

Cyclone Device Handbook, Volume 1



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

C5V1-2.4

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. 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 Corporation. 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.

I.S. EN ISO 9001

Contents

Chapter Revision Dates xi

About this Handbook xiii
How to Find Information xiii
How to Contact Altera xiii
Typographic Conventions xiv

Section I. Cyclone FPGA Family Data Sheet
Revision History 2–1

Chapter 1. Introduction
Introduction 1–1
Features 1–1
Document Revision History 1–3

Chapter 2. Cyclone Architecture
Functional Description 2–1
Logic Array Blocks 2–3
LAB Interconnects 2–3
LAB Control Signals 2–4
Logic Elements 2–5
LUT Chain and Register Chain 2–7
addnsub Signal 2–7
LE Operating Modes 2–7
MultiTrack Interconnect 2–12
Embedded Memory 2–18
Memory Modes 2–18
Parity Bit Support 2–20
Shift Register Support 2–20
Memory Configuration Sizes 2–21
Byte Enables 2–23
Control Signals and M4K Interface 2–23
Independent Clock Mode 2–25
Input/Output Clock Mode 2–25
Read/Write Clock Mode 2–28
Single-Port Mode 2–29
Global Clock Network and Phase-Locked Loops 2–29
Global Clock Network 2–29

Dual-Purpose Clock Pins 2–31
Combined Resources 2–31
PLLs 2–32
Clock Multiplication and Division 2–35
External Clock Inputs 2–36
External Clock Outputs 2–36
Clock Feedback 2–37
Phase Shifting 2–37
Lock Detect Signal 2–37
Programmable Duty Cycle 2–38
Control Signals 2–38
I/O Structure 2–39
External RAM Interfacing 2–46
DDR SDRAM and FCRAM 2–46
Programmable Drive Strength 2–49
Open-Drain Output 2–50
Slew-Rate Control 2–51
Bus Hold 2–51
Programmable Pull-Up Resistor 2–51
Advanced I/O Standard Support 2–52
LVDS I/O Pins 2–54
MultiVolt I/O Interface 2–54
Power Sequencing and Hot Socketing 2–55
Referenced Documents 2–56
Document Revision History 2–56

Chapter 3. Configuration and Testing

IEEE Std. 1149.1 (JTAG) Boundary Scan Support 3–1
SignalTap II Embedded Logic Analyzer 3–5
Configuration 3–5
Operating Modes 3–6
Configuration Schemes 3–6
Referenced Documents 3–7
Document Revision History 3–7

Chapter 4. DC and Switching Characteristics

Operating Conditions 4–1
Power Consumption 4–8
Timing Model 4–9
Preliminary and Final Timing 4–9
Performance 4–10
Internal Timing Parameters 4–11
External Timing Parameters 4–15
External I/O Delay Parameters 4–21
Maximum Input and Output Clock Rates 4–27
PLL Timing 4–29
Referenced Document 4–31

Document Revision History 4–31

Chapter 5. Reference and Ordering Information

Software 5–1
Device Pin-Outs 5–1
Ordering Information 5–1
Referenced Documents 5–2
Document Revision History 5–2

Section II. Clock Management

Revision History 5–1

Chapter 6. Using PLLs in Cyclone Devices

Introduction 6–1
Hardware Overview 6–1
Software Overview 6–4
Pins and Clock Network Connections 6–6
Hardware Features 6–8
Clock Multiplication and Division 6–8
Phase Shifting 6–9
Programmable Duty Cycle 6–10
External Clock Output 6–11
Control Signals 6–12
Clock Feedback Modes 6–13
Normal Mode 6–14
Zero Delay Buffer Mode 6–15
No Compensation 6–15
Pins 6–16
Board Layout 6–17
VCCA and GNDA 6–17
Jitter Considerations 6–19
Specifications 6–20
Software Support 6–20
Quartus II altpll Megafunction 6–20
altpll Input Ports 6–22
altpll Output Ports 6–23
MegaWizard Customization 6–23
MegaWizard Page Description 6–25
Compilation Report 6–31
Timing Analysis 6–33
Simulation 6–37
Global Clock Network 6–38
Dedicated Clock Input Pins 6–40
Dual-Purpose Clock I/O Pins 6–40
Combined Sources 6–41

Conclusion 6–43
Referenced Documents 6–44
Document Revision History 6–44

Section III. Memory

Revision History 7–1

Chapter 7. On-Chip Memory Implementations Using Cyclone Memory Blocks

Introduction 7–1
M4K Memory Features 7–1
Parity Bit Support 7–2
Byte-Enable Support 7–3
Power-up Conditions and Memory Initialization 7–4
Using M4K Memory 7–4
Implementing Single-Port Mode 7–5
Implementing Simple Dual-Port Mode 7–6
Implementing True Dual-Port Mode 7–8
Implementing Shift-Register Mode 7–11
Implementing ROM Mode 7–12
Implementing FIFO Buffers 7–12
Clock Modes 7–13
Independent Clock Mode 7–13
Input/Output Clock Mode 7–15
Read/Write Clock Mode 7–17
Single-Port Mode 7–18
Synchronous and Pseudo-Asynchronous Modes 7–19
Read-during-Write Operation at the Same Address 7–20
Same-Port Read-during-Write Mode 7–20
Mixed-Port Read-during-Write Mode 7–21
Conclusion 7–22
Referenced Documents 7–23
Document Revision History 7–23

Section IV. I/O Standards

Revision History 8–1

Chapter 8. Using Selectable I/O Standards in Cyclone Devices

Introduction 8–1
Supported I/O Standards 8–2
3.3-V LVTTL (EIA/JEDEC Standard JESD8-B) 8–3
3.3-V LVCMOS (EIA/JEDEC Standard JESD8-B) 8–3
2.5-V LVTTL Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-5) 8–3
2.5-V LVCMOS Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-5) . 8–4

1.8-V LVTTL Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-7) 8–4
1.8-V LVCMOS Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-7) . 8–4
1.5-V LVCMOS Normal and Wide Voltage Ranges (EIA/JEDEC Standard JESD8-11) 8–5
3.3-V (PCI Special Interest Group (SIG) PCI Local Bus Specification Revision 2.2) 8–5
SSTL-3 Class I and II (EIA/JEDEC Standard JESD8-8) 8–7
SSTL-2 Class I and II (EIA/JEDEC Standard JESD8-9A) 8–7
LVDS (ANSI/TIA/EIA Standard ANSI/TIA/EIA-644) 8–8
Differential SSTL-2 - EIA/JEDEC Standard JESD8-9A 8–9
Cyclone I/O Banks 8–9
Programmable Current Drive Strength 8–12
Hot Socketing 8–13
I/O Termination 8–13
Voltage-Referenced I/O Standard Termination 8–14
Differential I/O Standard Termination 8–14
Pad Placement and DC Guidelines 8–14
Differential Pad Placement Guidelines 8–14
V_{REF} Pad Placement Guidelines 8–15
DC Guidelines 8–18
Quartus II Software Support 8–18
Settings 8–18
Conclusion 8–22
More Information 8–22
References 8–22
Referenced Documents 8–23
Document Revision History 8–23

Chapter 9. High-Speed Differential Signaling in Cyclone Devices

Introduction 9–1
Cyclone High-Speed I/O Banks 9–2
Cyclone High-Speed I/O Interface 9–3
Clock Domains 9–3
LVDS Receiver and Transmitter 9–4
RSDS I/O Standard Support in Cyclone Devices 9–7
Designing with RSDS 9–9
RSDS Software Support 9–10
High-Speed I/O Timing in Cyclone Devices 9–11
LVDS Receiver and Transmitter Termination 9–15
Implementing Cyclone LVDS and RSDS I/O Pins in the Quartus II Software 9–16
Design Guidelines 9–18
Differential Pad Placement Guidelines 9–18
Board Design Considerations 9–18
Conclusion 9–19
Referenced Documents 9–19
Document Revision History 9–20

Section V. Design Considerations

Revision History 10–1

Chapter 10. Implementing Double Data Rate I/O Signaling in Cyclone Devices

Introduction 10–1
Double Data Rate Input 10–1
Double Data Rate Output 10–2
Bidirectional Double Data Rate 10–3
DDR Memory Support 10–4
Conclusion 10–4
Referenced Documents 10–4
Document Revision History 10–5

Chapter 11. Using Cyclone Devices in Multiple-Voltage Systems

Introduction 11–1
I/O Standards 11–1
MultiVolt I/O Operation 11–2
5.0-V Device Compatibility 11–3
Hot-Socketing 11–6
Devices Can Be Driven before Power-Up 11–6
I/O Pins Remain Tri-Stated during Power-Up 11–6
Signal Pins Do Not Drive the V_{CCIO} or V_{CCINT} Power Supplies 11–6
Power-Up Sequence 11–7
Power-On Reset 11–7
Conclusion 11–8
Document Revision History 11–8

Chapter 12. Designing with 1.5-V Devices

Introduction 12–1
Power Sequencing and Hot Socketing 12–1
Using MultiVolt I/O Pins 12–2
Voltage Regulators 12–3
Linear Voltage Regulators 12–4
Switching Voltage Regulators 12–6
Maximum Output Current 12–8
Selecting Voltage Regulators 12–8
Voltage Divider Network 12–10
1.5-V Regulator Circuits 12–10
1.5-V Regulator Application Examples 12–19
Synchronous Switching Regulator Example 12–20
Board Layout 12–21
Split-Plane Method 12–23
Conclusion 12–24
References 12–24
Referenced Documents 12–25
Document Revision History 12–25

Section VI. Configuration

Revision History 13–1

Chapter 13. Configuring Cyclone FPGAs

Introduction 13–1
Device Configuration Overview 13–1
Data Compression 13–3
Configuration Schemes 13–8
Active Serial Configuration (Serial Configuration Devices) 13–8
Passive Serial Configuration 13–18
JTAG-Based Configuration 13–31
Combining Configuration Schemes 13–45
Active Serial and JTAG 13–45
Device Configuration Pins 13–46
Referenced Documents 13–50
Document Revision History 13–51

Chapter 14. Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet

Introduction 14–1
Functional Description 14–2
Accessing Memory in Serial Configuration Devices 14–7
Active Serial FPGA Configuration 14–8
Serial Configuration Device Memory Access 14–12
Memory Array Organization 14–12
Operation Codes 14–20
Power and Operation 14–33
Power Mode 14–33
Power-On Reset 14–34
Error Detection 14–34
Timing Information 14–35
Programming and Configuration File Support 14–38
Operating Conditions 14–39
Pin Information 14–41
Package 14–43
Ordering Code 14–44
Referenced Documents 14–44
Document Revision History 14–44

Section VII. Cyclone Device Package Information

Revision History 15–1

Chapter 15. Package Information for Cyclone Devices

Introduction 15–1

Device and Package Cross Reference 15–1
Thermal Resistance 15–2
Package Outlines 15–2
Document Revision History 15–3

Chapter Revision Dates

The chapters in this book, *Cyclone Device Handbook, Volume 1*, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Introduction
Revised: *May 2008*
Part number: *C51001-1.5*

Chapter 2. Cyclone Architecture
Revised: *May 2008*
Part number: *C51002-1.6*

Chapter 3. Configuration and Testing
Revised: *May 2008*
Part number: *C51003-1.4*

Chapter 4. DC and Switching Characteristics
Revised: *May 2008*
Part number: *C51004-1.7*

Chapter 5. Reference and Ordering Information
Revised: *May 2008*
Part number: *C51005-1.4*

Chapter 6. Using PLLs in Cyclone Devices
Revised: *May 2008*
Part number: *C51006-1.5*

Chapter 7. On-Chip Memory Implementations Using Cyclone Memory Blocks
Revised: *May 2008*
Part number: *C51007-1.4*

Chapter 8. Using Selectable I/O Standards in Cyclone Devices
Revised: *May 2008*
Part number: *C51008-1.6*

Chapter 9. High-Speed Differential Signaling in Cyclone Devices
Revised: *May 2008*
Part number: *C51009-1.6*

Chapter 10. Implementing Double Data Rate I/O Signaling in Cyclone Devices
Revised: *May 2008*
Part number: *C51010-1.2*

Chapter 11. Using Cyclone Devices in Multiple-Voltage Systems
Revised: *May 2008*
Part number: *C51011-1.2*

Chapter 12. Designing with 1.5-V Devices
Revised: *May 2008*
Part number: *C51012-1.4*

Chapter 13. Configuring Cyclone FPGAs
Revised: *May 2008*
Part number: *C51013-1.8*

Chapter 14. Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet
Revised: *May 2008*
Part number: *C51014-3.1*

Chapter 15. Package Information for Cyclone Devices
Revised: *May 2008*
Part number: *C52006-1.3*

About this Handbook

This handbook provides comprehensive information about the Altera® Cyclone® family of devices.

How to Find Information

You can find more information in the following ways:

- The Adobe Acrobat Find feature, which searches the text of a PDF document. Click the binoculars toolbar icon to open the Find dialog box.
- Acrobat bookmarks, which serve as an additional table of contents in PDF documents.
- Thumbnail icons, which provide miniature previews of each page, provide a link to the pages.
- Numerous links, shown in green text, which allow you to jump to related information.

How to Contact Altera

For 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
Altera literature services	Email	literature@altera.com
Non-technical 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

This document uses the typographic conventions shown below.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: *<file name>*, *<project name>***.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input`. Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
	The hand points to information that requires special attention.
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

Section I. Cyclone FPGA Family Data Sheet

This section provides designers with the data sheet specifications for Cyclone® devices. The chapters contain feature definitions of the internal architecture, configuration and JTAG boundary-scan testing information, DC operating conditions, AC timing parameters, a reference to power consumption, and ordering information for Cyclone devices.

This section contains the following chapters:

- Chapter 1. Introduction
- Chapter 2. Cyclone Architecture
- Chapter 3. Configuration and Testing
- Chapter 4. DC and Switching Characteristics
- Chapter 5. Reference and Ordering Information

Revision History

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

1. Introduction

C51001-1.5

Introduction

The Cyclone® field programmable gate array family is based on a 1.5-V, 0.13-µm, all-layer copper SRAM process, with densities up to 20,060 logic elements (LEs) and up to 288 Kbits of RAM. With features like phase-locked loops (PLLs) for clocking and a dedicated double data rate (DDR) interface to meet DDR SDRAM and fast cycle RAM (FCRAM) memory requirements, Cyclone devices are a cost-effective solution for data-path applications. Cyclone devices support various I/O standards, including LVDS at data rates up to 640 megabits per second (Mbps), and 66- and 33-MHz, 64- and 32-bit peripheral component interconnect (PCI), for interfacing with and supporting ASSP and ASIC devices. Altera also offers new low-cost serial configuration devices to configure Cyclone devices.

Features

The Cyclone device family offers the following features:

- 2,910 to 20,060 LEs, see Table 1–1
- Up to 294,912 RAM bits (36,864 bytes)
- Supports configuration through low-cost serial configuration device
- Support for LVTTL, LVCMOS, SSTL-2, and SSTL-3 I/O standards
- Support for 66- and 33-MHz, 64- and 32-bit PCI standard
- High-speed (640 Mbps) LVDS I/O support
- Low-speed (311 Mbps) LVDS I/O support
- 311-Mbps RSDS I/O support
- Up to two PLLs per device provide clock multiplication and phase shifting
- Up to eight global clock lines with six clock resources available per logic array block (LAB) row
- Support for external memory, including DDR SDRAM (133 MHz), FCRAM, and single data rate (SDR) SDRAM
- Support for multiple intellectual property (IP) cores, including Altera® MegaCore® functions and Altera Megafunctions Partners Program (AMPP℠) megafunctions.

Table 1–1. Cyclone Device Features (Part 1 of 2)

Feature	EP1C3	EP1C4	EP1C6	EP1C12	EP1C20
LEs	2,910	4,000	5,980	12,060	20,060
M4K RAM blocks (128 × 36 bits)	13	17	20	52	64

Table 1–1. Cyclone Device Features (Part 2 of 2)

Feature	EP1C3	EP1C4	EP1C6	EP1C12	EP1C20
Total RAM bits	59,904	78,336	92,160	239,616	294,912
PLLs	1	2	2	2	2
Maximum user I/O pins *(1)*	104	301	185	249	301

Note to Table 1–1:
(1) This parameter includes global clock pins.

Cyclone devices are available in quad flat pack (QFP) and space-saving FineLine® BGA packages (see Tables 1–2 through 1–3).

Table 1–2. Cyclone Package Options and I/O Pin Counts

Device	100-Pin TQFP *(1)*	144-Pin TQFP *(1), (2)*	240-Pin PQFP *(1)*	256-Pin FineLine BGA	324-Pin FineLine BGA	400-Pin FineLine BGA
EP1C3	65	104	—	—	—	—
EP1C4	—	—	—	—	249	301
EP1C6	—	98	185	185	—	—
EP1C12	—	—	173	185	249	—
EP1C20	—	—	—	—	233	301

Notes to Table 1–2:
(1) TQFP: thin quad flat pack.
PQFP: plastic quad flat pack.
(2) Cyclone devices support vertical migration within the same package (i.e., designers can migrate between the EP1C3 device in the 144-pin TQFP package and the EP1C6 device in the same package).

Vertical migration means you can migrate a design from one device to another that has the same dedicated pins, JTAG pins, and power pins, and are subsets or supersets for a given package across device densities. The largest density in any package has the highest number of power pins; you must use the layout for the largest planned density in a package to provide the necessary power pins for migration.

For I/O pin migration across densities, cross-reference the available I/O pins using the device pin-outs for all planned densities of a given package type to identify which I/O pins can be migrated. The Quartus® II software can automatically cross-reference and place all pins for you when given a device migration list. If one device has power or ground pins, but these same pins are user I/O on a different device that is in the migration path,the Quartus II software ensures the pins are not used as user I/O in the Quartus II software. Ensure that these pins are connected

to the appropriate plane on the board. The Quartus II software reserves I/O pins as power pins as necessary for layout with the larger densities in the same package having more power pins.

Table 1–3. Cyclone QFP and FineLine BGA Package Sizes

Dimension	100-Pin TQFP	144-Pin TQFP	240-Pin PQFP	256-Pin FineLine BGA	324-Pin FineLine BGA	400-Pin FineLine BGA
Pitch (mm)	0.5	0.5	0.5	1.0	1.0	1.0
Area (mm^2)	256	484	1,024	289	361	441
Length × width (mm × mm)	16×16	22×22	34.6×34.6	17×17	19×19	21×21

Document Revision History

Table 1–4 shows the revision history for this document.

Table 1–4. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.5	Minor textual and style changes.	—
January 2007 v1.4	Added document revision history.	—
August 2005 v1.3	Minor updates.	—
October 2003 v1.2	Added 64-bit PCI support information.	—
September 2003 v1.1	• Updated LVDS data rates to 640 Mbps from 311 Mbps. • Updated RSDS feature information.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

2. Cyclone Architecture

C51002-1.6

Functional Description

Cyclone® devices contain a two-dimensional row- and column-based architecture to implement custom logic. Column and row interconnects of varying speeds provide signal interconnects between LABs and embedded memory blocks.

The logic array consists of LABs, with 10 LEs in each LAB. An LE is a small unit of logic providing efficient implementation of user logic functions. LABs are grouped into rows and columns across the device. Cyclone devices range between 2,910 to 20,060 LEs.

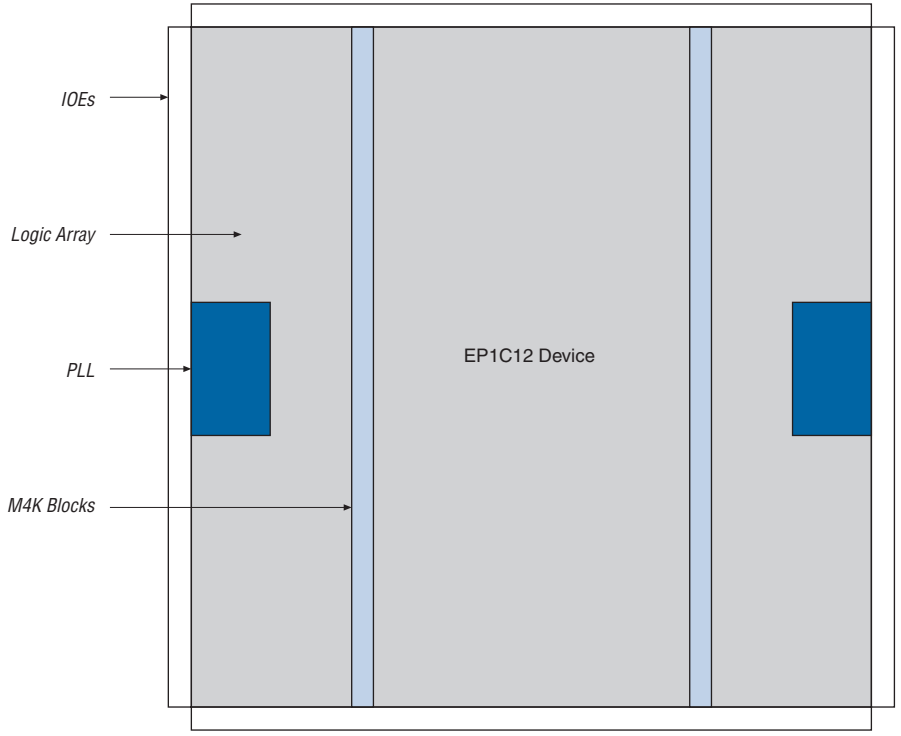
M4K RAM blocks are true dual-port memory blocks with 4K bits of memory plus parity (4,608 bits). These blocks provide dedicated true dual-port, simple dual-port, or single-port memory up to 36-bits wide at up to 250 MHz. These blocks are grouped into columns across the device in between certain LABs. Cyclone devices offer between 60 to 288 Kbits of embedded RAM.

Each Cyclone device I/O pin is fed by an I/O element (IOE) located at the ends of LAB rows and columns around the periphery of the device. I/O pins support various single-ended and differential I/O standards, such as the 66- and 33-MHz, 64- and 32-bit PCI standard and the LVDS I/O standard at up to 640 Mbps. Each IOE contains a bidirectional I/O buffer and three registers for registering input, output, and output-enable signals. Dual-purpose DQS, DQ, and DM pins along with delay chains (used to phase-align DDR signals) provide interface support with external memory devices such as DDR SDRAM, and FCRAM devices at up to 133 MHz (266 Mbps).

Cyclone devices provide a global clock network and up to two PLLs. The global clock network consists of eight global clock lines that drive throughout the entire device. The global clock network can provide clocks for all resources within the device, such as IOEs, LEs, and memory blocks. The global clock lines can also be used for control signals. Cyclone PLLs provide general-purpose clocking with clock multiplication and phase shifting as well as external outputs for high-speed differential I/O support.

Figure 2–1 shows a diagram of the Cyclone EP1C12 device.

Figure 2–1. Cyclone EP1C12 Device Block Diagram



The number of M4K RAM blocks, PLLs, rows, and columns vary per device. Table 2–1 lists the resources available in each Cyclone device.

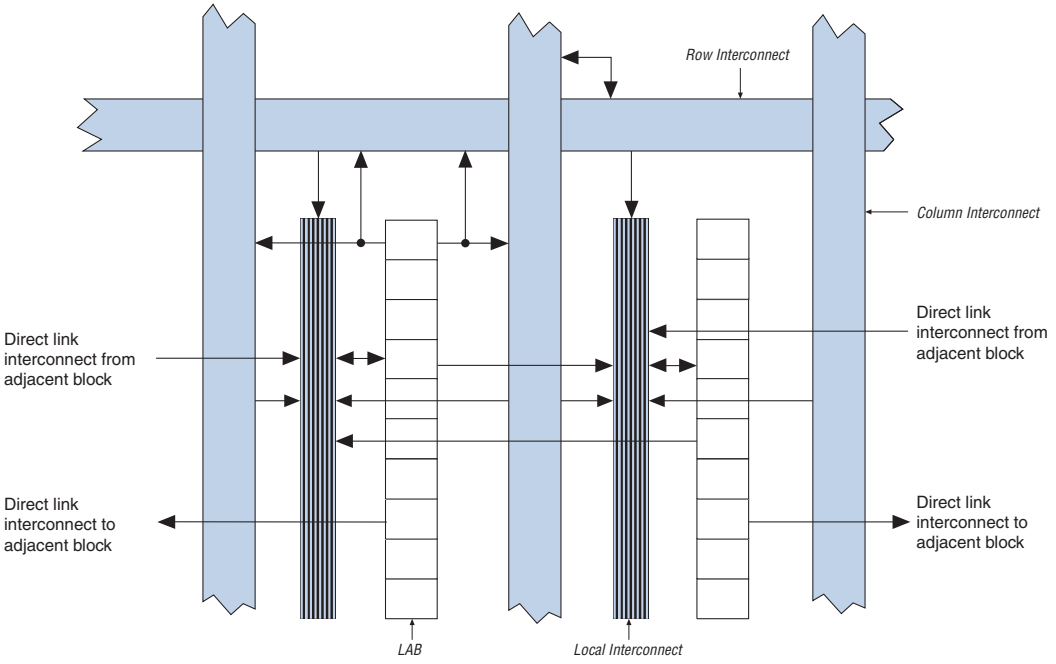
Table 2–1. Cyclone Device Resources

Device	M4K RAM		PLLs	LAB Columns	LAB Rows
	Columns	Blocks			
EP1C3	1	13	1	24	13
EP1C4	1	17	2	26	17
EP1C6	1	20	2	32	20
EP1C12	2	52	2	48	26
EP1C20	2	64	2	64	32

Logic Array Blocks

Each LAB consists of 10 LEs, LE carry chains, LAB control signals, a local interconnect, look-up table (LUT) chain, and register chain connection lines. The local interconnect transfers signals between LEs in the same LAB. LUT chain connections transfer the output of one LE's LUT to the adjacent LE for fast sequential LUT connections within the same LAB. Register chain connections transfer the output of one LE's register to the adjacent LE's register within a LAB. The Quartus® II Compiler places associated logic within a LAB or adjacent LABs, allowing the use of local, LUT chain, and register chain connections for performance and area efficiency. Figure 2–2 details the Cyclone LAB.

Figure 2–2. Cyclone LAB Structure

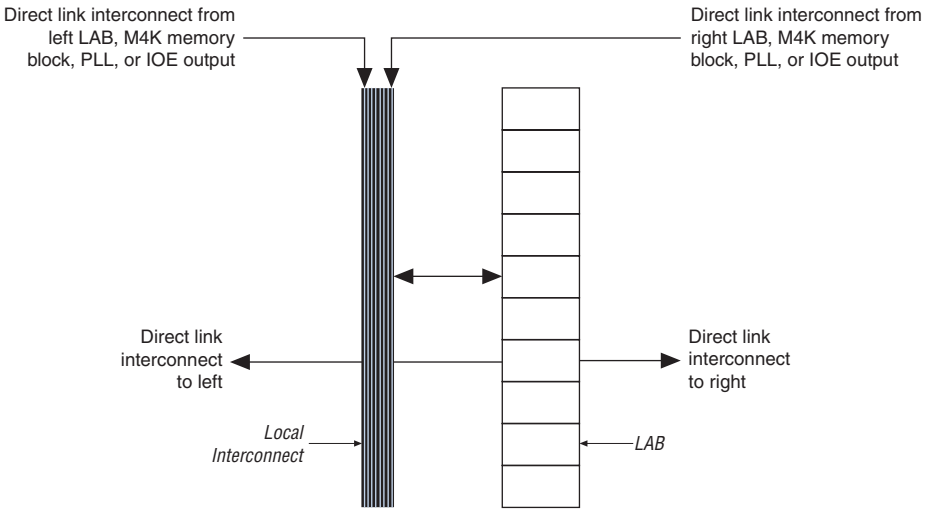


LAB Interconnects

The LAB local interconnect can drive LEs within the same LAB. The LAB local interconnect is driven by column and row interconnects and LE outputs within the same LAB. Neighboring LABs, PLLs, and M4K RAM blocks from the left and right can also drive a LAB's local interconnect 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 30 other LEs through fast local and direct link interconnects. Figure 2–3 shows the direct link connection.

Figure 2–3. 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, synchronous clear, asynchronous preset/load, synchronous load, and add/subtract control signals. This gives a maximum of 10 control signals at a time. Although synchronous load and clear signals are generally used when implementing counters, they can also be used with other functions.

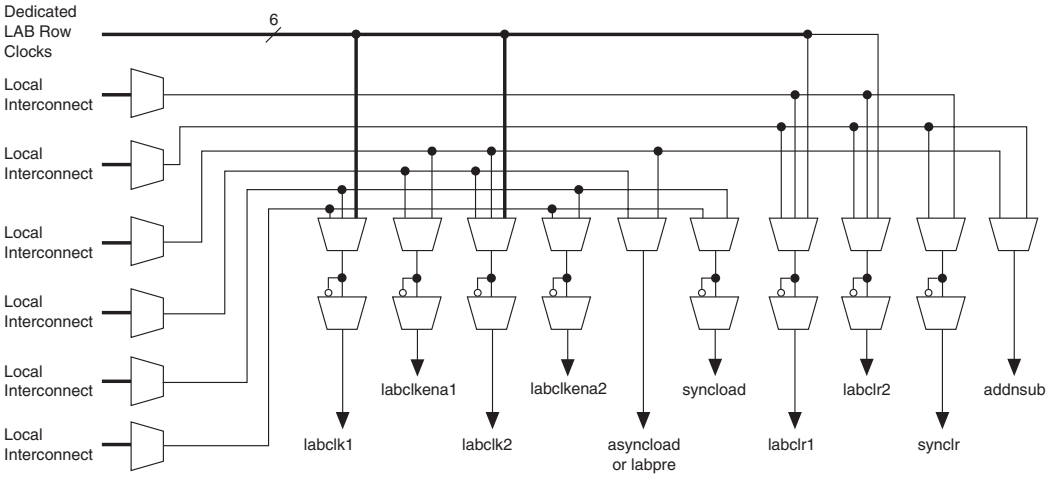
Each LAB can use two clocks and two clock enable signals. Each LAB's clock and clock enable signals are linked. For example, any LE in a particular LAB using the `labclk1` signal will also use `labclkena1`. If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock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

Each LAB can use two asynchronous clear signals and an asynchronous load/preset signal. The asynchronous load acts as a preset when the asynchronous load data input is tied high.

With the LAB-wide `addnsub` control signal, a single LE can implement a one-bit adder and subtractor. This saves LE resources and improves performance for logic functions such as DSP correlators and signed multipliers that alternate between addition and subtraction depending on data.

The LAB row clocks `[5..0]` and LAB local interconnect generate the LAB-wide control signals. The MultiTrack™ interconnect's inherent low skew allows clock and control signal distribution in addition to data. Figure 2–4 shows the LAB control signal generation circuit.

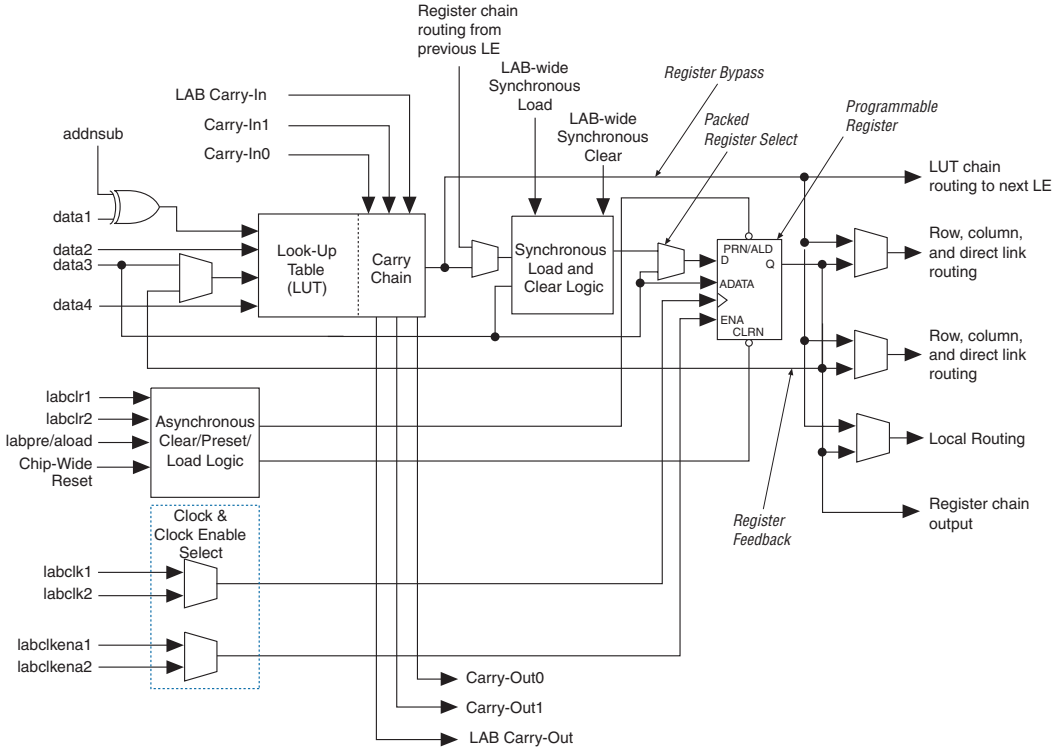
Figure 2–4. LAB-Wide Control Signals



Logic Elements

The smallest unit of logic in the Cyclone architecture, the LE, is compact and provides advanced features with efficient logic utilization. Each LE contains a four-input LUT, which is a function generator that can implement any function of four variables. In addition, each LE contains a programmable register and carry chain with carry select capability. A single LE also supports dynamic single bit addition or subtraction mode selectable by a LAB-wide control signal. Each LE drives all types of interconnects: local, row, column, LUT chain, register chain, and direct link interconnects. See Figure 2–5.

Figure 2–5. Cyclone LE



Each LE's programmable register can be configured for D, T, JK, or SR operation. Each register has data, true asynchronous load data, clock, clock enable, clear, and asynchronous load/preset inputs. Global signals, general-purpose I/O pins, or any internal logic can drive the register's clock and clear control signals. Either general-purpose I/O pins or internal logic can drive the clock enable, preset, asynchronous load, and asynchronous data. The asynchronous load data input comes from the `data3` input of the LE. For combinatorial 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 can drive these three outputs independently. Two LE outputs drive column or row and direct link routing connections and one drives 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. Another special packing mode allows the register output to feed back into the LUT of the same LE so that the register is packed with its own fan-out LUT. This provides another mechanism for improved fitting. The LE can also drive out registered and unregistered versions of the LUT output.

LUT Chain and Register Chain

In addition to the three general routing outputs, the LEs within a LAB have LUT chain and register chain outputs. LUT chain connections allow LUTs within the same LAB to cascade together for wide input functions. Register chain outputs allow registers within the same LAB to cascade together. The register chain output allows a LAB to use LUTs for a single combinatorial function and the registers to be used for an unrelated shift register implementation. These resources speed up connections between LABs while saving local interconnect resources. "MultiTrack Interconnect" on page 2–12 for more information on LUT chain and register chain connections.

addnsub Signal

The LE's dynamic adder/subtractor feature saves logic resources by using one set of LEs to implement both an adder and a subtractor. This feature is controlled by the LAB-wide control signal `addnsub`. The `addnsub` signal sets the LAB to perform either A + B or A –B. The LUT computes addition; subtraction is computed by adding the two's complement of the intended subtractor. The LAB-wide signal converts to two's complement by inverting the B bits within the LAB and setting carry-in = 1 to add one to the least significant bit (LSB). The LSB of an adder/subtractor must be placed in the first LE of the LAB, where the LAB-wide `addnsub` signal automatically sets the carry-in to 1. The Quartus II Compiler automatically places and uses the adder/subtractor feature when using adder/subtractor parameterized functions.

LE Operating Modes

The Cyclone LE can operate in one of the following modes:

- Normal mode
- Dynamic arithmetic mode

Each mode uses LE resources differently. In each mode, eight available inputs to the LE—the four data inputs from the LAB local interconnect, `carry-in0` and `carry-in1` from the previous LE, the LAB carry-in from the previous carry-chain LAB, and the register chain connection—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, asynchronous

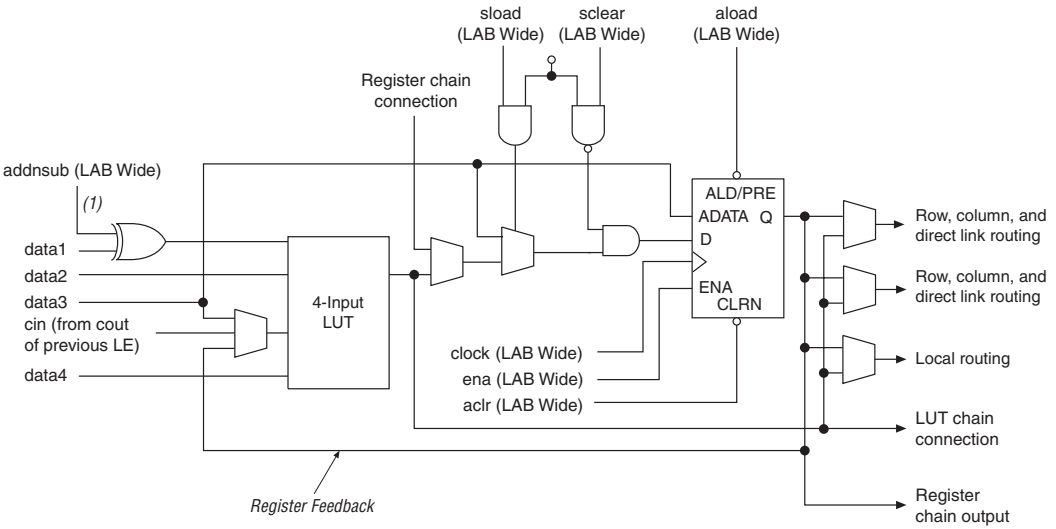
preset/load, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all LE modes. The `addnsub` control signal is allowed in arithmetic mode.

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

Normal Mode

The normal mode is suitable for general logic applications and combinatorial functions. In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT (see Figure 2–6). The Quartus II Compiler automatically selects the carry-in or the `data3` signal as one of the inputs to the LUT. Each LE can use LUT chain connections to drive its combinatorial output directly to the next LE in the LAB. Asynchronous load data for the register comes from the `data3` input of the LE. LEs in normal mode support packed registers.

Figure 2–6. LE in Normal Mode



Note to Figure 2–6:
(1) This signal is only allowed in normal mode if the LE is at the end of an adder/subtractor chain.

Dynamic Arithmetic Mode

The dynamic arithmetic mode is ideal for implementing adders, counters, accumulators, wide parity functions, and comparators. An LE in dynamic arithmetic mode uses four 2-input LUTs configurable as a dynamic adder/subtractor. The first two 2-input LUTs compute two summations based on a possible carry-in of 1 or 0; the other two LUTs generate carry outputs for the two chains of the carry select circuitry. As shown in Figure 2–7, the LAB carry-in signal selects either the `carry-in0` or `carry-in1` chain. The selected chain's logic level in turn determines which parallel sum is generated as a combinatorial or registered output. For example, when implementing an adder, the sum output is the selection of two possible calculated sums:

```
data1 + data2 + carry-in0
```

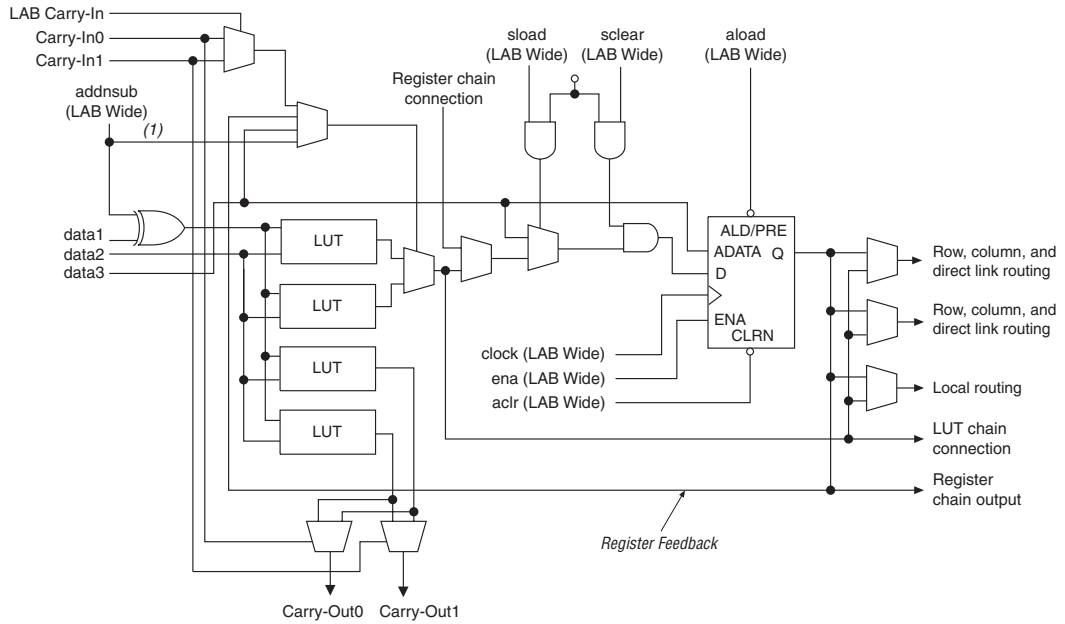
or

```
data1 + data2 + carry-in1
```

The other two LUTs use the `data1` and `data2` signals to generate two possible carry-out signals—one for a carry of 1 and the other for a carry of 0. The `carry-in0` signal acts as the carry select for the `carry-out0` output and `carry-in1` acts as the carry select for the `carry-out1` output. LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output.

The dynamic arithmetic mode also offers clock enable, counter enable, synchronous up/down control, synchronous clear, synchronous load, and dynamic adder/subtractor options. The LAB local interconnect data inputs generate the counter enable and synchronous up/down control signals. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. The Quartus II software automatically places any registers that are not used by the counter into other LABs. The `addnsub` LAB-wide signal controls whether the LE acts as an adder or subtractor.

Figure 2–7. LE in Dynamic Arithmetic Mode



Note to Figure 2–7:
(1) The `addnsub` signal is tied to the carry input for the first LE of a carry chain only.

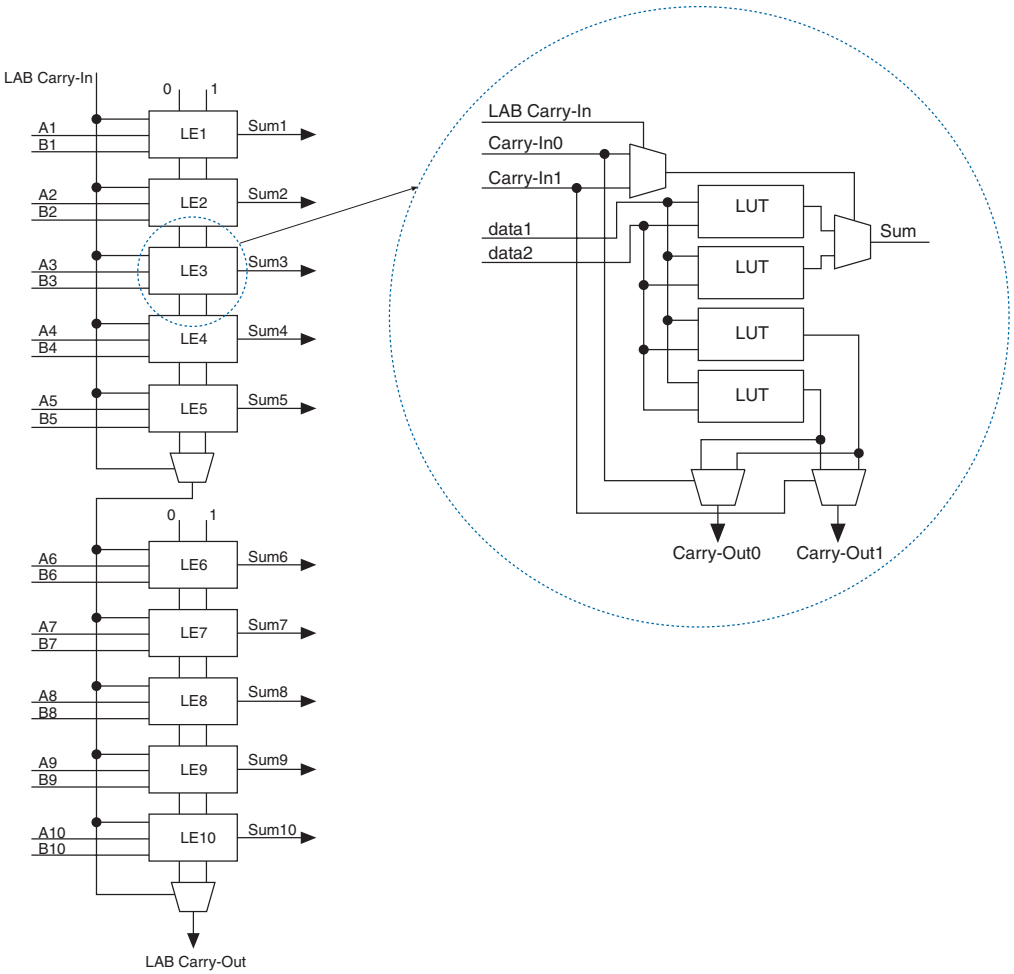
Carry-Select Chain

The carry-select chain provides a very fast carry-select function between LEs in dynamic arithmetic mode. The carry-select chain uses the redundant carry calculation to increase the speed of carry functions. The LE is configured to calculate outputs for a possible carry-in of 0 and carry-in of 1 in parallel. The `carry-in0` and `carry-in1` signals from a lower-order bit feed forward into the higher-order bit via the parallel carry chain and feed into both the LUT and the next portion of the carry chain. Carry-select chains can begin in any LE within a LAB.

