Cyclone III Device Handbook

Volume 1

101 Innovation Drive
San Jose, CA 95134
www.altera.com

CIII5V1-4.2

Document last updated for Altera Complete Design Suite version: 12.0
Document publication date: August 2012

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

ISO 9001:2008 Registered

Contents

Chapter Revision Dates ix

Section I. Device Core

Chapter 1. Cyclone III Device Family Overview

Cyclone III Device Family Features 1–1
Lowest Power FPGAs 1–1
Design Security Feature 1–2
Increased System Integration 1–2
Cyclone III Device Family Architecture 1–6
Logic Elements and Logic Array Blocks 1–6
Memory Blocks 1–7
Embedded Multipliers and Digital Signal Processing Support 1–7
Clock Networks and PLLs 1–8
I/O Features 1–8
High-Speed Differential Interfaces 1–8
Auto-Calibrating External Memory Interfaces 1–9
Support for Industry-Standard Embedded Processors 1–9
Hot Socketing and Power-On-Reset 1–9
SEU Mitigation 1–10
JTAG Boundary Scan Testing 1–10
Quartus II Software Support 1–10
Configuration 1–11
Remote System Upgrades 1–11
Design Security (Cyclone III LS Devices Only) 1–11
Reference and Ordering Information 1–12
Document Revision History 1–13

Chapter 2. Logic Elements and Logic Array Blocks in the Cyclone III Device Family

Logic Elements 2–1
LE Features 2–2
LE Operating Modes 2–3
Normal Mode 2–3
Arithmetic Mode 2–4
Logic Array Blocks 2–4
Topology 2–4
LAB Interconnects 2–5
LAB Control Signals 2–6
Document Revision History 2–7

Chapter 3. Memory Blocks in the Cyclone III Device Family

Overview 3–1
Control Signals 3–3
Parity Bit Support 3–3
Byte Enable Support 3–4
Packed Mode Support 3–5
Address Clock Enable Support 3–5
Mixed-Width Support 3–6

Asynchronous Clear . . . 3–7
Memory Modes . . . 3–7
Single-Port Mode . . . 3–8
Simple Dual-Port Mode . . . 3–9
True Dual-Port Mode . . . 3–11
Shift Register Mode . . . 3–12
ROM Mode . . . 3–13
FIFO Buffer Mode . . . 3–13
Clocking Modes . . . 3–14
Independent Clock Mode . . . 3–14
I/O Clock Mode . . . 3–14
Read or Write Clock Mode . . . 3–15
Single-Clock Mode . . . 3–15
Design Considerations . . . 3–15
Read-During-Write Operations . . . 3–15
Same-Port Read-During-Write Mode . . . 3–16
Mixed-Port Read-During-Write Mode . . . 3–16
Conflict Resolution . . . 3–17
Power-Up Conditions and Memory Initialization . . . 3–18
Power Management . . . 3–18
Document Revision History . . . 3–18

Chapter 4. Embedded Multipliers in the Cyclone III Device Family

Embedded Multiplier Block Overview . . . 4–2
Architecture . . . 4–3
Input Registers . . . 4–4
Multiplier Stage . . . 4–4
Output Registers . . . 4–5
Operational Modes . . . 4–5
18-Bit Multipliers . . . 4–6
9-Bit Multipliers . . . 4–6
Document Revision History . . . 4–8

Chapter 5. Clock Networks and PLLs in the Cyclone III Device Family

Clock Networks . . . 5–1
GCLK Network . . . 5–1
Clock Control Block . . . 5–4
GCLK Network Clock Source Generation . . . 5–6
GCLK Network Power Down . . . 5–7
clkena Signals . . . 5–8
PLLs in the Cyclone III Device Family . . . 5–9
Cyclone III Device Family PLL Hardware Overview . . . 5–10
External Clock Outputs . . . 5–10
Clock Feedback Modes . . . 5–11
Source-Synchronous Mode . . . 5–12
No Compensation Mode . . . 5–13
Normal Mode . . . 5–13
Zero Delay Buffer Mode . . . 5–14
Hardware Features . . . 5–15
Clock Multiplication and Division . . . 5–15
Post-Scale Counter Cascading . . . 5–16
Programmable Duty Cycle . . . 5–16
PLL Control Signals . . . 5–17

pfdena . . . 5–17
areset . . . 5–17
locked . . . 5–17
Clock Switchover . . . 5–18
Automatic Clock Switchover . . . 5–18
Manual Override . . . 5–20
Manual Clock Switchover . . . 5–21
Guidelines . . . 5–21
Programmable Bandwidth . . . 5–22
Phase Shift Implementation . . . 5–22
PLL Cascading . . . 5–24
PLL Reconfiguration . . . 5–26
PLL Reconfiguration Hardware Implementation . . . 5–26
Post-Scale Counters (C0 to C4) . . . 5–28
Scan Chain Description . . . 5–29
Charge Pump and Loop Filter . . . 5–30
Bypassing PLL Counter . . . 5–31
Dynamic Phase Shifting . . . 5–31
Spread-Spectrum Clocking . . . 5–33
PLL Specifications . . . 5–33
Document Revision History . . . 5–34

Section II. I/O Interfaces

Chapter 6. I/O Features in the Cyclone III Device Family

Cyclone III Device Family I/O Elements . . . 6–1
I/O Element Features . . . 6–2
Programmable Current Strength . . . 6–2
Slew Rate Control . . . 6–4
Open-Drain Output . . . 6–4
Bus Hold . . . 6–4
Programmable Pull-Up Resistor . . . 6–5
Programmable Delay . . . 6–5
PCI-Clamp Diode . . . 6–6
LVDS Transmitter Programmable Pre-Emphasis . . . 6–6
OCT Support . . . 6–7
On-Chip Series Termination with Calibration . . . 6–8
On-Chip Series Termination Without Calibration . . . 6–10
I/O Standards . . . 6–11
Termination Scheme for I/O Standards . . . 6–13
Voltage-Referenced I/O Standard Termination . . . 6–14
Differential I/O Standard Termination . . . 6–15
I/O Banks . . . 6–16
High-Speed Differential Interfaces . . . 6–20
External Memory Interfacing . . . 6–20
Pad Placement and DC Guidelines . . . 6–21
Pad Placement . . . 6–21
DC Guidelines . . . 6–21
Document Revision History . . . 6–21

Chapter 7. High-Speed Differential Interfaces in the Cyclone III Device Family

High-Speed I/O Interface . . . 7–1
High-Speed I/O Standards Support . . . 7–7
LVDS I/O Standard Support in the Cyclone III Device Family . . . 7–7
Designing with LVDS . . . 7–8
BLVDS I/O Standard Support in the Cyclone III Device Family . . . 7–8
Designing with BLVDS . . . 7–9
RSDS, Mini-LVDS, and PPDS I/O Standard Support in the Cyclone III Device Family . . . 7–10
Designing with RSDS, Mini-LVDS, and PPDS . . . 7–10
LVPECL I/O Support in the Cyclone III Device Family . . . 7–12
Differential SSTL I/O Standard Support in the Cyclone III Device Family . . . 7–13
Differential HSTL I/O Standard Support in the Cyclone III Device Family . . . 7–14
True Output Buffer Feature . . . 7–15
Programmable Pre-Emphasis . . . 7–15
High-Speed I/O Timing . . . 7–16
Design Guidelines . . . 7–17
Differential Pad Placement Guidelines . . . 7–17
Board Design Considerations . . . 7–17
Software Overview . . . 7–18
Document Revision History . . . 7–19

Chapter 8. External Memory Interfaces in the Cyclone III Device Family

Cyclone III Device Family Memory Interfaces Pin Support . . . 8–2
Data and Data Clock/Strobe Pins . . . 8–2
Optional Parity, DM, and Error Correction Coding Pins . . . 8–10
Address and Control/Command Pins . . . 8–10
Memory Clock Pins . . . 8–10
Cyclone III Device Family Memory Interfaces Features . . . 8–11
DDR Input Registers . . . 8–11
DDR Output Registers . . . 8–12
OCT . . . 8–13
PLL . . . 8–13
Document Revision History . . . 8–14

Section III. System Integration

Chapter 9. Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family

Configuration Features . . . 9–2
Configuration Data Decompression . . . 9–3
Configuration Requirement . . . 9–5
POR Circuit . . . 9–5
Configuration File Size . . . 9–7
Configuration and JTAG Pin I/O Requirements . . . 9–7
Configuration Process . . . 9–8
Power Up . . . 9–8
Reset . . . 9–8
Configuration . . . 9–9
Configuration Error . . . 9–9
Initialization . . . 9–10
User Mode . . . 9–10
Configuration Scheme . . . 9–11
AS Configuration (Serial Configuration Devices) . . . 9–12

Single-Device AS Configuration . . . 9–13
Multi-Device AS Configuration . . . 9–14
Configuring Multiple Cyclone III Device Family with the Same Design . . . 9–16
Guidelines for Connecting Serial Configuration Device to Cyclone III Device Family on AS Interface . . . 9–20
Programming Serial Configuration Devices . . . 9–21
AP Configuration (Supported Flash Memories) . . . 9–23
AP Configuration Supported Flash Memory . . . 9–24
Single-Device AP Configuration . . . 9–25
Multi-Device AP Configuration . . . 9–27
Byte-Wide Multi-Device AP Configuration . . . 9–28
Word-Wide Multi-Device AP Configuration . . . 9–29
Guidelines for Connecting Parallel Flash to Cyclone III Devices for the AP Interface . . . 9–30
Configuring With Multiple Bus Masters . . . 9–30
Estimating the AP Configuration Time . . . 9–32
Programming Parallel Flash Memories . . . 9–33
PS Configuration . . . 9–34
PS Configuration Using an External Host . . . 9–35
PS Configuration Timing . . . 9–38
PS Configuration Using a Download Cable . . . 9–40
FPP Configuration . . . 9–42
FPP Configuration Using an External Host . . . 9–43
FPP Configuration Timing . . . 9–47
JTAG Configuration . . . 9–48
Configuring Cyclone III Device Family with Jam STAPL . . . 9–55
Configuring Cyclone III Device Family with the JRunner Software Driver . . . 9–56
Combining JTAG and AS Configuration Schemes . . . 9–56
Programming Serial Configuration Devices In-System Using the JTAG Interface . . . 9–58
JTAG Instructions . . . 9–60
Changing the Start Boot Address of the AP Flash . . . 9–64
Device Configuration Pins . . . 9–64
Design Security . . . 9–70
Cyclone III LS Design Security Protection . . . 9–70
Security Against Copying . . . 9–71
Security Against Reverse Engineering . . . 9–71
Security Against Tampering . . . 9–71
AES Decryption Block . . . 9–71
Key Storage . . . 9–71
Cyclone III LS Design Security Solution . . . 9–72
Available Security Modes . . . 9–73
Volatile Key . . . 9–73
No Key Operation . . . 9–74
FACTORY Mode . . . 9–74
Remote System Upgrade . . . 9–74
Functional Description . . . 9–75
Enabling Remote Update . . . 9–76
Configuration Image Types . . . 9–77
Remote System Upgrade Mode . . . 9–77
Remote Update Mode . . . 9–77
Dedicated Remote System Upgrade Circuitry . . . 9–80
Remote System Upgrade Registers . . . 9–81
Remote System Upgrade State Machine . . . 9–84
User Watchdog Timer . . . 9–85
Quartus II Software Support . . . 9–86

Document Revision History 9–86

Chapter 10. Hot-Socketing and Power-On Reset in the Cyclone III Device Family
Hot-Socketing Specifications 10–1
Devices Driven Before Power-Up 10–2
I/O Pins Remain Tristated During Power-Up 10–2
Hot-Socketing Feature Implementation 10–3
POR Circuitry 10–3
Document Revision History 10–4

Chapter 11. SEU Mitigation in the Cyclone III Device Family
Error Detection Fundamentals 11–1
Configuration Error Detection 11–2
User Mode Error Detection 11–2
Automated SEU Detection 11–3
CRC_ERROR Pin 11–3
Error Detection Block 11–4
Error Detection Registers 11–4
Error Detection Timing 11–5
Software Support 11–7
Accessing Error Detection Block Through User Logic 11–8
Recovering from CRC Errors 11–10
Document Revision History 11–10

Chapter 12. IEEE 1149.1 (JTAG) Boundary-Scan Testing for the Cyclone III Device Family
IEEE Std. 1149.1 BST Architecture 12–1
IEEE Std. 1149.1 BST Operation Control 12–2
I/O Voltage Support in a JTAG Chain 12–5
Guidelines for IEEE Std. 1149.1 BST 12–6
Boundary-Scan Description Language Support 12–7
Document Revision History 12–7

Additional Information
How to Contact Altera Info–1
Typographic Conventions Info–1

Chapter Revision Dates

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

Chapter 1. Cyclone III Device Family Overview
Revised: *July 2012*
Part Number: *CIII51001-2.4*

Chapter 2. Logic Elements and Logic Array Blocks in the Cyclone III Device Family
Revised: *December 2011*
Part Number: *CIII51002-2.3*

Chapter 3. Memory Blocks in the Cyclone III Device Family
Revised: *December 2011*
Part Number: *CIII51004-2.3*

Chapter 4. Embedded Multipliers in the Cyclone III Device Family
Revised: *December 2011*
Part Number: *CIII51005-2.3*

Chapter 5. Clock Networks and PLLs in the Cyclone III Device Family
Revised: *July 2012*
Part Number: *CIII51006-4.1*

Chapter 6. I/O Features in the Cyclone III Device Family
Revised: *July 2012*
Part Number: *CIII51007-3.4*

Chapter 7. High-Speed Differential Interfaces in the Cyclone III Device Family
Revised: *December 2011*
Part Number: *CIII51008-4.0*

Chapter 8. External Memory Interfaces in the Cyclone III Device Family
Revised: *July 2012*
Part Number: *CIII51009-3.1*

Chapter 9. Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family
Revised: *August 2012*
Part Number: *CIII51016-2.2*

Chapter 10. Hot-Socketing and Power-On Reset in the Cyclone III Device Family
Revised: *July 2012*
Part Number: *CIII51011-3.4*

Chapter 11. SEU Mitigation in the Cyclone III Device Family
Revised: *December 2011*
Part Number: *CIII51013-2.3*

Chapter 12. IEEE 1149.1 (JTAG) Boundary-Scan Testing for the Cyclone III Device Family
Revised: *December 2011*
Part Number: *CIII51014-2.3*

Section I. Device Core

This section provides a complete overview of all features relating to the Cyclone® III device family.

This section includes the following chapters:

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

For information about the revision history for chapters in this section, refer to "Document Revision History" in each individual chapter.

1. Cyclone III Device Family Overview

CIII51001-2.4

Cyclone® III device family offers a unique combination of high functionality, low power and low cost. Based on Taiwan Semiconductor Manufacturing Company (TSMC) low-power (LP) process technology, silicon optimizations and software features to minimize power consumption, Cyclone III device family provides the ideal solution for your high-volume, low-power, and cost-sensitive applications. To address the unique design needs, Cyclone III device family offers the following two variants:

- Cyclone III—lowest power, high functionality with the lowest cost
- Cyclone III LS—lowest power FPGAs with security

With densities ranging from about 5,000 to 200,000 logic elements (LEs) and 0.5 Megabits (Mb) to 8 Mb of memory for less than ¼ watt of static power consumption, Cyclone III device family makes it easier for you to meet your power budget. Cyclone III LS devices are the first to implement a suite of security features at the silicon, software, and intellectual property (IP) level on a low-power and high-functionality FPGA platform. This suite of security features protects the IP from tampering, reverse engineering and cloning. In addition, Cyclone III LS devices support design separation which enables you to introduce redundancy in a single chip to reduce size, weight, and power of your application.

This chapter contains the following sections:

- "Cyclone III Device Family Features" on page 1–1
- "Cyclone III Device Family Architecture" on page 1–6
- "Reference and Ordering Information" on page 1–12

Cyclone III Device Family Features

Cyclone III device family offers the following features:

Lowest Power FPGAs

- Lowest power consumption with TSMC low-power process technology and Altera® power-aware design flow
- Low-power operation offers the following benefits:
  - Extended battery life for portable and handheld applications
  - Reduced or eliminated cooling system costs
  - Operation in thermally-challenged environments
- Hot-socketing operation support

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

ISO 9001:2008 Registered

Design Security Feature

Cyclone III LS devices offer the following design security features:

- Configuration security using advanced encryption standard (AES) with 256-bit volatile key
- Routing architecture optimized for design separation flow with the Quartus® II software
  - Design separation flow achieves both physical and functional isolation between design partitions
- Ability to disable external JTAG port
- Error Detection (ED) Cycle Indicator to core
  - Provides a pass or fail indicator at every ED cycle
  - Provides visibility over intentional or unintentional change of configuration random access memory (CRAM) bits
- Ability to perform zeroization to clear contents of the FPGA logic, CRAM, embedded memory, and AES key
- Internal oscillator enables system monitor and health check capabilities

Increased System Integration

- High memory-to-logic and multiplier-to-logic ratio
- High I/O count, low-and mid-range density devices for user I/O constrained applications
  - Adjustable I/O slew rates to improve signal integrity
  - Supports I/O standards such as LVTTL, LVCMOS, SSTL, HSTL, PCI, PCI-X, LVPECL, bus LVDS (BLVDS), LVDS, mini-LVDS, RSDS, and PPDS
  - Supports the multi-value on-chip termination (OCT) calibration feature to eliminate variations over process, voltage, and temperature (PVT)
- Four phase-locked loops (PLLs) per device provide robust clock management and synthesis for device clock management, external system clock management, and I/O interfaces
  - Five outputs per PLL
  - Cascadable to save I/Os, ease PCB routing, and reduce jitter
  - Dynamically reconfigurable to change phase shift, frequency multiplication or division, or both, and input frequency in the system without reconfiguring the device
- Remote system upgrade without the aid of an external controller
- Dedicated cyclical redundancy code checker circuitry to detect single-event upset (SEU) issues
- Nios® II embedded processor for Cyclone III device family, offering low cost and custom-fit embedded processing solutions

- Wide collection of pre-built and verified IP cores from Altera and Altera Megafunction Partners Program (AMPP) partners
- Supports high-speed external memory interfaces such as DDR, DDR2, SDR SDRAM, and QDRII SRAM
  - Auto-calibrating PHY feature eases the timing closure process and eliminates variations with PVT for DDR, DDR2, and QDRII SRAM interfaces

Cyclone III device family supports vertical migration that allows you to migrate your device to other devices with the same dedicated pins, configuration pins, and power pins for a given package-across device densities. This allows you to optimize device density and cost as your design evolves.

Table 1–1 lists Cyclone III device family features.

Table 1–1. Cyclone III Device Family Features

Family	Device	Logic Elements	Number of M9K Blocks	Total RAM Bits	18 x 18 Multipliers	PLLs	Global Clock Networks	Maximum User I/Os
Cyclone III	EP3C5	5,136	46	423,936	23	2	10	182
	EP3C10	10,320	46	423,936	23	2	10	182
	EP3C16	15,408	56	516,096	56	4	20	346
	EP3C25	24,624	66	608,256	66	4	20	215
	EP3C40	39,600	126	1,161,216	126	4	20	535
	EP3C55	55,856	260	2,396,160	156	4	20	377
	EP3C80	81,264	305	2,810,880	244	4	20	429
	EP3C120	119,088	432	3,981,312	288	4	20	531
Cyclone III LS	EP3CLS70	70,208	333	3,068,928	200	4	20	429
	EP3CLS100	100,448	483	4,451,328	276	4	20	429
	EP3CLS150	150,848	666	6,137,856	320	4	20	429
	EP3CLS200	198,464	891	8,211,456	396	4	20	429

Table 1–2 lists Cyclone III device family package options, I/O pins, and differential channel counts.

Table 1–2. Cyclone III Device Family Package Options, I/O pin and Differential Channel Counts *(1), (2), (3), (4), (5)*

Family	Package	E144 *(7)*	M164	P240	F256	U256	F324	F484	U484	F780
Cyclone III *(8)*	EP3C5	94, 22	106, 28	—	182, 68	182, 68	—	—	—	—
	EP3C10	94, 22	106, 28	—	182, 68	182, 68	—	—	—	—
	EP3C16	84, 19	92, 23	160, 47	168, 55	168, 55	—	346, 140	346, 140	—
	EP3C25	82, 18	—	148, 43	156, 54	156, 54	215, 83	—	—	—
	EP3C40	—	—	128, 26	—	—	195, 61	331, 127	331, 127	535, 227 *(6)*
	EP3C55	—	—	—	—	—	—	327, 135	327, 135	377, 163
	EP3C80	—	—	—	—	—	—	295, 113	295, 113	429, 181
	EP3C120	—	—	—	—	—	—	283, 106	—	531, 233
Cyclone III LS	EP3CLS70	—	—	—	—	—	—	294, 113	294, 113	429, 181
	EP3CLS100	—	—	—	—	—	—	294, 113	294, 113	429, 181
	EP3CLS150	—	—	—	—	—	—	226, 87	—	429, 181
	EP3CLS200	—	—	—	—	—	—	226, 87	—	429, 181

Notes to Table 1–2:

(1) For each device package, the first number indicates the number of the I/O pin; the second number indicates the differential channel count.

(2) For more information about device packaging specifications, refer to the Cyclone III Package and Thermal Resistance webpage.

(3) The I/O pin numbers are the maximum I/O counts (including clock input pins) supported by the device package combination and can be affected by the configuration scheme selected for the device.

(4) All packages are available in lead-free and leaded options.

(5) Vertical migration is not supported between Cyclone III and Cyclone III LS devices.

(6) The EP3C40 device in the F780 package supports restricted vertical migration. Maximum user I/Os are restricted to 510 I/Os if you enable migration to the EP3C120 and are using voltage referenced I/O standards. If you are not using voltage referenced I/O standards, you can increase the maximum number of I/Os.

(7) 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 on your PCB. Use this exposed pad for electrical connectivity and not for thermal purposes.

(8) All Cyclone III device UBGA packages are supported by the Quartus II software version 7.1 SP1 and later, with the exception of the UBGA packages of EP3C16, which are supported by the Quartus II software version 7.2.

Table 1–3 lists Cyclone III device family package sizes.

Table 1–3. Cyclone III Device Family Package Sizes

Family	Package	Pitch (mm)	Nominal Area (mm^2)	Length x Width (mm × mm)	Height (mm)
Cyclone III	E144	0.5	484	22 × 22	1.60
	M164	0.5	64	8 × 8	1.40
	P240	0.5	1197	34.6 × 34.6	4.10
	F256	1.0	289	17 × 17	1.55
	U256	0.8	196	14 × 14	2.20
	F324	1.0	361	19 × 19	2.20
	F484	1.0	529	23 × 23	2.60
	U484	0.8	361	19 × 19	2.20
	F780	1.0	841	29 × 29	2.60
Cyclone III LS	F484	1.0	529	23 × 23	2.60
	U484	0.8	361	19 × 19	2.20
	F780	1.0	841	29 × 29	2.60

Table 1–4 lists Cyclone III device family speed grades.

Table 1–4. Cyclone III Device Family Speed Grades (Part 1 of 2)

Family	Device	E144	M164	P240	F256	U256	F324	F484	U484	F780
Cyclone III	EP3C5	C7, C8, I7, A7	C7, C8, I7	—	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	—	—	—	—
	EP3C10	C7, C8, I7, A7	C7, C8, I7	—	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	—	—	—	—
	EP3C16	C7, C8, I7, A7	C7, C8, I7	C8	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	—	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	—
	EP3C25	C7, C8, I7, A7	—	C8	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	—	—	—
	EP3C40	—	—	C8	—	—	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	C6, C7, C8, I7, A7	C6, C7, C8, I7
	EP3C55	—	—	—	—	—	—	C6, C7, C8, I7	C6, C7, C8, I7	C6, C7, C8, I7
	EP3C80	—	—	—	—	—	—	C6, C7, C8, I7	C6, C7, C8, I7	C6, C7, C8, I7
	EP3C120	—	—	—	—	—	—	C7, C8, I7	—	C7, C8, I7

Table 1–4. Cyclone III Device Family Speed Grades (Part 2 of 2)

Family	Device	E144	M164	P240	F256	U256	F324	F484	U484	F780
Cyclone III LS	EP3CLS70	—	—	—	—	—	—	C7, C8, I7	C7, C8, I7	C7, C8, I7
	EP3CLS100	—	—	—	—	—	—	C7, C8, I7	C7, C8, I7	C7, C8, I7
	EP3CLS150	—	—	—	—	—	—	C7, C8, I7	—	C7, C8, I7
	EP3CLS200	—	—	—	—	—	—	C7, C8, I7	—	C7, C8, I7

Table 1–5 lists Cyclone III device family configuration schemes.

Table 1–5. Cyclone III Device Family Configuration Schemes

Configuration Scheme	Cyclone III	Cyclone III LS
Active serial (AS)	✓	✓
Active parallel (AP)	✓	—
Passive serial (PS)	✓	✓
Fast passive parallel (FPP)	✓	✓
Joint Test Action Group (JTAG)	✓	✓

Cyclone III Device Family Architecture

Cyclone III device family includes a customer-defined feature set that is optimized for portable applications and offers a wide range of density, memory, embedded multiplier, and I/O options. Cyclone III device family supports numerous external memory interfaces and I/O protocols that are common in high-volume applications. The Quartus II software features and parameterizable IP cores make it easier for you to use the Cyclone III device family interfaces and protocols.

The following sections provide an overview of the Cyclone III device family features.

Logic Elements and Logic Array Blocks

The logic array block (LAB) consists of 16 logic elements and a LAB-wide control block. An LE is the smallest unit of logic in the Cyclone III device family architecture. Each LE has four inputs, a four-input look-up table (LUT), a register, and output logic. The four-input LUT is a function generator that can implement any function with four variables.

For more information about LEs and LABs, refer to the *Logic Elements and Logic Array Blocks in the Cyclone III Device Family* chapter.

Memory Blocks

Each M9K memory block of the Cyclone III device family provides nine Kbits of on-chip memory capable of operating at up to 315 MHz for Cyclone III devices and up to 274 MHz for Cyclone III LS devices. The embedded memory structure consists of M9K memory blocks columns that you can configure as RAM, first-in first-out (FIFO) buffers, or ROM. The Cyclone III device family memory blocks are optimized for applications such as high throughout packet processing, embedded processor program, and embedded data storage.

The Quartus II software allows you to take advantage of the M9K memory blocks by instantiating memory using a dedicated megafunction wizard or by inferring memory directly from the VHDL or Verilog source code.

M9K memory blocks support single-port, simple dual-port, and true dual-port operation modes. Single-port mode and simple dual-port mode are supported for all port widths with a configuration of ×1, ×2, ×4, ×8, ×9, ×16, ×18, ×32, and ×36. True dual-port is supported in port widths with a configuration of ×1, ×2, ×4, ×8, ×9, ×16, and ×18.

For more information about memory blocks, refer to the *Memory Blocks in the Cyclone III Device Family* chapter.

Embedded Multipliers and Digital Signal Processing Support

Cyclone III devices support up to 288 embedded multiplier blocks and Cyclone III LS devices support up to 396 embedded multiplier blocks. Each block supports one individual 18 × 18-bit multiplier or two individual 9 × 9-bit multipliers.

The Quartus II software includes megafunctions that are used to control the operation mode of the embedded multiplier blocks based on user parameter settings. Multipliers can also be inferred directly from the VHDL or Verilog source code. In addition to embedded multipliers, Cyclone III device family includes a combination of on-chip resources and external interfaces, making them ideal for increasing performance, reducing system cost, and lowering the power consumption of digital signal processing (DSP) systems. You can use Cyclone III device family alone or as DSP device co-processors to improve price-to-performance ratios of DSP systems.

The Cyclone III device family DSP system design support includes the following features:

- DSP IP cores:
  - Common DSP processing functions such as finite impulse response (FIR), fast Fourier transform (FFT), and numerically controlled oscillator (NCO) functions
  - Suites of common video and image processing functions
- Complete reference designs for end-market applications
- DSP Builder interface tool between the Quartus II software and the MathWorks Simulink and MATLAB design environments
- DSP development kits

For more information about embedded multipliers and digital signal processing support, refer to the *Embedded Multipliers in Cyclone III Devices* chapter.

Clock Networks and PLLs

Cyclone III device family includes 20 global clock networks. You can drive global clock signals from dedicated clock pins, dual-purpose clock pins, user logic, and PLLs. Cyclone III device family includes up to four PLLs with five outputs per PLL to provide robust clock management and synthesis. You can use PLLs for device clock management, external system clock management, and I/O interfaces.

You can dynamically reconfigure the Cyclone III device family PLLs to enable auto-calibration of external memory interfaces while the device is in operation. This feature enables the support of multiple input source frequencies and corresponding multiplication, division, and phase shift requirements. PLLs in Cyclone III device family may be cascaded to generate up to ten internal clocks and two external clocks on output pins from a single external clock source.

For more PLL specifications and information, refer to the *Cyclone III Device Data Sheet*, *Cyclone III LS Device Data Sheet*, and *Clock Networks and PLLs in the Cyclone III Device Family* chapters.

I/O Features

Cyclone III device family has eight I/O banks. All I/O banks support single-ended and differential I/O standards listed in Table 1–6.

Table 1–6. Cyclone III Device Family I/O Standards Support

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 Cyclone III device family I/O also 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 III device family supports calibrated on-chip series termination (R_S OCT) or driver impedance matching (Rs) for single-ended I/O standards, with one OCT calibration block per side.

For more information, refer to the *I/O Features in the Cyclone III Device Family* chapter.

High-Speed Differential Interfaces

Cyclone III device family supports high-speed differential interfaces such as BLVDS, LVDS, mini-LVDS, RSDS, and PPDS. These high-speed I/O standards in Cyclone III device family provide high data throughput using a relatively small number of I/O pins and are ideal for low-cost applications. Dedicated differential output drivers on the left and right I/O banks can send data rates at up to 875 Mbps for Cyclone III devices and up to 740 Mbps for Cyclone III LS devices, without the need for external resistors. This saves board space or simplifies PCB routing. Top and bottom I/O banks support differential transmission (with the addition of an external resistor network) data rates at up to 640 Mbps for both Cyclone III and Cyclone III LS devices.

For more information, refer to the *High-Speed Differential Interfaces in the Cyclone III Device Family* chapter.

Auto-Calibrating External Memory Interfaces

Cyclone III device family supports common memory types such as DDR, DDR2, SDR SDRAM, and QDRII SRAM. DDR2 SDRAM memory interfaces support data rates up to 400 Mbps for Cyclone III devices and 333 Mbps for Cyclone III LS devices. Memory interfaces are supported on all sides of Cyclone III device family. Cyclone III device family has the OCT, DDR output registers, and 8-to-36-bit programmable DQ group widths features to enable rapid and robust implementation of different memory standards.

An auto-calibrating megafunction is available in the Quartus II software for DDR and QDR memory interface PHYs. This megafunction is optimized to take advantage of the Cyclone III device family I/O structure, simplify timing closure requirements, and take advantage of the Cyclone III device family PLL dynamic reconfiguration feature to calibrate PVT changes.

For more information, refer to the *External Memory Interfaces in the Cyclone III Device Family* chapter.

Support for Industry-Standard Embedded Processors

To quickly and easily create system-level designs using Cyclone III device family, you can select among the ×32-bit soft processor cores: Freescale®V1 Coldfire, ARM® Cortex M1, or Altera Nios® II, along with a library of 50 other IP blocks when using the system-on-a-programmable-chip (SOPC) Builder tool. SOPC Builder is an Altera Quartus II design tool that facilitates system-integration of IP blocks in an FPGA design. The SOPC Builder automatically generates interconnect logic and creates a testbench to verify functionality, saving valuable design time.

Cyclone III device family expands the peripheral set, memory, I/O, or performance of legacy embedded processors. Single or multiple Nios II embedded processors are designed into Cyclone III device family to provide additional co-processing power, or even replace legacy embedded processors in your system. Using the Cyclone III device family and Nios II together provide low-cost, high-performance embedded processing solutions, which in turn allow you to extend the life cycle of your product and improve time-to-market over standard product solutions.

Separate licensing of the Freescale and ARM embedded processors are required.

Hot Socketing and Power-On-Reset

Cyclone III device family features hot socketing (also known as hot plug-in or hot swap) and power sequencing support without the use of external devices. You can insert or remove a board populated with one or more Cyclone III device family during a system operation without causing undesirable effects to the running system bus or the board that was inserted into the system.

The hot socketing feature allows you to use FPGAs on PCBs that also contain a mixture of 3.3-V, 2.5-V, 1.8-V, 1.5-V, and 1.2-V devices. The Cyclone III device family hot socketing feature eliminates power-up sequence requirements for other devices on the board for proper FPGA operation.

For more information about hot socketing and power-on-reset, refer to the *Hot-Socketing and Power-on Reset in the Cyclone III Device Family* chapter.

SEU Mitigation

Cyclone III LS devices offer built-in error detection circuitry to detect data corruption due to soft errors in the CRAM cells. This feature allows CRAM contents to be read and verified to match a configuration-computed CRC value. The Quartus II software activates the built-in 32-bit CRC checker, which is part of the Cyclone III LS device.

For more information about SEU mitigation, refer to the *SEU Mitigation in the Cyclone III Device Family* chapter.

JTAG Boundary Scan Testing

Cyclone III device family supports the JTAG IEEE Std. 1149.1 specification. The boundary-scan test (BST) architecture offers the capability to test pin connections without using physical test probes and captures functional data while a device is operating normally. Boundary-scan cells in the Cyclone III device family can force signals onto pins or capture data from pins or from logic array signals. Forced test data is serially shifted into the boundary-scan cells. Captured data is serially shifted out and externally compared to expected results. In addition to BST, you can use the IEEE Std. 1149.1 controller for the Cyclone III LS device in-circuit reconfiguration (ICR).

For more information about JTAG boundary scan testing, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for the Cyclone III Device Family* chapter.

Quartus II Software Support

The Quartus II software is the leading design software for performance and productivity. It is the only complete design solution for CPLDs, FPGAs, and ASICs in the industry. The Quartus II software includes an integrated development environment to accelerate system-level design and seamless integration with leading third-party software tools and flows.

The Cyclone III LS devices provide both physical and functional separation between security critical design partitions. Cyclone III LS devices offer isolation between design partitions. This ensures that device errors do not propagate from one partition to another, whether unintentional or intentional. The Quartus II software design separation flow facilitates the creation of separation regions in Cyclone III LS devices by tightly controlling the routing in and between the LogicLock regions. For ease of use, the separation flow integrates in the existing incremental compilation flow.

For more information about the Quartus II software features, refer to the *Quartus II Handbook*.

Configuration

Cyclone III device family uses SRAM cells to store configuration data. Configuration data is downloaded to Cyclone III device family each time the device powers up. Low-cost configuration options include the Altera EPCS family serial flash devices as well as 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. Cyclone III device family supports the AS, PS, FPP, and JTAG configuration schemes. The AP configuration scheme is only supported in Cyclone III devices.

For more information about configuration, refer to the *Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family* chapter.

Remote System Upgrades

Cyclone III device family offers remote system upgrade without an external controller. The remote system upgrade capability in Cyclone III device family allows system upgrades from a remote location. Soft logic (either the Nios II embedded processor or user logic) implemented in Cyclone III device family can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to start a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, and can recover from an error condition by reverting to a safe configuration image. The dedicated circuitry also provides error status information. Cyclone III devices support remote system upgrade in the AS and AP configuration scheme. Cyclone III LS devices support remote system upgrade in the AS configuration scheme only.

For more information, refer to the *Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family* chapter.

Design Security (Cyclone III LS Devices Only)

Cyclone III LS devices offer design security features which play a vital role in the large and critical designs in the competitive military and commercial environments. Equipped with the configuration bit stream encryption and anti-tamper features, Cyclone III LS devices protect your designs from copying, reverse engineering and tampering. The configuration security of Cyclone III LS devices uses AES with 256-bit security key.

For more information, refer to the *Configuration, Design Security, and Remote System Upgrades in Cyclone III Device Family* chapter.

Reference and Ordering Information

Figure 1–1 and Figure 1–2 show the ordering codes for Cyclone III and Cyclone III LS devices.

Figure 1–1. Cyclone III Device Packaging Ordering Information

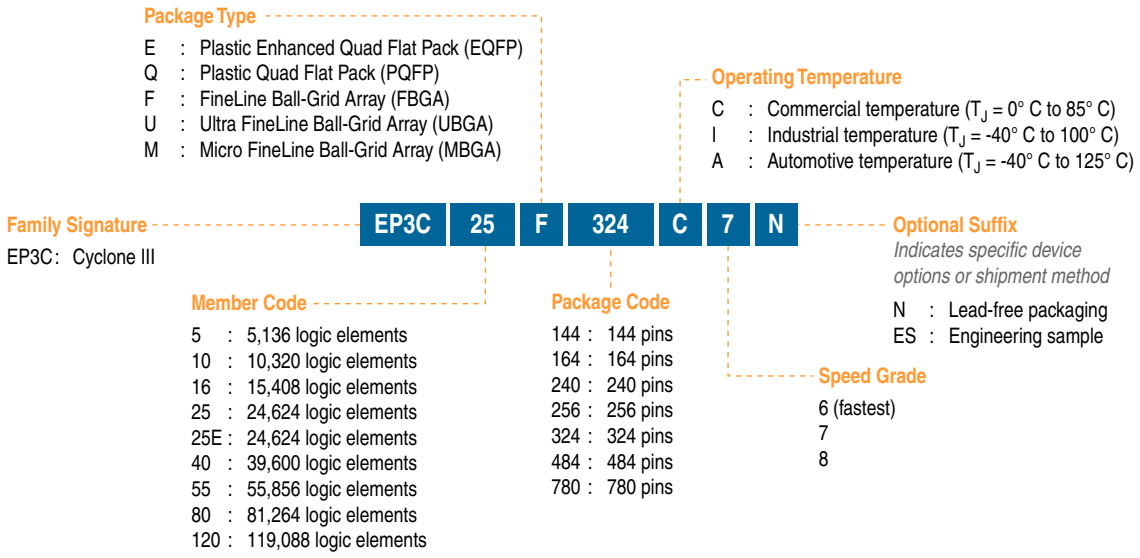
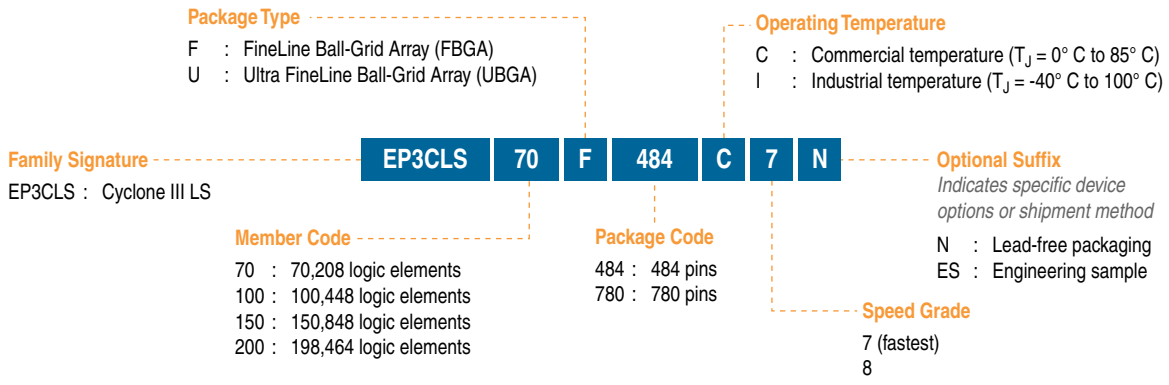


Figure 1–2. Cyclone III LS Device Packaging Ordering Information



Document Revision History

Table 1–7 lists the revision history for this document.

Table 1–7. Document Revision History

Date	Version	Changes
July 2012	2.4	Updated 484 pin package code in Figure 1–1.
December 2011	2.3	■ Updated Table 1–1 and Table 1–2. ■ Updated Figure 1–1 and Figure 1–2. ■ Updated hyperlinks. ■ Minor text edits.
December 2009	2.2	Minor text edits.
July 2009	2.1	Minor edit to the hyperlinks.
June 2009	2.0	■ Added Table 1–5. ■ Updated Table 1–1, Table 1–2, Table 1–3, and Table 1–4. ■ Updated “Introduction”, “Cyclone III Device Family Architecture”, “Embedded Multipliers and Digital Signal Processing Support ”, “Clock Networks and PLLs ”, “I/O Features ”, “High-Speed Differential Interfaces ”, “Auto-Calibrating External Memory Interfaces ”, “Quartus II Software Support”, “Configuration ”, and “Design Security (Cyclone III LS Devices Only)”. ■ Removed “Referenced Document” section.
October 2008	1.3	■ Updated “Increased System Integration” section. ■ Updated “Memory Blocks” section. ■ Updated chapter to new template.
May 2008	1.2	■ Added 164-pin Micro FineLine Ball-Grid Array (MBGA) details to Table 1–2, Table 1–3 and Table 1–4. ■ Updated Figure 1–2 with automotive temperature information. ■ Updated “Increased System Integration” section, Table 1–6, and “High-Speed Differential Interfaces” section with BLVDS information.
July 2007	1.1	■ Removed the text “Spansion” in “Increased System. ■ Integration” and “Configuration” sections. ■ Removed trademark symbol from “MultiTrack” in “MultiTrack Interconnect”. ■ Removed registered trademark symbol from “Simulink” and “MATLAB” from “Embedded Multipliers and Digital. ■ Signal Processing Support” section. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

2. Logic Elements and Logic Array Blocks in the Cyclone III Device Family

CIII51002-2.3

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 the Cyclone® III device family (Cyclone III and Cyclone III LS devices).

Logic Elements

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

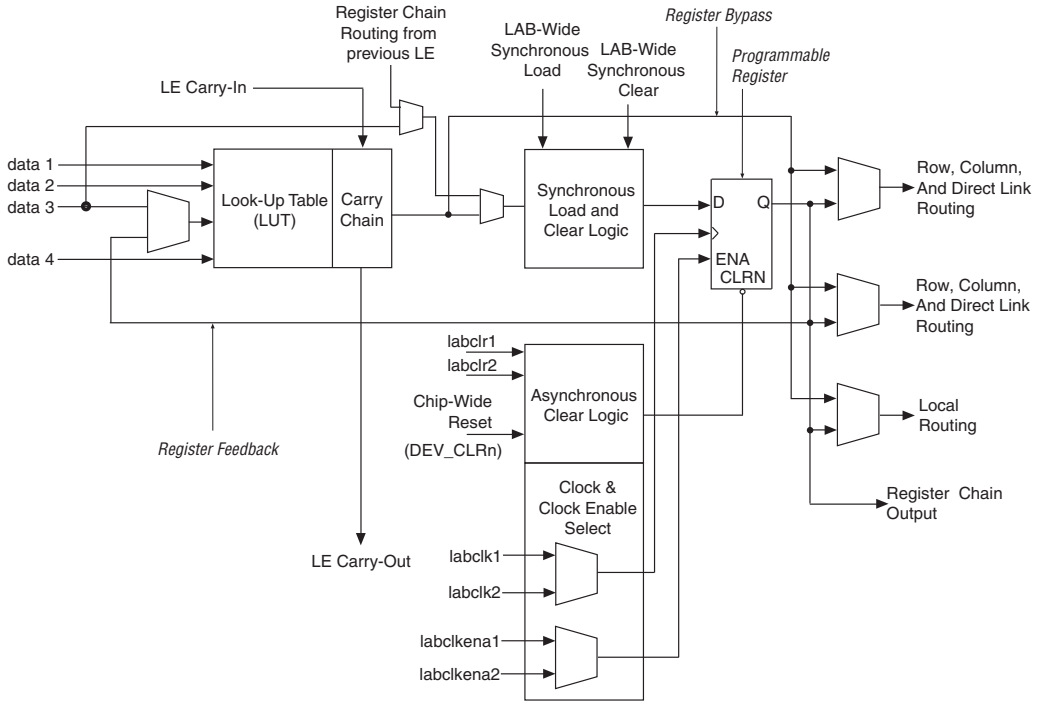
- A four-input look-up table (LUT), which can implement any function of four variables
- A programmable register
- A carry chain connection
- A register chain connection
- The ability to drive the following interconnects:
  - Local
  - Row
  - Column
  - Register chain
  - Direct link
- Register packing support
- Register feedback support

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Figure 2–1 shows the LEs for the Cyclone III device family.

Figure 2–1. Cyclone III Device Family 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 on 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 a 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 III device family LEs operate in the following modes:

- Normal mode
- Arithmetic mode

LE operating modes use LE resources differently. In each mode, there are six available inputs to the LE. These inputs include the four data inputs from the LAB local interconnect, the LE carry-in from the previous LE carry-chain, and the register chain connection. Each input is directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all LE modes.

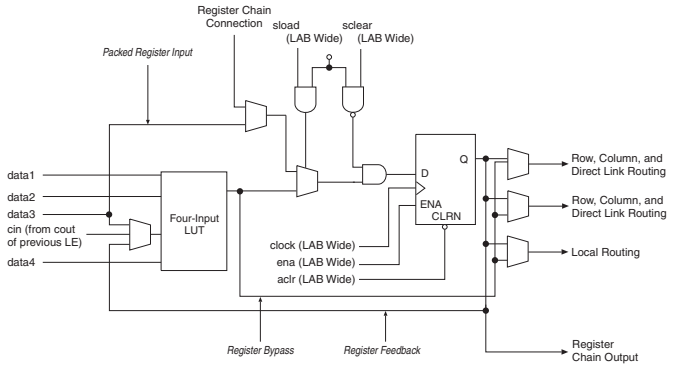
The Quartus® II software automatically chooses the appropriate mode for common functions, such as counters, adders, subtractors, and arithmetic functions, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions. You can also create special-purpose functions 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 III Device Family LEs in Normal Mode

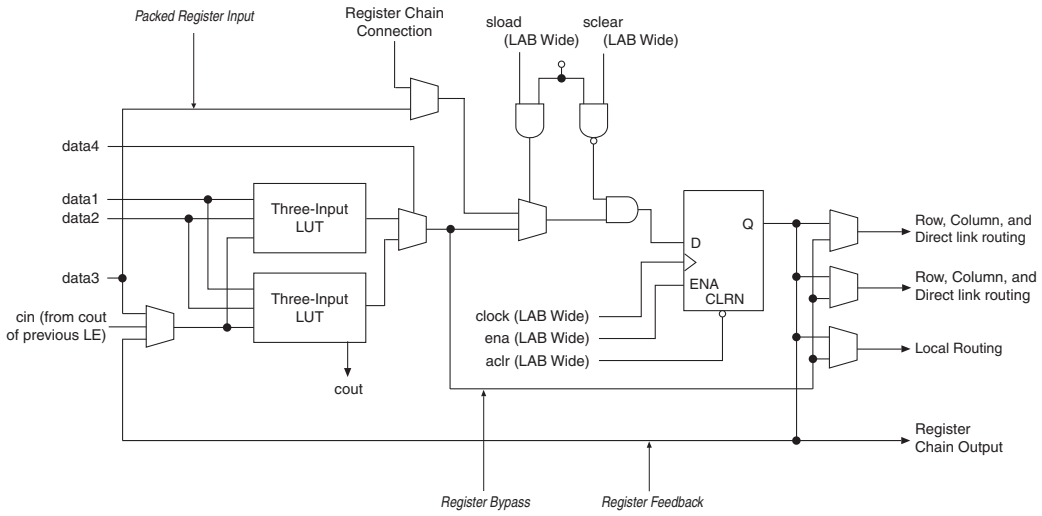


Arithmetic Mode

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 III Device Family 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 a 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:

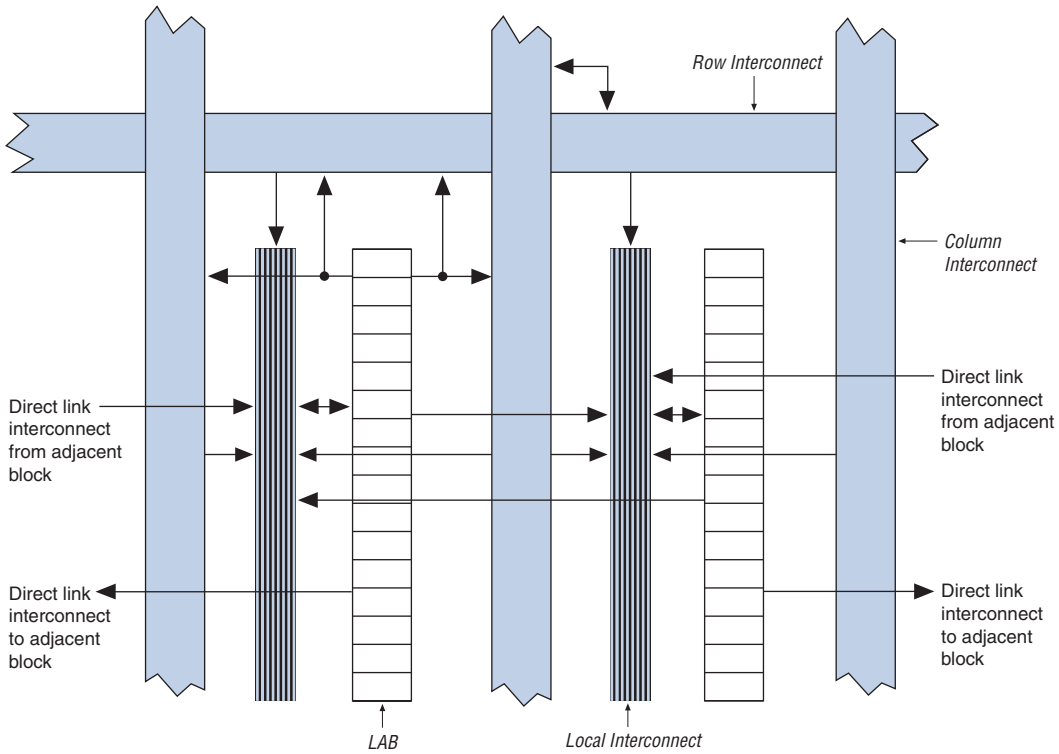
- 16 LEs

- 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 a LAB. The Quartus II Compiler places associated logic in a 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 the Cyclone III device family.

Figure 2–4. Cyclone III Device Family LAB Structure

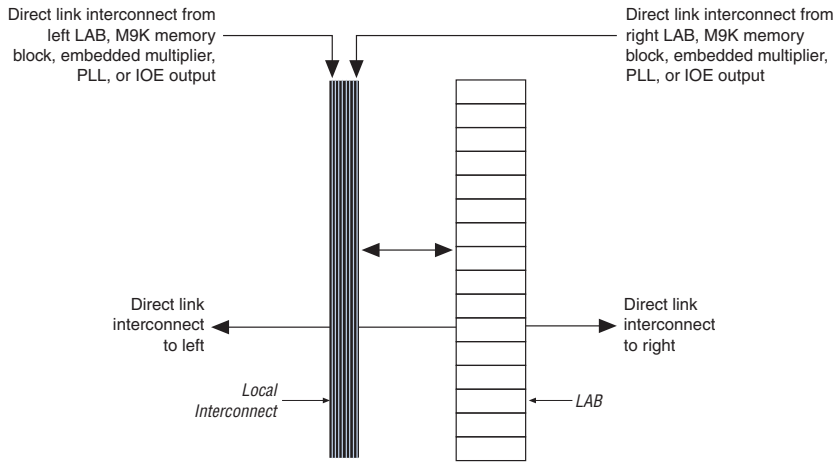


LAB Interconnects

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 III Device Family 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.

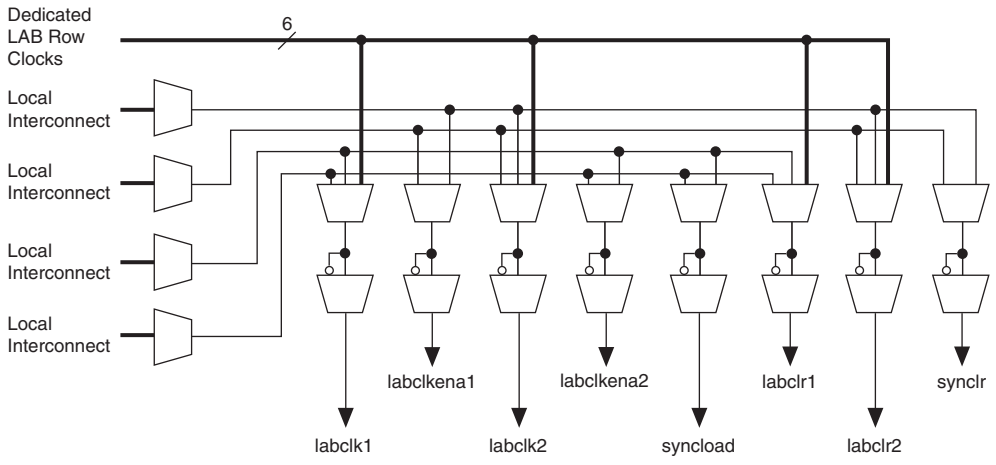
Synchronous clear and load signals are useful for implementing counters and other functions. The synchronous clear and synchronous load signals are LAB-wide signals that affect all registers in the LAB.

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 III Device Family 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. The Cyclone III device family only supports either a preset or asynchronous clear signal.

In addition to the clear port, the Cyclone III device family provides 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 lists the revision history for this document.

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

Date	Version	Changes
December 2011	2.3	Minor text edits.
December 2009	2.2	Minor changes to the text.
July 2009	2.1	Minor edit to the hyperlinks.
June 2009	2.0	Updated to include Cyclone III LS information ■ Updated chapter part number. ■ Updated “Introduction” on page 2–1. ■ Updated Figure 2–1 on page 2–2 and Figure 2–4 on page 2–5. ■ Updated “LAB Control Signals” on page 2–6.
October 2008	1.2	Updated chapter to new template.

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

Date	Version	Changes
July 2007	1.1	Removed trademark symbol from “MultiTrack” in “LAB Control Signals” section.
March 2007	1.0	Initial release.

3. Memory Blocks in the Cyclone III Device Family

CIII51004-2.3

The Cyclone® III device family (Cyclone III and Cyclone III LS devices) features embedded memory structures to address the on-chip memory needs of Altera® Cyclone III device family 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

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

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 the Cyclone III device family, refer to the *Cyclone III Device Family Overview* chapter.

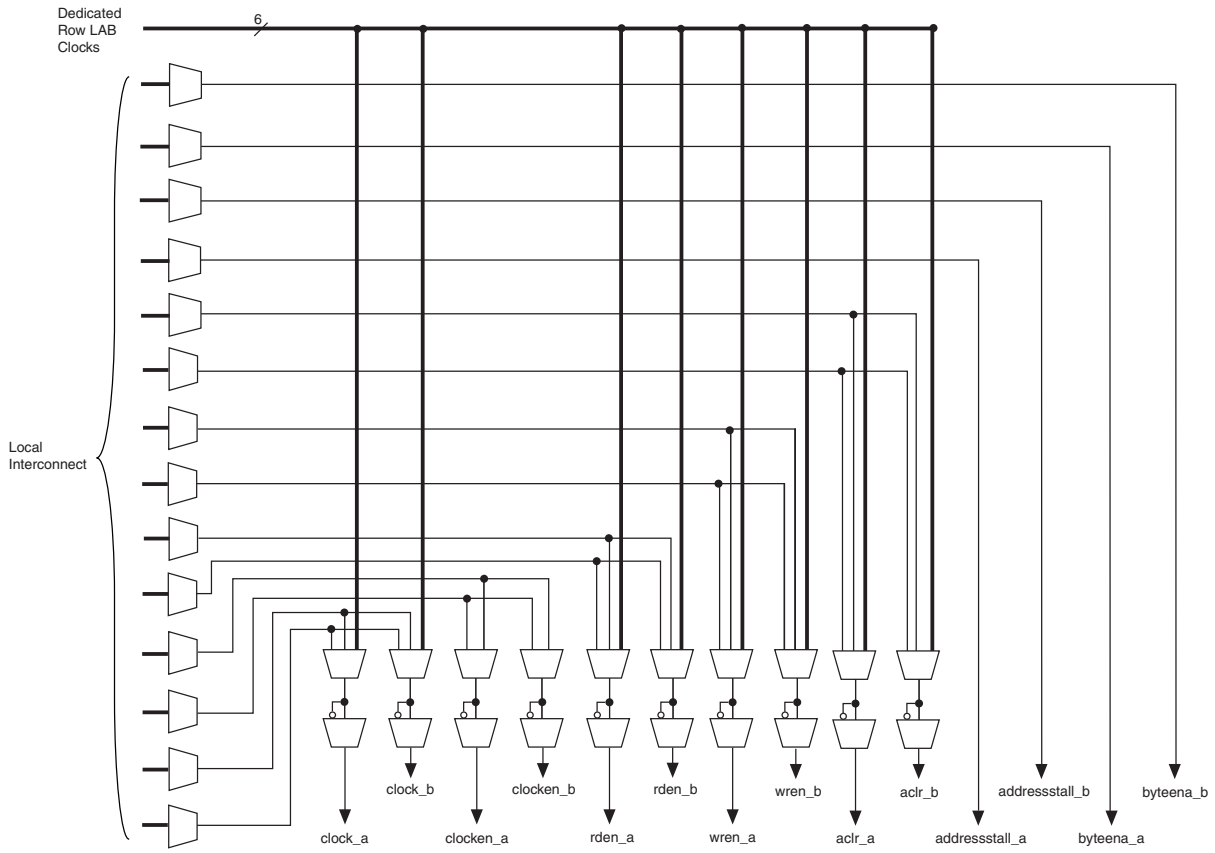
Control Signals

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.

Figure 3–1 shows how the register clock, clear, and control signals are implemented in the Cyclone III device family M9K memory block.

Figure 3–1. M9K Control Signal Selection



Parity Bit Support

Parity checking for error detection is possible with the parity bit along with internal logic resources. The Cyclone III device family 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

The Cyclone III device family 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 III Device Family 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 3–2 shows how the `wren` and `byteena` signals control the RAM operations.

Figure 3–2. Cyclone III Device Family byteena Functional Waveform *(1)*

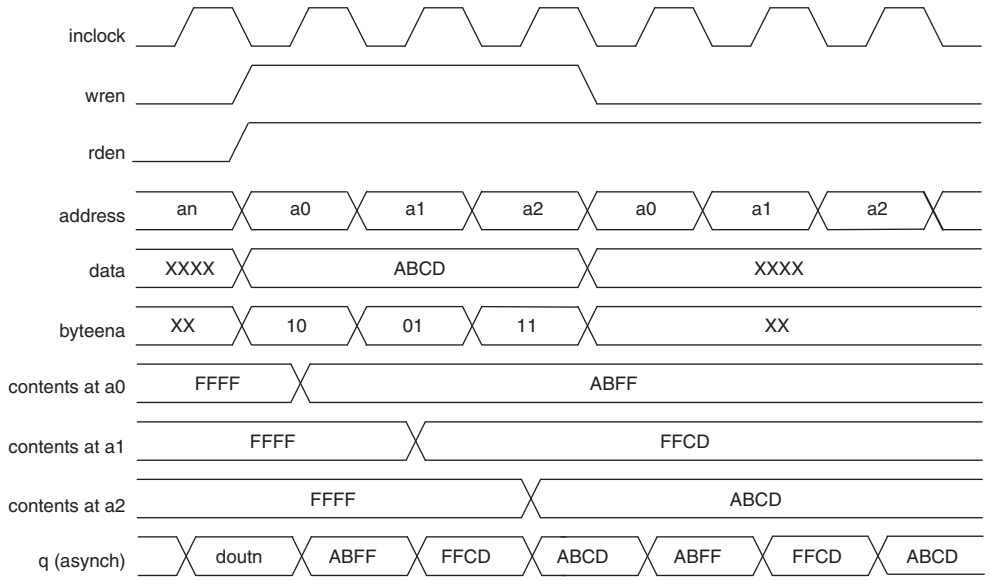


Figure 3–2. Cyclone III Device Family byteena Functional Waveform *(1)*

Note to Figure 3–2:

(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.

Packed Mode Support

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

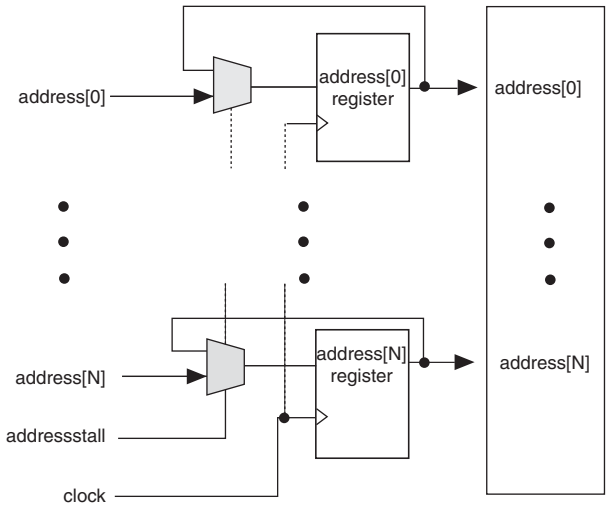
- Each of the two independent block sizes is less than or equal to half of the M9K block size. The maximum data width for each independent block is 18 bits wide.
- Each of the single-port memory blocks is configured in single-clock mode. For more information about packed mode support, refer to "Single-Port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

Address Clock Enable Support

Cyclone III device family 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 3–3 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 3–3. Cyclone III Device Family 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 3–4 and Figure 3–5 show the address clock enable waveform during read and write cycles, respectively.

Figure 3–4. Cyclone III Device Family Address Clock Enable During Read Cycle Waveform

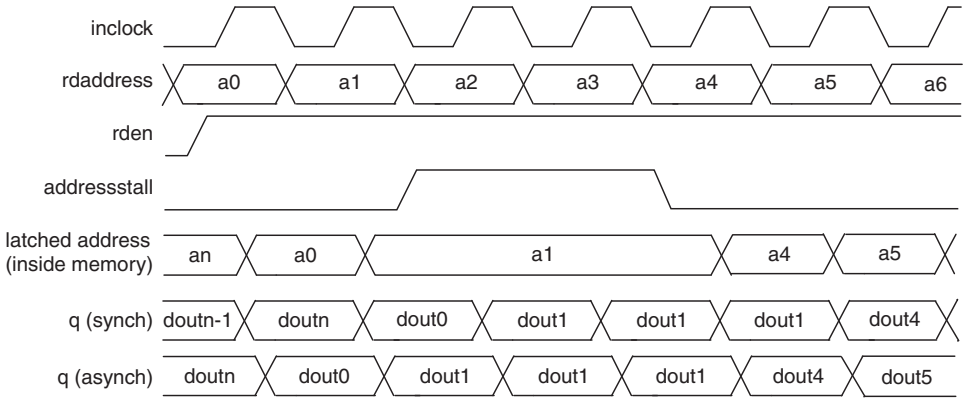
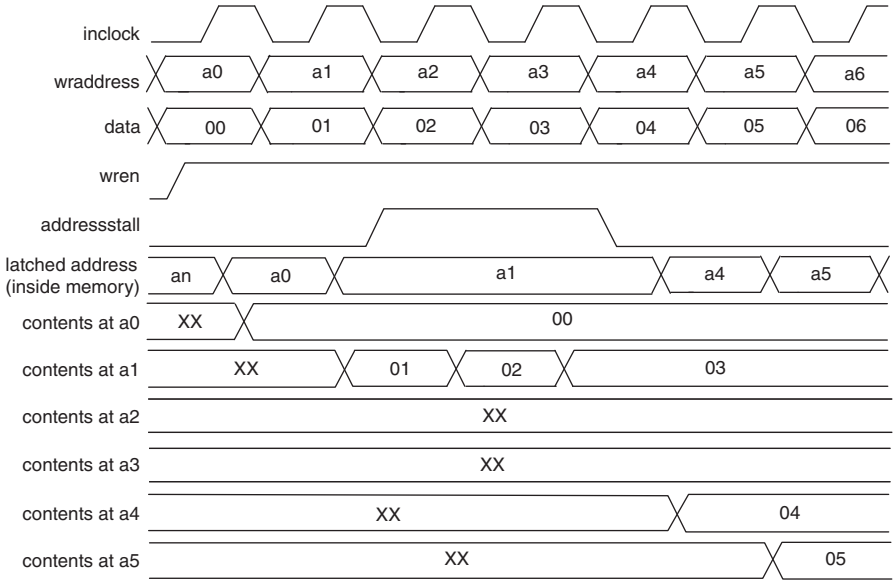


Figure 3–5. Cyclone III Device Family Address Clock Enable During Write Cycle Waveform



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.

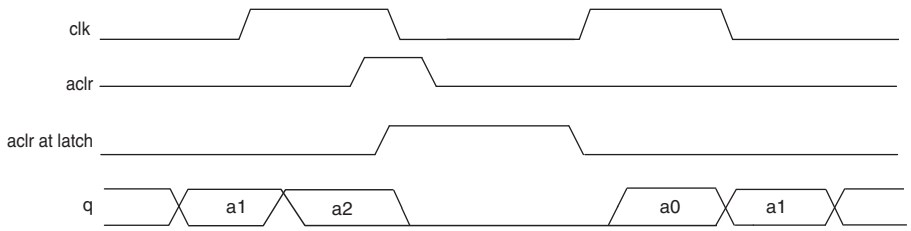
Asynchronous Clear

The Cyclone III device family supports 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 might corrupt the memory content.

Figure 3–6 shows the functional waveform for the asynchronous clear feature.

Figure 3–6. 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 *Internal Memory (RAM and ROM) 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 III device family M9K memory blocks allow you to implement fully-synchronous SRAM memory in multiple modes of operation. Cyclone III device family M9K memory blocks do not support asynchronous (unregistered) memory inputs.

M9K memory blocks support the following modes:

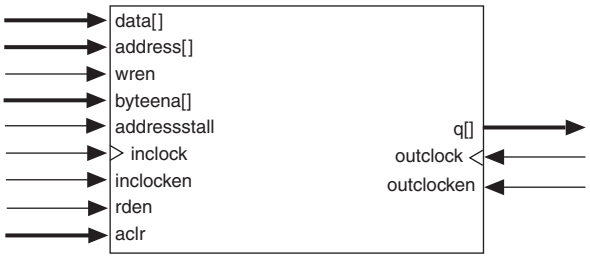
- Single-port
- Simple dual-port
- True dual-port
- Shift-register
- ROM
- FIFO

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 3–7 shows the single-port memory configuration for Cyclone III device family M9K memory blocks.

Figure 3–7. Single-Port Memory *(1), (2)*



Notes to Figure 3–7:

(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 3–5.

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.

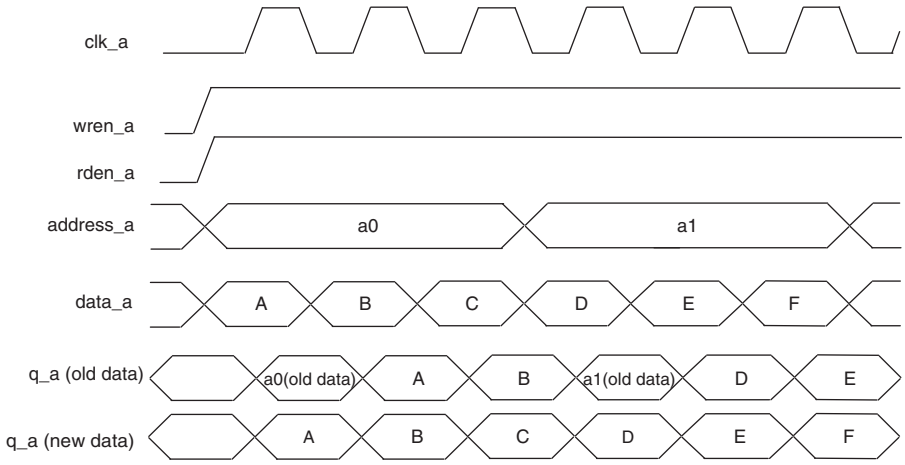
To choose the desired behavior, set the **Read-During-Write** option to either **New Data** or **Old Data** in the RAM MegaWizard Plug-In Manager in the Quartus II software. For more information about read-during-write mode, refer to "Read-During-Write Operations" on page 3–15.

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 3–8 shows timing waveforms 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.

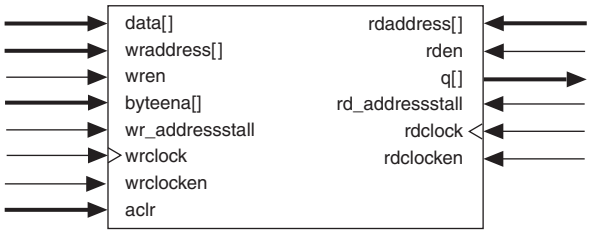
Figure 3–8. Cyclone III Device Family Single-Port Mode Timing Waveforms



Simple Dual-Port Mode

Simple dual-port mode supports simultaneous read and write operations to different locations. Figure 3–9 shows the simple dual-port memory configuration.

Figure 3–9. Cyclone III Device Family Simple Dual-Port Memory *(1)*



Note to Figure 3–9:

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

Cyclone III device family 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 III Device Family 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	✓	✓	✓	✓	✓	✓	—	—	—

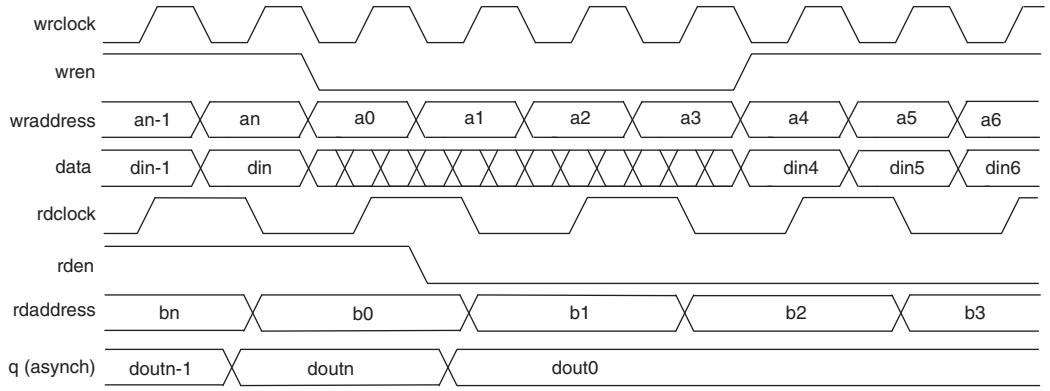
Table 3–3. Cyclone III Device Family M9K Block Mixed-Width Configurations (Simple Dual-Port Mode) ***(Part 2 of 2)***

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 3–10 shows the timing waveforms 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.

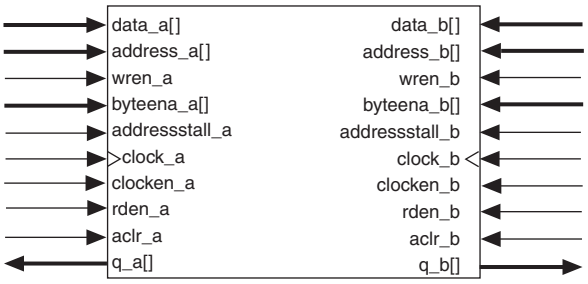
Figure 3–10. Cyclone III Device Family Simple Dual-Port Timing Waveforms



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 3–11 shows the Cyclone III device family true dual-port memory configuration.

Figure 3–11. Cyclone III Device Family True Dual-Port Memory *(1)*



Note to Figure 3–11:

(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 III Device Family M9K Block Mixed-Width Configurations (True Dual-Port Mode)

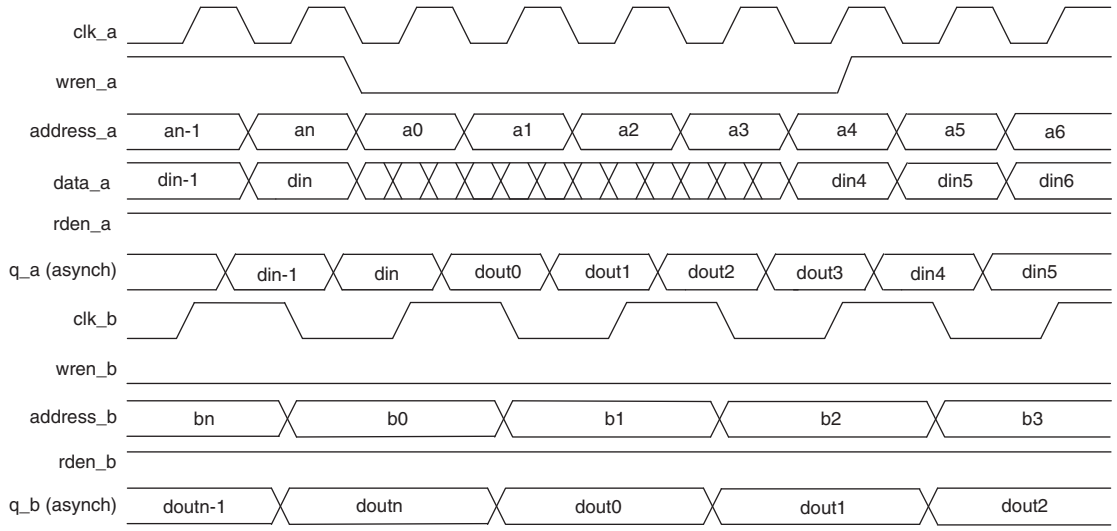
Read Port	Write Port						
	8192 × 1	4096 × 2	2048 × 4	1024 × 8	512 × 16	1024 × 9	512 × 18
8192 × 1	✓	✓	✓	✓	✓	—	—
4096 × 2	✓	✓	✓	✓	✓	—	—
2048 × 4	✓	✓	✓	✓	✓	—	—
1024 × 8	✓	✓	✓	✓	✓	—	—
512 × 16	✓	✓	✓	✓	✓	—	—
1024 × 9	—	—	—	—	—	✓	✓
512 × 18	—	—	—	—	—	✓	✓

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 III device family M9K memory blocks. You must handle address conflicts external to the RAM block.

Figure 3–12 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 3–12. Cyclone III Device Family True Dual-Port Timing Waveforms



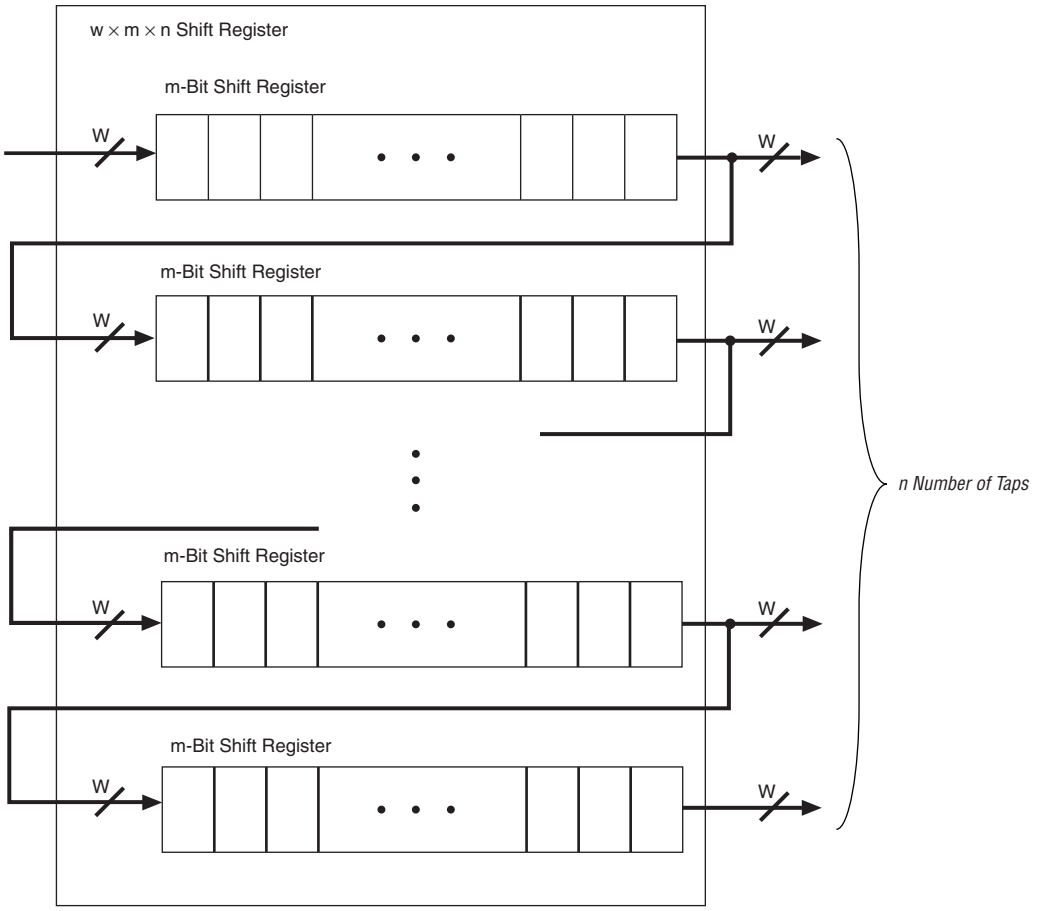
Shift Register Mode

Cyclone III device family 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 3–13 shows the Cyclone III device family M9K memory block in the shift register mode.

Figure 3–13. Cyclone III Device Family Shift Register Mode Configuration



ROM Mode

Cyclone III device family 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 III device family 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 III device family M9K memory blocks do not support simultaneous read and write from an empty FIFO buffer.

For more information about FIFO buffers, refer to the *SCFIFO and DCFIFO Megafunctions* user guide.

Clocking Modes

Cyclone III device family 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 III Device Family 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 III device family 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.

I/O Clock Mode

Cyclone III device family 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 III device family 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 III device family 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

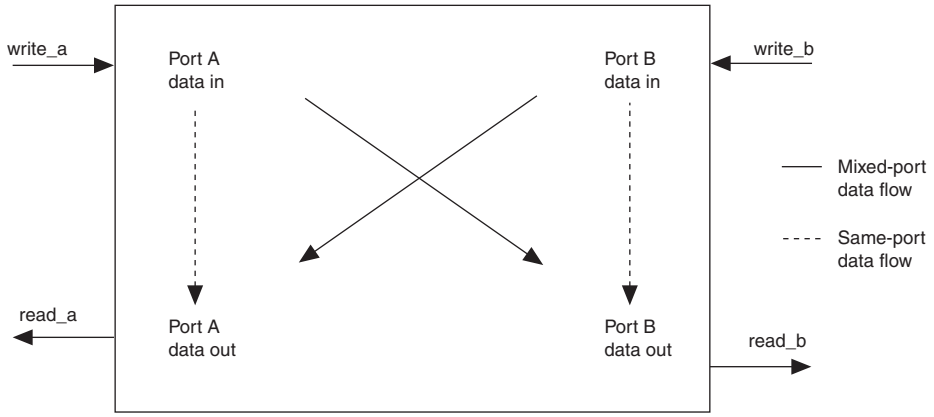
This section describes designing with M9K memory blocks.

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 3–14 shows the difference between these flows.

Figure 3–14. Cyclone III Device Family 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 3–15 and Figure 3–16 show sample functional waveforms of same port read-during-write behavior with both **New Data** and **Old Data** modes, respectively.

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

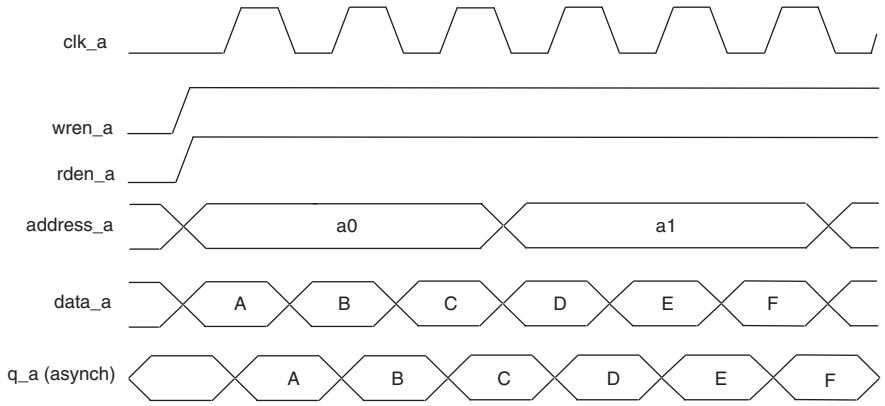
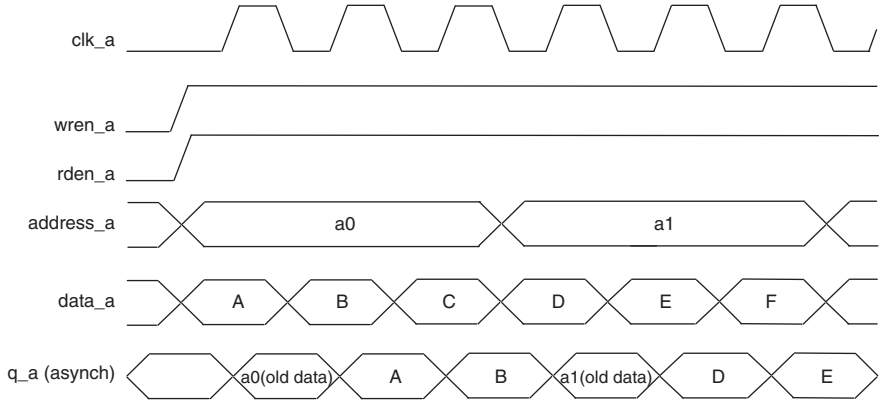


Figure 3–16. Same Port Read-During-Write: Old Data Mode



Mixed-Port Read-During-Write Mode

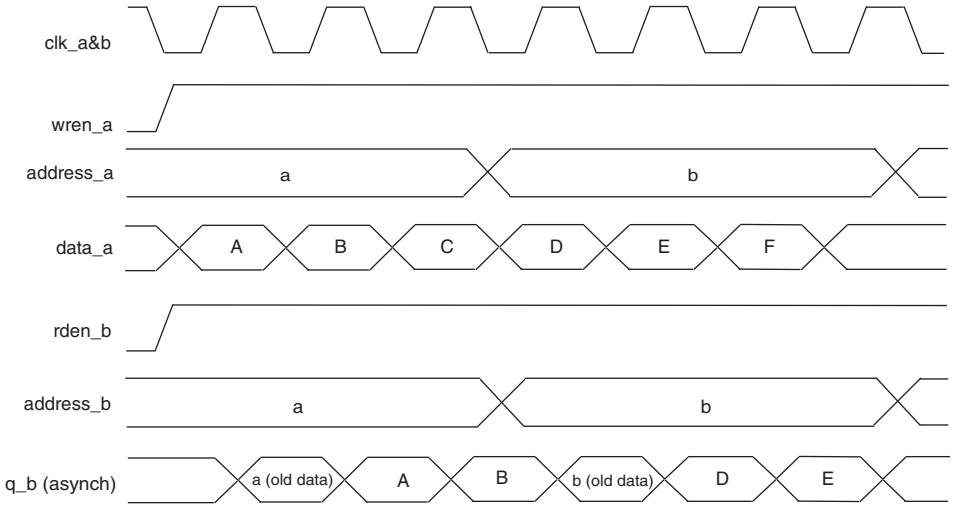
This mode applies to a RAM in simple or true dual-port mode, which has one port reading and the other port writing to the same address location with the same clock.

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 *Internal Memory (RAM and ROM) User Guide*.

Figure 3–17 shows a sample functional waveform of mixed port read-during-write behavior for the **Old Data** mode. In **Don't Care** mode, the old data is replaced with "Don't Care".

Figure 3–17. 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 the Cyclone III device family 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 **.mifs** 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 **.mifs**, refer to the *Internal Memory (RAM and ROM) User Guide* and the *Quartus II Handbook*.

Power Management

The M9K memory block clock enables of the Cyclone III device family 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 lists the revision history for this document.

Table 3–6. Document Revision History

Date	Version	Changes
December 2011	2.3	Minor text edits.
December 2009	2.2	Minor changes to the text.
July 2009	2.1	Made minor correction to the part number.
June 2009	2.0	Updated to include Cyclone III LS information ■ Updated chapter part number. ■ Updated “Introduction” on page 3–1. ■ Updated “Overview” on page 3–1. ■ Updated Table 3–1 on page 3–2. ■ Updated “Control Signals” on page 3–3. ■ Updated “Memory Modes” on page 3–8. ■ Updated “Simple Dual-Port Mode” on page 3–10. ■ Updated “Read or Write Clock Mode” on page 3–16.
October 2008	1.3	Updated chapter to new template.
May 2008	1.2	■ Revised the maximum performance of the M9K blocks to 315 MHz in “Introduction” and “Overview” sections, and in Table 3-1. ■ Updated “Address Clock Enable Support” section.
July 2007	1.1	Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

4. Embedded Multipliers in the Cyclone III Device Family

CIII51005-2.3

The Cyclone® III device family (Cyclone III and Cyclone III LS devices) includes a combination of on-chip resources and external interfaces that help to increase performance, reduce system cost, and lower the power consumption of digital signal processing (DSP) systems. The Cyclone III device family, 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 III and Cyclone III LS 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 4–2
- "Architecture" on page 4–3
- "Operational Modes" on page 4–5

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

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

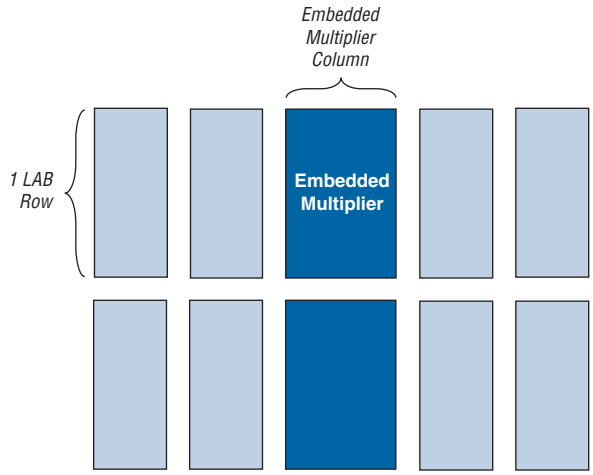


Table 4–1 lists the number of embedded multipliers and the multiplier modes that can be implemented in the Cyclone III device family.

Table 4–1. Number of Embedded Multipliers in the Cyclone III Device Family

Device Family	Device	Embedded Multipliers	9 × 9 Multipliers (1)	18 × 18 Multipliers (1)
Cyclone III	EP3C5	23	46	23
	EP3C10	23	46	23
	EP3C16	56	112	56
	EP3C25	66	132	66
	EP3C40	126	252	126
	EP3C55	156	312	156
	EP3C80	244	488	244
	EP3C120	288	576	288
Cyclone III LS	EP3CLS70	200	400	200
	EP3CLS100	276	552	276
	EP3CLS150	320	640	320
	EP3CLS200	396	792	396

Note to Table 4–1:

(1) These columns show the number of 9 × 9 or 18 × 18 multipliers for each device. The total number of multipliers for each device is not the sum of all the multipliers.

In addition to the embedded multipliers in the Cyclone III device family, 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.

Table 4–2 lists the total number of multipliers available in the Cyclone III device family using embedded multipliers and soft multipliers.

Table 4–2. Number of Multipliers in the Cyclone III Device Family

Device Family	Device	Embedded Multipliers	Soft Multipliers (16 × 16) (1)	Total Multipliers (2)
Cyclone III	EP3C5	23	—	23
	EP3C10	23	46	69
	EP3C16	56	56	112
	EP3C25	66	66	132
	EP3C40	126	126	252
	EP3C55	156	260	416
	EP3C80	244	305	549
	EP3C120	288	432	720
Cyclone III LS	EP3CLS70	200	333	533
	EP3CLS100	276	483	759
	EP3CLS150	320	666	986
	EP3CLS200	396	891	1287

Notes to Table 4–2:

(1) Soft multipliers are implemented in sum of multiplication mode. M9K memory blocks are configured with 18-bit data widths to support 16-bit coefficients. The sum of the coefficients requires 18-bits of resolution to account for overflow.

(2) The total number of multipliers may vary, depending on the multiplier mode you use.

For more information about M9K memory blocks of the Cyclone III device family, refer to the *Memory Blocks in the Cyclone III Device Family* chapter.

For more information about soft multipliers, refer to the *Implementing Multipliers in FPGA Devices* application note.

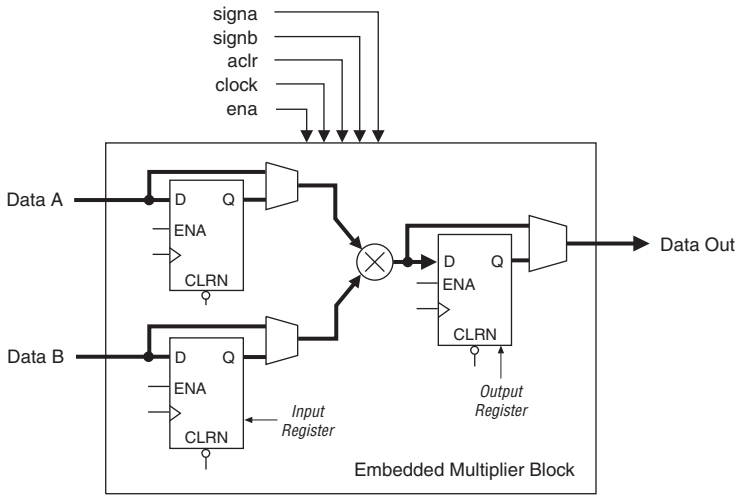
Architecture

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. Each multiplier input signal can be sent 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 to 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 in 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 4–5.

Each multiplier operand is a unique signed or unsigned number. Two signals, `signa` and `signb`, 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 4–3 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 4–3. 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 using 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-bit × 18-bit multiplier
- Up to two 9-bit × 9-bit independent multipliers

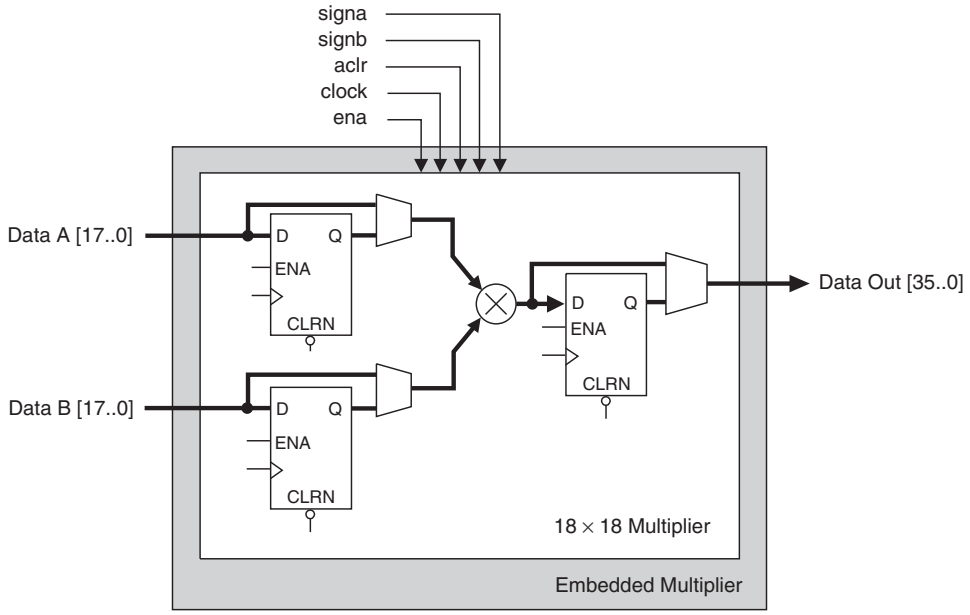
You can also use embedded multipliers of the Cyclone III device family to implement multiplier adder and multiplier accumulator functions, in which the multiplier portion of the function is implemented using embedded multipliers, and the adder or accumulator function is implemented in logic elements (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



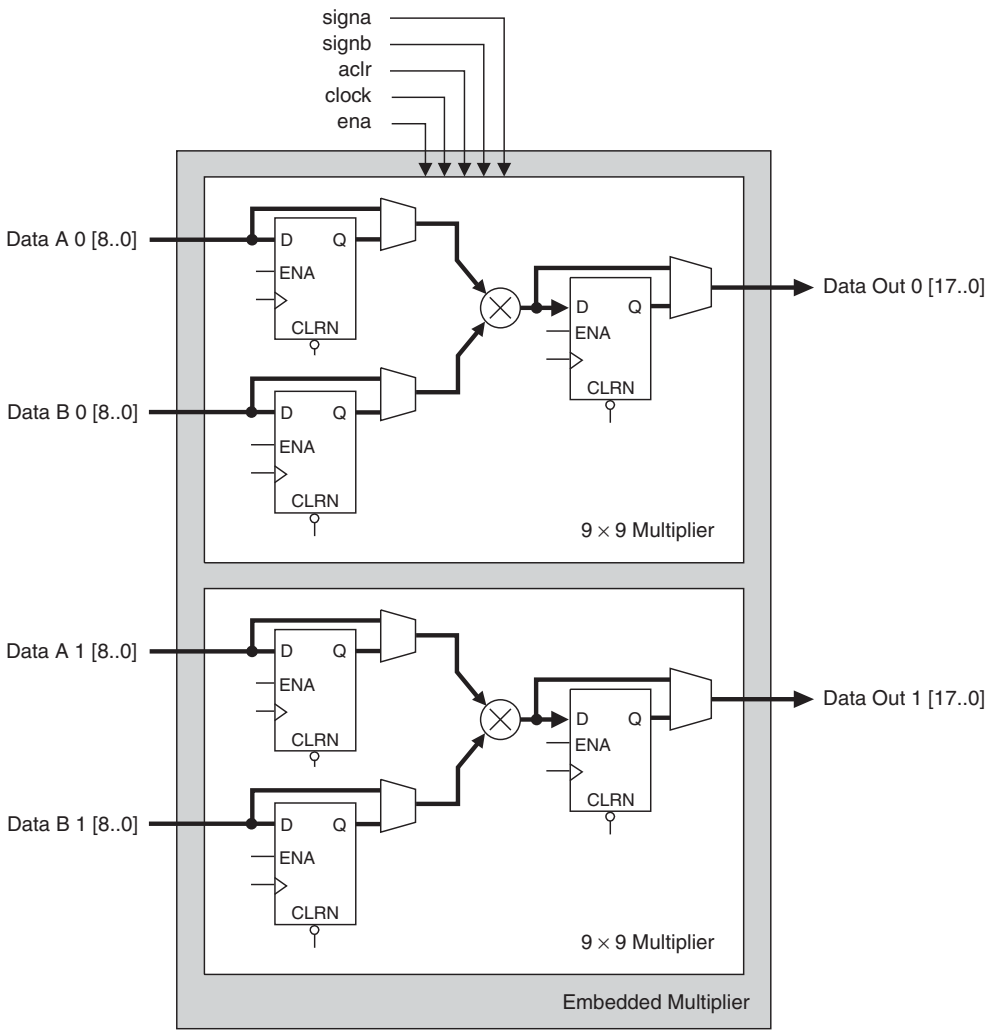
All 18-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Also, you can dynamically change the `signa` and `signb` signals and send these signals through dedicated input registers.

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 4–4 lists the revision history for this document.

Table 4–4. Document Revision History

Date	Version	Changes
December 2011	2.3	Minor text edits.
December 2009	2.2	Minor changes to the text.
July 2009	2.1	Made minor correction to the part number.
June 2009	2.0	Updated to include Cyclone III LS information ■ Updated chapter part number. ■ Updated “Introduction” on page 4–1. ■ Updated “Embedded Multiplier Block Overview” on page 4–1. ■ Updated Table 4–1 on page 4–2 and Table 4–2 on page 4–2. ■ Updated “Input Registers” on page 4–4.
October 2008	1.2	Updated chapter to new template.
July 2007	1.1	Added EP3C120 information. ■ Updated “Introduction” section. ■ Updated Table 4–1 and Table 4–2. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

5. Clock Networks and PLLs in the Cyclone III Device Family

CIII51006-4.1

This chapter describes the hierarchical clock networks and phase-locked loops (PLLs) with advanced features in the Cyclone® III device family (Cyclone III and Cyclone III LS devices).

This chapter includes the following sections:

- "Clock Networks" on page 5–1
- "PLLs in the Cyclone III Device Family" on page 5–9
- "Cyclone III Device Family PLL Hardware Overview" on page 5–10
- "Clock Feedback Modes" on page 5–11
- "Hardware Features" on page 5–15
- "Programmable Bandwidth" on page 5–22
- "Phase Shift Implementation" on page 5–22
- "PLL Cascading" on page 5–24
- "PLL Reconfiguration" on page 5–26
- "Spread-Spectrum Clocking" on page 5–33
- "PLL Specifications" on page 5–33

Clock Networks

The Cyclone III device family provides up to 16 dedicated clock pins (`CLK[15..0]`) that can drive the global clocks (GCLKs). The Cyclone III device family supports four dedicated clock pins on each side of the device except EP3C5 and EP3C10 devices. EP3C5 and EP3C10 devices only support four dedicated clock pins on the left and right sides of the device.

For more information about the number of GCLK networks in each device density, refer to the *Cyclone III 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.

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

ISO 9001:2008 Registered

Table 5–1 lists the connectivity of the clock sources to the GCLK networks.

Table 5–1. Cyclone III Device Family GCLK Network Connections (Part 1 of 2)

GCLK Network Clock Sources	GCLK Networks (1)																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CLK0/DIFFCLK_0p	✓	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
CLK1/DIFFCLK_0n	—	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	✓
CLK15/DIFFCLK_6p (2)	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—
PLL1_C0 (3)	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C1 (3)	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C2 (3)	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C3 (3)	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL1_C4 (3)	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
PLL2_C0 (3)	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
PLL2_C1 (3)	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—	—	—	—	—
PLL2_C2 (3)	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
PLL2_C3 (3)	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—	—
PLL2_C4 (3)	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—	—	—	—
PLL3_C0	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—	—
PLL3_C1	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—	—	—	—	—
PLL3_C2	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—	—
PLL3_C3	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—	—
PLL3_C4	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—	—	—	—
PLL4_C0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓	—

Table 5–1. Cyclone III Device Family GCLK Network Connections (Part 2 of 2)

GCLK Network Clock Sources	GCLK Networks (1)																			
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
PLL4_C1	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	✓
PLL4_C2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—	—
PLL4_C3	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓	—
PLL4_C4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	✓
DPCLK0	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK1	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
DPCLK7 (4) CDPCLK0, or CDPCLK7 (2), (5)	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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 5–1:

(1) EP3C5 and EP3C10 devices only have GCLK networks 0 to 9.
(2) These pins apply to all devices in the Cyclone III device family except EP3C5 and EP3C10 devices.
(3) EP3C5 and EP3C10 devices only have phase-locked loops (PLLs) 1 and 2.
(4) This pin applies only to EP3C5 and EP3C10 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 III Device Family Pin Connection Guidelines* on the Altera® website.

Clock Control Block

The clock control block drives 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 5–2 lists the sources that can feed the clock control block, which in turn feeds the GCLKs.

Table 5–2. 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 the Cyclone III device family, 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.

Normal I/O pins cannot drive the PLL input clock port.

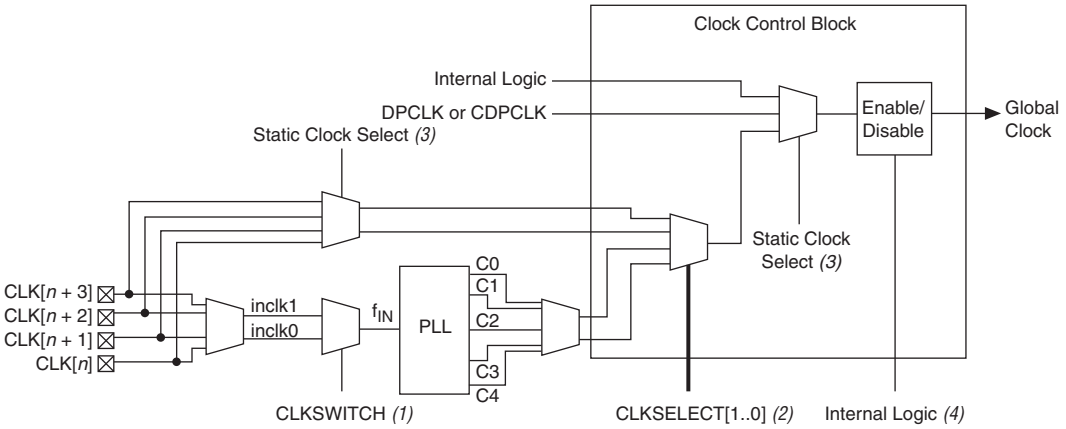
The output from the clock control block in turn feeds the corresponding GCLK. The GCLK can drive the PLL input if the clock control block inputs are outputs of another PLL or dedicated clock input pins. The clock control blocks are at the device periphery; there are a maximum of 20 clock control blocks available per Cyclone III device family.

The control block has two functions:

- Dynamic GCLK clock source selection (not applicable for `DPCLK` or `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 is 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) You can use internal logic to enable or disable the GCLK in user mode.

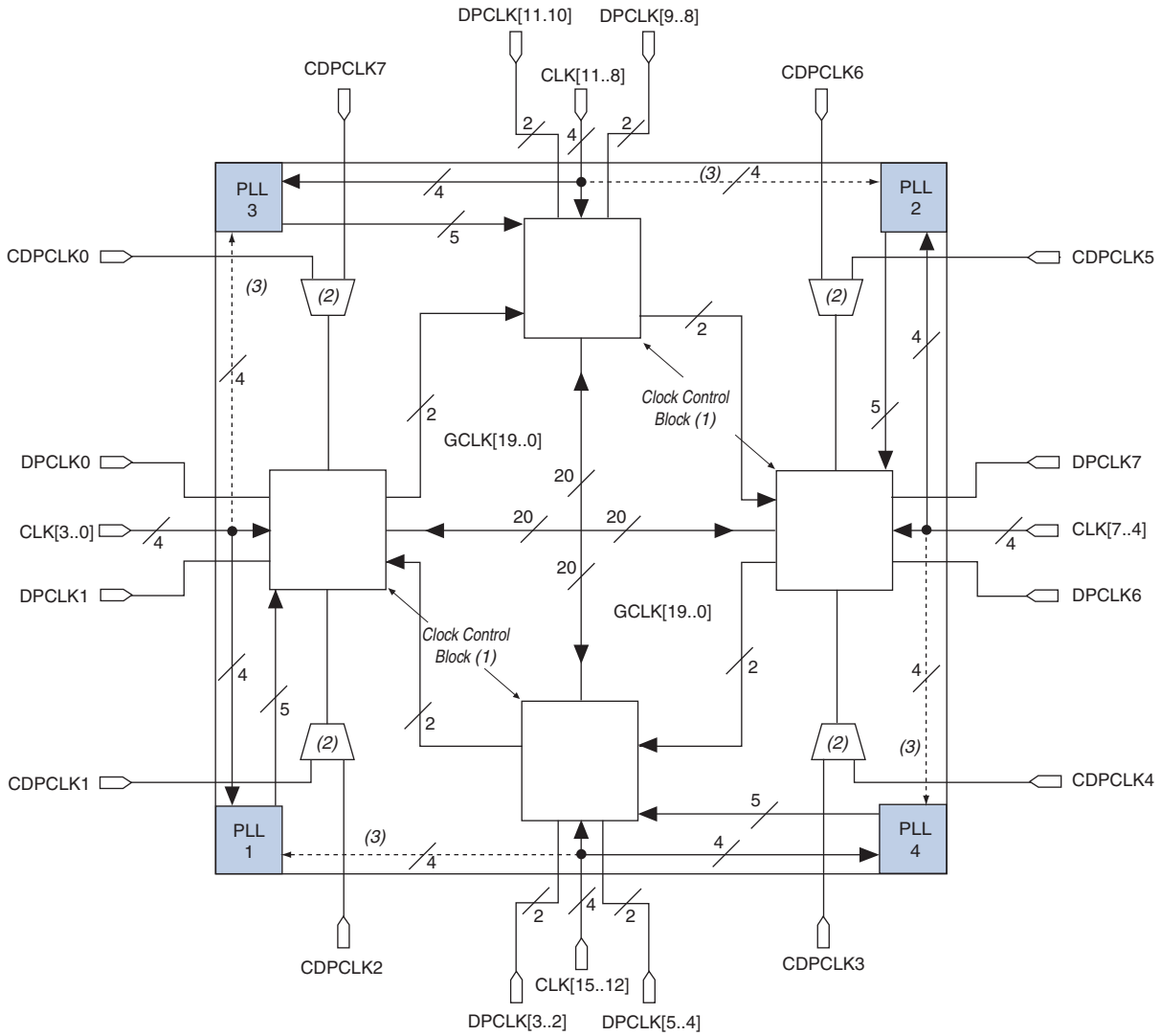
Each PLL generates five clock outputs through the `c[4..0]` counters. Two of these clocks can drive the GCLK through a clock control block, as shown in Figure 5–1.

For more information about how to use the clock control block in the Quartus® II software, refer to the *Clock Control Block (ALTCLKCTRL) Megafunction User Guide*.

GCLK Network Clock Source Generation

Figure 5–2 shows Cyclone III device family PLLs, clock inputs, and clock control block location for different device densities.

Figure 5–2. PLL, CLK[], DPCLK[], and Clock Control Block Locations in the Cyclone III Device Family *(1)*



Notes to Figure 5–2:

(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 pins.

(3) Remote clock pins can feed PLLs over dedicated clock paths. However, these paths are not fully compensated.

The inputs to the five clock control blocks on each side must be chosen from among the following clock sources:

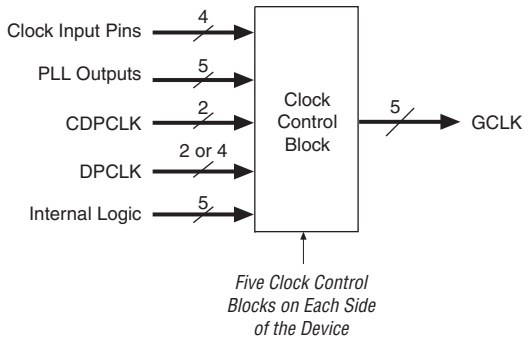
- Four clock input pins
- Five PLL counter outputs
- Two DPCLK pins and two CDPCLK pins from both the left and right sides, and four DPCLK pins and two CDPCLK pins from both the top and bottom
- Five signals from internal logic

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 5–5.

Out of these five 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 5–3 shows a simplified version of the five clock control blocks on each side of the Cyclone III device family periphery.

Figure 5–3. Clock Control Blocks on Each Side of the Cyclone III Device Family *(1)*



Note to Figure 5–3:

(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 the Cyclone III device family GCLK (power down) by 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 of the GCLKs in the Cyclone III device family.

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 5–5.

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

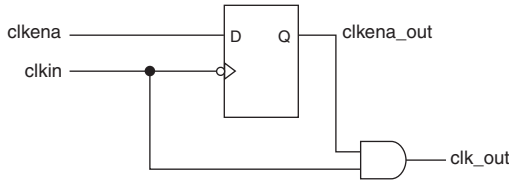
For more information, refer to the *Clock Control Block (ALTCLKCTRL) Megafunction User Guide*.

clkena Signals

The Cyclone III device family supports `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 5–4 shows how to implement the `clkena` signal.

Figure 5–4. clkena Implementation

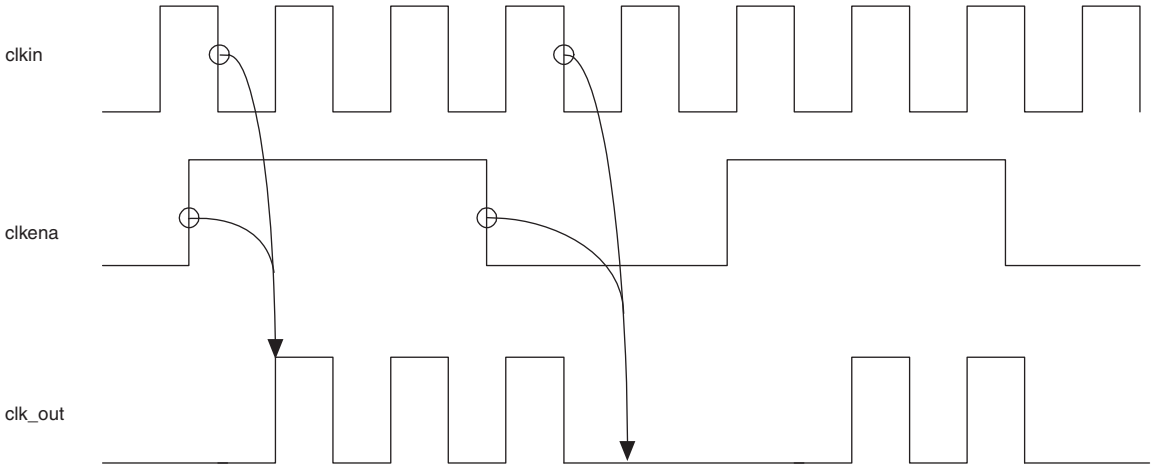


The `clkena` circuitry controlling the output C0 of the PLL to an output pin is implemented with two registers instead of a single register, as shown in Figure 5–4.

Figure 5–5 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 5–5. 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 deasserting 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 the Cyclone III Device Family

The Cyclone III device family offers up to four 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 PLLs in each device density, refer to the *Cyclone III Device Family Overview* chapter.

The Cyclone III device family PLLs have the same core analog structure.

Table 5–3 lists the features available in the Cyclone III device family PLLs.