The speed advantage of the carry-select chain is in the parallel pre-computation of carry chains. Since the LAB carry-in selects the precomputed carry chain, not every LE is in the critical path. Only the propagation delays between LAB carry-in generation (LE 5 and LE 10) are now part of the critical path. This feature allows the Cyclone architecture to implement high-speed counters, adders, multipliers, parity functions, and comparators of arbitrary width.

Figure 2–8 shows the carry-select circuitry in a LAB for a 10-bit full adder. One portion of the LUT generates the sum of two bits using the input signals and the appropriate carry-in bit; the sum is routed to the output of the LE. The register can be bypassed for simple adders or used for accumulator functions. Another portion of the LUT generates carry-out bits. A LAB-wide carry-in bit selects which chain is used for the addition of given inputs. The carry-in signal for each chain, `carry-in0` or `carry-in1`, selects the carry-out to carry forward to the carry-in signal of the next-higher-order bit. The final carry-out signal is routed to an LE, where it is fed to local, row, or column interconnects.

Figure 2–8. Carry Select Chain



The Quartus II Compiler automatically creates carry chain logic during design processing, or you can create it manually 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 10 LEs by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically allowing fast horizontal connections to M4K memory blocks. A carry chain can continue as far as a full column.

Clear and Preset Logic Control

LAB-wide signals control the logic for the register's clear and preset signals. The LE directly supports an asynchronous clear and preset function. The register preset is achieved through the asynchronous load of a logic high. The direct asynchronous preset does not require a NOT-gate push-back technique. Cyclone devices support simultaneous preset/ asynchronous load and clear signals. An asynchronous clear signal takes precedence if both signals are asserted simultaneously. Each LAB supports up to two clears and one preset signal.

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

MultiTrack Interconnect

In the Cyclone architecture, connections between LEs, M4K memory blocks, and device I/O pins are provided by the MultiTrack interconnect structure with DirectDrive™ technology. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different speeds used for inter- and intra-design block connectivity. The Quartus II Compiler automatically places critical design paths on faster interconnects to improve design performance.

DirectDrive technology is a deterministic routing technology that ensures identical routing resource usage for any function regardless of placement within the device. The MultiTrack interconnect and DirectDrive technology simplify the integration stage of block-based designing by eliminating the re-optimization cycles that typically follow design changes and additions.

The MultiTrack interconnect consists of row and column interconnects that span fixed distances. A routing structure with fixed length resources for all devices allows predictable and repeatable performance when

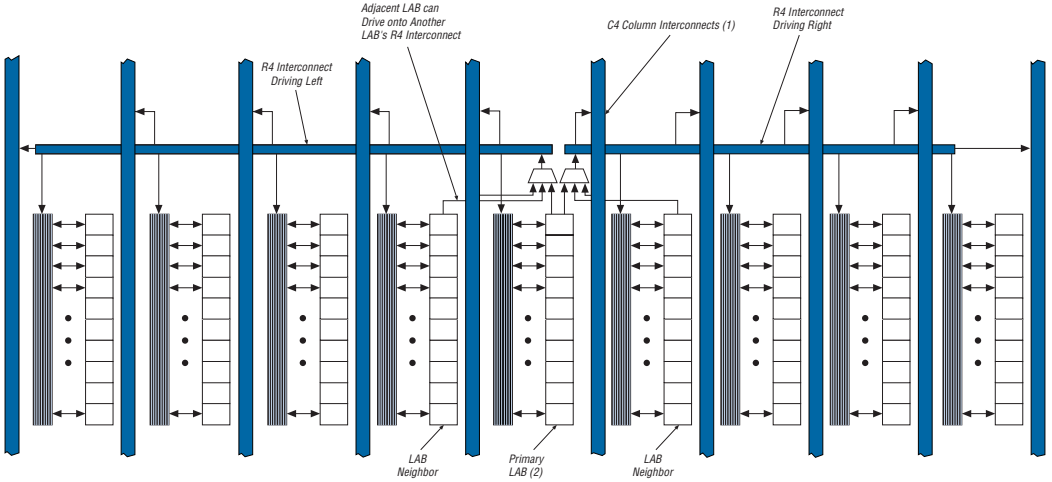
migrating through different device densities. Dedicated row interconnects route signals to and from LABs, PLLs, and M4K memory blocks within the same row. These row resources include:

- Direct link interconnects between LABs and adjacent blocks
- R4 interconnects traversing four blocks to the right or left

The direct link interconnect allows a LAB or M4K memory block to drive into the local interconnect of its left and right neighbors. Only one side of a PLL block interfaces with direct link and row interconnects. The direct link interconnect provides fast communication between adjacent LABs and/or blocks without using row interconnect resources.

The R4 interconnects span four LABs, or two LABs and one M4K RAM block. These resources are used for fast row connections in a four-LAB region. Every LAB has its own set of R4 interconnects to drive either left or right. Figure 2–9 shows R4 interconnect connections from a LAB. R4 interconnects can drive and be driven by M4K memory blocks, PLLs, and row IOEs. For LAB interfacing, a primary LAB or LAB neighbor can drive a given R4 interconnect. For R4 interconnects that drive to the right, the primary LAB and right neighbor can drive on to the interconnect. For R4 interconnects that drive to the left, the primary LAB and its left neighbor can drive on to the interconnect. R4 interconnects can drive other R4 interconnects to extend the range of LABs they can drive. R4 interconnects can also drive C4 interconnects for connections from one row to another.

Figure 2–9. R4 Interconnect Connections



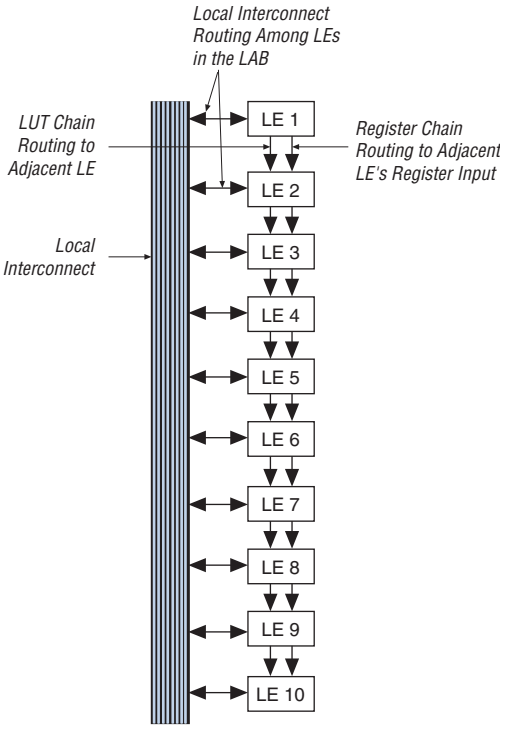
Notes to Figure 2–9:
(1) C4 interconnects can drive R4 interconnects.
(2) This pattern is repeated for every LAB in the LAB row.

The column interconnect operates similarly to the row interconnect. Each column of LABs is served by a dedicated column interconnect, which vertically routes signals to and from LABs, M4K memory blocks, and row and column IOEs. These column resources include:

- LUT chain interconnects within a LAB
- Register chain interconnects within a LAB
- C4 interconnects traversing a distance of four blocks in an up and down direction

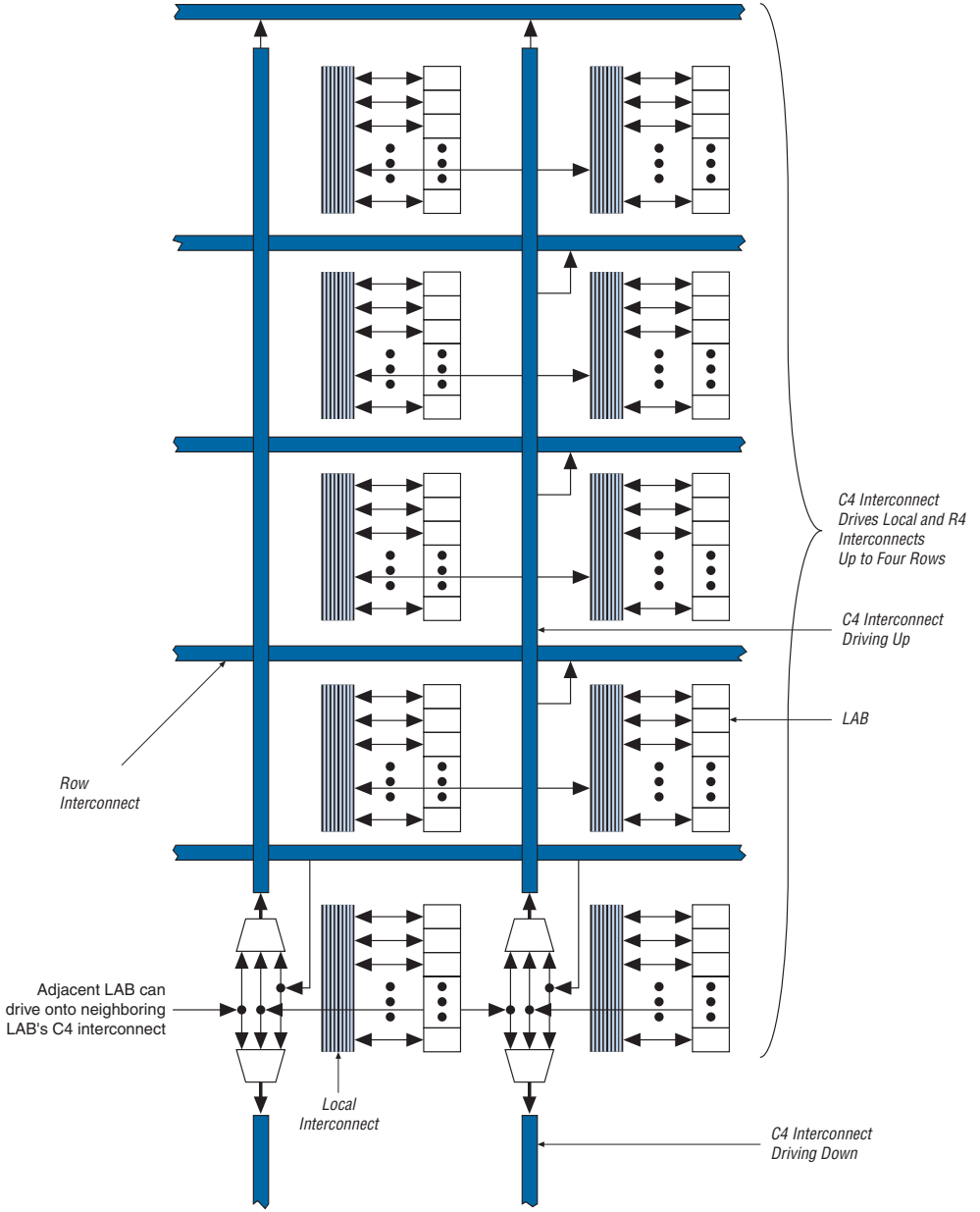
Cyclone devices include an enhanced interconnect structure within LABs for routing LE output to LE input connections faster using LUT chain connections and register chain connections. The LUT chain connection allows the combinatorial output of an LE to directly drive the fast input of the LE right below it, bypassing the local interconnect. These resources can be used as a high-speed connection for wide fan-in functions from LE 1 to LE 10 in the same LAB. The register chain connection allows the register output of one LE to connect directly to the register input of the next LE in the LAB for fast shift registers. The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance. Figure 2–10 shows the LUT chain and register chain interconnects.

Figure 2–10. LUT Chain and Register Chain Interconnects



The C4 interconnects span four LABs or M4K blocks up or down from a source LAB. Every LAB has its own set of C4 interconnects to drive either up or down. Figure 2–11 shows the C4 interconnect connections from a LAB in a column. The C4 interconnects can drive and be driven by all types of architecture blocks, including PLLs, M4K memory blocks, and column and row IOEs. For LAB interconnection, a primary LAB or its LAB neighbor can drive a given C4 interconnect. C4 interconnects can drive each other to extend their range as well as drive row interconnects for column-to-column connections.

Figure 2–11. C4 Interconnect Connections *Note (1)*



Note to Figure 2–11:
(1) Each C4 interconnect can drive either up or down four rows.

All embedded blocks communicate with the logic array similar to LAB-to-LAB interfaces. Each block (i.e., M4K memory or PLL) connects to row and column interconnects and has local interconnect regions driven by row and column interconnects. These blocks also have direct link interconnects for fast connections to and from a neighboring LAB.

Table 2–2 shows the Cyclone device's routing scheme.

Table 2–2. Cyclone Device Routing Scheme

Source	Destination										
	LUT Chain	Register Chain	Local Interconnect	Direct Link Interconnect	R4 Interconnect	C4 Interconnect	LE	M4K RAM Block	PLL	Column IOE	Row IOE
LUT Chain	—	—	—	—	—	—	✓	—	—	—	—
Register Chain	—	—	—	—	—	—	✓	—	—	—	—
Local Interconnect	—	—	—	—	—	—	✓	✓	✓	✓	✓
Direct Link Interconnect	—	—	✓	—	—	—	—	—	—	—	—
R4 Interconnect	—	—	✓	—	✓	✓	—	—	—	—	—
C4 Interconnect	—	—	✓	—	✓	✓	—	—	—	—	—
LE	✓	✓	✓	✓	✓	✓	—	—	—	—	—
M4K RAM Block	—	—	✓	✓	✓	✓	—	—	—	—	—
PLL	—	—	—	✓	✓	✓	—	—	—	—	—
Column IOE	—	—	—	—	—	✓	—	—	—	—	—
Row IOE	—	—	—	✓	✓	✓	—	—	—	—	—

Embedded Memory

The Cyclone embedded memory consists of columns of M4K memory blocks. EP1C3 and EP1C6 devices have one column of M4K blocks, while EP1C12 and EP1C20 devices have two columns (refer to Table 1–1 on page 1–1 for total RAM bits per density). Each M4K block can implement various types of memory with or without parity, including true dual-port, simple dual-port, and single-port RAM, ROM, and FIFO buffers. The M4K blocks support the following features:

- 4,608 RAM bits
- 250 MHz performance
- True dual-port memory
- Simple dual-port memory
- Single-port memory
- Byte enable
- Parity bits
- Shift register
- FIFO buffer
- ROM
- Mixed clock mode

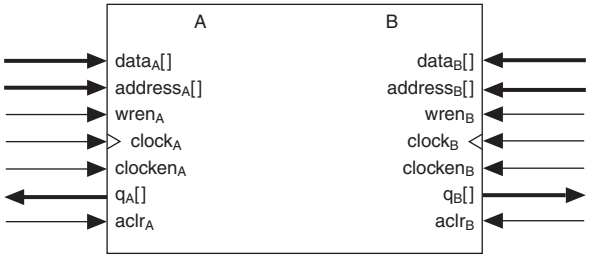


Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Memory Modes

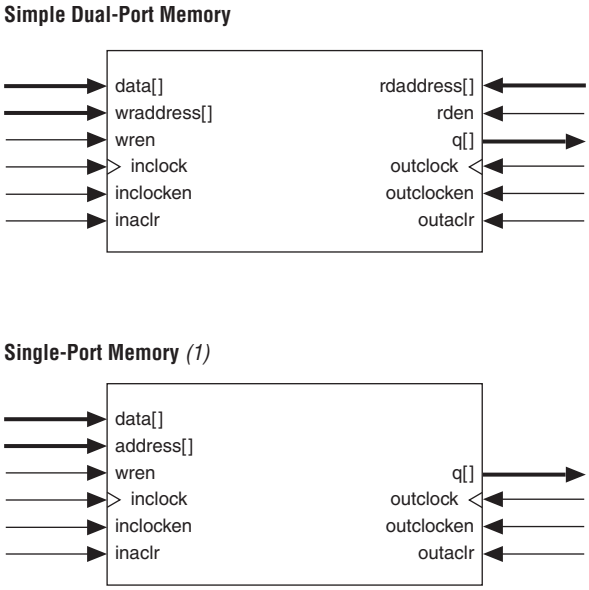
The M4K memory blocks include input registers that synchronize writes and output registers to pipeline designs and improve system performance. M4K blocks offer a true dual-port mode to support any combination of two-port operations: two reads, two writes, or one read and one write at two different clock frequencies. Figure 2–12 shows true dual-port memory.

Figure 2–12. True Dual-Port Memory Configuration



In addition to true dual-port memory, the M4K memory blocks support simple dual-port and single-port RAM. Simple dual-port memory supports a simultaneous read and write. Single-port memory supports non-simultaneous reads and writes. Figure 2–13 shows these different M4K RAM memory port configurations.

Figure 2–13. Simple Dual-Port and Single-Port Memory Configurations



Note to Figure 2–13:

(1) Two single-port memory blocks can be implemented in a single M4K block as long as each of the two independent block sizes is equal to or less than half of the M4K block size.

The memory blocks also enable mixed-width data ports for reading and writing to the RAM ports in dual-port RAM configuration. For example, the memory block can be written in ×1 mode at port A and read out in ×16 mode from port B.

The Cyclone memory architecture can implement fully synchronous RAM by registering both the input and output signals to the M4K RAM block. All M4K memory block inputs are registered, providing synchronous write cycles. In synchronous operation, the memory block generates its own self-timed strobe write enable (`wren`) signal derived from a global clock. In contrast, a circuit using asynchronous RAM must generate the RAM `wren` signal while ensuring its data and address signals meet setup and hold time specifications relative to the `wren`

signal. The output registers can be bypassed. Pseudo-asynchronous reading is possible in the simple dual-port mode of M4K blocks by clocking the read enable and read address registers on the negative clock edge and bypassing the output registers.

When configured as RAM or ROM, you can use an initialization file to pre-load the memory contents.

Two single-port memory blocks can be implemented in a single M4K block as long as each of the two independent block sizes is equal to or less than half of the M4K block size.

The Quartus II software automatically implements larger memory by combining multiple M4K memory blocks. For example, two 256×16-bit RAM blocks can be combined to form a 256×32-bit RAM block. Memory performance does not degrade for memory blocks using the maximum number of words allowed. Logical memory blocks using less than the maximum number of words use physical blocks in parallel, eliminating any external control logic that would increase delays. To create a larger high-speed memory block, the Quartus II software automatically combines memory blocks with LE control logic.

Parity Bit Support

The M4K blocks support a parity bit for each byte. The parity bit, along with internal LE logic, can implement parity checking for error detection to ensure data integrity. You can also use parity-size data words to store user-specified control bits. Byte enables are also available for data input masking during write operations.

Shift Register Support

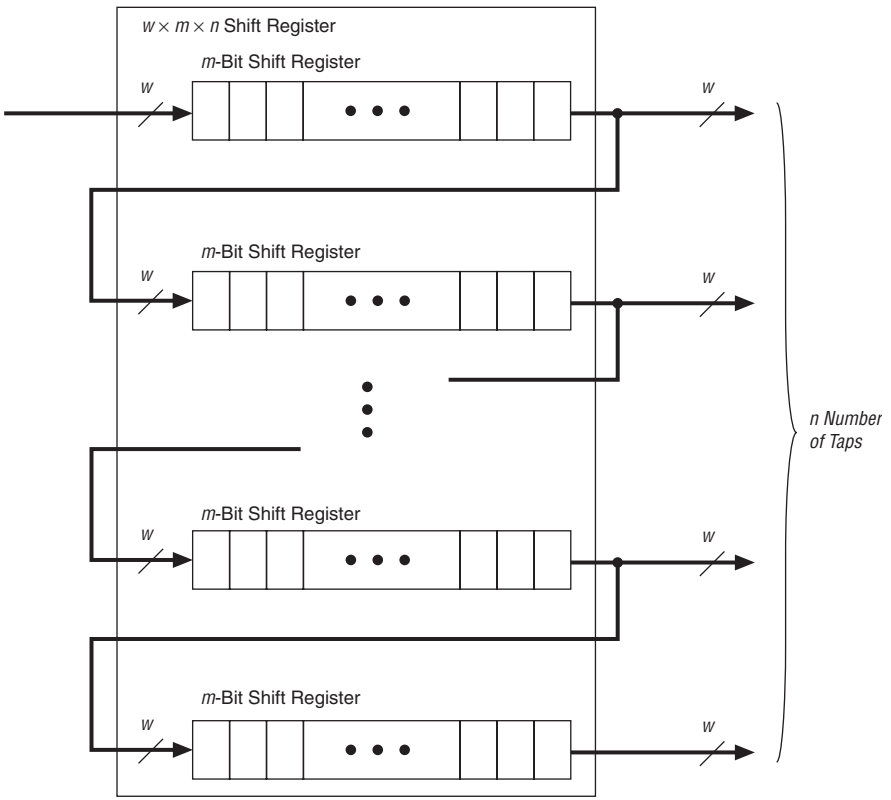
You can configure M4K memory blocks to implement shift registers for DSP applications such as pseudo-random number generators, multi-channel filtering, auto-correlation, and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flip-flops, which can quickly consume many logic cells and routing resources 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 and provides a more efficient implementation with the dedicated circuitry.

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). The size of a $w \times m \times n$ shift register must be less than or equal to the maximum number of memory bits in the M4K block (4,608 bits). The total number of shift

register outputs (number of taps n × width w) must be less than the maximum data width of the M4K RAM block (×36). To create larger shift registers, multiple memory blocks are cascaded together.

Data is written into each address location at the falling edge of the clock and read from the address at the rising edge of the clock. The shift register mode logic automatically controls the positive and negative edge clocking to shift the data in one clock cycle. Figure 2–14 shows the M4K memory block in the shift register mode.

Figure 2–14. Shift Register Memory Configuration



Memory Configuration Sizes

The memory address depths and output widths can be configured as 4,096 × 1, 2,048 × 2, 1,024 × 4, 512 × 8 (or 512 × 9 bits), 256 × 16 (or 256 × 18 bits), and 128 × 32 (or 128 × 36 bits). The 128 × 32- or 36-bit configuration

is not available in the true dual-port mode. Mixed-width configurations are also possible, allowing different read and write widths. Tables 2–3 and 2–4 summarize the possible M4K RAM block configurations.

Table 2–3. M4K RAM Block Configurations (Simple Dual-Port)

Read Port	Write Port								
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	128 × 32	512 × 9	256 × 18	128 × 36
4K × 1	✓	✓	✓	✓	✓	✓	—	—	—
2K × 2	✓	✓	✓	✓	✓	✓	—	—	—
1K × 4	✓	✓	✓	✓	✓	✓	—	—	—
512 × 8	✓	✓	✓	✓	✓	✓	—	—	—
256 × 16	✓	✓	✓	✓	✓	✓	—	—	—
128 × 32	✓	✓	✓	✓	✓	✓	—	—	—
512 × 9	—	—	—	—	—	—	✓	✓	✓
256 × 18	—	—	—	—	—	—	✓	✓	✓
128 × 36	—	—	—	—	—	—	✓	✓	✓

Table 2–4. M4K RAM Block Configurations (True Dual-Port)

Port A	Port B						
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	512 × 9	256 × 18
4K × 1	✓	✓	✓	✓	✓	—	—
2K × 2	✓	✓	✓	✓	✓	—	—
1K × 4	✓	✓	✓	✓	✓	—	—
512 × 8	✓	✓	✓	✓	✓	—	—
256 × 16	✓	✓	✓	✓	✓	—	—
512 × 9	—	—	—	—	—	✓	✓
256 × 18	—	—	—	—	—	✓	✓

When the M4K RAM block is configured as a shift register block, you can create a shift register up to 4,608 bits ($w \times m \times n$).

Byte Enables

M4K blocks support byte writes when the write port has a data width of 16, 18, 32, or 36 bits. The byte enables allow the input data to be masked so the device can write to specific bytes. The unwritten bytes retain the previous written value. Table 2–5 summarizes the byte selection.

Table 2–5. Byte Enable for M4K Blocks *Notes (1), (2)*

byteena[3..0]	datain ×18	datain ×36
[0] = 1	[8..0]	[8..0]
[1] = 1	[17..9]	[17..9]
[2] = 1	—	[26..18]
[3] = 1	—	[35..27]

Notes to Table 2–5:
(1) Any combination of byte enables is possible.
(2) Byte enables can be used in the same manner with 8-bit words, i.e., in ×16 and ×32 modes.

Control Signals and M4K Interface

The M4K blocks allow for different clocks on their inputs and outputs. Either of the two clocks feeding the block can clock M4K block registers (`renwe`, address, byte enable, `datain`, and output registers). Only the output register can be bypassed. The six `labclk` signals or local interconnects can drive the control signals for the A and B ports of the M4K block. LEs can also control the `clock_a`, `clock_b`, `renwe_a`, `renwe_b`, `clr_a`, `clr_b`, `clocken_a`, and `clocken_b` signals, as shown in Figure 2–15.

The R4, C4, and direct link interconnects from adjacent LABs drive the M4K block local interconnect. The M4K blocks can communicate with LABs on either the left or right side through these row resources or with LAB columns on either the right or left with the column resources. Up to 10 direct link input connections to the M4K block are possible from the left adjacent LABs and another 10 possible from the right adjacent LAB. M4K block outputs can also connect to left and right LABs through 10 direct link interconnects each. Figure 2–16 shows the M4K block to logic array interface.

Figure 2–15. M4K RAM Block Control Signals

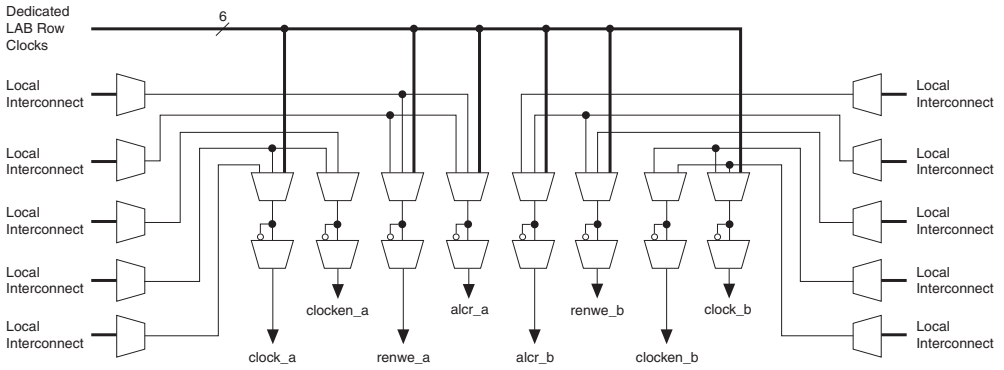
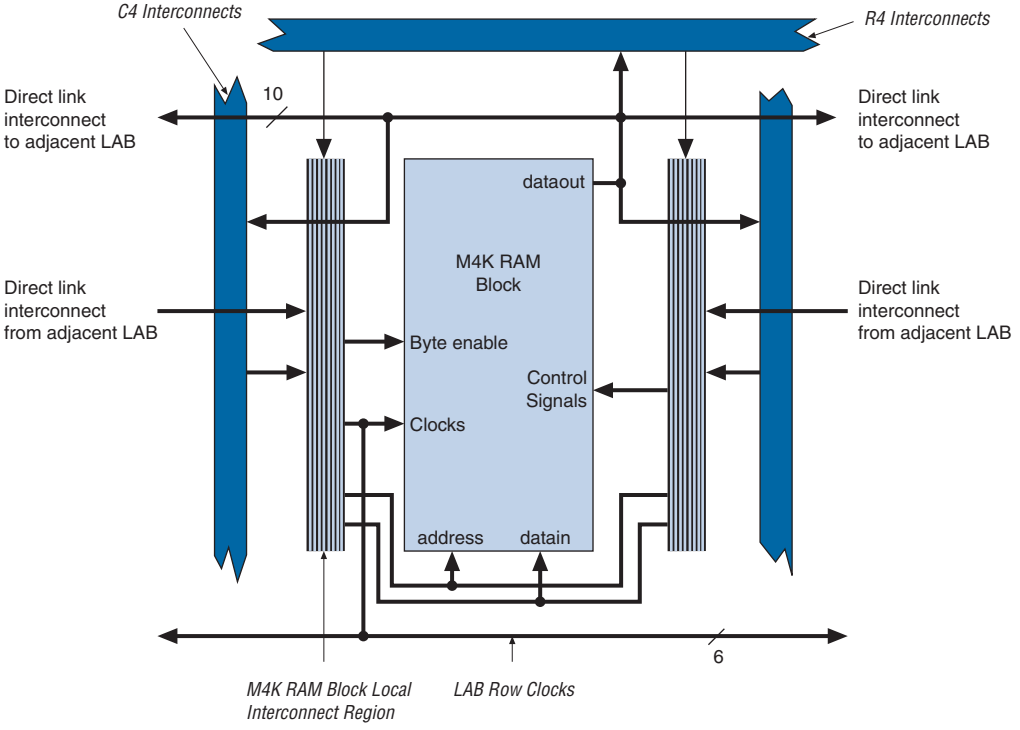


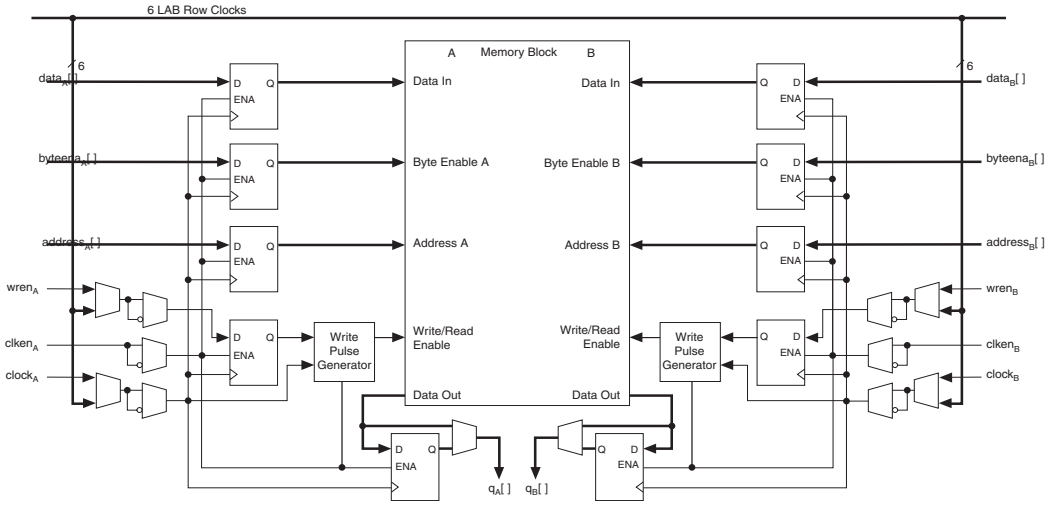
Figure 2–16. M4K RAM Block LAB Row Interface



Independent Clock Mode

The M4K memory blocks implement independent clock mode for true dual-port memory. In this mode, a separate clock is available for each port (ports A and B). Clock A controls all registers on the port A side, while clock B controls all registers on the port B side. Each port, A and B, also supports independent clock enables and asynchronous clear signals for port A and B registers. Figure 2–17 shows an M4K memory block in independent clock mode.

Figure 2–17. Independent Clock Mode *Notes (1), (2)*



Notes to Figure 2–17:

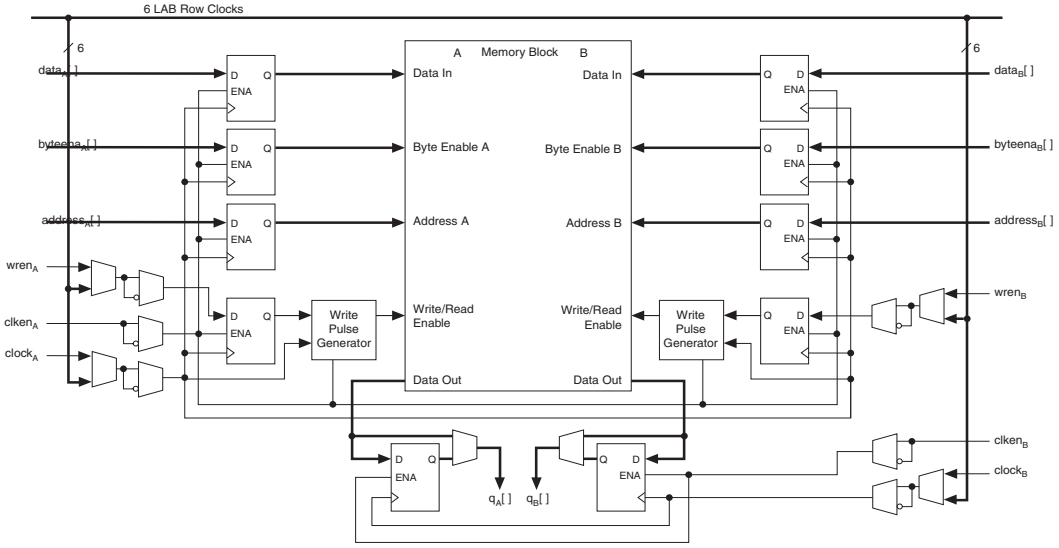
(1) All registers shown have asynchronous clear ports.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Input/Output Clock Mode

Input/output clock mode can be implemented for both the true and simple dual-port memory modes. On each of the two ports, A or B, one clock controls all registers for inputs into the memory block: data input, `wren`, and address. The other clock controls the block's data output registers. Each memory block port, A or B, also supports independent clock enables and asynchronous clear signals for input and output registers. Figures 2–18 and 2–19 show the memory block in input/output clock mode.

Figure 2–18. Input/Output Clock Mode in True Dual-Port Mode *Notes (1), (2)*

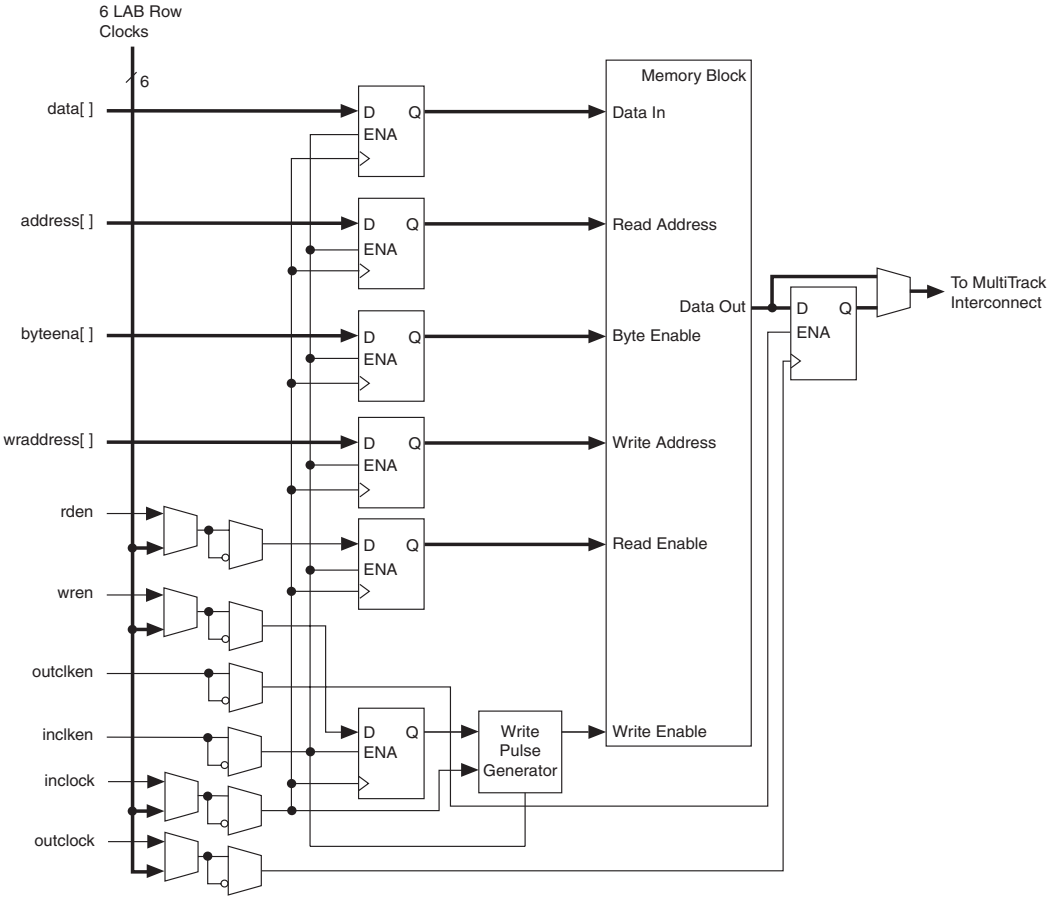


Notes to Figure 2–18:

(1) All registers shown have asynchronous clear ports.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Figure 2–19. Input/Output Clock Mode in Simple Dual-Port Mode *Notes (1), (2)*



Notes to Figure 2–19:

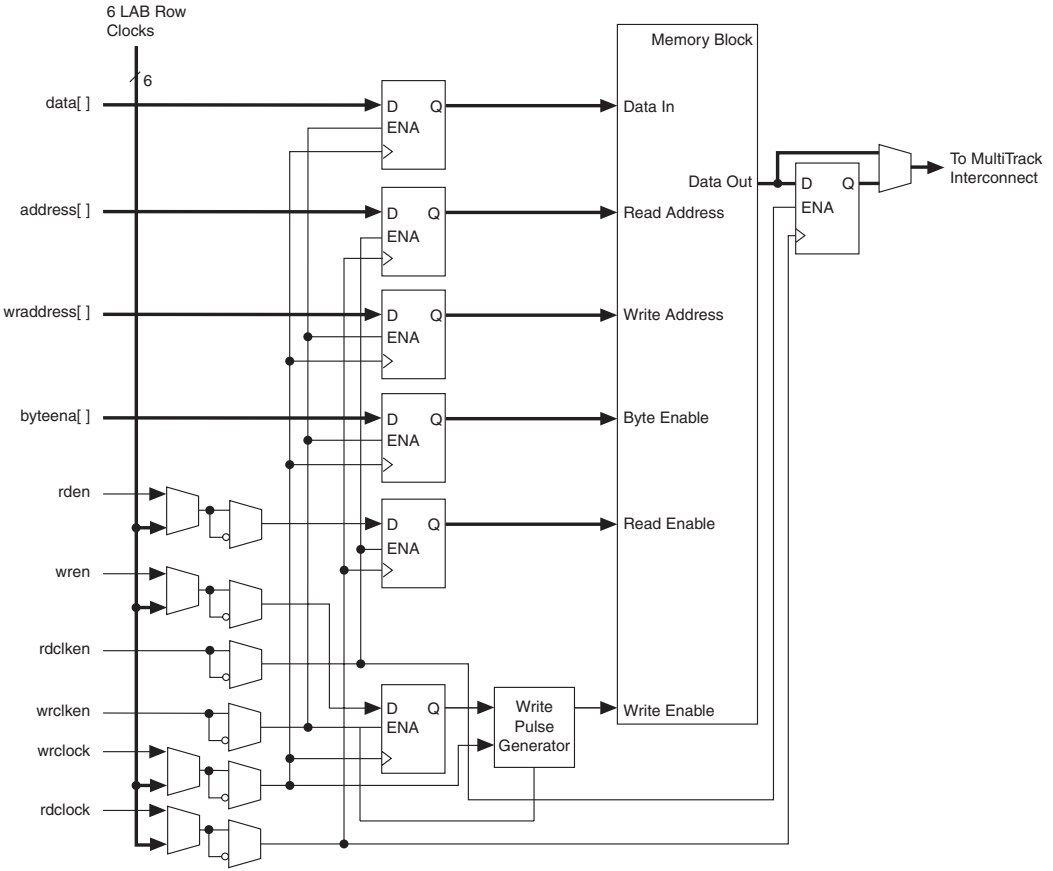
(1) All registers shown except the rden register have asynchronous clear ports.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Read/Write Clock Mode

The M4K memory blocks implement read/write clock mode for simple dual-port memory. You can use up to two clocks in this mode. The write clock controls the block's data inputs, `wraddress`, and `wren`. The read clock controls the data output, `rdaddress`, and `rden`. The memory blocks support independent clock enables for each clock and asynchronous clear signals for the read- and write-side registers. Figure 2–20 shows a memory block in read/write clock mode.

Figure 2–20. Read/Write Clock Mode in Simple Dual-Port Mode *Notes (1), (2)*



Notes to Figure 2–20:

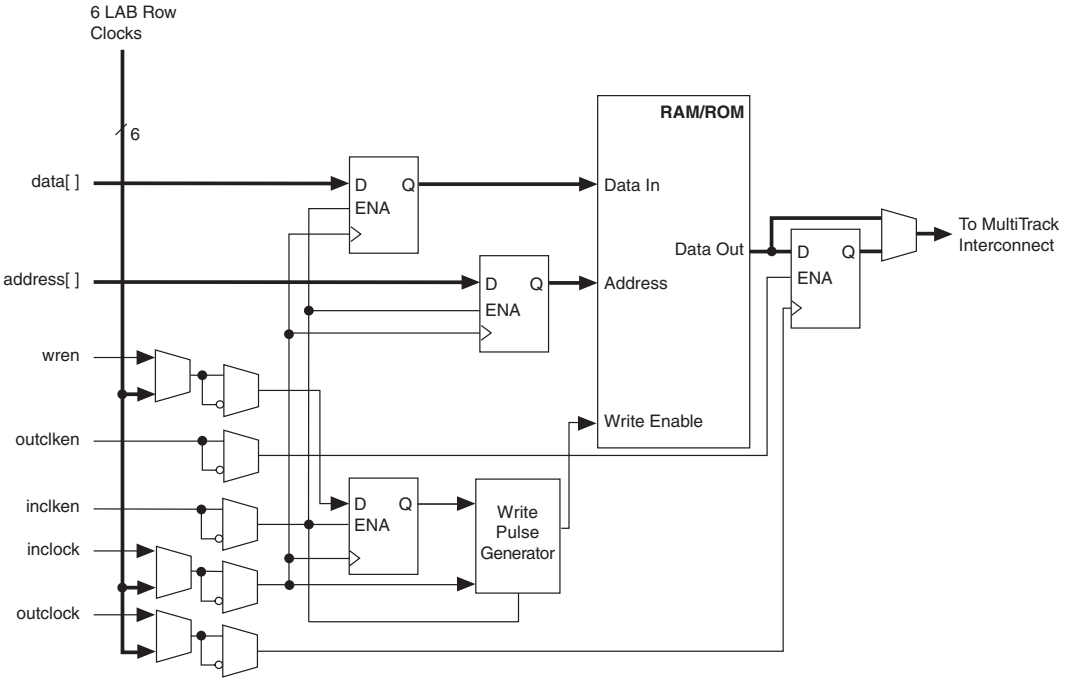
(1) All registers shown except the rden register have asynchronous clear ports.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Single-Port Mode

The M4K memory blocks also support single-port mode, used when simultaneous reads and writes are not required. See Figure 2–21. A single M4K memory block can support up to two single-port mode RAM blocks if each RAM block is less than or equal to 2K bits in size.

Figure 2–21. Single-Port Mode *Note (1)*



Note to Figure 2–21:

(1) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Global Clock Network and Phase-Locked Loops

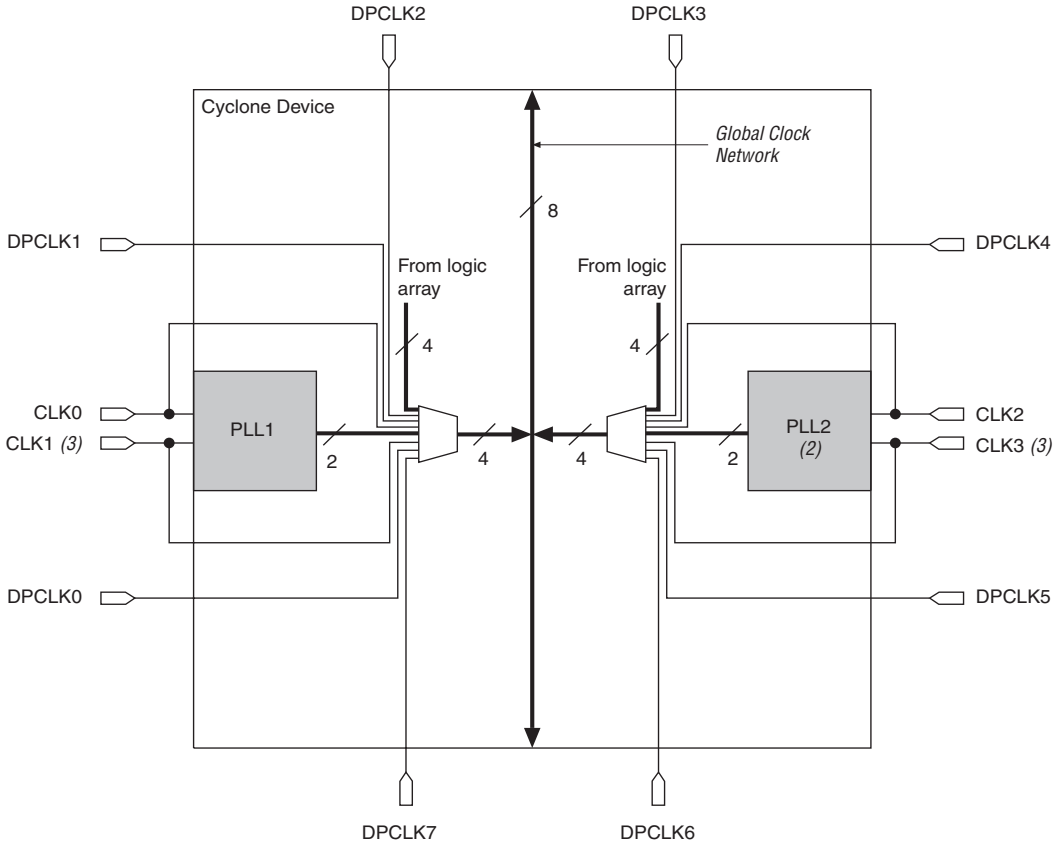
Cyclone devices provide a global clock network and up to two PLLs for a complete clock management solution.

Global Clock Network

There are four dedicated clock pins (`CLK[3..0]`, two pins on the left side and two pins on the right side) that drive the global clock network, as shown in Figure 2–22. PLL outputs, logic array, and dual-purpose clock (`DPCLK[7..0]`) pins can also drive the global clock network.

The eight global clock lines in the global clock network drive throughout the entire device. The global clock network can provide clocks for all resources within the device—IOEs, LEs, and memory blocks. The global clock lines can also be used for control signals, such as clock enables and synchronous or asynchronous clears fed from the external pin, or DQS signals for DDR SDRAM or FCRAM interfaces. Internal logic can also drive the global clock network for internally generated global clocks and asynchronous clears, clock enables, or other control signals with large fanout. Figure 2–22 shows the various sources that drive the global clock network.

Figure 2–22. Global Clock Generation *Note (1)*



Notes to Figure 2–22:

(1) The EP1C3 device in the 100-pin TQFP package has five DPCLK pins (DPCLK2, DPCLK3, DPCLK4, DPCLK6, and DPCLK7).

(2) EP1C3 devices only contain one PLL (PLL 1).

(3) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins CLK1 and CLK3.

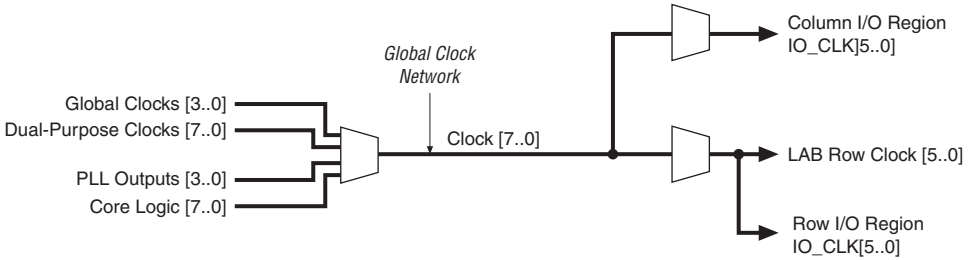
Dual-Purpose Clock Pins

Each Cyclone device except the EP1C3 device has eight dual-purpose clock pins, DPCLK[7..0] (two on each I/O bank). EP1C3 devices have five DPCLK pins in the 100-pin TQFP package. These dual-purpose pins can connect to the global clock network (see Figure 2–22) for high-fanout control signals such as clocks, asynchronous clears, presets, and clock enables, or protocol control signals such as TRDY and IRDY for PCI, or DQS signals for external memory interfaces.

Combined Resources

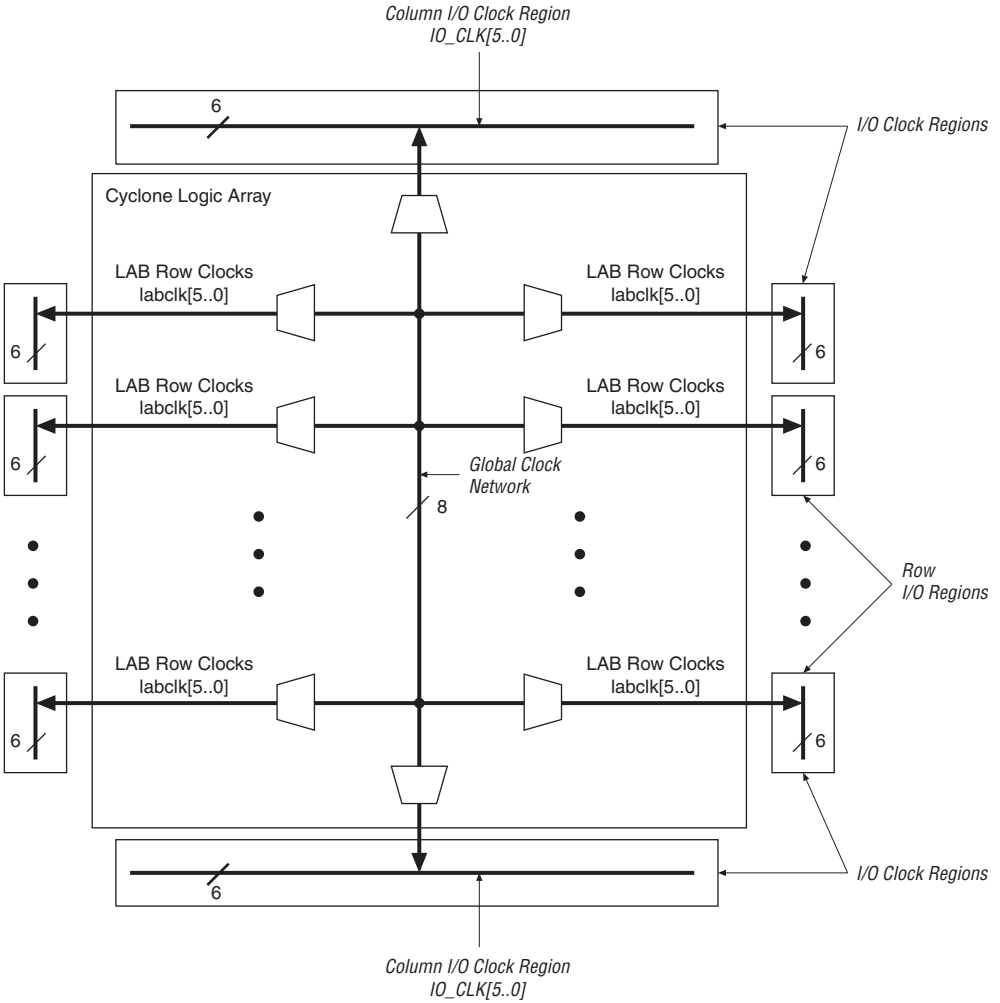
Each Cyclone device contains eight distinct dedicated clocking resources. The device uses multiplexers with these clocks to form six-bit buses to drive LAB row clocks, column IOE clocks, or row IOE clocks. See Figure 2–23. Another multiplexer at the LAB level selects two of the six LAB row clocks to feed the LE registers within the LAB.

Figure 2–23. Global Clock Network Multiplexers



IOE clocks have row and column block regions. Six of the eight global clock resources feed to these row and column regions. Figure 2–24 shows the I/O clock regions.

Figure 2–24. I/O Clock Regions



PLLs

Cyclone PLLs provide general-purpose clocking with clock multiplication and phase shifting as well as outputs for differential I/O support. Cyclone devices contain two PLLs, except for the EP1C3 device, which contains one PLL.

Table 2–6 shows the PLL features in Cyclone devices. Figure 2–25 shows a Cyclone PLL.

Table 2–6. Cyclone PLL Features

Feature	PLL Support
Clock multiplication and division	*m*/(*n* × post-scale counter) *(1)*
Phase shift	Down to 125-ps increments *(2)*, *(3)*
Programmable duty cycle	Yes
Number of internal clock outputs	2
Number of external clock outputs	One differential or one single-ended *(4)*

Notes to Table 2–6:

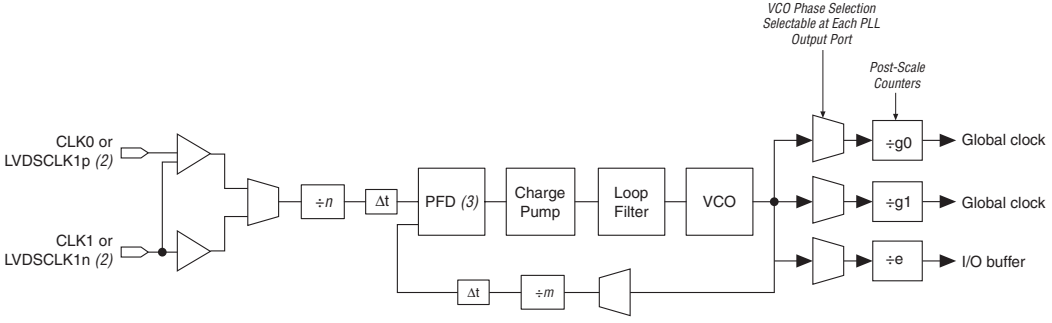
(1) The *m* counter ranges from 2 to 32. The *n* counter and the post-scale counters range from 1 to 32.

(2) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by 8.

(3) For degree increments, Cyclone devices can shift all output frequencies in increments of 45°. Smaller degree increments are possible depending on the frequency and divide parameters.

(4) The EP1C3 device in the 100-pin TQFP package does not support external clock output. The EP1C6 device in the 144-pin TQFP package does not support external clock output from PLL2.

Figure 2–25. Cyclone PLL *Note (1)*



Notes to Figure 2–25:

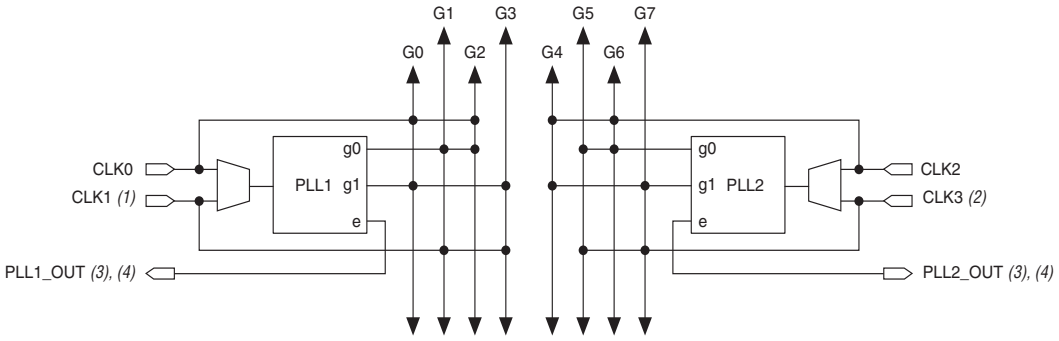
(1) The EP1C3 device in the 100-pin TQFP package does not support external outputs or LVDS inputs. The EP1C6 device in the 144-pin TQFP package does not support external output from PLL2.

(2) LVDS input is supported via the secondary function of the dedicated clock pins. For PLL 1, the `CLK0` pin's secondary function is `LVDSCLK1p` and the `CLK1` pin's secondary function is `LVDSCLK1n`. For PLL 2, the `CLK2` pin's secondary function is `LVDSCLK2p` and the `CLK3` pin's secondary function is `LVDSCLK2n`.

(3) PFD: phase frequency detector.

Figure 2–26 shows the PLL global clock connections.

Figure 2–26. Cyclone PLL Global Clock Connections



Notes to Figure 2–26:

(1) PLL 1 supports one single-ended or LVDS input via pins CLK0 and CLK1.

(2) PLL2 supports one single-ended or LVDS input via pins CLK2 and CLK3.

(3) PLL1_OUT and PLL2_OUT support single-ended or LVDS output. If external output is not required, these pins are available as regular user I/O pins.

(4) The EP1C3 device in the 100-pin TQFP package does not support external clock output. The EP1C6 device in the 144-pin TQFP package does not support external clock output from PLL2.

Table 2–7 shows the global clock network sources available in Cyclone devices.

Table 2–7. Global Clock Network Sources (Part 1 of 2)

Source		GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
PLL Counter Output	PLL1 G0	—	✓	✓	—	—	—	—	—
	PLL1 G1	✓	—	—	✓	—	—	—	—
	PLL2 G0 *(1)*	—	—	—	—	—	✓	✓	—
	PLL2 G1 *(1)*	—	—	—	—	✓	—	—	✓
Dedicated Clock Input Pins	CLK0	✓	—	✓	—	—	—	—	—
	CLK1 *(2)*	—	✓	—	✓	—	—	—	—
	CLK2	—	—	—	—	✓	—	✓	—
	CLK3 *(2)*	—	—	—	—	—	✓	—	✓

Table 2–7. Global Clock Network Sources (Part 2 of 2)

Source		GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
Dual-Purpose Clock Pins	DPCLK0 *(3)*	—	—	—	✓	—	—	—	—
	DPCLK1 *(3)*	—	—	✓	—	—	—	—	—
	DPCLK2	✓	—	—	—	—	—	—	—
	DPCLK3	—	—	—	—	✓	—	—	—
	DPCLK4	—	—	—	—	—	—	✓	—
	DPCLK5 *(3)*	—	—	—	—	—	—	—	✓
	DPCLK6	—	—	—	—	—	✓	—	—
	DPCLK7	—	✓	—	—	—	—	—	—

Notes to Table 2–7:
(1) EP1C3 devices only have one PLL (PLL 1).
(2) EP1C3 devices in the 100-pin TQFP package do not have dedicated clock pins `CLK1` and `CLK3`.
(3) EP1C3 devices in the 100-pin TQFP package do not have the `DPCLK0`, `DPCLK1`, or `DPCLK5` pins.

Clock Multiplication and Division

Cyclone PLLs provide clock synthesis for PLL output ports using $m/(n \times$ post scale counter) scaling factors. The input clock is divided by a pre-scale divider, n, and is then multiplied by the m feedback factor. The control loop drives the VCO to match $f_{IN} \times (m/n)$. Each output port has a unique post-scale counter to divide down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO is set to the least-common multiple of the output frequencies that meets its frequency specifications. Then, the post-scale dividers scale down the output frequency for each output port. For example, if the output frequencies required from one PLL are 33 and 66 MHz, the VCO is set to 330 MHz (the least-common multiple in the VCO's range).

Each PLL has one pre-scale divider, n, that can range in value from 1 to 32. Each PLL also has one multiply divider, m, that can range in value from 2 to 32. Global clock outputs have two post scale G dividers for global clock outputs, and external clock outputs have an E divider for external clock output, both ranging from 1 to 32. The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered.

External Clock Inputs

Each PLL supports single-ended or differential inputs for source-synchronous receivers or for general-purpose use. The dedicated clock pins (CLK[3..0]) feed the PLL inputs. These dual-purpose pins can also act as LVDS input pins. See Figure 2–25.

Table 2–8 shows the I/O standards supported by PLL input and output pins.

Table 2–8. PLL I/O Standards

I/O Standard	CLK Input	EXTCLK Output
3.3-V LVTTL/LVCMOS	✓	✓
2.5-V LVTTL/LVCMOS	✓	✓
1.8-V LVTTL/LVCMOS	✓	✓
1.5-V LVCMOS	✓	✓
3.3-V PCI	✓	✓
LVDS	✓	✓
SSTL-2 class I	✓	✓
SSTL-2 class II	✓	✓
SSTL-3 class I	✓	✓
SSTL-3 class II	✓	✓
Differential SSTL-2	—	✓

For more information on LVDS I/O support, refer to "LVDS I/O Pins" on page 2–54.

External Clock Outputs

Each PLL supports one differential or one single-ended output for source-synchronous transmitters or for general-purpose external clocks. If the PLL does not use these PLL_OUT pins, the pins are available for use as general-purpose I/O pins. The PLL_OUT pins support all I/O standards shown in Table 2–8.

The external clock outputs do not have their own V_{CC} and ground voltage supplies. Therefore, to minimize jitter, do not place switching I/O pins next to these output pins. The EP1C3 device in the 100-pin TQFP package

does not have dedicated clock output pins. The EP1C6 device in the 144-pin TQFP package only supports dedicated clock outputs from PLL 1.

Clock Feedback

Cyclone PLLs have three modes for multiplication and/or phase shifting:

- Zero delay buffer mode—The external clock output pin is phase-aligned with the clock input pin for zero delay.
- Normal mode—If the design uses an internal PLL clock output, the normal mode compensates for the internal clock delay from the input clock pin to the IOE registers. The external clock output pin is phase shifted with respect to the clock input pin if connected in this mode. You defines which internal clock output from the PLL should be phase-aligned to compensate for internal clock delay.
- No compensation mode—In this mode, the PLL will not compensate for any clock networks.

Phase Shifting

Cyclone PLLs have an advanced clock shift capability that enables programmable phase shifts. You can enter a phase shift (in degrees or time units) for each PLL clock output port or for all outputs together in one shift. You can perform phase shifting in time units with a resolution range of 125 to 250 ps. The finest resolution equals one eighth of the VCO period. The VCO period is a function of the frequency input and the multiplication and division factors. Each clock output counter can choose a different phase of the VCO period from up to eight taps. You can use this clock output counter along with an initial setting on the post-scale counter to achieve a phase-shift range for the entire period of the output clock. The phase tap feedback to the m counter can shift all outputs to a single phase. The Quartus II software automatically sets the phase taps and counter settings according to the phase shift entered.

Lock Detect Signal

The lock output indicates that there is a stable clock output signal in phase with the reference clock. Without any additional circuitry, the lock signal may toggle as the PLL begins tracking the reference clock. Therefore, you may need to gate the lock signal for use as a system-control signal. For correct operation of the lock circuit below –20 C, $f_{IN/N} > 200$ MHz.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on each PLL post-scale counter (g0, g1, e). The duty cycle setting is achieved by a low- and high-time count setting for the post-scale dividers. The Quartus II software uses the frequency input and the required multiply or divide rate to determine the duty cycle choices.

Control Signals

There are three control signals for clearing and enabling PLLs and their outputs. You can use these signals to control PLL resynchronization and the ability to gate PLL output clocks for low-power applications.

The `pllenable` signal enables and disables PLLs. When the `pllenable` signal is low, the clock output ports are driven by ground and all the PLLs go out of lock. When the `pllenable` signal goes high again, the PLLs relock and resynchronize to the input clocks. An input pin or LE output can drive the `pllenable` signal.

The `areset` signals are reset/resynchronization inputs for each PLL. Cyclone devices can drive these input signals from input pins or from LEs. When `areset` is driven high, the PLL counters will reset, clearing the PLL output and placing the PLL out of lock. When driven low again, the PLL will resynchronize to its input as it relocks.

The `pfdena` signals control the phase frequency detector (PFD) output with a programmable gate. If you disable the PFD, the VCO will operate at its last set value of control voltage and frequency with some drift, and the system will continue running when the PLL goes out of lock or the input clock disables. By maintaining the last locked frequency, the system has time to store its current settings before shutting down. You can either use their own control signal or gated locked status signals to trigger the `pfdena` signal.



For more information about Cyclone PLLs, refer to *Using PLLs in Cyclone Devices* chapter in the *Cyclone Device Handbook*.

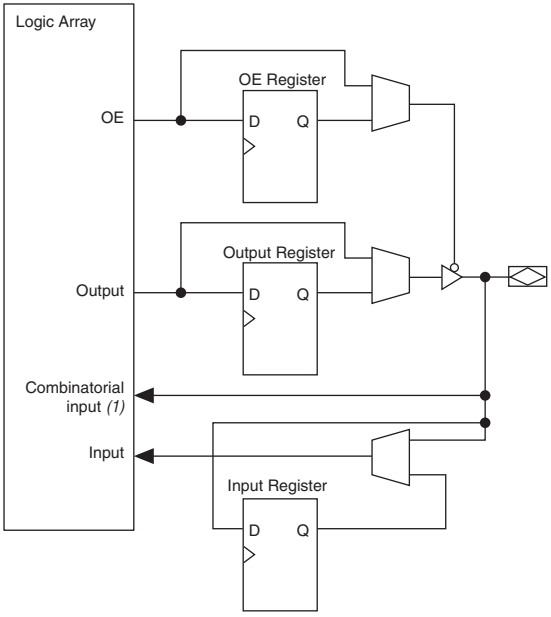
I/O Structure

IOEs support many features, including:

- Differential and single-ended I/O standards
- 3.3-V, 64- and 32-bit, 66- and 33-MHz PCI compliance
- Joint Test Action Group (JTAG) boundary-scan test (BST) support
- Output drive strength control
- Weak pull-up resistors during configuration
- Slew-rate control
- Tri-state buffers
- Bus-hold circuitry
- Programmable pull-up resistors in user mode
- Programmable input and output delays
- Open-drain outputs
- DQ and DQS I/O pins

Cyclone device IOEs contain a bidirectional I/O buffer and three registers for complete embedded bidirectional single data rate transfer. Figure 2–27 shows the Cyclone IOE structure. The IOE contains one input register, one output register, and one output enable register. You can use the input registers for fast setup times and output registers for fast clock-to-output times. Additionally, you can use the output enable (OE) register for fast clock-to-output enable timing. The Quartus II software automatically duplicates a single OE register that controls multiple output or bidirectional pins. IOEs can be used as input, output, or bidirectional pins.

Figure 2–27. Cyclone IOE Structure

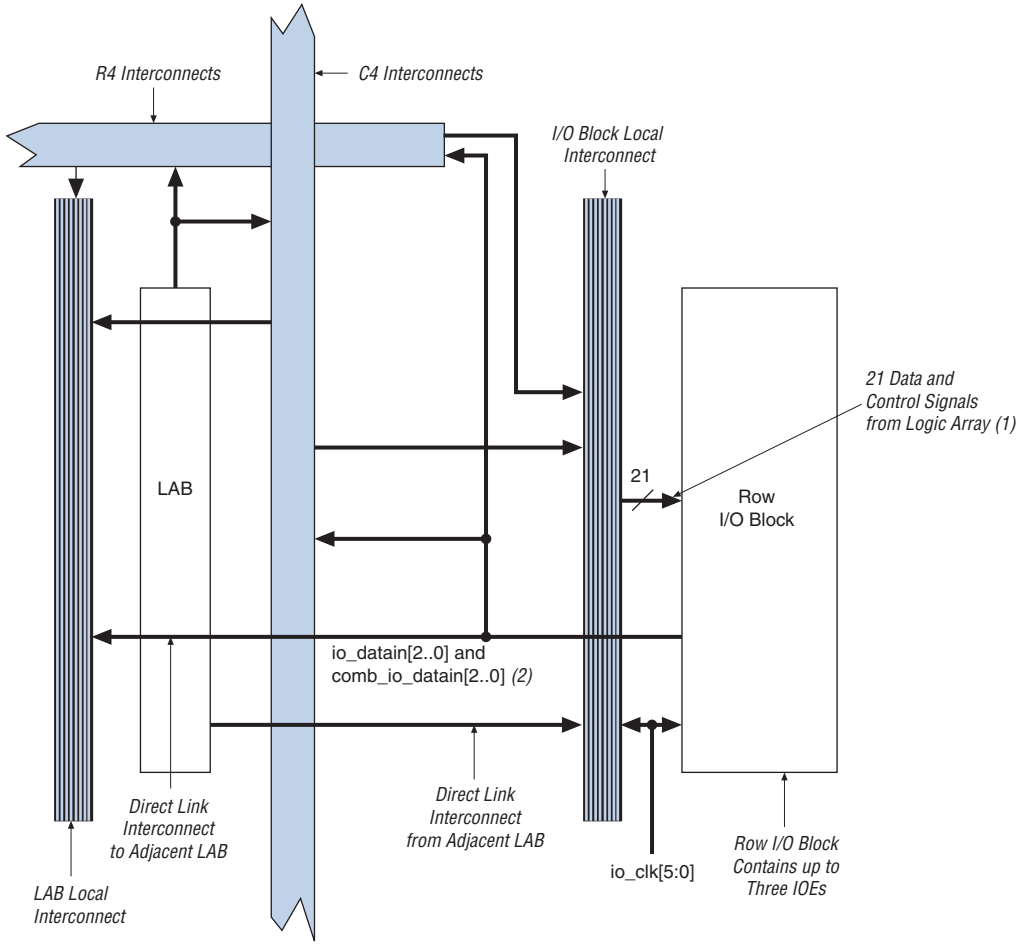


Note to Figure 2–27:

(1) There are two paths available for combinatorial inputs to the logic array. Each path contains a unique programmable delay chain.

The IOEs are located in I/O blocks around the periphery of the Cyclone device. There are up to three IOEs per row I/O block and up to three IOEs per column I/O block (column I/O blocks span two columns). The row I/O blocks drive row, column, or direct link interconnects. The column I/O blocks drive column interconnects. Figure 2–28 shows how a row I/O block connects to the logic array. Figure 2–29 shows how a column I/O block connects to the logic array.

Figure 2–28. Row I/O Block Connection to the Interconnect

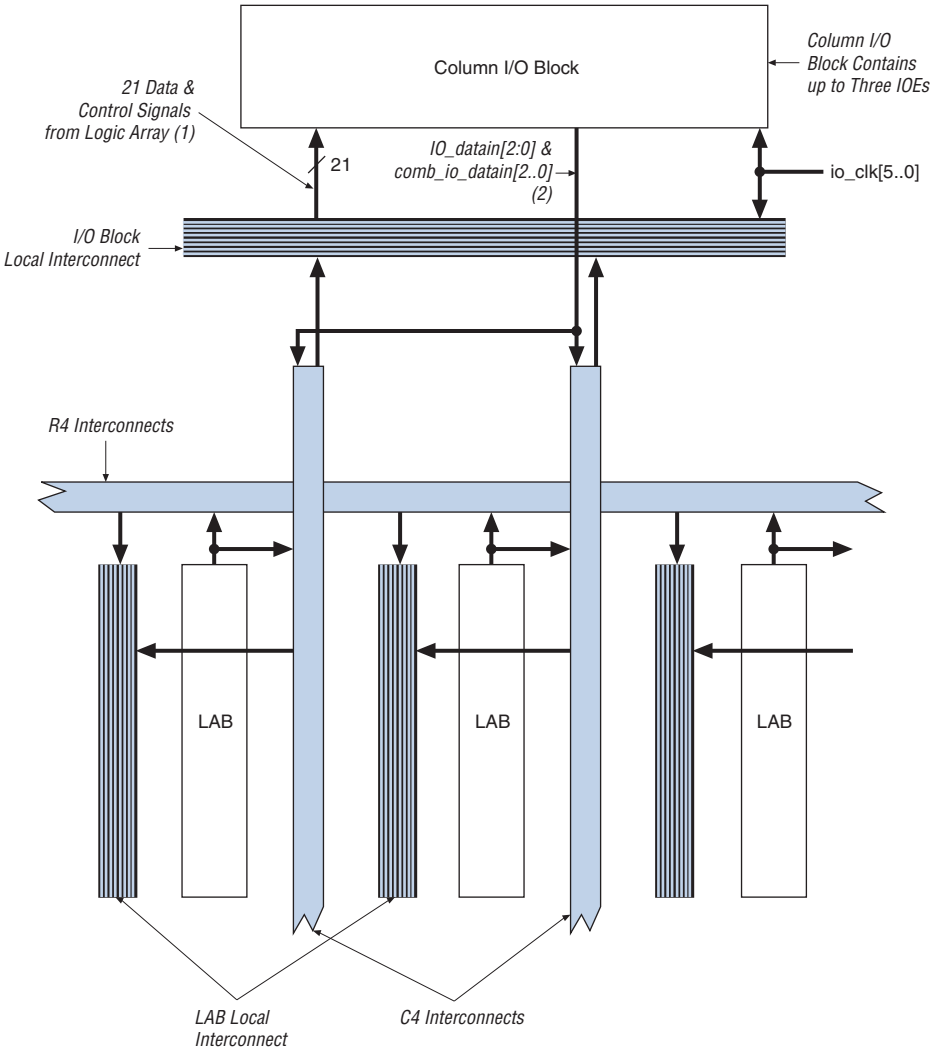


Notes to Figure 2–28:

(1) The 21 data and control signals consist of three data out lines, `io_dataout[2..0]`, three output enables, `io_coe[2..0]`, three input clock enables, `io_cce_in[2..0]`, three output clock enables, `io_cce_out[2..0]`, three clocks, `io_cclk[2..0]`, three asynchronous clear signals, `io_caclr[2..0]`, and three synchronous clear signals, `io_csclr[2..0]`.

(2) Each of the three IOEs in the row I/O block can have one `io_datain` input (combinatorial or registered) and one `comb_io_datain` (combinatorial) input.

Figure 2–29. Column I/O Block Connection to the Interconnect



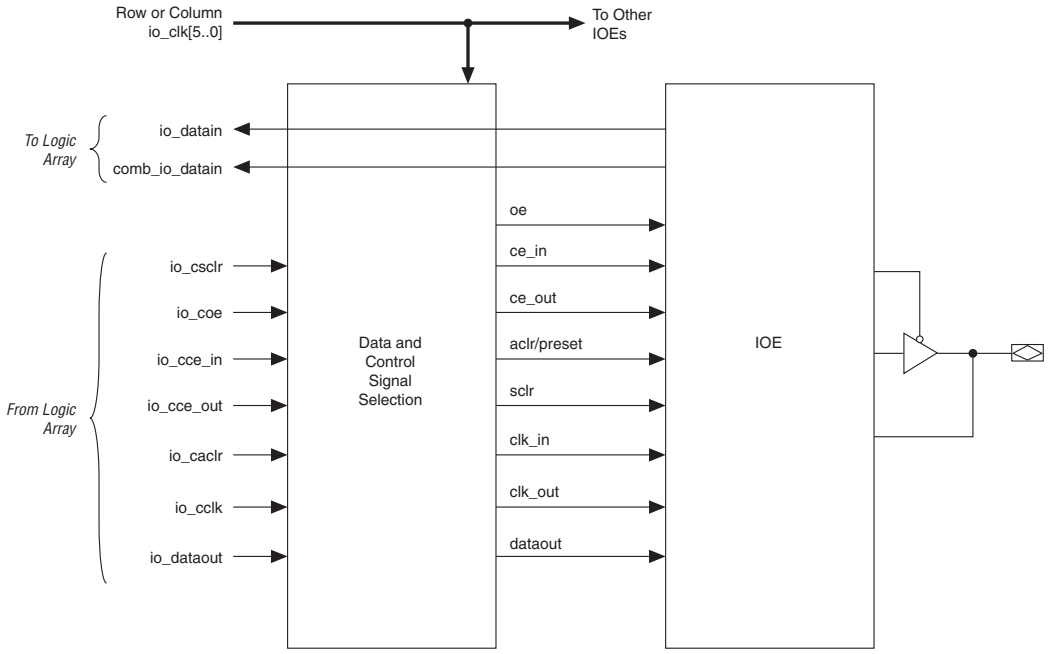
Notes to Figure 2–29:

(1) The 21 data and control signals consist of three data out lines, `io_dataout[2..0]`, three output enables, `io_coe[2..0]`, three input clock enables, `io_cce_in[2..0]`, three output clock enables, `io_cce_out[2..0]`, three clocks, `io_cclk[2..0]`, three asynchronous clear signals, `io_caclr[2..0]`, and three synchronous clear signals, `io_csclr[2..0]`.

(2) Each of the three IOEs in the column I/O block can have one `io_datain` input (combinatorial or registered) and one `comb_io_datain` (combinatorial) input.

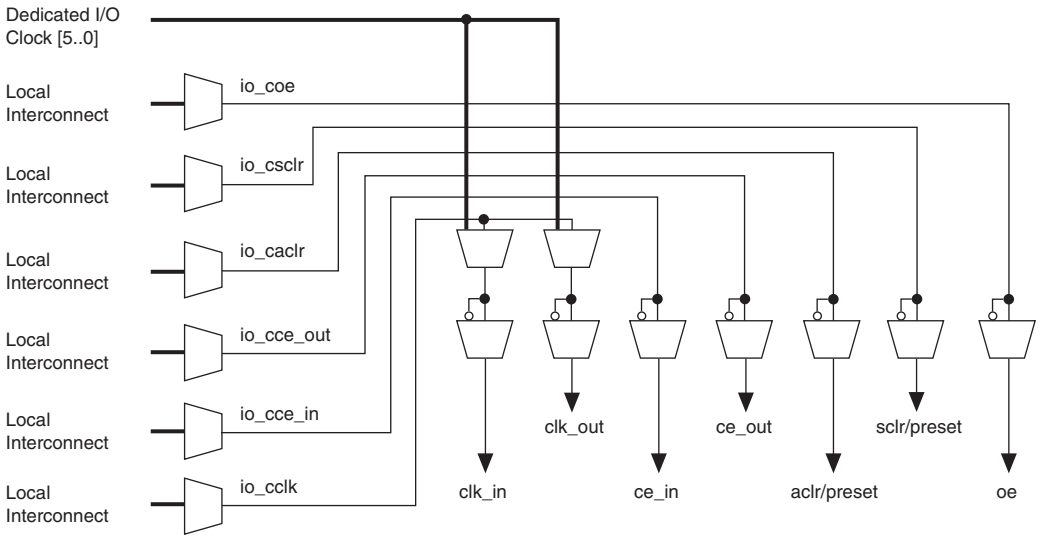
The pin's datain signals can drive the logic array. The logic array drives the control and data signals, providing a flexible routing resource. The row or column IOE clocks, `io_clk[5..0]`, provide a dedicated routing resource for low-skew, high-speed clocks. The global clock network generates the IOE clocks that feed the row or column I/O regions (see "Global Clock Network and Phase-Locked Loops" on page 2–29). Figure 2–30 illustrates the signal paths through the I/O block.

Figure 2–30. Signal Path through the I/O Block



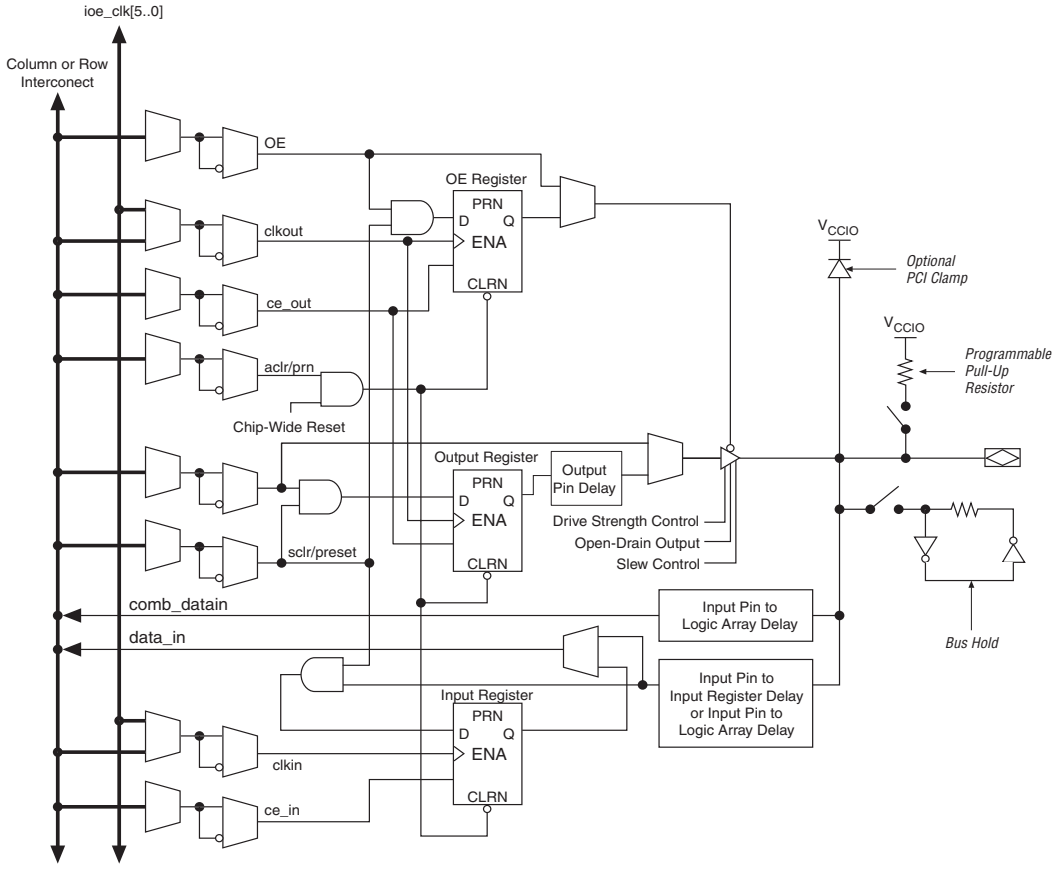
Each IOE contains its own control signal selection for the following control signals: `oe`, `ce_in`, `ce_out`, `aclr/preset`, `sclr/preset`, `clk_in`, and `clk_out`. Figure 2–31 illustrates the control signal selection.

Figure 2–31. Control Signal Selection per IOE



In normal bidirectional operation, you can use the input register for input data requiring fast setup times. The input register can have its own clock input and clock enable separate from the OE and output registers. The output register can be used for data requiring fast clock-to-output performance. The OE register is available for fast clock-to-output enable timing. The OE and output register share the same clock source and the same clock enable source from the local interconnect in the associated LAB, dedicated I/O clocks, or the column and row interconnects. Figure 2–32 shows the IOE in bidirectional configuration.

Figure 2–32. Cyclone IOE in Bidirectional I/O Configuration



The Cyclone device IOE includes programmable delays to ensure zero hold times, minimize setup times, or increase clock to output times.

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 combinatorial logic may not require the delay. Programmable delays decrease input-pin-to-logic-array and IOE input register delays. 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. Table 2–9 shows the programmable delays for Cyclone devices.

Table 2–9. Cyclone Programmable Delay Chain

Programmable Delays	Quartus II Logic Option
Input pin to logic array delay	Decrease input delay to internal cells
Input pin to input register delay	Decrease input delay to input registers
Output pin delay	Increase delay to output pin

There are two paths in the IOE for a combinatorial input to reach the logic array. Each of the two paths can have a different delay. This allows you adjust delays from the pin to internal LE registers that reside in two different areas of the device. The designer sets the two combinatorial input delays by selecting different delays for two different paths under the **Decrease input delay to internal cells** logic option in the Quartus II software. When the input signal requires two different delays for the combinatorial input, the input register in the IOE is no longer available.

The IOE registers in Cyclone devices share the same source for clear or preset. The designer can program preset or clear for each individual IOE. The designer 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 another device's active-low input upon power up. If one register in an IOE uses a preset or clear signal then all registers in the IOE must use that same signal if they require preset or clear. Additionally a synchronous reset signal is available to the designer for the IOE registers.

External RAM Interfacing

Cyclone devices support DDR SDRAM and FCRAM interfaces at up to 133 MHz through dedicated circuitry.

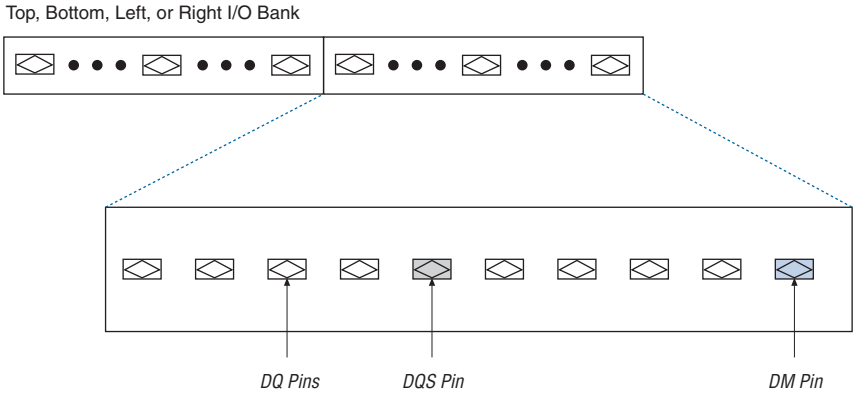
DDR SDRAM and FCRAM

Cyclone devices have dedicated circuitry for interfacing with DDR SDRAM. All I/O banks support DDR SDRAM and FCRAM I/O pins. However, the configuration input pins in bank 1 must operate at 2.5 V because the SSTL-2 V_{CCIO} level is 2.5 V. Additionally, the configuration

output pins (nSTATUS and CONF_DONE) and all the JTAG pins in I/O bank 3 must operate at 2.5 V because the V_{CCIO} level of SSTL-2 is 2.5 V. I/O banks 1, 2, 3, and 4 support DQS signals with DQ bus modes of × 8.

For × 8 mode, there are up to eight groups of programmable DQS and DQ pins, I/O banks 1, 2, 3, and 4 each have two groups in the 324-pin and 400-pin FineLine BGA packages. Each group consists of one DQS pin, a set of eight DQ pins, and one DM pin (see Figure 2–33). Each DQS pin drives the set of eight DQ pins within that group.

Figure 2–33. Cyclone Device DQ and DQS Groups in ×8 Mode *Note (1)*



Note to Figure 2–33:
(1) Each DQ group consists of one DQS pin, eight DQ pins, and one DM pin.

Table 2–10 shows the number of DQ pin groups per device.

Table 2–10. DQ Pin Groups (Part 1 of 2)

Device	Package	Number of × 8 DQ Pin Groups	Total DQ Pin Count
EP1C3	100-pin TQFP *(1)*	3	24
	144-pin TQFP	4	32
EP1C4	324-pin FineLine BGA	8	64
	400-pin FineLine BGA	8	64

Table 2–10. DQ Pin Groups (Part 2 of 2)

Device	Package	Number of ×8 DQ Pin Groups	Total DQ Pin Count
EP1C6	144-pin TQFP	4	32
	240-pin PQFP	4	32
	256-pin FineLine BGA	4	32
EP1C12	240-pin PQFP	4	32
	256-pin FineLine BGA	4	32
	324-pin FineLine BGA	8	64
EP1C20	324-pin FineLine BGA	8	64
	400-pin FineLine BGA	8	64

Note to Table 2–10:
(1) EP1C3 devices in the 100-pin TQFP package do not have any DQ pin groups in I/O bank 1.

A programmable delay chain on each DQS pin allows for either a 90° phase shift (for DDR SDRAM), or a 72° phase shift (for FCRAM) which automatically center-aligns input DQS synchronization signals within the data window of their corresponding DQ data signals. The phase-shifted DQS signals drive the global clock network. This global DQS signal clocks DQ signals on internal LE registers.