Table 5–3. Cyclone III Device Family PLL Hardware Features

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	✓
Loss of lock detection	✓

Notes to Table 5–3:

(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.

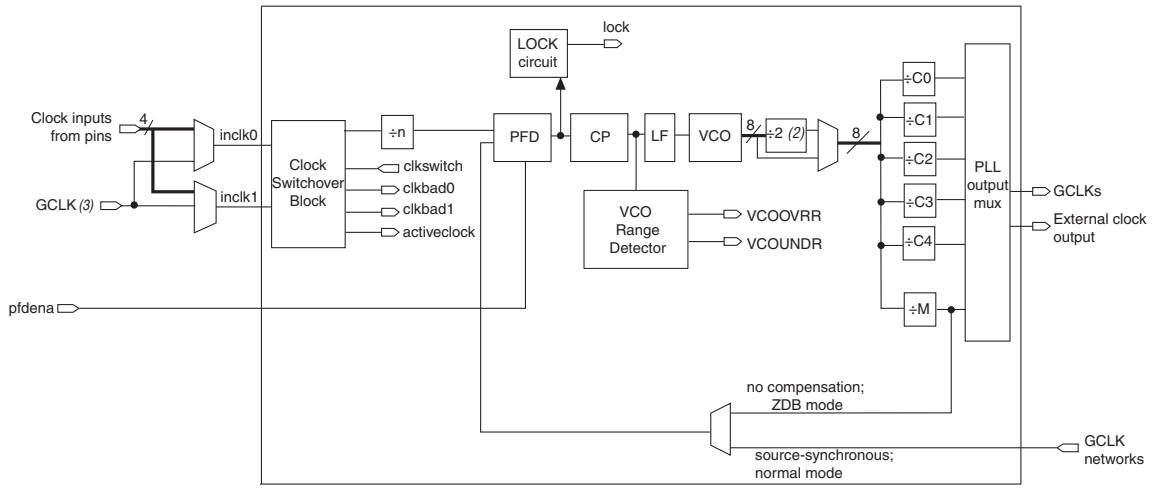
(3) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by eight. For degree increments, the Cyclone III device family can shift all output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the frequency and divide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

Cyclone III Device Family PLL Hardware Overview

This section gives a hardware overview of the Cyclone III device family PLL.

Figure 5–6 shows a simplified block diagram of the major components of the PLL of the Cyclone III device family.

Figure 5–6. Cyclone III Device Family PLL Block Diagram *(1)*



Notes to Figure 5–6:

(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.

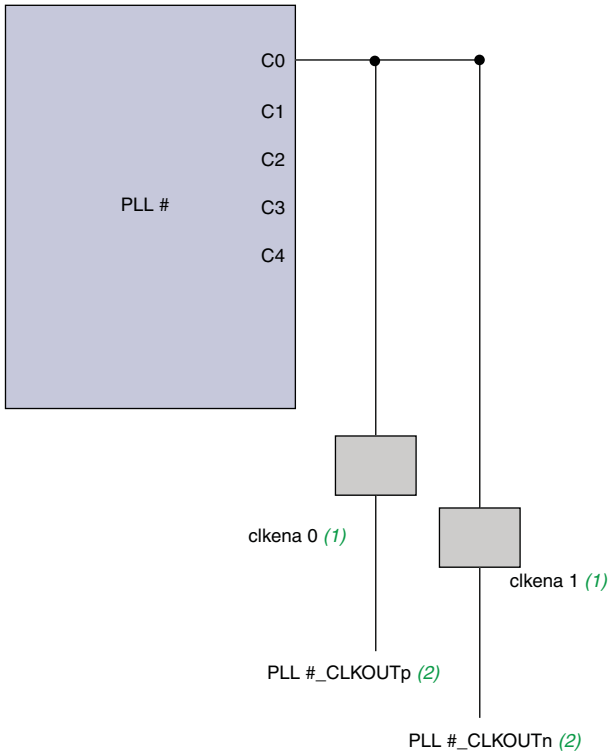
The VCO post-scale counter K is used to divide the supported VCO range by two. The VCO frequency reported by the Quartus II software in the PLL summary section of the compilation report takes into consideration the VCO post-scale counter value. Therefore, if the VCO post-scale counter has a value of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

External Clock Outputs

Each PLL of the Cyclone III device family 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 5–7, without going through the GCLK. Other output counters can feed other I/O pins through the GCLK.

Figure 5–7 shows the external clock outputs for PLLs.

Figure 5–7. External Clock Outputs for PLLs



Notes to Figure 5–7:

(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 or one differential clock output.

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 (in the top and bottom banks) as well as LVDS, LVPECL, differential HSTL, and differential SSTL.

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

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

Clock Feedback Modes

Cyclone III device family PLLs support up to four different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle.

Input and output delays are fully compensated by the PLL only when you are using the dedicated clock input pins associated with a given PLL as the clock sources. For example, when using PLL1 in normal mode, the clock delays from the input pin to the PLL and the PLL clock output-to-destination register are fully compensated, provided that the clock input pin is one of the following four pins:

- CLK0
- CLK1
- CLK2
- CLK3

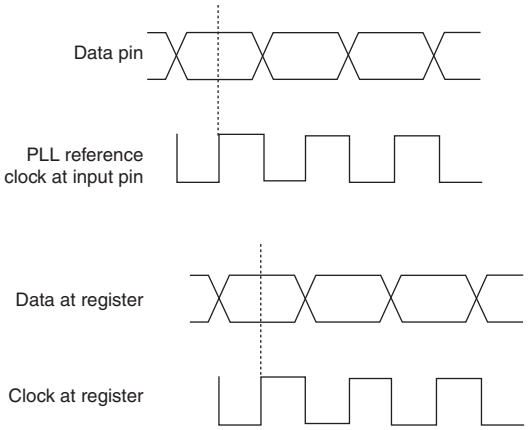
When driving the PLL using the GCLK network, the input and output delays may not be fully compensated in the Quartus II software.

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 5–8 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 5–8. 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

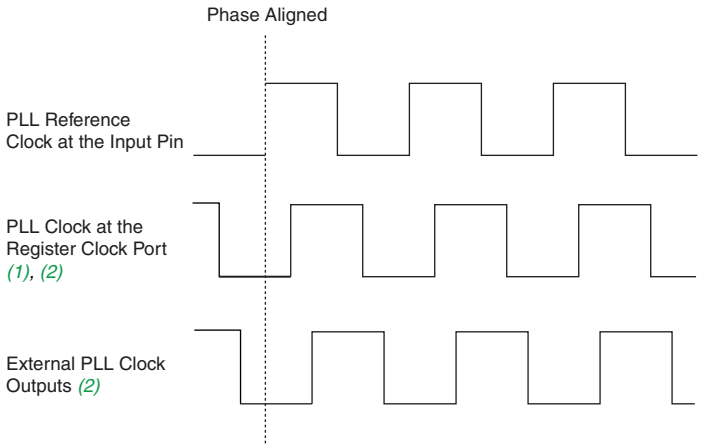
Set the input pin to the register delay chain in the I/O element to zero in the Quartus II software for all data pins clocked by a source-synchronous mode PLL. Also, all data pins must use the **PLL COMPENSATED logic** option in the Quartus II software.

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 5–9 shows a waveform example of the phase relationship of the PLL clock in this mode.

Figure 5–9. Phase Relationship Between PLL Clocks in No Compensation Mode



Notes to Figure 5–9:

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

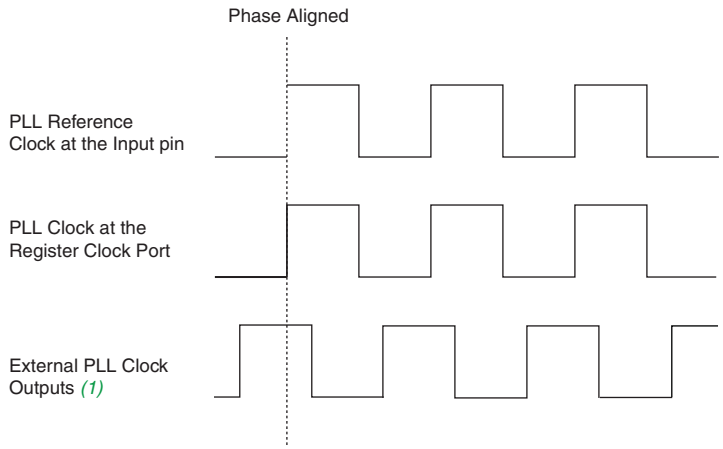
(2) The PLL clock outputs can lead or lag the PLL input clocks.

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 5–10 shows a waveform example of the phase relationship of the PLL clocks in this mode.

Figure 5–10. Phase Relationship Between PLL Clocks in Normal Mode



Note to Figure 5–10:

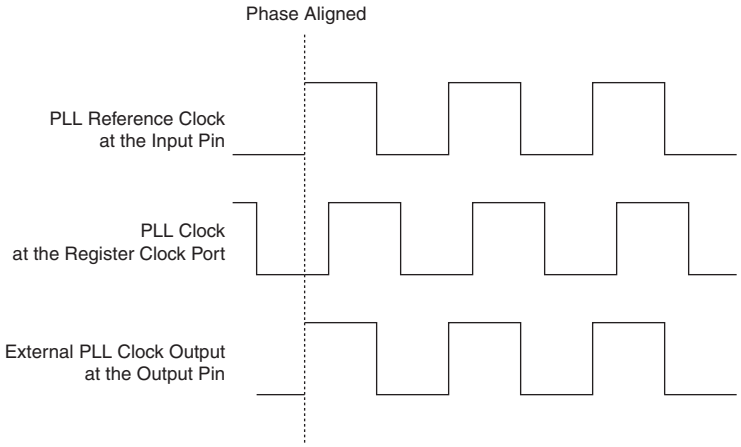
(1) The external clock output can lead or lag the PLL internal clock signals.

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 5–11 shows an example waveform of the phase relationship of the PLL clocks in ZDB mode.

Figure 5–11. Phase Relationship Between PLL Clocks in ZDB Mode



Hardware Features

Cyclone III device family 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 III device family 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.

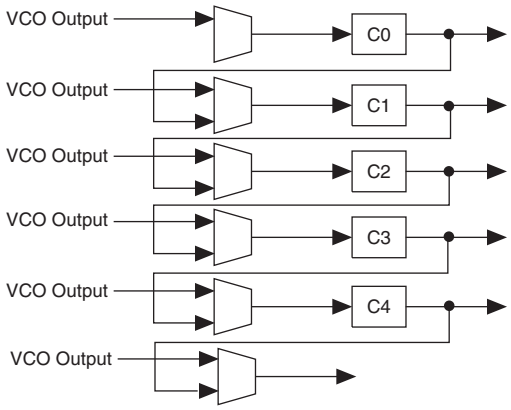
The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTPLL megafunction.

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

Post-Scale Counter Cascading

Cyclone III device family PLLs 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 5–12.

Figure 5–12. 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 following three signals to observe and control the PLL operation and resynchronization.

pfdena

Use the `pfdena` signal to maintain the last locked frequency so that your system has time to store its current settings before shutting down. The `pfdena` signal controls the PFD output with a programmable gate. If you disable the PFD, the VCO operates at its last set value of control voltage and frequency with some long-term drift to a lower frequency.

areset

The `areset` signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals. When driven high, the PLL counters reset, clearing the PLL output and placing the PLL out of lock. The VCO is then set back to its nominal setting. When driven low again, the PLL resynchronizes to its input as it re-locks.

You must include the `areset` signal in your designs if one of the following conditions is true:

- PLL reconfiguration or clock switchover enabled in your design
- Phase relationships between the PLL input clock and output clocks must be maintained after a loss-of-lock condition

If the input clock to the PLL is toggling or unstable upon power up, assert the `areset` signal after the input clock is stable and within specifications.

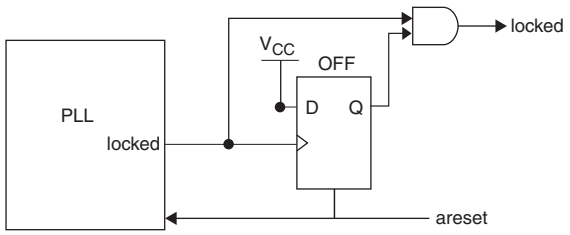
locked

The `locked` output indicates that the PLL has locked onto the reference clock and the PLL clock outputs are operating at the desired phase and frequency set in the Quartus II MegaWizard™ Plug-in Manager.

Altera recommends that you use the `areset` and `locked` signals in your designs to control and observe the status of your PLL.

This implementation is illustrated in Figure 5–13.

Figure 5–13. Locked Signal Implementation



If you use the SignalTap® II tool to probe the `locked` signal before the D flip-flop, the `locked` signal goes low only when `areset` is deasserted. If the `areset` signal is not enabled, the extra logic is not implemented in the ALTPLL megafunction.

For more information about the PLL control signals, refer to the *Phase-Locked Loop (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`.

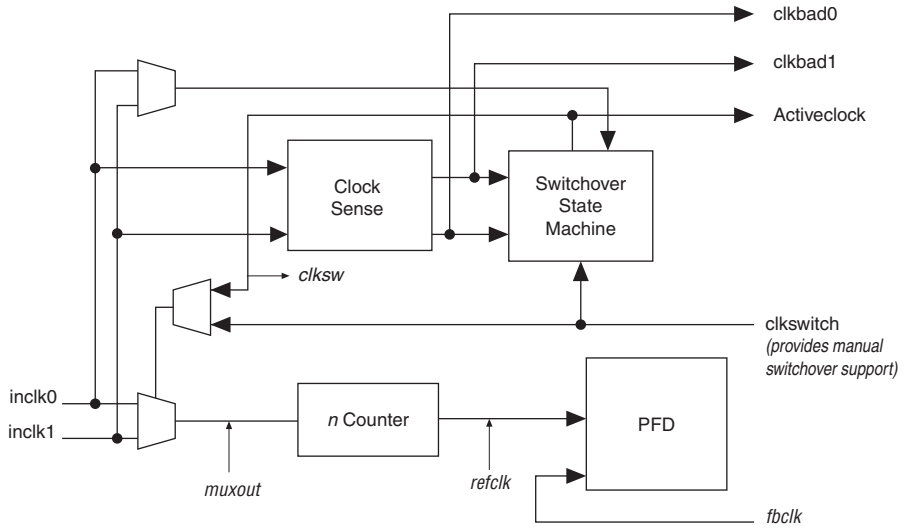
Automatic Clock Switchover

Cyclone III device family PLLs 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—`clkbad[0]`, `clkbad[1]`, 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 5–14 shows the block diagram of the switchover circuit built into the PLL.

Figure 5–14. Automatic Clock Switchover Circuit

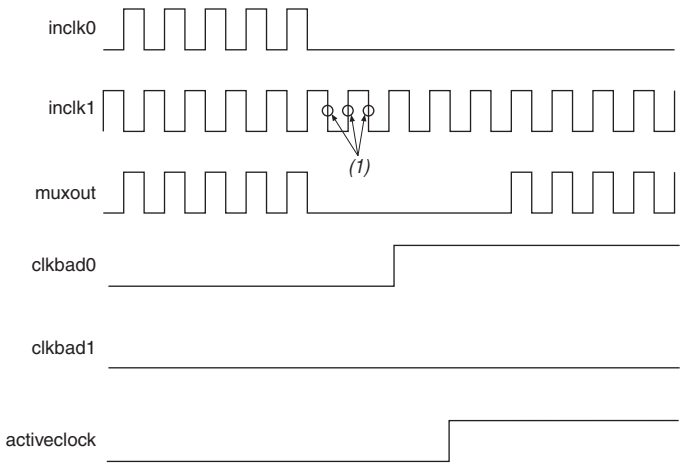


There are two ways to use the clock switchover feature:

- 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 5–14. 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 5–15 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 `clkbad[0]` 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 5–15. Automatic Switchover Upon Clock Loss Detection *(1)*



Note to Figure 5–15:

(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

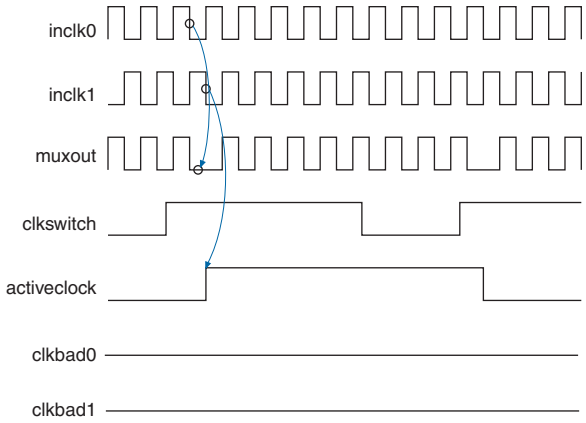
If you are using the automatic switchover, you must switch input clocks with the manual override feature with the `clkswitch` input.

Figure 5–16 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. 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.

If CLKSWITCH = 1, the automatic switchover function is overridden. While the `clkswitch` signal is high, further switchover action is blocked.

Figure 5–16. Clock Switchover Using the clkswitch Control *(1)*



Note to Figure 5–16:

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

Manual Clock Switchover

Cyclone III device family PLLs 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 is 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 *Phase-Locked Loop (ALTPLL) Megafunction User Guide*.

Guidelines

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

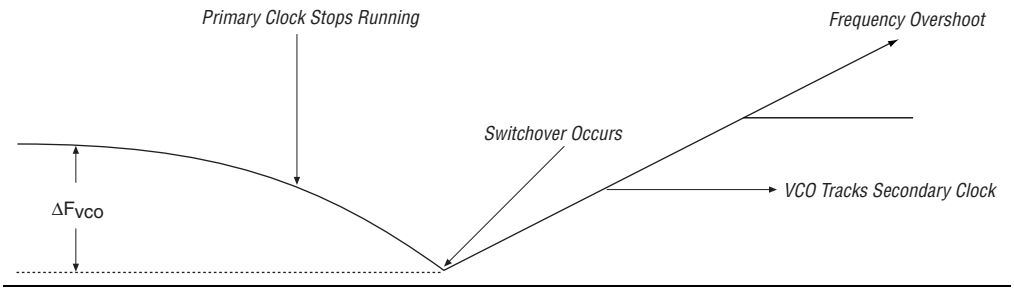
- Clock loss detection and automatic clock switchover requires that the `inclk0` and `inclk1` frequencies be within 20% of each other. Failing to meet this requirement causes the `clkbad[0]` and `clkbad[1]` signals to function improperly.
- When using manual clock switchover, the difference between `inclk0` 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, you must be aware that 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 5–17 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 5–17. 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 `clkbad[0]` and `clkbad[1]` 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. Cyclone III device family PLLs 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 the Cyclone III device family. 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 purely based on counter settings, which are independent of process, voltage, and temperature.

You can phase shift the output clocks from the Cyclone III device family PLLs in either:

- Fine resolution using VCO phase taps, or
- 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

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

Note to Equation 5–1:

(1) 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 Φfine = 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

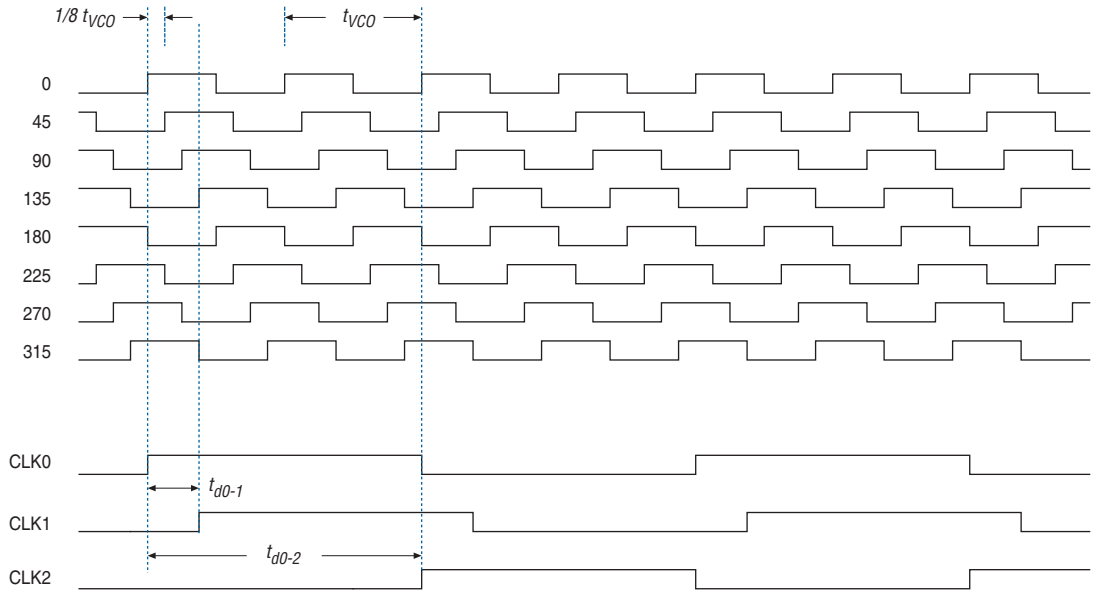
$$\Phi_{coarse} = \frac{C-1}{f_{VCO}} = \frac{(C-1)N}{Mf_{REF}}$$

Note to Equation 5–2:

(1) C is the count value set for the counter delay time—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 5–18 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 5–18. Delay Insertion Using VCO Phase Output and Counter Delay Time



You can use the coarse and fine phase shifts to implement clock delays in the Cyclone III device family.

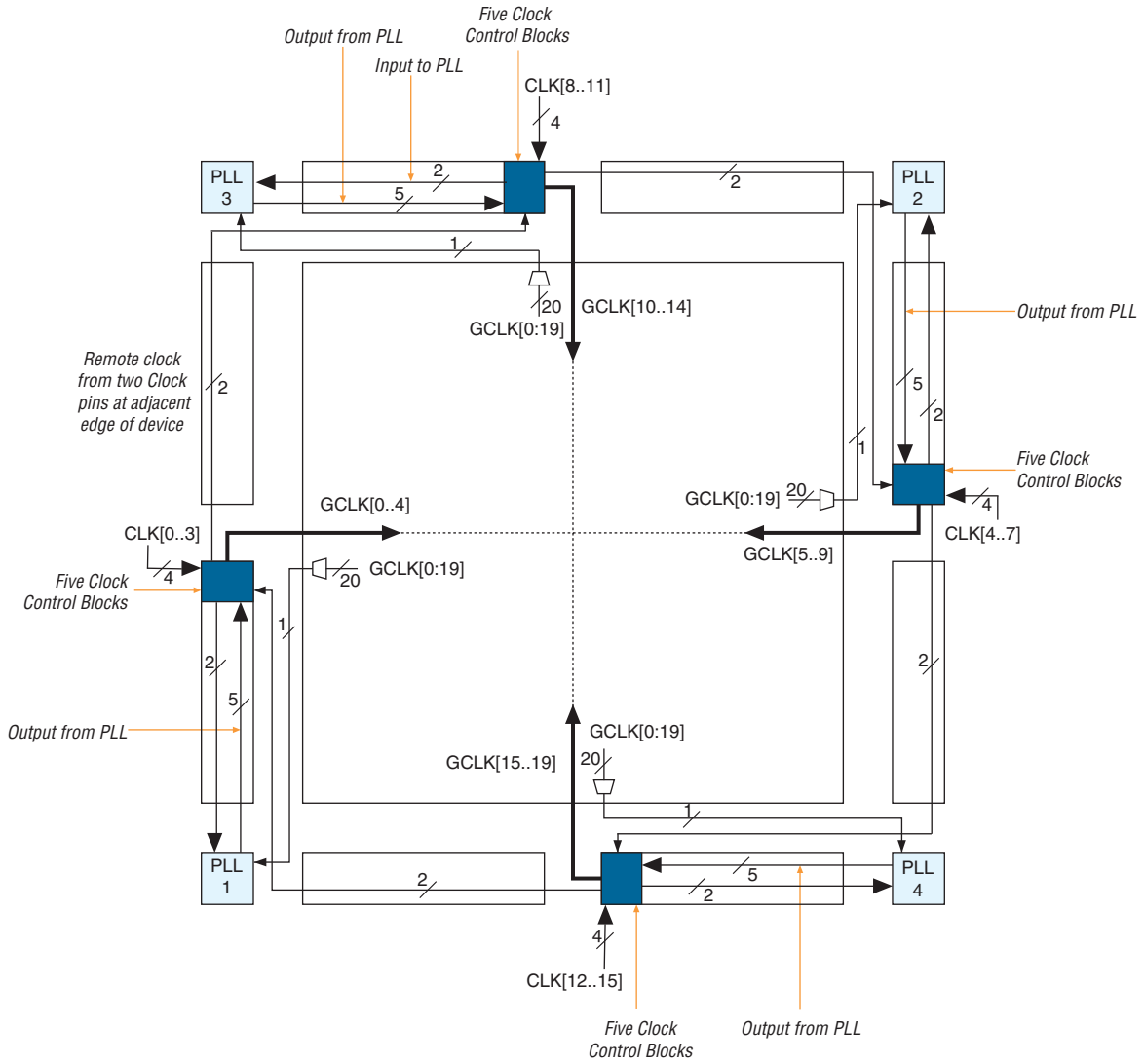
The Cyclone III device family supports dynamic phase shifting of VCO phase taps only. The phase shift is configurable for any number of times. Each phase shift takes about one scanclk cycle, allowing you to implement large phase shifts quickly.

PLL Cascading

Two PLLs are cascaded to each other through the clock 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.

Figure 5–19 shows using GCLK while cascading PLLs.

Figure 5–19. PLL Cascading Using GCLK



Consider the following guidelines when cascading PLLs:

- Set the primary PLL to low bandwidth to help filter jitter. Set the secondary PLL to high bandwidth to be able to track the jitter from the primary PLL. You can view the Quartus II software compilation report file to ensure the PLL bandwidth ranges do not overlap. If the bandwidth ranges overlap, jitter peaking can occur in the cascaded PLL scheme.

  ☞ You can get an estimate of the PLL deterministic jitter and static phase error (SPE) by using the TimeQuest Timing Analyzer in the Quartus II software. Use the SDC command "derive_clock_uncertainty" to direct TimeQuest to generate a report titled "PLLJ_PLLSPE_INFO.txt" in your project directory. Then, use "set_clock_uncertainty" commands to add jitter and SPE values to your clock constraints.

- Keep the secondary PLL in a reset state until the primary PLL has locked to ensure the phase settings are correct on the secondary PLL.
- You cannot connect either of the `inclk` ports of any PLLs in the cascaded scheme to clock outputs from PLLs in the cascaded scheme.

PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In Cyclone III device family PLLs, 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.

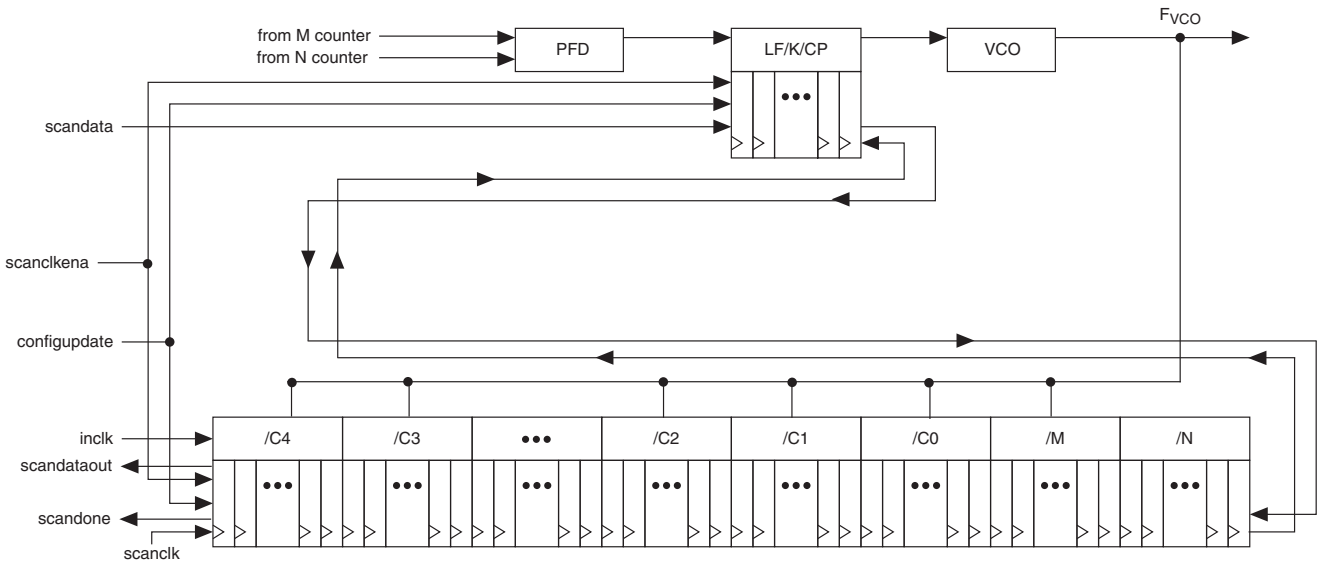
PLL Reconfiguration Hardware Implementation

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 5–20 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 5–20. PLL Reconfiguration Scan Chain



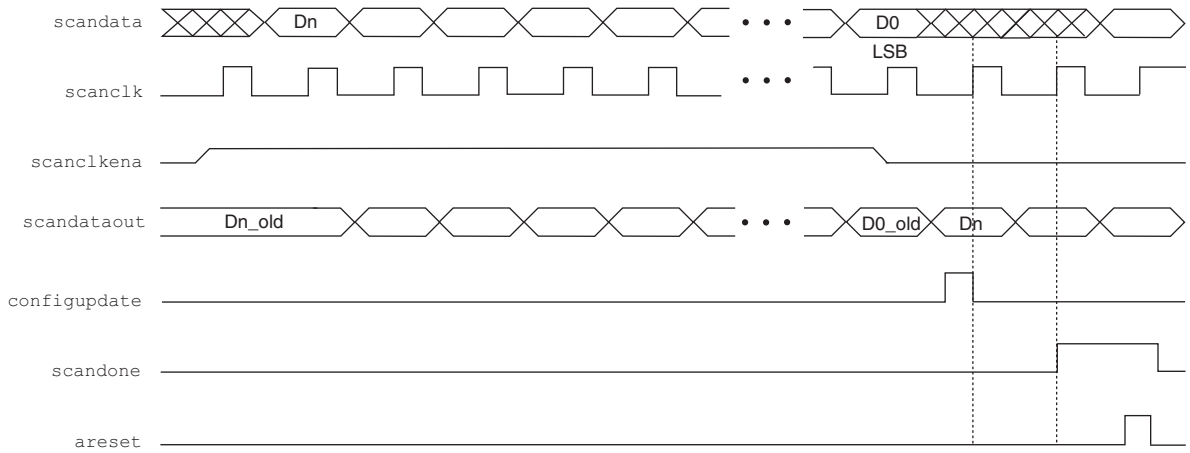
The counter settings are updated synchronously to the clock frequency of the individual counters. Therefore, not all counters update simultaneously.

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` (`Dn`).
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 deasserted 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 5–21 shows a functional simulation of the PLL reconfiguration feature.

Figure 5–21. 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, which 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 is set to 5 and 5 respectively, 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 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 III device family PLLs have a 144-bit scan chain.

Table 5–4 lists the number of bits for each component of the PLL.

Table 5–4. Cyclone III Device Family 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 5–4:

(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 5–22 shows the scan chain order of the PLL components.

Figure 5–22. PLL Component Scan Chain Order

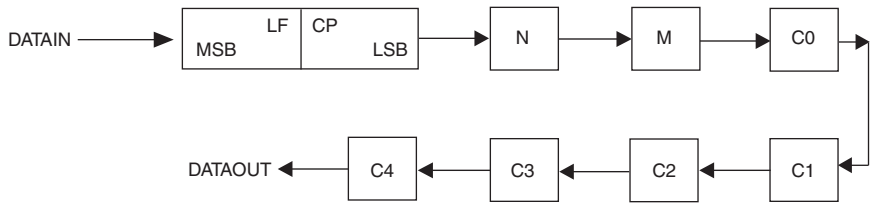
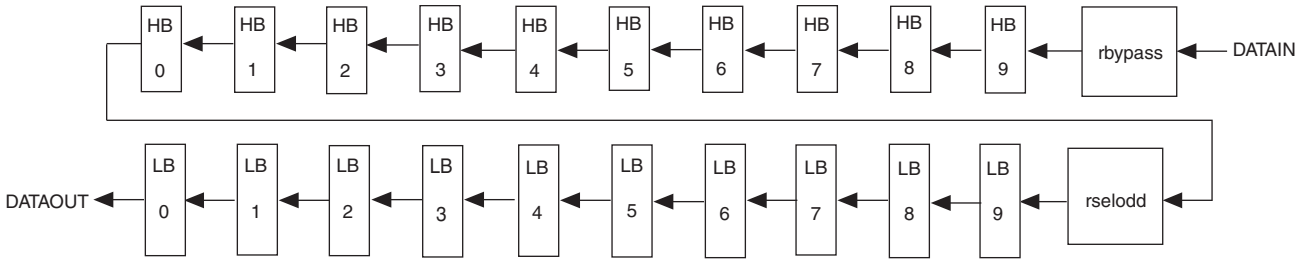


Figure 5–23 shows the scan chain bit order sequence for one PLL post-scale counter in Cyclone III device family PLLs.

Figure 5–23. Scan Chain Bit Order



For more information about the PLL scan chain, refer to the *Implementing PLL Reconfiguration in Cyclone III Devices* application note.

Charge Pump and Loop Filter

You can reconfigure the charge pump and loop filter settings to update the PLL bandwidth in real time. Table 5–5 through Table 5–7 list the possible settings for charge pump (ICP), loop filter resistor (R), and capacitor (C) values for Cyclone III device family PLLs.

Table 5–5. Charge Pump Bit Control

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

Table 5–6. 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 5–7. Loop Filter Control of High Frequency Capacitor

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

Bypassing PLL Counter

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

Table 5–8 lists the settings for bypassing the counters in Cyclone III device family PLLs.

Table 5–8. 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 5–8:

(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 5–9 lists the control signals that are used for dynamic phase shifting.

Table 5–9. 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

Table 5–9. Dynamic Phase Shifting Control Signals (Part 2 of 2)

Signal Name	Description	Source	Destination
PHASESTEP	Logic high enables dynamic phase shifting.	Logic array or I/O pins	PLL reconfiguration circuit
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. Deasserts on rising edge of SCANCLK.	PLL reconfiguration circuit	Logic array or I/O pins

Table 5–10 lists the PLL counter selection based on the corresponding PHASECOUNTERSELECT setting.

Table 5–10. Phase Counter Select Mapping

PHASECOUNTERSELECT [2]	[1]	[0]	Selects
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 step, you must perform the following procedures:

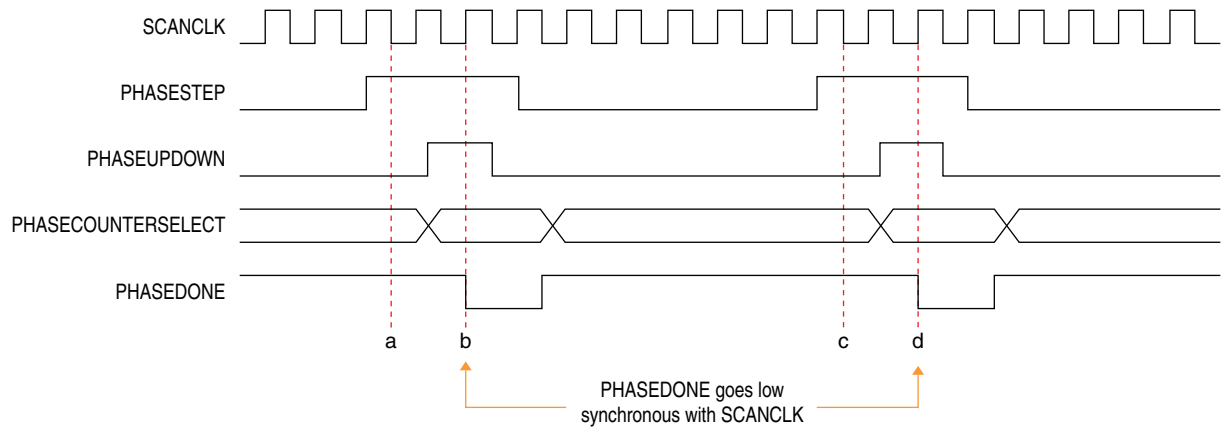
1. Set PHASEUPDOWN and PHASECOUNTERSELECT as required.
2. Assert PHASESTEP for at least two SCANCLK cycles. Each PHASESTEP pulse allows one phase shift.
3. Deassert PHASESTEP after PHASEDONE goes low.
4. Wait for PHASEDONE to go high.
5. Repeat steps 1 through 4 as many times as required to perform multiple 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 5–24 shows the dynamic phase shifting waveform.

Figure 5–24. Timing Diagram for 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 *Phase-Locked Loop Reconfiguration (ALTPLL_RECONFIG) Megafunction* user guide.

Spread-Spectrum Clocking

The Cyclone III device family 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. Cyclone III device family PLLs 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, which is specified in the fitter report. The Cyclone III device family cannot generate spread-spectrum signals internally.

PLL Specifications

For information about PLL specifications, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

Document Revision History

Table 5–11 lists the revision history for this document.

Table 5–11. Document Revision History (Part 1 of 2)

Date	Version	Changes
July 2012	4.1	Updated Figure 5–2.
November 2011	4.0	■ Minor edits to Equation 5–1 and Equation 5–2. ■ Updated Table 5–5. ■ Updated Figure 5–6, Figure 5–13, Figure 5–19, and Figure 5–24. ■ Updated "Clock Control Block" on page 5–4, "Manual Override" on page 5–20, "PLL Cascading" on page 5–24, and "Dynamic Phase Shifting" on page 5–31. ■ Minor text edits.
December 2009	3.2	Minor changes to the text.
July 2009	3.1	Made minor correction to the part number.
June 2009	3.0	■ Updated to include Cyclone III LS information. ■ Updated chapter part number. ■ Updated "Clock Networks" on page 5–1. ■ Updated Table 5–1 on page 5–2, Table 5–3 on page 5–9. ■ Updated "PLLs in the Cyclone III Device Family" on page 5–9. ■ Updated "PLL Reconfiguration Hardware Implementation" on page 5–25. ■ Updated "Spread-Spectrum Clocking" on page 5–32.
October 2008	2.1	■ Updated the "Dynamic Phase Shifting" and "Introduction" sections. ■ Updated Figure 5–2, Figure 5–8, and Figure 5–24. ■ Updated chapter to new template.
May 2008	2.0	■ Updated Figure 5–2 and added (Note 3). ■ Updated "clkena Signals" section. ■ Updated Figure 5–8 and added (Note 3). ■ Updated "PLL Control Signals" section. ■ Updated "PLL Cascading" section. ■ Updated "Cyclone III PLL Hardware Overview" section. ■ Updated Table 5–6, Table 5–3, Table 5–7. ■ Updated Figure 5–14. ■ Updated "PLL Cascading" section. ■ Updated "Clock Multiplication and Division" section. ■ Updated Step 6–32 in "PLL Reconfiguration Hardware Implementation" section. ■ Updated "Spread-Spectrum Clocking" section. ■ Updated Figure 5–29. ■ Updated "VCCD and GND" section. ■ Added "Power Consumption" section.
September 2007	1.2	■ Updated "Board Layout" section and removed Figure 5-30.

Table 5–11. Document Revision History (Part 2 of 2)

Date	Version	Changes
July 2007	1.1	■ Updated document with EP3C120 information. ■ Updated Table 5–1 and Table 5–4 with EP3C120 information. ■ Updated “Clock Control Block” section. ■ Updated locked signal information in “PLL Control Signals” section and added Figure 5–16. ■ Updated “Manual Override” section, updated “Manual Clock Switchover” section. ■ Added new “Programmable Bandwidth” section with Figure 5–21 and Figure 5–22. ■ Replaced Figure 5-30 with correct diagram. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

Section II. I/O Interfaces

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

This section includes the following chapters:

- Chapter 6, I/O Features in the Cyclone III Device Family
- Chapter 7, High-Speed Differential Interfaces in the Cyclone III Device Family
- Chapter 8, External Memory Interfaces in the Cyclone III Device Family

For information about the revision history for chapters in this section, refer to "Document Revision History" in each individual chapter.

6. I/O Features in the Cyclone III Device Family

CIII51007-3.4

This chapter describes the I/O features offered in the Cyclone® III device family (Cyclone III and Cyclone III LS devices).

The I/O capabilities of the Cyclone III device family 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 the Cyclone III device family 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. Altera's 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 contains the following sections:

- "Cyclone III Device Family I/O Elements" on page 6–1
- "I/O Element Features" on page 6–2
- "OCT Support" on page 6–7
- "I/O Standards" on page 6–11
- "Termination Scheme for I/O Standards" on page 6–13
- "I/O Banks" on page 6–16
- "Pad Placement and DC Guidelines" on page 6–21

Cyclone III Device Family I/O Elements

Cyclone III device family 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.

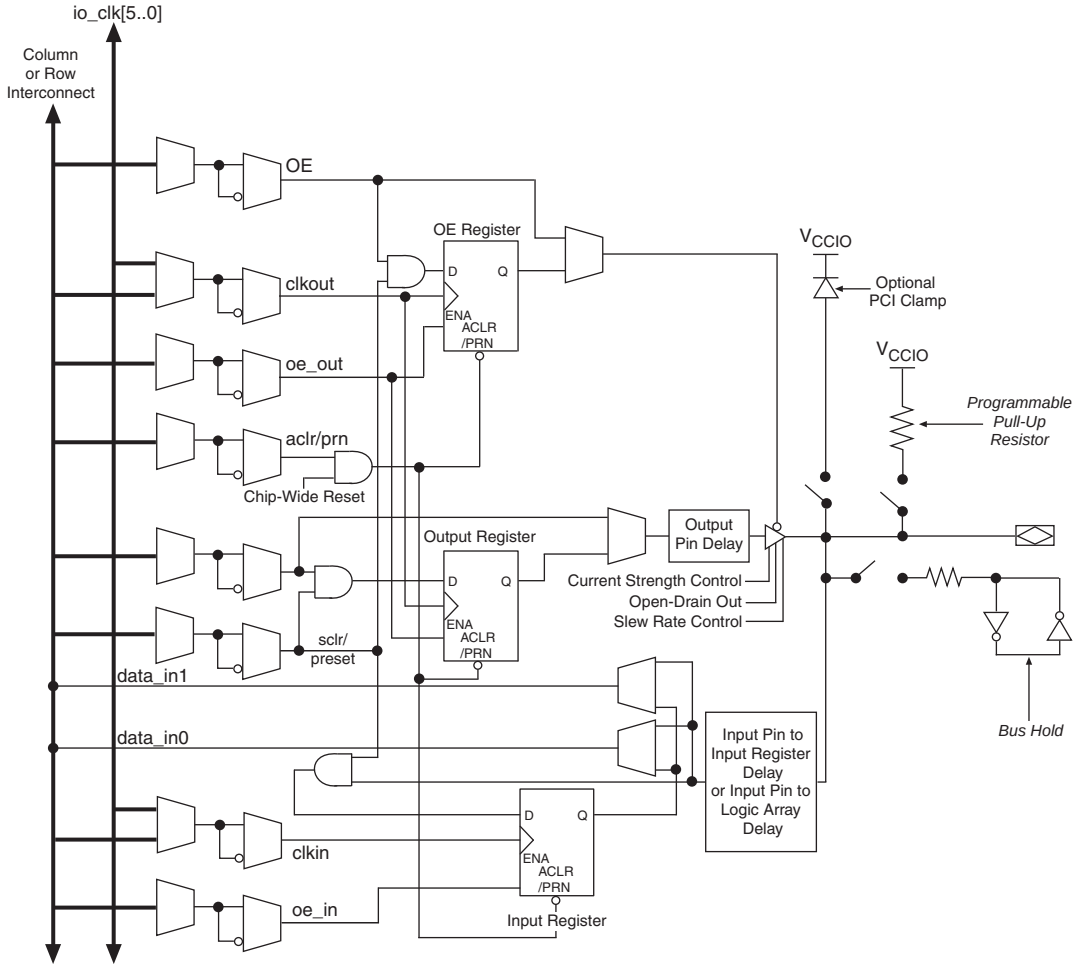
The IOE contains one input register, two output registers, and two output-enable (OE) registers. The two output registers and two OE registers are used for DDR applications. You can use input registers for fast setup times and output registers for fast clock-to-output times. Additionally, you can use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

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

ISO 9001:2008 Registered

Figure 6–1 shows the Cyclone III device family IOE structure.

Figure 6–1. Cyclone III Device Family IOE in a Bidirectional I/O Configuration



I/O Element Features

The Cyclone III device family IOE offers a range of programmable features for an I/O pin. These features increase the flexibility of I/O utilization and provide an alternative to reduce the usage of external discrete components to on-chip, such as a pull-up resistor and a diode.

Programmable Current Strength

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

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

Table 6–1 lists 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 is not available.

Table 6–1. Programmable Current Strength *(1)*

I/O Standard	I_{OH}/I_{OL} Current Strength Setting (mA)	
	Top and Bottom I/O Pins	Left and Right I/O Pins
1.2-V LVCMOS	2, 4, 6, 8, 10,12	2, 4, 6, 8,10
1.5-V LVCMOS	2, 4, 6, 8, 10, 12,16	2, 4, 6, 8, 10, 12,16
1.8-V LVTTL/LVCMOS	2, 4, 6, 8, 10, 12,16	2, 4, 6, 8, 10, 12,16
2.5-V LVTTL/LVCMOS	4, 8, 12,16	4, 8, 12,16
3.0-V LVCMOS	4, 8, 12,16	4, 8, 12,16
3.0-V LVTTL	4, 8, 12,16	4, 8, 12,16
3.3-V LVCMOS *(2)*	***2***	***2***
3.3-V LVTTL *(2)*	4, ***8***	4, ***8***
HSTL-12 Class I	8, 10,12	8, 10
HSTL-12 Class II	14	—
HSTL-15 Class I	8, 10, 12	8, 10, 12
HSTL-15 Class II	16	16
HSTL-18 Class I	8, 10, 12	8, 10, 12
HSTL-18 Class II	16	16
SSTL-18 Class I	8, 10, 12	8, 10, 12
SSTL-18 Class II	12, 16	12, 16
SSTL-2 Class I	8, 12	8, 12
SSTL-2 Class II	16	16
BLVDS	8, 12, 16	8, 12, 16

Notes to Table 6–1:

(1) The default 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 default current setting in the Quartus II software is highlighted in bold italic for **3.3-V LVTTL** and **3.3-V LVCMOS** I/O standards.

For information about how to interface the Cyclone III device family with 3.3-, 3.0-, or 2.5-V systems, refer to the guidelines provided in *AN 447: Interfacing Cyclone III and Cyclone IV Devices with 3.3/3.0/2.5-V LVTTL/LVCMOS I/O Systems*.

Slew Rate Control

The output buffer for each Cyclone III device family I/O pin provides optional programmable output slew-rate control. The Quartus II software allows three settings for programmable slew rate control—0, 1, and 2—where 0 is the slow slew rate and 2 is the fast slew rate. The default setting is 2. A faster slew rate provides high-speed transitions for high-performance systems. 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 or without 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**, and **3.3-V LVCMOS** I/O standards. Only fast slew rate (default) setting is available.

Open-Drain Output

The Cyclone III device family provides 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 III device family 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 III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

Programmable Pull-Up Resistor

Each Cyclone III device family 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, 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 remain tri-stated even if the programmable pull-up option is enabled.

Programmable Delay

The Cyclone III device family IOE includes programmable delays to ensure zero hold times, minimize setup times, increase clock-to-output times, or 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 6–2 lists the programmable delays for the Cyclone III device family.

Table 6–2. Cyclone III Device Family Programmable Delay Chain

Programmable Delays	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

The Cyclone III device family provides 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 **3.3-V LVTTL**, **3.3-V LVCMOS**, **3.0-V LVTTL**, **3.0-V LVCMOS**, **2.5-V LVTTL/LVCMOS**, **PCI**, or **PCI-X**, the PCI clamp diode is enabled by default in the Quartus II software.

For more information about the Cyclone III device family PCI-clamp diode 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*.

LVDS Transmitter Programmable Pre-Emphasis

The Cyclone III device family true **LVDS** transmitter supports programmable pre-emphasis. Programmable pre-emphasis is used to compensate the frequency-dependent attenuation of the transmission line. It increases the amplitude of the high-frequency components of the output signal, which cancels out much of the high-frequency loss of the transmission line.

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 as well.

For more information about the Cyclone III device family high-speed differential interface support, refer to the *High-Speed Differential Interfaces in the Cyclone III Device Family* chapter.

OCT Support

The Cyclone III device family features OCT to provide output impedance matching and termination capabilities. OCT helps to prevent reflections and maintain signal integrity while minimizing the need for external resistors in high pin-count ball grid array (BGA) packages.

The Cyclone III device family provides output driver on-chip impedance matching and on-chip series termination for single-ended outputs and bidirectional pins. For bidirectional pins, OCT is active only for output.

When using on-chip series termination, programmable current strength is not available.

There are two ways to implement OCT in the Cyclone III device family:

- OCT with calibration
- OCT without calibration

Table 6–3 lists the I/O standards that support output impedance matching and series termination.

Table 6–3. Selectable I/O Drivers for On-Chip Series Termination with and Without Calibration Setting

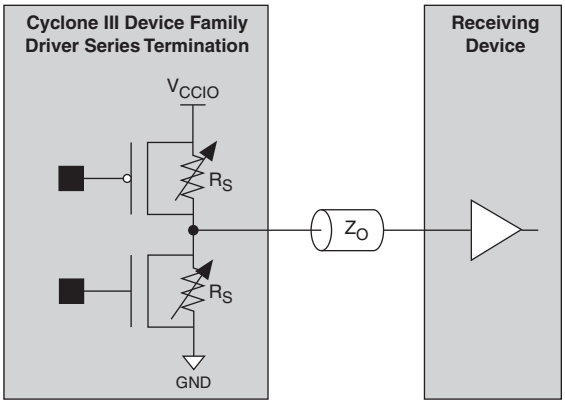
I/O Standard	On-Chip Series Termination with Calibration Setting, in ohms (Ω)		On-Chip Series Termination Without Calibration Setting, in ohms (Ω)	
	Row I/O	Column I/O	Row I/O	Column I/O
3.0-V LVTTL/LVCMOS	50, 25	50, 25	50, 25	50, 25
2.5-V LVTTL/LVCMOS	50, 25	50, 25	50, 25	50, 25
1.8-V LVTTL/LVCMOS	50, 25	50, 25	50, 25	50, 25
1.5-V LVCMOS	50, 25	50, 25	50, 25	50, 25
1.2-V LVCMOS	50	50, 25	50	50, 25
SSTL-2 Class I	50	50	50	50
SSTL-2 Class II	25	25	25	25
SSTL-18 Class I	50	50	50	50
SSTL-18 Class II	25	25	25	25
HSTL-18 Class I	50	50	50	50
HSTL-18 Class II	25	25	25	25
HSTL-15 Class I	50	50	50	50
HSTL-15 Class II	25	25	25	25
HSTL-12 Class I	50	50	50	50
HSTL-12 Class II	—	25	—	25

On-Chip Series Termination with Calibration

The Cyclone III device family supports on-chip series termination with calibration in all banks. The on-chip series termination calibration circuit compares the total impedance of the output buffer to the external 25-Ω ±1% or 50-Ω ±1% resistors connected to the RUP and RDN pins, and dynamically adjusts the output 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 output buffer.

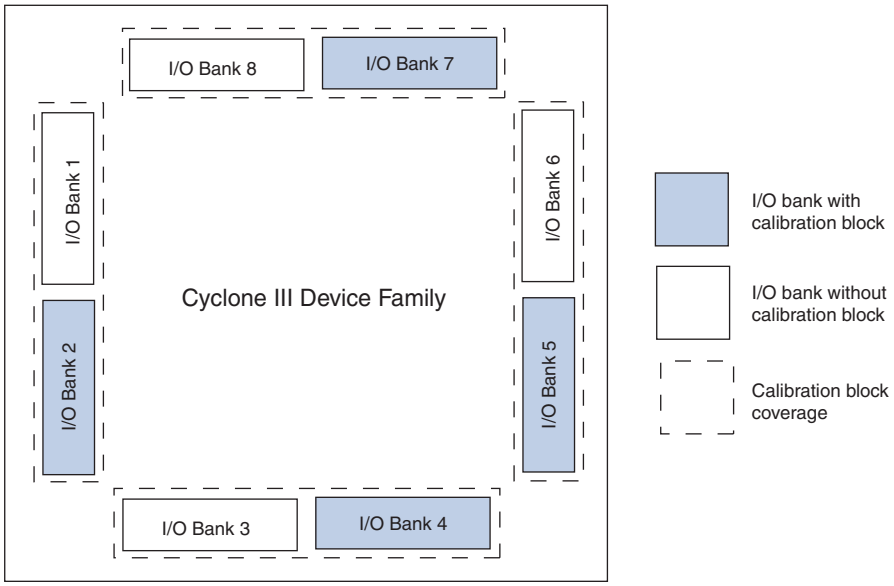
Figure 6–2. Cyclone III Device Family On-Chip Series Termination with Calibration



OCT with calibration is achieved using the OCT calibration block circuitry. There is one OCT calibration block in banks 2, 4, 5, and 7. 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}s, only the bank in which the calibration block resides can enable OCT calibration.

Figure 6–3 shows the top-level view of the OCT calibration blocks placement.

Figure 6–3. Cyclone III Device Family OCT Block Placement

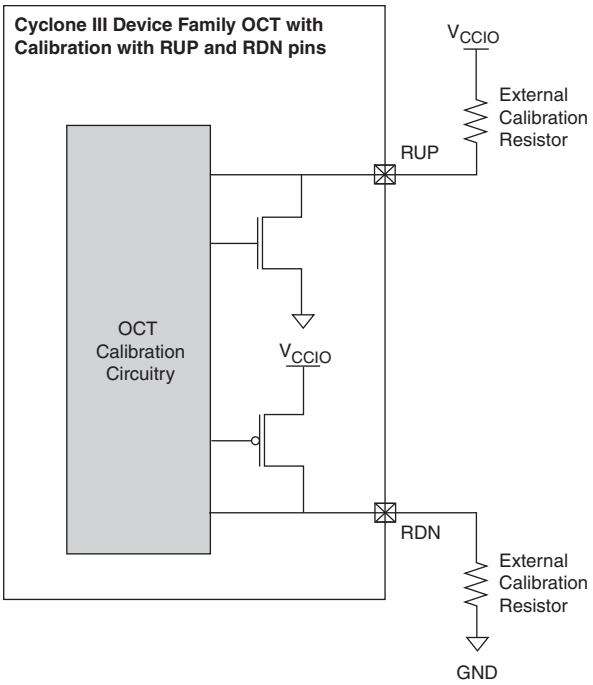


Each calibration block comes with a pair of RUP and RDN pins. When used for calibration, the RUP pin is connected to V_{CCIO} through an external 25-Ω ±1% or 50-Ω ±1% resistor for an on-chip series termination value of 25 Ω or 50 Ω, respectively. The RDN pin is connected to GND through an external 25-Ω ±1% or 50-Ω ±1% resistor for an on-chip series termination 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. For an estimate of the maximum possible current through the external calibration resistors, assume a minimum resistance of 0 Ω on the RUP and RDN pins during calibration.