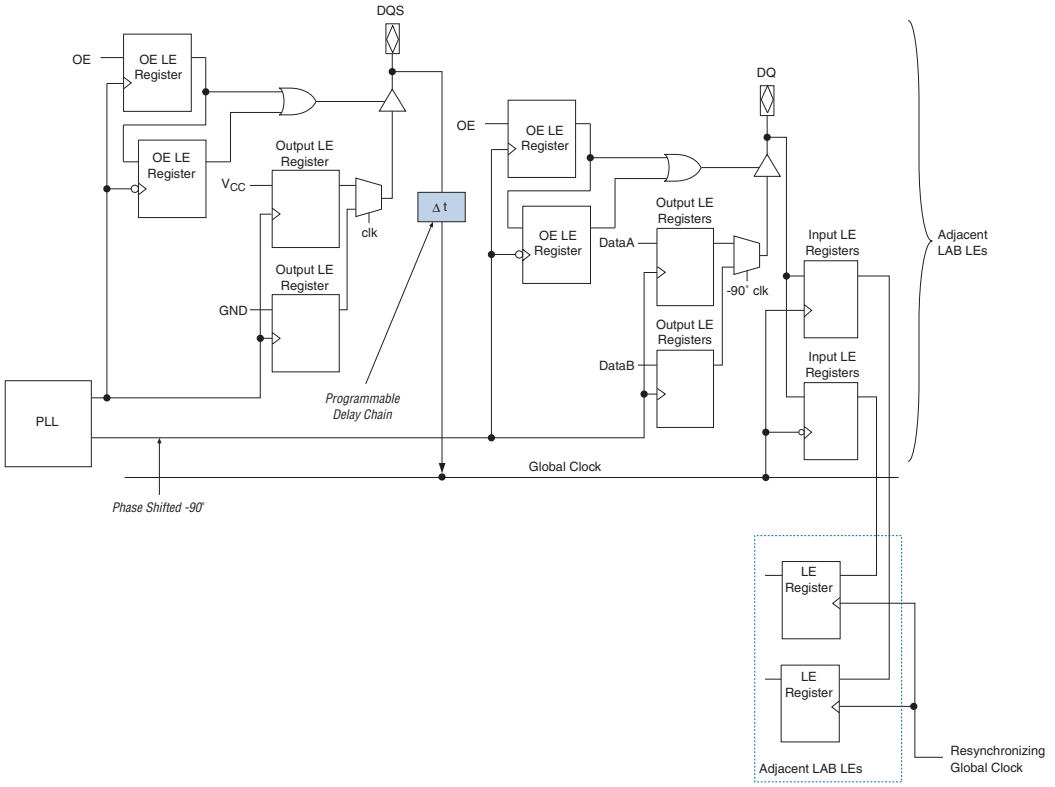
These DQS delay elements combine with the PLL's clocking and phase shift ability to provide a complete hardware solution for interfacing to high-speed memory.

The clock phase shift allows the PLL to clock the DQ output enable and output paths. The designer should use the following guidelines to meet 133 MHz performance for DDR SDRAM and FCRAM interfaces:

- The DQS signal must be in the middle of the DQ group it clocks
- Resynchronize the incoming data to the logic array clock using successive LE registers or FIFO buffers
- LE registers must be placed in the LAB adjacent to the DQ I/O pin column it is fed by

Figure 2–34 illustrates DDR SDRAM and FCRAM interfacing from the I/O through the dedicated circuitry to the logic array.

Figure 2–34. DDR SDRAM and FCRAM Interfacing



Programmable Drive Strength

The output buffer for each Cyclone device I/O pin has a programmable drive strength control for certain I/O standards. The LVTTL and LVCMOS standards have several levels of drive strength that the designer can control. SSTL-3 class I and II, and SSTL-2 class I and II support a minimum setting, the lowest drive strength that guarantees the I_{OH}/I_{OL}

of the standard. Using minimum settings provides signal slew rate control to reduce system noise and signal overshoot. Table 2–11 shows the possible settings for the I/O standards with drive strength control.

Table 2–11. Programmable Drive Strength *Note (1)*

I/O Standard	I_{OH}/I_{OL} Current Strength Setting (mA)
LVTTL (3.3 V)	4
	8
	12
	16
	24 *(2)*
LVCMOS (3.3 V)	2
	4
	8
	12 *(2)*
LVTTL (2.5 V)	2
	8
	12
	16 *(2)*
LVTTL (1.8 V)	2
	8
	12 *(2)*
LVCMOS (1.5 V)	2
	4
	8 *(2)*

Notes to Table 2–11:

(1) SSTL-3 class I and II, SSTL-2 class I and II, and 3.3-V PCI I/O Standards do not support programmable drive strength.

(2) This is the default current strength setting in the Quartus II software.

Open-Drain Output

Cyclone devices provide an optional open-drain (equivalent to an open-collector) output for each I/O pin. This open-drain output enables the device to provide system-level control signals (e.g., interrupt and write-enable signals) that can be asserted by any of several devices.

Slew-Rate Control

The output buffer for each Cyclone device I/O pin has a programmable output slew-rate control that can be configured for low noise or high-speed performance. A faster slew rate provides high-speed transitions for high-performance systems. However, these fast transitions may introduce noise transients into the system. A slow slew rate reduces system noise, but adds a nominal delay to rising and falling edges. Each I/O pin has an individual slew-rate control, allowing the designer to specify the slew rate on a pin-by-pin basis. The slew-rate control affects both the rising and falling edges.

Bus Hold

Each Cyclone device I/O pin provides an optional bus-hold feature. The bus-hold circuitry can hold the signal on an I/O pin at its last-driven state. Since 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 where noise can cause unintended high-frequency switching. The designer can select this feature individually for each I/O pin. The bus-hold output will drive no higher than V_{CCIO} to prevent overdriving signals. If the bus-hold feature is enabled, the device cannot use the programmable pull-up option. Disable the bus-hold feature when the I/O pin is configured for differential signals.

The bus-hold circuitry uses a resistor with a nominal resistance (RBH) of approximately 7 kΩ to pull the signal level to the last-driven state. Table 4–15 on page 4–6 gives the specific sustaining current for each V_{CCIO} voltage level driven through this resistor and overdrive current used to identify the next-driven input level.

The 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.

Programmable Pull-Up Resistor

Each Cyclone device I/O pin provides an optional programmable pull-up resistor during user mode. If the designer enables this feature for an I/O pin, the pull-up resistor (typically 25 kΩ) holds the output to the V_{CCIO} level of the output pin's bank. Dedicated clock pins do not have the optional programmable pull-up resistor.

Advanced I/O Standard Support

Cyclone device IOEs support the following I/O standards:

- 3.3-V LVTTL/LVCMOS
- 2.5-V LVTTL/LVCMOS
- 1.8-V LVTTL/LVCMOS
- 1.5-V LVCMOS
- 3.3-V PCI
- LVDS
- RSDS
- SSTL-2 class I and II
- SSTL-3 class I and II
- Differential SSTL-2 class II (on output clocks only)

Table 2–12 describes the I/O standards supported by Cyclone devices.

Table 2–12. Cyclone I/O Standards

I/O Standard	Type	Input Reference Voltage (V_{REF}) (V)	Output Supply Voltage (V_{CCIO}) (V)	Board Termination Voltage (V_{TT}) (V)
3.3-V LVTTL/LVCMOS	Single-ended	N/A	3.3	N/A
2.5-V LVTTL/LVCMOS	Single-ended	N/A	2.5	N/A
1.8-V LVTTL/LVCMOS	Single-ended	N/A	1.8	N/A
1.5-V LVCMOS	Single-ended	N/A	1.5	N/A
3.3-V PCI *(1)*	Single-ended	N/A	3.3	N/A
LVDS *(2)*	Differential	N/A	2.5	N/A
RSDS *(2)*	Differential	N/A	2.5	N/A
SSTL-2 class I and II	Voltage-referenced	1.25	2.5	1.25
SSTL-3 class I and II	Voltage-referenced	1.5	3.3	1.5
Differential SSTL-2 *(3)*	Differential	1.25	2.5	1.25

Notes to Table 2–12:

(1) There is no megafunction support for EP1C3 devices for the PCI compiler. However, EP1C3 devices support PCI by using the LVTTL 16-mA I/O standard and drive strength assignments in the Quartus II software. The device requires an external diode for PCI compliance.

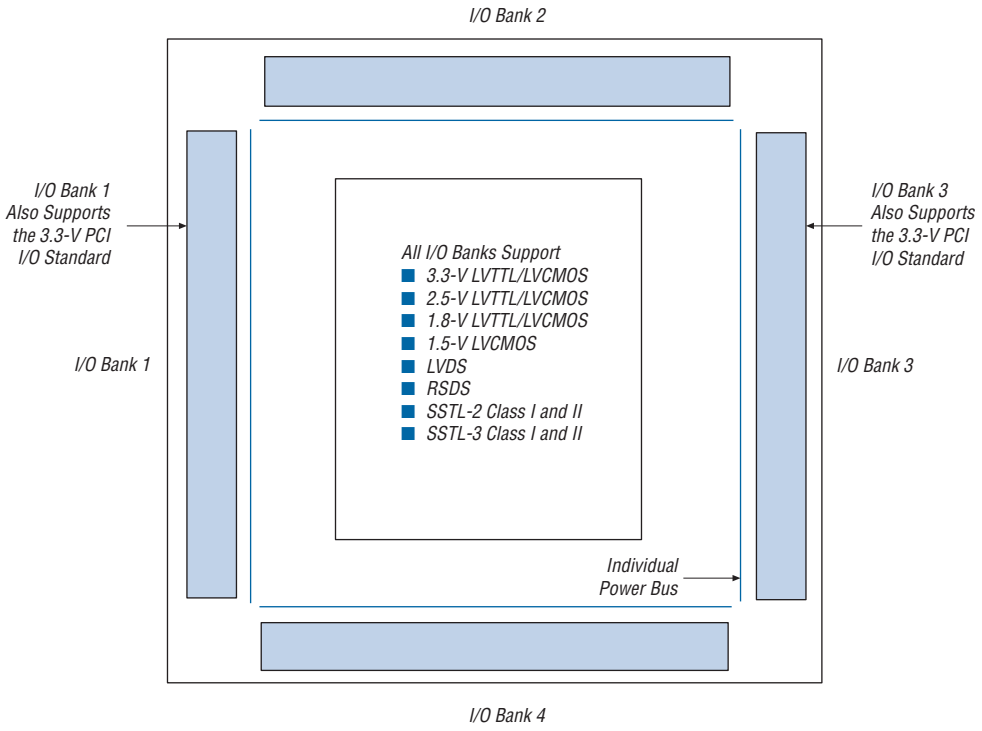
(2) EP1C3 devices in the 100-pin TQFP package do not support the LVDS and RSDS I/O standards.

(3) This I/O standard is only available on output clock pins (`PLL_OUT` pins). EP1C3 devices in the 100-pin package do not support this I/O standard as it does not have `PLL_OUT` pins.

Cyclone devices contain four I/O banks, as shown in Figure 2–35. I/O banks 1 and 3 support all the I/O standards listed in Table 2–12. I/O banks 2 and 4 support all the I/O standards listed in Table 2–12 except the 3.3-V PCI standard. I/O banks 2 and 4 contain dual-purpose DQS, DQ,

and DM pins to support a DDR SDRAM or FCRAM interface. I/O bank 1 can also support a DDR SDRAM or FCRAM interface, however, the configuration input pins in I/O bank 1 must operate at 2.5 V. I/O bank 3 can also support a DDR SDRAM or FCRAM interface, however, all the JTAG pins in I/O bank 3 must operate at 2.5 V.

Figure 2–35. Cyclone I/O Banks *Notes (1), (2)*



Notes to Figure 2–35:
(1) Figure 2–35 is a top view of the silicon die.
(2) Figure 2–35 is a graphic representation only. Refer to the pin list and the Quartus II software for exact pin locations.

Each I/O bank has its own VCCIO pins. A single device can support 1.5-V, 1.8-V, 2.5-V, and 3.3-V interfaces; each individual bank can support a different standard with different I/O voltages. Each bank also has dual-purpose VREF pins to support any one of the voltage-referenced standards (e.g., SSTL-3) independently. If an I/O bank does not use voltage-referenced standards, the V_{REF} pins are available as user I/O pins.

Each I/O bank can support multiple standards with the same V_{CCIO} for input and output pins. For example, when V_{CCIO} is 3.3-V, a bank can support LVTTL, LVCMOS, 3.3-V PCI, and SSTL-3 for inputs and outputs.

LVDS I/O Pins

A subset of pins in all four I/O banks supports LVDS interfacing. These dual-purpose LVDS pins require an external-resistor network at the transmitter channels in addition to 100-Ω termination resistors on receiver channels. These pins do not contain dedicated serialization or deserialization circuitry; therefore, internal logic performs serialization and deserialization functions.

Table 2–13 shows the total number of supported LVDS channels per device density.

Table 2–13. Cyclone Device LVDS Channels

Device	Pin Count	Number of LVDS Channels
EP1C3	100	*(1)*
	144	34
EP1C4	324	103
	400	129
EP1C6	144	29
	240	72
	256	72
EP1C12	240	66
	256	72
	324	103
EP1C20	324	95
	400	129

Note to Table 2–13:

(1) EP1C3 devices in the 100-pin TQFP package do not support the LVDS I/O standard.

MultiVolt I/O Interface

The Cyclone architecture supports the MultiVolt I/O interface feature, which allows Cyclone devices in all packages to interface with systems of different supply voltages. The devices have one set of V_{CC} pins for internal operation and input buffers (V_{CCINT}), and four sets for I/O output drivers (V_{CCIO}).

The Cyclone V_{CCINT} pins must always be connected to a 1.5-V power supply. If the V_{CCINT} level is 1.5 V, then input pins are 1.5-V, 1.8-V, 2.5-V, and 3.3-V tolerant. The V_{CCIO} pins can be connected to either a 1.5-V, 1.8-V, 2.5-V, or 3.3-V power supply, depending on the output requirements. The output levels are compatible with systems of the same voltage as the power supply (i.e., when V_{CCIO} pins are connected to a 1.5-V power supply, the output levels are compatible with 1.5-V systems). When V_{CCIO} pins are connected to a 3.3-V power supply, the output high is 3.3-V and is compatible with 3.3-V or 5.0-V systems. Table 2–14 summarizes Cyclone MultiVolt I/O support.

Table 2–14. Cyclone MultiVolt I/O Support *Note (1)*

V_{CCIO} (V)	Input Signal					Output Signal				
	1.5 V	1.8 V	2.5 V	3.3 V	5.0 V	1.5 V	1.8 V	2.5 V	3.3 V	5.0 V
1.5	✓	✓	✓ (2)	✓ (2)	—	✓	—	—	—	—
1.8	✓	✓	✓ (2)	✓ (2)	—	✓ (3)	✓	—	—	—
2.5	—	—	✓	✓	—	✓ (5)	✓ (5)	✓	—	—
3.3	—	—	✓ (4)	✓	✓ (6)	✓ (7)	✓ (7)	✓ (7)	✓	✓ (8)

Notes to Table 2–14:

(1) The PCI clamping diode must be disabled to drive an input with voltages higher than V_{CCIO}.
(2) When V_{CCIO} = 1.5-V or 1.8-V and a 2.5-V or 3.3-V input signal feeds an input pin, higher pin leakage current is expected. Turn on **Allow voltage overdrive for LVTTL / LVCMOS input pins** in the Assignments > Device > Device and Pin Options > Pin Placement tab when a device has this I/O combinations.
(3) When V_{CCIO} = 1.8-V, a Cyclone device can drive a 1.5-V device with 1.8-V tolerant inputs.
(4) When V_{CCIO} = 3.3-V and a 2.5-V input signal feeds an input pin, the V_{CCIO} supply current will be slightly larger than expected.
(5) When V_{CCIO} = 2.5-V, a Cyclone device can drive a 1.5-V or 1.8-V device with 2.5-V tolerant inputs.
(6) Cyclone devices can be 5.0-V tolerant with the use of an external resistor and the internal PCI clamp diode.
(7) When V_{CCIO} = 3.3-V, a Cyclone device can drive a 1.5-V, 1.8-V, or 2.5-V device with 3.3-V tolerant inputs.
(8) When V_{CCIO} = 3.3-V, a Cyclone device can drive a device with 5.0-V LVTTL inputs but not 5.0-V LVCMOS inputs.

Power Sequencing and Hot Socketing

Because Cyclone devices can be used in a mixed-voltage environment, they have been designed specifically to tolerate any possible power-up sequence. Therefore, the V_{CCIO} and V_{CCINT} power supplies may be powered in any order.

Signals can be driven into Cyclone devices before and during power up without damaging the device. In addition, Cyclone devices do not drive out during power up. Once operating conditions are reached and the device is configured, Cyclone devices operate as specified by the user.

Referenced Documents

This chapter references the following document:

- *Using PLLs in Cyclone Devices* chapter in the *Cyclone Device Handbook*

Document Revision History

Table 2–15 shows the revision history for this chapter.

Table 2–15. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.6	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.5	• Added document revision history. • Updated Figures 2–17, 2–18, 2–19, 2–20, 2–21, and 2–32.	—
August 2005 v1.4	Minor updates.	—
February 2005 v1.3	• Updated JTAG chain limits. Added test vector information. • Corrected Figure 2-12. • Added a note to Tables 2-17 through 2-21 regarding violating the setup or hold time.	—
October 2003 v1.2	• Updated phase shift information. • Added 64-bit PCI support information.	—
September 2003 v1.1	Updated LVDS data rates to 640 Mbps from 311 Mbps.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

3. Configuration and Testing

C51003-1.4

IEEE Std. 1149.1 (JTAG) Boundary Scan Support

All Cyclone® devices provide JTAG BST circuitry that complies with the IEEE Std. 1149.1a-1990 specification. JTAG boundary-scan testing can be performed either before or after, but not during configuration. Cyclone devices can also use the JTAG port for configuration together with either the Quartus® II software or hardware using either Jam Files (**.jam**) or Jam Byte-Code Files (**.jbc**).

Cyclone devices support reconfiguring the I/O standard settings on the IOE through the JTAG BST chain. The JTAG chain can update the I/O standard for all input and output pins any time before or during user mode. Designers can use this ability for JTAG testing before configuration when some of the Cyclone pins drive or receive from other devices on the board using voltage-referenced standards. Since the Cyclone device might not be configured before JTAG testing, the I/O pins might not be configured for appropriate electrical standards for chip-to-chip communication. Programming those I/O standards via JTAG allows designers to fully test I/O connection to other devices.

The JTAG pins support 1.5-V/1.8-V or 2.5-V/3.3-V I/O standards. The TDO pin voltage is determined by the V_{CCIO} of the bank where it resides. The bank V_{CCIO} selects whether the JTAG inputs are 1.5-V, 1.8-V, 2.5-V, or 3.3-V compatible.

Cyclone devices also use the JTAG port to monitor the operation of the device with the SignalTap® II embedded logic analyzer. Cyclone devices support the JTAG instructions shown in Table 3–1.

Table 3–1. Cyclone JTAG Instructions (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 0000	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.

Table 3–1. Cyclone JTAG Instructions (Part 2 of 2)

JTAG Instruction	Instruction Code	Description
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 0000 0110	Selects the IDCODE register and places it between TDI and TDO, allowing the IDCODE to be serially shifted out of TDO.
HIGHZ *(1)*	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 *(1)*	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 a Cyclone device via the JTAG port with a MasterBlaster™ or ByteBlasterMV™ download cable, or when using a Jam File or Jam Byte-Code 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	00 0000 1101	Allows configuration of I/O standards through the JTAG chain for JTAG testing. Can be executed before, after, or during configuration. Stops configuration if executed during configuration. Once issued, the CONFIG_IO instruction will hold nSTATUS low to reset the configuration device. nSTATUS is held low until the device is reconfigured.
SignalTap II instructions	—	Monitors internal device operation with the SignalTap II embedded logic analyzer.

Note to Table 3–1:
(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

In the Quartus II software, there is an Auto Usercode feature where you can choose to use the checksum value of a programming file as the JTAG user code. If selected, the checksum is automatically loaded to the USERCODE register. Choose Assignments > Device > Device and Pin Options > General. Turn on **Auto Usercode**.

The Cyclone device instruction register length is 10 bits and the USERCODE register length is 32 bits. Tables 3–2 and 3–3 show the boundary-scan register length and device IDCODE information for Cyclone devices.

Table 3–2. Cyclone Boundary-Scan Register Length

Device	Boundary-Scan Register Length
EP1C3	339
EP1C4	930
EP1C6	582
EP1C12	774
EP1C20	930

Table 3–3. 32-Bit Cyclone Device IDCODE

Device	IDCODE (32 bits) *(1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) *(2)*
EP1C3	0000	0010 0000 1000 0001	000 0110 1110	1
EP1C4	0000	0010 0000 1000 0101	000 0110 1110	1
EP1C6	0000	0010 0000 1000 0010	000 0110 1110	1
EP1C12	0000	0010 0000 1000 0011	000 0110 1110	1
EP1C20	0000	0010 0000 1000 0100	000 0110 1110	1

Notes to Table 3–3:

(1) The most significant bit (MSB) is on the left.

(2) The IDCODE's least significant bit (LSB) is always 1.

Figure 3–1 shows the timing requirements for the JTAG signals.

Figure 3–1. Cyclone JTAG Waveforms

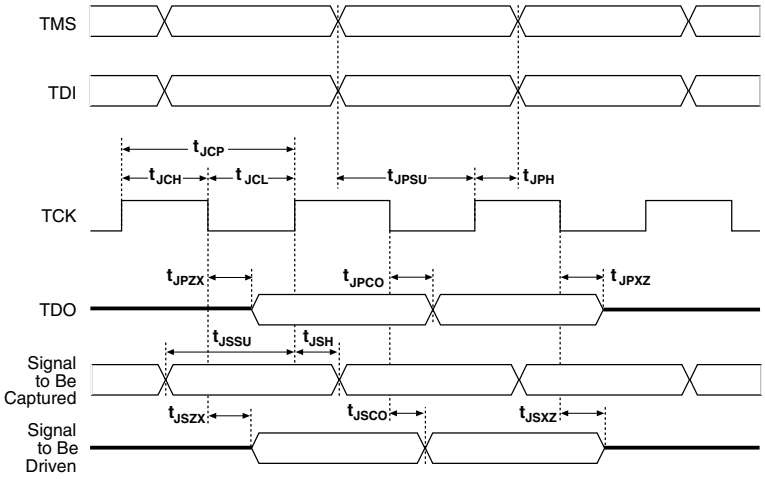


Table 3–4 shows the JTAG timing parameters and values for Cyclone devices.

Table 3–4. Cyclone JTAG Timing Parameters and Values

Symbol	Parameter	Min	Max	Unit
t_{JCP}	TCK clock period	100	—	ns
t_{JCH}	TCK clock high time	50	—	ns
t_{JCL}	TCK clock low time	50	—	ns
t_{JPSU}	JTAG port setup time	20	—	ns
t_{JPH}	JTAG port hold time	45	—	ns
t_{JPCO}	JTAG port clock to output	—	25	ns
t_{JPZX}	JTAG port high impedance to valid output	—	25	ns
t_{JPXZ}	JTAG port valid output to high impedance	—	25	ns
t_{JSSU}	Capture register setup time	20	—	ns
t_{JSH}	Capture register hold time	45	—	ns
t_{JSCO}	Update register clock to output	—	35	ns
t_{JSZX}	Update register high impedance to valid output	—	35	ns
t_{JSXZ}	Update register valid output to high impedance	—	35	ns



Cyclone devices must be within the first 8 devices in a JTAG chain. All of these devices have the same JTAG controller. If any of the Cyclone devices are in the 9th or after they will fail configuration. This does not affect the SignalTap® II logic analyzer.



For more information on JTAG, refer to the following documents:

- *AN 39: IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*
- *Jam Programming & Test Language Specification*

SignalTap II Embedded Logic Analyzer

Cyclone devices feature the SignalTap II embedded logic analyzer, which monitors design operation over a period of time through the IEEE Std. 1149.1 (JTAG) circuitry. A designer can analyze internal logic at speed without bringing internal signals to the I/O pins. This feature is particularly important for advanced packages, such as FineLine BGA packages, because it can be difficult to add a connection to a pin during the debugging process after a board is designed and manufactured.

Configuration

The logic, circuitry, and interconnects in the Cyclone architecture are configured with CMOS SRAM elements. Altera FPGAs are reconfigurable and every device is tested with a high coverage production test program so the designer does not have to perform fault testing and can instead focus on simulation and design verification.

Cyclone devices are configured at system power-up with data stored in an Altera configuration device or provided by a system controller. The Cyclone device's optimized interface allows the device to act as controller in an active serial configuration scheme with the new low-cost serial configuration device. Cyclone devices can be configured in under 120 ms using serial data at 20 MHz. The serial configuration device can be programmed via the ByteBlaster II download cable, the Altera Programming Unit (APU), or third-party programmers.

In addition to the new low-cost serial configuration device, Altera offers in-system programmability (ISP)-capable configuration devices that can configure Cyclone devices via a serial data stream. The interface also enables microprocessors to treat Cyclone devices as memory and configure them by writing to a virtual memory location, making reconfiguration easy. After a Cyclone device has been configured, it can be reconfigured in-circuit by resetting the device and loading new data. Real-time changes can be made during system operation, enabling innovative reconfigurable computing applications.

Operating Modes

The Cyclone architecture uses SRAM configuration elements that require configuration data to be loaded each time the circuit powers up. The process of physically loading the SRAM data into the device is called configuration. During initialization, which occurs immediately after configuration, the device resets registers, enables I/O pins, and begins to operate as a logic device. Together, the configuration and initialization processes are called command mode. Normal device operation is called user mode.

SRAM configuration elements allow Cyclone devices to be reconfigured in-circuit by loading new configuration data into the device. With real-time reconfiguration, the device is forced into command mode with a device pin. The configuration process loads different configuration data, reinitializes the device, and resumes user-mode operation. Designers can perform in-field upgrades by distributing new configuration files either within the system or remotely.

A built-in weak pull-up resistor pulls all user I/O pins to V_{CCIO} before and during device configuration.

The configuration pins support 1.5-V/1.8-V or 2.5-V/3.3-V I/O standards. The voltage level of the configuration output pins is determined by the V_{CCIO} of the bank where the pins reside. The bank V_{CCIO} selects whether the configuration inputs are 1.5-V, 1.8-V, 2.5-V, or 3.3-V compatible.

Configuration Schemes

Designers can load the configuration data for a Cyclone device with one of three configuration schemes (see Table 3–5), chosen on the basis of the target application. Designers can use a configuration device, intelligent controller, or the JTAG port to configure a Cyclone device. A low-cost configuration device can automatically configure a Cyclone device at system power-up.

Multiple Cyclone devices can be configured in any of the three configuration schemes by connecting the configuration enable (nCE) and configuration enable output (nCEO) pins on each device.

Table 3–5. Data Sources for Configuration

Configuration Scheme	Data Source
Active serial	Low-cost serial configuration device
Passive serial (PS)	Enhanced or EPC2 configuration device, MasterBlaster or ByteBlasterMV download cable, or serial data source
JTAG	MasterBlaster or ByteBlasterMV download cable or a microprocessor with a Jam or JBC file

Referenced Documents

This chapter references the following documents:

- *AN 39: IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*
- *Jam Programming & Test Language Specification*

Document Revision History

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

Table 3–6. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.4	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.3	• Added document revision history. • Updated handpara note below Table 3–4.	—
August 2005 V1.2	Minor updates.	—
February 2005 V1.1	Updated JTAG chain limits. Added information concerning test vectors.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

4. DC and Switching Characteristics

C51004-1.7

Operating Conditions

Cyclone® devices are offered in both commercial, industrial, and extended temperature grades. However, industrial-grade and extended-temperature-grade devices may have limited speed-grade availability.

Tables 4–1 through 4–16 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for Cyclone devices.

Table 4–1. Cyclone Device Absolute Maximum Ratings *Notes (1), (2)*

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage	With respect to ground *(3)*	–0.5	2.4	V
V_{CCIO}			–0.5	4.6	V
V_{CCA}	Supply voltage	With respect to ground *(3)*	–0.5	2.4	V
V_I	DC input voltage		–0.5	4.6	V
I_{OUT}	DC output current, per pin		–25	25	mA
T_{STG}	Storage temperature	No bias	–65	150	°C
T_{AMB}	Ambient temperature	Under bias	–65	135	°C
T_J	Junction temperature	BGA packages under bias	—	135	°C

Table 4–2. Cyclone Device Recommended Operating Conditions (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	*(4)*	1.425	1.575	V
V_{CCIO}	Supply voltage for output buffers, 3.3-V operation	*(4)*	3.00	3.60	V
	Supply voltage for output buffers, 2.5-V operation	*(4)*	2.375	2.625	V
	Supply voltage for output buffers, 1.8-V operation	*(4)*	1.71	1.89	V
	Supply voltage for output buffers, 1.5-V operation	*(4)*	1.4	1.6	V
V_I	Input voltage	*(3)*, *(5)*	–0.5	4.1	V

Table 4–2. Cyclone Device Recommended Operating Conditions (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_O	Output voltage		0	V_{CCIO}	V
T_J	Operating junction temperature	For commercial use	0	85	° C
		For industrial use	–40	100	° C
		For extended-temperature use	–40	125	° C

Table 4–3. Cyclone Device DC Operating Conditions *Note (6)*

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
I_I	Input pin leakage current	$V_I = V_{CCIOmax}$ to 0 V *(8)*	–10	—	10	µA
I_{OZ}	Tri-stated I/O pin leakage current	$V_O = V_{CCIOmax}$ to 0 V *(8)*	–10	—	10	µA
I_{CC0}	V_{CC} supply current (standby) (All M4K blocks in power-down mode) *(7)*	EP1C3	—	4	—	mA
		EP1C4	—	6	—	mA
		EP1C6	—	6	—	mA
		EP1C12	—	8	—	mA
		EP1C20	—	12	—	mA
R_{CONF} *(9)*	Value of I/O pin pull-up resistor before and during configuration	V_I = 0 V; V_{CCI0} = 3.3 V	15	25	50	kΩ
		V_I = 0 V; V_{CCI0} = 2.5 V	20	45	70	kΩ
		V_I = 0 V; V_{CCI0} = 1.8 V	30	65	100	kΩ
		V_I = 0 V; V_{CCI0} = 1.5 V	50	100	150	kΩ
	Recommended value of I/O pin external pull-down resistor before and during configuration	—	—	1	2	kΩ

Table 4–4. LVTTL Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage	—	3.0	3.6	V
V_{IH}	High-level input voltage	—	1.7	4.1	V
V_{IL}	Low-level input voltage	—	–0.5	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –4 to –24 mA *(11)*	2.4	—	V
V_{OL}	Low-level output voltage	I_{OL} = 4 to 24 mA *(11)*	—	0.45	V

Table 4–5. LVCMOS Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage	—	3.0	3.6	V
V_{IH}	High-level input voltage	—	1.7	4.1	V
V_{IL}	Low-level input voltage	—	–0.5	0.7	V
V_{OH}	High-level output voltage	V_{CCIO} = 3.0, I_{OH} = –0.1 mA	V_{CCIO} – 0.2	—	V
V_{OL}	Low-level output voltage	V_{CCIO} = 3.0, I_{OL} = 0.1 mA	—	0.2	V

Table 4–6. 2.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage	—	2.375	2.625	V
V_{IH}	High-level input voltage	—	1.7	4.1	V
V_{IL}	Low-level input voltage	—	–0.5	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –0.1 mA	2.1	—	V
		I_{OH} = –1 mA	2.0	—	V
		I_{OH} = –2 to –16 mA *(11)*	1.7	—	V
V_{OL}	Low-level output voltage	I_{OL} = 0.1 mA	—	0.2	V
		I_{OH} = 1 mA	—	0.4	V
		I_{OH} = 2 to 16 mA *(11)*	—	0.7	V

Table 4–7. 1.8-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage	—	1.65	1.95	V
V_{IH}	High-level input voltage	—	0.65 × V_{CCIO}	2.25 *(12)*	V
V_{IL}	Low-level input voltage	—	–0.3	0.35 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 to –8 mA *(11)*	V_{CCIO} – 0.45	—	V
V_{OL}	Low-level output voltage	I_{OL} = 2 to 8 mA *(11)*	—	0.45	V

Table 4–8. 1.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage	—	1.4	1.6	V
V_{IH}	High-level input voltage	—	0.65 × V_{CCIO}	V_{CCIO} + 0.3 *(12)*	V
V_{IL}	Low-level input voltage	—	–0.3	0.35 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 mA *(11)*	0.75 × V_{CCIO}	—	V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(11)*	—	0.25 × V_{CCIO}	V

Table 4–9. 2.5-V LVDS I/O Specifications *Note (13)*

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage	—	2.375	2.5	2.625	V
V_{OD}	Differential output voltage	R_L = 100 Ω	250	—	550	mV
Δ V_{OD}	Change in V_{OD} between high and low	R_L = 100 Ω	—	—	50	mV
V_{OS}	Output offset voltage	R_L = 100 Ω	1.125	1.25	1.375	V
Δ V_{OS}	Change in V_{OS} between high and low	R_L = 100 Ω	—	—	50	mV
V_{TH}	Differential input threshold	V_{CM} = 1.2 V	–100	—	100	mV
V_{IN}	Receiver input voltage range	—	0.0	—	2.4	V
R_L	Receiver differential input resistor	—	90	100	110	Ω

Table 4–10. 3.3-V PCI Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage	—	3.0	3.3	3.6	V
V_{IH}	High-level input voltage	—	0.5 × V_{CCIO}	—	V_{CCIO} + 0.5	V
V_{IL}	Low-level input voltage	—	–0.5	—	0.3 × V_{CCIO}	V

Table 4–10. 3.3-V PCI Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{OH}	High-level output voltage	I_{OUT} = –500 µA	0.9 × V_{CCIO}	—	—	V
V_{OL}	Low-level output voltage	I_{OUT} = 1,500 µA	—	—	0.1 × V_{CCIO}	V

Table 4–11. SSTL-2 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage	—	2.375	2.5	2.625	V
V_{TT}	Termination voltage	—	V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage	—	1.15	1.25	1.35	V
V_{IH}	High-level input voltage	—	V_{REF} + 0.18	—	3.0	V
V_{IL}	Low-level input voltage	—	–0.3	—	V_{REF} – 0.18	V
V_{OH}	High-level output voltage	I_{OH} = –8.1 mA *(11)*	V_{TT} + 0.57	—	—	V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(11)*	—	—	V_{TT} – 0.57	V

Table 4–12. SSTL-2 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage	—	2.3	2.5	2.7	V
V_{TT}	Termination voltage	—	V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage	—	1.15	1.25	1.35	V
V_{IH}	High-level input voltage	—	V_{REF} + 0.18	—	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage	—	–0.3	—	V_{REF} – 0.18	V
V_{OH}	High-level output voltage	I_{OH} = –16.4 mA *(11)*	V_{TT} + 0.76	—	—	V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(11)*	—	—	V_{TT} – 0.76	V

Table 4–13. SSTL-3 Class I Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage	—	3.0	3.3	3.6	V
V_{TT}	Termination voltage	—	V_{REF} – 0.05	V_{REF}	V_{REF} + 0.05	V

Table 4–13. SSTL-3 Class I Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{REF}	Reference voltage	—	1.3	1.5	1.7	V
V_{IH}	High-level input voltage	—	V_{REF} + 0.2	—	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage	—	–0.3	—	V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA *(11)*	V_{TT} + 0.6	—	—	V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA *(11)*	—	—	V_{TT} – 0.6	V

Table 4–14. SSTL-3 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage	—	3.0	3.3	3.6	V
V_{TT}	Termination voltage	—	V_{REF} – 0.05	V_{REF}	V_{REF} + 0.05	V
V_{REF}	Reference voltage	—	1.3	1.5	1.7	V
V_{IH}	High-level input voltage	—	V_{REF} + 0.2	—	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage	—	–0.3	—	V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = –16 mA *(11)*	V_{TT} + 0.8	—	—	V
V_{OL}	Low-level output voltage	I_{OL} = 16 mA *(11)*	—	—	V_{TT} – 0.8	V

Table 4–15. Bus Hold Parameters

Parameter	Conditions	V_{CCIO} Level								Unit
		1.5 V		1.8 V		2.5 V		3.3 V		
		Min	Max	Min	Max	Min	Max	Min	Max	
Low sustaining current	$V_{IN} > V_{IL}$ (maximum)	—	—	30	—	50	—	70	—	µA
High sustaining current	$V_{IN} < V_{IH}$ (minimum)	—	—	–30	—	–50	—	–70	—	µA
Low overdrive current	0 V < V_{IN} < V_{CCIO}	—	—	—	200	—	300	—	500	µA
High overdrive current	0 V < V_{IN} < V_{CCIO}	—	—	—	–200	—	–300	—	–500	µA

Table 4–16. Cyclone Device Capacitance *Note (14)*

Symbol	Parameter	Typical	Unit
C_{IO}	Input capacitance for user I/O pin	4.0	pF
C_{LVDS}	Input capacitance for dual-purpose LVDS/user I/O pin	4.7	pF
C_{VREF}	Input capacitance for dual-purpose V_{REF}/user I/O pin.	12.0	pF
C_{DPCLK}	Input capacitance for dual-purpose DPCLK/user I/O pin.	4.4	pF
C_{CLK}	Input capacitance for CLK pin.	4.7	pF

Notes to Tables 4–1 through 4–16:

(1) Refer to the *Operating Requirements for Altera Devices Data Sheet*.

(2) Conditions beyond those listed in Table 4–1 may cause permanent damage to a device. Additionally, device operation at the absolute maximum ratings for extended periods of time may have adverse affects on the device.

(3) Minimum DC input is –0.5 V. During transitions, the inputs may undershoot to –2.0 V or overshoot to 4.6 V for input currents less than 100 mA and periods shorter than 20 ns.

(4) Maximum V_{CC} rise time is 100 ms, and V_{CC} must rise monotonically.

(5) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT} and V_{CCIO} are powered.

(6) Typical values are for T_A = 25° C, V_{CCINT} = 1.5 V, and V_{CCIO} = 1.5 V, 1.8 V, 2.5 V, and 3.3 V.

(7) V_I = ground, no load, no toggling inputs.

(8) This value is specified for normal device operation. The value may vary during power-up. This applies for all V_{CCIO} settings (3.3, 2.5, 1.8, and 1.5 V).

(9) R_{CONF} is the measured value of internal pull-up resistance when the I/O pin is tied directly to GND. R_{CONF} value will be lower if an external source drives the pin higher than V_{CCIO}.

(10) Pin pull-up resistance values will lower if an external source drives the pin higher than V_{CCIO}.

(11) Drive strength is programmable according to values in *Cyclone Architecture* chapter in the *Cyclone Device Handbook*.

(12) Overdrive is possible when a 1.5 V or 1.8 V and a 2.5 V or 3.3 V input signal feeds an input pin. Turn on "Allow voltage overdrive" for LVTTL/LVCMOS input pins in the Assignments > Device > Device and Pin Options > Pin Placement tab when a device has this I/O combination. However, higher leakage current is expected.

(13) The Cyclone LVDS interface requires a resistor network outside of the transmitter channels.

(14) Capacitance is sample-tested only. Capacitance is measured using time-domain reflections (TDR). Measurement accuracy is within ±0.5 pF.

Power Consumption

Designers can use the Altera web Early Power Estimator to estimate the device power.

Cyclone devices require a certain amount of power-up current to successfully power up because of the nature of the leading-edge process on which they are fabricated. Table 4–17 shows the maximum power-up current required to power up a Cyclone device.

Table 4–17. Cyclone Maximum Power-Up Current (I_{CCINT}) Requirements (In-Rush Current)

Device	Commercial Specification	Industrial Specification	Unit
EP1C3	150	180	mA
EP1C4	150	180	mA
EP1C6	175	210	mA
EP1C12	300	360	mA
EP1C20	500	600	mA

Notes to Table 4–17:

(1) The Cyclone devices (except for the EP1C20 device) meet the power up specification for Mini PCI.

(2) The lot codes 9G0082 to 9G2999, or 9G3109 and later comply to the specifications in Table 4–17 and meet the Mini PCI specification. Lot codes appear at the top of the device.

(3) The lot codes 9H0004 to 9H29999, or 9H3014 and later comply to the specifications in this table and meet the Mini PCI specification. Lot codes appear at the top of the device.

Designers should select power supplies and regulators that can supply this amount of current when designing with Cyclone devices. This specification is for commercial operating conditions. Measurements were performed with an isolated Cyclone device on the board. Decoupling capacitors were not used in this measurement. To factor in the current for decoupling capacitors, sum up the current for each capacitor using the following equation:

$$I = C\ (dV/dt)$$

The exact amount of current that is consumed varies according to the process, temperature, and power ramp rate. If the power supply or regulator can supply more current than required, the Cyclone device may consume more current than the maximum current specified in Table 4–17. However, the device does not require any more current to successfully power up than what is listed in Table 4–17.

The duration of the I_{CCINT} power-up requirement depends on the V_{CCINT} voltage supply rise time. The power-up current consumption drops when the V_{CCINT} supply reaches approximately 0.75 V. For example, if the V_{CCINT} rise time has a linear rise of 15 ms, the current consumption spike drops by 7.5 ms.

Typically, the user-mode current during device operation is lower than the power-up current in Table 4–17. Altera recommends using the Cyclone Power Calculator, available on the Altera web site, to estimate the user-mode I_{CCINT} consumption and then select power supplies or regulators based on the higher value.

Timing Model

The DirectDrive technology and MultiTrack interconnect ensure predictable performance, accurate simulation, and accurate timing analysis across all Cyclone device densities and speed grades. This section describes and specifies the performance, internal, external, and PLL timing specifications.

All specifications are representative of worst-case supply voltage and junction temperature conditions.

Preliminary and Final Timing

Timing models can have either preliminary or final status. The Quartus® II software issues an informational message during the design compilation if the timing models are preliminary. Table 4–18 shows the status of the Cyclone device timing models.

Preliminary status means the timing model is subject to change. Initially, timing numbers are created using simulation results, process data, and other known parameters. These tests are used to make the preliminary numbers as close to the actual timing parameters as possible.

Final timing numbers are based on actual device operation and testing. These numbers reflect the actual performance of the device under worst-case voltage and junction temperature conditions.

Table 4–18. Cyclone Device Timing Model Status

Device	Preliminary	Final
EP1C3	—	✓
EP1C4	—	✓
EP1C6	—	✓
EP1C12	—	✓
EP1C20	—	✓

Performance

The maximum internal logic array clock tree frequency is limited to the specifications shown in Table 4–19.

Table 4–19. Clock Tree Maximum Performance Specification

Parameter	Definition	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Units
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Clock tree f_{MAX}	Maximum frequency that the clock tree can support for clocking registered logic	—	—	405	—	—	320	—	—	275	MHz

Table 4–20 shows the Cyclone device performance for some common designs. All performance values were obtained with the Quartus II software compilation of library of parameterized modules (LPM) functions or megafunctions. These performance values are based on EP1C6 devices in 144-pin TQFP packages.

Table 4–20. Cyclone Device Performance

Resource Used	Design Size and Function	Mode	Resources Used			Performance		
			LEs	M4K Memory Bits	M4K Memory Blocks	-6 Speed Grade (MHz)	-7 Speed Grade (MHz)	-8 Speed Grade (MHz)
LE	16-to-1 multiplexer	—	21	—	—	405.00	320.00	275.00
	32-to-1 multiplexer	—	44	—	—	317.36	284.98	260.15
	16-bit counter	—	16	—	—	405.00	320.00	275.00
	64-bit counter *(1)*	—	66	—	—	208.99	181.98	160.75

Table 4–20. Cyclone Device Performance

Resource Used	Design Size and Function	Mode	Resources Used			Performance		
			LEs	M4K Memory Bits	M4K Memory Blocks	-6 Speed Grade (MHz)	-7 Speed Grade (MHz)	-8 Speed Grade (MHz)
M4K memory block	RAM 128 × 36 bit	Single port	—	4,608	1	256.00	222.67	197.01
	RAM 128 × 36 bit	Simple dual-port mode	—	4,608	1	255.95	222.67	196.97
	RAM 256 × 18 bit	True dual-port mode	—	4,608	1	255.95	222.67	196.97
	FIFO 128 × 36 bit	—	40	4,608	1	256.02	222.67	197.01
	Shift register 9 × 4 × 128	Shift register	11	4,536	1	255.95	222.67	196.97

Note to Table 4–20:
(1) The performance numbers for this function are from an EP1C6 device in a 240-pin PQFP package.

Internal Timing Parameters

Internal timing parameters are specified on a speed grade basis independent of device density. Tables 4–21 through 4–24 describe the Cyclone device internal timing microparameters for LEs, IOEs, M4K memory structures, and MultiTrack interconnects.

Table 4–21. LE Internal Timing Microparameter Descriptions

Symbol	Parameter
t_{SU}	LE register setup time before clock
t_{H}	LE register hold time after clock
t_{CO}	LE register clock-to-output delay
t_{LUT}	LE combinatorial LUT delay for data-in to data-out
t_{CLR}	Minimum clear pulse width
t_{PRE}	Minimum preset pulse width
t_{CLKHL}	Minimum clock high or low time

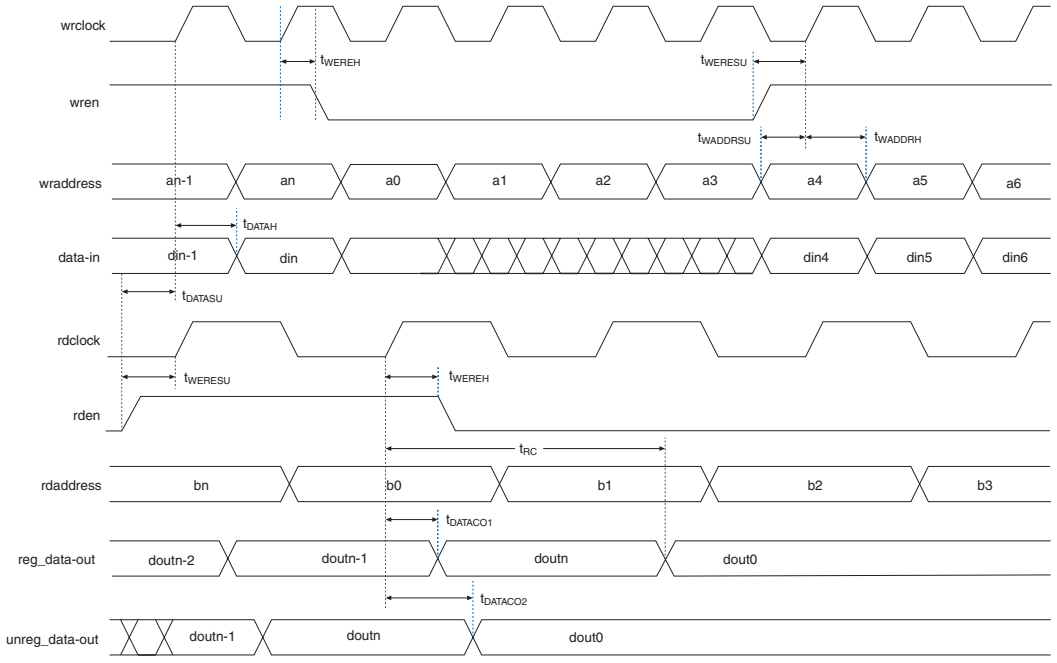
Table 4–22. IOE Internal Timing Microparameter Descriptions	
Symbol	**Parameter**
t_{SU}	IOE input and output register setup time before clock
t_{H}	IOE input and output register hold time after clock
t_{CO}	IOE input and output register clock-to-output delay
$t_{PIN2COMBOUT_R}$	Row input pin to IOE combinatorial output
$t_{PIN2COMBOUT_C}$	Column input pin to IOE combinatorial output
$t_{COMBIN2PIN_R}$	Row IOE data input to combinatorial output pin
$t_{COMBIN2PIN_C}$	Column IOE data input to combinatorial output pin
t_{CLR}	Minimum clear pulse width
t_{PRE}	Minimum preset pulse width
t_{CLKHL}	Minimum clock high or low time

Table 4–23. M4K Block Internal Timing Microparameter Descriptions	
Symbol	**Parameter**
t_{M4KRC}	Synchronous read cycle time
t_{M4KWC}	Synchronous write cycle time
$t_{M4KWERESU}$	Write or read enable setup time before clock
$t_{M4KWEREH}$	Write or read enable hold time after clock
$t_{M4KBESU}$	Byte enable setup time before clock
t_{M4KBEH}	Byte enable hold time after clock
$t_{M4KDATAASU}$	A port data setup time before clock
$t_{M4KDATAAH}$	A port data hold time after clock
$t_{M4KADDRASU}$	A port address setup time before clock
$t_{M4KADDRAH}$	A port address hold time after clock
$t_{M4KDATABSU}$	B port data setup time before clock
$t_{M4KDATABH}$	B port data hold time after clock
$t_{M4KADDRBSU}$	B port address setup time before clock
$t_{M4KADDRBH}$	B port address hold time after clock
$t_{M4KDATACO1}$	Clock-to-output delay when using output registers
$t_{M4KDATACO2}$	Clock-to-output delay without output registers
$t_{M4KCLKHL}$	Minimum clock high or low time
t_{M4KCLR}	Minimum clear pulse width

Table 4–24. Routing Delay Internal Timing Microparameter Descriptions	
Symbol	**Parameter**
t_{R4}	Delay for an R4 line with average loading; covers a distance of four LAB columns
t_{C4}	Delay for an C4 line with average loading; covers a distance of four LAB rows
t_{LOCAL}	Local interconnect delay

Figure 4–1 shows the memory waveforms for the M4K timing parameters shown in Table 4–23.

Figure 4–1. Dual-Port RAM Timing Microparameter Waveform



Internal timing parameters are specified on a speed grade basis independent of device density. Tables 4–25 through 4–28 show the internal timing microparameters for LEs, IOEs, TriMatrix memory structures, DSP blocks, and MultiTrack interconnects.

Table 4–25. LE Internal Timing Microparameters

Symbol	-6		-7		-8		Unit
	Min	Max	Min	Max	Min	Max	
t_{SU}	29	—	33	—	37	—	ps
t_{H}	12	—	13	—	15	—	ps
t_{CO}	—	173	—	198	—	224	ps
t_{LUT}	—	454	—	522	—	590	ps
t_{CLR}	129	—	148	—	167	—	ps
t_{PRE}	129	—	148	—	167	—	ps
t_{CLKHL}	1,234	—	1,562	—	1,818	—	ps

Table 4–26. IOE Internal Timing Microparameters

Symbol	-6		-7		-8		Unit
	Min	Max	Min	Max	Min	Max	
t_{SU}	348	—	400	—	452	—	ps
t_{H}	0	—	0	—	0	—	ps
t_{CO}	—	511	—	587	—	664	ps
$t_{PIN2COMBOUT_R}$	—	1,130	—	1,299	—	1,469	ps
$t_{PIN2COMBOUT_C}$	—	1,135	—	1,305	—	1,475	ps
$t_{COMBIN2PIN_R}$	—	2,627	—	3,021	—	3,415	ps
$t_{COMBIN2PIN_C}$	—	2,615	—	3,007	—	3,399	ps
t_{CLR}	280	—	322	—	364	—	ps
t_{PRE}	280	—	322	—	364	—	ps
t_{CLKHL}	1,234	—	1,562	—	1,818	—	ps

Table 4–27. M4K Block Internal Timing Microparameters

Symbol	-6		-7		-8		Unit
	Min	Max	Min	Max	Min	Max	
t_{M4KRC}	—	4,379		5,035		5,691	ps
t_{M4KWC}	—	2,910		3,346		3,783	ps
$t_{M4KWERESU}$	72	—	82	—	93	—	ps
$t_{M4KWEREH}$	43	—	49	—	55	—	ps
$t_{M4KBESU}$	72	—	82	—	93	—	ps
t_{M4KBEH}	43	—	49	—	55	—	ps
$t_{M4KDATAASU}$	72	—	82	—	93	—	ps
$t_{M4KDATAAH}$	43	—	49	—	55	—	ps
$t_{M4KADDRASU}$	72	—	82	—	93	—	ps
$t_{M4KADDRAH}$	43	—	49	—	55	—	ps
$t_{M4KDATABSU}$	72	—	82	—	93	—	ps
$t_{M4KDATABH}$	43	—	49	—	55	—	ps
$t_{M4KADDRBSU}$	72	—	82	—	93	—	ps
$t_{M4KADDRBH}$	43	—	49	—	55	—	ps
$t_{M4KDATACO1}$	—	621	—	714	—	807	ps
$t_{M4KDATACO2}$	—	4,351	—	5,003	—	5,656	ps
$t_{M4KCLKHL}$	1,234	—	1,562	—	1,818	—	ps
t_{M4KCLR}	286	—	328	—	371	—	ps

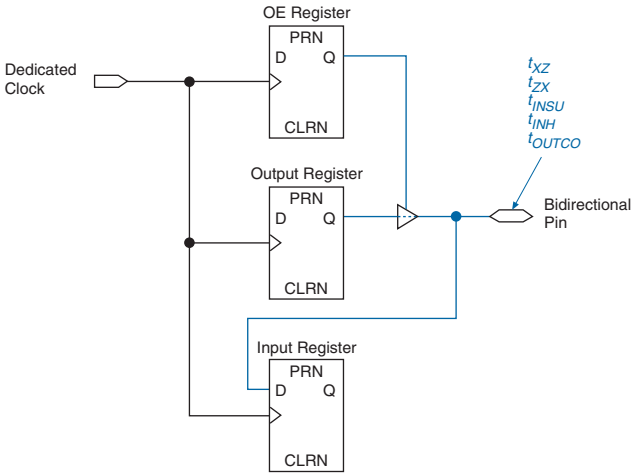
Table 4–28. Routing Delay Internal Timing Microparameters

Symbol	-6		-7		-8		Unit
	Min	Max	Min	Max	Min	Max	
t_{R4}	—	261	—	300	—	339	ps
t_{C4}	—	338	—	388	—	439	ps
t_{LOCAL}	—	244	—	281	—	318	ps

External Timing Parameters

External timing parameters are specified by device density and speed grade. Figure 4–2 shows the timing model for bidirectional IOE pin timing. All registers are within the IOE.

Figure 4–2. External Timing in Cyclone Devices



All external I/O timing parameters shown are for 3.3-V LVTTL I/O standard with the maximum current strength and fast slew rate. For external I/O timing using standards other than LVTTL or for different current strengths, use the I/O standard input and output delay adders in Tables 4–40 through 4–44.

Table 4–29 shows the external I/O timing parameters when using global clock networks.

Table 4–29. Cyclone Global Clock External I/O Timing Parameters *Notes (1), (2)* ***(Part 1 of 2)***

Symbol	Parameter	Conditions
t_{INSU}	Setup time for input or bidirectional pin using IOE input register with global clock fed by CLK pin	—
t_{INH}	Hold time for input or bidirectional pin using IOE input register with global clock fed by CLK pin	—
t_{OUTCO}	Clock-to-output delay output or bidirectional pin using IOE output register with global clock fed by CLK pin	C_{LOAD} = 10 pF
$t_{INSUPLL}$	Setup time for input or bidirectional pin using IOE input register with global clock fed by Enhanced PLL with default phase setting	—
t_{INHPLL}	Hold time for input or bidirectional pin using IOE input register with global clock fed by enhanced PLL with default phase setting	—

Table 4–29. Cyclone Global Clock External I/O Timing Parameters Notes (1), (2) (Part 2 of 2)		
Symbol	**Parameter**	**Conditions**
$t_{OUTCOPLL}$	Clock-to-output delay output or bidirectional pin using IOE output register with global clock enhanced PLL with default phase setting	C_{LOAD} = 10 pF

Notes to Table 4–29:
(1) These timing parameters are sample-tested only.
(2) These timing parameters are for IOE pins using a 3.3-V LVTTL, 24-mA setting. Designers should use the Quartus II software to verify the external timing for any pin.

Tables 4–30 through 4–31 show the external timing parameters on column and row pins for EP1C3 devices.

Table 4–30. EP1C3 Column Pin Global Clock External I/O Timing Parameters

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	3.085	—	3.547	—	4.009	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	4.073	2.000	4.682	2.000	5.295	ns
$t_{INSUPLL}$	1.795	—	2.063	—	2.332	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	2.306	0.500	2.651	0.500	2.998	ns

Table 4–31. EP1C3 Row Pin Global Clock External I/O Timing Parameters

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	3.157	—	3.630	—	4.103	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.984	2.000	4.580	2.000	5.180	ns
$t_{INSUPLL}$	1.867	—	2.146	—	2.426	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	2.217	0.500	2.549	0.500	2.883	ns

Tables 4–32 through 4–33 show the external timing parameters on column and row pins for EP1C4 devices.

Table 4–32. EP1C4 Column Pin Global Clock External I/O Timing Parameters Note (1)

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.471	—	2.841	—	3.210	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.937	2.000	4.526	2.000	5.119	ns
$t_{INSUPLL}$	1.471	—	1.690	—	1.910	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	2.080	0.500	2.392	0.500	2.705	ns

Table 4–33. EP1C4 Row Pin Global Clock External I/O Timing Parameters Note (1)

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.600	—	2.990	—	3.379	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.991	2.000	4.388	2.000	5.189	ns
$t_{INSUPLL}$	1.300	—	1.494	—	1.689	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	2.234	0.500	2.569	0.500	2.905	ns

Note to Tables 4–32 and 4–33:
(1) Contact Altera Applications for EP1C4 device timing parameters.

Tables 4–34 through 4–35 show the external timing parameters on column and row pins for EP1C6 devices.

Table 4–34. EP1C6 Column Pin Global Clock External I/O Timing Parameters

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.691	—	3.094	—	3.496	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.917	2.000	4.503	2.000	5.093	ns
$t_{INSUPLL}$	1.513	—	1.739	—	1.964	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	2.038	0.500	2.343	0.500	2.651	ns

Table 4–35. EP1C6 Row Pin Global Clock External I/O Timing Parameters

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.774	—	3.190	—	3.605	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.817	2.000	4.388	2.000	4.963	ns
$t_{INSUPLL}$	1.596	—	1.835	—	2.073	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	1.938	0.500	2.228	0.500	2.521	ns

Tables 4–36 through 4–37 show the external timing parameters on column and row pins for EP1C12 devices.

Table 4–36. EP1C12 Column Pin Global Clock External I/O Timing Parameters (Part 1 of 2)

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.510	—	2.885	—	3.259	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
tO_{UTCO}	2.000	3.798	2.000	4.367	2.000	4.940	ns
$t_{INSUPLL}$	1.588	—	1.824	—	2.061	—	ns

Table 4–36. EP1C12 Column Pin Global Clock External I/O Timing Parameters (Part 2 of 2)

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	1.663	0.500	1.913	0.500	2.164	ns

Table 4–37. EP1C12 Row Pin Global Clock External I/O Timing Parameters

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.620	—	3.012	—	3.404	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.671	2.000	4.221	2.000	4.774	ns
$t_{INSUPLL}$	1.698	—	1.951	—	2.206	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	1.536	0.500	1.767	0.500	1.998	ns

Tables 4–38 through 4–39 show the external timing parameters on column and row pins for EP1C20 devices.

Table 4–38. EP1C20 Column Pin Global Clock External I/O Timing Parameters

Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.417	—	2.779	—	3.140	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.724	2.000	4.282	2.000	4.843	ns
$t_{INSUPLL}$	1.417	—	1.629	—	1.840	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	1.667	0.500	1.917	0.500	2.169	ns

Table 4–39. EP1C20 Row Pin Global Clock External I/O Timing Parameters							
Symbol	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
t_{INSU}	2.417	—	2.779	—	3.140	—	ns
t_{INH}	0.000	—	0.000	—	0.000	—	ns
t_{OUTCO}	2.000	3.724	2.000	4.282	2.000	4.843	ns
t_{XZ}	—	3.645	—	4.191	—	4.740	ns
t_{ZX}	—	3.645	—	4.191	—	4.740	ns
$t_{INSUPLL}$	1.417	—	1.629	—	1.840	—	ns
t_{INHPLL}	0.000	—	0.000	—	0.000	—	ns
$t_{OUTCOPLL}$	0.500	1.667	0.500	1.917	0.500	2.169	ns
t_{XZPLL}	—	1.588	—	1.826	—	2.066	ns
t_{ZXPLL}	—	1.588	—	1.826	—	2.066	ns

External I/O Delay Parameters

External I/O delay timing parameters for I/O standard input and output adders and programmable input and output delays are specified by speed grade independent of device density.

Tables 4–40 through 4–45 show the adder delays associated with column and row I/O pins for all packages. If an I/O standard is selected other than LVTTL 4 mA with a fast slew rate, add the selected delay to the external t_{CO} and t_{SU} I/O parameters shown in Tables 4–25 through 4–28.

Table 4–40. Cyclone I/O Standard Column Pin Input Delay Adders (Part 1 of 2)							
I/O Standard	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
LVCMOS	—	0	—	0	—	0	ps
3.3-V LVTTL	—	0	—	0	—	0	ps
2.5-V LVTTL	—	27	—	31	—	35	ps
1.8-V LVTTL	—	182	—	209	—	236	ps
1.5-V LVTTL	—	278	—	319	—	361	ps
SSTL-3 class I	—	–250	—	–288	—	–325	ps
SSTL-3 class II	—	–250	—	–288	—	–325	ps
SSTL-2 class I	—	–278	—	–320	—	–362	ps

Table 4–40. Cyclone I/O Standard Column Pin Input Delay Adders (Part 2 of 2)

I/O Standard	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
SSTL-2 class II		–278	—	–320	—	–362	ps
LVDS		–261	—	–301	—	–340	ps

Table 4–41. Cyclone I/O Standard Row Pin Input Delay Adders

I/O Standard	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
LVCMOS	—	0	—	0	—	0	ps
3.3-V LVTTL	—	0	—	0	—	0	ps
2.5-V LVTTL	—	27	—	31	—	35	ps
1.8-V LVTTL	—	182	—	209	—	236	ps
1.5-V LVTTL	—	278	—	319	—	361	ps
3.3-V PCI *(1)*	—	0	—	0	—	0	ps
SSTL-3 class I	—	–250	—	–288	—	–325	ps
SSTL-3 class II	—	–250	—	–288	—	–325	ps
SSTL-2 class I	—	–278	—	–320	—	–362	ps
SSTL-2 class II	—	–278	—	–320	—	–362	ps
LVDS	—	–261	—	–301	—	–340	ps

Table 4–42. Cyclone I/O Standard Output Delay Adders for Fast Slew Rate on Column Pins (Part 1 of 2)

Standard		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
LVCMOS	2 mA	—	0	—	0	—	0	ps
	4 mA	—	–489	—	–563	—	–636	ps
	8 mA	—	–855	—	–984	—	–1,112	ps
	12 mA	—	–993	—	–1,142	—	–1,291	ps
3.3-V LVTTL	4 mA	—	0	—	0	—	0	ps
	8 mA	—	–347	—	–400	—	–452	ps
	12 mA	—	–858	—	–987	—	–1,116	ps
	16 mA	—	–819	—	–942	—	–1,065	ps
	24 mA	—	–993	—	–1,142	—	–1,291	ps

Table 4–42. Cyclone I/O Standard Output Delay Adders for Fast Slew Rate on Column Pins (Part 2 of 2)

Standard		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
2.5-V LVTTL	2 mA	—	329	—	378	—	427	ps
	8 mA	—	–661	—	–761	—	–860	ps
	12 mA	—	–655	—	–754	—	–852	ps
	16 mA	—	–795	—	–915	—	–1034	ps
1.8-V LVTTL	2 mA	—	4	—	4	—	5	ps
	8 mA	—	–208	—	–240	—	–271	ps
	12 mA	—	–208	—	–240	—	–271	ps
1.5-V LVTTL	2 mA	—	2,288	—	2,631	—	2,974	ps
	4 mA	—	608	—	699	—	790	ps
	8 mA	—	292	—	335	—	379	ps
SSTL-3 class I		—	–410	—	–472	—	–533	ps
SSTL-3 class II		—	–811	—	–933	—	–1,055	ps
SSTL-2 class I		—	–485	—	–558	—	–631	ps
SSTL-2 class II		—	–758	—	–872	—	–986	ps
LVDS		—	–998	—	–1,148	—	–1,298	ps

Table 4–43. Cyclone I/O Standard Output Delay Adders for Fast Slew Rate on Row Pins (Part 1 of 2)

Standard		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
LVCMOS	2 mA	—	0	—	0	—	0	ps
	4 mA	—	–489	—	–563	—	–636	ps
	8 mA	—	–855	—	–984	—	–1,112	ps
	12 mA	—	–993	—	–1,142	—	–1,291	ps
3.3-V LVTTL	4 mA	—	0	—	0	—	0	ps
	8 mA	—	–347	—	–400	—	–452	ps
	12 mA	—	–858	—	–987	—	–1,116	ps
	16 mA	—	–819	—	–942	—	–1,065	ps
	24 mA	—	–993	—	–1,142	—	–1,291	ps
2.5-V LVTTL	2 mA	—	329	—	378	—	427	ps
	8 mA	—	–661	—	–761	—	–860	ps
	12 mA	—	–655	—	–754	—	–852	ps
	16 mA	—	–795	—	–915	—	–1,034	ps

Table 4–43. Cyclone I/O Standard Output Delay Adders for Fast Slew Rate on Row Pins (Part 2 of 2)

Standard		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
1.8-V LVTTL	2 mA	—	1,290	—	1,483	—	1,677	ps
	8 mA	—	4	—	4	—	5	ps
	12 mA	—	–208	—	–240	—	–271	ps
1.5-V LVTTL	2 mA	—	2,288	—	2,631	—	2,974	ps
	4 mA	—	608	—	699	—	790	ps
	8 mA	—	292	—	335	—	379	ps
3.3-V PCI *(1)*		—	–877	—	–1,009	—	–1,141	ps
SSTL-3 class I		—	–410	—	–472	—	–533	ps
SSTL-3 class II		—	–811	—	–933	—	–1,055	ps
SSTL-2 class I		—	–485	—	–558	—	–631	ps
SSTL-2 class II		—	–758	—	–872	—	–986	ps
LVDS		—	–998	—	–1,148	—	–1,298	ps

Table 4–44. Cyclone I/O Standard Output Delay Adders for Slow Slew Rate on Column Pins (Part 1 of 2)

I/O Standard		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
LVCMOS	2 mA	—	1,800	—	2,070	—	2,340	ps
	4 mA	—	1,311	—	1,507	—	1,704	ps
	8 mA	—	945	—	1,086	—	1,228	ps
	12 mA	—	807	—	928	—	1,049	ps
3.3-V LVTTL	4 mA	—	1,831	—	2,105	—	2,380	ps
	8 mA	—	1,484	—	1,705	—	1,928	ps
	12 mA	—	973	—	1,118	—	1,264	ps
	16 mA	—	1,012	—	1,163	—	1,315	ps
	24 mA	—	838	—	963	—	1,089	ps
2.5-V LVTTL	2 mA	—	2,747	—	3,158	—	3,570	ps
	8 mA	—	1,757	—	2,019	—	2,283	ps
	12 mA	—	1,763	—	2,026	—	2,291	ps
	16 mA	—	1,623	—	1,865	—	2,109	ps
1.8-V LVTTL	2 mA	—	5,506	—	6,331	—	7,157	ps
	8 mA	—	4,220	—	4,852	—	5,485	ps
	12 mA	—	4,008	—	4,608	—	5,209	ps

Table 4–44. Cyclone I/O Standard Output Delay Adders for Slow Slew Rate on Column Pins (Part 2 of 2)

I/O Standard		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
1.5-V LVTTL	2 mA	—	6,789	—	7,807	—	8,825	ps
	4 mA	—	5,109	—	5,875	—	6,641	ps
	8 mA	—	4,793	—	5,511	—	6,230	ps
SSTL-3 class I		—	1,390	—	1,598	—	1,807	ps
SSTL-3 class II		—	989	—	1,137	—	1,285	ps
SSTL-2 class I		—	1,965	—	2,259	—	2,554	ps
SSTL-2 class II		—	1,692	—	1,945	—	2,199	ps
LVDS		—	802	—	922	—	1,042	ps

Table 4–45. Cyclone I/O Standard Output Delay Adders for Slow Slew Rate on Row Pins (Part 1 of 2)

I/O Standard		-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
LVCMOS	2 mA	—	1,800	—	2,070	—	2,340	ps
	4 mA	—	1,311	—	1,507	—	1,704	ps
	8 mA	—	945	—	1,086	—	1,228	ps
	12 mA	—	807	—	928	—	1,049	ps
3.3-V LVTTL	4 mA	—	1,831	—	2,105	—	2,380	ps
	8 mA	—	1,484	—	1,705	—	1,928	ps
	12 mA	—	973	—	1,118	—	1,264	ps
	16 mA	—	1,012	—	1,163	—	1,315	ps
	24 mA	—	838	—	963	—	1,089	ps
2.5-V LVTTL	2 mA	—	2,747	—	3,158	—	3,570	ps
	8 mA	—	1,757	—	2,019	—	2,283	ps
	12 mA	—	1,763	—	2,026	—	2,291	ps
	16 mA	—	1,623	—	1,865	—	2,109	ps
1.8-V LVTTL	2 mA	—	5,506	—	6,331	—	7,157	ps
	8 mA	—	4,220	—	4,852	—	5,485	ps
	12 mA	—	4,008	—	4,608	—	5,209	ps
1.5-V LVTTL	2 mA	—	6,789	—	7,807	—	8,825	ps
	4 mA	—	5,109	—	5,875	—	6,641	ps
	8 mA	—	4,793	—	5,511	—	6,230	ps
3.3-V PCI		—	923	—	1,061	—	1,199	ps

Table 4–45. Cyclone I/O Standard Output Delay Adders for Slow Slew Rate on Row Pins (Part 2 of 2)

I/O Standard	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
SSTL-3 class I	—	1,390	—	1,598	—	1,807	ps
SSTL-3 class II	—	989	—	1,137	—	1,285	ps
SSTL-2 class I	—	1,965	—	2,259	—	2,554	ps
SSTL-2 class II	—	1,692	—	1,945	—	2,199	ps
LVDS	—	802	—	922	—	1,042	ps

Note to Tables 4–40 through 4–45:
(1) EP1C3 devices do not support the PCI I/O standard.

Tables 4–46 through 4–47 show the adder delays for the IOE programmable delays. These delays are controlled with the Quartus II software options listed in the Parameter column.

Table 4–46. Cyclone IOE Programmable Delays on Column Pins

Parameter	Setting	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
Decrease input delay to internal cells	Off	—	155	—	178	—	201	ps
	Small	—	2,122	—	2,543	—	2,875	ps
	Medium	—	2,639	—	3,034	—	3,430	ps
	Large	—	3,057	—	3,515	—	3,974	ps
	On	—	155	—	178	—	201	ps
Decrease input delay to input register	Off	—	0	—	0	—	0	ps
	On	—	3,057	—	3,515	—	3,974	ps
Increase delay to output pin	Off	—	0	—	0	—	0	ps
	On	—	552	—	634	—	717	ps

Table 4–47. Cyclone IOE Programmable Delays on Row Pins

Parameter	Setting	-6 Speed Grade		-7 Speed Grade		-8 Speed Grade		Unit
		Min	Max	Min	Max	Min	Max	
Decrease input delay to internal cells	Off	—	154	—	177	—	200	ps
	Small	—	2,212	—	2,543	—	2,875	ps
	Medium	—	2,639	—	3,034	—	3,430	ps
	Large	—	3,057	—	3,515	—	3,974	ps
	On	—	154	—	177	—	200	ps
Decrease input delay to input register	Off	—	0	—	0	—	0	ps
	On	—	3,057	—	3,515	—	3,974	ps
Increase delay to output pin	Off	—	0	—	0	—	0	ps
	On	—	556	—	639	—	722	ps

Note to Table 4–47:
(1) EPC1C3 devices do not support the PCI I/O standard.

Maximum Input and Output Clock Rates

Tables 4–48 and 4–49 show the maximum input clock rate for column and row pins in Cyclone devices.

Table 4–48. Cyclone Maximum Input Clock Rate for Column Pins

I/O Standard	-6 Speed Grade	-7 Speed Grade	-8 Speed Grade	Unit
LVTTL	464	428	387	MHz
2.5 V	392	302	207	MHz
1.8 V	387	311	252	MHz
1.5 V	387	320	243	MHz
LVCMOS	405	374	333	MHz
SSTL-3 class I	405	356	293	MHz
SSTL-3 class II	414	365	302	MHz
SSTL-2 class I	464	428	396	MHz
SSTL-2 class II	473	432	396	MHz
LVDS	567	549	531	MHz

Table 4–49. Cyclone Maximum Input Clock Rate for Row Pins				
I/O Standard	-6 Speed Grade	-7 Speed Grade	-8 Speed Grade	Unit
LVTTL	464	428	387	MHz
2.5 V	392	302	207	MHz
1.8 V	387	311	252	MHz
1.5 V	387	320	243	MHz
LVCMOS	405	374	333	MHz
SSTL-3 class I	405	356	293	MHz
SSTL-3 class II	414	365	302	MHz
SSTL-2 class I	464	428	396	MHz
SSTL-2 class II	473	432	396	MHz
3.3-V PCI *(1)*	464	428	387	MHz
LVDS	567	549	531	MHz

Note to Tables 4–48 through 4–49:
(1) EP1C3 devices do not support the PCI I/O standard. These parameters are only available on row I/O pins.

Tables 4–50 and 4–51 show the maximum output clock rate for column and row pins in Cyclone devices.

Table 4–50. Cyclone Maximum Output Clock Rate for Column Pins				
I/O Standard	-6 Speed Grade	-7 Speed Grade	-8 Speed Grade	Unit
LVTTL	304	304	304	MHz
2.5 V	220	220	220	MHz
1.8 V	213	213	213	MHz
1.5 V	166	166	166	MHz
LVCMOS	304	304	304	MHz
SSTL-3 class I	100	100	100	MHz
SSTL-3 class II	100	100	100	MHz
SSTL-2 class I	134	134	134	MHz
SSTL-2 class II	134	134	134	MHz
LVDS	320	320	275	MHz

Note to Table 4–50:
(1) EP1C3 devices do not support the PCI I/O standard.

Table 4–51. Cyclone Maximum Output Clock Rate for Row Pins				
I/O Standard	-6 Speed Grade	-7 Speed Grade	-8 Speed Grade	Unit
LVTTL	296	285	273	MHz
2.5 V	381	366	349	MHz
1.8 V	286	277	267	MHz
1.5 V	219	208	195	MHz
LVCMOS	367	356	343	MHz
SSTL-3 class I	169	166	162	MHz
SSTL-3 class II	160	151	146	MHz
SSTL-2 class I	160	151	142	MHz
SSTL-2 class II	131	123	115	MHz
3.3-V PCI *(1)*	66	66	66	MHz
LVDS	320	303	275	MHz

Note to Tables 4–50 through 4–51:
(1) EP1C3 devices do not support the PCI I/O standard. These parameters are only available on row I/O pins.

PLL Timing

Table 4–52 describes the Cyclone FPGA PLL specifications.

Table 4–52. Cyclone PLL Specifications (Part 1 of 2)				
Symbol	Parameter	Min	Max	Unit
f_{IN}	Input frequency (-6 speed grade)	15.625	464	MHz
	Input frequency (-7 speed grade)	15.625	428	MHz
	Input frequency (-8 speed grade)	15.625	387	MHz
f_{IN} DUTY	Input clock duty cycle	40.00	60	%
t_{IN} JITTER	Input clock period jitter	—	± 200	ps
f_{OUT_EXT} (external PLL clock output)	PLL output frequency (-6 speed grade)	15.625	320	MHz
	PLL output frequency (-7 speed grade)	15.625	320	MHz
	PLL output frequency (-8 speed grade)	15.625	275	MHz

Table 4–52. Cyclone PLL Specifications (Part 2 of 2)

Symbol	Parameter	Min	Max	Unit
f_{OUT} (to global clock)	PLL output frequency (-6 speed grade)	15.625	405	MHz
	PLL output frequency (-7 speed grade)	15.625	320	MHz
	PLL output frequency (-8 speed grade)	15.625	275	MHz
t_{OUT} DUTY	Duty cycle for external clock output (when set to 50%)	45.00	55	%
t_{JITTER} *(1)*	Period jitter for external clock output	—	±300 *(2)*	ps
t_{LOCK} *(3)*	Time required to lock from end of device configuration	10.00	100	µs
f_{VCO}	PLL internal VCO operating range	500.00	1,000	MHz
-	Minimum areset time	10	—	ns
N, G0, G1, E	Counter values	1	32	integer

Notes to Table 4–52:

(1) The t_{JITTER} specification for the `PLL[2..1]_OUT` pins are dependent on the I/O pins in its V_{CCIO} bank, how many of them are switching outputs, how much they toggle, and whether or not they use programmable current strength or slow slew rate.

(2) $f_{OUT} \geq 100$ MHz. When the PLL external clock output frequency (f_{OUT}) is smaller than 100 MHz, the jitter specification is 60 mUI.

(3) $f_{IN/N}$ must be greater than 200 MHz to ensure correct lock detect circuit operation below –20 C. Otherwise, the PLL operates with the specified parameters under the specified conditions.

Referenced Document

This chapter references the following documents:

- *Cyclone Architecture* chapter in the *Cyclone Device Handbook*
- *Operating Requirements for Altera Devices Data Sheet*

Document Revision History

Table 4–53 shows the revision history for this chapter.

Table 4–53. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.7	Minor textual and style changes. Added "Referenced Document" section.	—
January 2007 v1.6	• Added document revision history. • Added new row for V_{CCA} details in Table 4–1. • Updated R_{CONF} information in Table 4–3. • Added new *Note (12)* on voltage overdrive information to Table 4–7 and Table 4–8. • Updated *Note (9)* on R_{CONF} information to Table 4–3. • Updated information in "External I/O Delay Parameters" section. • Updated speed grade information in Table 4–46 and Table 4–47. • Updated LVDS information in Table 4–51.	—
August 2005 v1.5	Minor updates.	—
February 2005 v1.4	• Updated information on Undershoot voltage. Updated Table 4-2. • Updated Table 4-3. • Updated the undershoot voltage from 0.5 V to 2.0 V in Note 3 of Table 4-16. • Updated Table 4-17.	—
January 2004 v.1.3	• Added extended-temperature grade device information. Updated Table 4-2. • Updated I_{CC0} information in Table 4-3.	—
October 2003 v.1.2	• Added clock tree information in Table 4-19. • Finalized timing information for EP1C3 and EP1C12 devices. Updated timing information in Tables 4-25 through 4-26 and Tables 4-30 through 4-51. • Updated PLL specifications in Table 4-52.	—

July 2003 v1.1	Updated timing information. Timing finalized for EP1C6 and EP1C20 devices. Updated performance information. Added PLL Timing section.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

5. Reference and Ordering Information

C51005-1.4

Software

Cyclone® devices are supported by the Altera® Quartus® II design software, which provides a comprehensive environment for system-on-a-programmable-chip (SOPC) design. The Quartus II software includes HDL and schematic design entry, compilation and logic synthesis, full simulation and advanced timing analysis, SignalTap® II logic analysis, and device configuration.



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

The Quartus II software supports the Windows 2000/NT/98, Sun Solaris, Linux Red Hat v7.1 and HP-UX operating systems. It also supports seamless integration with industry-leading EDA tools through the NativeLink® interface.

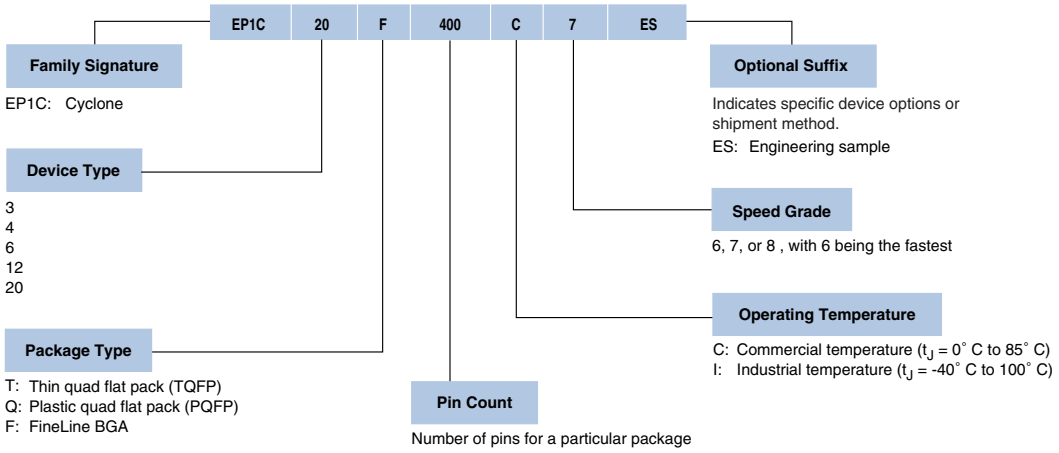
Device Pin-Outs

Device pin-outs for Cyclone devices are available on the Altera website (www.altera.com) and in the *Cyclone Device Handbook*.

Ordering Information

Figure 5–1 describes the ordering codes for Cyclone devices. For more information about a specific package, refer to the *Package Information for Cyclone Devices* chapter in the *Cyclone Device Handbook*.

Figure 5–1. Cyclone Device Packaging Ordering Information



Referenced Documents

This chapter references the following documents:

- *Package Information for Cyclone Devices* chapter in the *Cyclone Device Handbook*
- *Quartus II Handbook*

Document Revision History

Table 5–1 shows the revision history for this chapter.

Table 5–1. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.4	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.3	Added document revision history.	—
August 2005 v1.2	Minor updates.	—

February 2005 v1.1	Updated Figure 5-1.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

Section II. Clock Management

This section provides information on the Cyclone phase-lock loops (PLLs). The PLLs assist designers in managing clocks internally and also have the ability to drive off chip to control system-level clock networks. This chapter contains detailed information on the features, the interconnections to the logic array and off chip, and the specifications for Cyclone PLLs.

This section contains the following chapter:

- Chapter 6. Using PLLs in Cyclone Devices

Revision History

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

6. Using PLLs in Cyclone Devices

C51006-1.5

Introduction

Cyclone® FPGAs offer phase locked loops (PLLs) and a global clock network for clock management solutions. Cyclone PLLs offer clock multiplication and division, phase shifting, programmable duty cycle, and external clock outputs, allowing system-level clock management and skew control. The Altera® Quartus® II software enables Cyclone PLLs and their features without using any external devices. This chapter explains how to design and enable Cyclone PLL features.

PLLs are commonly used to synchronize internal device clocks with an external clock, run internal clocks at higher frequencies than an external clock, minimize clock delay and clock skew, and reduce or adjust clock-to-out (t_{CO}) and set-up (t_{SU}) times.

Hardware Overview

Cyclone FPGAs contain up to two PLLs per device. Table 6–1 shows which PLLs are available for each Cyclone FPGA.

Table 6–1. Cyclone FPGA PLL Availability

Device	PLL1 *(1)*	PLL2 *(2)*
EP1C3	✓	—
EP1C4	✓	✓
EP1C6	✓	✓
EP1C12	✓	✓
EP1C20	✓	✓

Notes to Table 6–1:
(1) Located on the center left side of the device.
(2) Located on the center right side of the device.

Table 6–2 provides an overview of available Cyclone PLL features.

Table 6–2. Cyclone PLL Features

Feature	Description
Clock multiplication and division	*M*/(*N* × post-scale counter) *(1)*
Phase shift	Down to 125-ps increments *(2)*, *(3)*
Programmable duty cycle	✓
Number of internal clock outputs	Two per PLL
Number of external clock outputs *(4)*	One per PLL
Locked port can feed logic array	✓
PLL clock outputs can feed logic array	✓

Notes to Table 6–2:

(1) *M*, N, and post-scale counter values range from 1 to 32.

(2) The smallest phase shift is determined by the Voltage Control Oscillator (VCO) period divided by 8.

(3) For degree increments, Cyclone FPGAs can shift output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the multiplication/division ratio needed on the PLL clock output.

(4) The EP1C3 device in the 100-pin thin quad flat pack (TQFP) package does not have support for a PLL LVDS input or an external clock output. The EP1C6 PLL2 in the 144-pin TQFP package does not support an external clock output.

Cyclone PLL Blocks

The main goal of a PLL is to synchronize the phase and frequency of an internal/external clock to an input reference clock. There are a number of components that comprise a PLL to achieve this phase alignment.

Cyclone PLLs align the rising edge of the reference input clock to a feedback clock using a phase-frequency detector (PFD). The falling edges are determined by the duty cycle specifications. The PFD produces an up or down signal that determines whether the VCO needs to operate at a higher or lower frequency. The PFD output is applied to the charge pump and loop filter, which produces a control voltage for setting the frequency of the VCO. If the PFD produces an up signal, then the VCO frequency increases, while a down signal causes the VCO frequency to decrease.

The PFD outputs these up and down signals to a charge pump. If the charge pump receives an up signal, current is driven into the loop filter. Conversely, if it receives a down signal, current is drawn from the loop filter. The loop filter converts these up and down signals to a voltage that

is used to bias the VCO. The loop filter also removes glitches from the charge pump and prevents voltage over-shoot, which minimizes the jitter on the VCO.

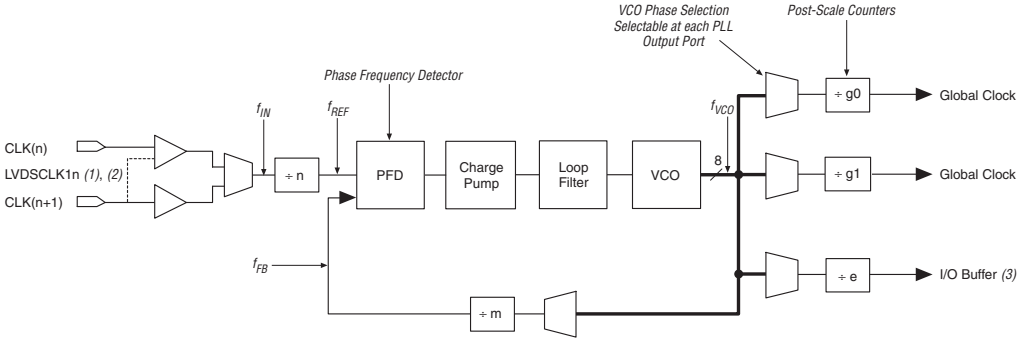
The voltage from the loop filter determines how fast the VCO operates. The VCO is implemented as a four-stage differential ring oscillator. A divide counter (*M*) is inserted in the feedback loop to increase the VCO frequency above the input reference frequency, making the VCO frequency (f_{VCO}) equal to *M* times the input reference clock (f_{REF}). The input reference clock (f_{REF}) to the PFD is equal to the input clock (f_{IN}) divided by the pre-scale counter (*N*). Therefore, the feedback clock (f_{FB}) that is applied to one input of the PFD is locked to the f_{REF} that is applied to the other input of the PFD.

The VCO output can feed up to three post-scale counters (G0, G1, and E). These post-scale counters allow a number of harmonically-related frequencies to be produced within the PLL.

Additionally, the PLL has internal delay elements to compensate for routing on the global clock networks and I/O buffers of the external clock output pins. These internal delays are fixed and not accessible to the user.

Figure 6–1 shows a block diagram of the major components of a Cyclone PLL.

Figure 6–1. Cyclone PLL



Notes to Figure 6–1:

(1) The EP1C3 device in the 100-pin TQFP package does not have support for a PLL LVDS input.

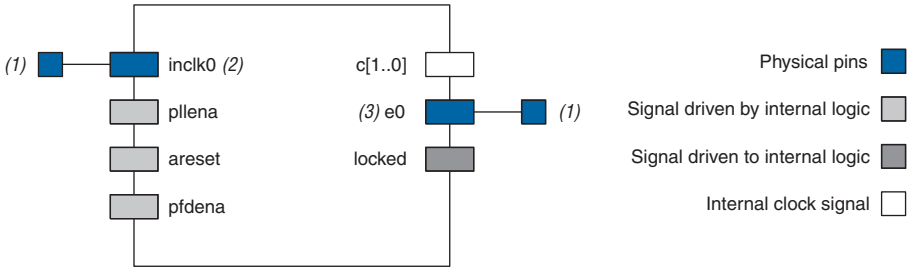
(2) If you are using the LVDS standard, then both `CLK` pins of that PLL are used. LVDS input is supported via the secondary function of the dedicated `CLK` pins. For PLL1, the `CLK0` pin's secondary function is `LVDSCLK1p` and the `CLK1` pin's secondary function is `LVDSCLK1n`. For PLL2, the `CLK2` pin's secondary function is `LVDSCLK2p` and the `CLK3` pin's secondary function is `LVDSCLK2n`.