Figure 6–4 shows the external calibration resistors setup on the RUP and RDN pins and the associated OCT calibration circuitry.

Figure 6–4. Cyclone III Device Family On-Chip Series Termination with Calibration Setup



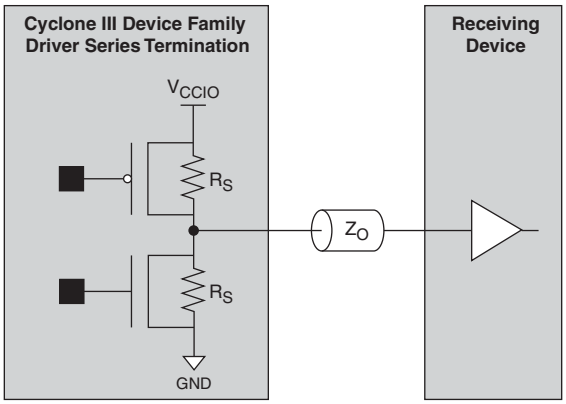
RUP and RDN pins go to a tri-state condition when calibration is completed or not running. These two pins are dual-purpose I/Os and function as regular I/Os if you do not use the calibration circuit.

On-Chip Series Termination Without Calibration

The Cyclone III device family supports driver impedance matching to the impedance of the transmission line, which is typically 25 Ω or 50 Ω. When used with the output drivers, OCT sets the output driver impedance to 25 Ω or 50 Ω. The Cyclone III device family also supports output driver series termination (R_S = 50 Ω) for **SSTL-2** and **SSTL-18**.

Figure 6–5 shows the single-ended I/O standards for OCT without calibration. The R_S shown is the intrinsic transistor impedance.

Figure 6–5. Cyclone III Device Family On-Chip Series Termination 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.

On-chip series termination is supported on any I/O bank. V_{CCIO} and V_{REF} must be compatible for all I/O pins to enable on-chip series termination 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} are not conflicting.

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 III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

I/O Standards

The Cyclone III device family supports multiple single-ended and differential I/O standards. Apart from 3.3-, 3.0-, 2.5-, 1.8-, and 1.5-V support, the Cyclone III device family also supports 1.2-V I/O standards.

Table 6–4 lists I/O standards supported by the Cyclone III device family and which I/O pins support them.

Table 6–4. Cyclone III Device Family Supported I/O Standards and Constraints (Part 1 of 2)

I/O Standard	Type	Standard Support	V_{CCIO} Level (in V)		Top and Bottom I/O Pins			Left and Right I/O Pins	
			Input	Output	CLK, DQS	PLL_OUT	User I/O Pins	CLK, DQS	User I/O Pins
3.3-V LVTTL, 3.3-V LVCMOS (1)	Single-ended	JESD8-B	3.3/3.0/2.5 (2)	3.3	✓	✓	✓	✓	✓
3.0-V LVTTL, 3.0-V LVCMOS (1)	Single-ended	JESD8-B	3.3/3.0/2.5 (2)	3.0	✓	✓	✓	✓	✓
2.5-V LVTTL / LVCMOS	Single-ended	JESD8-5	3.3/3.0/2.5 (2)	2.5	✓	✓	✓	✓	✓
1.8-V LVTTL / LVCMOS	Single-ended	JESD8-7	1.8/1.5 (2)	1.8	✓	✓	✓	✓	✓
1.5-V LVCMOS	Single-ended	JESD8-11	1.8/1.5 (2)	1.5	✓	✓	✓	✓	✓
1.2-V LVCMOS	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 (7)	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**	Differential (3)	JESD8-9A	—	2.5	—	✓	—	—	—
			2.5	—	✓	—	—	✓	—
Differential SSTL-18 Class I or **Class II**	Differential (3)	JESD815	—	1.8	—	✓	—	—	—
			1.8	—	✓	—	—	✓	—
Differential HSTL-18 Class I or **Class II**	Differential (3)	JESD8-6	—	1.8	—	✓	—	—	—
			1.8	—	✓	—	—	✓	—
Differential HSTL-15 Class I or **Class II**	Differential (3)	JESD8-6	—	1.5	—	✓	—	—	—
			1.5	—	✓	—	—	✓	—
Differential HSTL-12 Class I or **Class II**	Differential (3)	JESD8-16A	—	1.2	—	✓	—	—	—
			1.2	—	✓	—	—	✓	—
PPDS (4)	Differential	—	—	2.5	—	✓	✓	—	✓

Table 6–4. Cyclone III Device Family Supported I/O Standards and Constraints (Part 2 of 2)

I/O Standard	Type	Standard Support	V_{CCIO} Level (in V)		Top and Bottom I/O Pins			Left and Right I/O Pins	
			Input	Output	CLK, DQS	PLL_OUT	User I/O Pins	CLK, DQS	User I/O Pins
LVDS (8)	Differential	—	2.5	2.5	✓	✓	✓	✓	✓
RSDS and **mini-LVDS** (4)	Differential	—	—	2.5	—	✓	✓	—	✓
BLVDS (6)	Differential	—	2.5	2.5	—	—	✓	—	✓
LVPECL (5)	Differential	—	2.5	—	✓	—	—	✓	—

Notes to Table 6–4:

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

(2) The Cyclone III architecture supports the MultiVolt I/O interface feature that allows Cyclone III devices to interface with I/O systems that have different supply voltages.

(3) **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.

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

(5) **LVPECL** is only supported on clock inputs.

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

(7) Class I and Class II refer to output termination and do not apply to input. **1.2-V HSTL** input is supported at both column and row I/O regardless of class.

(8) True differential **LVDS**, **RSDS**, and **mini-LVDS** I/O standards are supported in left and right I/O pins while emulated differential **LVDS** (**LVDS_E_3R**), **RSDS** (**RSDS_E_3R**), and **mini-LVDS** (**LVDS_E_3R**) I/O standards are supported in both left and right, and top and bottom I/O pins.

The Cyclone III device family supports **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** and **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 6–6 and Figure 6–7.

Figure 6–6. Cyclone III Device Family HSTL I/O Standard Termination

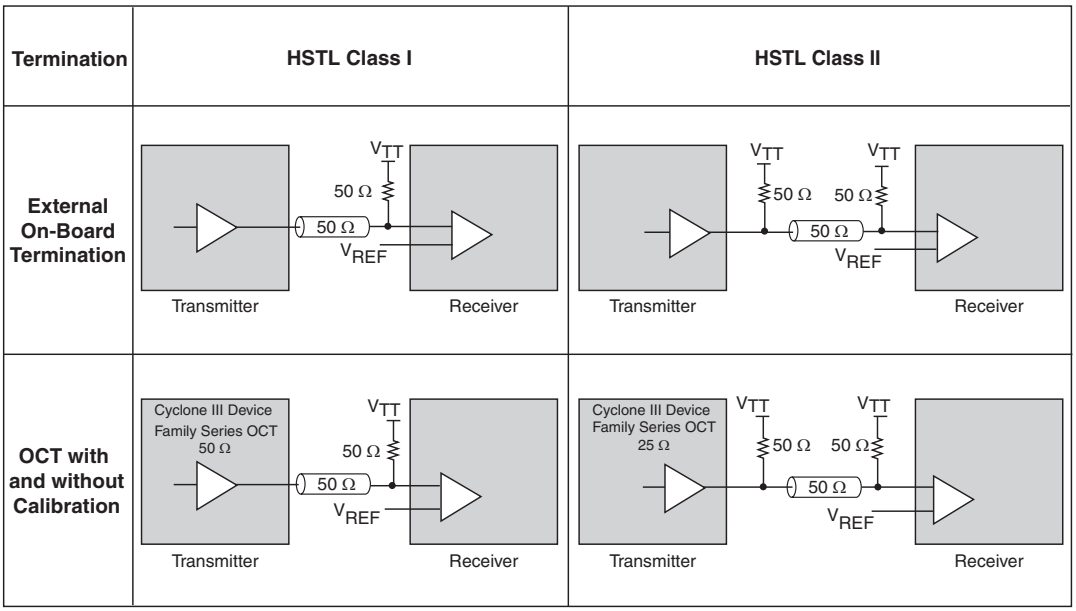
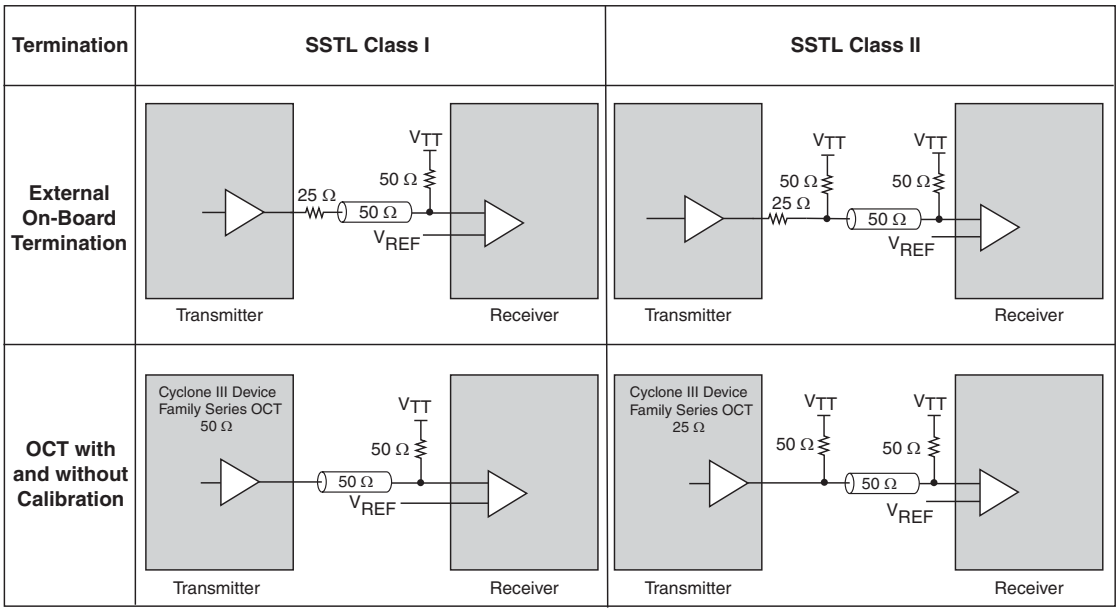


Figure 6–7. Cyclone III Device Family SSTL I/O Standard Termination



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 (Figure 6–8 and Figure 6–9).

The Cyclone III device family supports **differential SSTL-2** and **SSTL-18**, **differential HSTL-18**, **HSTL-15**, and **HSTL-12**, **PPDS**, **LVDS**, **RSDS**, **mini-LVDS**, and **differential LVPECL**.

Figure 6–8. Cyclone III Device Family Differential HSTL I/O Standard Termination

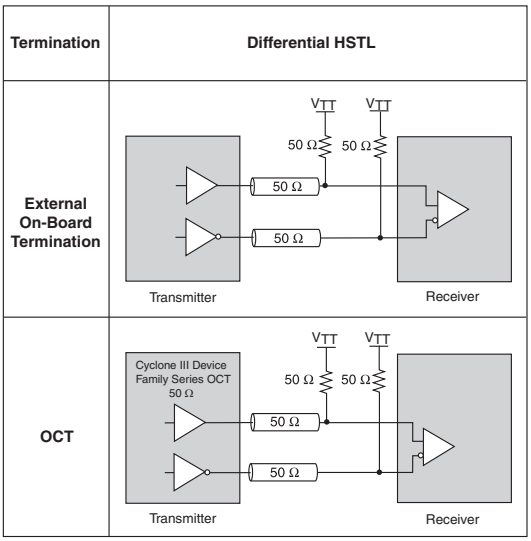
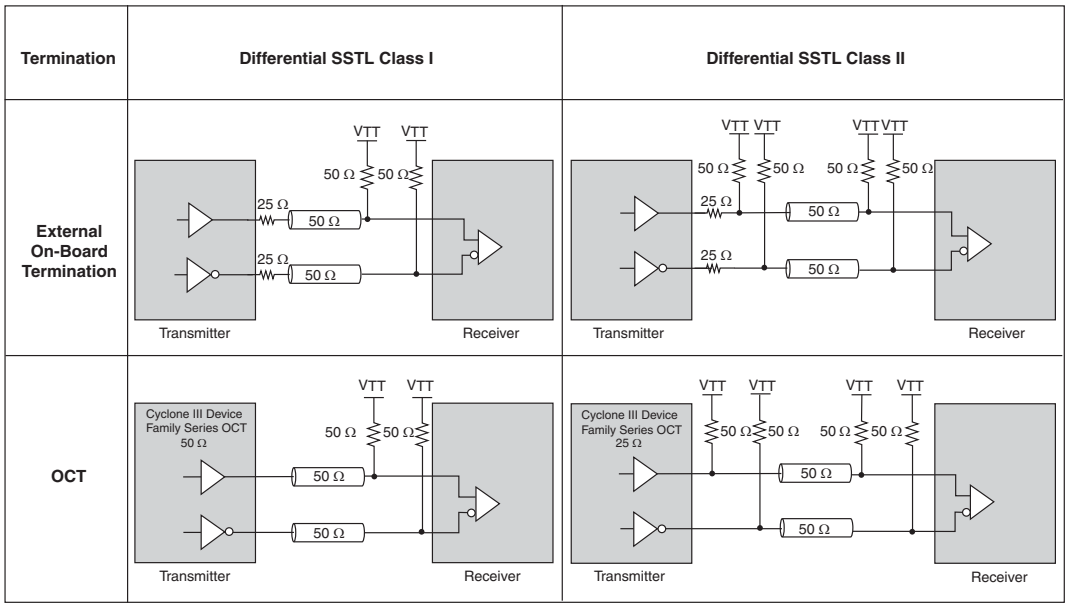


Figure 6–9. Cyclone III Device Family Differential SSTL I/O Standard Termination *(1)*



Note to Figure 6–9:

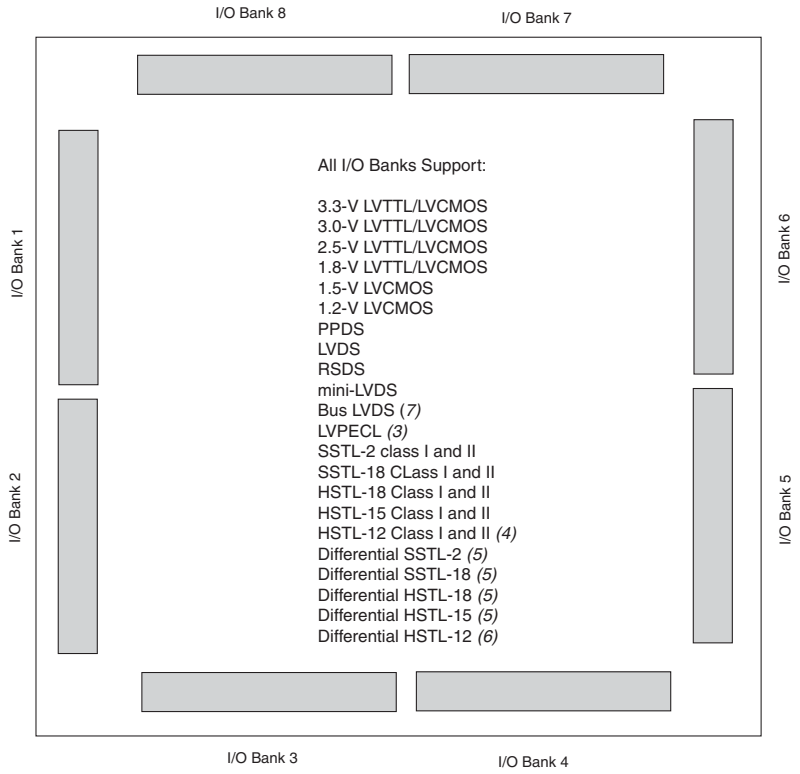
(1) Only Differential SSTL-2 I/O standard supports Class II output.

For information about the Cyclone III device family **differential PPDS**, **LVDS**, **mini LVDS**, **RSDS** I/O, and Bus LVDS (**BLVDS**) standard termination, refer to the *High-Speed Differential Interfaces in the Cyclone III Device Family* chapter.

I/O Banks

I/O pins on the Cyclone III device family are grouped together into I/O banks, and each bank has a separate power bus. Cyclone III and Cyclone III LS devices have eight I/O banks, as shown in Figure 6–10. 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.

Figure 6–10. Cyclone III Device Family I/O Banks *(1)*, *(2)*



Notes to Figure 6–10:

(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 the **LVDS** input buffer.

Table 6–5 lists the I/O standards supported when a pin is used as a regular I/O pin in the I/O banks of the Cyclone III device family.

Table 6–5. Cyclone III Device Family I/O Standards Support

I/O Standard	I/O Banks							
	1	2	3	4	5	6	7	8
3.3-V LVTTL/LVCMOS, **3.0-V LVTTL/LVCMOS**, **2.5-V LVTTL/LVCMOS**, **1.8-V LVTTL/LVCMOS**, **1.5-V LVCMOS**, **1.2V LVCMOS**, **3.0-V PCI/PCI-X**	✓	✓	✓	✓	✓	✓	✓	✓
SSTL-18 Class I/II, **SSTL-2 Class I/II**, **HSTL-18 Class I/II**, **HSTL-15 Class I/II**, **HSTL-12 Class I**	✓	✓	✓	✓	✓	✓	✓	✓
HSTL-12 Class II	—	—	✓	✓	—	—	✓	✓
Differential SSTL-2, **Differential SSTL-18**, **Differential HSTL-18**, **Differential HSTL-15**, **Differential HSTL-12**	*(1)*	*(1)*	*(1)*	*(1)*	*(1)*	*(1)*	*(1)*	*(1)*
PPDS *(2), (3)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*
LVDS *(2)*	✓	✓	✓	✓	✓	✓	✓	✓
BLVDS	✓	✓	✓	✓	✓	✓	✓	✓
RSDS and **mini-LVDS** *(2)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*	*(3)*
Differential LVPECL	*(4)*	*(4)*	*(4)*	*(4)*	*(4)*	*(4)*	*(4)*	*(4)*

Notes to Table 6–5:

(1) These differential I/O standards are supported only for clock inputs and dedicated `PLL_OUT` outputs.

(2) True differential (**PPDS**, **LVDS**, **mini-LVDS**, and **RSDS** I/O standards) outputs are supported in row I/O banks only. Differential outputs in column I/O banks require an external resistors network.

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

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

Each I/O bank of the Cyclone III device family 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 `VREFB1N0` group, `VREFB1N0` 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 III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

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

Table 6–6 lists the number of VREF pins in each I/O bank for Cyclone III and Cyclone III LS devices.

Table 6–6. Number of VREF Pins Per I/O Banks for Cyclone III and Cyclone III LS Devices (Part 1 of 2)

Family	Device	Package	Pin Count	I/O Banks							
				1	2	3	4	5	6	7	8
Cyclone III	EP3C5	EQFP	144	1	1	1	1	1	1	1	1
		MBGA	164	1	1	1	1	1	1	1	1
		FBGA	256	1	1	1	1	1	1	1	1
	EP3C10	EQFP	144	1	1	1	1	1	1	1	1
		MBGA	164	1	1	1	1	1	1	1	1
		FBGA	256	1	1	1	1	1	1	1	1
	EP3C16	EQFP	144	2	2	2	2	2	2	2	2
		MBGA	164	2	2	2	2	2	2	2	2
		PQFP	240	2	2	2	2	2	2	2	2
		FBGA	256	2	2	2	2	2	2	2	2
		FBGA	484	2	2	2	2	2	2	2	2
	EP3C25	EQFP	144	1	1	1	1	1	1	1	1
		PQFP	240	1	1	1	1	1	1	1	1
		FBGA	256	1	1	1	1	1	1	1	1
		FBGA	324	1	1	1	1	1	1	1	1
	EP3C40	PQFP	240	4	4	4	4	4	4	4	4
		FBGA	324	4	4	4	4	4	4	4	4
		FBGA	484	4	4	4	4	4	4	4	4
		FBGA	780	4	4	4	4	4	4	4	4
	EP3C55	FBGA	484	2	2	2	2	2	2	2	2
		FBGA	780	2	2	2	2	2	2	2	2
	EP3C80	FBGA	484	3	3	3	3	3	3	3	3
		FBGA	780	3	3	3	3	3	3	3	3
	EP3C120	FBGA	484	3	3	3	3	3	3	3	3
		FBGA	780	3	3	3	3	3	3	3	3

Table 6–6. Number of VREF Pins Per I/O Banks for Cyclone III and Cyclone III LS Devices (Part 2 of 2)

Family	Device	Package	Pin Count	I/O Banks							
				1	2	3	4	5	6	7	8
Cyclone III LS	EP3CLS70	UBGA	278	3	3	3	3	3	3	3	3
		FBGA	278	3	3	3	3	3	3	3	3
		FBGA	413	3	3	3	3	3	3	3	3
	EP3CLS100	UBGA	278	3	3	3	3	3	3	3	3
		FBGA	278	3	3	3	3	3	3	3	3
		FBGA	413	3	3	3	3	3	3	3	3
	EP3CLS150	FBGA	210	3	3	3	3	3	3	3	3
		FBGA	413	3	3	3	3	3	3	3	3
	EP3CLS200	FBGA	210	3	3	3	3	3	3	3	3
		FBGA	413	3	3	3	3	3	3	3	3

Each I/O bank of the Cyclone III device family has its own `VCCIO` pins. Each I/O bank can support only one V_{CCIO} setting from among 1.2, 1.5, 1.8, 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 III and Cyclone III LS 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 are provided in the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.
- 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 III and Cyclone III LS 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 III and Cyclone III LS 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 III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

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.

For more information about the Cyclone III device family I/O interface with 3.3-, 3.0-, or 2.5-V **LVTTL/LVCMOS** systems, refer to *AN 447: Interfacing Cyclone III and Cyclone IV Devices with 3.3/3.0/2.5-V LVTTL/LVCMOS I/O Systems*.

High-Speed Differential Interfaces

The Cyclone III device family can send and receive data through **LVDS** signals. For the **LVDS** transmitter and receiver, the input and output pins of the Cyclone III device family support serialization and deserialization through internal logic.

The **BLVDS** extends the benefits of **LVDS** to multipoint applications such as in 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 the Cyclone III device family support **BLVDS** for user I/O pins.

The reduced swing differential signaling (**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 point-to-point differential signaling (**PPDS**) standard is the next generation of the **RSDS** standard introduced by National Semiconductor Corporation. The Cyclone III device family meets the National Semiconductor Corporation PPDS Interface Specification and supports the **PPDS** standard for outputs only. All the I/O banks of the Cyclone III device family support the **PPDS** standard for output pins only.

You can use I/O pins and internal logic to implement the **LVDS** I/O receiver and transmitter in the Cyclone III device family. Cyclone III and Cyclone III LS devices do not contain dedicated serialization or deserialization circuitry. Therefore, shift registers, internal PLLs, and IOEs are used to perform serial-to-parallel conversions on incoming data and parallel-to-serial conversion on outgoing data.

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 top and bottom I/O banks.

For more information about the Cyclone III device family high-speed differential interface support, refer to the *High-Speed Differential Interfaces in the Cyclone III Device Family* chapter.

External Memory Interfacing

The Cyclone III device family supports I/O standards required to interface with a broad range of external memory interfaces, such as DDR SDRAM, DDR2 SDRAM, and QDRII SRAM.

For more information about the Cyclone III device family external memory interface support, refer to the *External Memory Interfaces in the Cyclone III Device Family* chapter.

Pad Placement and DC Guidelines

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 are dependent 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.

Document Revision History

Table 6–7 lists the revision history for this document.

Table 6–7. Document Revision History (Part 1 of 3)

Date	Version	Changes
July 2012	3.4	Updated OCT with or without calibration note in "Slew Rate Control" section,
December 2011	3.3	■ Updated Table 6–1 and Table 6–4. ■ Updated "Programmable Pull-Up Resistor" on page 6–5, "OCT Support" on page 6–7, and "I/O Standards" on page 6–11. ■ Updated hyperlinks. ■ Minor text edits.
December 2009	3.2	Minor changes to the text.
July 2009	3.1	Made minor correction to the part number.
June 2009	3.0	Updated to include Cyclone III LS information ■ Updated chapter part number. ■ Updated "Introduction" on page 6–1, "PCI-Clamp Diode" on page 6–6, "On-Chip Series Termination Without Calibration" on page 6–10, "I/O Standards" on page 6–11, "I/O Banks" on page 6–16, "High-Speed Differential Interfaces" on page 6–20, and "External Memory Interfacing" on page 6–20. ■ Updated Table 6–6 on page 6–18.

Table 6–7. Document Revision History (Part 2 of 3)

Date	Version	Changes
October 2008	2.1	■ Added *(Note 6)* to Table 6–5. ■ Updated the "I/O Banks" section. ■ Updated the "Differential Pad Placement Guidelines" section. ■ Updated the "VREF Pad Placement Guidelines" section. ■ Removed any mention of "RSDS and PPDS are registered trademarks of National Semiconductor" from chapter. ■ Updated chapter to new template.
May 2008	2.0	Changes include addition of BLVDS information. ■ Added an introduction to "I/O Element Features" section. ■ Updated "Slew Rate Control" section. ■ Updated "Programmable Delay" section. ■ Updated Table 6–1 with BLVDS information. ■ Updated Table 6–2. ■ Updated "PCI-Clamp Diode" section. ■ Updated "LVDS Transmitter Programmable Pre-Emphasis" section. ■ Updated "On-Chip Termination with Calibration" section and added new Figure 6–9. ■ Updated Table 6–3 title. ■ Updated Table 6–4 unit. ■ Updated "I/O Standards" section and Table 6–5 with BLVDS information and added *(Note 5)*. ■ Updated "Differential I/O Standard Termination" section with BLVDS information. ■ Updated "I/O Banks" section. ■ Updated *(Note 2)* and added *(Note 7)* and BLVDS information to Figure 6–15. ■ Updated *(Note 2)* and added BLVDS information to Table 6–6. ■ Added MBGA package information to Table 6–7. ■ Deleted Table 6-8. ■ Updated "High-Speed Differential Interfaces" section with BLVDS information. ■ Updated "Differential Pad Placement Guidelines" section and added new Figure 6–16. ■ Updated "VREF Pad Placement Guidelines" section and added new Figure 6–17. ■ Updated Table 6–11. ■ Added new "DCLK Pad Placement Guidelines" section. ■ Updated "DC Guidelines" section.

Table 6–7. Document Revision History (Part 3 of 3)

Date	Version	Changes
July 2007	1.1	■ Updated feetpara note in “Programmable Current Strength” section. ■ Updated feetpara note in “Slew Rate Control” section. ■ Updated feetpara note in “Open-Drain Output” section. ■ Updated feetpara note in “Bus Hold” section. ■ Updated feetpara note in “Programmable Pull-Up Resistor” section. ■ Updated feetpara note in “PCI-Clamp Diode” section. ■ Updated Figure 6–13. ■ Updated Figure 6–14 and added Note (1). ■ Updated “I/O Banks” section. ■ Updated Note (5) to Figure 6–15. ■ Updated “DDR/DDR2 and QDRII Pads” section and corrected ‘cms’ to ‘cmd’. ■ Updated Note 3 in Table 6-8. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

7. High-Speed Differential Interfaces in the Cyclone III Device Family

CIII51008-4.0

This chapter describes the high-speed differential I/O features and resources in the Cyclone III device family.

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 Altera® Cyclone® III device family (Cyclone III and Cyclone III LS devices) supports LVDS, BLVDS, reduced swing differential signaling (RSDS), mini-LVDS, and point-to-point differential signaling (PPDS).

This chapter contains the following sections:

- "High-Speed I/O Interface" on page 7–1
- "High-Speed I/O Standards Support" on page 7–7
- "True Output Buffer Feature" on page 7–15
- "High-Speed I/O Timing" on page 7–16
- "Design Guidelines" on page 7–17
- "Software Overview" on page 7–18

High-Speed I/O Interface

Cyclone III device family I/Os are separated into eight I/O banks, as shown in Figure 7–1. Each bank has an independent power supply. True output drivers for LVDS, RSDS, mini-LVDS, and PPDS are on the left and right I/O banks. These I/O standards are also supported on the top and bottom I/O banks using external resistors. On the left and 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 all I/O banks.

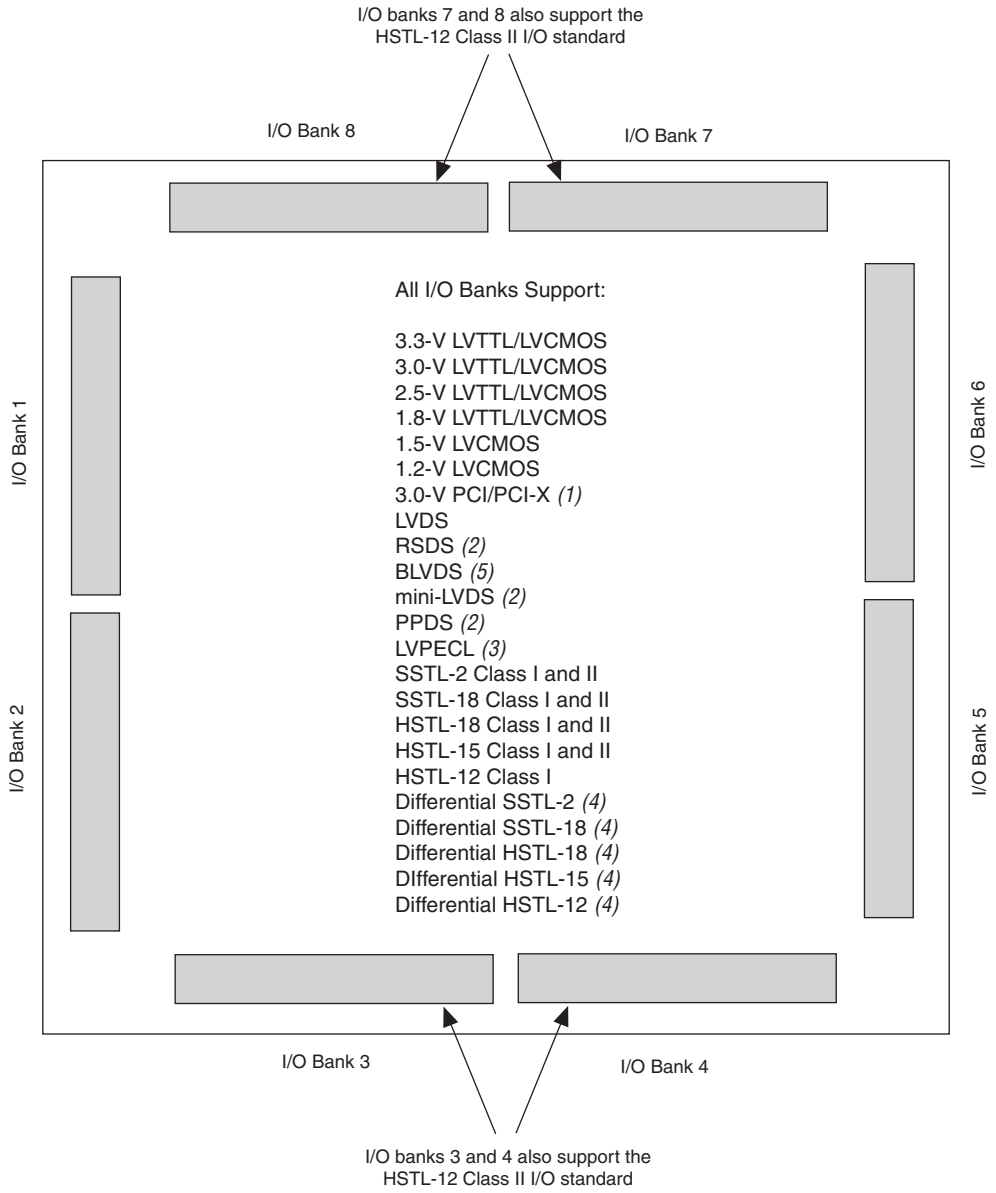
For more information about the location of Cyclone III device family true differential pins, refer to the Pin-Out Files for Altera Devices webpage on the Altera website.

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

Figure 7–1 shows the I/O banks of the Cyclone III device family.

Figure 7–1. Cyclone III Device Family I/O Banks



Notes to Figure 7–1:

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

(2) The RSDS, mini-LVDS, and PPDS I/O standards are only supported on output pins. These I/O standards are not supported on input pins.

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

(4) The differential SSTL-2, SSTL-18, HSTL-18, HSTL-15, and HSTL-12 I/O standards are only supported on dedicated 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.

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

Table 7–1 lists which I/O bank supports these I/O standards in the Cyclone III device family.

Table 7–1. Differential I/O Standards Supported in Cyclone III Device Family 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	Yes	Yes
	All	Three Resistors		
RSDS	1,2,5,6	Not Required	Yes	Not Supported
	3, 4, 7, 8	Three Resistors		
	All	Single Resistor		
mini-LVDS	1,2,5,6	Not Required	Yes	Not Supported
	All	Three Resistors		
PPDS	1,2,5,6	Not Required	Yes	Not Supported
	All	Three Resistors		
BLVDS (1)	All	Single Resistor	Yes	Yes
LVPECL (2)	All	NA	Not Supported	Yes
Differential SSTL-2 (3)	All	NA	Yes	Yes
Differential SSTL-18 (3)	All	NA	Yes	Yes
Differential HSTL-18 (3)	All	NA	Yes	Yes
Differential HSTL-15 (3)	All	NA	Yes	Yes
Differential HSTL-12 (3)	All	NA	Yes	Yes

Notes to Table 7–1:

(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 the Cyclone III device family. The Cyclone III device family does 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 7–2 lists the total number of supported row and column differential channels in the Cyclone III device family.

Table 7–2. Cyclone III Device Family Differential Channels (Part 1 of 2)

Cyclone III Device Family	Device	Package	Number of Differential Channels (1), (2)			
			User I/O	Clock Input	Clock Output	Total
Cyclone III Devices	EP3C5	E144	16	4	2	22
		F256	62	4	2	68
		M164	22	4	2	28
		U256	62	4	2	68
	EP3C10	E144	16	4	2	22
		F256	62	4	2	68
		M164	22	4	2	28
		U256	62	4	2	68
	EP3C16	E144	7	8	4	19
		E240	35	8	4	47
		F256	43	8	4	55
		F484	128	8	4	140
		M164	11	8	4	23
		U256	43	8	4	55
		U484	128	8	4	140
	EP3C25	E144	6	8	4	18
		E240	31	8	4	43
		F256	42	8	4	54
		F324	71	8	4	83
		U256	42	8	4	54
	EP3C40	E240	14	8	4	26
		F324	49	8	4	61
		F484	115	8	4	127
		F780	215	8	4	227
		U484	115	8	4	127
	EP3C55	F484	123	8	4	135
		F780	151	8	4	163
		U484	123	8	4	135
	EP3C80	F484	101	8	4	113
		F780	169	8	4	181
		U484	101	8	4	113
	EP3C120	F484	94	8	4	106
		F780	221	8	4	233

Table 7–2. Cyclone III Device Family Differential Channels (Part 2 of 2)

Cyclone III Device Family	Device	Package	Number of Differential Channels (1), (2)			
			User I/O	Clock Input	Clock Output	Total
Cyclone III LS Devices	EP3CLS70	U484	101	8	4	113
		F484	101	8	4	113
		F780	169	8	4	181
	EP3CLS100	U484	101	8	4	113
		F484	101	8	4	113
		F780	169	8	4	181
	EP3CLS150	F484	75	8	4	87
		F780	169	8	4	181
	EP3CLS200	F484	75	8	4	87
		F780	169	8	4	181

Notes to Table 7–2:

(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 the *I/O Features in the Cyclone III Device Family* chapter.

Table 7–3 lists the numbers of differential channels that can be migrated in Cyclone III devices.

Table 7–3. Cyclone III Devices Migratable Differential Channels (1) (Part 1 of 2)

Package Type	Migration Between Devices	Migratable Channels		
		User I/O	CLK	Total
E144	EP3C5 and EP3C10	16	4	20
	EP3C5 and EP3C16	5	4	9
	EP3C5 and EP3C25	6	4	10
	EP3C10 and EP3C16	5	4	9
	EP3C10 and EP3C25	6	4	10
	EP3C16 and EP3C25	5	8	13
M164	EP3C5 and EP3C10	22	4	26
	EP3C5 and EP3C16	11	4	15
	EP3C10 and EP3C16	19	4	14
Q240	EP3C16 and EP3C25	23	8	31
	EP3C16 and EP3C40	11	8	19
	EP3C25 and EP3C40	12	8	20
F256	EP3C5 and EP3C10	62	4	66
	EP3C5 and EP3C16	39	4	43
	EP3C5 and EP3C25	40	4	44
	EP3C10 and EP3C16	39	4	43
	EP3C10 and EP3C25	40	4	44
	EP3C16 and EP3C25	33	8	41

Table 7–3. Cyclone III Devices Migratable Differential Channels (1) (Part 2 of 2)

Package Type	Migration Between Devices	Migratable Channels		
		User I/O	CLK	Total
U256	EP3C5 and EP3C10	62	4	66
	EP3C5 and EP3C16	39	4	43
	EP3C5 and EP3C25	40	4	44
	EP3C10 and EP3C16	39	4	43
	EP3C10 and EP3C25	40	4	44
	EP3C16 and EP3C25	33	8	41
F324	EP3C25 and EP3C40	47	8	55
F484	EP3C16 and EP3C40	102	8	110
	EP3C16 and EP3C55	98	8	106
	EP3C16 and EP3C80	79	8	87
	EP3C16 and EP3C120	72	8	80
	EP3C40 and EP3C55	102	8	110
	EP3C40 and EP3C80	84	8	92
	EP3C40 and EP3C120	74	8	82
	EP3C55 and EP3C80	98	8	106
	EP3C55 and EP3C120	85	8	93
	EP3C80 and EP3C120	88	8	96
U484	EP3C16 and EP3C40	102	8	110
	EP3C16 and EP3C55	98	8	106
	EP3C16 and EP3C80	79	8	87
	EP3C40 and EP3C55	102	8	110
	EP3C40 and EP3C80	84	8	92
	EP3C55 and EP3C80	98	8	106
F780	EP3C40 and EP3C55	46	8	54
	EP3C40 and EP3C80	51	8	59
	EP3C40 and EP3C120	54	8	62
	EP3C55 and EP3C80	144	8	152
	EP3C55 and EP3C120	142	8	150
	EP3C80 and EP3C120	160	8	168

Note to Table 7–3:

(1) The migratable differential channels for Cyclone III devices are not directly migratable to Cyclone III LS devices and vice versa.

Table 7–4 lists the numbers of differential channels that can be migrated in Cyclone III LS devices.

Table 7–4. Cyclone III LS Devices Migratable Differential Channels *(1)*

Package Type	Migration between Devices	Migratable Channels			
		User I/O	Clock Input	Clock Output	Total
U484	EP3CLS70 and EP3CLS100	101	8	4	113
F484	EP3CLS70 and EP3CLS100	101	8	4	113
	EP3CLS70 and EP3CLS150	71	8	4	83
	EP3CLS70 and EP3CLS200	71	8	4	83
	EP3CLS100 and EP3CLS150	71	8	4	83
	EP3CLS100 and EP3CLS200	71	8	4	83
	EP3CLS150 and EP3CLS200	75	8	4	87
F780	EP3CLS70 and EP3CLS100	169	8	4	181
	EP3CLS70 and EP3CLS150	169	8	4	181
	EP3CLS70 and EP3CLS200	169	8	4	181
	EP3CLS100 and EP3CLS150	169	8	4	181
	EP3CLS100 and EP3CLS200	169	8	4	181
	EP3CLS150 and EP3CLS200	169	8	4	181

Note to Table 7–4:

(1) The migratable differential channels for Cyclone III devices are not directly migratable to Cyclone III LS devices and vice versa.

High-Speed I/O Standards Support

This section provides information about the high-speed I/O standards supported in the Cyclone III device family.

LVDS I/O Standard Support in the Cyclone III Device Family

The LVDS I/O standard is a high-speed, low-voltage swing, low power, and general purpose I/O interface standard. The Cyclone III device family meets 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 more information about the LVDS I/O standard electrical specifications in the Cyclone III device family, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

Designing with LVDS

Cyclone III device family I/O banks support LVDS I/O standard. The 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 7–2 shows a point-to-point LVDS interface using Cyclone III device family true LVDS output and input buffers.

Figure 7–2. Cyclone III Device Family LVDS Interface with True Output Buffer on the Left and Right I/O Banks

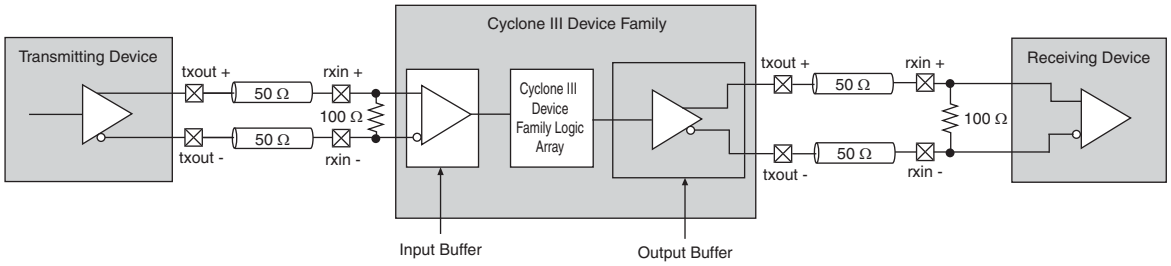
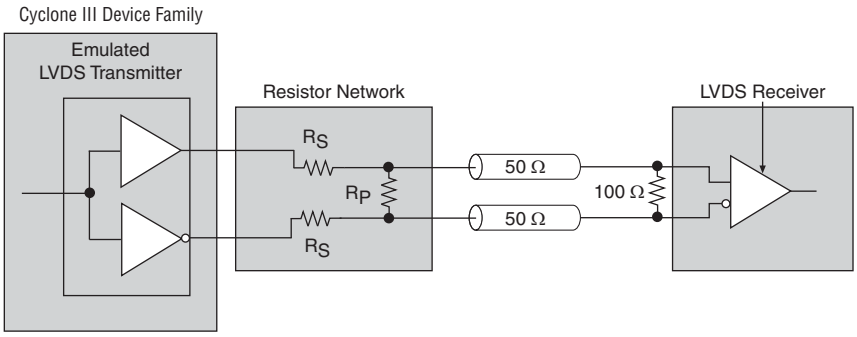


Figure 7–3 shows a point-to-point LVDS interface with Cyclone III device family LVDS using two single-ended output buffers and external resistors.

Figure 7–3. LVDS Interface with External Resistor Network on the Top and Bottom I/O Banks *(1)*



Note to Figure 7–3:

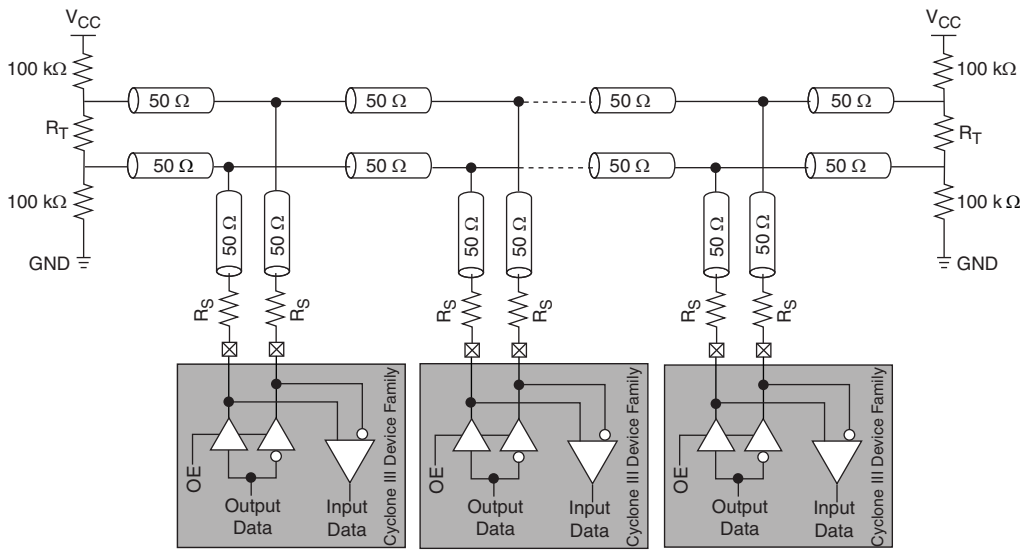
(1) R_S = 120 Ω; R_P = 170 Ω

BLVDS I/O Standard Support in the Cyclone III Device Family

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 7–4 shows a typical BLVDS topology with multiple transmitter and receiver pairs.

Figure 7–4. BLVDS Topology with Cyclone III Device Family Transmitters and Receivers



The BLVDS I/O standard is supported on all I/O banks of the Cyclone III device family. 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 about BLVDS I/O features and electrical specifications, refer to the *I/O Features in the Cyclone III Device Family* chapter in volume 1 of the *Cyclone III Device Handbook* and the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

For more information and design examples about implementing the BLVDS interfaces in the Cyclone III device family, refer to *AN 522: Implementing Bus LVDS Interface in Supported Altera Device Families*.

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.

RSDS, Mini-LVDS, and PPDS I/O Standard Support in the Cyclone III Device Family

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. The Cyclone III device family meets 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 more information about the Cyclone III device family RSDS, mini-LVDS, and PPDS output electrical specifications, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

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 III device family I/O banks support RSDS, mini-LVDS, and PPDS output standards. The left and 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 7–5 shows a RSDS, mini-LVDS, or PPDS interface with a true output buffer.

Figure 7–5. Cyclone III Device Family RSDS, Mini-LVDS, or PPDS Interface with True Output Buffer on the Left and Right I/O Banks

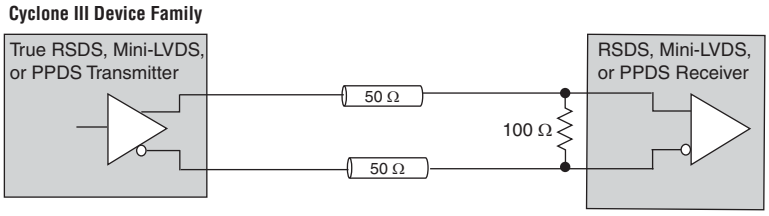
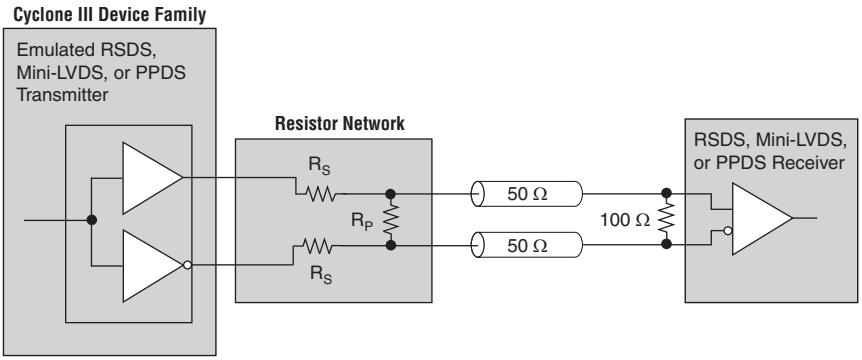


Figure 7–6 shows a RSDS, mini-LVDS, or PPDS interface with two singled-ended output buffers and external resistors.

Figure 7–6. RSDS, Mini-LVDS, or PPDS Interface with External Resistor Network on the Top and Bottom I/O Banks *(1)*



Note to Figure 7–6:

(1) R_S = 120 Ω; R_P = 170 Ω

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 7–1.

Equation 7–1.

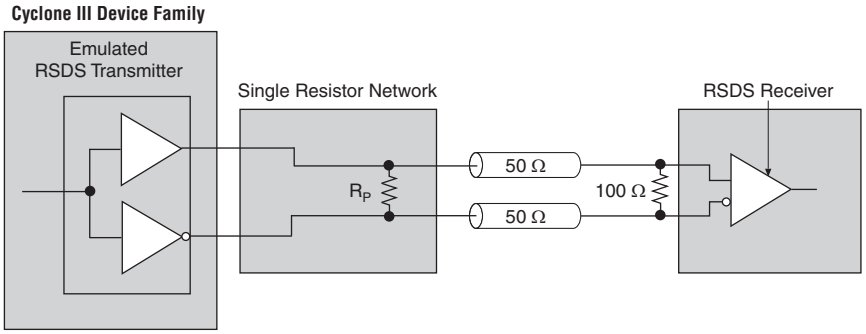
$$\frac{R_S \times \frac{R_P}{2}}{R_S + \frac{R_P}{2}} = 50\ \Omega$$

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

You can use a single external resistor instead of using three resistors in the resistor network for an RSDS interface, as shown in Figure 7–7. 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 7–7 shows the RSDS interface with a single resistor network on the top and bottom I/O banks.

Figure 7–7. RSDS Interface with Single Resistor Network on the Top and Bottom I/O Banks



Note to Figure 7–7:

(1) R_P = 100 Ω

LVPECL I/O Support in the Cyclone III Device Family

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. The Cyclone III device family supports 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 more information about the LVPECL I/O standard electrical specification, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

AC coupling is required when the LVPECL common mode voltage of the output buffer is higher than the Cyclone III device family LVPECL input common mode voltage.

Figure 7–8 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 III device family LVPECL input buffer specification (Figure 7–9).

Figure 7–8. LVPECL AC-Coupled Termination

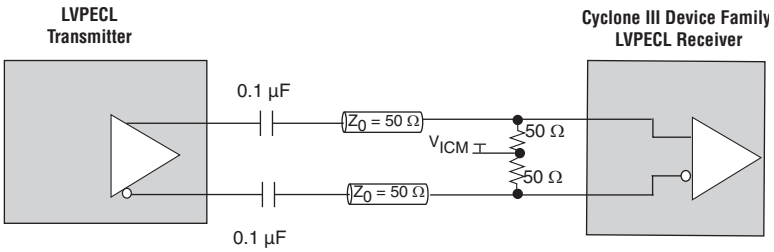
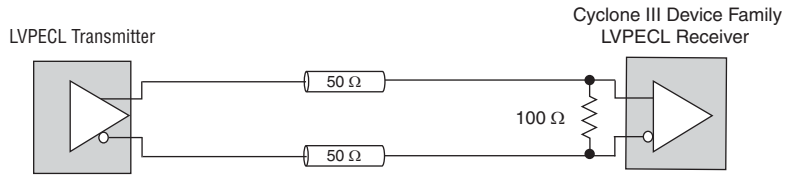


Figure 7–9 shows the LVPECL DC-coupled termination.

Figure 7–9. LVPECL DC-Coupled Termination



Differential SSTL I/O Standard Support in the Cyclone III Device Family

The differential SSTL I/O standard is a memory-bus standard used for applications such as high-speed DDR SDRAM interfaces. The Cyclone III device family supports 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 × V_{CCIO} to which termination resistors are connected.

For more information about the differential SSTL electrical specifications, refer to the *I/O Features in the Cyclone III Device Family* chapter and the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

Figure 7–10 shows the differential SSTL Class I interface.

Figure 7–10. Differential SSTL Class I Interface

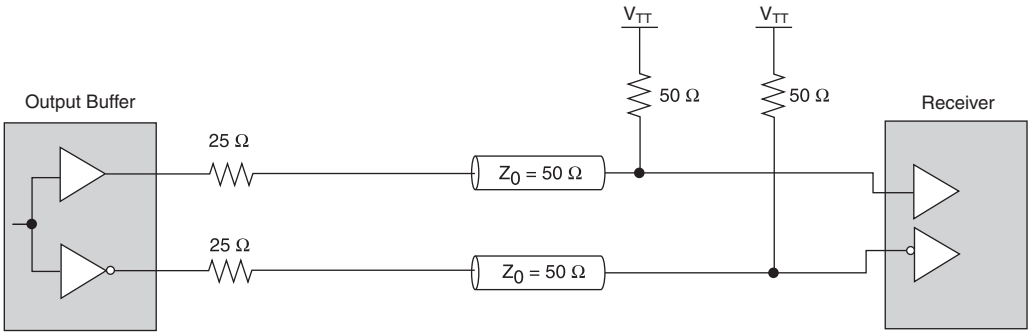
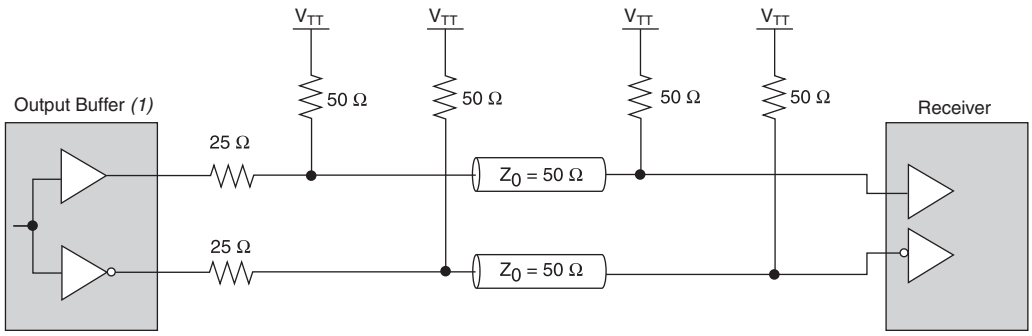


Figure 7–11 shows the differential SSTL Class II interface.

Figure 7–11. Differential SSTL Class II Interface



Note to Figure 7–11:

(1) PLL output clock pins do not support differential SSTL-18 Class II I/O standard.

Differential HSTL I/O Standard Support in the Cyclone III Device Family

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. The Cyclone III device family supports 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#_CLKOUT`*p* and `PLL#_CLKOUT`*n*), 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 × V_{CCIO} to which termination resistors are connected.

For more information about the differential HSTL signaling characteristics, refer to the *I/O Features in the Cyclone III Device Family*, *Cyclone III Device Data Sheet*, and *Cyclone III LS Device Data Sheet* chapters.

Figure 7–12 shows the differential HSTL Class I interface.