(3) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not support an external clock output.

Software Overview

Cyclone PLLs are enabled in the Quartus II software by using the `altpll` megafunction. Figure 6–2 shows the available ports (as they are named in the Quartus II `altpll` megafunction) of Cyclone PLLs and their sources and destinations. It is important to note that the `c[1..0]` and `e0` clock output ports from `altpll` are driven by the post-scale counters G0, G1, and E (not necessarily in that order). The G0 and G1 counters feed the internal global clock network on the `c0` and `c1` PLL outputs, and the E counter feeds the PLL external clock output pin on the `e0` PLL output.

Figure 6–2. Cyclone PLL Signals



Notes to Figure 6–2:

(1) You can assign these signals to either a single-ended I/O standard or LVDS.
(2) `Inclk0` must be driven by the dedicated clock input pin(s).
(3) `e0` drives the dual-purpose `PLL[2..1]_OUT` pins.

Tables 6–3 and 6–4 describe the Cyclone PLL input and output ports.

Table 6–3. PLL Input Signals

Port	Description	Source	Destination
inclk0	Clock input to PLL.	Dedicated clock input pin *(1)*	÷n counter
pllena *(2)*	pllena is an active-high signal that acts as a combined enable and reset signal for the PLL. You can use It for enabling or disabling one or two PLLs. When this signal is driven low, the PLL clock output ports are driven to GND and the PLL loses lock. Once this signal is driven high again, the lock process begins and the PLL re-synchronizes to its input reference clock. You can drive the pllena port from internal logic or any general-purpose I/O pin.	Logic array *(3)*	PLL control signal
areset	areset is an active-high signal that resets all PLL counters to their initial values. When this signal is driven high, the PLL resets its counters, clears the PLL outputs, and loses lock. Once this signal is driven low again, the lock process begins and the PLL re-synchronizes to its input reference clock. You can drive the areset port from internal logic or any general-purpose I/O pin.	Logic array *(3)*	PLL control signal
pfdena	pfdena is an active-high signal that enables or disables the up/down output signals from the PFD. When pfdena is driven low, the PFD is disabled, while the VCO continues to operate. The PLL clock outputs continue to toggle regardless of the input clock, but can experience some long-term drift. Because the output clock frequency does not change for some time, you can use the pfdena port as a shutdown or cleanup function when a reliable input clock is no longer available. You can drive the pfdena port from internal logic or any general-purpose I/O pin.	Logic array *(3)*	PFD

Notes to Table 6–3:

(1) The inclk0 port to the PLL must be driven by the dedicated clock input pin(s).

(2) There is no dedicated pllena pin for all PLLs, allowing you to choose either one pllena pin for both PLLs or each PLL can have its own pllena pin.

(3) Logic array source means that you can drive the port from internal logic or any general-purpose I/O pin.

Table 6–4. PLL Output Signals

Port	Description	Source	Destination
c[1..0]	PLL clock outputs driving the internal global clock network.	PLL post-scale counter G0 or G1	Global clock network *(1)*
e0 *(2)*	PLL clock output driving the single-ended or LVDS external clock output pin(s).	PLL post-scale counter E	PLL[2..1]_OUT pin(s) *(3)*
locked	Gives the status of the PLL lock. When the PLL is locked, this port drives logic high. When the PLL is out of lock, this port drives logic low. The locked port can pulse high and low during the PLL lock process.	PLL lock detect	Logic array *(4)*

Notes to Table 6–4:

(1) C[1..0] can also drive to any general-purpose I/O pin through the global clock network.

(2) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not have support for the external clock output PLL[2..1]_OUT.

(3) The PLL[2..1]_OUT pins are dual-purpose pins. If these pins are not required, they are available for use as general-purpose I/O pins.

(4) Logic array destination means that you can drive the port to internal logic or any general-purpose I/O pin.

In the Quartus II software, you define which internal clock output from the PLL (c0 or c1) should be compensated. This PLL clock output is phase-aligned with respect to the PLL input clock. For example, if c0 is specified as the compensation clock in normal mode, the compensation is based on the c0 routing on the global clock network.

Pins and Clock Network Connections

You must drive Cyclone PLLs by the dedicated clock input pins CLK[3..0]. Inverted clocks and internally generated clocks cannot drive the PLL. Table 6–5 shows which dedicated clock pin drives which PLL input clock port.



A single clock input pin cannot drive both PLLs, but a single clock input pin can feed both registers in the logic array, as well as the PLL `inclk` port.

Table 6–5. PLL Input Clock Sources

Clock Input Pins *(1)*	PLL1	PLL2 *(2)*
CLK0	✓	—
CLK1	✓	—
CLK2	—	✓
CLK3	—	✓

Notes to Table 6–5:
(1) If you are using the LVDS standard, then both `CLK` pins driving that PLL are used.
(2) The EP1C3 device only supports PLL1.

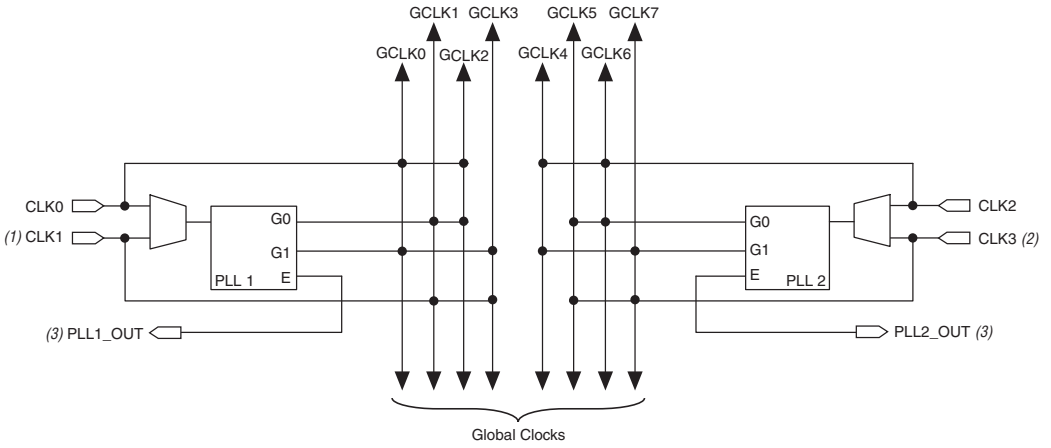
The `c[1..0]` and `e0` clock output ports from `altpll` are driven by the PLL post-scale counters G0, G1, and E (not necessarily in that order). The G0 and G1 counters feed the internal global clock network on the `c0` and `c1` PLL outputs, and the E counter feeds the PLL external clock output pin on the `e0` PLL output. Table 6–6 shows which global clock network can be driven by which PLL post-scale counter output.

Table 6–6. PLL Output Clock Destinations onto the Global Clock Network

PLL	Counter Output	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
PLL 1	G0	—	✓	✓	—	—	—	—	—
	G1	✓	—	—	✓	—	—	—	—
PLL2	G0	—	—	—	—	—	✓	✓	—
	G1	—	—	—	—	✓	—	—	✓

Figure 6–3 summarizes Tables 6–5 and 6–6 by showing the PLL input and output clock connections.

Figure 6–3. Cyclone PLL Clock Connections



Notes to Figure 6–3:

(1) PLL1 supports one single-ended or LVDS input via the `CLK0` and `CLK1` pins.

(2) PLL2 supports one single-ended or LVDS input via the `CLK2` and `CLK3` pins.

(3) `PLL1_OUT` and `PLL2_OUT` support single-ended or LVDS outputs. If the external clock output is not required, these pins are available as general-purpose I/O pins.

You can invert the clock outputs of the PLL at the logic array block (LAB) and at the input/output element (IOE) level.

Hardware Features

Cyclone PLLs have a number of advanced features available, including clock multiplication and division, phase shifting, programmable duty cycles, external clock outputs, and control signals.

Clock Multiplication and Division

Cyclone PLLs provide clock synthesis for PLL output ports using $M/(N \times \text{post-scale})$ scaling factors. There is one pre-scale divider (N) and one multiply counter (M) per PLL. N and post-scale counter values range from 1 to 32. The M counter ranges from 2 to 32. The input clock (f_{IN}) is divided by a pre-scale counter (N) to produce the input reference clock (f_{REF}) to the PFD. f_{REF} is then multiplied by the M feedback factor. The control loop drives the VCO frequency to match $f_{IN} \times (M/N)$. See the following equations:

$$f_{REF} = f_{IN}/N$$
$$f_{VCO} = f_{REF} \times M = f_{IN} \times (M/N)$$

Each output port has a unique post-scale counter to divide down the high-frequency VCO. There are three post-scale counters (G0, G1, and E) that range from 1 to 32. See the following equations:

$$f_{C0} = f_{VCO}/G0 = f_{IN} \times (M/(N \times G0))$$
$$f_{C1} = f_{VCO}/G1 = f_{IN} \times (M/(N \times G1))$$
$$f_{E} = f_{VCO}/E = f_{IN} \times (M/(N \times E))$$



`c0` and `c1` can use either post-scale counter, G0 or G1.

For multiple PLL outputs with different frequencies, the VCO is set to the least common multiple of the output frequencies that meets the VCO frequency specifications. Then, the post-scale counters scale down the output frequency for each PLL clock output port. For example, if clock output frequencies required from one PLL are 33 and 66 MHz, the VCO is set to 330 MHz (the least common multiple in the VCO's range).

Phase Shifting

Cyclone PLLs have advanced clock shift capability to provide programmable phase shifting. You can enter the desired phase shift in the `altpll` MegaWizard® Plug-In Manager and the Quartus II software automatically sets and displays the closest phase shift achievable. You can enter the phase shift in degrees, or units of time, for each PLL clock output port. This feature is supported on all three PLL post-scale counters, G0, G1, and E and is supported for all available clock feedback modes.

Phase shifting is performed with respect to the PLL clock output that is compensated. For example, you have a 100 MHz input clock and request a × 1 multiplication with a +90° phase shift on `c0` and a × 1 multiplication with a +45° phase shift on `c1`. If you choose to compensate for the `c0` clock output, the PLL uses a zero phase-shifted `c0` clock as a reference point to produce the +90° phase shift on `c0`. Since `c0` is the compensated clock, it is phase-shifted +90° from the input clock. The `c1` clock also uses the zero phase-shifted `c0` reference to produce the +45° phase shift on `c1`.

For fine phase adjustment, each PLL clock output counter can choose a different phase of the VCO from up to eight phase taps. In addition, each clock output counter can use a unique initial count setting to achieve individual coarse phase shift selection, in steps of one VCO period. The Quartus II software can use this clock output counter, along with an initial setting on the post-scale counter, to achieve a phase shift range for the entire period of the output clock. You can phase shift the PLL clock output up to ±180°. The Quartus II software automatically sets the phase taps and counter settings according to the phase shift requested.

The resolution of the fine phase adjustment is dependent on the input frequency and the multiplication/division factors (i.e., it is a function of the VCO period), with the finest step being equal to an eighth (× 0.125) of the VCO period. The minimum phase shift is $1/(8 \times f_{VCO})$ or $N/(8 \times M \times f_{IN})$. In Cyclone FPGAs, the VCO ranges from 500 to 1,000 MHz. Therefore, phase shifting can be performed with a resolution range of 1/(8 × 1,000 MHz) to 1/(8 × 500 MHz), which is 125 to 250 ps in time units.

Because there are eight VCO phase taps, the maximum step size is 45°. Smaller steps are possible, depending on the multiplication and division ratio necessary on the output clock port. The equation to determine the precision of the phase shifting in degrees is 45° divided by the post-scale counter value. For example, if you have an input clock of 125 MHz with × 1, the post-scale counter G0 is 3. Therefore, the smallest phase shift step is (45°/3 = 15°) and possible phase-shift values would be multiples of 15°.

This type of phase shift provides the highest precision since it is the least sensitive to process, voltage and temperature variation.

Programmable Duty Cycle

The programmable duty cycle feature allows you to set the duty cycle of the PLL clock outputs. The duty cycle is the ratio of the clock output high/low time to the total clock cycle time, which is expressed as a percentage of high time. This feature is supported on all three PLL post-scale counters (G0, G1, and E).

The duty cycle is set by using a low- and high-time count setting for the post-scale counters. The Quartus II software uses the input frequency and target multiply/divide ratio to select the post-scale counter. The precision of the duty cycle is determined by the post-scale counter value chosen on a PLL clock output and is defined as 50% divided by the post-scale counter value. For example, if the post-scale counter value is 3, the allowed duty cycle precision would be 50% divided by 3 equaling 16.67%. Because the `altpll` megafunction does not accept non-integer values for the duty cycle values, the allowed duty cycles are 17, 33, 50, and 67%.

Due to hard limitations, you cannot achieve a duty cycle of 84% because you cannot achieve the closest value to 100% for a given counter value. However, you can achieve a duty cycle of 84% by choosing a 17% duty cycle and inverting the PLL clock output. For example, if the G0 counter is 10, increments of 5% are possible for duty cycle choices between 5 and 90%.

External Clock Output

Each PLL supports one single-ended or LVDS external clock output for general-purpose external clocks, or for source-synchronous transmitters. The output of the E counter drives the PLL external clock output (e0), which can only feed to the PLL[2..1]_OUT pins and not to internal logic. You can use PLL[2..1]_OUT in all three clock feedback modes.



The EP1C3 device in the 100-pin package, and the EP1C6 PLL2 in the 144-pin package, do not have support for an external clock output.

The PLL[2..1]_OUT pins are dual-purpose pins, meaning if the pins are not required by the PLL, they are available for use as general-purpose I/O pins. The I/O standards supported by the PLL[2..1]_OUT pins are listed in Table 6–7.

Table 6–7. Supported I/O Standards for Cyclone PLL Pins

I/O Standard	Inclk	PLL[2..1]_OUT *(1)*
LVTTL	✓	✓
LVCMOS	✓	✓
2.5-V	✓	✓
1.8-V	✓	✓
1.5-V	✓	✓
3.3-V PCI	✓	✓
LVDS *(2)*	✓	✓
SSTL-2 Class I	✓	✓
SSTL-2 Class II	✓	✓
SSTL-3 Class I	✓	✓
SSTL-3 Class II	✓	✓
Differential SSTL-2 Class II	—	✓

Notes to Table 6–7:

(1) The EP1C3 device in the 100-pin TQFP package and the EP1C6 PLL2 in the 144-pin TQFP package do not support an external clock output.

(2) The EP1C3 device in the 100-pin TQFP package does not support an LVDS input.

Since the pllena and locked signal can be driven by or driven to general-purpose I/O pins, respectively, they support all Cyclone I/O standards.

The Cyclone external clock output pins (PLL[2..1]_OUT) do not have a separate V_{CC} and GND bank internal to the device. The PLL[2..1]_OUT pins share a V_{CCIO} bank with neighboring I/O pins. Only the I/O pins in the same bank have an effect on the PLL[2..1]_OUT pins. Therefore, to minimize jitter on the PLL[2..1]_OUT pins, I/O pins directly adjacent to these pins should be either inputs or they should not be used. For more information about board design guidelines, see "Jitter Considerations" on page 6–19.

Control Signals

There are four available control signals, pllena, areset, pfdena, and locked, in Cyclone PLLs that provide added PLL management.

pllena

The PLL enable signal, pllena, enables or disables the PLL. You can either enable/disable a single PLL (by connecting pllena port independently) or multiple PLLs (by connecting pllena ports together). The pllena signal is an active-high signal. When pllena is low, the PLL clock output ports are driven to logic low and the PLL loses lock. All PLL counters, including gated lock counter return to default state. When pllena goes high again, the PLL relocks and resynchronizes to the input clock. Therefore, pllena is an active-high signal. In Cyclone FPGAs, you can feed the pllena port from internal logic or any general-purpose I/O pin because there is no dedicated pllena pin. This feature offers added flexibility, since each PLL can have its own pllena control circuitry, or both PLLs can share the same pllena circuitry. The pllena signal is optional, and when it is not enabled in the software, the port is internally tied to VCC.

areset

The PLL areset signal is the reset or resynchronization input for each PLL. The areset signal should be asserted every time the PLL loses lock to guarantee correct phase relationship between the PLL input and output clocks. Users should include the areset signal in designs where phase relationship between input and output clocks need to be maintained after a loss of lock condition. The areset signal is an active high signal and, when driven high, the PLL counters reset, clearing the PLL output and causing the PLL to lose lock. The clock outputs of the PLL are driven to ground as long as areset is active. When areset transitions low, the PLL will resynchronize to its input clock as the PLL relocks. If the target VCO frequency is below this nominal frequency, the PLL clock output frequency will start at a higher value than desired during the lock process. In this case, Altera recommends monitoring the gated locked signal to ensure the PLL is fully in lock before enabling the clock outputs

from the PLL. Cyclone FPGAs can drive this PLL input signal from LEs or any general-purpose I/O pin. The areset signal is optional. When it is not enabled in the Quartus II software, the port is internally tied to GND.

pfdena

The pfdena signal controls the PFD output in the PLL with a programmable gate. If you disable the PFD by driving areset low, the VCO operates at its last set control voltage and frequency value with some long-term drift to a lower frequency. The VCO frequency can drift up to +/- 5% over 25 us. Even though the PLL clock outputs continue to toggle regardless of the input clock, the PLL could lose lock. The system continues running when the PLL goes out of lock, or if the input clock is disabled. Because the last locked output frequency does not change for some time, you can use the pfdena port as a shutdown or cleanup function when a reliable input clock is no longer available. By maintaining this frequency, the system has time to store its current settings before shutting down. If the pfdena signal goes high again, the PLL relocks and resynchronizes to the input clock. Therefore, the pfdena pin is an active-high signal. You can drive the pfdena input signal by any general-purpose I/O pin, or from internal logic. This signal is optional, and when it is not enabled in the software, the port is internally tied to VCC.

locked

When the locked output is at a logic-high level, this level indicates a stable PLL clock output in phase with the PLL reference input clock. Without any additional circuitry, the locked port may toggle as the PLL begins tracking the reference clock. The locked port of the PLL can feed any general-purpose I/O pin and/or internal logic. This locked signal is optional, but is useful in monitoring the PLL lock process.

Whenever the PLL loses lock for any reason (be it excessive inclk jitter, power supply noise, etc.), the PLL must be reset with the areset signal to guarantee correct phase relationship between the PLL output clocks. If the phase relationship between the input clock versus output clock, and between different output clocks from the PLL is not important in your design, the PLL need not be reset.

Clock Feedback Modes

Cyclone PLLs support three feedback modes: normal, zero delay buffer, and no compensation. Unlike other Altera device families, Cyclone PLLs do not have support for external feedback mode. All three supported

clock feedback modes allow for multiplication/division, phase shifting, and programmable duty cycle. The following sections give a brief description of each mode.



The phase relationship shown in Figure 6–4 through 6–6 are for the default phase shift setting of 0°. Changing the phase-shift setting will change the relationships.

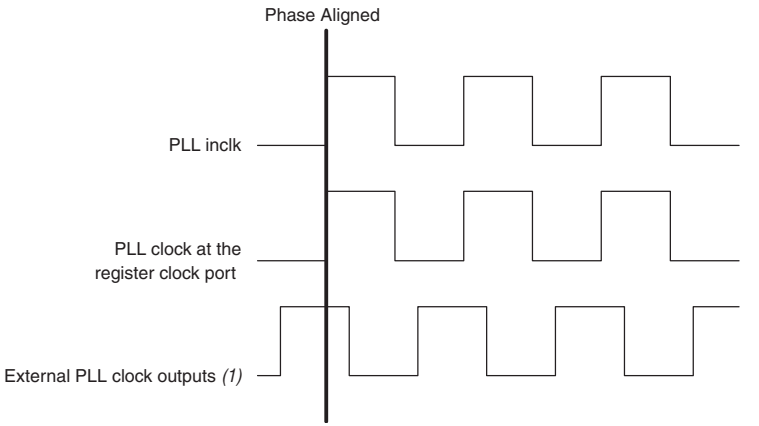
Normal Mode

In normal mode, the PLL phase aligns the input reference clock with the clock signal at the ports of the registers in the logic array or the IOE to compensate for the internal global clock network delay. In the `altpll` MegaWizard Plug-In Manager, you can define which internal clock output from the PLL (`c0` or `c1`) should be compensated.

If the external clock output (`PLL[2..1]_OUT`) is used in this mode, there will be a phase shift with respect to the clock input pin. Similarly, if you use the internal PLL clock outputs to drive general-purpose I/O pins, there will be a phase shift with respect to the clock input pin.

Figure 6–4 shows an example waveform of the PLL clocks' phase relationship in normal mode.

Figure 6–4. Phase Relationship Between PLL Clocks in Normal Mode

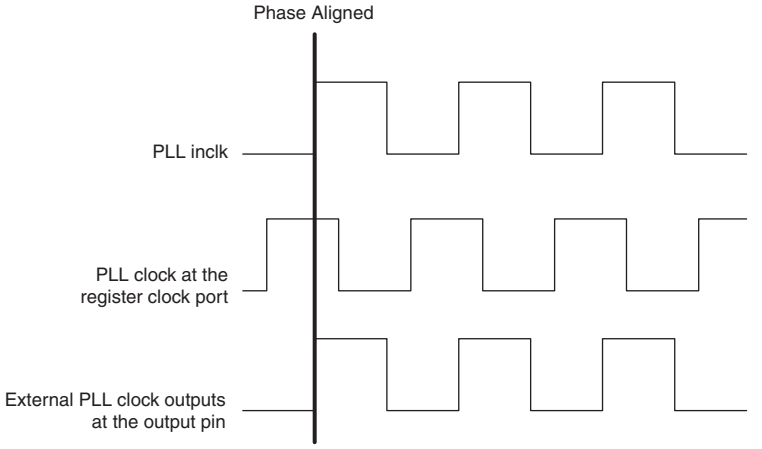


Note to Figure 6–4:
(1) The external clock output can lead or lag the PLL clock signals.

Zero Delay Buffer Mode

The clock signal on the PLL external clock output pin (PLL[2..1]_OUT) is phase-aligned with the PLL input clock pin for zero delay. If you use the c[1..0] ports to drive internal clock ports, there will be a phase shift with respect to the input clock pin. Figure 6–5 shows an example waveform of the PLL clocks' phase relationship in zero delay buffer mode.

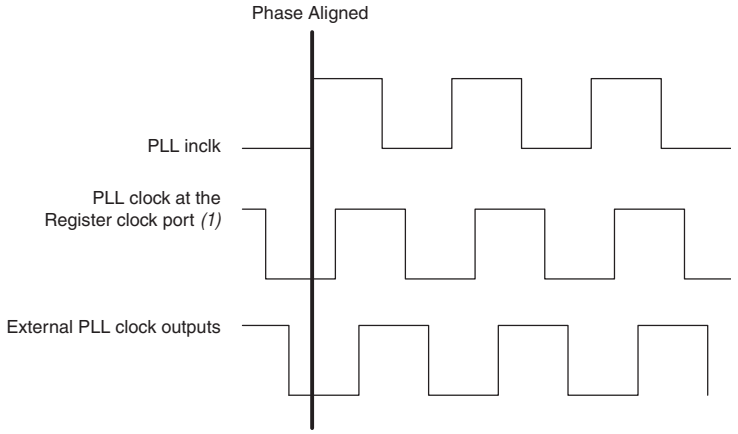
Figure 6–5. Phase Relationship Between PLL Clocks in Zero Delay Buffer Mode



No Compensation

In this mode, the PLL does not compensate for any clock networks, which leads to better jitter performance because the 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 6–6 shows an example waveform of the PLL clocks' phase relationship in no compensation mode.

Figure 6–6. Phase Relationship Between PLL Clocks in No Compensation Mode



Note to Figure 6–6:
(1) Internal clocks fed by the PLL are in phase alignment with each other.

Pins

Table 6–8 describes the Cyclone PLL-related physical pins and their functionality.

Table 6–8. Cyclone PLL Pins (Part 1 of 2)

Pin Name	Description
CLK0	Single-ended or LVDS p-pin that can drive the inclk0 port of PLL1.
CLK1 (1)	Single-ended or LVDS n-pin that can drive the inclk0 port of PLL1.
CLK2	Single-ended or LVDS p-pin that can drive the inclk0 port of PLL2.
CLK3 (1)	Single-ended or LVDS n-pin that can drive the inclk0 port of PLL2.
PLL1_OUTp (2) PLL1_OUTn (2)	Single-ended or LVDS pins driven by the e0 port from PLL1. If not used by the PLL, these are available as general-purpose I/O pins.
PLL2_OUTp (2) PLL2_OUTn (2)	Single-ended or LVDS pins driven by the e0 port from PLL2. If not used by the PLL, these are available as general-purpose I/O pins.
VCCA_PLL1 (3)	Analog power for PLL1. Even if the PLL is not used, you must connect this pin to 1.5 V.
GNDA_PLL1 (4)	Analog ground for PLL1. You can connect this pin to the GND plane on the board.
VCCA_PLL2 (3)	Analog power for PLL2. Even if the PLL is not used, you must connect this pin to 1.5 V.

Table 6–8. Cyclone PLL Pins (Part 2 of 2)	
Pin Name	**Description**
GNDA_PLL2 *(4)*	Analog ground for PLL2. You can connect this pin to the GND plane on the board.
GNDG_PLL1 *(5)*	Guard ring ground for PLL1. You can connect this pin to the GND plane on the board.
GNDG_PLL2 *(5)*	Guard ring ground for PLL2. You can connect this pin to the GND plane on the board.

Notes to Table 6–8:
(1) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins CLK1 and CLK3.
(2) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not support an external clock output.
(3) Refer to "Board Layout" on page 6–17 for filtering and other recommendations.
(4) The EP1C3 device in the 100-pin TQFP package, and the EP1C6 PLL2 in the 144-pin TQFP package do not have a separate GNDA_PLL pin. They are internally tied to GND.
(5) The Guard ring power (VCCG_PLL) is tied internally to V_{CCINT}.

Board Layout

Cyclone PLLs contain analog components that are embedded in a digital device. These analog components have separate power and ground pins to provide immunity against noise generated by the digital components. These separate VCC and GND pins are used to isolate circuitry and improve noise resistance.

V_{CCA} and GNDA

Each PLL has separate VCC and GND pairs for their analog circuitry. The analog circuit power and ground pin for each PLL is called VCCA_PLL# and GNDA_PLL# (# represents the PLL number). Even if the PLL is not used, the V_{CCA} power must be connected to a 1.5-V supply. The power connected to V_{CCA} must be isolated from the power to the rest of the Cyclone FPGA, or any other digital device on the board. The following sections describe three different methods for isolating V_{CCA}.

Separate V_{CCA} Power Plane

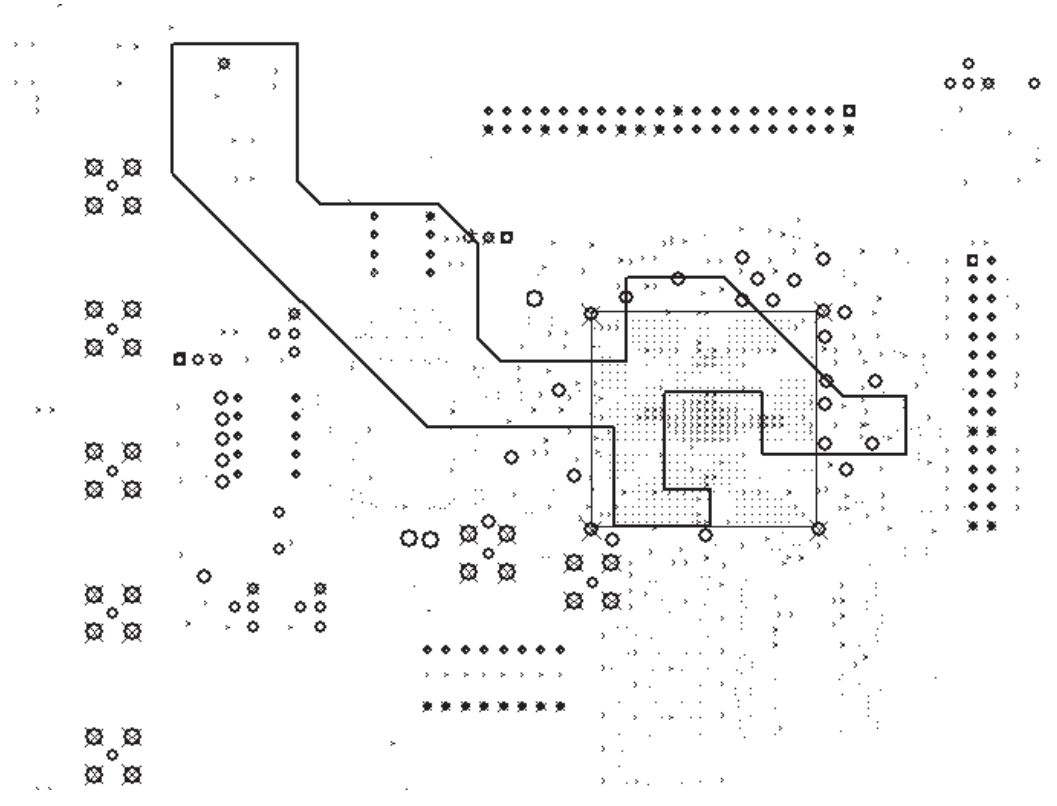
The designer of a mixed-signal system would have already partitioned the system into analog and digital sections, each with its own power planes on the board. In this case, you can connect V_{CCA} to the analog 1.5-V power plane.

Partitioned V_{CCA} Island within V_{CCINT} Plane

Most systems using Altera devices are fully digital, so there is not a separate analog power plane readily available on the board. Adding new planes to the board may be expensive. Therefore, you can create islands

for VCCA_PLL. The dielectric boundary that creates the island is approximately 25 mils thick. Figure 6–7 shows a partitioned plane within V_{CCINT} for V_{CCA}.

Figure 6–7. V_{CCINT} Plane Partitioned for V_{CCA} Island



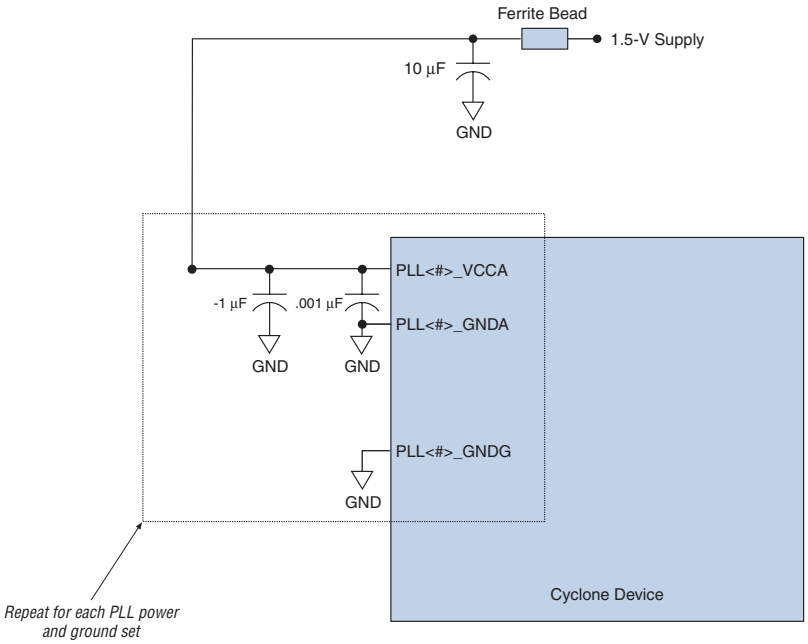
Thick V_{CCA} Traces

Due to board restraints, it may not be possible to partition a V_{CCA} island. Instead, run a thick trace from the power supply to each of the VCCA pins. The traces should be at least 20 mils thick.

In all cases, each VCCA pin must be filtered with a decoupling circuit shown in Figure 6–8. You must place a ferrite bead and a 10-µF tantalum parallel capacitor where the power enters the board. Choose a ferrite bead that exhibits high impedance at frequencies of 50 MHz or higher. Each VCCA pin must be decoupled with a 0.1-µF and a 0.001-µF parallel

combination of ceramic capacitors located as close as possible to the Cyclone FPGA. You can connect the GNDA pins directly to the same GND plane as the digital GND of the device.

Figure 6–8. PLL Power Schematic for Cyclone PLLs



For more information about board design guidelines, refer to *AN 75: High-Speed Board Designs*.

Jitter Considerations

If the input clocks have any low-frequency jitter (below the PLL bandwidth), the PLL attempts to track it, which increases the jitter seen at the PLL clock output. To minimize this effect, avoid placing noisy signals in the same V_{CCIO} bank as those that power the PLL clock input buffer. This is only important if the PLL input clock is assigned to 3.3-V or 2.5-V LVTTL or LVCMOS I/O standards. With these I/O standards, V_{CCIO}

powers the input clock buffer. Therefore, any noise on this V_{CCIO} supply can affect jitter performance. For all other I/O standards the input buffers are powered by V_{CCINT}.

Because Cyclone external clock output pins (PLL[2..1]_OUT) do not have a separate V_{CC} and GND bank, you should avoid placing noisy output signals directly next to these pins. Therefore, Altera recommends that PLL[2..1]_OUT neighboring I/O pins should be either inputs pins or not used at all. If noisy outputs are placed next to the PLL[2..1]_OUT pins, they could inject noise through ground bounce or V_{CC} sag and mutual pin inductance, which would result in worse jitter performance on the PLL[2..1]_OUT pins.

Additionally, you should take into consideration the number of simultaneously switching outputs within the same V_{CCIO} bank as the PLL[2..1]_OUT pins. Altera recommends that you switch as few outputs simultaneously in the same direction as possible in these V_{CCIO} banks. Also, if you have switching outputs in the same V_{CCIO} bank as the PLL[2..1]_OUT pins, Altera recommends that you use the low current strength and/or slow slew rate options on those output pins as they will help to improve the jitter performance.

Specifications

Refer to the *DC and Switching Characteristics* chapter of the *Cyclone Device Handbook* for Cyclone FPGA PLL specifications.

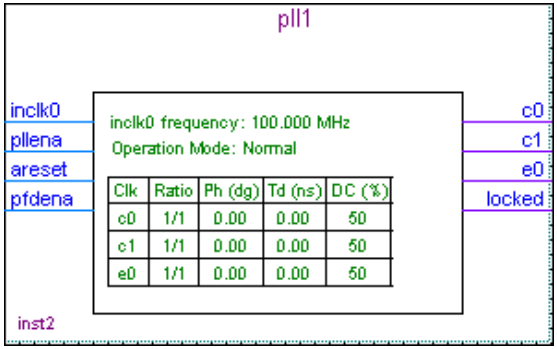
Software Support

Support for Cyclone PLLs is available in the Quartus II software by using the altpll megafunction. The following section describes how the altpll megafunction enables the various Cyclone PLL features and options. This section includes the megafunction symbol, the input and output ports, a description of the MegaWizard Plug-In Manager options, and example MegaWizard screen shots.

Quartus II altpll Megafunction

Figure 6–9 shows the altpll megafunction symbol in the Quartus II software.

Figure 6–9. altpll Megafunction Symbol Targeted for Cyclone FPGAs



Refer to Quartus II Help for the `altpll` megafunction AHDL functional prototypes (applicable to Verilog HDL), VHDL component declaration, and parameter descriptions.

altpll Input Ports

Table 6–9 shows the input ports of the altpll megafunction and describes their function.

Table 6–9. Input Ports of the altpll Megafunction

Port Name	Required	Description
inclk0 *(1)*	Yes	The input clock port that drives the PLL.
pllena *(2)*	No	pllena is an active-high signal, which acts as a combined enable and reset signal for the PLL. You can use it for enabling or disabling one or both PLLs. When this signal is driven low, the PLL clock output ports are driven to GND and the PLL loses lock. Once this signal is driven high again, the lock process begins and the PLL re-synchronizes to its input reference clock. The pllena port can be driven from internal logic or any general-purpose I/O pin.
areset *(2)*	No	areset is an active-high signal, which resets all PLL counters to their initial values. When this signal is driven high, the PLL resets its counters, clears the PLL outputs, and loses lock. Once this signal is driven low again, the lock process begins and the PLL re-synchronizes to its input reference clock. You can drive the areset port from internal logic or any general-purpose I/O pin.
pfdena *(2)*	No	pfdena is an active-high signal, which enables or disables the up/down output signals from the PFD. When pfdena is driven low, the PFD is disabled, while the VCO continues to operate. PLL clock outputs continue to toggle regardless of the input clock, but can experience some long-term drift. Because the output clock frequency does not change for some time, you can use the pfdena port as a shutdown or cleanup function when a reliable input clock is no longer available. You can drive the pfdena port from internal logic or any general-purpose I/O pin.

Notes to Table 6–9:
(1) The inclk0 port to the PLL must be driven by the dedicated clock input pin(s).
(2) See "Control Signals" on page 6–12 for further details.

altpll Output Ports

Table 6–10 shows the output ports of the altpll megafunction and describes their function.

Table 6–10. Output Ports of the altpll Megafunction

Port Name	Required	Description
c[1..0] *(1)*	No	Clock output of the PLL that drives the internal global clock network.
e0 *(1)*	No	Clock output that feeds the external clock output pins, PLL[2..1]_OUT.
locked *(2)*	No	Gives the status of the PLL lock. When the PLL is locked, this port drives logic high. When the PLL is out of lock, this port drives logic low. The locked port can pulse high and low during the PLL lock process.

Notes to Table 6–10:
(1) Either the internal or external clock output of the PLL must be selected.
(2) See "Control Signals" on page 6–12 for further details.

MegaWizard Customization

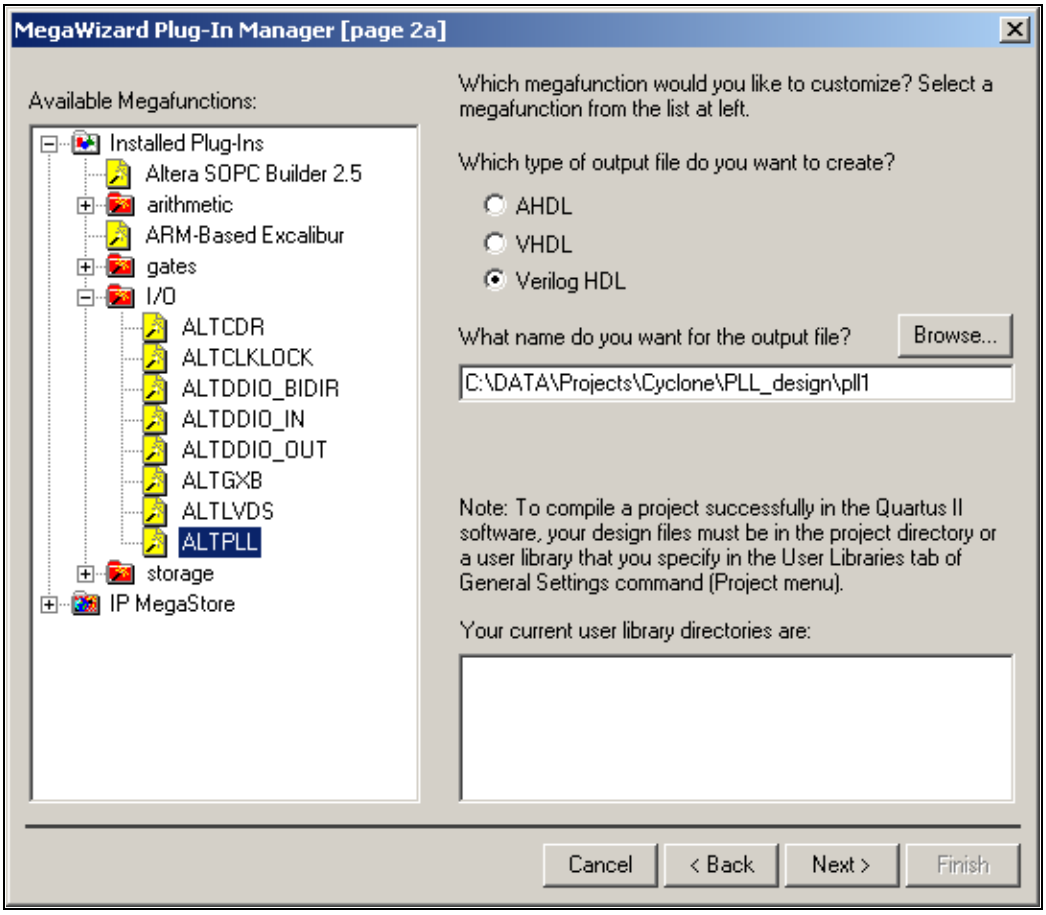
You can use the MegaWizard Plug-In Manager to set the altpll megafunction options for each PLL instance in your design.



If you instantiate the altpll megafunction without using the MegaWizard Plug-In Manager, search for "altpll" in the Quartus II Help for a list of the altpll parameters.

In the MegaWizard Plug-In Manager, select the altpll megafunction in the I/O directory from the **Available Megafunctions** dialog box (see Figure 6–10). The altclklock megafunction is also available from the Quartus II software for backward compatibility, but instantiates the new altpll megafunction when targeting Cyclone FPGAs.

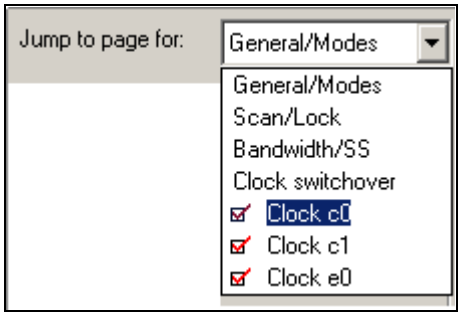
Figure 6–10. altpll Megafunction Selection in the MegaWizard Plug-In Manager



The `altpll` MegaWizard Plug-In Manager has separate pages that apply to Cyclone PLLs. The MegaWizard will gray-out options that are unavailable in Cyclone PLLs. During compilation, theQuartus II Compiler verifies the `altpll` parameters selected against the available PLLs, and any PLL or input clock location assignments.

At the top right-hand corner of each page of the `altpll` MegaWizard Plug-In Manager, there is a **jump to page** drop-down list (see Figure 6–11). This drop-down list allows you to jump to any particular `altpll` MegaWizard page and set those options.

Figure 6–11. Jump to Page Drop-Down List in the altpll MegaWizard Plug-In



MegaWizard Page Description

This section describes the options available on the `altpll` MegaWizard pages. Each of the MegaWizard pages are shown. Tables 6–11 through 6–13 describe the features or settings on that page that apply to Cyclone PLLs. Use these tables, along with the hardware descriptions of the PLL features, to determine appropriate settings for your PLL instance.

You can use the **General/Modes** (Page 1) of the `altpll` MegaWizard Plug-In Manager for selecting the target device family, clock input frequency, general control signal selection, and clock feedback operation mode (see Figure 6–12 and Table 6–11).

Figure 6–12. altpll MegaWizard Plug-In Manager (Page 1)

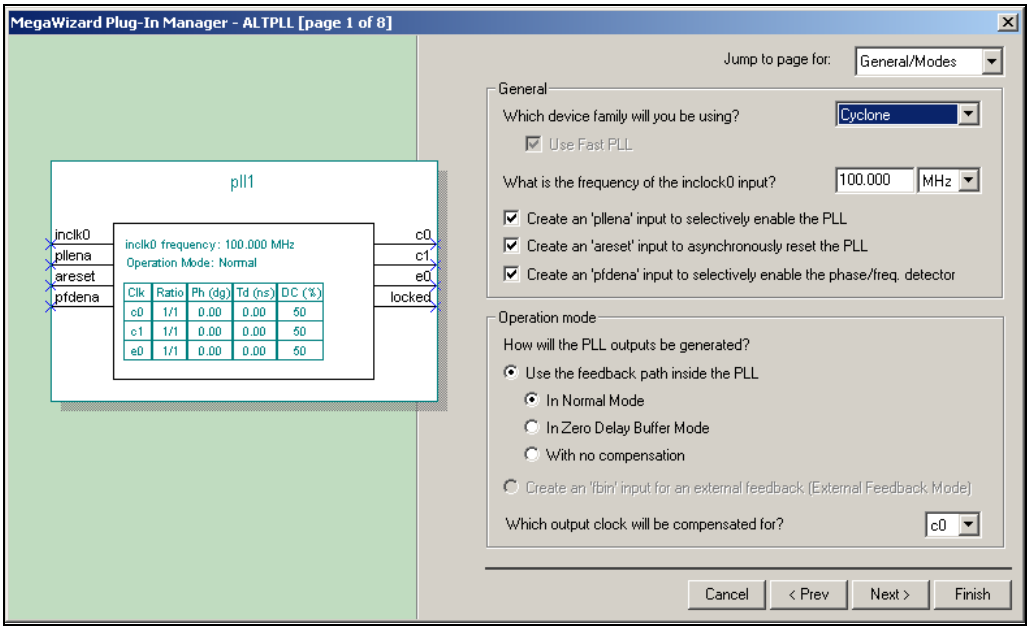


Table 6–11. altpll MegaWizard Plug-In Options Page 1 (Part 1 of 2)

Function	Description
Which device family will you be using?	This chapter explains all `altpll` options that apply when Cyclone is the target device family selected.
What is the frequency of the `inclock0` input	The frequency for the PLL input clock, `inclock0`.
Create an `pllena` input to selectively enable the PLL	Creates a `pllena` port for this PLL instance. See Table 6–9 for `pllena` port description.
Create an `areset` input to asynchronously reset the PLL	Creates a `areset` port for this PLL instance. See Table 6–9 for `areset` port description.
Create an `pfdena` input to selectively enable the PFD	Creates a `pfdena` port for this PLL instance. See Table 6–9 for `pfdena` port description.