Figure 7–12. Differential HSTL Class I Interface

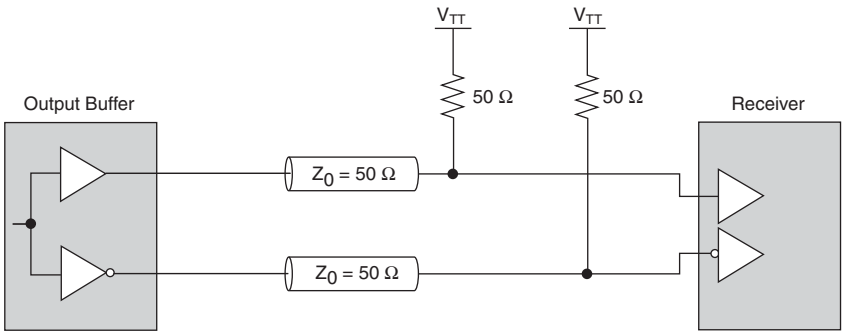
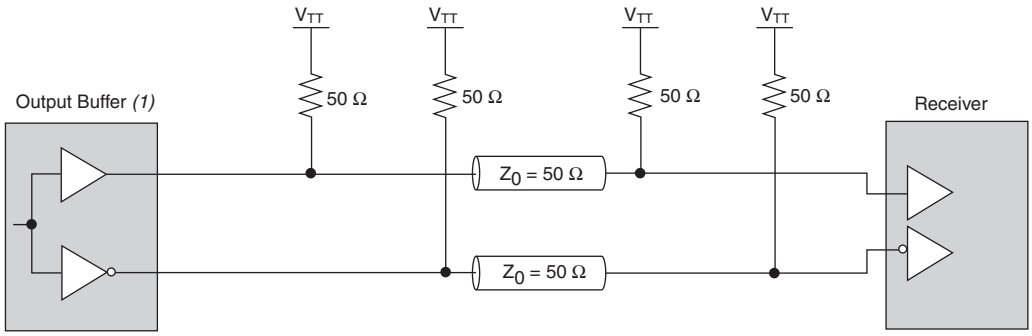


Figure 7–13 shows the differential HSTL Class II interface.

Figure 7–13. Differential HSTL Class II Interface



Note to Figure 7–13:

(1) PLL output clock pins do not support differential HSTL Class II I/O standard.

True Output Buffer Feature

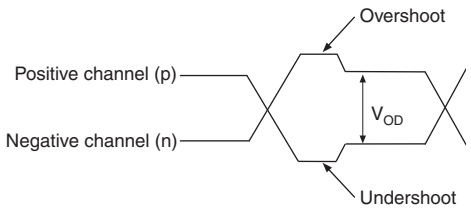
Cyclone III device family true differential transmitters offer programmable pre-emphasis—you can choose to turn it on or off. The default setting is on.

Programmable Pre-Emphasis

The programmable pre-emphasis boosts the high frequencies of the output signal to compensate the frequency-dependent 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.

Figure 7–14 shows the differential output signal with pre-emphasis.

Figure 7–14. 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 the Cyclone III device family. 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 the Cyclone III device family.

Table 7–5 lists the parameters of the timing diagram as shown in Figure 7–15.

Table 7–5. High-Speed I/O Timing Definitions

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.
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: $RSKM = \frac{(TUI - SW - TCCS}{2}$
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 7–5:

(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 7–15. High-Speed I/O Timing Diagram

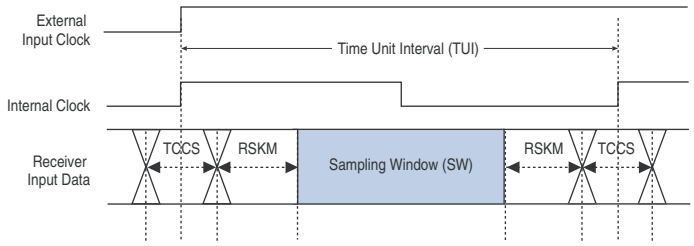
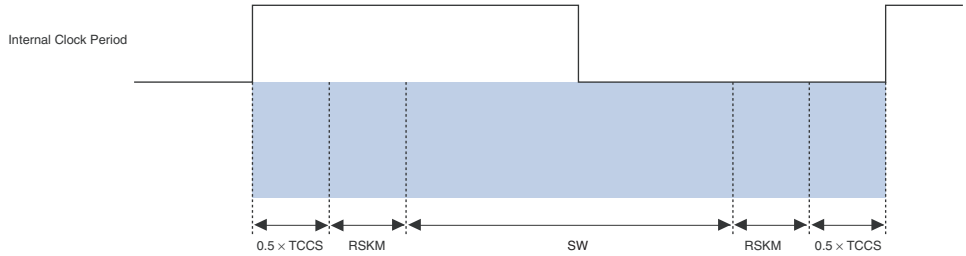


Figure 7–16 shows the Cyclone III device family high-speed I/O timing budget.

Figure 7–16. Cyclone III Device Family High-Speed I/O Timing Budget *(1)*



Note to Figure 7–16:

(1) The equation for the high-speed I/O timing budget is:
eriod = 0.5 × TCCS + RSKM + SW + RSKM + 0.5 × TCCS

For more information, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters in volume 2 of the *Cyclone III Device Handbook*.

Design Guidelines

This section provides guidelines for designing with the Cyclone III device family.

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.

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.

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*.

Board Design Considerations

This section explains how to achieve the optimal performance from the Cyclone III device family 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 the Cyclone III device family.

Use the following general guidelines for improved 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 transmitter-channel-to-channel skew (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 more information about PCB layout guidelines, refer to the *High-Speed Board Layout Guidelines* and *Guidelines for Designing High-Speed FPGA PCBs* application notes.

Software Overview

Cyclone III device family 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. The Cyclone III device family uses the I/O registers and LE registers to improve the timing performance and support the SERDES. Altera Quartus II software allows you to design your high-speed interfaces using the 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 III device family resources to create high-speed I/O interfaces in the most effective manner.

When you are using the Cyclone III device family with the ALTLVDS megafunction, the interface always sends the MSB of your parallel data first.

For more information about designing your high-speed I/O systems interfaces using the ALTLVDS megafunction, refer to the *LVDS SERDES Transmitter/Receiver (ALTLVDS_TX amd ALTLVDS_RX) Megafunction User Guide* and the *Quartus II Handbook*.

Document Revision History

Table 7–6 lists the revision history for this document.

Table 7–6. Document Revision History (Part 1 of 2)

Date	Version	Changes
December 2011	4.0	■ Updated Table 7–2. ■ Updated "Differential SSTL I/O Standard Support in the Cyclone III Device Family" on page 7–13, "Differential HSTL I/O Standard Support in the Cyclone III Device Family" on page 7–14, and "Differential Pad Placement Guidelines" on page 7–17. ■ Updated hyperlinks. ■ Minor text edits.
December 2009	3.2	Minor changes to the text.
July 2009	3.1	Made minor correction to the part number.
June 2009	3.0	Updated to include Cyclone III LS information ■ Updated chapter part number. ■ Updated "Introduction" on page 7–1, "High-Speed I/O Interface" on page 7–1, "High-Speed I/O Standards Support" on page 7–7, "LVDS I/O Standard Support in Cyclone III Family Devices" on page 7–7, "Designing with LVDS" on page 7–8, "BLVDS I/O Standard Support in Cyclone III Family Devices" on page 7–8, "RSDS, Mini-LVDS, and PPDS I/O Standard Support in Cyclone III Family Devices" on page 7–10, "LVPECL I/O Support in Cyclone III Family Devices" on page 7–12, "Differential SSTL I/O Standard Support in Cyclone III Family Devices" on page 7–13, and "Differential HSTL I/O Standard Support in Cyclone III Family Devices" on page 7–14. ■ Updated Figure 7–1 on page 7–2, Figure 7–4 on page 7–9, and Figure 7–5 on page 7–10. ■ Updated Table 7–1 on page 7–3, Table 7–2 on page 7–4, Table 7–3 on page 7–5, and Table 7–4 on page 7–7.
October 2008	1.3	■ Updated Table 7–2. ■ Updated Table 7–1. ■ Updated "BLVDS I/O Standard Support in Cyclone III Devices". ■ Updated "Software Overview". ■ Removed registered trademark symbols for RSDS and PPDS. ■ Removed any mention of "RSDS and PPDS are registered trademarks of National Semiconductor" in this chapter. ■ Updated chapter to new template.

Table 7–6. Document Revision History (Part 2 of 2)

Date	Version	Changes
May 2008	1.2	Changes include addition of BLVD information ■ Updated "Introduction" section with BLVDS information. ■ Updated Figure 7–1 with BLVDS information and added Note 5. ■ Updated Table 7–1 and added BLVDS information. ■ Updated "Cyclone III High-Speed I/O Banks" section with BLVDS information. ■ Updated Table 7–2 and 7–6. ■ Added new section "BLVDS I/O Standard Support in Cyclone III Devices". ■ Updated Note 4 to Figure 7–4. ■ Updated Note 1 to Figure 7–10. ■ Updated Note 1 to Figure 7–11. ■ Updated Note 1 to Figure 7–14. ■ Updated "Mini-LVDS I/O Standard Support in Cyclone III Devices" section. ■ Updated Note 1 to Figure 7–17. ■ Updated "LVPECL I/O Support in Cyclone III Devices" section. ■ Added new Figure 7–18.
July 2007	1.1	■ Added note that PLL output clock pins do not support Class II type of selected differential I/O standards. ■ Added Table 8–3 that lists the number of differential channels which are migratable across densities and packages. ■ Updated *(Note 4)* to Figure 7–1. ■ Updated *(Note 3)* to Table 7–1. ■ Added new Table 7–3. ■ Added *(Note 1)* to Figure 7–21. ■ Added *(Note 1)* to Figure 7–23. ■ Added chapter TOC and "Referenced Documents" section.
March 2007	1.0	Initial release.

8. External Memory Interfaces in the Cyclone III Device Family

CIII51009-3.1

In addition to an abundant supply of on-chip memory, Cyclone® III device family (Cyclone III and Cyclone III LS devices) can easily interface to a broad range of external memory, including DDR2 SDRAM, DDR SDRAM, and QDRII 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 III device family supports QDR II interfaces electrically, but Altera does not supply controller or physical layer (PHY) megafunctions for QDR II interfaces.

This chapter includes a description of the hardware interfaces for external memory interfaces available in Cyclone III device family.

This chapter contains the following sections:

- "Cyclone III Device Family Memory Interfaces Pin Support" on page 8–2
- "Cyclone III Device Family Memory Interfaces Features" on page 8–11

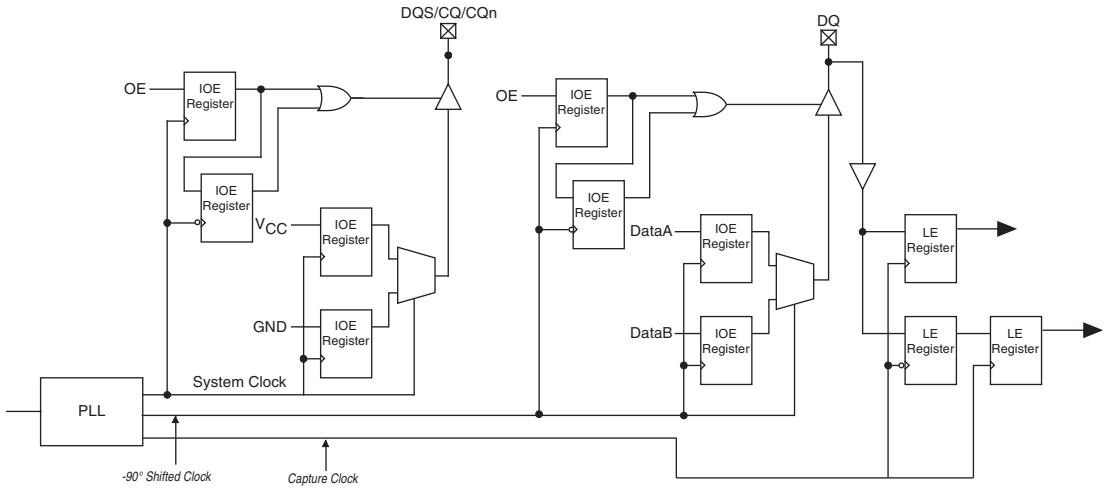
For more information about external memory system performance specifications, board design guidelines, timing analysis, simulation, and debugging information, refer to the External Memory Interfaces page.

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

ISO 9001:2008 Registered

Figure 8–1 shows the block diagram of a typical external memory interface data path in Cyclone III device family.

Figure 8–1. Cyclone III Device Family External Memory Data Path *(1)*



Note to Figure 8–1:

(1) All clocks shown here are global clocks.

Cyclone III Device Family Memory Interfaces Pin Support

Cyclone III device family uses 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 III device family supports all these different pins.

Data and Data Clock/Strobe Pins

Cyclone III device family 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 III device family supports 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 III device family `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 III device family, `DQS` is used only during write mode in DDR2 and DDR SDRAM interfaces. Cyclone III device family ignores `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 III device family does not support differential strobe pins, which is an optional feature in the DDR2 SDRAM device.

When you use the Altera Memory Controller MegaCore®, the PHY is instantiated for you. For more information about the memory interface data path, refer to the External Memory Interfaces page.

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 III device family 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 III device family can support DQ and DQS signals with DQ-bus modes of ×8, ×9, ×16, ×18, ×32, and ×36. DDR2 and DDR SDRAM interfaces use ×8 mode DQS group regardless of the interface width. For 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 which 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.

Table 8–1 lists the number of DQS or DQ groups supported on each side of the Cyclone III device only.

Table 8–1. Cyclone III Device DQS and DQ Bus Mode Support for Each Side of the Device (Part 1 of 4)

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP3C5	144-pin EQFP (1)	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Top (2)	1	0	0	0	—	—
		Bottom (3), (4)	1	0	0	0	—	—
	164-pin MBGA (1)	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Top (2)	1	0	0	0	—	—
		Bottom (3), (4)	1	0	0	0	—	—
	256-pin FineLine BGA/256-pin Ultra FineLine BGA (1)	Left (4), (5)	1	1	0	0	—	—
		Right (4), (6)	1	1	0	0	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—

Table 8–1. Cyclone III Device DQS and DQ Bus Mode Support for Each Side of the Device (Part 2 of 4)

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP3C10	144-pin EQFP (1)	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Top (2)	1	0	0	0	—	—
		Bottom (3), (4)	1	0	0	0	—	—
	164-pin MBGA (1)	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Top (2)	1	0	0	0	—	—
		Bottom (3), (4)	1	0	0	0	—	—
	256-pin FineLine BGA/256-pin Ultra FineLine BGA (1)	Left (4), (5)	1	1	0	0	—	—
		Right (4), (6)	1	1	0	0	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
EP3C16	144-pin EQFP (1)	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Top (2)	1	0	0	0	—	—
		Bottom (3), (4)	1	0	0	0	—	—
	164-pin MBGA (1)	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Top (2)	1	0	0	0	—	—
		Bottom (3), (4)	1	0	0	0	—	—
	240-pin PQFP (1)	Left (4), (7)	1	1	0	0	—	—
		Right (3), (4)	1	0	0	0	—	—
		Top	1	1	0	0	—	—
		Bottom	1	1	0	0	—	—
	256-pin FineLine BGA/256-pin Ultra FineLine BGA (1)	Left (4), (5)	1	1	0	0	—	—
		Right (4), (6)	1	1	0	0	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	484-pin FineLine BGA/484-pin Ultra FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1

Table 8–1. Cyclone III Device DQS and DQ Bus Mode Support for Each Side of the Device (Part 3 of 4)

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP3C25	144-pin EQFP (1)	Left	0	0	0	0	—	—
		Right	0	0	0	0	—	—
		Top (2)	1	0	0	0	—	—
		Bottom (3), (4)	1	0	0	0	—	—
	240-pin PQFP (1)	Left (4), (7)	1	1	0	0	—	—
		Right (3), (4)	1	0	0	0	—	—
		Top	1	1	0	0	—	—
		Bottom	1	1	0	0	—	—
	256-pin FineLine BGA/256-pin Ultra FineLine BGA (1)	Left (4), (5)	1	1	0	0	—	—
		Right (4), (6)	1	1	0	0	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	324-pin FineLine BGA (1)	Left	2	2	1	1	—	—
		Right (8)	2	2	1	1	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
EP3C40	240-pin PQFP	Left (4), (7)	1	1	0	0	0	0
		Right (3), (4)	1	0	0	0	0	0
		Top	1	1	0	0	0	0
		Bottom	1	1	0	0	0	0
	324-pin FineLine BGA	Left	2	2	1	1	0	0
		Right (8)	2	2	1	1	0	0
		Top	2	2	1	1	0	0
		Bottom	2	2	1	1	0	0
	484-pin FineLine BGA/484-pin Ultra FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1

Table 8–1. Cyclone III Device DQS and DQ Bus Mode Support for Each Side of the Device (Part 4 of 4)

Device	Package	Side	Number ×8 Groups	Number ×9 Groups	Number ×16 Groups	Number ×18 Groups	Number ×32 Groups	Number ×36 Groups
EP3C55	484-pin FineLine BGA/484-pin Ultra FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1
EP3C80	484-pin FineLine BGA/484-pin Ultra FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1
EP3C120	484-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	4	2	2	2	1	1
		Bottom	4	2	2	2	1	1
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1

Notes to Table 8–1:

(1) This device package does not support ×32 or ×36 mode.

(2) For the top side of the device, RUP, RDN, 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 the RUP and RDN pins for on-chip termination (OCT) calibration or if you are using PLLCLKOUT3n and PLLCLKOUT3p.

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

(4) The RUP and RDN pins are shared with the DQ pins. You cannot use these groups if you are using the RUP and RDN pins for OCT calibration.

(5) The ×8 DQ group can be formed in Bank 2.

(6) The ×8 DQ group can be formed in Bank 5.

(7) There is no DM and BWS# pins support for these groups.

(8) The RUP pin is shared with the DQ pin to gain ×9 or ×18 DQ group. You cannot use these groups if you are using the RUP and RDN pins for OCT calibration.

Table 8–2 lists the numbers of DQS or DQ groups supported on each side of the Cyclone III LS device only.

Table 8–2. Cyclone III LS Device DQS and DQ Bus Mode Support for Each Side of the Device

Device	Package	Side	Number of ×8 Groups	Number of ×9 Groups	Number of ×16 Groups	Number of ×18 Groups	Number of ×32 Groups	Number of ×36 Groups
EP3CLS70	484-pin FineLine BGA/ 484-pin Ultra FineLine BGA (1)	Left	2	2	1	1	—	—
		Right	2	2	1	1	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1
EP3CLS100	484-pin FineLine BGA/ 484-pin Ultra FineLine BGA (1)	Left	2	2	1	1	—	—
		Right	2	2	1	1	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1
EP3CLS150	484-pin FineLine BGA (1)	Left	2	2	1	1	—	—
		Right	2	2	1	1	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1
EP3CLS200	484-pin FineLine BGA (1)	Left	2	2	1	1	—	—
		Right	2	2	1	1	—	—
		Top	2	2	1	1	—	—
		Bottom	2	2	1	1	—	—
	780-pin FineLine BGA	Left	4	2	2	2	1	1
		Right	4	2	2	2	1	1
		Top	6	2	2	2	1	1
		Bottom	6	2	2	2	1	1

Note to Table 8–2:

(1) This device package does not support x32 or 36 mode.

For more information about device package outline, refer to the Package and Thermal Resistance page.

DQS pins are listed in the Cyclone III and Cyclone III LS 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), left (L) 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), left (L) 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 III and Cyclone III LS pin tables; for example:

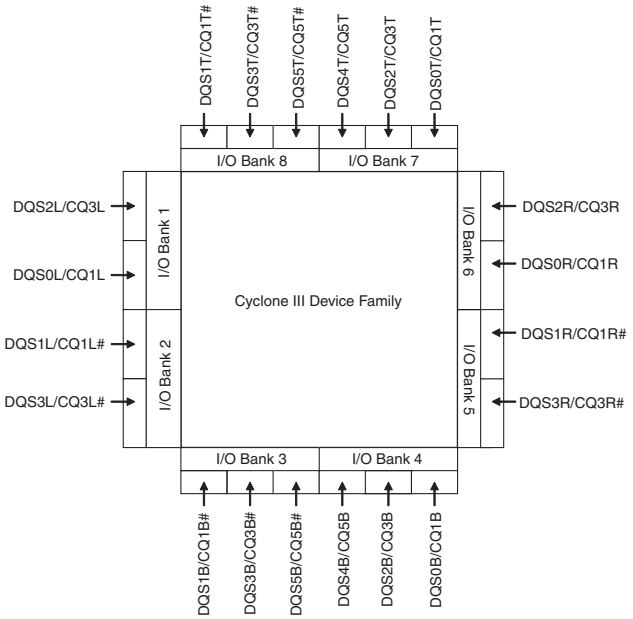
- For DDR2 or DDR SDRAM, ×8 DQ group DQ3B[7:0] pins are associated with the DQS3B pin (same 3B group index)
- For QDR II SRAM, ×9 Q read-data group DQ3L[8..0] pins are associated with DQS2L/CQ3L and DQS3L/CQ3L# pins (same 3L group index)

The Quartus® II software issues an error message if a DQ group is not placed properly with its associated DQS.

Figure 8–2 shows the location and numbering of the DQS, DQ, or CQ# pins in the Cyclone III device family I/O banks.

For maximum timing performance, Altera recommends that the data groups for external memory interfaces must always be within the same side of a device.

Figure 8–2. DQS, CQ, or CQ# Pins in Cyclone III Device Family I/O Banks *(1)*

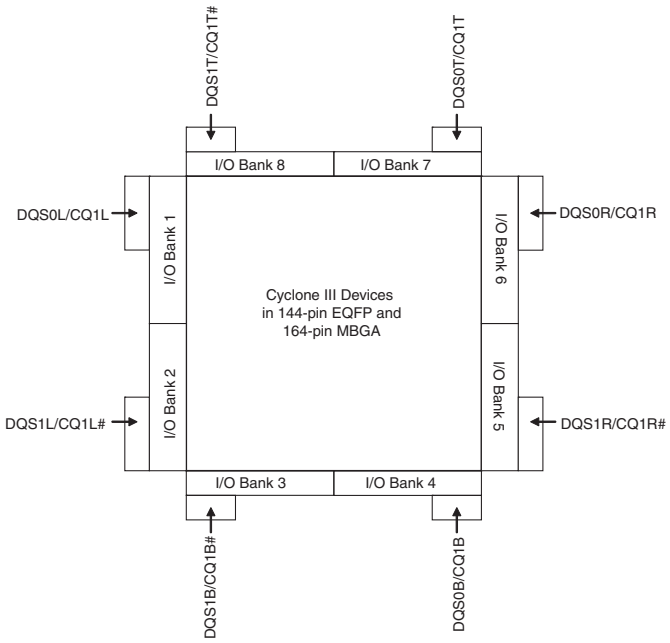


Note to Figure 8–2:

(1) The DQS, CQ, or CQ# pin locations in this diagram apply to all packages in Cyclone III device family except devices in 144-pin EQFP and 164-pin MBGA.

Figure 8–3 shows the location and numbering of the DQS, DQ, or CQ# pins in I/O banks of the Cyclone III device in the 144-pin EQFP and 164-pin MBGA packages only.

Figure 8–3. DQS, CQ, or CQ# Pins for Devices in the 144-Pin EQFP and 164-Pin MBGA Packages



In Cyclone III device family, 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 III device family supports 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 III device family because the parity pins are treated and configured similar 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 III device family, 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 III device family. 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 III device family to generate the address and control or command signals to the memory device.

Cyclone III device family does 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 III device family.

For more information about CK/CK# pins placement, refer to the "Pin Connection Guidelines Tables" section in the *Planning Pin and FPGA Resources* chapter of the *External Memory Interface Handbook*.

Cyclone III Device Family Memory Interfaces Features

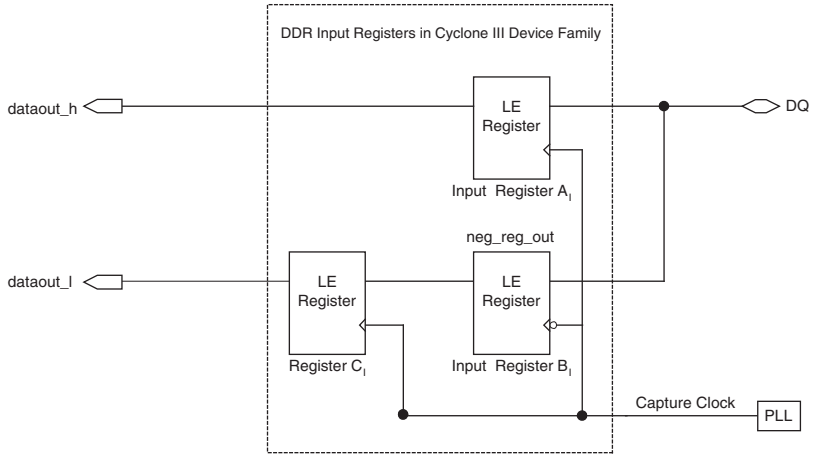
This section describes Cyclone III device family memory interfaces, including DDR input registers, DDR output registers, OCT, and phase-lock loops (PLLs).

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 8–4 shows Cyclone III device family DDR input registers.

Figure 8–4. Cyclone III Device Family DDR Input Registers



The DDR data is first fed to two registers, input register A_I and input register B_I.

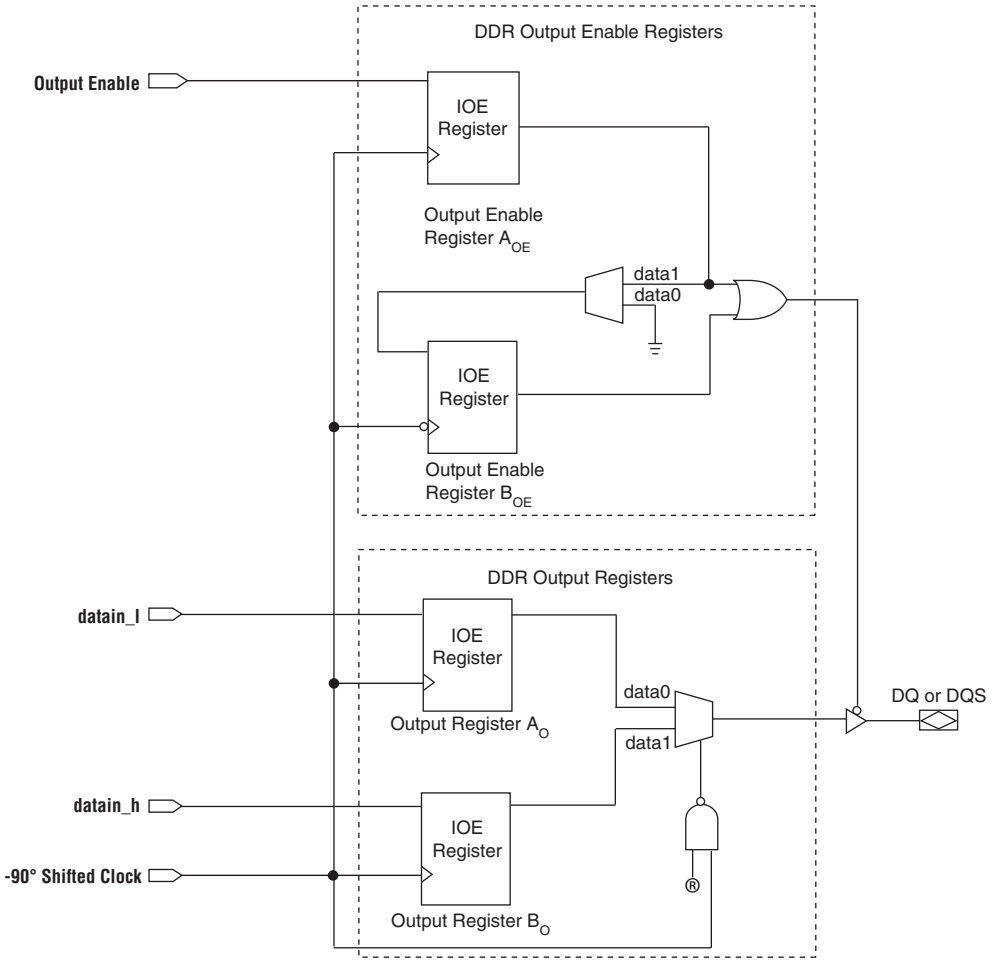
- Input register A_I captures the DDR data present during the rising edge of the clock
- Input register B_I captures the DDR data present during the falling edge of the clock
- Register C_I aligns the data before it is synchronized with the system clock

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 III device family; 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 8–5 shows how Cyclone III device family dedicated write DDIO block is implemented in the I/O element (IOE) registers.

Figure 8–5. Cyclone III Device Family Dedicated Write DDIO



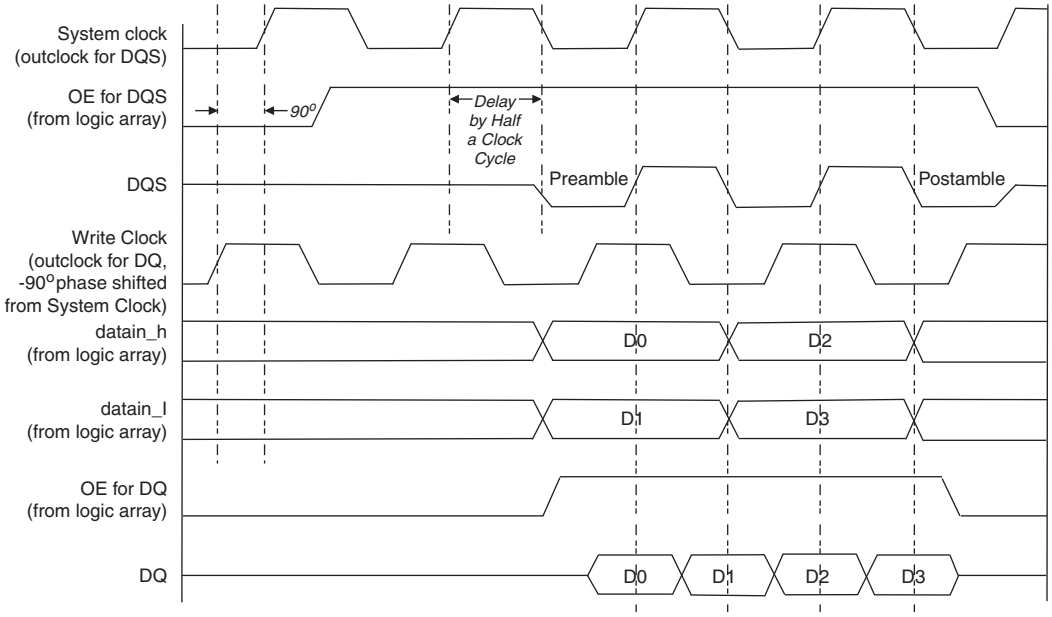
The two DDR output registers are located in the I/O element (IOE) block. Two serial data streams routed through `datain_l` and `datain_h`, are fed into two registers, `output register Ao` and `output register Bo`, respectively, on the same clock edge. The output from `output register Ao` is captured 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 III device family IOE registers, refer to the *I/O Features in the Cyclone III Device Family* chapter.

Figure 8–6 shows how the second output enable register extends the DQS high-impedance state by half a clock cycle during a write operation.

Figure 8–6. Extending the OE Disable by Half a Clock Cycle for a Write Transaction *(1)*



Note to Figure 8–6:

(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

Cyclone III device family supports 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 Cyclone III device family OCT calibration block, refer to the *Cyclone III 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.

For more information about the usage of PLL outputs by the ALTMEMPHY megafunction, refer to the External Memory Interfaces page.

For more information about Cyclone III device family PLL, refer to the *Clock Networks and PLLs in the Cyclone III Device Family* chapter.

Document Revision History

Table 8–3 lists the revision history for this document.

Table 8–3. Document Revision History

Date	Version	Changes
July 2012	3.1	Finalized Table 8–2.
December 2011	3.0	■ Updated “Data and Data Clock/Strobe Pins” on page 8–2 and “Memory Clock Pins” on page 8–10. ■ Updated hyperlinks. ■ Minor text edits.
January 2010	2.3	■ Removed Tables 8-1, 8-2, 8-3, and 8-4. ■ Changed links to reference *Literature: External Memory Interfaces*.
December 2009	2.2	Minor changes to the text.
July 2009	2.1	Made minor correction to the part number.
June 2009	2.0	■ Updated chapter part number. ■ Updated “Introduction” on page 8–1. ■ Updated Table 8–1 on page 8–1, Table 8–2 on page 8–2, Table 8–3 on page 8–3, Table 8–4 on page 8–4, and Table 8–5 on page 8–7. Updated notes to Table 8–6 on page 8–10. Updated “Data and Data Clock/Strobe Pins” on page 8–5. ■ Updated note to Figure 8–2 on page 8–12. ■ Updated “Optional Parity, DM, and Error Correction Coding Pins” on page 8–13. ■ Updated “Address and Control/Command Pins” on page 8–14.
October 2008	1.3	■ Updated “Introduction”, “DDR Input Registers” and “Conclusion” sections. ■ Updated chapter to new template.
May 2008	1.2	■ Added *(Note 4)* to Figure 8–3. ■ Updated Table 8–3 and Table 8-5. Added new Table 8–4. ■ Updated *(Note 1)* to Figure 8-4. Updated Figure 8–5 and 8–14.
July 2007	1.1	■ Updated “Data and Data Clock/Strobe Pins” section. ■ Updated Table 8–5. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

Section III. System Integration

This section includes the following chapters:

- Chapter 9, Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family
- Chapter 10, Hot-Socketing and Power-On Reset in the Cyclone III Device Family
- Chapter 11, SEU Mitigation in the Cyclone III Device Family
- Chapter 12, IEEE 1149.1 (JTAG) Boundary-Scan Testing for the Cyclone III Device Family

For information about the revision history for chapters in this section, refer to "Document Revision History" in each individual chapter.

9. Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family

CIII51016-2.2

This chapter describes the configuration, design security, and remote system upgrades in Cyclone® III devices. The Cyclone III device family (Cyclone III and Cyclone III LS devices) uses SRAM cells to store configuration data. Configuration data must be downloaded to Cyclone III device family each time the device powers up because SRAM memory is volatile.

The Cyclone III device family is configured using one of the following configuration schemes:

- Fast Active serial (AS)
- Active parallel (AP) for Cyclone III devices only
- Passive serial (PS)
- Fast passive parallel (FPP)
- Joint Test Action Group (JTAG)

All configuration schemes use an external configuration controller (for example, MAX® II devices or a microprocessor), a configuration device, or a download cable.

The Cyclone III device family offers the following configuration features:

- Configuration data decompression
- Design security (for Cyclone III LS devices only)
- Remote system upgrade

As Cyclone III LS devices play a role in larger and more critical designs in competitive commercial and military environments, it is increasingly important to protect your designs from copying, reverse engineering, and tampering. Cyclone III LS devices address these concerns with 256-bit advanced encryption standard (AES) programming file encryption and anti-tamper feature support to prevent tampering. For more information about the design security feature in Cyclone III LS devices, refer to "Design Security" on page 9–70.

System designers face difficult challenges such as shortened design cycles, evolving standards, and system deployments in remote locations. The Cyclone III device family helps 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, reduce time-to-market, and extend product life. Remote system upgrades can also be implemented with the advanced Cyclone III device family features such as real-time decompression of configuration data. For more information about the remote system upgrade feature in Cyclone III device family, refer to "Remote System Upgrade" on page 9–74.

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

ISO 9001:2008 Registered

Subscribe

This chapter describes the Cyclone III device family configuration features and describes how to configure Cyclone III device family using the supported configuration schemes. This chapter also includes configuration pin descriptions and the Cyclone III device family configuration file formats. In this chapter, the generic term "device" includes all Cyclone III device family.

This chapter contains the following sections:

- "Configuration Features" on page 9–2
- "Design Security" on page 9–70
- "Remote System Upgrade" on page 9–74

Configuration Features

Cyclone III device family offers configuration data decompression to reduce configuration file storage, provides design security feature to protect your configuration data (for Cyclone III LS devices only), and provides remote system upgrade to allow you to remotely update your Cyclone III device family designs.

Table 9–1 lists which configuration methods you can use in each configuration scheme.

Table 9–1. Cyclone III Device Family Configuration Features (Part 1 of 2)

Configuration Scheme	Configuration Method	Decompression	Remote System Upgrade (1)	Design Security (Cyclone III LS Devices Only)
Fast Active Serial Standard (AS Standard POR)	Serial Configuration Device	✓	✓	✓
Fast Active Serial Fast (AS Fast POR)	Serial Configuration Device	✓	✓	✓
Active Parallel ×16 Standard (AP Standard POR, for Cyclone III devices only)	Supported Flash Memory (2)	—	✓	—
Active Parallel ×16 Fast (AP Fast POR, for Cyclone III devices only)	Supported Flash Memory (2)	—	✓	—
Passive Serial Standard (PS Standard POR)	External Host with Flash Memory	✓	—	✓
	Download Cable	✓	—	✓ (3)
Passive Serial Fast (PS Fast POR)	External Host with Flash Memory	✓	—	✓
	Download Cable	✓	—	✓ (3)
Fast Passive Parallel Fast (FPP Fast POR)	External Host with Flash Memory	—	—	✓

Table 9–1. Cyclone III Device Family Configuration Features (Part 2 of 2)

Configuration Scheme	Configuration Method	Decompression	Remote System Upgrade (1)	Design Security (Cyclone III LS Devices Only)
JTAG based configuration	External Host with Flash Memory	—	—	—
	Download Cable	—	—	—

Notes to Table 9–1:

(1) Remote update mode is supported when using the remote system upgrade feature. You can enable or disable remote update mode with an option setting in the Quartus® II software. For more information about the remote system upgrade feature, refer to "Remote System Upgrade" on page 9–74.

(2) For more information about the supported families for the Micron commodity parallel flash, refer to Table 9–11 on page 9–24.

(3) The design security feature is not supported using a SRAM Object File (**.sof**).

The design security feature is for Cyclone III LS devices only and is available in all configuration schemes except the JTAG-based configuration. The decompression feature is not supported when you have enabled the design security feature.

When using a serial configuration scheme such as PS or fast AS, the configuration time is the same whether or not you have enabled the design security feature. A ×4 `DCLK` is required if you use the FPP scheme with the design security feature.

Cyclone III devices support remote system upgrade in AS and AP configuration schemes. Cyclone III LS devices only support remote system upgrade in the AS configuration scheme.

This section only describes the decompression feature. For more information about the design security and remote system upgrade, refer to "Design Security" on page 9–70 and "Remote System Upgrade" on page 9–74.

Configuration Data Decompression

Cyclone III device family supports 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 III device family. During configuration, Cyclone III device family decompress the bitstream in real time and program SRAM cells. The decompression feature is not supported when you have enabled the design security feature.

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

Cyclone III device family supports decompression in the AS and PS configuration schemes. Decompression is not supported in the AP, FPP, or JTAG-based configuration schemes. In PS mode, use the Cyclone III device family decompression feature to reduce configuration time.

Altera recommends using the Cyclone III device family decompression feature during AS configuration if you must save configuration memory space in the serial configuration device.

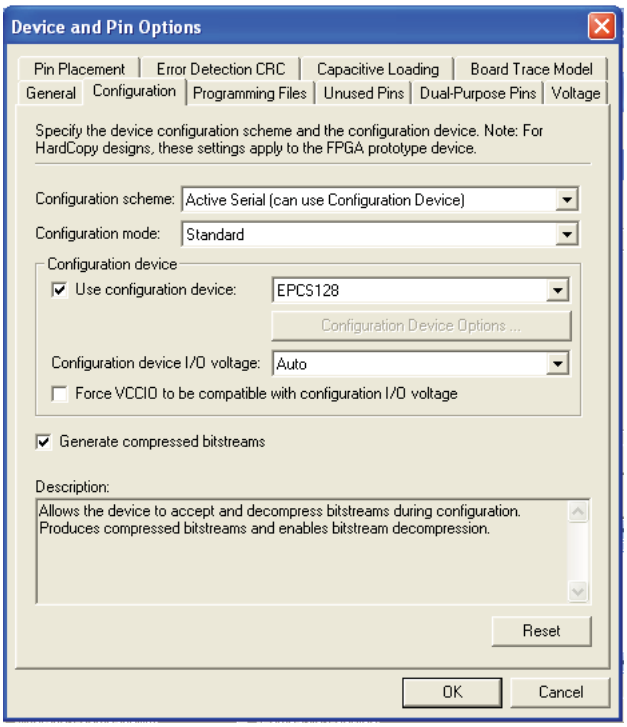
When you enable compression, the Quartus II software generates configuration files with compressed configuration data. This compressed file reduces the storage requirements in the configuration device or flash memory and decreases the time needed to send the bitstream to the Cyclone III device family.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

- Before design compilation (using the Compiler Settings menu).
- After design compilation (use 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** (Figure 9–1).
5. Click **OK**.
6. In the **Settings** dialog box, click **OK**.

Figure 9–1. Enabling Compression for Cyclone III Device Family Bitstreams in Compiler Settings



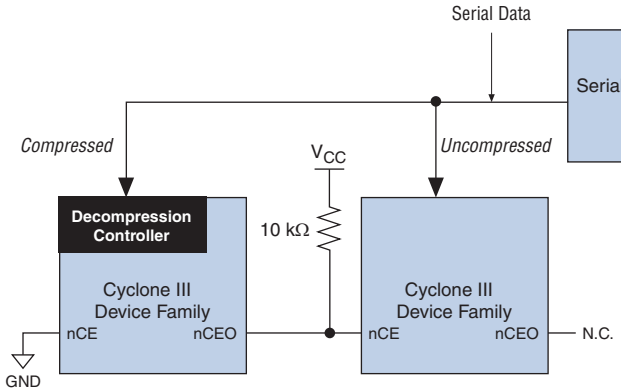
To enable compression when creating programming files from the **Convert Programming Files** window, follow these steps:

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

2. Under **Output programming file**, from the pull-down menu, select your desired file type.
3. If you select the Programmer Object File (**.pof**), you must specify a configuration device, directly under the file type.
4. In the **Input files to convert** box, select **SOF Data**.
5. Click **Add File** to browse to the Cyclone III device family **.sof**s.
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 devices in Cyclone III device family are cascaded, you can selectively enable the compression feature for each device in the chain. Figure 9–2 shows a chain of two devices in Cyclone III device family. 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 from the **Convert Programming Files** dialog box from the File menu in the Quartus II software.

Figure 9–2. Compressed and Uncompressed Configuration Data in the Same Configuration File



Configuration Requirement

The following section describes power-on-reset (POR) for Cyclone III device family.

POR Circuit

The POR circuit keeps the device in the reset state until the power supply voltage levels have stabilized after device power-up. After device power-up, the device does not release `nSTATUS` until the required voltages listed in table Table 9–4 on page 9–8 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 Cyclone III device family, you can select either a fast POR time or standard POR time depending on the MSEL pin settings. The fast POR time is 3 ms < TPOR < 9 ms for the fast configuration time. The standard POR time is 50 ms < TPOR < 200 ms, which has a lower power-ramp rate.

Table 9–2 lists the supported POR times for each configuration scheme.

Table 9–2. Cyclone III Device Family Supported POR Times Across Configuration Schemes (1)

Configuration Scheme	Fast POR Time (3 ms< TPOR < 9 ms)	Standard POR Time (50 ms< TPOR < 200 ms)	Configuration Voltage Standard (V) (2)
Fast Active Serial Standard (AS Standard POR)	—	✓	3.3
Fast Active Serial Standard (AS Standard POR)	—	✓	3.0/2.5
Fast Active Serial Fast (AS Fast POR)	✓	—	3.3
Fast Active Serial Fast (AS Fast POR)	✓	—	3.0/2.5
Active Parallel ×16 Standard (AP Standard POR, for Cyclone III devices only)	—	✓	3.3
Active Parallel ×16 Standard (AP Standard POR, for Cyclone III devices only)	—	✓	3.0/2.5
Active Parallel ×16 Standard (AP Standard POR, for Cyclone III devices only)	—	✓	1.8
Active Parallel ×16 Fast (AP Fast POR, for Cyclone III devices only)	✓	—	3.3
Active Parallel ×16 Fast (AP Fast POR, for Cyclone III devices only)	✓	—	1.8
Passive Serial Standard (PS Standard POR)	—	✓	3.3/3.0/2.5
Passive Serial Fast (PS Fast POR)	✓	—	3.3/3.0/2.5
Fast Passive Parallel Fast (FPP Fast POR)	✓	—	3.3/3.0/2.5
Fast Passive Parallel Fast (FPP Fast POR)	✓	—	1.8/1.5
JTAG-based configuration	(3)	(3)	—

Notes to Table 9–2:

(1) Altera recommends connecting the MSEL pins to V_{CCA} or GND depending on the MSEL pin settings.

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

(3) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored. However, the POR time is dependent on the MSEL pin settings.

In some applications, it is necessary for a device to wake up very quickly to begin operation. The Cyclone III device family offers 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 POR or standard POR options with the MSEL pin settings.

The automotive application is for Cyclone III devices only. The Cyclone III devices fast wake-up time meets the requirement of common bus standards in automotive applications, such as Media Orientated Systems Transport (MOST) and Controller Area Network (CAN).

For more information about wake-up time and the POR circuit, refer to the *Hot-Socketing and Power-On Reset in Cyclone III Devices* chapter.

Configuration File Size

Table 9–3 lists the uncompressed configuration file sizes for the Cyclone III device family. To calculate the amount of storage space required for multiple device configurations, add the file size of each device together.

Table 9–3. Cyclone III Device Family Uncompressed Raw Binary File (.rbf) Sizes

Device		Data Size (bits)
Cyclone III	EP3C5	3,000,000
	EP3C10	3,000,000
	EP3C16	4,100,000
	EP3C25	5,800,000
	EP3C40	9,600,000
	EP3C55	14,900,000
	EP3C80	20,000,000
	EP3C120	28,600,000
Cyclone III LS	EP3CLS70	26,766,760
	EP3CLS100	26,766,760
	EP3CLS150	50,610,728
	EP3CLS200	50,610,728

Use the data in Table 9–3 only 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 are using compression, the file size varies after each compilation because the compression ratio is design dependent.

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 III devices are manufactured using the TSMC 65-nm low-k dielectric process; Cyclone III LS devices are manufactured using the TSMC 60-nm low-k dielectric process. Although Cyclone III device family uses TSMC 2.5-V transistor technology in the I/O buffers, the devices are compatible and able to interface with 2.5-, 3.0-, 3.3-V configuration voltage standards. However, you must follow specific requirements when interfacing Cyclone III device family with 2.5-, 3.0-, 3.3-V configuration voltage standards.

All I/O inputs must maintain a maximum AC voltage of 4.1 V. When using a JTAG configuration scheme or a serial configuration device in an AS configuration scheme, you must connect a 25-Ω series resistor at the near end of the `TDO` and `TDI` pin or the serial configuration device for the `DATA[0]` pin. When cascading Cyclone III device family in a multi-device configuration, you must connect the repeater buffers between the master and slave devices for `DATA` and `DCLK`.

The output resistance of the repeater buffers must fit the maximum overshoot equation shown in Equation 9–1:

Equation 9–1. (1)

$$0.8Z_O \leq R_E \leq 1.8Z_O$$

Note to Equation 9–1:

(1) Z_0 is the transmission line impedance and R_E is the equivalent resistance of the output buffer.

Configuration Process

This section describes the configuration process.

For more information about the configuration cycle state machine of Altera® FPGAs, 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, the V_{CCIO} for all the I/O banks must be powered up to the appropriate level for the device to exit POR.

To begin configuration, the required voltages listed in Table 9–4 must be powered up to the appropriate voltage levels.

Table 9–4. Power-Up Voltage for Cyclone III Device Family Configuration

Device	Voltage that must be Powered-Up (1)
Cyclone III	V_{CCINT}, V_{CCA}, V_{CCIO} (2)
Cyclone III LS	V_{CCBAT}, V_{CCINT}, V_{CCA}, V_{CCIO} (2)

Notes to Table 9–4:

(1) Voltages must be powered up to the appropriate voltage levels to begin configuration.

(2) V_{CCIO} is for banks in which the configuration and JTAG pins reside.

Reset

When `nCONFIG` or `nSTATUS` is low, the device is in reset. After power-up, the Cyclone III device family goes through POR. POR delay depends on the MSEL pin settings, which correspond to your configuration scheme.

Depending on the configuration scheme, a fast or standard POR time is available. POR time for fast POR ranges between 3–9 ms. POR time for standard POR, which has a lower power-ramp rate, ranges between 50–200 ms.

During POR, the device resets, holds `nSTATUS` and `CONF_DONE` low, and tri-states all user I/O pins.

The configuration bus is not tri-stated in POR stage if the `MSEL` pins are set to AS or AP mode. To tri-state the configuration bus for AS and AP configuration schemes, you must tie `nCE` high and `nCONFIG` low. For more information about the hardware implementation, refer to "Configuring With Multiple Bus Masters" on page 9–30.

When the device exits POR, all user I/O pins continue to tri-state. The user I/O pins and dual-purpose I/O pins have weak pull-up resistors that are always enabled (after POR) before and during configuration. After POR, the Cyclone III device family releases `nSTATUS`, which is pulled high by an external 10-kΩ pull-up resistor and enters configuration mode.

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 begins.

Cyclone III LS devices are accessible by limited JTAG instructions after POR. For more information about enabling full JTAG instructions access, refer to "JTAG Instructions" on page 9–60.

For more information about the value of weak pull-up resistors on the I/O pins that are on before and during configuration, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters.

Configuration

Configuration data is latched into the Cyclone III device family 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 configuration is complete and initialization of the device can begin. The `CONF_DONE` pin must have an external 10-kΩ pull-up resistor for the device to initialize.

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 III device family is reset. The Cyclone III device family 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 III device family, reconfiguration begins.

Configuration Error

If an error occurs during configuration, the Cyclone III device family asserts 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 from the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Cyclone III device family 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 III device family, the clock source for initialization is either a 10-MHz (typical) internal oscillator (separate from the AS internal oscillator) or an 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 a proper initialization. When using the internal oscillator, you do not need 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 from 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, the Cyclone III device family requires a certain amount of clock cycles to initialize and to enter user mode.

Table 9–5 lists the required clock cycles for proper initialization in Cyclone III device family.

Table 9–5. Initialization Clock Cycles Required in Cyclone III Device Family

Device	Initialization Clock Cycles
Cyclone III	3,185
Cyclone III LS	3,192

Table 9–6 lists the maximum CLKUSR frequency (f_{MAX}) for Cyclone III device family.

Table 9–6. Maximum CLKUSR Frequency for Cyclone III Device Family

Device	f_{MAX} (MHz)
Cyclone III	133
Cyclone III LS	100

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 that 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 from 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 listed in Table 9–7.

The MSEL pins are powered by V_{CCINT}. The MSEL[3..0] pins have 9-kΩ internal pull-down resistors that are always active.

☞ Smaller Cyclone III devices or package options (E144, M164, Q240, F256, and U256 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 III devices without the MSEL[3] pin. To configure these devices with other supported configuration schemes, select the MSEL[2..0] pins according to the MSEL settings in Table 9–7.

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

☞ 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**.

Table 9–7. Cyclone III Device Family Configuration Schemes (1) (Part 1 of 2)

Configuration Scheme	MSEL				Configuration Voltage Standard (V) (2), (3)
	3	2	1	0	
Fast Active Serial Standard (AS Standard POR)	0	0	1	0	3.3
Fast Active Serial Standard (AS Standard POR)	0	0	1	1	3.0/2.5
Fast Active Serial Fast (AS Fast POR)	1	1	0	1	3.3
Fast Active Serial Fast (AS Fast POR)	0	1	0	0	3.0/2.5
Active Parallel ×16 Standard (AP Standard POR, for Cyclone III devices only)	0	1	1	1	3.3
Active Parallel ×16 Standard (AP Standard POR, for Cyclone III devices only)	1	0	1	1	3.0/2.5
Active Parallel ×16 Standard (AP Standard POR, for Cyclone III devices only)	1	0	0	0	1.8
Active Parallel ×16 Fast (AP Fast POR, for Cyclone III devices only)	0	1	0	1	3.3

Table 9–7. Cyclone III Device Family Configuration Schemes (1) (Part 2 of 2)

Configuration Scheme	MSEL				Configuration Voltage Standard (V) (2), (3)
	3	2	1	0	
Active Parallel ×16 Fast (AP Fast POR, for Cyclone III devices only)	0	1	1	0	1.8
Passive Serial Standard (PS Standard POR)	0	0	0	0	3.3/3.0/2.5
Passive Serial Fast (PS Fast POR)	1	1	0	0	3.3/3.0/2.5
Fast Passive Parallel Fast (FPP Fast POR) (4)	1	1	1	0	3.3/3.0/2.5
Fast Passive Parallel Fast (FPP Fast POR) (for Cyclone III devices only) (4)	1	1	1	1	1.8/1.5
Fast Passive Parallel Fast (FPP Fast POR) (for Cyclone III LS devices only)	0	0	0	1	1.8/1.5
Fast Passive Parallel Fast (FPP Fast POR) with Encryption (for Cyclone III LS devices only)	0	1	0	1	3.3/3.0/2.5
Fast Passive Parallel Fast (FPP Fast POR) with Encryption (for Cyclone III LS devices only)	0	1	1	0	1.8/1.5
JTAG-based configuration (5)	(6)	(6)	(6)	(6)	—

Notes to Table 9–7:

(1) Altera recommends connecting the MSEL pins to V_{CCA} or GND depending on the MSEL pin settings.

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

(3) You must follow specific requirements when interfacing Cyclone III device family with 2.5-, 3.0-, and 3.3-V configuration voltage standards. For more information about these requirements, refer to "Configuration and JTAG Pin I/O Requirements" on page 9–7.

(4) FPP configuration is not supported in the Cyclone III E144 device package of Cyclone III devices.

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

(6) 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 the JTAG configuration.

AS Configuration (Serial Configuration Devices)

In the AS configuration scheme, Cyclone III device family is configured using 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) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

In Cyclone III device family, the active master clock frequency runs at a maximum of 40 MHz, and typically at 30 MHz. Cyclone III device family only work with serial configuration devices that support up to 40 MHz.

Serial configuration devices provide a serial interface to access configuration data. During device configuration, Cyclone III device family reads configuration data using the serial interface, decompress 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.

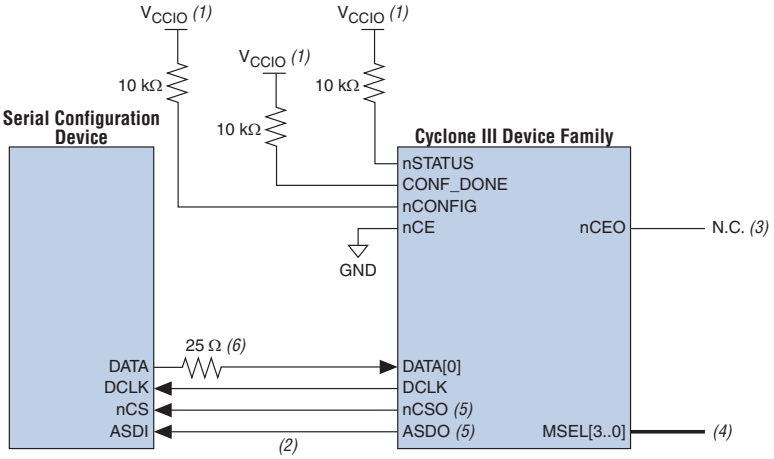
Single-Device AS Configuration

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

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

This four-pin interface connects to Cyclone III device family pins, as shown in Figure 9–3.