Table 6–11. altpll MegaWizard Plug-In Options Page 1 (Part 2 of 2)

Function	Description
Use the feedback path inside the PLL	This option sets the OPERATION_MODE parameter to either normal, zero delay buffer, or no compensation mode. In normal mode, the PLL feedback path comes from a global clock network, which minimizes the clock delay to registers for that specific PLL clock output. You can specify which PLL output is compensated for by using the COMPENSATE_CLOCK parameter. In zero delay buffer mode, the PLL feedback path is confined to the dedicated PLL external output pin. The clock signal driven off-chip on the PLL_OUT pin is phase aligned with the PLL clock input for a minimized delay between clock input and external clock output. If the PLL is also used to drive the internal clock network, a corresponding phase shift of that clock network results. In no compensation mode, the PLL feedback path is confined to the PLL loop; it does not come from the global clock network or an external source. There is no clock network compensation, but this mode minimizes jitter on clocks. This mode may lead to positive hold times on IOE registers; you can use manual phase shifting to compensate for positive hold times. For more information, see "Clock Feedback Modes" on page 6–13.
Which output clock will be compensated?	Indicates which output port of the PLL is compensated. For normal mode, you can select c0 or c1.

You can use **Scan/Lock** (Page 2) for selecting the locked output port (see Figure 6–13 and Table 6–12).

Figure 6–13. altpll MegaWizard Plug-In Manager (Page 2)

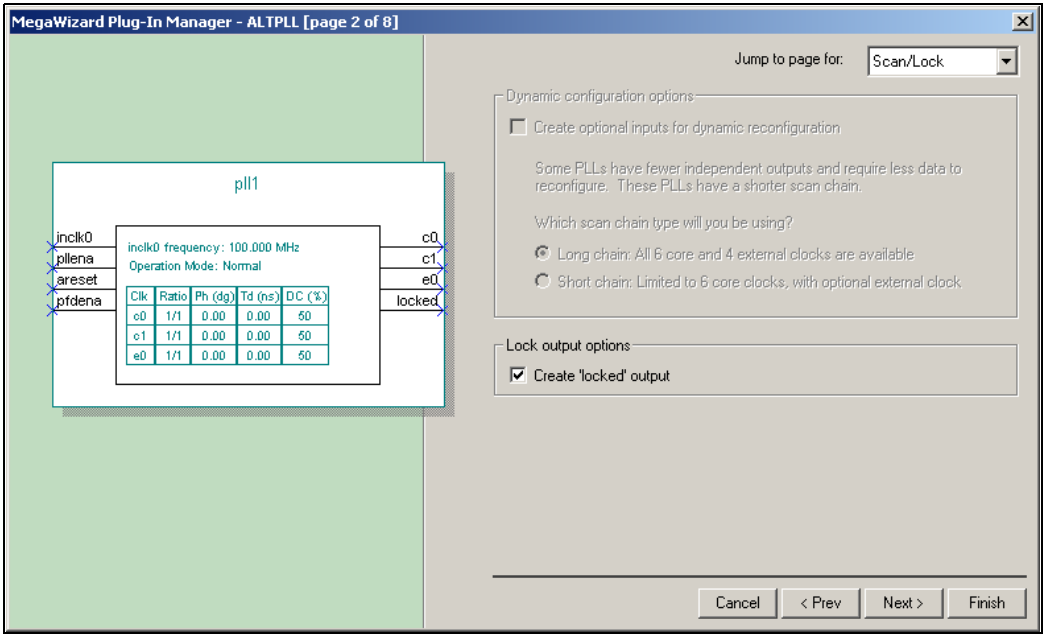
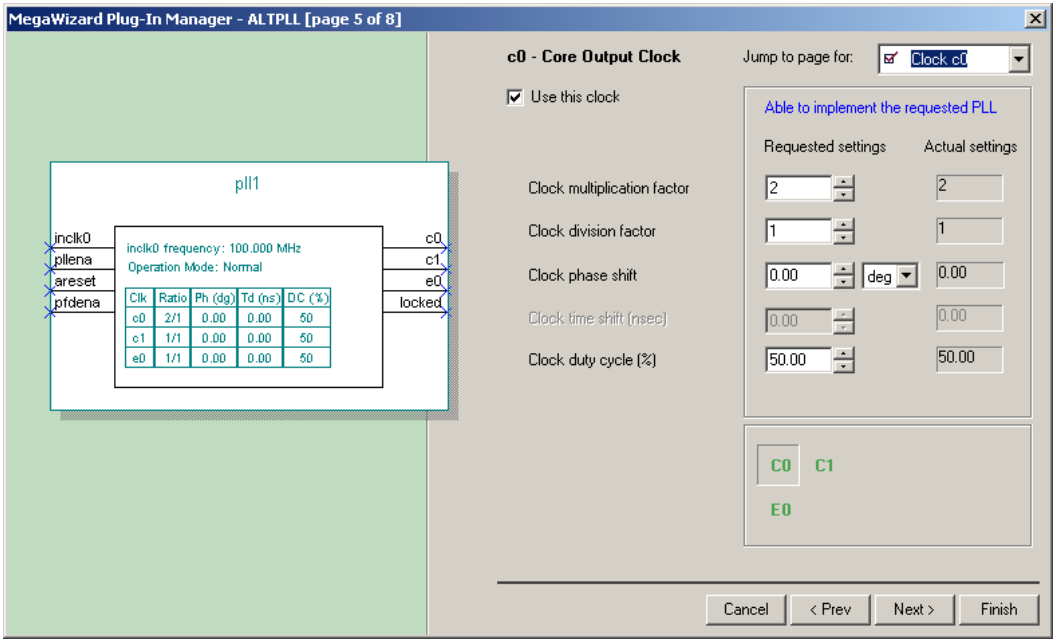


Table 6–12. altpll MegaWizard Plug-In Options Page 2

Function	Description
Create "locked" output	Creates a locked output port to indicate PLL lock. See `locked` port description in Table 6–10.

The options on the next two pages of the MegaWizard Plug-In Manager, (Pages 3 to 4, titled **Bandwidth/SS** and **Clock Switchover**) are not supported in Cyclone FPGAs.

Figure 6–14. altpll MegaWizard Plug-In Manager Pages 5 of 8



The last 3 pages of the MegaWizard Plug-In Manager (Pages 5 to 7) allow you to set the multiplication/division factors, phase shift, and duty cycle for each PLL output port (see Figure 6–14 and Table 6–13).

Each page represents the settings for one PLL clock output port. Table 6–13 describes the options for Pages 5 to 8.

Table 6–13. altpll MegaWizard Plug-In Options Pages 5 of 8

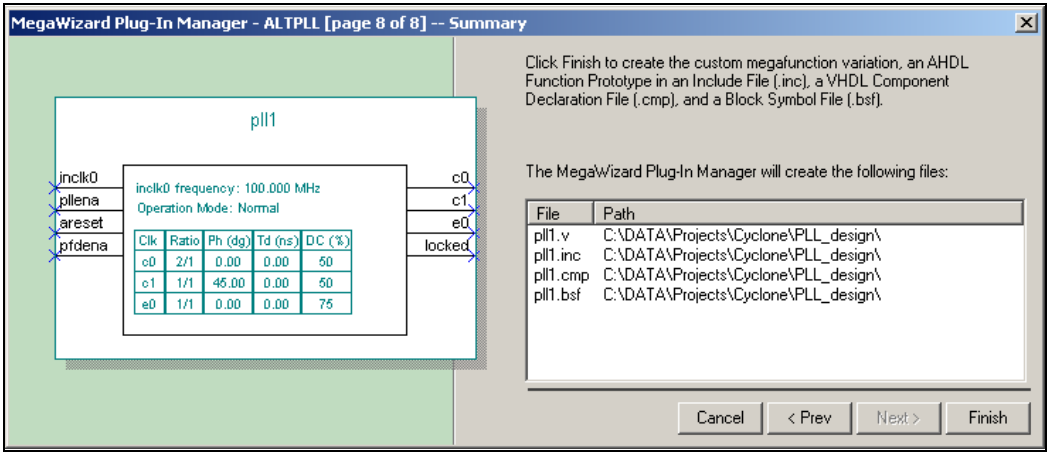
Function	Description
Clock multiplication factor (ratio)	Specifies the clock multiplication for this PLL output. The multiplication factor cannot be greater than 32.
Clock division factor (ratio)	Specifies the clock division for this PLL output.
Clock phase shift (Ph)	Sets the programmable phase shift for the clock output with respect to the PLL clock output that is compensated. The equation to determine the precision of the phase shifting in degrees is (45° divided by the post-scale counter value). Therefore, the maximum step size is 45°, and smaller steps are possible, depending on the multiplication/division ratio necessary on the clock output port. For example, if you have an input clock of 125 MHz with ×1, the post-scale counter G0 is 3. Therefore, the smallest phase shift step is 15°, and additional phase shifting is in 15° increments. The up/down buttons cycle through the possible phase shift settings with the default *M* and post-scale dividers that the MegaWizard Pug-In Manager has chosen for your target frequency and multiplication/division ratio. It is possible to get other granularities of phase shifts if you manually enter a number into the phase shift field. For example, you can override the MegaWizard-chosen values and manually enter 7.5°. The MegaWizard Plug-In Manager verifies this is possible by using *M* = 6 and G0 = 6. The MegaWizard Plug-In Manager tries to achieve the closest phase shift possible. For example, if you enter 10°, the MegaWizard Plug-In Manager verifies that 9° is possible by using *M* = 5 and G0 = 5. For more information, see "Phase Shifting" on page 6–9.
Clock duty cycle (DC)	Specifies the clock duty cycle of the PLL clock output. The equation to determine the precision of the duty cycle is (50% divided by the post-scale counter value). For example, if post-scale counter G0 is 3, the allowed duty cycles are 50% divided by 3, equaling 16.67%. Because the `altpll` megafunction does not accept non-integer values for the duty cycle values, the allowed duty cycles are 17, 33, 50, and 67%. Due to hard limitations, a duty cycle of 84% cannot be achieved because the closest value to 100% cannot be achieved for a given counter value. However, you can achieve a duty cycle of 84% by choosing a 17% duty cycle and inverting the PLL clock output. Use the up/down buttons to cycle through all possible settings. For more information, see "Programmable Duty Cycle" on page 6–10.

Page 8 is the summary page and tells you what files the MegaWizard Plug-In Manager will create (see Figure 6–15).



You can click **Finish** at anytime while in the MegaWizard Plug-In Manager to update the files.

Figure 6–15. altpll MegaWizard Plug-In Manager Page 8

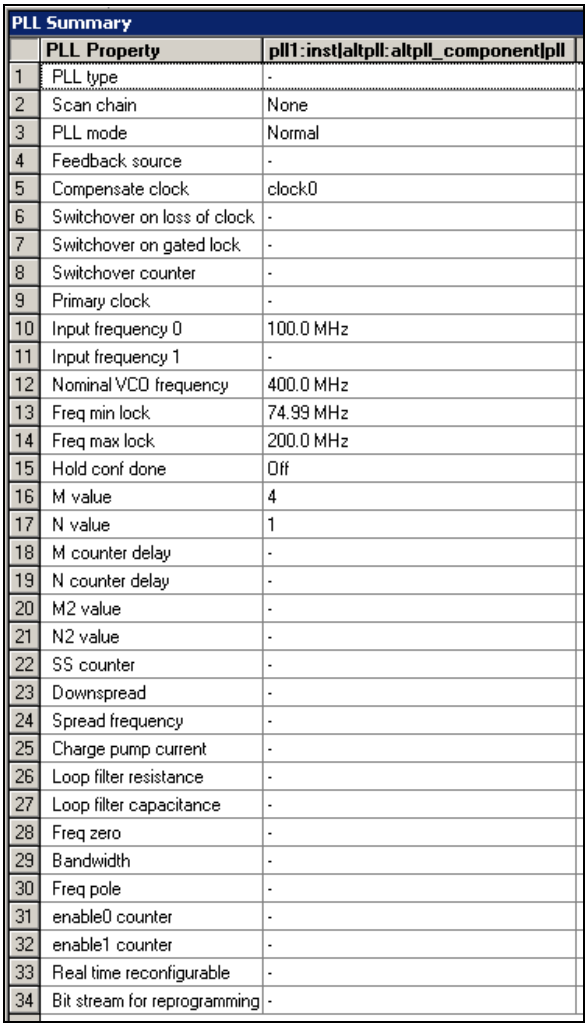


Compilation Report

During compilation, an information message displays whether the requested multiplication/division factors, and/or phase shift, and/or duty cycle were achieved. If you enter an invalid multiplication/division ratio, compilation fails, and the Quartus II software displays an error message. If you enter an invalid phase shift or duty cycle value, the compilation proceeds, and you will receive an information message displaying the best alternative values chosen by the Quartus II software.

The **Resource Section** of the compilation report provides two PLL reports: the **PLL Summary** and the **PLL Usage** reports. The **PLL Summary** provides information on each PLL's parameters (see Figure 6–16). The **PLL Summary** is column-based in the report file, where each column represents a different PLL instance. Table 6–14 lists and explains the parameters shown in the **PLL Summary** report. PLL properties not listed in Table 6–14 do not apply to Cyclone PLLs.

Figure 6–16. PLL Summary Report



PLL Summary

	PLL Property	pll1:inst|altpll:altpll_component|pll
1	PLL type	-
2	Scan chain	None
3	PLL mode	Normal
4	Feedback source	-
5	Compensate clock	clock0
6	Switchover on loss of clock	-
7	Switchover on gated lock	-
8	Switchover counter	-
9	Primary clock	-
10	Input frequency 0	100.0 MHz
11	Input frequency 1	-
12	Nominal VCO frequency	400.0 MHz
13	Freq min lock	74.99 MHz
14	Freq max lock	200.0 MHz
15	Hold conf done	Off
16	M value	4
17	N value	1
18	M counter delay	-
19	N counter delay	-
20	M2 value	-
21	N2 value	-
22	SS counter	-
23	Downspread	-
24	Spread frequency	-
25	Charge pump current	-
26	Loop filter resistance	-
27	Loop filter capacitance	-
28	Freq zero	-
29	Bandwidth	-
30	Freq pole	-
31	enable0 counter	-
32	enable1 counter	-
33	Real time reconfigurable	-
34	Bit stream for reprogramming	-

Table 6–14. PLL Summary in Compilation Report File (Part 1 of 2)

PLL Property	Description
PLL mode	Clock feedback mode
Compensate clock	Indicates which PLL clock output (`clock0`, `clock1`, or `extclock0`) port is compensated
Input frequency 0	Clock input frequency for `inclk0`

Table 6–14. PLL Summary in Compilation Report File (Part 2 of 2)	
PLL Property	**Description**
Nominal VCO frequency	Shows the VCO frequency; $f_{VCO} = f_{IN} \times M/N$
Freq min lock	Shows the minimum PLL input clock frequency for which the current combination of *M*/*N* still provides a valid VCO lock
Freq max lock	Shows the maximum PLL input clock frequency for which the current combination of *M*/*N* still provides a valid VCO lock
M value	*M* counter value
N value	*N* counter value

The **PLL Usage** report shows the breakdown information for each PLL clock output (see Figure 6–17). This report is categorized by PLL clock output ports, such that each row represents a different PLL clock output used in your design. Table 6–15 lists and explains the parameters shown in the **PLL Usage** report file in a row format. PLL parameters not listed in Table 6–15 do not apply to Cyclone PLLs.

Figure 6–17. PLL Usage Report

PLL Usage

	Name	Output Clock	Mult	Div	Output Frequency	Phase Shift	D	Duty Cycle	Counter	C	Counter Value	High / Low	Initial	VCO Tap
1	pll1:inst|altpll:altpll_component|_clk0	clock0	2	1	200.0 MHz	0 (0 ps)	0..	50/50	G1	-	2	1/1 Even	1	0
2	pll1:inst|altpll:altpll_component|_clk1	clock1	1	1	100.0 MHz	45 (1250 ps)	0..	50/50	G0	-	4	2/2 Even	1	4
3	pll1:inst|altpll:altpll_component|_extclk0	extclock0	1	1	100.0 MHz	0 (0 ps)	0..	75/25	E0	-	4	3/1 Even	1	0

Timing Analysis

Table 6–15 shows the usage in the compilation report file.

Table 6–15. PLL Usage in Compilation Report File (Part 1 of 2)	
PLL Parameter	**Description**
`Name`	Indicates the PLL instance name and clock output reported.
`Output Clock`	Indicates the PLL clock output (`clock0`, `clock1`, or `extclock0`) for which the parameter information in this row applies. This is the clock port specified in the MegaWizard Plug-In Manager (`c0`, `c1`, `e0`).
`Mult`	Overall multiplication ratio.
`Div`	Overall division ratio.
`Output Frequency`	Output frequency for this output clock.
`Phase Shift`	Achieved phase shift in degrees and units of time (can differ from user-entered value).

Table 6–15. PLL Usage in Compilation Report File (Part 2 of 2)

PLL Parameter	Description
Duty Cycle	Duty cycle for this clock output.
Counter	Post-scale counter used for this clock output, which counter (G0, G1, E0) feeds the clock output.
Counter Value	Value of post-scale counter.
High/Low	High- and low-time counts that make up the counter value. The ratio of high- and low-counts is directly proportional to the duty cycle.
Initial	Initial value for this post-scale counter (achieves the coarse granularity for phase shifting). Specifies the initial number of VCO cycles before starting the counter.
VCO Tap	VCO tap ranges from 0 to 7 (achieves fine granularity for phase shift in units of 1/8 of the VCO period).

The register-to-register timing for each PLL clock output that drives the logic array is reported with slack. The timing analysis section of the report file provides slack information in a clock requirement line for each PLL clock output.

You can derive f_{MAX} numbers from the slack reporting. The microparameters t_{CO}, t_{SU}, and the path delay are given for a `List Path` command on the Actual Maximum P2P timing in the Slack Report window. You can add and invert these to find the f_{MAX} for that path. See the following equation:

$$f_{MAX} = 1/(<\textit{register to register delay}> - <\textit{clock skew delay}> + <\textit{micro setup delay}> + <\textit{micro clock to output delay}>)$$

During timing analysis for Cyclone designs using PLLs, the project clock settings override the PLL input clock frequency and duty cycle settings. It is important to note the following:

- A warning during compilation reports that the project clock settings override the PLL clock settings.
- The project clock setting overrides the PLL clock settings for timing-driven compilation. When you compile a design with timing-driven compilation turned on, you are overconstraining the design so that the fitter can give you a better f_{MAX} performance. For example, if the PLL is set to output a 150 MHz clock, you can set a project clock setting for 170 MHz so that the fitter tries to achieve a design performance of 170 MHz.

- The Compiler checks the lock frequency range of the PLL. If the frequency specified in the project clock settings is outside the lock frequency range, the PLL clock settings will not be overridden.
- Overriding the PLL clock settings only changes the timing requirements; it does not change the overall multiplication/division and phase delay on each clock output of the PLL. The MegaWizard Plug-In Manager does not use the project clock settings to determine the `altpll` parameters.
- Performing a timing analysis without recompiling your design does not change the programming files. You must recompile your design to update the programming files.
- A Default Required f_{MAX} setting does not override the PLL clock settings. Only individual clock settings will override the PLL clock settings.

This capability is useful when you have configured a Cyclone device and want to see if your timing requirements are met when you feed the PLL a different input clock than what is specified for the PLL parameters. Therefore, this feature allows you to overwrite the PLL input clock frequency settings for timing analysis, meaning you do not have to re-synthesize or re-fit your design. The following procedure allows you to override the PLL input frequency setting and re-generate timing analysis.

1. Choose **Timing Settings** (Project menu).
2. Click on the **Clock Settings** tab.
3. Under **Specify circuit frequency as**, select **Settings for individual clock signals**.
4. Click **New**.
5. In the **New Clock Settings** dialog box, type a <name> for the new clock settings in the **Clock settings** box.
6. If you want to specify timing requirements for an absolute clock, follow these steps:
    a. Under **Relationship to other clock settings**, select **Independent of other clock settings**.
    b. In the **Required fMAX** box, type the required frequency (f_{MAX}) of the clock signal and select a time unit from the list.

c. In the **Duty Cycle** list, specify the required duty cycle for the clock.



Cyclone PLLs accept input clocks with 40 to 60% duty cycle.

d. If you want to include external delays to and from device pins in the f_{MAX} calculations, turn on **Include external delays to and from device pins in fMAX calculations**.

e. Click **OK**.

7. Click **OK** to close the Timing Settings window.
8. Open the **Assignment Organizer** dialog box (Tools menu).
9. Click on the **By Node** tab.
10. Under *Mode*, select **Edit specific entity & node settings for**.
11. If necessary, copy a specific PLL input clock pin name to the **Name** box using the **Node Finder** dialog box.
12. Under **Assignment Categories**, click the + icon next to **Timing**.
13. Click on **Click here to add a new assignment**.
14. Under **Assignment**, select **Clock Settings** in the **Name** list, and select the <name> of the clock settings you created in step 5.
15. Under **Stored in assignments for**, select **This instance only**, **This instance in all occurrences of its parent entity**, or **Other**.
16. Click **Add**.
17. Click **OK** or **Apply**.
18. Select **Start Timing Analysis (**Processing Menu).

Simulation

The altpll megafunction supports behavioral and timing simulation in both the Quartus II software and supported third-party simulation tools. You can simulate all digital aspects of the PLL, but none of the analog aspects. Simulation supports all control signals and clock outputs. Table 6–16 explains the simulation support for altpll.

Table 6–16. altpll Simulation Support for Cyclone FPGAs

Feature	Simulation Support
pllena	The pllena signal is modeled. When this signal is driven low, the PLL loses lock and the PLL clock outputs are driven to logic low.
areset	The areset signal is modeled. When this signal is driven high, the PLL loses lock and the PLL clock outputs are driven to logic low. Frequency over-shoot on the PLL clock outputs is not modeled.
pfdena	The pfdena control signal is modeled. When this signal is driven low, the PLL's locked output is undefined and the PLL clock outputs continue to toggle at their last set frequency. The finite frequency long-term drift of the VCO is not modeled.
locked	The locked signal is modeled for a high-bandwidth condition only. The PLL locks or relocks within 2 to 10 cycles during simulation, and does not necessarily reflect the real lock time.
Frequency input change	If the input frequency of the PLL is changed in simulation, the model checks that $f_{IN} \times (M/N)$ is within the VCO frequency range and loses lock if outside the VCO operating range.
Jitter	Jitter is not modeled in simulation.

You can use the altpll behavioral model to simulate the Cyclone PLLs. The Cyclone behavioral model instantiation must follow the same guidelines and restrictions as the design entry. The altpll behavioral and timing models do not simulate jitter, lock time, or VCO drift.

The behavioral models for altpll reside in the **\quartus\eda\sim_lib directory**. **ALTERA_MF.VHD** contains the VHDL behavioral models and can be used for Cyclone designs that instantiate altpll. **ALTERA_MF.v** contains the Verilog HDL behavioral models. The behavioral model does not perform parameter error checking, and you must specify only valid values.



You must set the resolution of the simulator to units of pico seconds (ps) to simulate the model successfully. A larger resolution rounds off the calculations, providing incorrect results.

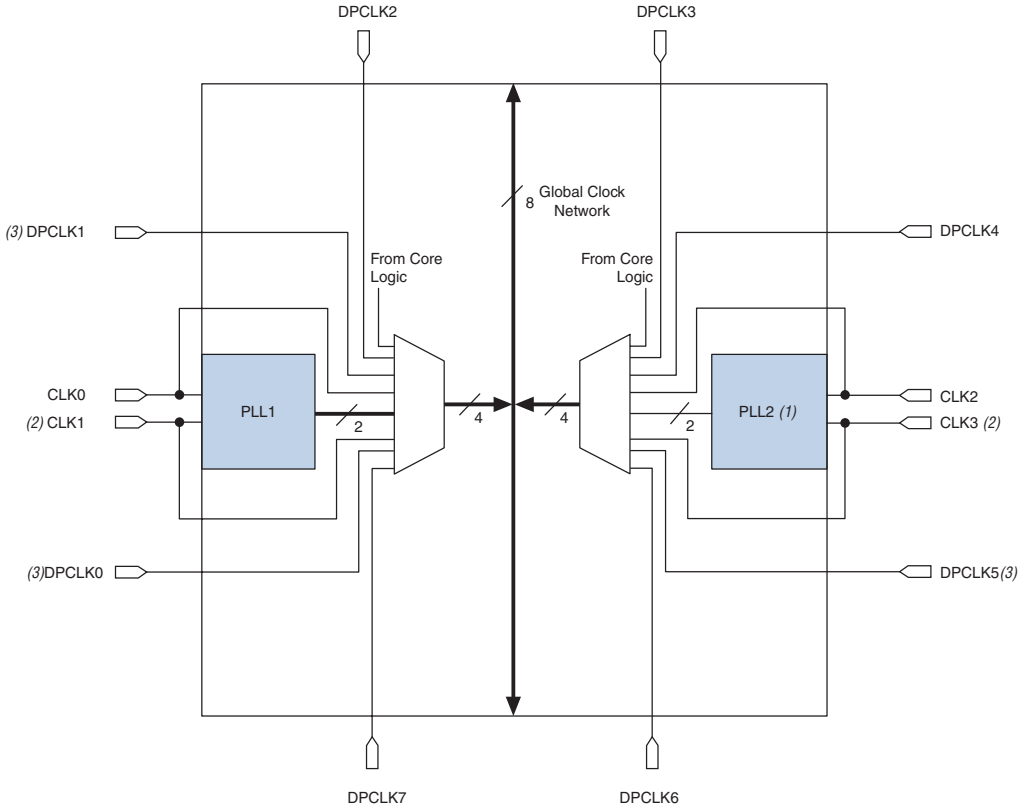
Global Clock Network

Cyclone FPGAs have eight global clock networks. The four dedicated clock input pins (CLK[3..0]), eight dual-purpose clock pins (DPCLK[7..0]), and PLL clock outputs can drive the global clock networks. In addition, internal logic for internally-generated global clocks and asynchronous clears, clock enables, or other control signals with large fanout can drive the global clock networks.

The eight global clock lines that comprise the global clock network drive throughout the entire device. You can use the global clock network as clock sources for all device resources, including IOEs, logic elements (LEs), and memory blocks. You can also use global clock resources for control signals, such as clock enables and synchronous or asynchronous clears fed from external pins.

Figure 6–18 shows the global clock network resources.

Figure 6–18. Global Clock Generation



Notes to Figure 6–18:

(1) The EP1C3 device contains PLL1 only.

(2) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins `CLK1` and `CLK3`.

(3) The EP1C3 device in the 100-pin TQFP package has five `DPCLK` pins (`DPCLK2`, `DPCLK3`, `DPCLK4`, `DPCLK6`, and `DPCLK7`). For more information, see "“Dual-Purpose Clock I/O Pins” on page 6–40.

Dedicated Clock Input Pins

Cyclone FPGAs have up to four dedicated clock input pins (CLK[3..0], two on the left and right side of the device. You can use the CLK[3..0] pins to drive the PLLs, or directly drive them onto the global clock network. Table 6–17 shows which clock pins drive which global clock network.

Table 6–17. Dedicated Clock Input Pin Connections to Global Clock Network

Clock Input Pin	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
CLK0	✓	—	✓	—	—	—	—	—
CLK1 *(1)*	—	✓	—	✓	—	—	—	—
CLK2	—	—	—	—	✓	—	✓	—
CLK3 *(1)*	—	—	—	—	—	✓	—	✓

Note to Table 6–17:
(1) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins CLK1 and CLK3.

Dual-Purpose Clock I/O Pins

Cyclone FPGAs can have up to eight dual-purpose clock pins, DPCLK[7..0] (two on each side of the device). These dual-purpose pins can connect to the global clock network. You can use the DPCLK[7..0] pins for high fanout control signals, such as asynchronous clears, presets, clock enables, or protocol control signals (e.g., TRDY and IRDY for PCI, or DQS signals for external memory interfaces). These pins are also available as general-purpose I/O pins, meaning they can be inputs, outputs, or bidirectional pins. Table 6–18 shows which dual-purpose clock pins drive which global clock network in Cyclone FPGAs.

Table 6–18. Dual-Purpose Clock I/O Connections to the Global Clock Network (Part 1 of 2)

Dual-Purpose Clock Pin	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
DPCLK0 *(1)*	—	—	—	✓	—	—	—	—
DPCLK1 *(1)*	—	—	✓	—	—	—	—	—
DPCLK2	✓	—	—	—	—	—	—	—
DPCLK3	—	—	—	—	✓	—	—	—
DPCLK4	—	—	—	—	—	—	✓	—

Table 6–18. Dual-Purpose Clock I/O Connections to the Global Clock Network (Part 2 of 2)

Dual-Purpose Clock Pin	GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
DPCLK5 *(1)*	—	—	—	—	—	—	—	✓
DPCLK6	—	—	—	—	—	✓	—	—
DPCLK7	—	✓	—	—	—	—	—	—

Note to Table 6–18:
(1) The EP1C3 device in the 100-pin TQFP package does not have the DPCLK0, DPCLK1, or DPCLK5 pins.

Combined Sources

Table 6–19 shows which combined sources drive which global clock network.

Table 6–19. Global Clock Network Sources (Part 1 of 2)

Source		GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
PLL Counter Outputs	PLL1 G0	—	✓	✓	—	—	—	—	—
	PLL1 G1	✓	—	—	✓	—	—	—	—
	PLL2 G0 *(1)*	—	—	—	—	—	✓	✓	—
	PLL2 G1 *(1)*	—	—	—	—	✓	—	—	✓
Dedicated Clock Input Pins	CLK0	✓	—	✓	—	—	—	—	—
	CLK1 *(2)*	—	✓	—	✓	—	—	—	—
	CLK2	—	—	—	—	✓	—	✓	—
	CLK3 *(2)*	—	—	—	—	—	✓	—	✓

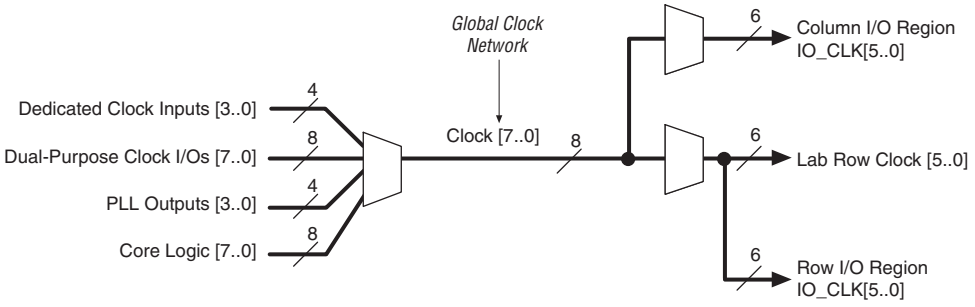
Table 6–19. Global Clock Network Sources (Part 2 of 2)

Source		GCLK0	GCLK1	GCLK2	GCLK3	GCLK4	GCLK5	GCLK6	GCLK7
Dual-Purpose Clock Pins	DPCLK0	—	—	—	✓	—	—	—	—
	DPCLK1 *(3)*	—	—	✓	—	—	—	—	—
	DPCLK2	✓	—	—	—	—	—	—	—
	DPCLK3	—	—	—	—	✓	—	—	—
	DPCLK4	—	—	—	—	—	—	✓	—
	DPCLK5	—	—	—	—	—	—	—	✓
	DPCLK6	—	—	—	—	—	✓	—	—
	DPCLK7	—	✓	—	—	—	—	—	—

Notes to Table 6–19:
(1) The EP1C3 device only has PLL1.
(2) The EP1C3 device in the 100-pin TQFP package does not have dedicated clock pins CLK1 and CLK3.
(3) The EP1C3 device does not have DPCLK1.

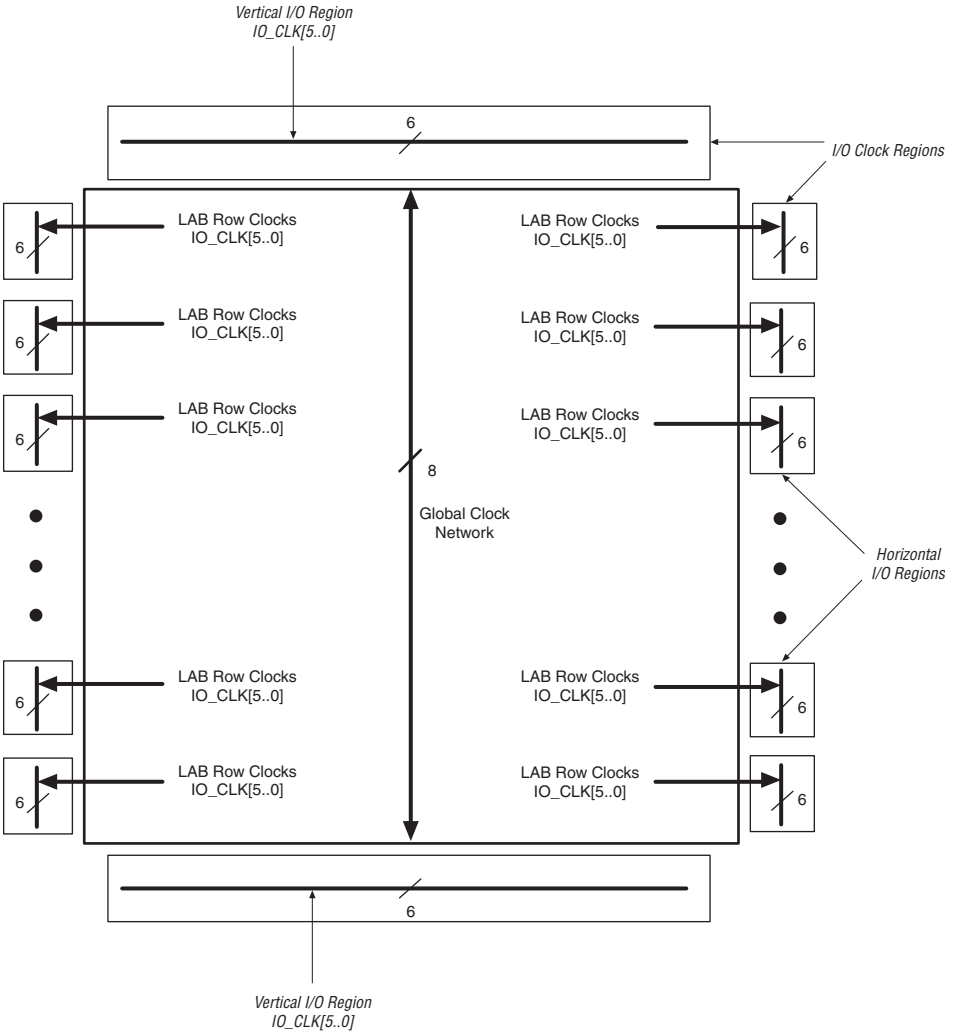
In the Cyclone FPGA, there are eight distinct dedicated global clock networks. Multiplexers are used with these clocks to form six-bit buses to drive LAB row clocks, column IOE clocks, or row IOE clocks (see Figure 6–19). Another multiplexer is used at the LAB level to select two of the six row clocks to feed the LE registers within the LAB.

Figure 6–19. Global Clock Network Multiplexers



IOE clocks have horizontal (row) and vertical (column) block regions that are clocked by six I/O clock signals chosen from the eight global clock resources. Figure 6–20 shows the I/O clock regions.

Figure 6–20. I/O Clock Regions



Conclusion

Cyclone PLLs provide significant features such as $M/(N \times \text{post-scale})$ multiplication/division, phase shift, and programmable duty cycle for your cost-sensitive clock synthesis applications. The reduction in clock delay, and the elimination of clock skew within the device, improves design speed. Cyclone PLL features simplify board design by running the internal logic of the device at a faster rate than the input clock frequency.

Referenced Documents

This chapter references the following documents:

- *AN 75: High-Speed Board Designs*
- *DC and Switching Characteristics* chapter of the *Cyclone Device Handbook*

Document Revision History

Table 6–20 shows the revision history for this chapter.

Table 6–20. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.5	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.4	• Added document revision history. • Updated information about `plena` signal in "Control Signals" section. • Updated "Zero Delay Buffer Mode" section. • Updated Figure 6–5.	—
August 2005 v1.3	Minor updates.	—
October 2003 v1.2	Updated phase shift information.	—
July 2003 v1.1	Updated input and output frequency specifications.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

Section III. Memory

This section provides information on the M4K embedded memory blocks internal to Cyclone devices.

It contains the following:

- Chapter 7. On-Chip Memory Implementations Using Cyclone Memory Blocks

Revision History

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

7. On-Chip Memory Implementations Using Cyclone Memory Blocks

C51007-1.4

Introduction

Cyclone® devices feature embedded memory blocks that can be easily configured to support a wide range of system requirements. These M4K memory blocks present a very flexible and fast memory solution that you can use to provide excellent memory bandwidth and density for a host of cost-sensitive applications.

You can use M4K memory blocks in various memory modes, including single-port, simple dual-port, true dual-port (also known as bidirectional dual-port), shift-register, ROM, and first-in first-out (FIFO) mode. M4K memory blocks also include advanced features such as support for byte-enable operation, parity-bit-based error correction, and mixed-port widths. This chapter describes these modes and other characteristics of the M4K memory blocks.

M4K Memory Features

Table 7–1 summarizes the features supported by the M4K memory block.

Table 7–1. Summary of M4K Memory Features (Part 1 of 2)

Performance	250 MHz
Total RAM bits (including parity bits)	4,608
Configurations	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36 *(1)*
Parity bits	✓
Byte enable	✓
Single-port memory	✓
Simple dual-port memory	✓
True dual-port memory	✓
Embedded shift register	✓
ROM	✓

Table 7–1. Summary of M4K Memory Features (Part 2 of 2)	
FIFO buffer	✓
Simple dual-port mixed width support	✓
True dual-port mixed width support	✓
Memory initialization (.mif)	✓
Mixed-clock mode	✓
Power-up condition	Outputs cleared
Register clears	Input and output registers *(2)*
Same-port read-during-write	New data available at positive clock edge
Mixed-port read-during-write	Outputs set to unknown or old data

Notes to Table 7–1:

(1) The Altera® Quartus® II software will automatically cascade or concatenate multiple M4K memory blocks to provide deeper or wider memory functions.

(2) Asserting the clear port of the `rden` and byte-enable registers drives the output of these registers high.

Table 7–2 shows the memory capacity for M4K memory blocks in each Cyclone device.

Table 7–2. M4K Memory Distribution in Cyclone Devices			
Device	**Columns**	**Blocks**	**Total RAM Bits**
EP1C3	1	13	59,904
EP1C4	1	17	78,336
EP1C6	1	20	92,160
EP1C12	2	52	239,616
EP1C20	2	64	294,912

Parity Bit Support

M4K memory blocks support an optional parity bit for each data byte. Of the 4,608 bits of storage space available in an M4K block, 512 are available for use as parity-bit storage. The parity bit, along with logic implemented in logic elements (LEs), can facilitate parity-checking methods of error detection to ensure data integrity. You can also use parity-size data words to store user-specified control bits or as extra data bits to provide support for 9-bit, 18-bit, or 36-bit wide memories.

Byte-Enable Support

Byte-enable signals can be used to mask the input data so that only specific bytes in memory are overwritten. The unwritten bytes retain the data value that was last written to them. The write-enable signal (`wren`) is used in conjunction with byte-enable signals (`byteena`) to control the M4K block's write operations. The default value for the `byteena` signal is high (enabled), in which case no bytes are masked and writing is controlled only by the `wren` signals.

Asserting the clear port of the byte-enable register drives the byte-enable signal to its default high level.

M4K blocks support byte write operations when the write port has a data width of 16, 18, 32, or 36 bits. Table 7–3 summarizes how `byteena` controls which bits are masked.

Table 7–3. Byte Enable for M4K Blocks *Notes (1), (2)*

byteena	datain × 18	datain × 36
[0] = 1	[8..0]	[8..0]
[1] = 1	[17..9]	[17..9]
[2] = 1	—	[26..18]
[3] = 1	—	[35..27]

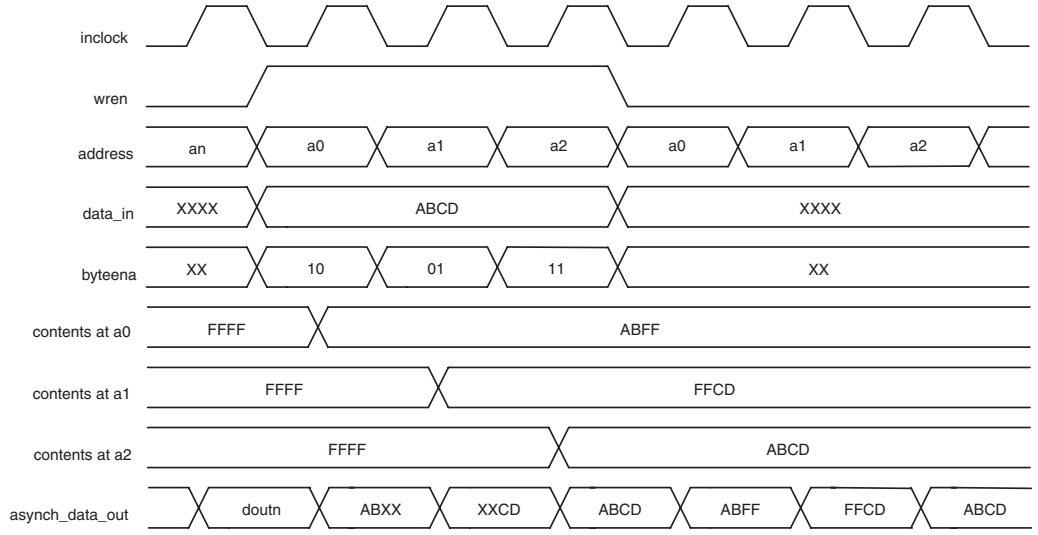
Notes to Table 7–3:

(1) Any combination of byte-enable signals is possible.

(2) Byte enables can be used in the same manner with 8-bit words, i.e., in × 16 and × 32 modes.

Figure 7–1 shows how both the `wren` and the `byteena` signals control the write operations of the RAM.

Figure 7–1. Byte-Enable Operation Functional Waveform



Power-up Conditions and Memory Initialization

Upon power-up, M4K memory is in an idle state. The outputs always power-up to zero, regardless of whether the output registers are used or bypassed. Even if a memory initialization file is used to pre-load the contents of the RAM block, the outputs will still power-up cleared. For example, if address 0 is pre-initialized to FF, the M4K blocks power-up with the output at 00.

Using M4K Memory

M4K memory blocks include input registers that synchronize write operations and output registers to pipeline designs and improve system performance. All M4K memory blocks are fully synchronous, meaning that all inputs are registered, but outputs can be either registered or combinatorial. M4K memory can emulate asynchronous memory.

Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

For more information, refer to *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs*.