Figure 9–3. Single-Device AS Configuration



Notes to Figure 9–3:

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

(2) Cyclone III device family uses 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 MSEL[3..0], refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or GND.

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

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

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

When connecting a serial configuration device to a Cyclone III device family in the single-device AS configuration, you must connect a 25-Ω series resistor at the near end of the serial configuration device for DATA[0]. The 25-Ω resistor in the series works to minimize the driver impedance mismatch with the board trace and reduce overshoot seen at the Cyclone III device family DATA[0] input pin.

In a single-device AS configuration, the maximum board loading and board trace length between the supported serial configuration device and the Cyclone III device family must follow the recommendations in Table 9–9 on page 9–20.

The DCLK generated by the Cyclone III device family controls the entire configuration cycle and provides timing for the serial interface. Cyclone III device family uses a 40-MHz internal oscillator to generate DCLK. There are some variations in the internal oscillator frequency because of the process, voltage, and temperature conditions in Cyclone III device family. The internal oscillator is designed to ensure that its maximum frequency is guaranteed to meet the EPCS device specifications.

☞ EPCS1 does not support Cyclone III device family because of its insufficient memory capacity.

Table 9–8 lists the AS DCLK output frequency for Cyclone III device family.

Table 9–8. AS DCLK Output Frequency

Oscillator	Minimum	Typical	Maximum	Unit
40 MHz	20	30	40	MHz

In the AS configuration scheme, the serial configuration device latches input and control signals on the rising edge of DCLK and drives out configuration data on the falling edge. Cyclone III device family drives out control signals on the falling edge of DCLK and latch configuration data on the falling edge of DCLK.

In configuration mode, the Cyclone III device family enables the serial configuration device by driving the nCSO output pin low, which connects to the nCS pin of the configuration device. The Cyclone III device family 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 III device family.

After all the configuration bits are received by the Cyclone III device family, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ resistor. Initialization begins only after the CONF_DONE signal reaches a logic-high level. 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 weak internal pull-up resistors. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

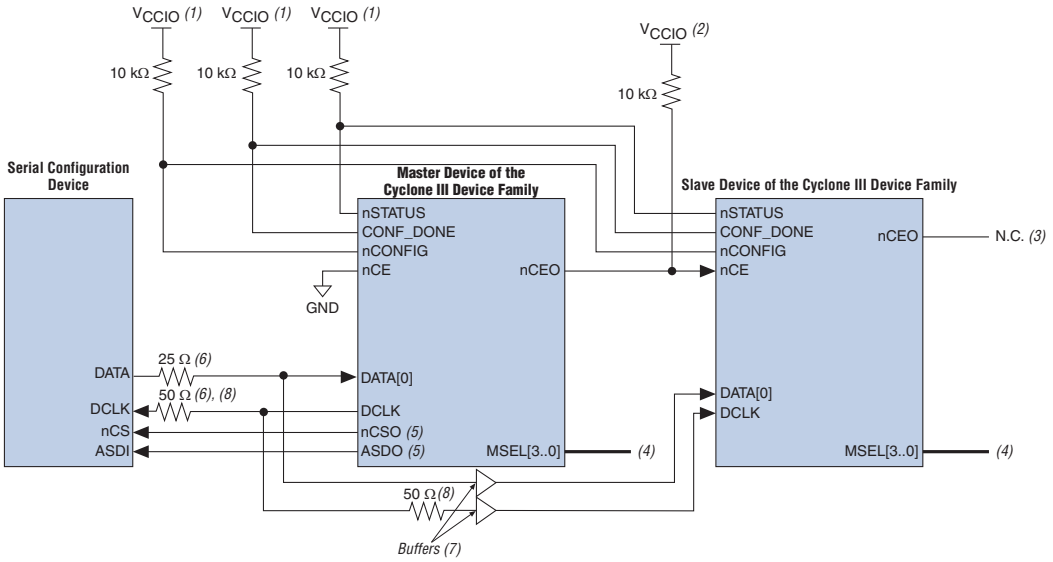
The timing parameters for AS mode are not listed here because the t_{CF2CD}, t_{CF2ST0}, t_{CFG}, t_{STATUS}, t_{CF2ST1}, and t_{CD2UM} timing parameters are identical to the timing parameters for PS mode listed in Table 9–13 on page 9–39.

Multi-Device AS Configuration

You can configure multiple Cyclone III device family using a single serial configuration device. You can cascade multiple Cyclone III device family 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 III device family. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA[0] pins of each device in the chain are connected (Figure 9–4).

Figure 9–4. Multi-device AS Configuration



Notes to Figure 9–4:

(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) 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 III device family in AS mode and the slave devices in PS mode. To connect MSEL[3..0] for the master device in AS mode and slave devices in PS mode, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or GND.

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

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

(7) Connect the repeater buffers between the master and slave devices of the Cyclone III device family 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 9–7.

(8) 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.

The first Cyclone III device family in the chain is the configuration master and controls the configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Cyclone III device family is configuration slaves and you must connect their MSEL pins to select the PS configuration scheme. Any other Altera device that supports PS configuration can also be part of the chain as a configuration slave.

When connecting a serial configuration device to the Cyclone III device family in the multi-device AS configuration, you must connect a 25-Ω series resistor at the near end of the serial configuration device for DATA[0].

In the multi-device AS configuration, the board trace length between the serial configuration device to the master device of the Cyclone III device family must follow the recommendations in Table 9–9 on page 9–20. You must also connect the repeater buffers between the master and slave devices of the Cyclone III device family 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 9–7.

As shown in Figure 9–4 on page 9–15, the nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors. 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.

Although you can cascade Cyclone III device family, 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 devices configuration bitstreams.

Configuring Multiple Cyclone III Device Family with the Same Design

Certain designs require you to configure multiple Cyclone III device family with the same design through a configuration bitstream or a **.sof**. You can do this using the following methods:

- Multiple SRAM Object Files
- Single SRAM Object File

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

Multiple SRAM Object Files

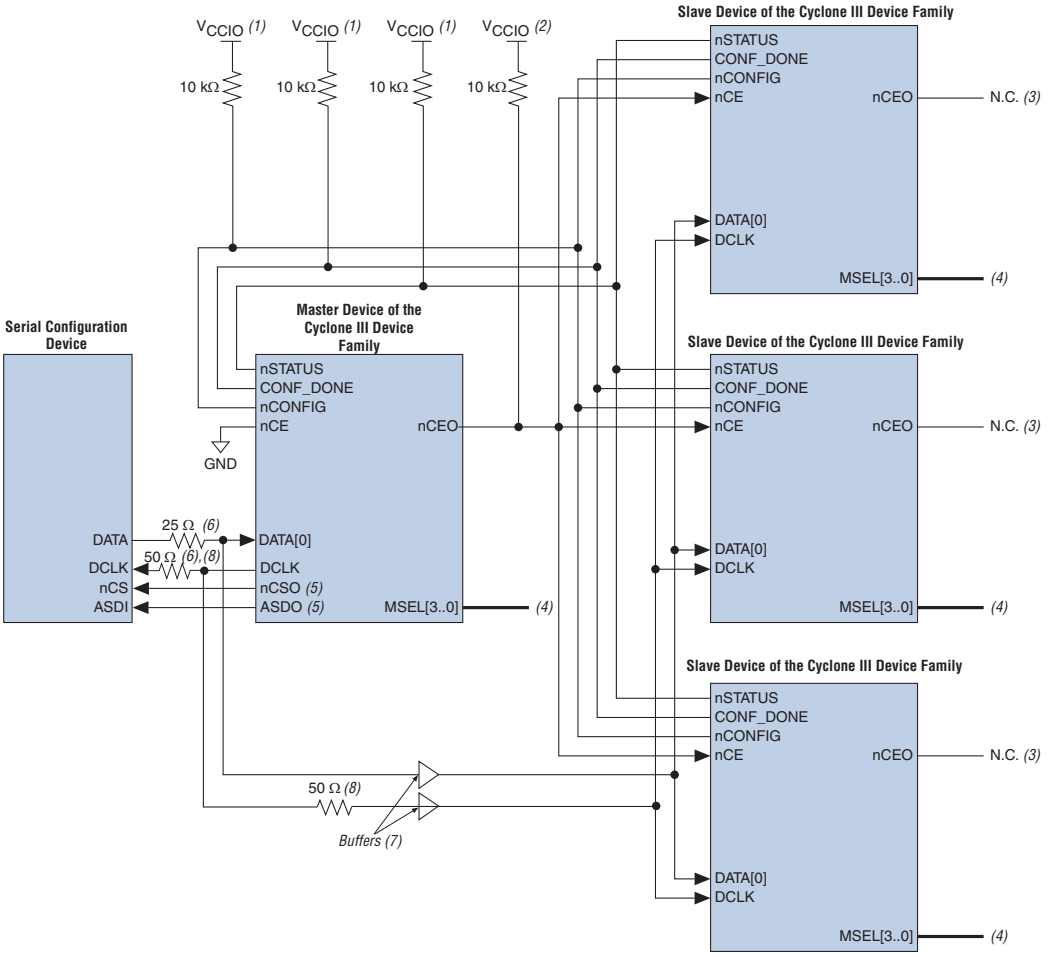
Two copies of the **.sof** are stored in the serial configuration device. Use the first copy to configure the master device of the Cyclone III device family and the second copy to configure all the remaining slave devices concurrently. All slave devices must be of the same density and package. The setup is similar to Figure 9–4 on page 9–15, in which the master device is set up in AS mode and the slave devices are set up in PS mode.

To configure four identical Cyclone III device family with the same **.sof**, you must set up the chain similar to Figure 9–5. The first device is the master device and its MSEL pins must be set to select the AS configuration. The other three slave devices are set up for concurrent configuration and their MSEL pins must be set to select the 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 using the setup in Figure 9–5 is that you can have a different **.sof** for the master device. However, all the slave devices must be configured with the same **.sof**. In this configuration method, you can either compress or uncompress the **.sof**s.

You can still use this method if the master and slave devices use the same **.sof**.

Figure 9–5. Multi-Device AS Configuration where the Devices Receive the Same Data with Multiple SRAM Object Files



Notes to Figure 9–5:

(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 `MSEL[3..0]` for the master device in AS mode and the slave devices in PS mode, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or GND.

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

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

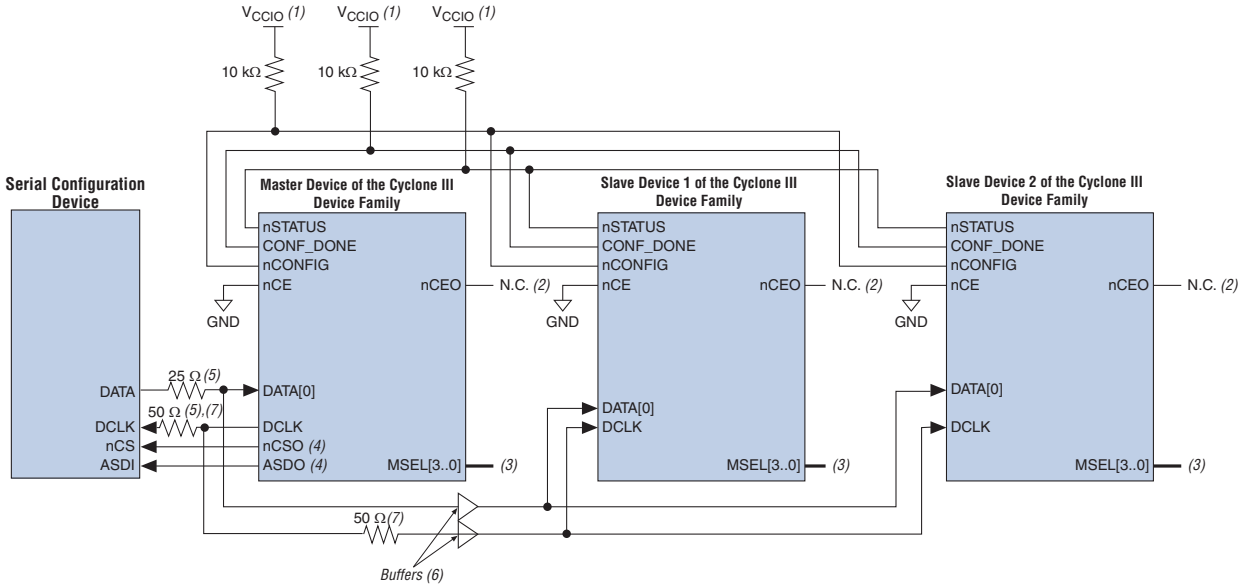
(7) Connect the repeater buffers between the master and slave devices for `DATA[0]` and `DCLK`. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

(8) 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.

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**. This setup is shown in Figure 9–6 where the master is set up in AS mode and the slave devices are set up in PS mode. You must set up one or more slave devices in the chain. All the slave devices must be set up as shown in Figure 9–6.

Figure 9–6. Multi-Device AS Configuration where the Devices Receive the Same Data with a Single .sof



Notes to Figure 9–6:

(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 III device family in AS mode and the slave devices in PS mode. To connect `MSEL[3..0]` for the master device in AS mode and slave devices in PS mode, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or GND.

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

(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 9–7.

(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.

In this setup, all the Cyclone III device family in the chain are connected for concurrent configuration. This can reduce the AS configuration time because all the Cyclone III device family is configured in one configuration cycle. Connect the `nCE` input pins of all the Cyclone III device family to ground. You can either leave the `nCEO` output pins on all the Cyclone III device family 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 III device family.

Altera recommends putting a buffer before the DATA and DCLK output from the master device to avoid signal strength and 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**s. 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 Serial Configuration Device to Cyclone III Device Family on AS Interface

For single- and multi-device AS configurations, the board trace length and loading between the supported serial configuration device and Cyclone III device family must follow the recommendations listed in Table 9–9.

Table 9–9. Maximum Trace Length and Loading for the AS Configuration

Cyclone III Device Family AS Pins	Maximum Board Trace Length from the Cyclone III Device Family to the Serial Configuration Device (Inches)	Maximum Board Load (pF)
DCLK	10	15
DATA[0]	10	30
nCSO	10	30
ASDO	10	30

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 III device family. This serial interface is clocked by the Cyclone III device family DCLK output (generated from an internal oscillator). Equation 9–2 and Equation 9–3 show the configuration time estimation for the Cyclone III device family.

Equation 9–2.

$$\text{Size} \times \left(\frac{\text{maximum DCLK period}}{1\text{ bit}}\right) = \text{estimated maximum configuration ti}$$

Equation 9–3.

$$3{,}500{,}000\text{ bits} \times \left(\frac{50\text{ ns}}{1\text{ bit}}\right) = 175\text{ ms}$$

To estimate the typical configuration time, use the typical DCLK period shown in Figure 9–7 on page 9–22. With a typical DCLK period of 33.33 ns, the typical configuration time is 116.7 ms. Enabling compression reduces the amount of configuration data that is sent to the Cyclone III device family, 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 using the USB-Blaster™ or ByteBlaster™ II download cable. Alternatively, you can program them using 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 using 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 III device family is 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 GND and V_{CC}, respectively.

To perform in-system programming of a serial configuration device using the AS programming interface, the diodes and capacitors must be placed as close as possible to the Cyclone III device family. Ensure that the diodes and capacitors maintain a maximum AC voltage of 4.1 V (Figure 9–7).

If you wish to use the same setup shown in Figure 9–7 to perform in-system programming of a serial configuration device and single- or multi-device AS configuration, you do not need a series resistor on the `DATA` line at the near end of the serial configuration device. The existing diodes and capacitors are sufficient.

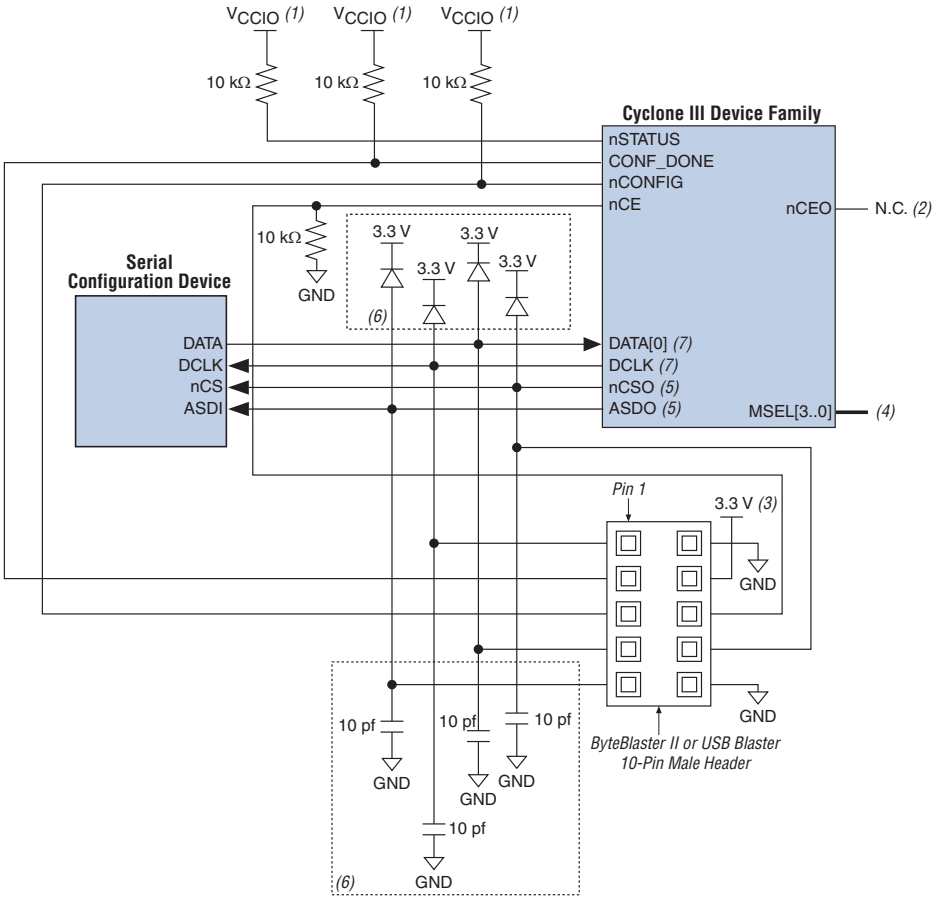
Altera has developed the Serial FlashLoader (SFL), a JTAG-based in-system programming solution for Altera serial configuration devices. The SFL is a bridge design for the Cyclone III device family that uses its JTAG interface to access the EPCS JIC (JTAG Indirect Configuration Device Programming) file 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 III device family, 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 9–7 shows the download cable connections to the serial configuration device.

Figure 9–7. In-System Programming of Serial Configuration Devices



Notes to Figure 9–7:

(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 `MSEL[3..0]`, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or ground.

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

(6) The diodes and capacitors must be placed as close as possible to the Cyclone III device family. 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 III device family AS configuration input pins due to possible overshoot when programming the serial configuration device using a download cable. For effective voltage clamping, Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes. For more information about the interface guidelines using Schottky diodes, refer to *AN 523: Cyclone III Configuration Interface Guidelines with EPCS Devices.*

(7) When cascading Cyclone III device family 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 9–7.

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 using 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 the 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 (www.altera.com).

AP Configuration (Supported Flash Memories)

The AP configuration scheme is for Cyclone III devices only. In the AP configuration scheme, Cyclone III 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 the 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 III devices or package options do not support the AP configuration scheme and do not have the `MSEL[3]` pin. Table 9–10 lists the supported AP configuration scheme for each Cyclone III device.

Table 9–10. Supported AP Configuration Scheme for Cyclone III Devices

Device	Package Options								
	E144	M164	Q240	F256	F324	F484	F780	U256	U484
EP3C5	—	—	—	—	—	—	—	—	—
EP3C10	—	—	—	—	—	—	—	—	—
EP3C16	—	—	—	—	—	✓	—	—	✓
EP3C25	—	—	—	—	✓	—	—	—	—
EP3C40	—	—	—	—	✓	✓	✓	—	✓
EP3C55	—	—	—	—	—	✓	✓	—	✓
EP3C80	—	—	—	—	—	✓	✓	—	✓
EP3C120	—	—	—	—	—	✓	✓	—	—

During device configuration, Cyclone III 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 Memory

The AP configuration controller in Cyclone III devices is designed to interface with the Micron P30 Parallel NOR flash family and the Micron P33 Parallel NOR flash family, which are two industry standard flash families. Unlike serial configuration devices, both of the flash families supported in the 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 and P33 flash families 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 III devices use a 40-MHz oscillator for the AP configuration scheme.

Table 9–11 lists the supported families of the commodity parallel flash for the AP configuration scheme.

Table 9–11. Supported Commodity Flash for the AP Configuration Scheme for Cyclone III Devices *(1)*

Flash Memory Density	Micron P30 Flash Family *(2)*	Micron P33 Flash Family (3)
64 Mbit	✓	✓
128 Mbit	✓	✓
256 Mbit	✓	✓

Notes to Table 9–11:

(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.

The AP configuration scheme of Cyclone III devices supports the Micron P30 and P33 family 64-, 128-, and 256-Mbit flash memories. Configuring Cyclone III 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.

☞ You must refer to the respective flash data sheets to check for supported speed grades and package options.

The AP configuration scheme in Cyclone III devices supports flash speed grades of 40 MHz and above. However, the 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 III device accesses flash memory in user mode.

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

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

Following are the control signals 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#`)

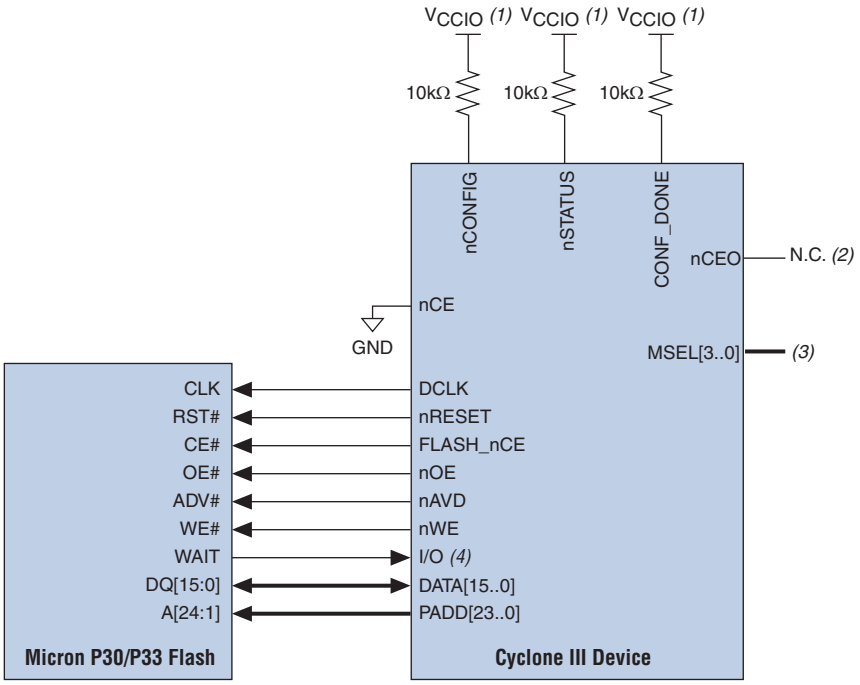
The supported parallel flash memories output a control signal (`WAIT`) to Cyclone III devices to indicate when synchronous data is ready on the data bus. Cyclone III devices have a 24-bit address bus connecting to the address bus (`A[24:1]`) of the flash memory. A 16-bit bidirectional data bus (`DATA[15..0]`)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

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

- `DCLK`
- `nRESET`
- `FLASH_nCE`
- `nOE`
- `nAVD`
- `nWE`

The interface for the Micron P30 flash memory and P33 flash memory connects to Cyclone III device pins, as shown in Figure 9–8.

Figure 9–8. Single-Device AP Configuration Using Micron P30 and P33 Flash Memory



Notes to Figure 9–8:

(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 9–7 on page 9–11. 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 a 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 the supported parallel flash and Cyclone III devices must follow the recommendations listed in Table 9–12 on page 9–30.

If you use the AP configuration scheme for Cyclone III 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/P33 flash and the Cyclone III device in the AP configuration scheme.

There are no series resistors required in the AP configuration mode for Cyclone III 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 III devices, all shared pins are still subject to the 4.1-V limit and may require series resistors.

The 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`.

- `nRESET` is an active-low hard reset
- `FLASH_nCE` is an active-low chip enable
- `nOE` is an active-low output enable for the `DATA[15..0]` bus and `WAIT` pin
- `nAVD` is an active-low address valid signal and is used to write addresses into the flash
- `nWE` is an active-low write enable and is used to write data into the flash
- `PADD[23..0]` bus is the address bus supplied to the flash
- `DATA[15..0]` bus is a bidirectional bus used to supply and read data to and from the flash, with the flash output controlled by `nOE`

The serial clock (`DCLK`) generated by Cyclone III devices controls the entire configuration cycle and provides timing for the parallel interface. Cyclone III devices use a 40-Mhz internal oscillator to generate `DCLK`. The oscillator is the same oscillator used in the AS configuration scheme. The active `DCLK` output frequency is listed in Table 9–8 on page 9–14.

Multi-Device AP Configuration

You can cascade multiple Cyclone III 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. 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 III device. The `nCONFIG`, `nSTATUS`, `CONF_DONE`, `DCLK`, `DATA[15..8]`, and `DATA[7..0]` pins of each device in the chain are connected (Figure 9–9 on page 9–28 and Figure 9–10 on page 9–29).

The first Cyclone III device in the chain, as shown in Figure 9–9 on page 9–28 and Figure 9–10 on page 9–29, is the configuration master device and controls the configuration of the entire chain. Connect its MSEL pins to select the AP configuration scheme. The remaining Cyclone III devices are used as configuration slaves. 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.

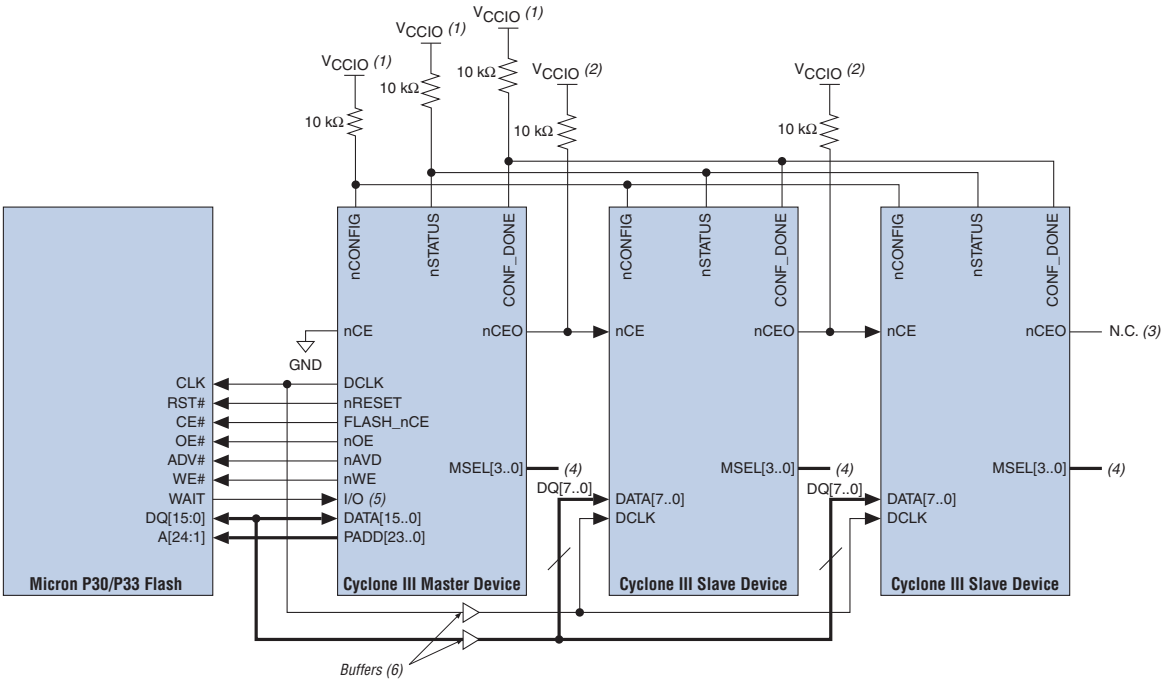
The following are the configurations for the DATA[15..0] bus in a multi-device AP configuration:

- 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 9–9.

Figure 9–9. Byte-Wide Multi-Device AP Configuration



Notes to Figure 9–9:

(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 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or GND.

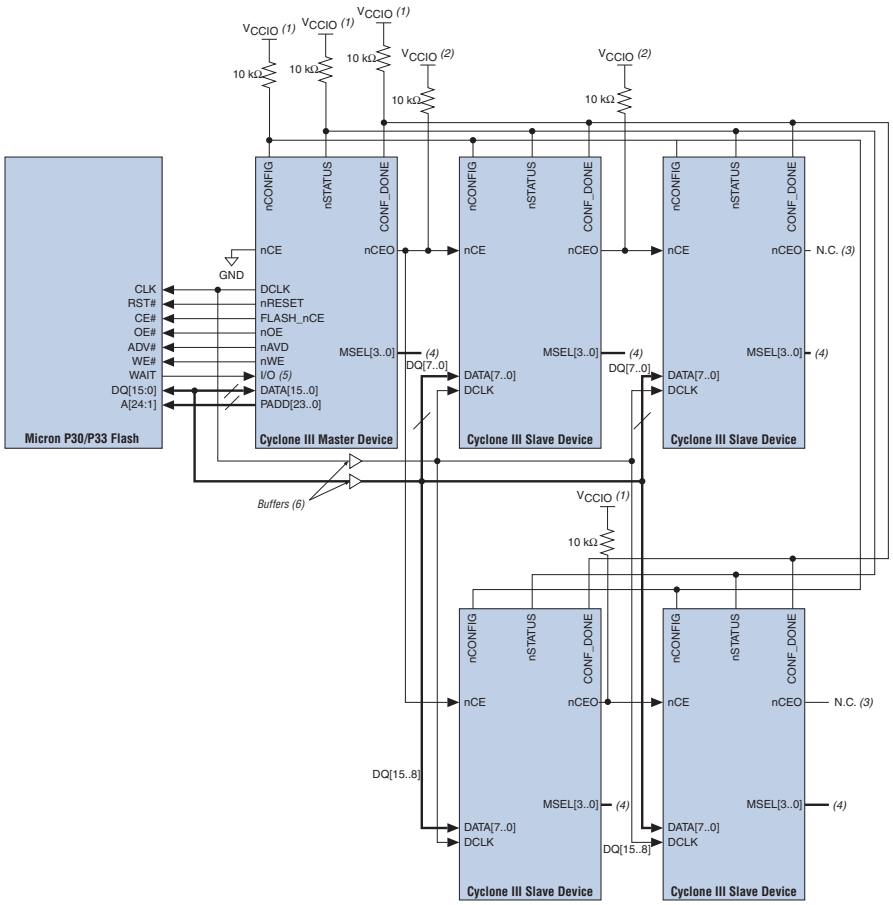
(5) The AP configuration ignores the WAIT signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

(6) Connect the repeater buffers between the 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 9–7.

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 `DATA[7..0]` and the remaining slave devices are connected to the MSB of `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 9–10.

Figure 9–10. Word-Wide Multi-Device AP Configuration



Notes to Figure 9–10:

(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 9–7 on page 9–11. 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 III 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 9–7.

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 9–12.

As shown in Figure 9–9 and Figure 9–10, the nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors. 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 III Devices for the AP Interface

For the single- and multi-device AP configuration, the board trace length and loading between the supported parallel flash and Cyclone III devices must follow the recommendations listed in Table 9–12. These recommendations also apply to an AP configuration with multiple bus masters.

Table 9–12. Maximum Trace Length and Loading for the AP Configuration

Cyclone III AP Pins	Maximum Board Trace Length from the Cyclone III Device to the 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 9–12:

(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.

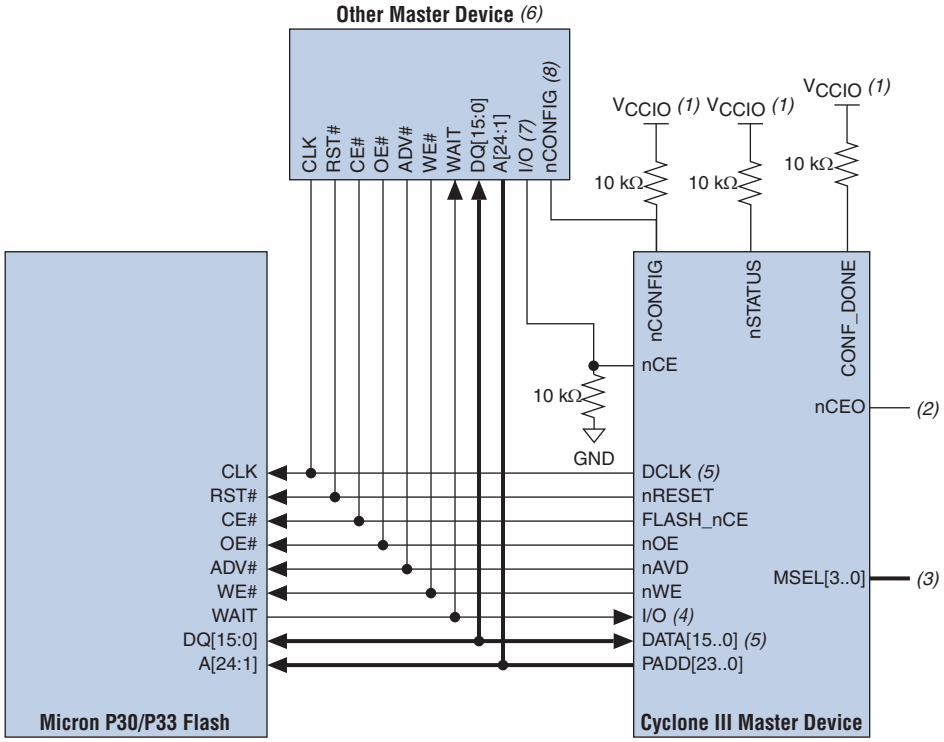
Configuring With Multiple Bus Masters

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 III device and override the weak 10 kΩ pull-down resistor on the nCE pin. This resets the master Cyclone III device and causes it to tri-state its AP configuration bus. The other master then takes control of the AP configuration bus. After the other master 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 9–11 shows the AP configuration with multiple bus masters.

Figure 9–11. AP Configuration with Multiple Bus Masters



Notes to Figure 9–11:

(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 9–7 on page 9–11. 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 III 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 9–7.

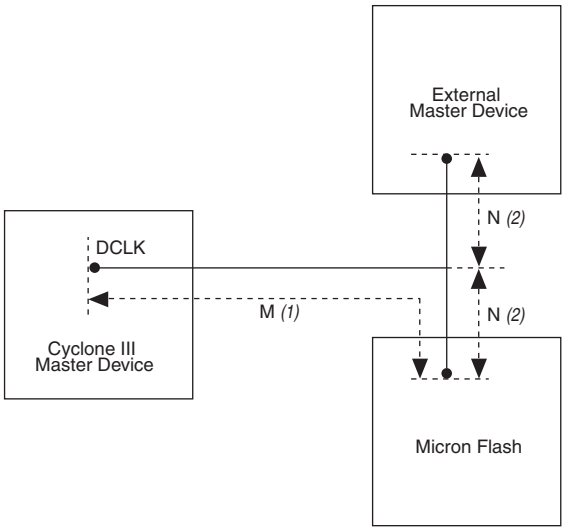
(6) The other master device must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

(7) The other master device can control the AP configuration bus by driving the nCE pin 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 rather than tied to V_{CCIO}.

Figure 9–12 shows the recommended balanced star routing for multiple bus master interfaces to minimize signal integrity issue.

Figure 9–12. Balanced Star Routing



Notes to Figure 9–12:

(1) Altera does not recommend *M* to exceed six inches as listed in Table 9–12 on page 9–30.

(2) Altera recommends using a balanced star routing. Try to 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 the AP Configuration Time

AP configuration time is dominated by the time it takes to transfer data from the parallel flash to the Cyclone III devices. This parallel interface is clocked by the Cyclone III `DCLK` output (generated from an internal oscillator). As listed in Table 9–8 on page 9–14, 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. Therefore, the maximum configuration time estimation for an EP3C40 device (9,600,000 bits of uncompressed data) is defined in Equation 9–4 and Equation 9–5.

Equation 9–4.

$$\text{Size} \times \left(\frac{\text{maximum DCLK period}}{\text{16 bits per DCLK cycle}}\right) = \text{estimated maximum configuration ti}$$

Equation 9–5.

$$9{,}600{,}000 \text{ bits} \times \left(\frac{50 \text{ ns}}{16 \text{ bits}}\right) = 30 \text{ ms}$$

To estimate a typical configuration time, use the typical `DCLK` period listed in Table 9–8 on page 9–14. With a typical `DCLK` period of 33.33 ns, the typical configuration time is 20 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 9–11 on page 9–24.

Cyclone III devices in a single- or multiple-device chains support in-system parallel flash programming with the JTAG interface using the flash loader megafunction. For Cyclone III devices, the board-intelligent host or download cable uses four JTAG pins to program the parallel flash in system, even if the host or download cable cannot access the configuration pins of the parallel flash.

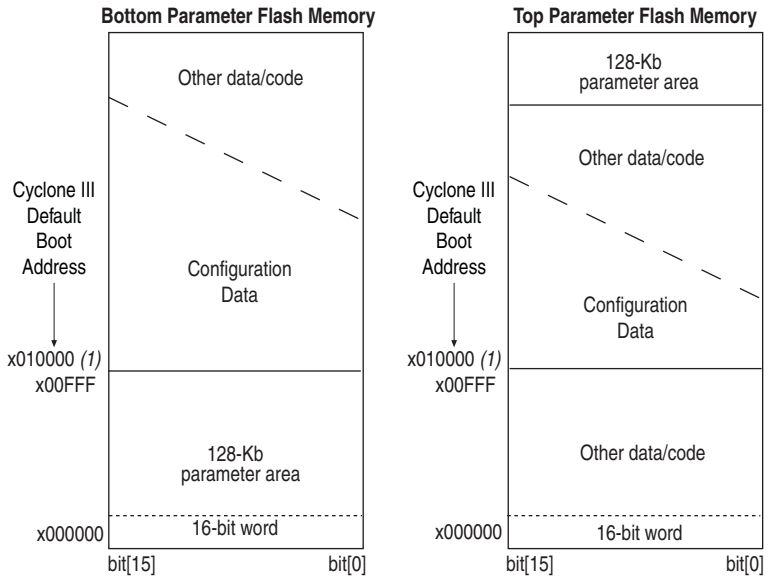
For more information about using the JTAG pins on Cyclone III devices to program the parallel flash in-system, refer to *AN 478: Using FPGA-Based Parallel Flash Loader (PFL) with the Quartus II Software*.

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 9–13). In the Quartus II software, the default configuration boot address is 0x020000 because it is represented in 8-bit byte addressing. Cyclone III devices configure from word address 0x010000, which is equivalent to byte address 0x020000.

The Quartus II software uses byte addressing for the default configuration boot address. You must set the start address field to **0x020000**.

The default configuration boot addressing 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. The configuration boot address in the AP configuration scheme is shown in Figure 9–13. You can change the default configuration default boot address **0x010000** 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 9–60.

Figure 9–13. Configuration Boot Address in AP Flash Memory Map



Note to Figure 9–13:

(1) The default configuration boot address is x010000 when represented in 16-bit word addressing.

PS Configuration

You can perform PS configuration on Cyclone III device family 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 III device family using the `DATA[0]` pin at each rising edge of `DCLK`.

If your system already contains a common flash interface (CFI) flash memory, you can use it for the Cyclone III device family configuration storage as well. The MAX II PFL feature provides an efficient method to program CFI flash memory devices through the JTAG interface and provides the logic to control the configuration from the flash memory device to the Cyclone III device family. Both PS and FPP configuration schemes are supported using the PFL feature.

For more information about the PFL, refer to *Parallel Flash Loader Megafunction User Guide*.

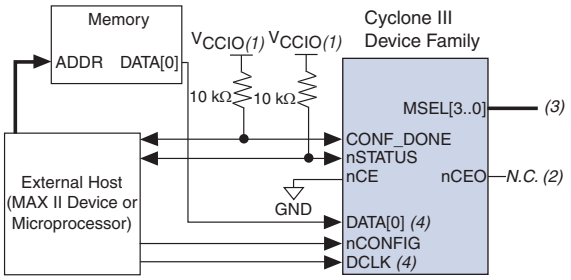
Cyclone III device family does not support enhanced configuration devices for PS or FPP configurations.

PS Configuration Using an External Host

In the PS configuration scheme, you can use an intelligent host such as MAX II or microprocessor that controls the transfer of configuration data from a storage device, such as flash memory, to the target Cyclone III device family. You can store the configuration data in **.rbf**, **.hex**, or **.ttf** format.

Figure 9–14 shows the configuration interface connections between a Cyclone III device family and an external host device for a single-device configuration.

Figure 9–14. Single-Device PS Configuration Using an External Host



Notes to Figure 9–14:

(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 `MSEL[3..0]`, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or ground.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. `DATA[0]` and `DCLK` must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

To begin 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 the `DATA[0]` pin. If you are using configuration data in a **.rbf**, **.ttf**, or **.hex** file, 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 III device family receives configuration data on the `DATA[0]` pin 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 and the device enters the initialization state.

Two `DCLK` falling edges are required after `CONF_DONE` goes high to begin device initialization.

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 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.

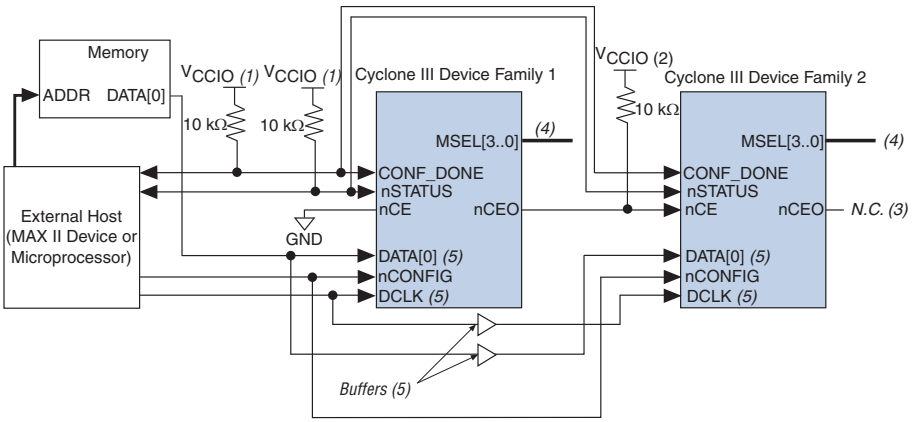
The configuration clock (DCLK) speed must be below the specified system frequency to ensure correct configuration (Figure 9–19 on page 9–42). No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If a configuration error occurs during configuration and the **Auto-restart configuration after error** option is turned on, the Cyclone III device family releases nSTATUS after a reset time-out period (a maximum of 230 µs). After nSTATUS is released and pulled high by a pull-up resistor, the external host device tries to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the external host device must generate a low-to-high transition (with a low pulse of at least 500 ns) on nCONFIG to restart the configuration process.

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 the 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 9–15 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 the Cyclone III device family is cascaded for multi-device configuration.

Figure 9–15. Multi-Device PS Configuration Using an External Host



Notes to Figure 9–15:

(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 MSEL[3..0], refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or ground.

(5) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

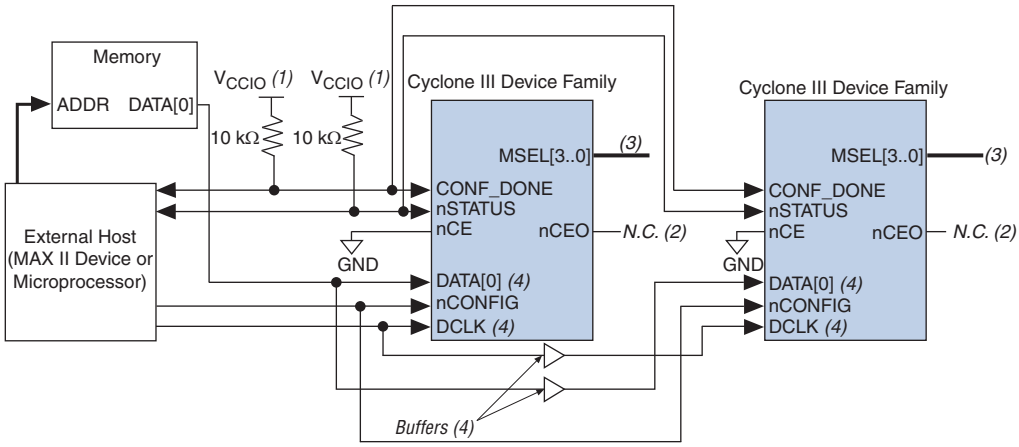
In a multi-device PS configuration, the nCE pin of the first device is connected to GND while its nCEO pin is connected to the nCE pin of the next device in the chain. The nCE input of the last device comes from the previous device, while its nCEO pin is left floating. After the first device completes 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. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

If any device detects an error, configuration stops for the entire chain and the entire chain must be reconfigured 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. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE) are connected to every device in the chain. Configuration signals can 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 9–16 shows a multi-device PS configuration when both Cyclone III device family is receiving the same configuration data.

Figure 9–16. Multi-Device PS Configuration When Both Devices Receive the Same Data



Notes to Figure 9–16:

(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) 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 `MSEL[3..0]`, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or ground.

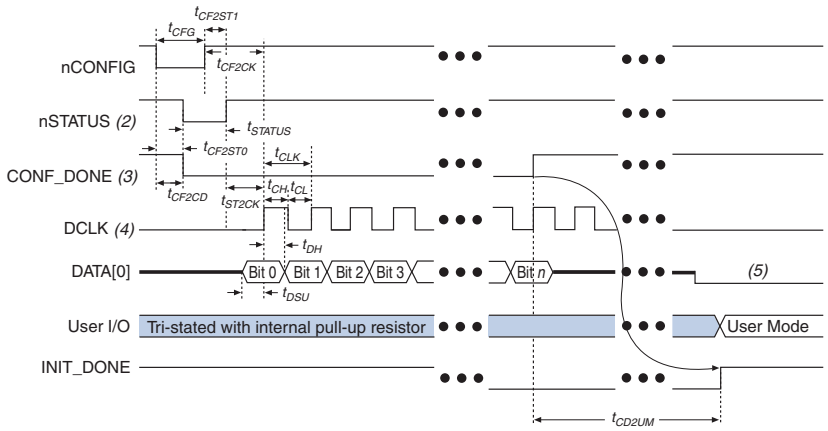
(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. `DATA[0]` and `DCLK` must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

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 9–17 shows the timing waveform for a PS configuration when using an external host device as an external host.

Figure 9–17. PS Configuration Timing Waveform (1)



Notes to Figure 9–17:

(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 III device family 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 III device family output pin and must not be driven externally.

(5) Do not leave the DATA[0] pin floating after configuration. Drive it high or low, whichever is more convenient.

Table 9–13 lists the PS configuration timing parameters for Cyclone III device family.

Table 9–13. PS Configuration Timing Parameters for Cyclone III Device Family (Part 1 of 2)

Symbol	Parameter	Minimum	Maximum	Unit
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	800 (1)	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	800 (2)	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	800 (1)	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	5	—	ns
t_{DH}	Data hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	3.2	—	ns
t_{CL}	DCLK low time	3.2	—	ns
t_{CLK}	DCLK period	7.5	—	ns
f_{MAX}	DCLK frequency	—	100 (4)	MHz
t_{CD2UM}	CONF_DONE high to user mode (3)	300	650	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—

Table 9–13. PS Configuration Timing Parameters for Cyclone III Device Family (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Unit
t_{CD2UMC}	CONF_DONE high to user mode with **CLKUSR** option on	t_{CD2CU} + (initialization clock cycles × CLKUSR period) (5)	—	—

Notes to Table 9–13:

(1) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(2) This value is applicable if you do not delay configuration by externally holding nSTATUS low.

(3) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting the device.

(4) Cyclone III devices can support a DCLK f_{MAX} of 133 MHz. Cyclone III LS devices can support a DCLK f_{MAX} of 100 MHz.

(5) For more information about the initialization clock cycles required in Cyclone III device family, refer to Table 9–5 on page 9–10.

PS Configuration Using a Download Cable

In this section, the generic term "download cable" includes the Altera USB-Blaster universal serial bus (USB) port download cable, MasterBlaster™ serial/USB communications cable, ByteBlaster II parallel port download cable, the ByteBlasterMV™ parallel port download cable, and the Ethernet-Blaster communications cable.

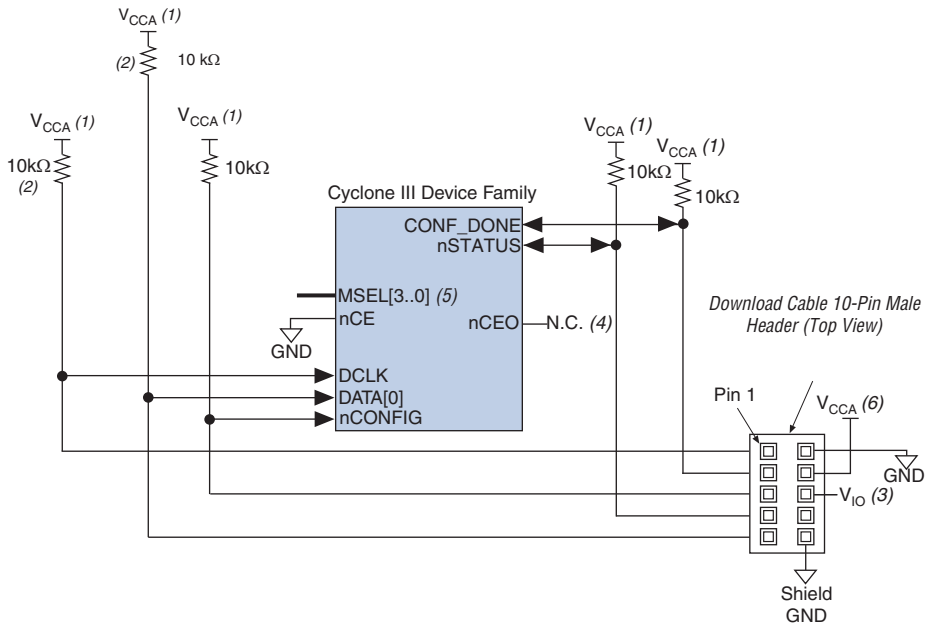
In the PS configuration with a download cable, an intelligent host (such as a PC) transfers data from a storage device to the device using the download cable.

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 when an error occurs. Additionally, the **Enable user-supplied start-up clock (CLKUSR)** option has no effect on the device initialization because this option is disabled in the **.sof** when programming the device using the Quartus II programmer and download cable. Therefore, if you turn on the **CLKUSR** option, you do not need to provide a clock on CLKUSR when you are configuring the device with the Quartus II programmer and a download cable.

Figure 9–18 shows PS configuration for Cyclone III device family using a download cable.

Figure 9–18. PS Configuration Using a USB-Blaster, MasterBlaster, ByteBlaster II, ByteBlasterMV, or Ethernet Blaster Cable



Notes to Figure 9–18:

(1) The pull-up resistor must be connected to the same supply voltage as the V_{CCA} supply.

(2) You only need the pull-up resistors on `DATA[0]` and `DCLK` 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 are also using a configuration device, you do not need the pull-up resistors on `DATA[0]` and `DCLK`.

(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*. For the USB Blaster, ByteBlaster II, ByteBlaster MV, and Ethernet Blaster, 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 `MSEL[3..0]`, refer to Table 9–7 on page 9–11 for PS configuration schemes. Connect the MSEL pins directly to V_{CCA} or GND.

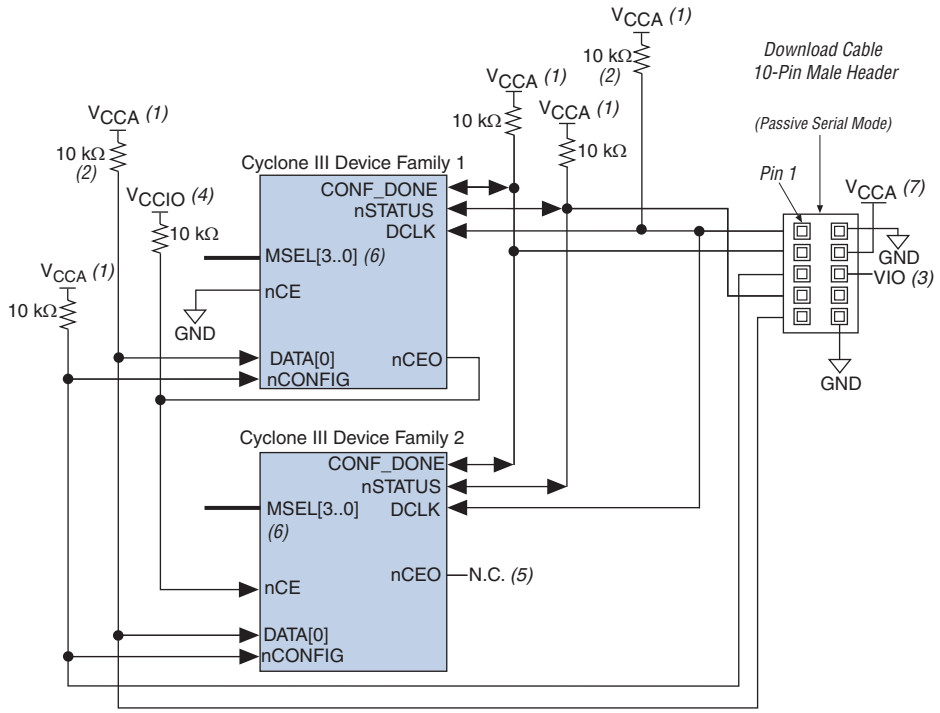
(6) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5- V supply from V_{CCA}. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a V_{CC} power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V 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 III device family by connecting the `nCEO` pin of each device to the `nCE` pin of the subsequent device. The `nCE` pin of the first device is connected to GND while its `nCEO` pin is connected to the `nCE` pin of the next device in the chain. The `nCE` input of the last device comes from the previous device while its `nCEO` pin is left floating. All other configuration pins, `nCONFIG`, `nSTATUS`, `DCLK`, `DATA[0]`, and `CONF_DONE` are connected to every device in the chain. Because all `CONF_DONE` pins are tied together, all devices in the chain initialize and enter user mode at the same time.

In addition, the entire chain halts configuration if any device detects an error because the `nSTATUS` pins are tied together. The **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs.

Figure 9–19 shows PS configuration for multi Cyclone III device family using a MasterBlaster, USB-Blaster, ByteBlaster II, or ByteBlasterMV cable.

Figure 9–19. Multi-Device PS Configuration Using a USB-Blaster, MasterBlaster, ByteBlaster II, ByteBlasterMV, or Ethernet Blaster Cable



Notes to Figure 9–19:

(1) The pull-up resistor must be connected to the same supply voltage as the V_{CCA} supply.

(2) You only need the pull-up resistors on `DATA[0]` and `DCLK` 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 are also using a configuration device, you do not need the pull-up resistors on `DATA[0]` and `DCLK`.

(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 ByteBlasterMV, this pin is a no connect. In USB-Blaster, ByteBlaster II, and Ethernet Blaster, 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[3..0]` for PS configuration schemes, refer to Table 9–7 on page 9–11. 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 III device family is designed to meet the increasing demand for faster configuration time. Cyclone III device family is designed with the capability of receiving byte-wide configuration data per clock cycle.

You can perform the FPP configuration of Cyclone III device family 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 III device family 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 III device family. Both PS and FPP configuration schemes are supported using this PFL feature.

For more information about the PFL, refer to *Parallel Flash Loader Megafunction User Guide*.

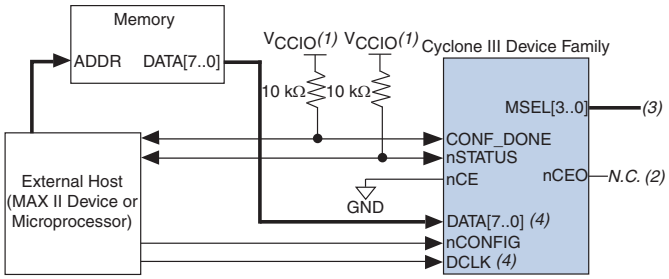
Cyclone III device family does not support enhanced configuration devices for PS or FPP configurations.

FPP configuration is not supported in the E144 package of Cyclone III devices.

FPP Configuration Using an External Host

The FPP configuration using an external host provides a fast method to configure Cyclone III device family. 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 III device family. You can store configuration data in either 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 9–20 shows the configuration interface connections between the Cyclone III device family and an external device for single-device configuration.

Figure 9–20. Single-Device FPP Configuration Using an External Host



Notes to Figure 9–20:

(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 `MSEL[3..0]`, refer to Table 9–7 on page 9–11. 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 equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

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 III device family receives 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 the 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 III device family requires certain amount of clock cycles to initialize properly and enter user mode. For more information about the initialization clock cycles required in the Cyclone III device family, refer to Table 9–5 on page 9–10. For more information about the supported CLKUSR f_{MAX} value for Cyclone III device family, refer to Table 9–14 on page 9–47.

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.

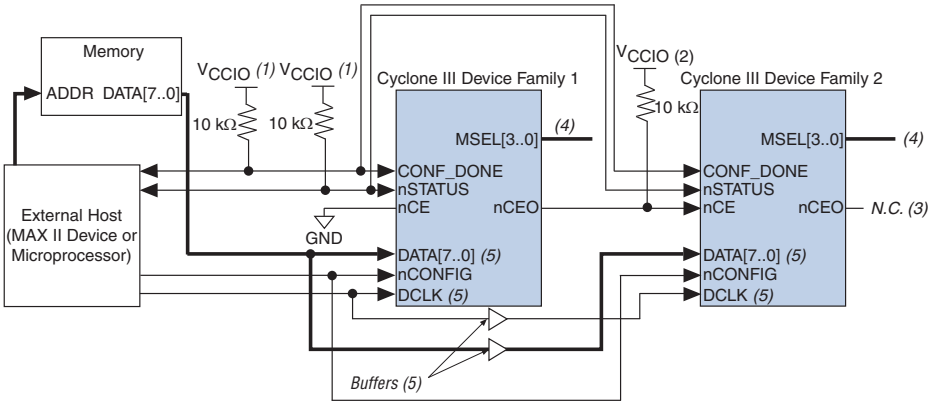
The DCLK speed must be below the specified system frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If a configuration error occurs during configuration and the **Auto-restart configuration after error** option is turned on, the Cyclone III device family releases nSTATUS after a reset time-out period (a maximum of 230 µs). After nSTATUS is released and pulled high by a pull-up resistor, the external host device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the external host device must generate a low-to-high transition (with a low pulse of at least 500 ns) on nCONFIG to restart the configuration process.

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 9–21 shows how to configure multiple devices using a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the Cyclone III device family is cascaded for a multi-device configuration.

Figure 9–21. Multi-Device FPP Configuration Using an External Host



Notes to Figure 9–21:

(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 `MSEL[3..0]`, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or ground.

(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 equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

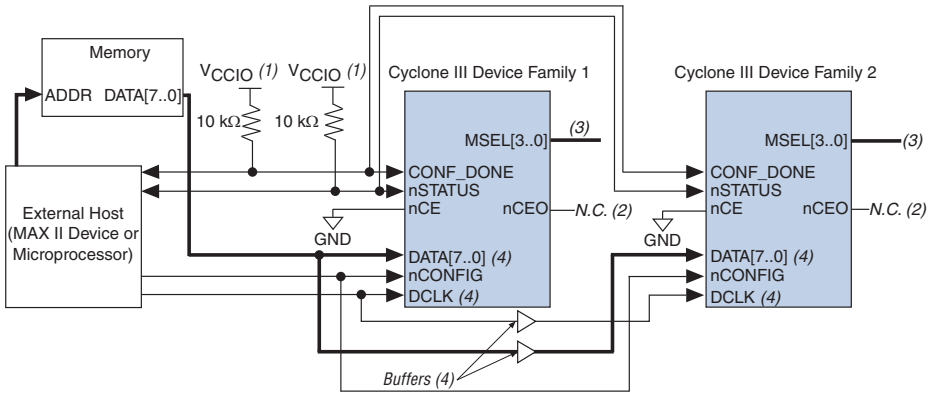
In a multi-device FPP configuration, the `nCE` pin of the first device is connected to GND while its `nCEO` pin is connected to the `nCE` pin of the next device in the chain. The `nCE` input of the last device comes from the previous device while its `nCEO` pin is left floating. After the first device 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. The 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.

If a system has multiple devices that contain the same configuration data, tie all device nCE inputs to GND and leave nCEO pins floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals can 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 9–22 shows multi-device FPP configuration when both Cyclone III device family is receiving the same configuration data.

Figure 9–22. Multi-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 9–22:

(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) 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 MSEL[3..0], refer to Table 9–7 on page 9–11. 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 equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

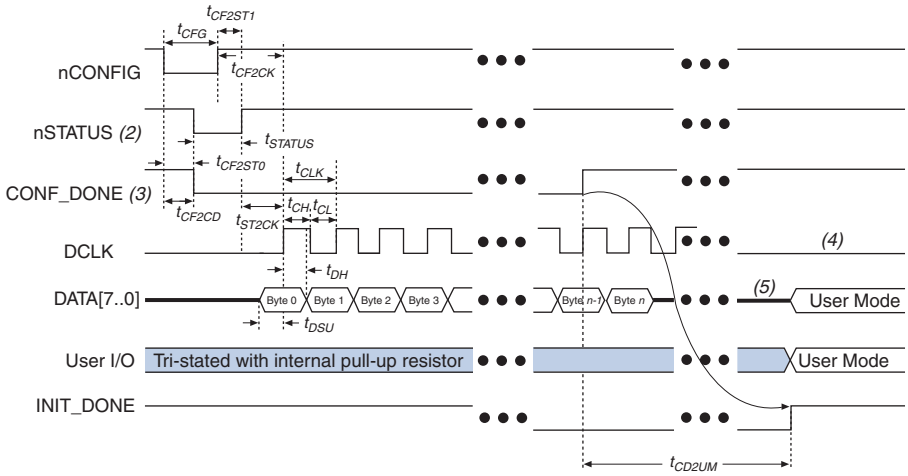
You can use a single configuration chain to configure Cyclone III device family with other Altera devices that support the FPP configuration. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device starts reconfiguration in all devices, tie all the device CONF_DONE and nSTATUS pins together.

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*.

FPP Configuration Timing

Figure 9–23 shows the timing waveform for FPP configuration when using an external host.

Figure 9–23. FPP Configuration Timing Waveform (1)



Notes to Figure 9–23:

(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 III device family 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 user I/O pin after configuration; the state of the pin depends on the dual-purpose pin settings.

Table 9–14 lists the FPP configuration timing parameters for Cyclone III device family.

Table 9–14. FPP Timing Parameters for Cyclone III Device Family (Part 1 of 2)

Symbol	Parameter	Minimum	Maximum	Unit
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 (1)	µs
t_{CF2ST1}	`nCONFIG` high to `nSTATUS` high	—	230 (1)	µs
t_{CF2CK}	`nCONFIG` high to first rising edge on `DCLK`	230 (1)	—	µs
t_{ST2CK}	`nSTATUS` high to first rising edge of `DCLK`	2	—	µs
t_{DSU}	`DATA` setup time before rising edge on `DCLK`	5	—	ns
t_{DH}	`DATA` hold time after rising edge on `DCLK`	0	—	ns
t_{CH}	`DCLK` high time	3.2	—	ns
t_{CL}	`DCLK` low time	3.2	—	ns
t_{CLK}	`DCLK` period	7.5	—	ns
f_{MAX}	`DCLK` frequency	—	100 (3)	MHz
t_{CD2UM}	`CONF_DONE` high to user mode (2)	300	650	µs

Table 9–14. FPP Timing Parameters for Cyclone III Device Family (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Unit
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} + (initialization clock cycles × CLKUSR period) (4)	—	—

Notes to Table 9–14:

(1) This value is applicable if users do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(2) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device.

(3) Cyclone III EP3C5, EP3C10, EP3C16, EP3C25, and EP3C40 devices support a DCLK f_{MAX} of 133 MHz. Cyclone III EP3C55, EP3C80, EP3C120 and all the Cyclone III LS devices support a DCLK f_{MAX} of 100 MHz.

(4) For more information about the initialization clock cycles required in Cyclone III device family, refer to Table 9–5 on page 9–10.

JTAG Configuration

JTAG has developed a specification for boundary-scan testing. This boundary-scan test (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 operating normally. You can also use the JTAG circuitry to shift configuration data into the device. The Quartus II software automatically generates **.sof**s that are used for JTAG configuration with a download cable in the Quartus II software programmer.

For more information about JTAG boundary-scan testing, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices* chapter.

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

For the Cyclone III LS device, JTAG programming is disabled if the device was already configured using the PS or AS mode. After POR, the Cyclone III LS device allows only mandatory JTAG 1149.1 instructions (BYPASS, SAMPLE/RELOAD, EXTEST, and FACTORY). For more information, refer to "JTAG Instructions" on page 9–60.

The four required pins for a device operating in JTAG mode are TDI, TDO, TMS, and TCK. The TCK pin has an internal weak pull-down resistor while the TDI and TMS pins have weak internal pull-up resistors (typically 25 kΩ). The TDO output pin is powered by V_{CCIO} in I/O bank 1. All the JTAG input pins are powered by the V_{CCIO} pin. All the JTAG pins support only LVTTL I/O standard. All user I/O pins are tri-stated during JTAG configuration. Table 9–15 lists the function of each JTAG pin.

The TDO output is powered by the V_{CCIO} power supply of I/O bank 1.

For more information about how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices* chapter.

Table 9–15. Dedicated JTAG Pins

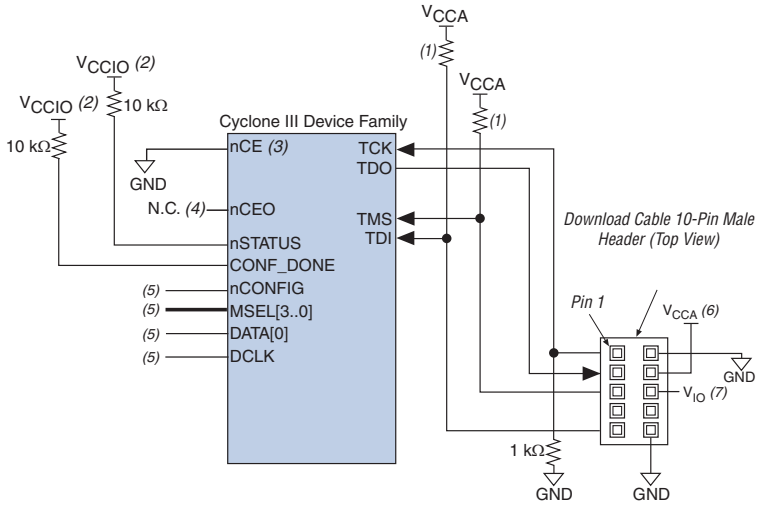
Pin Name	Pin Type	Description
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data shifts in on the rising edge of TCK. The TDI pin is powered by the V_{CCIO} supply. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to V_{CC}.
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. The TDO pin is powered by V_{CCIO} in I/O bank 1. 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. The TMS pin is powered by the V_{CCIO} supply. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to V_{CC}.
TCK	Test clock input	Clock input to the BST circuitry. Some operations occur at the rising edge while others occur at the falling edge. The TCK pin is powered by the V_{CCIO} supply. If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to GND.

You can download data to the device on the PCB through the USB-Blaster, MasterBlaster, ByteBlaster II, ByteBlasterMV download cable, and Ethernet-Blaster communications cable during JTAG configuration. Configuring devices using a cable is similar to programming devices in-system. Figure 9–24 and Figure 9–25 show the JTAG configuration of a single Cyclone III device family.

For device V_{CCIO} of 2.5, 3.0, and 3.3 V, refer to Figure 9–24. 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}, and you must pull TCK to ground.

For device V_{CCIO} of 1.2, 1.5, and 1.8 V, refer to Figure 9–25. You can power up the V_{CC} of the download cabled with the supply from V_{CCIO}.

Figure 9–24. 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 9–24:

(1) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

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

(3) The `nCE` pin must be connected to GND or driven low for successful JTAG configuration.

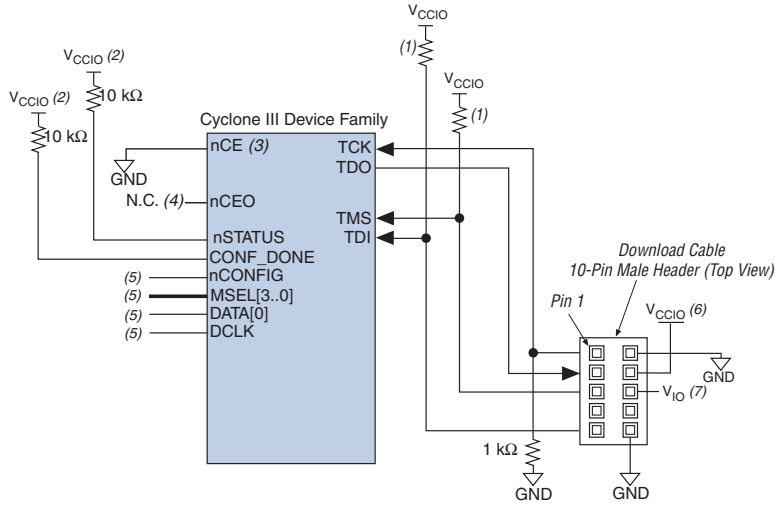
(4) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(5) Connect the `nCONFIG` and `MSEL[3..0]` 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[3..0]` pins to ground. In addition, pull `DCLK` and `DATA[0]` either high or low, whichever is convenient on your board.

(6) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, ByteBlasterMV, or Ethernet Blaster 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) 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*. In USB-Blaster, ByteBlaster II, ByteBlasterMV, and Ethernet Blaster, this pin is a no connect.

Figure 9–25. 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 9–25:

(1) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

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

(3) The `nCE` must be connected to GND or driven low for successful JTAG configuration.

(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) Connect the `nCONFIG` and `MSEL[3..0]` 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[3..0]` pins to ground. In addition, pull `DCLK` and `DATA[0]` either high or low, whichever is convenient on your board.

(6) Power up the V_{CC} of the ByteBlaster II, USB-Blaster, or Ethernet Blaster cable with supply from V_{CCIO}. The ByteBlaster II, USB-Blaster, and Ethernet 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*, *USB-Blaster Download Cable User Guide* and *Ethernet Blaster Communications Cable User Guide*.

(7) 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.

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 upon completion. At the end of configuration, the software checks the state of `CONF_DONE` through the JTAG port. When the Quartus II software generates a **.jam** for a multi-device chain, it contains instructions to have all devices in the chain initialize 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 cycle to perform device initialization.

Cyclone III device family has dedicated JTAG pins that function as JTAG pins. You can perform JTAG testing on Cyclone III device family before, during, and after configuration. Cyclone III device family supports 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 using the ACTIVE_DISENGAGE and CONFIG_IO instructions.

The CONFIG_IO instruction allows I/O buffers to be configured using the JTAG port and when issued after the ACTIVE_DISENGAGE instruction interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Cyclone III device family 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 III device family, 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 CONFIG_IO, ACTIVE_DISENGAGE, and ACTIVE_ENGAGE JTAG instructions in Cyclone III device family. For more information about the instruction flow, refer to "JTAG Instructions" on page 9–60.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Cyclone III device family do not affect JTAG boundary-scan or programming operations. Toggling these pins does not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration, consider the dedicated configuration pins. Table 9–16 lists how these pins must be connected during JTAG configuration.

Table 9–16. Dedicated Configuration Pin Connections During JTAG Configuration

Signal	Description
nCE	On all Cyclone III device family in the chain, nCE must be driven low by connecting it to ground, pulling it low using a resistor or driving it by some control circuitry. For devices that are also in multi-device AS, AP, PS, or FPP configuration chains, the nCE pins must be connected to GND during JTAG configuration or JTAG configured in the same order as the configuration chain.
nCEO	On all Cyclone III device family in the chain, nCEO is left floating or connected to the nCE of the next device.
MSEL[3..0]	These pins must not be left floating. These pins support whichever non-JTAG configuration that is 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 using a resistor or driven high by some control circuitry.
nSTATUS	Pull to the V_{CCIO} supply of the bank in which the pin resides using 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 using a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin must be pulled up to the 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.

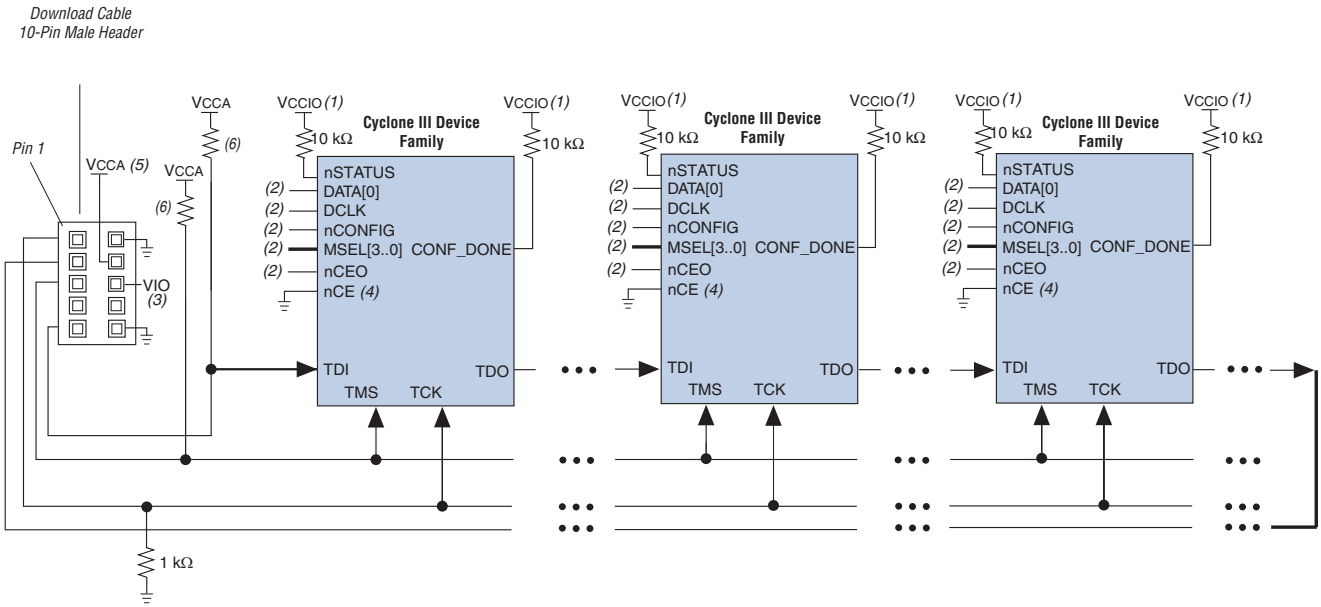
When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the `TCK`, `TDI`, 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 using JTAG BST circuitry. Figure 9–26 and Figure 9–27 show a multi-device JTAG configuration.

For the device V_{CCIO} of 2.5, 3.0, and 3.3 V, refer to Figure 9–26. 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 V_{CCIO} of 1.2, 1.5, and 1.8 V, refer to Figure 9–27. You can power up the V_{CC} of the download cable with the supply from V_{CCIO}.

Figure 9–26. 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 9–26:

(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[3..0]` 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[3..0]` pins to ground. In addition, pull `DCLK` and `DATA[0]` either high or low, whichever is convenient on your board.

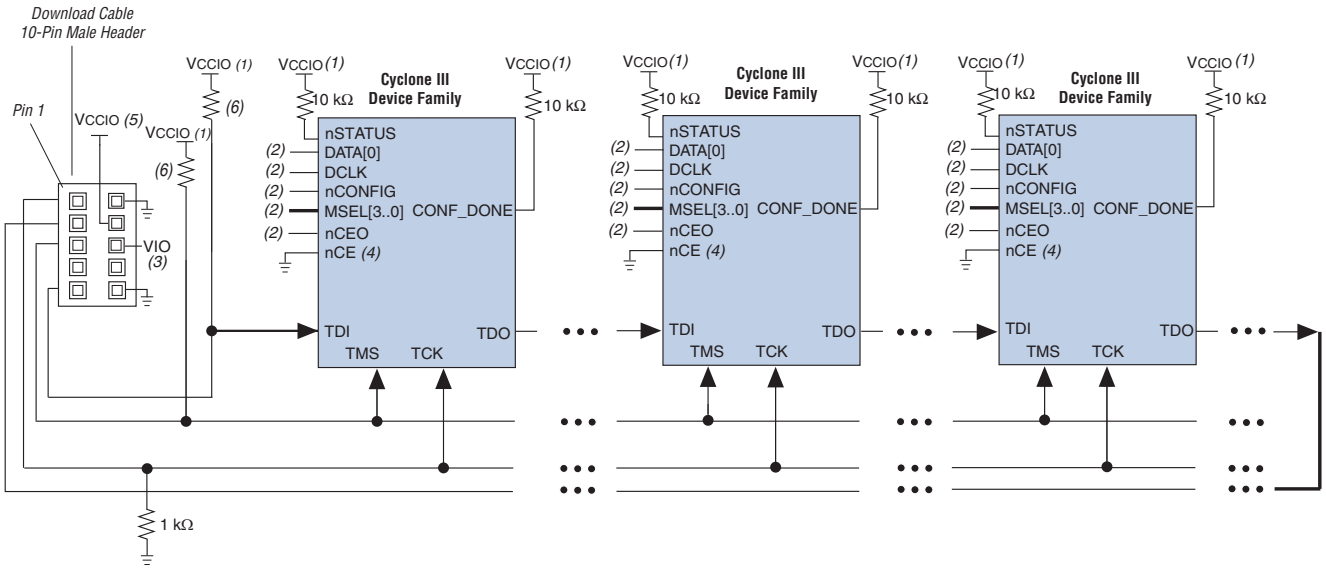
(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the ByteBlasterMV cable, this pin is a no connect. In the USB-Blaster and ByteBlaster II cables, this pin is connected to `nCE` when it is used for AS programming, otherwise it is a no connect.

(4) The `nCE` pin must be connected to ground 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 User Guide*.

(6) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

Figure 9–27. 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 9–27:

(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[3..0]` 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[3..0]` pins to ground. In addition, pull `DCLK` and `DATA[0]` 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) The `nCE` pin must be connected to ground 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) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

All I/O inputs must maintain a maximum AC voltage of 4.1 V. If a non-Cyclone III device family is cascaded in the JTAG-chain, `TDO` of the non-Cyclone III device family driving into `TDI` of the Cyclone III device family must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

The `nCE` pin must be connected to GND or driven low during JTAG configuration. In multi-device AS, AP, PS, and FPP configuration chains, the `nCE` pin of the first device is connected to GND while its `nCEO` pin is connected to the `nCE` pin of the next device in the chain. The inputs of the `nCE` pin of the last device come from the previous device while its `nCEO` pin is left floating. In addition, 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, every device must be configured when you perform JTAG configuration.

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 9–26 or Figure 9–27, in which each of the `CONF_DONE` and `nSTATUS` signals are isolated so that each device can enter user mode individually.

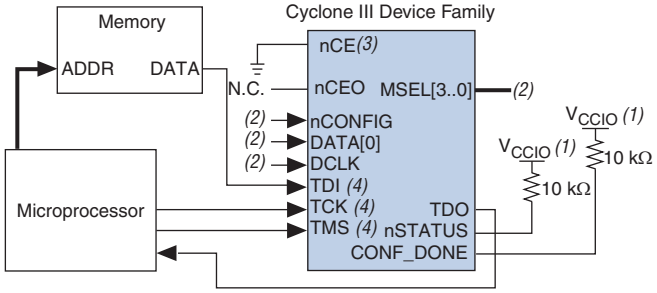
After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, ensure that the nCE pins are connected 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 pin 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 III device family 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 9–28 shows JTAG configuration of a Cyclone III device family with a microprocessor.

Figure 9–28. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 9–28:

(1) The pull-up resistor must be connected to a supply that provides an acceptable input signal for all devices in the chain.

(2) Connect the nCONFIG and MSEL[3..0] 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[3..0] pins to ground. In addition, pull DCLK and DATA[0] either high or low, whichever is convenient on your board.

(3) The nCE pin must be connected 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 equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 9–7.

Configuring Cyclone III Device Family 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 III Device Family with the JRunner Software Driver

The JRunner software driver allows you to configure Cyclone III device family 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.

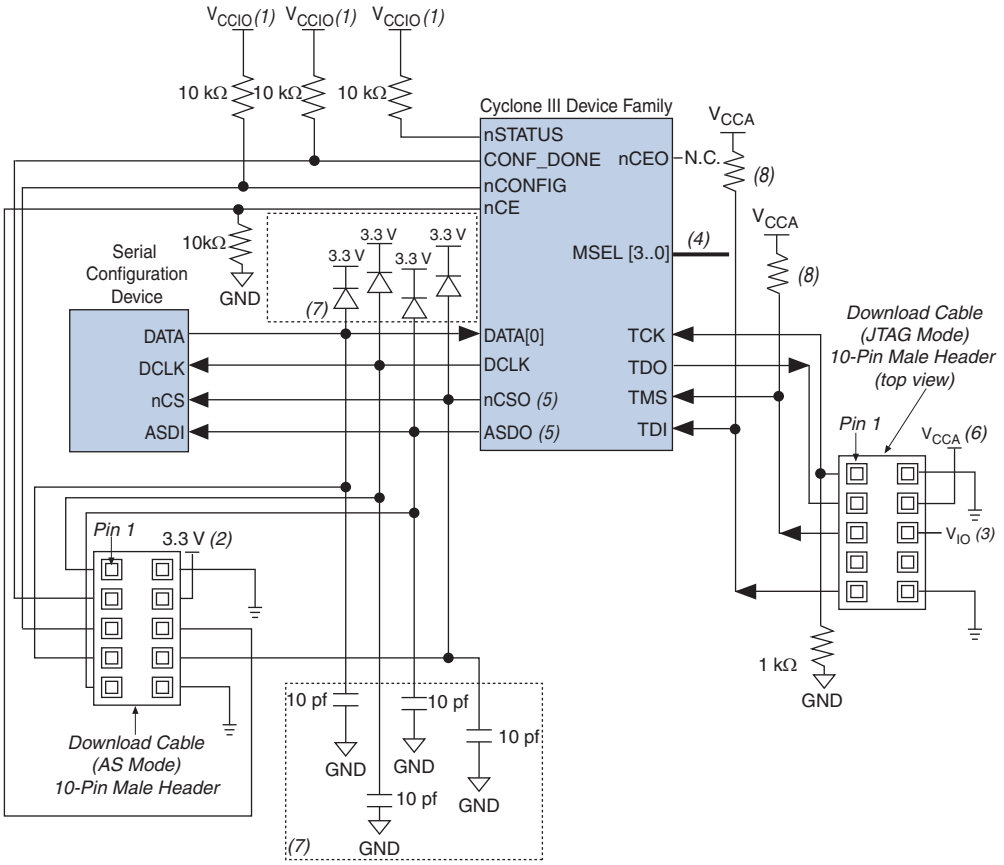
The **.rbf** used by the JRunner software driver cannot be a compressed **.rbf** because the JRunner software driver uses JTAG-based configuration. During JTAG-based configuration, the real-time decompression feature is not available.

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 9–29). This setup uses two 10-pin download cable headers on the board. One download cable is used in JTAG mode to configure the Cyclone III device family directly using the JTAG interface. The other download cable is used in AS mode to program the serial configuration device in-system using the AS programming interface. The `MSEL[3..0]` pins must be set to select AS configuration mode (Table 9–7 on page 9–11). If you try configuring the device using both schemes simultaneously, the JTAG configuration takes precedence and the AS configuration terminates.

Figure 9–29. Combining JTAG and AS Configuration Schemes



Notes to Figure 9–29:

(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 ByteBlaster II, USB-Blaster, or Ethernet 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. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In ByteBlasterMV, this pin is a no connect. In USB-Blaster and ByteBlaster II, 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[3..0]` for AS configuration schemes, refer to Table 9–7 on page 9–11. Connect the MSEL pins directly to V_{CCA} or GND.

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

(6) Power up V_{CC} of the ByteBlaster II, USB-Blaster, ByteBlasterMV, or Ethernet Blaster 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) You must place the diodes and capacitors as close as possible to the Cyclone III device family. For effective voltage clamping, Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes. For more information about the interface guidelines using Schottky diodes, refer to *AN 523: Cyclone III Configuration Interface Guidelines with EPCS Devices.*

(8) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

Programming Serial Configuration Devices In-System Using the JTAG Interface

Cyclone III device family in a single-device or in a multiple-device chain supports in-system programming of a serial configuration device with the JTAG interface using the SFL design. The intelligent host or download cable of the board can use the four JTAG pins on the Cyclone III device family 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 III device family that uses its 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. When using this feature, the slave devices in the multiple device chain which are configured by the serial configuration device do not need to be configured. To use this feature successfully, set the `MSEL[3..0]` pins of the master device to select the AS configuration scheme (Table 9–7 on page 9–11). The serial configuration device in-system programming through the Cyclone III device family JTAG interface has three stages, which are described in the following sections:

- "Loading the SFL Design" on page 9–58
- "ISP of the Configuration Device" on page 9–59
- "Reconfiguration" on page 9–60

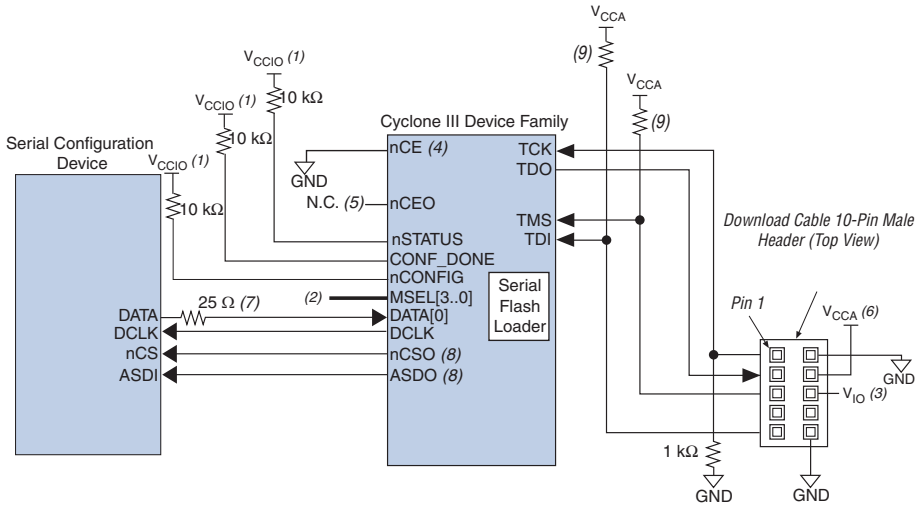
Loading the SFL Design

The SFL design is a design inside the Cyclone III device family that bridges the JTAG interface and the 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 a 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 9–30 shows the JTAG configuration of a single Cyclone III device family with a SFL design.

Figure 9–30. Programming Serial Configuration Devices In-System Using the JTAG Interface



Notes to Figure 9–30:

(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[3..0]` for AS configuration schemes, refer to Table 9–7 on page 9–11. 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. V_{IO} must match the V_{CCA} of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In ByteBlasterMV, this pin is a no connect. In USB-Blaster, ByteBlaster II, and Ethernet Blaster, this pin is connected to `nCE` when it is used for AS programming, otherwise it 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 ByteBlaster II, USB-Blaster, ByteBlasterMV, or Ethernet Blaster 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) Connect the series resistor at the near end of the serial configuration device.

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

(9) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

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 III device family JTAG interface. The JTAG interface sends the programming data for the serial configuration device to the Cyclone III device family first. The Cyclone III device family 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 III device family does not reconfigure by itself. 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 III device family and the serial configuration device configures all the devices in the chain with your user design.

For more information about SFL, refer to *AN 370: Using the Serial FlashLoader with Quartus II Software.*

JTAG Instructions

This section describes the instructions that are necessary for JTAG configuration for the Cyclone III device family. Table 9–17 lists the supported JTAG instructions.

Table 9–17. JTAG Instructions

JTAG Instruction	Cyclone III Device	Cyclone III LS Device
CONFIG_IO	✓	✓
ACTIVE_DISENGAGE	✓	✓
ACTIVE_ENGAGE	✓	✓
EN_ACTIVE_CLK	✓	—
DIS_ACTIVE_CLK	✓	—
APFC_BOOT_ADDR	✓	—
FACTORY (1)	—	✓
KEY_PROG_VOL (2)	—	✓
KEY_CLR_VREG (2)	—	✓

Notes to Table 9–17:

(1) In Cyclone III LS devices, the CONFIG_IO, ACTIVE_DISENGAGE, PULSE_NCONFIG, and PROGRAM instructions are supported, provided that the FACTORY instruction is executed. It is not necessary to execute the FACTORY instruction prior to the JTAG configuration in Cyclone III devices because this instruction is used for Cyclone III LS devices only.

(2) Use the KEY_PROG_VOL and KEY_CLR_VREG instructions for the design security feature. For more information, refer to "Design Security" on page 9–70.

For more information about the JTAG binary instruction code, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing for Cyclone III Devices* chapter.

For Cyclone III LS devices, the device can only allow mandatory JTAG 1149.1 instructions after POR. These instructions are BYPASS, SAMPLE/PRELOAD, EXTEST and FACTORY. To enable the access of other JTAG instructions, issue the FACTORY instruction. The FACTORY instruction puts the device in a state in which it is ready for in-house testing and board-level testing and it must be executed before configuration starts. When this instruction is executed, the CRAM bits content and volatile key are cleared and the device is reset.

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 III device family or waiting for a configuration device to complete configuration. After the configuration is interrupted and JTAG testing is complete, the part must be reconfigured using the PULSE_NCONFIG JTAG instruction or by pulsing the nCONFIG pin low.

You can issue the CONFIG_IO instruction any time during user mode. The CONFIG_IO instruction cannot be issued when nCONFIG pin is asserted low (during power up) or immediately after issuing a JTAG instruction that triggers reconfiguration. For more information about the wait-time for issuing the CONFIG_IO instruction, refer to Table 9–18.

When using CONFIG_IO instruction, you must meet the following timing restrictions:

- CONFIG_IO instruction cannot be issued during the nCONFIG pin low
- Observe 230 µs minimum wait time after any of the following conditions are met:
  - nCONFIG pin goes high
  - Issuing the PULSE_NCONFIG instruction
  - Issuing the ACTIVE_ENGAGE instruction, before issuing the CONFIG_IO instruction
- Wait 230 µs after power up with nCONFIG pin high before issuing the CONFIG_IO instruction (or wait for the nSTATUS pin to go high)

Table 9–18. Wait Time for Issuing the CONFIG_IO Instruction

Wait Time	Time
Wait time after the nCONFIG pin is released	230 µs
Wait time after PULSE_NCONFIG or ACTIVE_ENGAGE is issued	230 µs

Use the ACTIVE_DISENGAGE instruction with CONFIG_IO instruction to interrupt configuration. Table 9–19 lists the sequence of instructions to use for various CONFIG_IO usage scenarios.

Table 9–19. JTAG CONFIG_IO (without JTAG_PROGRAM) Instruction Flows (1) (Part 1 of 2)

JTAG Instruction	Configuration Scheme and Current State of the Cyclone III Device Family											
	Prior to User Mode (Interrupting Configuration)				User Mode				Power Up			
	PS	FPP	AS	AP (4)	PS	FPP	AS	AP (4)	PS	FPP	AS	AP (4)
FACTORY	NA	NA	NA	NA	NA	NA	NA	NA	R	R	R	NA
ACTIVE_DISENGAGE	0	0	0	0	0	0	0	0	—	—	—	—
CONFIG_IO	R	R	R	R	R	R	R	R	NA	NA	NA	NA
JTAG Boundary Scan Instructions (no JTAG_PROGRAM)	0	0	0	0	0	0	0	0	—	—	—	—

Table 9–19. JTAG CONFIG_IO (without JTAG_PROGRAM) Instruction Flows (1) (Part 2 of 2)

JTAG Instruction	Configuration Scheme and Current State of the Cyclone III Device Family											
	Prior to User Mode (Interrupting Configuration)				User Mode				Power Up			
	PS	FPP	AS	AP (4)	PS	FPP	AS	AP (4)	PS	FPP	AS	AP (4)
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 9–19:

(1) "R" indicates that the instruction is to be executed 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.

(4) AP configuration is for Cyclone III devices only.

The CONFIG_IO instruction does not hold the nSTATUS pin low until reconfiguration. You must disengage the active configuration controllers (AS and AP) by issuing the ACTIVE_DISENGAGE and ACTIVE_ENGAGE instructions when the 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 9–20). This puts the active configuration controllers into the idle state. The active configuration controller is re-engaged after user mode is reached using JTAG programming (Table 9–20).

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 the nCONFIG pin 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 the reconfiguration in configuration modes; therefore, it is not necessary to pull the nCONFIG pin 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 active configuration controller is the AS controller when the MSEL pins are set to AS configuration scheme and the AP controller when the MSEL pins are set to the AP configuration scheme. The two purposes of placing the active controllers in an idle state are:

- To ensure that they are not trying to configure the device in their respective configuration modes 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 III device family if the MSEL pins are set to an active configuration scheme (AS or AP). If the ACTIVE_DISENGAGE instruction is issued during a passive configuration scheme (PS or FPP), it has no effect on the Cyclone III device family. Similarly, the CONFIG_IO instruction is issued after an ACTIVE_DISENGAGE instruction, but is no longer required to properly halt configuration. Table 9–20 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 9–20. JTAG Programming Instruction Flows *(1)*

JTAG Instruction	Configuration Scheme and Current State of the Cyclone III Device											
	Prior to User Mode (Interrupting Configuration)				User Mode				Power Up			
	PS	FPP	AS	AP (2)	PS	FPP	AS	AP (2)	PS	FPP	AS	AP (2)
FACTORY	NA	NA	NA	NA	NA	NA	NA	NA	R	R	R	NA
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

Notes to Table 9–20:

(1) "R" indicates that the instruction is required to 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 to be executed in this mode.

(2) AP configuration is for Cyclone III devices only.

In AS or AP configuration schemes, the ACTIVE_DISENGAGE instruction puts the active configuration controllers into idle state. If a successful JTAG programming is executed, the active controllers are automatically re-engaged after user mode is reached using JTAG programming. This causes the active controllers to transition to their respective user mode states.

If JTAG programming fails to get the Cyclone III device family to enter user mode and re-engage active programming, there are available methods to achieve this for the AS or AP configuration schemes:

- When in the 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.
- When in the 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 III device family in the active configuration scheme specified by the MSEL pin settings.

The ACTIVE_ENGAGE instruction functions as the PULSE_NCONFIG instruction when the device is in passive configuration schemes (PS or FPP). 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 (AS or AP) controllers.

Changing the Start Boot Address of the AP Flash

In the AP configuration scheme, for Cyclone III 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 III 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 9–21 through Table 9–23 describe the connections and functionality of all the configuration-related pins on Cyclone III device family.

Table 9–21 lists the Cyclone III device family pin configuration.

Table 9–21. Cyclone III Device Family Configuration Pin Summary

Bank	Description	Input/Output	Dedicated	Powered By	Configuration Mode
1	FLASH_nCE, nCSO	Output	—	V_{CCIO}	AS, AP (2)
6	CRC_ERROR	Output	—	V_{CCIO}/Pull-up (1)	Optional, all modes
1	DATA[0]	Input	Yes	V_{CCIO}	PS, FPP, AS
		Bidirectional		V_{CCIO}	AP (2)
1	DATA[1], ASDO	Input	—	V_{CCIO}	FPP
		Output		V_{CCIO}	AS
		Bidirectional		V_{CCIO}	AP (2)
8	DATA[7..2]	Input	—	V_{CCIO}	FPP
		Bidirectional		V_{CCIO}	AP (2)
8	DATA[15..8]	Bidirectional		V_{CCIO}	AP (2)
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	Input	Yes	V_{CCIO}	PS, FPP
		Output		V_{CCIO}	AS, AP (2)
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[3..0]	Input	Yes	V_{CCINT}	All modes
1	TDO	Output	Yes	V_{CCIO}	JTAG
7	PADD[14..0]	Output	—	V_{CCIO}	AP (2)
8	PADD[19..15]	Output	—	V_{CCIO}	AP (2)
6	PADD[23..20]	Output	—	V_{CCIO}	AP (2)
1	nRESET	Output	—	V_{CCIO}	AP (2)
6	nAVD	Output	—	V_{CCIO}	AP (2)
6	nOE	Output	—	V_{CCIO}	AP (2)
6	nWE	Output	—	V_{CCIO}	AP (2)
5	DEV_OE	Input	—	V_{CCIO}	Optional, AP (2)
5	DEV_CLRn	Input	—	V_{CCIO}	Optional, AP (2)

Notes to Table 9–21:

(1) In Cyclone III devices, the CRC_ERROR pin is a dedicated output by default. Optionally, you can enable the CRC_ERROR pin as an open-drain output in the **CRC Error Detection** tab from the **Device and Pin Options** dialog box.

(2) AP configuration is for Cyclone III devices only.

Table 9–22 lists the dedicated configuration pins that must be connected properly on your board for successful configuration. Some of these pins may not be required for your configuration scheme.

Table 9–22. Dedicated Configuration Pins on Cyclone III Device Family (Part 1 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
MSEL [3..0]	N/A	All	Input	4-bit configuration input that sets the Cyclone III device family configuration scheme. These pins must be hardwired to V_{CCA} or GND. The MSEL[3..0] pins have internal 9-kΩ pull-down resistors that are always active. Some of the smaller devices or package options of Cyclone III devices do not have the MSEL[3] pin; therefore, the AP configuration scheme is not supported.
nCONFIG	N/A	All	Input	Configuration control input. Pulling this pin low with external circuitry during user mode causes the Cyclone III device family 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 III device family 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 III device family) 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, nCONFIG must be pulled low.
CONF_DONE	N/A	All	Bidirectional open-drain	■ Status output. The target Cyclone III device family drives the CONF_DONE pin low before and during configuration. After all configuration data is received without error and the initialization cycle starts, the target device releases CONF_DONE. ■ Status input. After all 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 the CONF_DONE pin.

Table 9–22. Dedicated Configuration Pins on Cyclone III Device Family (Part 2 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nCE	N/A	All	Input	Active-low chip enable. The nCE pin activates the Cyclone III device family with a low signal to allow configuration. The nCE pin must be held low during configuration, initialization, and user-mode. In a single-device configuration, it must be tied low. In a multi-device configuration, nCE of the first device is tied low while its nCEO pin is connected to the nCE pin of the next device in the chain. The nCE pin must also be held 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 is 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.
FLASH_nCE, nCSO (1), (2)	I/O	AS, AP (3)	Output	Output control signal from the Cyclone III device family to the serial configuration device in AS mode that enables the configuration device. This pin functions as the nCSO pin in AS mode and the FLASH_NCE pin in AP mode. Output control signal from the Cyclone III device to the parallel flash in AP mode that enables the flash. Connects to the CE# pin on the Micron P30 or P33 flash. (3) This pin has an internal pull-up resistor that is always active.
DCLK (1), (2)	N/A	PS, FPP, AS, AP (3)	Input (PS, FPP). Output (AS, AP (3))	In PS and FPP configuration, DCLK is the clock input used to clock data from an external source into the target Cyclone III device family. Data is latched into the device on the rising edge of DCLK. In AS mode, DCLK is an output from the Cyclone III device family 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 III device that provides timing for the configuration interface. (3) In active configuration schemes (AS or AP), this pin will be 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 passive schemes (PS or FPP) that use a control host, DCLK must be driven 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

Table 9–22. Dedicated Configuration Pins on Cyclone III Device Family (Part 3 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DATA[0] (1), (2)	I/O	PS, FPP, AS, AP (3)	Input (PS, FPP, AS). Bidirectional (AP) (3)	Data input. In serial configuration modes, bit-wide configuration data is presented to the target Cyclone III device family 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 and 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. (3)
DATA[1], ASDO (1), (2)	I/O	FPP, AS, AP (3)	Input (FPP), Output (AS). Bidirectional (AP) (3)	Data input in non-AS mode. Control signal from the Cyclone III device family to the serial configuration device in AS mode used to read out configuration data. The DATA[1] pin functions as the ASDO pin in AS mode. 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 PS configuration scheme, DATA[1] functions as 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 AP configuration scheme, which is for Cyclone III devices only, the byte-wide or word-wide configuration data is presented to the target Cyclone III device on DATA[7..0] or DATA[15..0], respectively. After AP configuration, DATA[1] is a dedicated bidirectional pin with optional user control. (3)
DATA[7..2]	I/O	FPP, AP (3)	Inputs (FPP). Bidirectional (AP) (3)	Data inputs. In AS or PS configuration schemes, they 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. The byte-wide or word-wide configuration data is presented to the target Cyclone III device on DATA[7..0] or DATA[15..0], respectively, in the AP configuration scheme (for Cyclone III devices only). After AP configuration, DATA[7..2] are dedicated bidirectional pins with optional user control. (3)
DATA[15..8]	I/O	AP (3)	Bidirectional	Data inputs. Word-wide configuration data is presented to the target Cyclone III device on DATA[15..0]. In PS, FPP, or AS configuration schemes, they 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.

Table 9–22. Dedicated Configuration Pins on Cyclone III Device Family (Part 4 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
PADD[23..0]	I/O	AP (3)	Output	24-bit address bus from the Cyclone III device to the parallel flash in AP mode. Connects to the A[24:1] bus on the Micron P30 or P33 flash.
nRESET	I/O	AP (3)	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 (3)	Output	Active-low address valid output. Driving the nAVD pin low during a read or write operation indicates to the parallel flash that 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 (3)	Output	Active-low output enable to the parallel flash. Driving the nOE pin low during a read operation enables the parallel flash outputs (DATA[15..0]). Connects to the OE# pin on the Micron P30 or P33 flash.
nWE	I/O	AP (3)	Output	Active-low write enable to the parallel flash. Driving the nWE pin low during a write operation 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 9–22:

(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) To tri-state the AS configuration pins in user mode, turn on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box in the **Configuration** tab. This option tri-states the DCLK, DATA0, nCSO, and ASDO pins.

(3) AP configuration scheme is for Cyclone III devices only.

Table 9–23 lists the optional configuration pins. If these optional configuration pins are not enabled 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 9–23. 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.
INIT_DONE	N/A if option is on. I/O if option is off.	Output open-drain	Status pin used to indicate when the device has initialized and is in user-mode. When nCONFIG is low and during the beginning of configuration, the INIT_DONE pin is tri-stated and pulled high due to an external 10-kΩ pull-up resistor. 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 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. This pin is enabled by turning on the **Enable device-wide reset (DEV_CLRn)** option in the Quartus II software.

Design Security

The design security feature is for Cyclone III LS devices only. The design security feature is not supported in Cyclone III devices.

Cyclone III LS Design Security Protection

Cyclone III LS device designs are protected from copying, reverse engineering, and tampering using configuration bitstream encryption and anti-tamper features.

Security Against Copying

The volatile key is securely stored in the Cyclone III LS device and cannot be read out through any interfaces. The information of your design cannot be copied because the configuration file read-back feature is not supported in Cyclone III LS devices.

Security Against Reverse Engineering

Reverse engineering from an encrypted configuration file is very difficult and time consuming because Cyclone III LS configuration file formats are proprietary and the file contains million of bits which require specific decryption. Reverse engineering the Cyclone III LS device is just as difficult because the device is manufactured on the advanced 60-nm process technology.

Security Against Tampering

Cyclone III LS devices support the following anti-tamper features:

- Ability to limit JTAG instruction set and provides protection against configuration data readback over the JTAG port
- Ability to clear contents of FPGA logic, configuration memory, user memory, and volatile key
- Error detection (ED) cycle indicator to core Cyclone III LS devices provide a pass or fail indicator at every ED cycle and visibility over intentional or unintentional change of CRAM bits.

For more information about anti-tamper protection for Cyclone III LS devices, refer to *AN 593: Anti-Tamper Protection for Cyclone III LS Devices*.

For more information about the implementation of secure configuration flow in Quartus II, refer to *AN 589: Using Design Security Feature in Cyclone III LS Devices*.

AES Decryption Block

The main purpose of the AES decryption block is to decrypt the configuration bitstream prior to entering configuration. Prior to receiving encrypted data, you must enter and store the 256-bit volatile key in the device with battery backup. The key is scrambled prior to storing it in the key storage to make it more difficult for anyone to retrieve the stored key using de-capsulation of the device.

Key Storage

Cyclone III LS devices support volatile key programming. Table 9–24 lists the volatile key features.

Table 9–24. Security Key Features (Part 1 of 2)

Volatile Key Features	Description
Key programmability	Reprogrammable and erasable
External battery	Required
Key programming method (1)	On-board

Table 9–24. Security Key Features (Part 2 of 2)

Volatile Key Features	Description
Design protection	Secure against copying, reverse engineering, and tampering

Note to Table 9–24:

(1) Key programming is carried out using the JTAG interface.

AES volatile key zeroization is supported in Cyclone III LS devices. The volatile key clear and key program JTAG instructions from the device core is supported to protect Cyclone III LS devices against tampering. You can clear and reprogram the key from the device core whenever tampering attempt is detected by executing the `KEY_CLR_VREG` and `KEY_PROG_VOL` JTAG instructions to clear and reprogram the volatile key, and then reset the Cyclone III LS device by pulling the `nCONFIG` pin low for at least 500 ns. When `nCONFIG` returns to a logic-high level and `nSTATUS` is released by the Cyclone III LS device, reconfiguration begins to configure the Cyclone III LS device with a benign or unencrypted configuration file. After configuration is successfully completed, observe the `cyclecomplete` signal from error detection block to ensure that reconfigured CRAM bits content is correct for at least one error detection cycle. You can also observe the `cyclecomplete` and `crcerror` signals for any unintentional CRAM bits change.

`cyclecomplete` is a signal that is routed from the error detection block to the core for the purpose of every complete error detection cycle. You must include the `cycloneiiils_crcblock` WYSIWYG atom in your design to use the `cyclecomplete` signal. For more information about the SEU mitigation, refer to the *SEU Mitigation in Cyclone III Devices* chapter.

V_{CCBAT} is a dedicated power supply for the volatile key storage and not shared with other on-chip power supplies, such as V_{CCIO} or V_{CC}. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition. The nominal voltage for this supply is 3.0 V, while its valid operating range is from 1.2 to 3.3 V. If you do not use the volatile security key, you may connect the V_{CCBAT} to a 1.8-V, 2.5-V, or 3.0-V power supply.

After power-up, wait for 200 ms (Standard POR) or 9 ms (Fast POR) before beginning the key programming to ensure that V_{CCBAT} is at its full rail.

As an example, BR1220 (-30°C to +80°C) and BR2477A (-40 C to +125°C) are lithium coin-cell type batteries used for volatile key storage purposes.

For more information about the battery specifications, refer to the *Cyclone III LS Device Data Sheet* chapter.

Cyclone III LS Design Security Solution

Cyclone III LS devices are SRAM-based devices. To provide design security, Cyclone III LS devices require a 256-bit volatile key for configuration bitstream encryption.

The Cyclone III LS design security feature provides routing architecture optimization for design separation flow with the Quartus II software. Design separation flow achieves both physical and functional isolation between design partitions.

For more information about the design separation flow, refer to *AN 567: Quartus II Design Separation Flow.*

You can carry out secure configuration in Steps 1–3, as shown in Figure 9–31:

1. Generate the encryption key programming file and encrypt the configuration data.

   The Quartus II configuration software uses the user-defined 256-bit volatile keys to generate a key programming file and an encrypted configuration file. The encrypted configuration file is stored in an external memory, such as a flash memory or a configuration device.

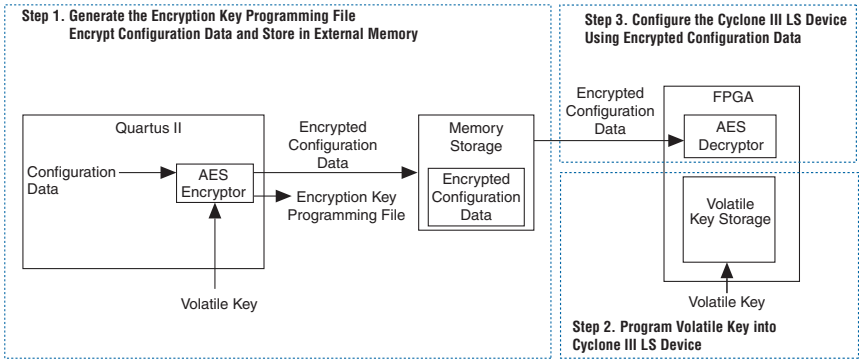
2. Program the volatile key into the Cyclone III LS device.

   Program the user-defined 256-bit volatile keys into the Cyclone III LS device through the JTAG interface.

3. Configure the Cyclone III LS device.

   At system power-up, the external memory device sends the encrypted configuration file to the Cyclone III LS device.

Figure 9–31. Cyclone III LS Secure Configuration Flow (1)



Note to Figure 9–31:

(1) Step 1, Step 2, and Step 3 correspond to the procedure detailed in "Cyclone III LS Design Security Solution".

Available Security Modes

There are several security modes available on Cyclone III LS devices, they are:

- Volatile Key
- No Key Operation
- `FACTORY` Mode

Volatile Key

Secure operation with volatile key programmed and required external battery—this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board-level testing only.

No Key Operation

Only unencrypted configuration bitstreams are allowed to configure the device.

FACTORY Mode

After power up, Cyclone III LS devices must be in `FACTORY` mode to program the volatile key. The `FACTORY` private JTAG instruction must be issued after the device successfully exits from POR and before the device starts loading the core configuration data to enable access to all other instructions from the JTAG pins. The device configuration data and AES volatile key are cleared if the `FACTORY` instruction is executed.

Table 9–25 lists the configuration bitstream and the configuration mode supported for each security mode.

Table 9–25. Security Modes Supported

Mode	Function	Configuration File	Allowed Configuration Mode
Volatile Key	Secure	Encrypted	PS with AES (without decompression). FPP with AES (without decompression). Remote update fast AS with AES (without decompression). Fast AS (without decompression).
	Board-Level Testing	Unencrypted	All configuration modes that do not engage the design security feature.
No Key	—	Unencrypted	All configuration modes that do not engage the design security feature.
`FACTORY`	Volatile Key Programming	—	—

Remote System Upgrade

Cyclone III devices support remote system upgrade in AS and AP configuration schemes. Cyclone III LS devices support remote system upgrade in the AS configuration scheme only. Remote system upgrade can also be implemented with advanced Cyclone III features such as real-time decompression of configuration data in the AS configuration scheme.

- The serial configuration device uses the AS configuration scheme to configure Cyclone III or Cyclone III LS devices
- The supported parallel flash uses the AP configuration scheme to configure Cyclone III devices
- Remote system upgrade is not supported in the multi-device configuration chain for any configuration scheme.

Functional Description

The dedicated remote system upgrade circuitry in Cyclone III device family manages remote configuration and provides error detection, recovery, and status information. User logic or a Nios® II processor implemented in the Cyclone III device family 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, and depends on the configuration scheme that you use.

The remote system upgrade process of Cyclone III device family involves the following steps:

1. A Nios II processor (or user logic) implemented in the Cyclone III device family 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 9–32 shows the steps required for performing remote configuration updates (the numbers in Figure 9–32 coincide with steps 1–4).

Figure 9–32. Functional Diagram of Cyclone III Device Family Remote System Upgrade

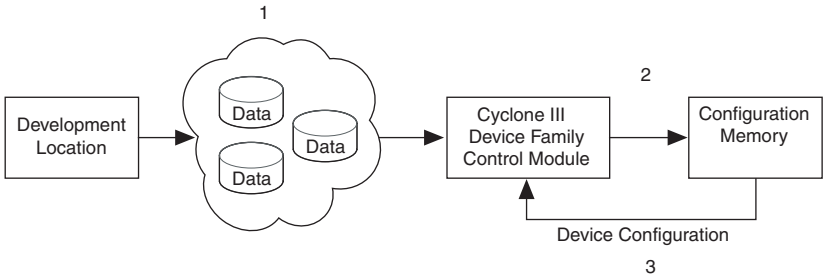
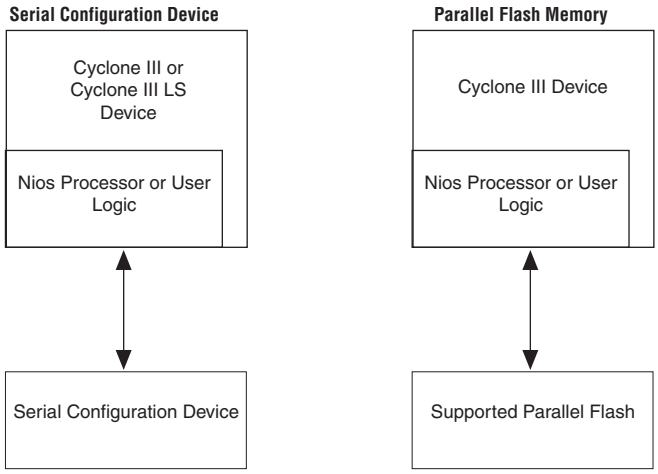


Figure 9–33 shows the block diagrams to implement remote system upgrade with the AS and AP configuration schemes.

Figure 9–33. Remote System Upgrade Block Diagrams for AS and AP Configuration Schemes



Remote system upgrade only supports single-device configuration.

When using remote system upgrade in Cyclone III devices, you must set the mode select pins (`MSEL [3.0]`) to the AS or AP configuration scheme. When using remote system upgrade in Cyclone III LS devices, you must set `MSEL [3..0]` to the AS configuration scheme. The MSEL pin setting in remote system upgrade mode is the same as standard configuration mode. Standard configuration mode refers to normal Cyclone III device family configuration mode with no support for remote system upgrades, and the remote system upgrade circuitry is disabled. When using remote system upgrade in Cyclone III device family, you must enable the **remote update mode** option setting in the Quartus II software. For more information, refer to "Enabling Remote Update" on page 9–76.

Enabling Remote Update

You can enable or disable remote update for Cyclone III device family 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 in the Quartus II software:

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 III device family 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 with the 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 dedicated circuitry. Application images implement user-defined functionality in the target Cyclone III device family. You can include the default application image functionality in the factory image.

Remote System Upgrade Mode

In remote update mode, the Cyclone III device family loads the factory configuration image after power-up. The user-defined factory configuration determines which application configuration is to be loaded and triggers a reconfiguration cycle. The factory configuration can also contain application logic.

When used with configuration memory, 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

When a Cyclone III device family is first powered up in remote update in the AS configuration scheme, 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` when using the AS configuration scheme. A factory configuration image is a bitstream for Cyclone III device family 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. This corresponds to the start address location 0x000000 in the serial configuration device.

When you use the AP configuration in Cyclone III devices, the Cyclone III 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

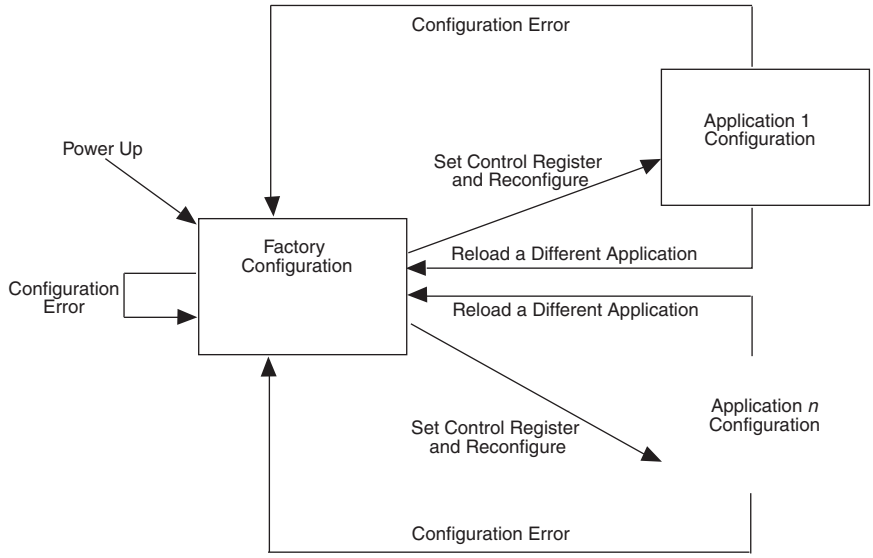
You can change the default factory configuration address to any desired address using the `APFC_BOOT_ADDR` JTAG instruction. The factory configuration image is stored in non-volatile memory and is never updated or modified using remote access. This corresponds to the default start address location 0x010000 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 "JTAG Instructions" on page 9–60.

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 which application configuration is to be loaded into the Cyclone III device family
- 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 9–34 shows the transitions between the factory and application configurations in remote update mode.

Figure 9–34. 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 "User Watchdog Timer" on page 9–85.

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 III device family, specifying the cause of the reconfiguration.

The following actions cause the remote system upgrade status register to be written:

- `nSTATUS` driven low externally
- Internal CRC error
- User watchdog timer time-out
- A configuration reset (logic array `nCONFIG` signal or external `nCONFIG` pin assertion)

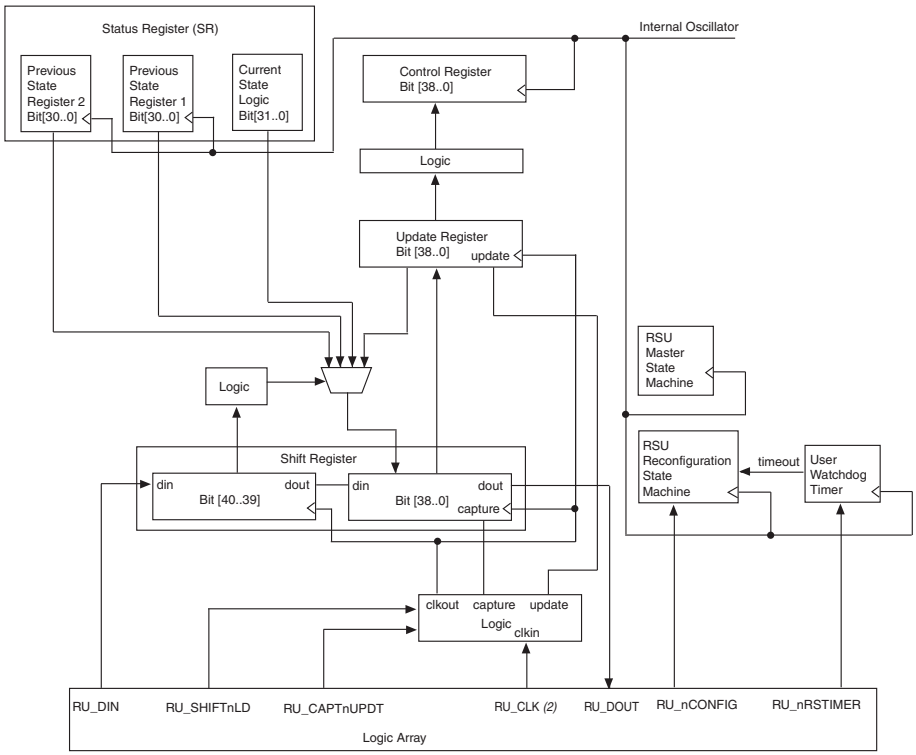
Cyclone III device family automatically load the factory configuration located at address `boot_address[23:0] = 24'b0` for the AS configuration scheme, and default address `boot_address[23:0] = 24'h010000` (or the updated address if the default address is changed) for the AP configuration scheme. This 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 III device family successfully load the application configuration, the devices enter user mode. In user mode, the soft logic (Nios II processor or state machine and the remote communication interface) assists the Cyclone III device family 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 explains the implementation of the Cyclone III device family remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces to the user-defined factory application configurations implemented in the Cyclone III device family 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 9–35 shows the data path of the remote system upgrade block.

Figure 9–35. Remote System Upgrade Circuit Data Path (1)



Notes to Figure 9–35:

(1) 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) RU_CLK refers to 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. These registers are listed in Table 9–26.

Table 9–26. 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 writes 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 to 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.

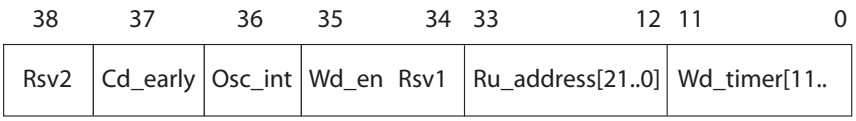
The control and status registers of the remote system upgrade are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the shift and update registers of the remote system upgrade are clocked by the maximum frequency of 40-MHz user clock input (`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 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.

The control register bit positions are shown in Figure 9–36 and listed in Table 9–27. In the figure, the numbers show the bit position of a setting in a register. For example, bit number 35 is the enable bit for the watchdog timer.

Figure 9–36. Remote System Upgrade Control Register



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 `CONF_DONE` pin is asserted too early, the device resets and then reconfigures the factory configuration image. The internal oscillator, as startup state machine clock (`Osc_int`) option bit, ensures a functional startup clock to eliminate the hanging of startup when enabled. When all option bits are turned on, they provide complete coverage for the programming and startup portions of the application configuration. It is strongly recommended that you turn 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.

Table 9–27. 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 9–27:

(1) Option bit for the application 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 III device family logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- External configuration reset (`nCONFIG`) assertion
- User watchdog timer time out

Table 9–28 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, and the MSEL pin setting is set to AS or AP configuration scheme. The status register bit in Table 9–28 lists the bit positions in a 32-bit logic.

Table 9–28. Remote System Upgrade Current State Logic Contents In Status Register *(1)*

Current State Logic	Status Register Bit	Definition	Description
Factory information (2)	31:30	Master State Machine current state	The current state of the RSU 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 part 1 (3)	31:30	Master State Machine current state	The current state of the RSU 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 part 2 (3)	31:30	Master State Machine current state	The current state of the RSU 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

Notes to Table 9–28:

(1) The MSEL pin setting is in the AS or AP configuration scheme.
(2) The RSU master state machine is in factory configuration.
(3) The RSU 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 9–29 lists the contents of the previous state register 1 and previous state register 2 in the status register when the MSEL pin setting is set to the AS or AP scheme. The status register bit in Table 9–29 shows the bit positions in a 31-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 9–29. Remote System Upgrade Previous State Register 1 and Previous State Register 2 Contents in Status Register (1)

Status Register Bit	Definition	Description
30	nCONFIG source	One-hot, active-high field that describes the reconfiguration source that caused the Cyclone III device family 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 III device family to leave the previous application configuration.
23:0	Boot address	The address used by the configuration scheme to load the previous application configuration.

Note to Table 9–29:

(1) The MSEL pin settings are in the AS configuration scheme.

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

Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (Table 9–26 on page 9–81). 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 `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 9–30 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 9–30. 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 stalling the device indefinitely. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Cyclone III device family.

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. 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.

Table 9–31 lists the operating range of the 10-MHz internal oscillator.

Table 9–31. 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 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.

The user watchdog timer is not enabled during the configuration cycle of the device. Errors during configuration are detected by the CRC engine. Also, the timer is disabled for factory configuration. Functional errors must not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

By default, the user watchdog timer is disabled in factory configurations and enabled in user-mode application configurations. If you do not want to use the watchdog timer feature, disable this feature in the factory configuration.

Quartus II Software Support

Implementation in your design requires a remote system upgrade interface between the Cyclone III device family logic array and the 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, the ALTREMOTE_UPDATE megafunction and the 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 9–32 lists the revision history for this document.

Table 9–32. Document Revision History (Part 1 of 2)

Date	Version	Changes
August 2012	2.2	Updated Micron P30 and P33 Parallel NOR flash devices.
July 2012	2.1	Finalized Table 9–3, Table 9–13, and Table 9–14.
December 2011	2.0	■ Updated “Configuration Features” on page 9–2, “Reset” on page 9–8,“AS Configuration (Serial Configuration Devices)” on page 9–12, “Single-Device AS Configuration” on page 9–13, “AP Configuration Supported Flash Memory” on page 9–24, “Single-Device AP Configuration” on page 9–25, “JTAG Configuration” on page 9–48, and “User Watchdog Timer” on page 9–85. ■ Removed the “Overriding the Internal Oscillator” section from “JTAG Configuration”. ■ Updated Figure 9–11, Figure 9–24, Figure 9–25, Figure 9–26, Figure 9–27, Figure 9–29, Figure 9–30. ■ Updated Table 9–13, Table 9–18, and Table 9–22. ■ Replaced links to *AN 386: Using the Parallel Flash Loader with the Quartus II Software links to Parallel Flash Loader Megafunction User Guide.*

Table 9–32. Document Revision History (Part 2 of 2)

Date	Version	Changes
December 2009	1.2	■ Updated Table 9–7, Table 9–10, Table 9–22, and Table 9–28. ■ Updated Figure 9–23 and Figure 9–30. ■ Updated the “Programming Serial Configuration Devices” and “Security Against Tampering” sections. ■ Minor changes to the text.
July 2009	1.1	Made a minor correction to the part number.
June 2009	1.0	Initial release.

10. Hot-Socketing and Power-On Reset in the Cyclone III Device Family

CIII51011-3.4

The Cyclone® III device family (Cyclone III and Cyclone III LS devices) offers hot-socketing, which is also known as hot plug-in or hot swap, and power sequencing support without the use of any external devices. You can insert or remove Cyclone III device family or a board in a system during system operation without causing undesirable effects to the running system bus or the board that is inserted into the system.

The hot-socketing feature removes some of the difficulties that you encounter when you use Cyclone III device family on a PCB that contains a mixture of 3.3, 3.0, 2.5, 1.8, 1.5, and 1.2 V devices. With the hot-socketing feature of Cyclone III device family, you no longer need to ensure a proper power up sequence for each device on the board.

Cyclone III device family hot-socketing feature provides:

- Board or device insertion and removal without external components or board manipulation
- Support for any power-up sequence
- Non-intrusive I/O buffers to system buses during hot insertion

This chapter also describes the power-on reset (POR) circuitry in Cyclone III device family. The POR circuitry keeps the devices in the reset state until the power supplies are in operating range.

This chapter contains the following sections:

- "Hot-Socketing Specifications" on page 10–1
- "Hot-Socketing Feature Implementation" on page 10–3
- "POR Circuitry" on page 10–3

Hot-Socketing Specifications

Cyclone III device family is a hot-socketing compliant without the need for any external components or special design requirements. Hot-socketing support in Cyclone III device family has the following advantages:

- You can drive the device before power-up without damaging the device.
- I/O pins remain tristated during power-up. The device does not drive out before or during power-up, therefore not affecting other buses in operation.

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

ISO 9001:2008 Registered

Devices Driven Before Power-Up

You can drive signals into I/O pins, dedicated input pins, and dedicated clock pins of Cyclone III device family before or during power-up or power down without damaging the device. The Cyclone III device family supports any power-up or power down sequence (V_{CCIO}, V_{CCINT}) to simplify system level design.

I/O Pins Remain Tristated During Power-Up

The output buffers of Cyclone III device family are turned off during system power up or power down. Cyclone III device family does not drive out until the device is configured and working in recommended operating conditions. The I/O pins are tristated until the device enters user mode with a weak pull-up resistor (R) to V_{CCIO}.

You can power-up or power down the V_{CCIO}, V_{CCA}, and V_{CCINT} pins in any sequence. The V_{CCIO}, V_{CCA}, and V_{CCINT} pins must have a monotonic rise to their steady state levels. The maximum power ramp rate is 3 ms for fast POR time and 50 ms for standard POR time. The minimum power ramp rate is 50 µs. V_{CCIO} for all I/O banks must be powered up during device operation. All V_{CCA} pins must be powered to 2.5 V (even when PLLs are not used), and must be powered up and powered down at the same time. V_{CCD_PLL} must always be connected to V_{CCINT} through a decoupling capacitor and ferrite bead. During hot-socketing, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

Cyclone III device family meets the following hot-socketing specification:

- The hot-socketing DC specification is $| I_{IOPIN} | < 300$ uA
- The hot-socketing AC specification is $| I_{IOPIN} | < 8$ mA for the ramp rate of 10 ns or more

For ramp rates faster than 10 ns on I/O pins, $|I_{IOPIN}|$ is obtained with the equation $I = C\ dv/dt$, in which C is the I/O pin capacitance and dv/dt is the slew rate. The hot-socketing specification takes into account the pin capacitance but not board trace and external loading capacitance. You must consider additional or separate capacitance for trace, connector, and loading. I_{IOPIN} is the current for any user I/O pins on the device. The DC specification applies when all V_{CC} supplied to the device is stable in the powered-up or powered-down conditions.

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 ground 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 III device family are immune to latch up during hot-socketing.

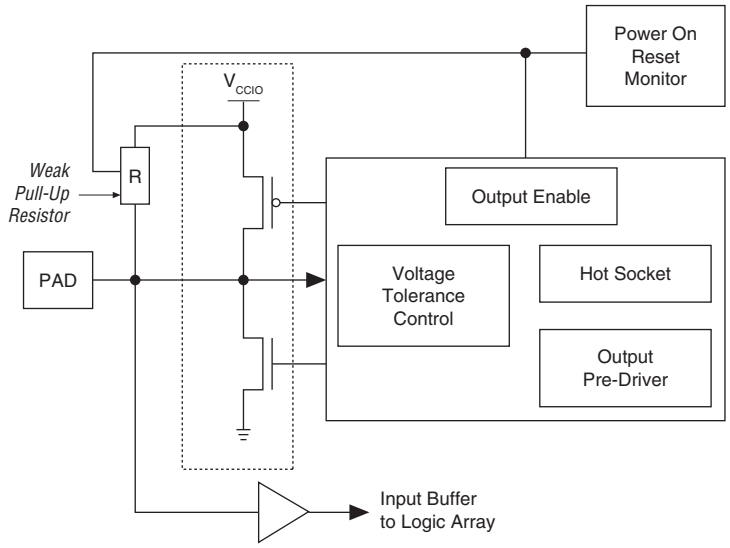
For more information about the hot-socketing specification, refer to the *Cyclone III Device Data Sheet* and *Cyclone III LS Device Data Sheet* chapters and the *Hot-Socketing and Power-Sequencing Feature and Testing for Altera Devices* white paper.

Hot-Socketing Feature Implementation

Each I/O pin has the circuitry shown in Figure 10–1. The hot-socketing circuit does not include `CONF_DONE`, `nCEO`, and `nSTATUS` pins to ensure that they are able to operate during configuration. Thus, it is expected behavior for these pins to drive out during power up and power down sequences.

Figure 10–1 shows the hot-socketing circuit block diagram for Cyclone III device family.

Figure 10–1. Hot-socketing Circuit Block Diagram for Cyclone III Device Family



The POR circuit monitors the voltage level of power supplies and keeps the I/O pins tristated until the device is in user mode. The weak pull-up resistor (R) in Cyclone III device family I/O element (IOE) keeps the I/O pins from floating. The 3.3-V tolerance control circuit permits the I/O pins to be driven by 3.3 V before V_{CCIO}, V_{CC}, and V_{CCA} supplies are powered up, and it prevents the I/O pins from driving out when the device is not in user mode.

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.

POR Circuitry

Cyclone III device family contains 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 tristated until the V_{CC} reaches the recommended operating levels. In addition, the POR circuitry also ensures the V_{CCIO} level of I/O banks 1, 6, 7, and 8 that contains configuration pins reach an acceptable level before configuration is triggered.

The POR circuit of the Cyclone III device monitors the V_{CCINT}, V_{CCIO} (banks 1, 6, 7, and 8), and V_{CCA} pins during power-on. The enhanced POR circuit of the Cyclone III LS device includes monitoring V_{CCBAT} to ensure that V_{CCBAT} is always at the minimum requirement voltage level.

The V_{CCBAT} power supply is the new design security feature power supply introduced for Cyclone III LS devices only. Cyclone III devices do not have V_{CCBAT} power supply.

After Cyclone III device family enters user mode, the POR circuit continues to monitor the V_{CCINT} or 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 III device family offers the Fast-On feature to support fast wake-up time applications. For Cyclone III device family, the `MSEL[3..0]` pin settings determine the POR time (t_{POR}) of the device. Fast POR ranges from 3 ms to 9 ms, while standard POR ranges from 50 ms to 200 ms.

If you cannot meet the maximum V_{CC} ramp time requirement, use an external component to hold `nCONFIG` low until the power supplies have reached their minimum recommend operating levels. Otherwise, the device may not properly configure and enter user mode.

For more information about the `MSEL[3..0]` pin settings, refer to the *Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family* chapter.

For more information about the V_{CCBAT} pin connection, refer to the *Cyclone III Device Family Pin Connection Guidelines*.

Document Revision History

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

Table 10–1. Document Revision History (Part 1 of 2)

Date	Version	Changes
July 2012	3.4	Updated tolerance control circuit voltage level in the "Hot-Socketing Feature Implementation" section.
December 2011	3.3	■ Updated "POR Circuitry" on page 10–3. ■ Updated hyperlinks. ■ Minor text edits.
December 2009	3.2	Minor changes to the text.
July 2009	3.1	Made minor correction to the part number.
June 2009	3.0	■ Updated chapter part number. ■ Updated "I/O Pins Remain Tristated During Power-Up" on page 10–2. ■ Updated "Hot-Socketing Feature Implementation" on page 10–3. ■ Updated "POR Circuitry" on page 10–4.

Table 10–1. Document Revision History (Part 2 of 2)

Date	Version	Changes
October 2008	1.2	■ Updated chapter to new template. ■ Added handnote to the “Cyclone III Hot-Socketing Specifications” section.
July 2007	1.1	■ Updated “I/O Pins Remain Tri-stated During Power-Up” section. ■ Updated Figure 10–3. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

11. SEU Mitigation in the Cyclone III Device Family

CIII51013-2.3

Dedicated circuitry built into the Cyclone® III device family (Cyclone III and Cyclone III LS devices) consists of a cyclical redundancy check (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

This chapter describes how to activate and use the error detection CRC feature in user mode and describes how to recover from configuration errors caused by CRC error. Using the CRC error detection feature for Cyclone III device family does not impact fitting or performance.

This chapter contains the following sections:

- "Error Detection Fundamentals" on page 11–1
- "Configuration Error Detection" on page 11–2
- "User Mode Error Detection" on page 11–2
- "Automated SEU Detection" on page 11–3
- "CRC_ERROR Pin" on page 11–3
- "Table 11–2 lists the CRC_ERROR pin." on page 11–4
- "Error Detection Block" on page 11–4
- "Error Detection Timing" on page 11–5
- "Software Support" on page 11–7
- "Recovering from CRC Errors" on page 11–10

Error Detection Fundamentals

Error detection determines if the data received through an input device is corrupted during transmission. In validating the data, the transmitter uses a function to calculate a checksum value for the data and appends the checksum to the original data frame. The receiver uses the same calculation methodology to generate a checksum for the received data frame and compares the received checksum to the transmitted checksum. If the two checksum values are equal, the received data frame is correct and no data corruption has occurred during transmission or storage.

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

The error detection CRC feature in Cyclone III device family puts theory into practice. In user mode, the error detection CRC feature in Cyclone III device family ensures the integrity of the configuration data.

Configuration Error Detection

In configuration mode, a frame-based CRC is stored in the configuration data and contains the CRC value for each data frame.

During configuration, Cyclone III device family calculates 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.

For Cyclone III device family, the CRC is computed by the Quartus® II software and downloaded into the device as part of the configuration bit stream. These devices store the CRC in the 32-bit storage register at the end of the configuration mode.

User Mode Error Detection

Soft errors are changes in a configuration random-access memory (CRAM) bit state due to an ionizing particle. Cyclone III device family has 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 *n*`CONFIG` to low).

The Cyclone III device family 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 III device family 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 `0x000000`, which results in a `0` on the output signal `CRC_ERROR`. 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 failure induced, you can restore the 32-bit CRC value to the correct CRC value using the same instruction and inserting the correct value.

☞ Be sure to read out the correct value before updating it with a known bad value.

In user mode, Cyclone III device family supports the CHANGE_EDREG JTAG instruction, which allows you to write to the 32-bit storage register. You can use Jam™ STAPL files (**.jam**) to automate the testing and verification process. This instruction can only be executed 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 switch to use the CRC circuit to check for real errors induced by an SEU.

Table 11–1 lists the CHANGE_EDREG JTAG instructions.

Table 11–1. 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, to clear the CRC error and restore the original CRC value, power cycle the device or perform the following procedure:

1. After the configuration completes, use JTAG instruction CHANGE_EDREG to shift out the correct precomputed CRC value and load the wrong CRC value to the CRC storage register. The CRC_ERROR pin will be asserted and shows that a CRC error is detected.
2. Use JTAG instruction CHANGE_EDREG to shift in the correct precomputed CRC value. The CRC_ERROR pin is deasserted and shows that the error detection CRC circuitry is working.

Automated SEU Detection

Cyclone III device family offers 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 III device family, 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, and you must decide whether to reconfigure the FPGA by strobing the *n*CONFIG pin low or ignore the error.

CRC_ERROR Pin

A specific error detection pin, CRC_ERROR, is required to monitor the results of the error detection circuitry during user mode.

Table 11–2 lists the CRC_ERROR pin.

Table 11–2. CRC_ERROR Pin Description

Device	CRC_ERROR Pin Type	Description
Cyclone III	Dedicated Output or Open Drain Output (Optional)	By default, the Quartus II software sets the CRC_ERROR pin as a dedicated output. If the CRC_ERROR pin is used as a dedicated output, you must ensure that the V_{CCIO} of the bank in which the pin resides meets the input voltage specification of the system receiving the signal. Optionally, you can set this pin to be an open-drain output by enabling the option in the Quartus II software from the **Error Detection CRC** tab of the **Device and Pin Options** dialog box. Using the pin as an open-drain provides an advantage on the voltage leveling. To use this pin as open-drain, you can tie this pin to V_{CCIO} of Bank 1 through a 10-kΩ pull-resistor. Alternatively, depending on the voltage input specification of the system receiving the signal, you can tie the pull-up resistor to a different pull-up voltage.
Cyclone III LS	Open Drain Output	To use the CRC_ERROR pin, you can either tie this pin to V_{CCIO} through a 10-kΩ pull-up resistor, or depending on input voltage specification of the system receiving the signal, you can tie this pin to a different pull-up voltage.

For more information about the CRC_ERROR pin information for Cyclone III device family, refer to the Cyclone III Pin-Out Files for Altera Devices page on the Altera® website.

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

Error Detection Block

Table 11–3 lists the types of CRC detection to check the configuration bits.

Table 11–3. 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, and 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 11–1 shows the block diagram of the error detection block and the two related 32-bit registers: the signature register and the storage register.

Figure 11–1. Error Detection Block Diagram

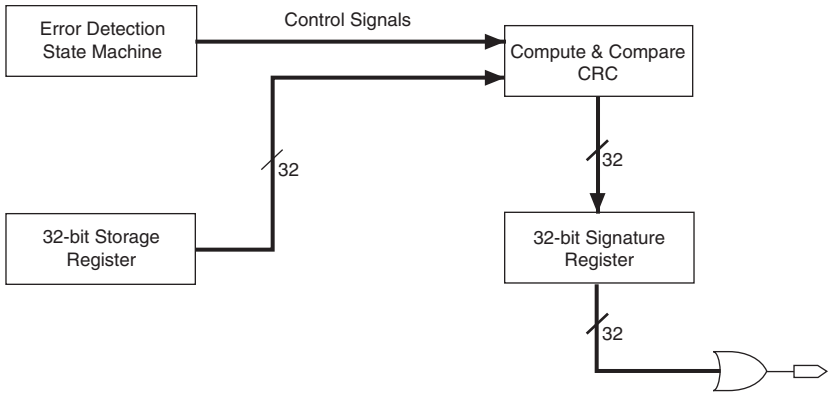


Table 11–4 lists the registers shown in Figure 11–1.

Table 11–4. 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 11–1) 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 has detected 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 11–5 lists the minimum and maximum error detection frequencies.

Table 11–5. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (2^v)
Cyclone III device family	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" on page 11–7). 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 11–1.

Equation 11–1. Error Detection Frequency

$$\text{Error detection frequency} = \frac{80\ \text{MHz}}{2^n}$$

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

Table 11–6 lists the estimated time for each CRC calculation with minimum and maximum clock frequencies for Cyclone III device family.

Table 11–6. CRC Calculation Time

Device		Minimum Time (ms) (1)	Maximum Time (s) (2)
Cyclone III	EP3C5	5	2.29
	EP3C10	5	2.29
	EP3C16	7	3.17
	EP3C25	9	4.51
	EP3C40	15	7.48
	EP3C55	23	11.77
	EP3C80	31	15.81
	EP3C120	45	22.67
Cyclone III LS	EP3CLS70	42	21.24
	EP3CLS100	42	21.24
	EP3CLS150	79	40.27
	EP3CLS200	79	40.27

Notes to Table 11–6:

(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.

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.

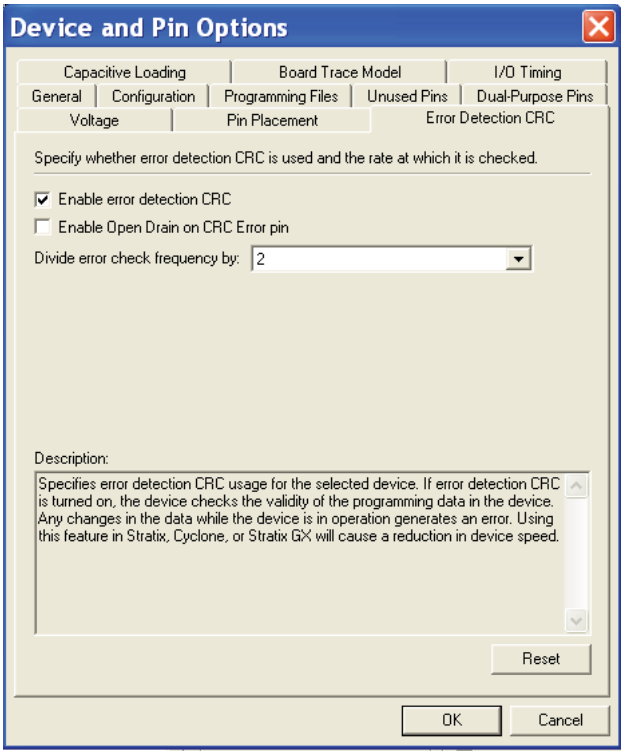
To enable the error detection feature using CRC, perform the following steps:

1. Open the Quartus II software and load a project using Cyclone III device family.
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**, as shown in Figure 11–2.
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 11–5 on page 11–6.

   The divisor value divides down 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 11–2. Enabling the Error Detection CRC Feature in the Quartus II Software



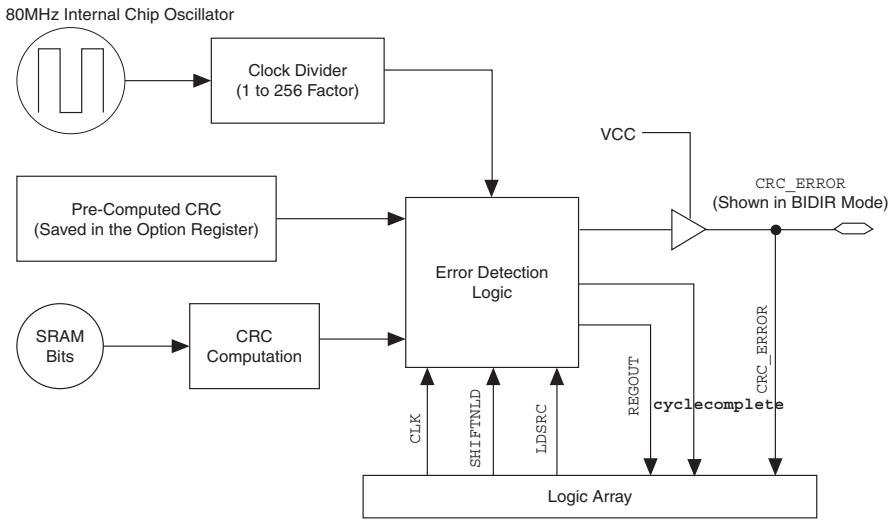
For Cyclone III LS devices, the "**Enable Open Drain on CRC Error Pin**" option is not available because the Quartus II software sets the CRC_ERROR pin for the Cyclone III LS device as open drain output by default.

Accessing Error Detection Block Through User Logic

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

Figure 11–3 shows the error detection block diagram in FPGA devices and shows the interface that the WYSIWYG atom enables in your design.

Figure 11–3. Error Detection Block Diagram



The user logic is affected by the soft error failure, thus 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 <device>_crcblock WYSIWYG atom, you must name the atom for each Cyclone III device family accordingly.

Table 11–7 lists the name of the WYSIWYG atom for Cyclone III device family.

Table 11–7. WYSIWYG Atoms

Device	WYSIWYG Atom
Cyclone III	`cycloneiii_crcblock`
Cyclone III LS	`cycloneiiils_crcblock`

To enable the `cycloneiii_crcblock` primitive in version 8.0 SP1 or earlier of the Quartus II software, turn on the error detection CRC feature in the **Device and Pins Options** dialog box. This is not required when you are using version 8.1 and later of the Quartus II software.

Example 11–1 shows an example of how to define the input and output ports of a WYSIWYG atom in a Cyclone III LS device.

Example 11–1. Error Detection Block Diagram

```
cycloneiiils_crcblock<crcblock_name>

(

.clk(<clock source>),

.shiftnld(<shiftnld source>),

.ldsrc(<ldsrc source>),

.crcerror(<crcerror out destination>),

.regout(<output destination>),

.cyclecomplete(<cyclecomplete destination>),

);
```

Table 11–8 lists the input and output ports that must be included in the atom. The input and output ports of the atoms for Cyclone III device family are similar, except for the `cyclecomplete` port which is for Cyclone III LS devices only.

Table 11–8. CRC Block Input and Output Ports (Part 1 of 2)

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, 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. This port is required.

Table 11–8. CRC Block Input and Output Ports (Part 2 of 2)

Port	Input/Output	Definition
.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. This signal must be connected 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, this signal must be connected 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 VCC (Figure 11–3 on page 11–8).
.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.
.cyclecomplete (<cyclone complete indicator output>)	Output	This signal is for `cycloneiiils_crcblock` only. This output signal is synchronized to the internal oscillator of the device (80-MHz internal oscillator), and not to the `clk` port. The signal asserts high for one clock cyclone every time an error detection cyclone completes.

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 *n*`CONFIG` 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 11–9 lists the revision history for this document.

Table 11–9. Document Revision History (Part 1 of 2)

Date	Version	Changes
December 2011	2.3	■ Updated "User Mode Error Detection" on page 11–2. ■ Update hyperlinks. ■ Minor text edits.
December 2009	2.2	Minor changes to the text.

Table 11–9. Document Revision History (Part 2 of 2)

Date	Version	Changes
July 2009	2.1	Made minor correction to the part number.
June 2009	2.0	■ Updated chapter part number. ■ Updated “Introduction” on page 11–1. ■ Updated Table 11–6 on page 11–6 and Table 11–8 on page 11–9. ■ Updated Figure 11–2 on page 11–7. ■ Updated “Accessing Error Detection Block Through User Logic” on page 11–8.
October 2008	1.3	■ Added chapter “Accessing Error Detection Block through User Logic” to document. ■ Updated chapter to new template.
May 2008	1.2	■ Updated Table 11-2.
July 2007	1.1	■ Added chapter TOC to document.
March 2007	1.0	Initial release.

12. IEEE 1149.1 (JTAG) Boundary-Scan Testing for the Cyclone III Device Family

CIII51014-2.3

This chapter provides guidelines on using the IEEE Std. 1149.1 boundary-scan test (BST) circuitry in Cyclone® III device family (Cyclone III and Cyclone III LS devices). BST architecture tests pin connections without using physical test probes, and captures functional data while a device is operating normally. Boundary-scan cells (BSCs) in a device can force signals onto pins or capture data from pin or logic array signals. Forced test data is serially shifted into the boundary-scan cells. Captured data is serially shifted out and externally compared to expected results.

This chapter contains the following sections:

- "IEEE Std. 1149.1 BST Architecture" on page 12–1
- "IEEE Std. 1149.1 BST Operation Control" on page 12–2
- "I/O Voltage Support in a JTAG Chain" on page 12–5
- "Guidelines for IEEE Std. 1149.1 BST" on page 12–6
- "Boundary-Scan Description Language Support" on page 12–7

IEEE Std. 1149.1 BST Architecture

Cyclone III device family operating in the IEEE Std. 1149.1 BST mode use four required pins:

- `TDI`
- `TDO`
- `TMS`
- `TCK`

The `TCK` pin has an internal weak pull-down resistor, while the `TDI` and `TMS` pins have weak internal pull-up resistors. The `TDO` output pin and all the JTAG input pins are powered by the V_{CCIO} supply of bank 1A. All user I/O pins are tri-stated during JTAG configuration.

For recommendations on how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to "I/O Voltage Support in a JTAG Chain" on page 12–5.

For more information about the description and functionality of all JTAG pins, registers used by the IEEE Std. 1149.1 BST circuitry, and the test access port (TAP) controller, refer to *AN39: IEEE 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*.

© 2011 Altera Corporation. All rights reserved. ALTERA, ARRIA, CYCLONE, HARDCOPY, MAX, MEGACORE, NIOS, QUARTUS and STRATIX words and logos are trademarks of Altera Corporation and registered in the U.S. Patent and Trademark Office and in other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders as described at www.altera.com/common/legal.html. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

ISO 9001:2008 Registered

IEEE Std. 1149.1 BST Operation Control

Table 12–1 lists the boundary-scan register length for devices in Cyclone III device family.

Table 12–1. Boundary-Scan Register Length for Cyclone III Device Family

Family	Device	Boundary-Scan Register Length
Cyclone III	EP3C5	603
	EP3C10	603
	EP3C16	1,080
	EP3C25	732
	EP3C40	1,632
	EP3C55	1,164
	EP3C80	1,314
	EP3C120	1,620
Cyclone III LS	EP3CLS70	1,314
	EP3CLS100	1,314
	EP3CLS150	1,314
	EP3CLS200	1,314

Table 12–2 lists the `IDCODE` information for devices in Cyclone III device family.

Table 12–2. Device IDCODE for Cyclone III Device Family

Family	Device	IDCODE (32 Bits) (1)			
		Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) (2)
Cyclone III	EP3C5	0000	0010 0000 1111 0001	000 0110 1110	1
	EP3C10	0000	0010 0000 1111 0001	000 0110 1110	1
	EP3C16	0000	0010 0000 1111 0010	000 0110 1110	1
	EP3C25	0000	0010 0000 1111 0011	000 0110 1110	1
	EP3C40	0000	0010 0000 1111 0100	000 0110 1110	1
	EP3C55	0000	0010 0000 1111 0101	000 0110 1110	1
	EP3C80	0000	0010 0000 1111 0110	000 0110 1110	1
	EP3C120	0000	0010 0000 1111 0111	000 0110 1110	1
Cyclone III LS	EP3CLS70	0000	0010 0111 0000 0001	000 0110 1110	1
	EP3CLS100	0000	0010 0111 0000 0000	000 0110 1110	1
	EP3CLS150	0000	0010 0111 0000 0011	000 0110 1110	1
	EP3CLS200	0000	0010 0111 0000 0010	000 0110 1110	1

Notes to Table 12–2:

(1) The MSB is on the left.

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

Cyclone III device family supports the IEEE Std. 1149.1 (JTAG) instructions as listed in Table 12–3.

Table 12–3. IEEE Std. 1149.1 (JTAG) Instructions Supported by Cyclone III Device Family (Part 1 of 2)

JTAG Instruction	Instruction Code	Description
SAMPLE/PRELOAD	00 0000 0101	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation, and permits an initial data pattern to be output at the device pins. Also used by the SignalTap® II embedded logic analyzer.
EXTEST (1)	00 0000 1111	Allows the external circuitry and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	11 1111 1111	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.
USERCODE	00 0000 0111	Selects the 32-bit USERCODE register and places it between the TDI and TDO pins, allowing the USERCODE to be serially shifted out of TDO.
IDCODE	00 000 0110	Selects the IDCODE register and places it between TDI and TDO, allowing the IDCODE to be serially shifted out of TDO. IDCODE is the default instruction at power up and in TAP RESET state.
HIGHZ	00 0000 1011	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation, while tri-stating all of the I/O pins.
CLAMP	00 0000 1010	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding I/O pins to a state defined by the data in the boundary scan register.
ICR Instructions	—	Used when configuring Cyclone III device family using the JTAG port with a USB-Blaster™ ByteBlaster™ II, MasterBlaster™ or ByteBlasterMV™ download cable, or when using a Jam File, or JBC File via an embedded processor.
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is unaffected.
CONFIG_IO (2)	00 0000 1101	Allows I/O reconfiguration through JTAG ports using the IOCSR for JTAG testing. This is executed after or during configurations. nSTATUS pin must go high before you can issue the CONFIG_IO instruction.
EN_ACTIVE_CLK (2)	01 1110 1110	Allows CLKUSR pin signal to replace the internal oscillator as the configuration clock source.
DIS_ACTIVE_CLK (2)	10 1110 1110	Allows you to revert the configuration clock source from CLKUSR pin signal set by EN_ACTIVE_CLK back to the internal oscillator.
ACTIVE_DISENGAGE (2)	10 1101 0000	Places the active configuration mode controllers into idle state prior to CONFIG_IO to configure the IOCSR or perform board level testing.
ACTIVE_ENGAGE (2)	10 1011 0000	This instruction might be used in AS and AP configuration schemes to re-engage the active controller.
APFC_BOOT_ADDR (2), (3)	10 0111 0000	Places the 22-bit active boot address register between the TDI and TDO pins, allowing a new active boot address to be serially shifted into TDI and into the active parallel (AP) flash controller. In remote system upgrade, the PFC_BOOT_ADDR instruction sets the boot address for the factory configuration.

Table 12–3. IEEE Std. 1149.1 (JTAG) Instructions Supported by Cyclone III Device Family (Part 2 of 2)

JTAG Instruction	Instruction Code	Description
FACTORY (4)	10 1000 0001	Enables access to all other JTAG instructions (other than BYPASS, SAMPLE/PRELOAD and EXTEST instructions, which are supported upon power up). This instruction also clears the device configuration data and advanced encryption standard (AES) volatile key.
KEY_PROG_VOL (4)	01 1010 1101	Used to enter and store the security key into volatile registers. When this instruction is executed, TDI is connected to a 512-bit volatile key scan chain. TDO is not connected to the end of this scan chain.
KEY_CLR_VREG (4)	00 0010 1001	Clears the volatile verify register which signifies the validity of the volatile keys stored in the registers. You must clear the volatile verify register by issuing this command whenever you attempt to program a new volatile key. This instruction must be asserted for at least 10 TCK cycles.

Notes to Table 12–3:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

(2) For more information about how to use CONFIG_IO, EN_ACTIVE_CLK, DIS_ACTIVE_CLK, ACTIVE_DISENGAGE, ACTIVE_ENGAGE and APFC_BOOT_ADDR instructions for Cyclone III device family, refer to the *Configuration, Design Security, and Remote System Upgrades in Cyclone III Devices* chapter.

(3) APFC_BOOT_ADDR instruction is not supported in Cyclone III LS devices.

(4) For Cyclone III LS devices only. For more information about how to program the security key into the volatile registers, refer to the *Configuration, Design Security, and Remote System Upgrades in Cyclone III Devices* chapter.

The IEEE Std. 1149.1 BST circuitry is enabled upon device power-up. You can perform BST on Cyclone III device family before, after, and during configuration. Cyclone III device family supports the BYPASS, IDCODE and SAMPLE instructions during configuration without interrupting configuration. To send all other JTAG instructions, interrupt the configuration using the CONFIG_IO instruction except for active configuration schemes in which the ACTIVE_DISENGAGE instruction is used instead.

The CONFIG_IO instruction allows you to configure I/O buffers via the JTAG port, and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring Cyclone III device family. Alternatively, you can wait for the configuration device to complete configuration. After configuration is interrupted and JTAG BST is complete, you must reconfigure the part via JTAG (PULSE_NCONFIG instruction) or by pulsing nCONFIG low.

When you perform JTAG boundary-scan testing before configuration, the nCONFIG pin must be held low.

For more information about the following topics, refer to *AN39: IEEE 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*:

- TAP controller state-machine
- Timing requirements for IEEE Std. 1149.1 signals
- Instruction mode
- Mandatory JTAG instructions (SAMPLE/PRELOAD, EXTEST and BYPASS)
- Optional JTAG instructions (IDCODE, USERCODE, CLAMP and HIGHZ)

The following information is only applicable to Cyclone III LS devices:

- Only the three mandatory JTAG 1149.1 JTAG instructions (`BYPASS`, `SAMPLE/PRELOAD`, `EXTEST`) and the `FACTORY` private instruction are supported from the JTAG pins upon power up. The `FACTORY` instruction (instruction code: `10 1000 0001`) must be issued before the device starts loading the core configuration data to enable access to all other JTAG instructions. This instruction also clears the device configuration data and AES volatile key.
- `IDCODE` instruction is not supported upon power-up, prior to issuing the `FACTORY` instruction. However, it is the default instruction when the TAP controller is in the reset state. Without loading any instructions, you can go to the `Shift_DR` state and shift out the JTAG Device ID.
- `IDCODE`, `CONFIG_IO`, `ACTIVE_DISENGAGE`, `HIGHZ`, `CLAMP`, `USERCODE` and `PULSE_NCONFIG` instructions are supported, provided that the `FACTORY` instruction is executed.

I/O Voltage Support in a JTAG Chain

A JTAG chain can contain several different devices. However, you must be cautious if the chain contains devices that have different V_{CCIO} levels. The output voltage level of the `TDO` pin must meet the specifications of the `TDI` pin it drives. For Cyclone III device family, the `TDO` pin is powered by the V_{CCIO} power supply. Because the V_{CCIO} supply is 3.3 V, the `TDO` pin drives out 3.3 V.

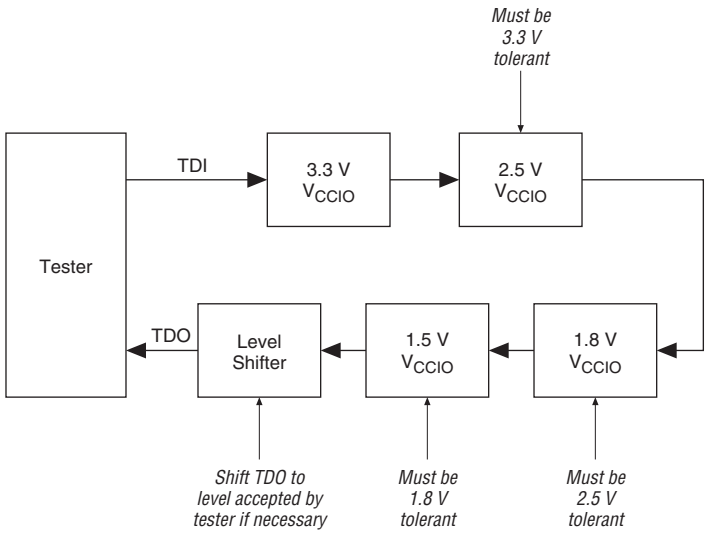
Devices can interface with each other although they might have different V_{CCIO} levels. For example, a device with a 3.3-V `TDO` pin can drive to a device with a 5.0-V `TDI` pin because 3.3 V meets the minimum TTL-level V_{IH} for the 5.0-V `TDI` pin. JTAG pins on Cyclone III device family can support the input levels of V_{CCIO} of bank 1A.

For multiple devices in a JTAG chain with 3.0-V or 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 must be built in such a way that a device with a higher V_{CCIO} level drives to 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 12–1 shows the JTAG chain of mixed voltages and how a level shifter is inserted in the chain.

Figure 12–1. JTAG Chain of Mixed Voltages



Guidelines for IEEE Std. 1149.1 BST

Use the following guidelines when performing BST with IEEE Std. 1149.1 devices:

- If the 10 bit checkerboard pattern (1010101010) does not shift out of the instruction register via the `TDO` pin during the first clock cycle of the `SHIFT_IR` state, the TAP controller did not reach the proper state. To solve this problem, try one of the following procedures:
  - Verify that the TAP controller has reached the `SHIFT_IR` state correctly. To advance the TAP controller to the `SHIFT_IR` state, return to the `RESET` state and send the code `01100` to the `TMS` pin.
  - Check the connections to the `VCC`, `GND`, `JTAG`, and dedicated configuration pins on the device.
- Perform a `SAMPLE/PRELOAD` test cycle prior to the first `EXTEST` test cycle to ensure that known data is present at the device pins when you enter the `EXTEST` mode. If the `OEJ` update register contains a 0, the data in the `OUTJ` update register is driven out. The state must be known and correct to avoid contention with other devices in the system.
- Do not perform `EXTEST` testing during ICR. This instruction is supported before or after ICR, but not during ICR. Use the `CONFIG_IO` instruction to interrupt configuration and then perform testing, or wait for configuration to complete.
- If testing is performed before configuration, hold the `nCONFIG` pin low.

CAUTION You must not invoke the following private instructions at any instance because these instructions can potentially damage the device, rendering the device useless:

- 1000010000
- 1001000000
- 1011100000

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 BST-capable device that can be tested. To download BSDL files for IEEE Std. 1149.1-compliant Cyclone III device family, visit the Altera Download Center.

BSDL files for IEEE std. 1149.1-compliant Cyclone III LS devices can also be generated using version 9.0 and later of the Quartus II software.

To perform BST on a configured device, a post configuration BSDL file that is customized to your design is required. Post configuration BSDL file generation with BSDL Customizer script (available on the Altera Download Center) is for Cyclone III devices only.

Use version 9.0 and later of the Quartus II software to create a post configuration BSDL file for Cyclone III LS devices.

For information on the procedures to generate the generic and post configuration BSDL files with Quartus II software, visit the Altera Download Center.

Document Revision History

Table 12–4 lists the revision history for this document.

Table 12–4. Document Revision History (Part 1 of 2)

Date	Version	Changes
December 2011	2.3	■ Updated "IEEE Std. 1149.1 BST Architecture" on page 12–1 and "I/O Voltage Support in a JTAG Chain" on page 12–5. ■ Minor text edits.
December 2009	2.2	Minor changes to the text.
July 2009	2.1	Made minor correction to the part number.
June 2009	2.0	■ Updated "Introduction" on page 12–1, "IEEE Std. 1149.1 BST Architecture" on page 12–1, "IEEE Std. 1149.1 BST Operation Control" on page 12–2, "Guidelines for IEEE Std. 1149.1 BST" on page 12–6, and "Boundary-Scan Description Language Support" on page 12–7. ■ Updated Table 12–1 on page 12–2, Table 12–2 on page 12–2, and Table 12–3 on page 12–3.
October 2008	1.3	Updated chapter to new template.
May 2008	1.2	Minor textual changes.

Table 12–4. Document Revision History (Part 2 of 2)

Date	Version	Changes
July 2007	1.1	■ Updated “IEEE Std.1149.1 Boundary-Scan Register” section. ■ Updated IDCODE information and removed SignalTap II instructions in Table 12-4. ■ Updated “BST for Configured Devices” section. ■ Added a guideline to “Guidelines for IEEE Std. 1149.1 Boundary-Scan Testing” section. ■ Added chapter TOC and “Referenced Documents” section.
March 2007	1.0	Initial release.

This chapter provides additional information about the document and Altera.

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.