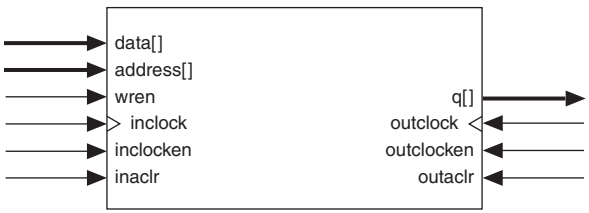
M4K memory blocks can operate in various modes, including:

- Single-port
- Simple dual-port
- True dual-port (bidirectional dual-port)
- Shift-register
- ROM
- FIFO

Implementing Single-Port Mode

Single-port mode supports non-simultaneous read and write operations. Figure 7–2 shows the single-port memory configuration for M4K blocks.

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



Note to Figure 7–2:
(1) Two single-port memory blocks can be implemented in a single M4K block.

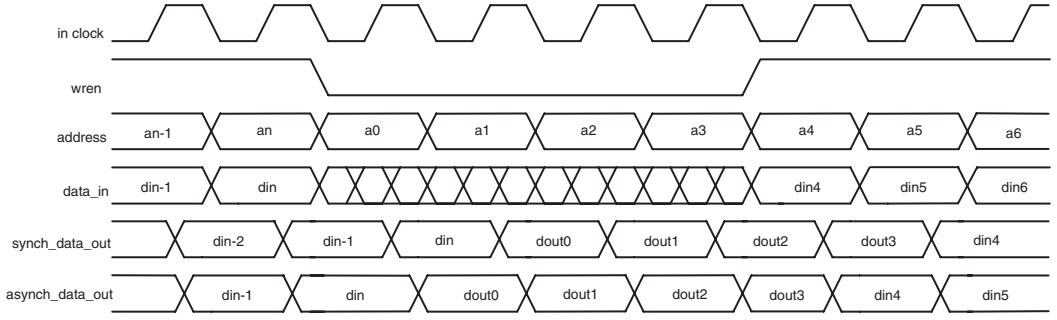
M4K memory blocks can also be divided in half and used for two independent single-port RAM blocks. The Quartus II software automatically uses this method of single-port memory packing when running low on memory resources. When deliberately assigning two single-port memories to one M4K block, first ensure that each of the two independent RAM blocks is equal to or less than half the size of the M4K block.

In the single-port RAM configuration, the outputs can only be in read-during-write mode, which means that during the write operation, data written to the RAM flows through to the RAM outputs. When the output registers are bypassed, the new data is available on the rising edge of the same clock cycle on which it was written.

For more information about read-during-write mode, see "Read-during-Write Operation at the Same Address" on page 7–20.

Figure 7–3 shows timing waveforms for read and write operations in single-port mode.

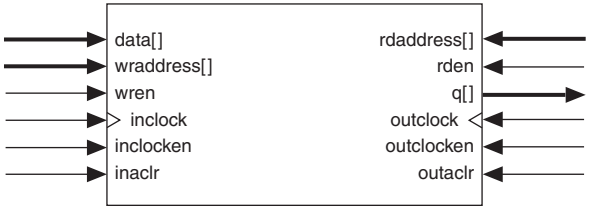
Figure 7–3. Single-Port Timing Waveforms



Implementing Simple Dual-Port Mode

Simple dual-port memory supports simultaneous read and write operations. Figure 7–4 shows the simple dual-port memory configuration for M4K blocks.

Figure 7–4. Simple Dual-Port Memory *Note (1)*



Note to Figure 7–4:
(1) Simple dual-port RAM supports read/write clock mode in addition to the input/output clock mode shown.

M4K memory supports mixed-width configurations, allowing different read and write port widths. This capability is useful for many applications, including implementing serializer-deserializers (SERDES) as well as interfacing with buses of differing widths. Table 7–4 shows the mixed-width configurations supported by the M4K blocks in Cyclone devices.

Table 7–4. M4K Block Mixed-Width Configurations (Simple Dual-Port Mode)

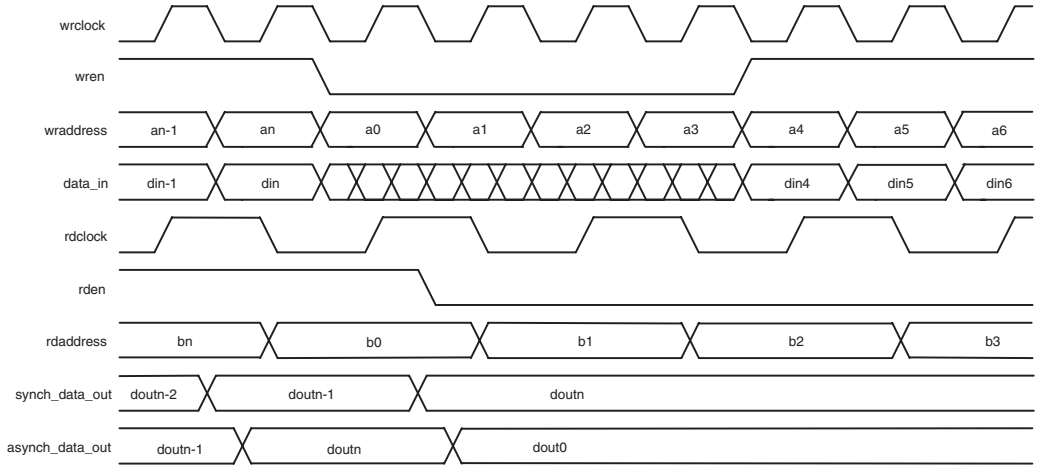
Read Port	Write Port								
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	128 × 32	512 × 9	256 × 18	128 × 36
4K × 1	✓	✓	✓	✓	✓	✓	—	—	—
2K × 2	✓	✓	✓	✓	✓	✓	—	—	—
1K × 4	✓	✓	✓	✓	✓	✓	—	—	—
512 × 8	✓	✓	✓	✓	✓	✓	—	—	—
256 × 16	✓	✓	✓	✓	✓	✓	—	—	—
128 × 32	✓	✓	✓	✓	✓	✓	—	—	—
512 × 9	—	—	—	—	—	—	✓	✓	✓
256 × 18	—	—	—	—	—	—	✓	✓	✓
128 × 36	—	—	—	—	—	—	✓	✓	✓

In simple dual-port mode, M4K blocks have one write-enable and one read-enable signal. On the M4K block, asserting the clear port of the `rden` register drives `rden` high, which allows the read operation to occur. When the read-enable signal is deactivated, the current data is retained at the output ports. If the read-enable signal is activated during a write operation with the same address location selected, the simple dual-port RAM output is either unknown or can be set to output the old data stored at the memory address.

For more information, see "Read-during-Write Operation at the Same Address" on page 7–20.

Figure 7–5 shows timing waveforms for read and write operations in simple dual-port mode.

Figure 7–5. Simple Dual-Port Timing Waveforms

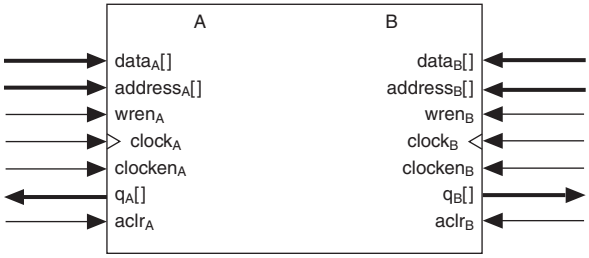


Implementing True Dual-Port Mode

M4K blocks offer a true dual-port mode to support any combination of two-port operations: two read operations, two write operations, or one read operation and one write operation at two different clock frequencies. True dual-port memory can be used to increase memory bandwidth in numerous applications. An example system that benefits from the use of true dual-port memory is a system containing an Altera Nios® embedded processor and a direct memory access (DMA) controller. Such a system will experience bottlenecks if the processor and the DMA controller need simultaneous access to single-port memory. The ability of both the processor and the DMA controller to access the M4K memory simultaneously, avoiding the need for arbitration, can dramatically improve bandwidth in this type of system.

Figure 7–6 shows the true dual-port memory configuration for M4K blocks.

Figure 7–6. True Dual-Port Memory *Note (1)*



Note to Figure 7–6:

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

The widest bit configuration of a single M4K block in true dual-port mode is 256 × 16-bit (or 256 × 18-bit with parity). The 128 × 32-bit (128 × 36-bit with parity) configuration of the M4K block is unavailable because the number of output drivers is equivalent to the maximum bit width of the M4K block. Because true dual-port RAM has outputs on two ports, the maximum width of the true dual-port RAM equals half of the total number of output drivers. However, multiple M4K blocks can be concatenated to support wider memory configurations. Table 7–5 lists the possible M4K RAM block configurations.

Table 7–5. M4K Block Mixed-Port Width Configurations (True Dual-Port Mode)

Port A	Port B						
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	512 × 9	256 × 18
4K × 1	✓	✓	✓	✓	✓	—	—
2K × 2	✓	✓	✓	✓	✓	—	—
1K × 4	✓	✓	✓	✓	✓	—	—
512 × 8	✓	✓	✓	✓	✓	—	—
256 × 16	✓	✓	✓	✓	✓	—	—
512 × 9	—	—	—	—	—	✓	✓
256 × 18	—	—	—	—	—	✓	✓

In true dual-port mode, the RAM outputs can only be configured for read-during-write mode. This means that during write operation, data being written to the A or B port of the RAM flows through to the A or B outputs, respectively. When the output registers are bypassed, the new data is available on the rising edge of the same clock cycle it was written on.

For sample waveforms and other information on mixed-port read-during-write mode, see "Read-during-Write Operation at the Same Address" on page 7–20.

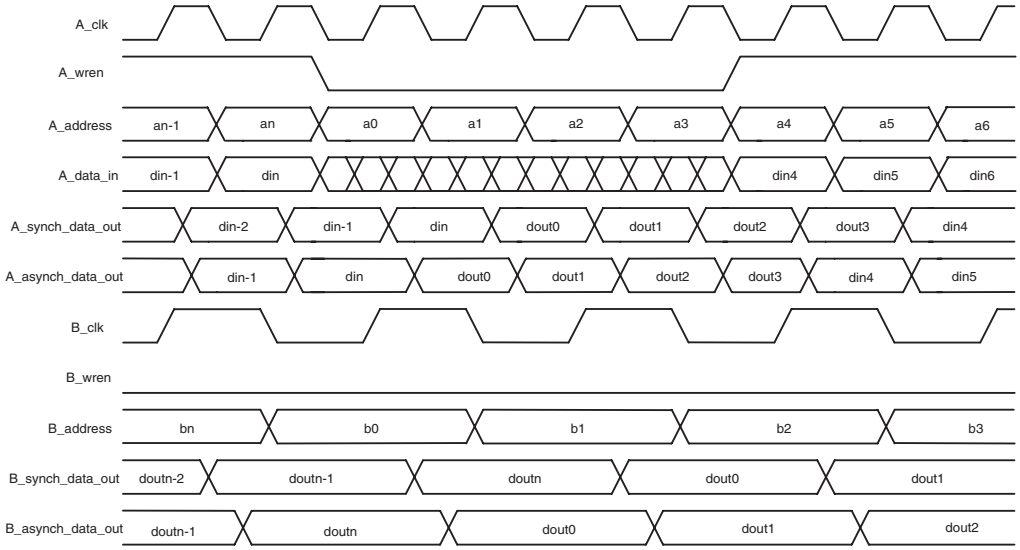
Potential write conflicts must be resolved external to the RAM because simultaneously writing to the same address location at both ports results in unknown data storage at that location. For a valid write operation to the same address of the RAM block, the rising edge of the write clock for port A must occur following the minimum write cycle time interval after the rising edge of the write clock for port B. Since data is written into the M4K blocks at the falling edge of the write clock, the rising edge of the write clock for port A should occur following half of the minimum write cycle time interval after the falling edge of the write clock for port B. If this timing is not met, the data stored in that particular address will be invalid.



For more information about the minimum synchronous write cycle time, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

Figure 7–7 shows true dual-port timing waveforms for a write operation at port A and a read operation at port B.

Figure 7–7. True Dual-Port Timing Waveforms



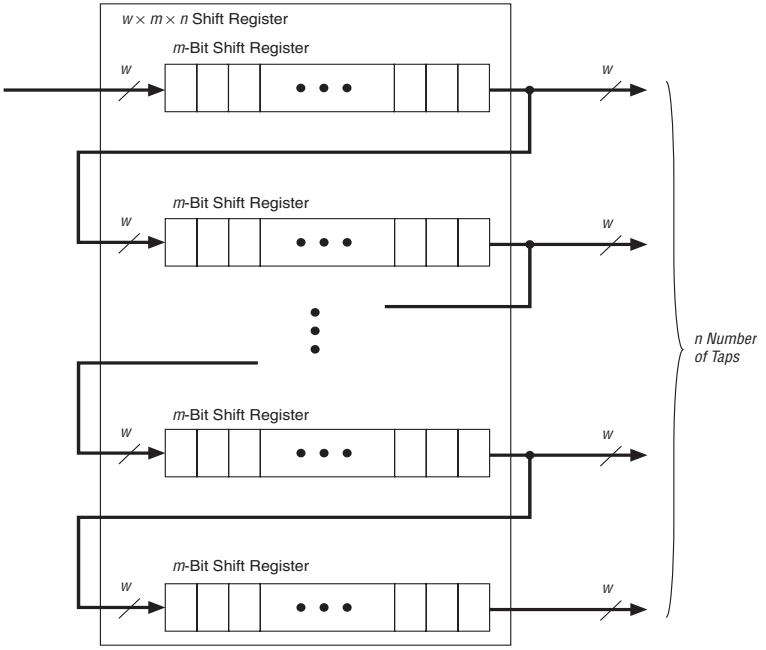
Implementing Shift-Register Mode

Embedded memory configurations can implement shift-register blocks 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 flip-flops that can quickly consume 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 and provides a more efficient implementation.

The size of a (w × m × n) shift register is determined by the input data width (w), the length of the taps (m), and the number of taps (n). The size of a (w × m × n) shift register must be less than or equal to the 4,608 bits. In addition, the size of (w × n) must be less than or equal to 36 bits. If a larger shift register is required, memory blocks can be cascaded together.

Data is written into each address location at the falling edge of the clock and read from the address at the rising edge of the clock. The shift-register mode logic automatically controls the positive and negative edge clocking to shift the data in one clock cycle. Figure 7–8 shows the M4K memory block in shift-register mode.

Figure 7–8. M4K Shift-Register Memory Configuration



Implementing ROM Mode

M4K blocks can also be configured as ROM. ROM can be initialized in an M4K block by using a memory initialization file (**.mif**). Because all M4K memory configurations must have synchronous inputs, the address lines of the ROM are registered. ROM outputs can be registered or combinatorial. The read operation of the ROM is identical to the read operation of the single-port RAM configuration.

Implementing FIFO Buffers

FIFO buffer outputs are always combinatorial. Simultaneous read and write operations from an empty FIFO buffer are not supported.

Clock Modes

Depending on the M4K memory mode, independent, input/output, read/write, and/or single-port clock modes are available. Table 7–6 shows the clock modes supported by the M4K memory modes.

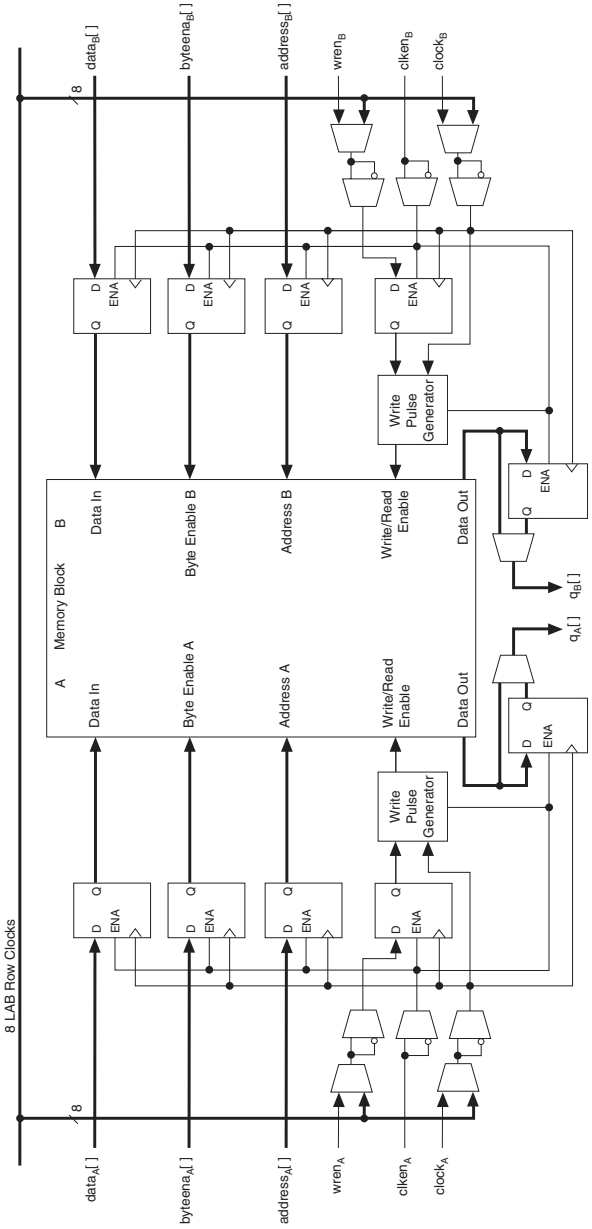
Table 7–6. M4K Memory Clock Modes

Clocking Mode	True-Dual Port Mode	Simple Dual-Port Mode	Single-Port Mode
Independent	✓	—	—
Input/output	✓	✓	—
Read/write	—	✓	—
Single-port	—	—	✓

Independent Clock Mode

M4K memory blocks can implement independent clock mode for true dual-port memory. In this mode, a separate clock is available for each port (A and 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-enable signals and asynchronous clear signals for port A and B registers. Figure 7–9 shows an M4K memory block in independent clock mode.

Figure 7–9. Independent Clock Mode *Note (1)*



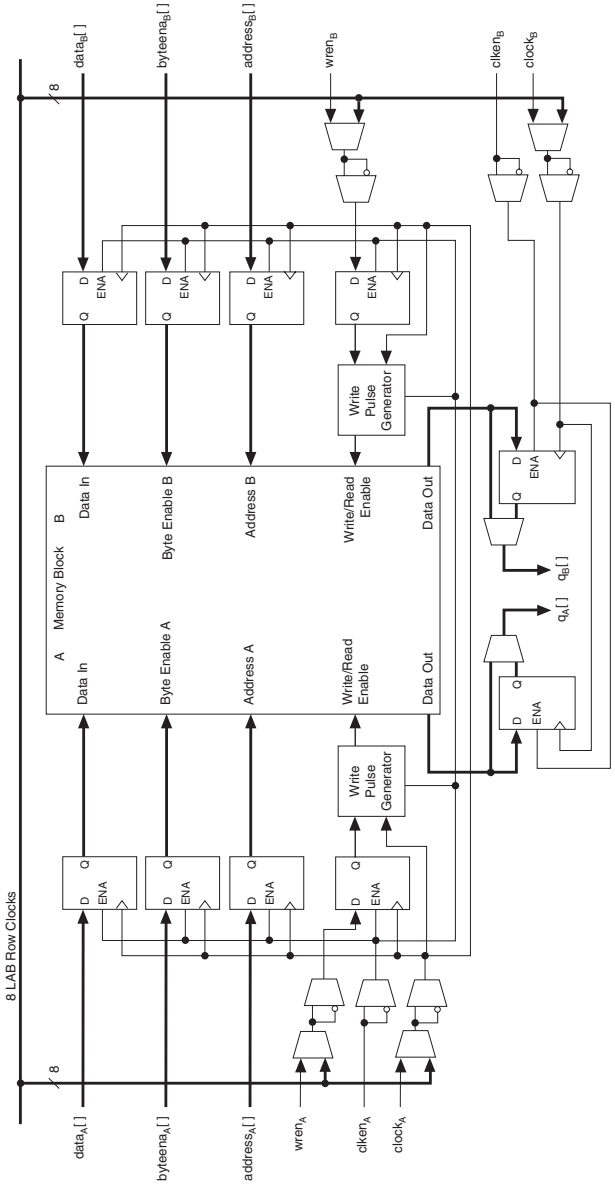
Note to Figure 7–9:

(1) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Input/Output Clock Mode

M4K memory blocks can implement input/output clock mode for true and simple dual-port memory. On each of the two ports, A and B, one clock controls all registers for inputs (data input, `wren`, and `address`) into the memory block. The other clock controls the block's data output registers. Each memory block port also supports independent clock enables and asynchronous clear signals for input and output registers. Figures 7–10 and 7–11 show the memory block in input/output clock mode for true and simple dual-port modes, respectively.

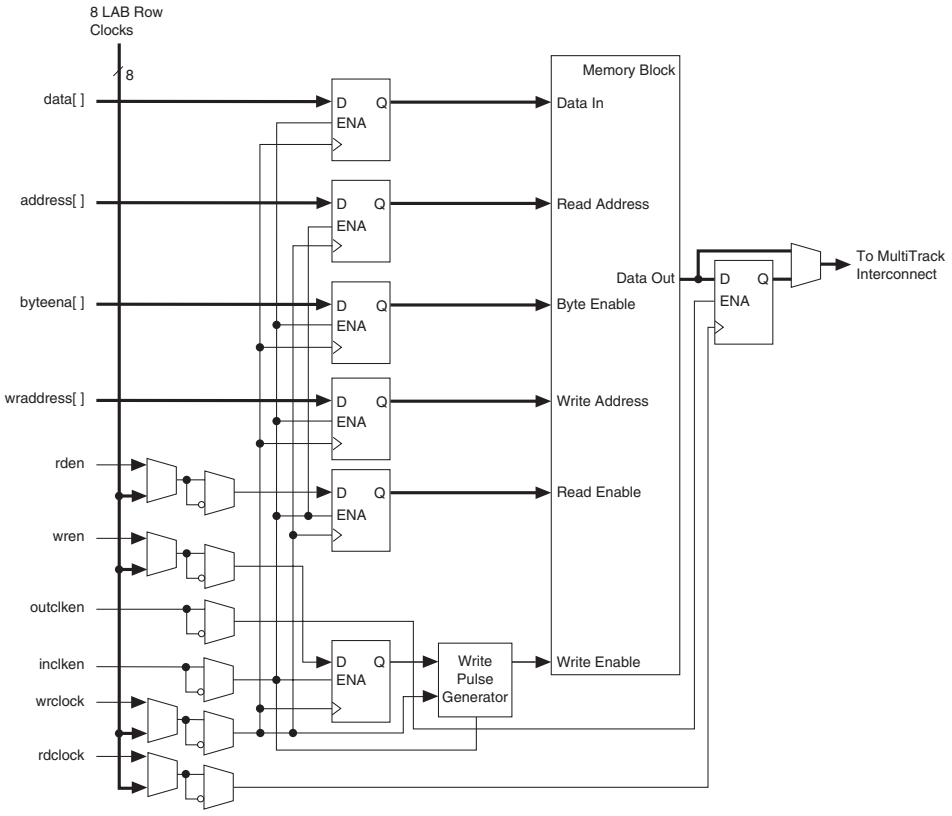
Figure 7–10. Input/Output Clock Mode in True Dual-Port Mode *Note (1)*



Note to Figure 7–10:

(1) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Figure 7–11. Input/Output Clock Mode in Simple Dual-Port Mode *Notes (1), (2)*



Notes to Figure 7–11:

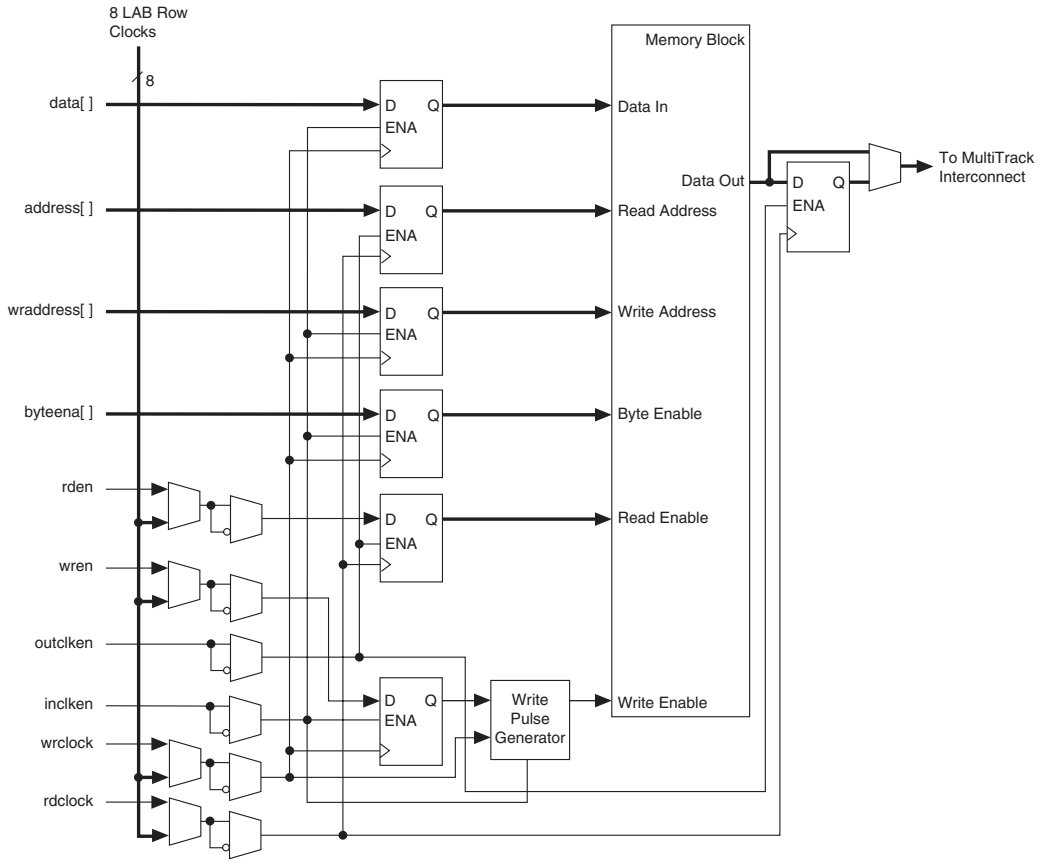
(1) For more information on the MultiTrack™ interconnect, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Read/Write Clock Mode

M4K memory blocks can implement read/write clock mode for simple dual-port memory. This mode can use up to two clocks. The write clock controls the block's data inputs, `wraddress`, and `wren`. The read clock controls the data output, `rdaddress`, and `rden`. The memory blocks support independent clock enables for each clock and asynchronous clear signals for the read- and write-side registers. Figure 7–12 shows a memory block in read/write clock mode.

Figure 7–12. Read/Write Clock Mode in Simple Dual-Port Mode *Notes (1), (2)*



Notes to Figure 7–12:

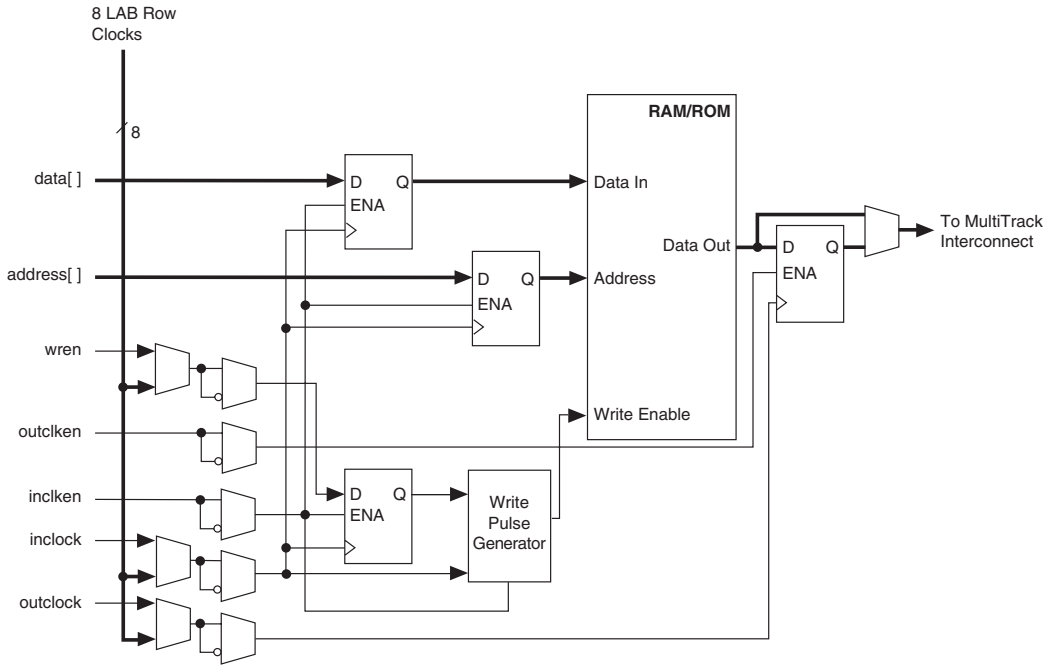
(1) For more information on the MultiTrack interconnect, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Single-Port Mode

The M4K memory blocks can implement single-port clock mode when simultaneous read and write operations are not required (see Figure 7–13). A single block in a memory block can support up to two single-port mode RAM blocks in M4K blocks.

Figure 7–13. Single-Port Mode *Notes (1), (2)*



Notes to Figure 7–13:

(1) For more information about the MultiTrack interconnect, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Synchronous and Pseudo-Asynchronous Modes

The M4K memory architecture implements synchronous, pipelined RAM by registering both the input and output signals to the RAM block. All M4K memory inputs are registered, providing synchronous write cycles. In synchronous operation, an M4K block generates its own self-timed strobe write enable (`wren`) signal derived from the global or regional clock. In contrast, a circuit using asynchronous RAM must generate the RAM `wren` signal while ensuring its data and address signals meet setup and hold time specifications relative to the `wren` signal. The output registers can be bypassed.

In an asynchronous memory, neither the input nor the output is registered. While Cyclone devices do not support asynchronous memory, they do support a pseudo-asynchronous read operation where the output data is available during the same clock cycle as when the read address is driven into it. Pseudo-asynchronous reading is possible in the simple and

true dual-port modes of the M4K blocks by clocking the read enable and read address registers on the negative clock edge and bypassing the output registers.

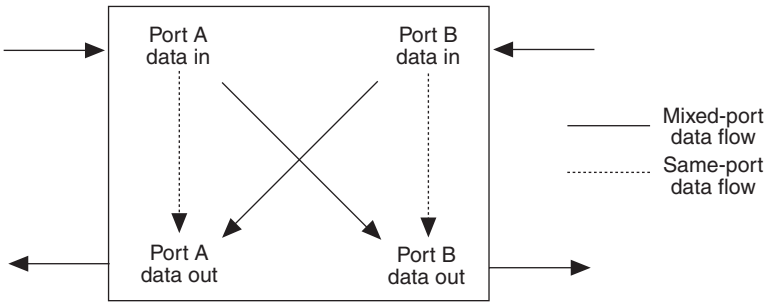
The clear signal for both asynchronous and synchronous mode for the memory are treated similarly in Cyclone devices. All inputs to the memory must be synchronous, therefore, the time it takes a clear signal to reset the input or output registers is synchronous to the clock.

For more information, refer to *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs*.

Read-during-Write Operation at the Same Address

The following two sections describe the functionality of the various M4K memory configurations when reading from an address during a write operation at that same address. There are two types of read-during-write operations: same-port and mixed-port. Figure 7–14 illustrates the difference in data flow between same-port and mixed-port read-during-write.

Figure 7–14. Read-during-Write Data Flow

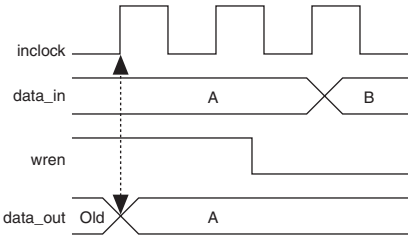


Same-Port Read-during-Write Mode

For read-during-write operation of a single-port RAM or the same port of a true dual-port RAM, the new data is available on the rising edge of the same clock cycle it was written on. See Figure 7–15 for a sample functional waveform.

When using byte-enable signals in true dual-port RAM mode, the outputs for the masked bytes on the same port are unknown. (See Figure 7–1.) The non-masked bytes are read out as shown in Figure 7–15.

Figure 7–15. Same-Port Read-during-Write Functionality *Note (1)*



Note to Figure 7–15:
(1) Outputs are not registered.

Mixed-Port Read-during-Write Mode

This mode is used when a RAM in simple or true dual-port mode has one port reading and the other port writing to the same address location with the same clock. You can configure the M4K memory block to operate in this mode and modify the parameter shown below using the MegaWizard® Plug-In Manager included with the Quartus II software.

The READ_DURING_WRITE_MODE_MIXED_PORTS parameter for M4K memory blocks determines whether or not to output the old data at the address. Setting this parameter to OLD_DATA outputs the old data at that address. Setting this parameter to DONT_CARE outputs an unknown value. During the instantiation of an ALTSYNCRAM or LPM_RAM_DP+ storage megafunction using the Quartus II software, the MegaWizard plug-in manager asks "How should the q output behave when reading a memory location that is being written from the other port?" Clicking "I don't care" assigns the DONT_CARE value to the parameter, and clicking "Old memory contents appear" assigns the OLD_DATA value to the parameter.



Altera recommends using the MegaWizard Plug-In Manager to create these memory megafunctions rather than directly creating instances. Once a storage megafunction is created using the MegaWizard Plug-In Manager, use the MegaWizard Plug-In Manager to make any necessary changes.

See Figures 7–16 and 7–17 for sample functional waveforms showing mixed-port read-during-write mode operation. These figures assume that the outputs are not registered.

Figure 7–16. Mixed-Port Read-during-Write: OLD_DATA

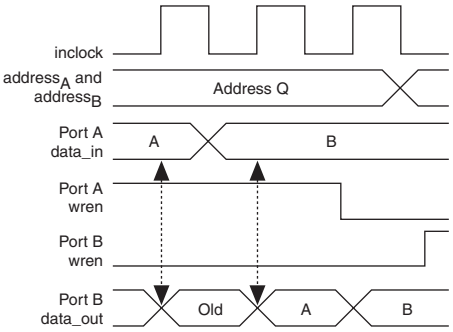
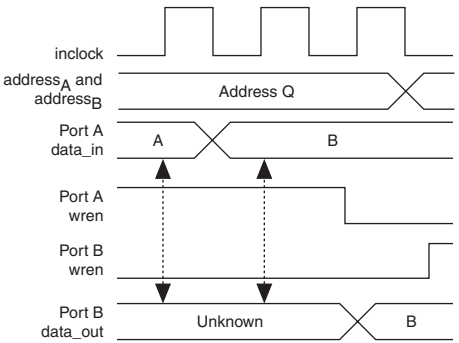


Figure 7–17. Mixed-Port Read-during-Write: DONT_CARE



Mixed-port read-during-write is not supported when two different clocks are used in a dual-port RAM. The output value is unknown during a mixed-port read-during-write operation.

For the minimum synchronous-write-cycle time, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

Conclusion

M4K memory blocks are a flexible memory solution available in Cyclone devices that provide advanced features such as byte-enable capability, parity bit storage capability, and shift-register mode, as well as mixed-port width support and true dual-port mode. This flexibility makes these embedded memory blocks well suited for a wide range of applications including ATM cell packet processing, header/cell storage, channelized functions, and program memory for processors.

Referenced Documents

This chapter references the following documents:

- *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs*
- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*

Document Revision History

Table 7–7 shows the revision history for this chapter.

Table 7–7. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.4	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.3	Added document revision history.	—
August 2005 v1.2	Minor updates.	—
February 2005 v1.1	Updated notes for Figures 7-9 through 7-13.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

Section IV. I/O Standards

This section provides information on the Cyclone FPGA I/O capabilities. It also includes information on selecting I/O standards for Cyclone devices in the Quartus II software.

This section contains the following chapters:

- Chapter 8. Using Selectable I/O Standards in Cyclone Devices
- Chapter 9. High-Speed Differential Signaling in Cyclone Devices

Revision History

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

8. Using Selectable I/O Standards in Cyclone Devices

C51008-1.6

Introduction

The proliferation of I/O standards and the need for improved I/O performance have made it critical that low-cost devices have flexible I/O capabilities. Selectable I/O capabilities such as SSTL-2, SSTL-3, and LVDS compatibility allow Cyclone® devices to connect to other devices on the same printed circuit board (PCB) that may require different operating and I/O voltages. With these aspects of implementation easily manipulated using the Altera Quartus® II software, the Cyclone device family enables system designers to use low-cost FPGAs while keeping pace with increasing design complexity.

This chapter is a guide to understanding the input/output capabilities of the Cyclone devices, including:

- Supported I/O Standards
- Cyclone I/O Banks
- Programmable Current Drive Strength
- Hot Socketing
- I/O Termination
- Pad Placement and DC Guidelines
- Quartus II Software Support

"Quartus II Software Support" on page 8–18 describes how to use the Quartus II software to specify device and pin options and assign pins to implement the above features of Cyclone devices.

Supported I/O Standards

Cyclone devices support the I/O standards shown in Table 8–1.



For more details about the I/O standards discussed in this section, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

Table 8–1. I/O Standards Supported by Cyclone Devices *Notes (1), (2)*

I/O Standard	Type	Input Voltage Level (V)	Output Voltage Level (V)	Input V_{REF} (V)	Output V_{CCIO} (V)	Termination V_{TT} (V)
3.3-V LVTTL/LVCMOS	Single-ended	3.3/2.5	3.3	N/A	3.3	N/A
2.5-V LVTTL/LVCMOS	Single-ended	3.3/2.5	2.5	N/A	2.5	N/A
1.8-V LVTTL/LVCMOS	Single-ended	3.3/2.5/1.8	1.8	N/A	1.8	N/A
1.5-V LVCMOS	Single-ended	3.3/2.5/1.8/1.5	1.5	N/A	1.5	N/A
PCI *(3)*	Single-ended	3.3	3.3	N/A	3.3	N/A
SSTL-3 Class I and II	Voltage-referenced	–0.3 to 3.9	3.3	1.5	3.3	1.5
SSTL-2 Class I and II	Voltage-referenced	–0.3 to 3.0	2.5	1.25	2.5	1.25
LVDS Compatibility	Differential	0 to 2.4	VOD = 0.25 to 0.55	N/A	2.5	N/A
RSDS Compatibility	Differential	0.1 to 1.4	VOD = 0.1 to 0.6	N/A	2.5	N/A
Differential SSTL - 2	Differential	N/A *(4)*	2.5	1.25	2.5	1.25

Notes to Table 8–1:

(1) The EP1C3 device in the 100-pin thin quad flat pack (TQFP) package does not have support for a PLL LVDS input or an external clock output.

(2) Cyclone devices have dual-purpose differential inputs. Outputs are balanced SSTL outputs requiring an external resistor divider.

(3) EP1C3 devices support PCI by using the LVTTL 16-mA I/O standard and drive strength assignments in the Quartus II software. The device requires an external diode for PCI compliance.

(4) This I/O standard is only available on output clock pins (`PLL_OUT` pins).

3.3-V LVTTL (EIA/JEDEC Standard JESD8-B)

The 3.3-V LVTTL I/O standard is a general-purpose, single-ended standard used for 3.3-V applications. The LVTTL standard defines the DC interface parameters for digital circuits operating from a 3.0-V/3.3-V power supply and driving or being driven by LVTTL-compatible devices.

The LVTTL input standard specifies a wider input voltage range of $-0.3\ V \leq V_I \leq 3.9\ V$. Altera recommends an input voltage range of $-0.5\ V \leq V_I \leq 4.1\ V$. The LVTTL standard does not require input reference voltages or board terminations. Cyclone devices support both input and output levels for 3.3-V LVTTL.

3.3-V LVCMOS (EIA/JEDEC Standard JESD8-B)

The 3.3-V LVCMOS I/O standard is a general-purpose, single-ended standard used for 3.3-V applications. The LVCMOS standard defines the DC interface parameters for digital circuits operating from a 3.0-V or 3.3-V power supply and driving or being driven by LVCMOS-compatible devices.

The LVCMOS standard specifies the same input voltage requirements as LVTTL ($-0.3\ V \leq V_I \leq 3.9\ V$). The output buffer drives to the rail to meet the minimum high-level output voltage requirements. The 3.3-V I/O Standard does not require input reference voltages or board terminations. Cyclone devices support both input and output levels specified by the 3.3-V LVCMOS I/O standard.

2.5-V LVTTL Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-5)

The 2.5-V I/O standard is used for 2.5-V LVTTL applications. This standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 2.5-V devices. The input and output voltage requirements are:

- The 2.5-V normal and wide range input standards specify an input voltage range of $-0.3\ V \leq VI \leq 3.0\text{-V}$.
- The normal range minimum high-level output voltage requirement (V_{OH}) is 2.1-V.
- The wide range minimum high-level output voltage requirement (V_{OH}) is $V_{CCIO} - 0.2\text{-V}$.

The 2.5-V standard does not require input reference voltages or board terminations. Cyclone devices support input and output levels for both 2.5-V LVTTL ranges.

2.5-V LVCMOS Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-5)

The 2.5-V I/O standard is used for 2.5-V LVCMOS applications. This standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 2.5-V parts. The input and output voltage ranges are:

- The 2.5-V normal and wide range input standards specify an input voltage range of – 0.3-V ≤ V_I ≤ 3.0-V.
- The normal range minimum V_{OH} requirement is 2.1 V.
- The wide range minimum V_{OH} requirement is V_{CCIO} – 0.2 V.

The 2.5-V standard does not require input reference voltages or board terminations. Cyclone devices support input and output levels for both 2.5-V LVCMOS ranges.

1.8-V LVTTL Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-7)

The 1.8-V I/O standard is used for 1.8-V LVTTL applications. This standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 1.8-V parts. The input and output voltage ranges are:

- The 1.8-V normal and wide range input standards specify an input voltage range of – 0.3 V ≤ V_I ≤ 2.25 V.
- The normal range minimum V_{OH} requirement is V_{CCIO} – 0.45 V.
- The wide range minimum V_{OH} requirement is V_{CCIO} – 0.2 V.

The 1.8-V standard does not require input reference voltages or board terminations. Cyclone devices support input and output levels for both normal and wide 1.8-V LVTTL ranges.

1.8-V LVCMOS Normal and Wide Voltage Ranges (EIA/JEDEC Standard EIA/JESD8-7)

The 1.8-V I/O standard is used for 1.8-V LVCMOS applications. This standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 1.8-V devices. The input and output voltage ranges are:

- The 1.8-V normal and wide range input standards specify an input voltage range of – 0.3 V ≤ V_I ≤ 2.25 V.
- The normal range minimum V_{OH} requirement is V_{CCIO} – 0.45 V.
- The wide range minimum V_{OH} requirement is V_{CCIO} – 0.2 V.

The 1.8-V standard does not require input reference voltages or board terminations. Cyclone devices support input and output levels for both normal and wide 1.8-V LVCMOS ranges.

1.5-V LVCMOS Normal and Wide Voltage Ranges (EIA/JEDEC Standard JESD8-11)

The 1.5-V I/O standard is used for 1.5-V applications. This standard defines the DC interface parameters for high-speed, low-voltage, non-terminated digital circuits driving or being driven by other 1.5-V devices. The input and output voltage ranges are:

- The 1.5-V normal and wide range input standards specify an input voltage range of $-0.3\ V \le V_I \le 1.9\text{-}V$.
- The normal range minimum V_{OH} requirement is 1.05 V.
- The wide range minimum V_{OH} requirement is $V_{CCIO} - 0.2\text{-}V$.

The 1.5-V standard does not require input reference voltages or board terminations. Cyclone devices support input and output levels for both normal and wide 1.5-V LVCMOS ranges.

3.3-V (PCI Special Interest Group (SIG) PCI Local Bus Specification Revision 2.2)

The PCI local bus specification is used for applications that interface to the PCI local bus, which provides a processor-independent data path between highly integrated peripheral controller components, peripheral add-in boards, and processor/memory systems. The conventional PCI specification revision 2.2 defines the PCI hardware environment including the protocol, electrical, mechanical, and configuration specifications for the PCI devices and expansion boards. This standard requires 3.3-V V_{CCIO}. The 3.3-V PCI standard does not require input reference voltages or board terminations.

The side I/O pins on all Cyclone devices (except the EP1C3 device) are fully compliant with the 3.3-V PCI Local Bus Specification Revision 2.2 and meet 32-bit/66-MHz operating frequency and timing requirements. The EP1C3 device supports the PCI I/O standard by using the LVTTL 16-mA setting and an external diode. The top and bottom I/O pins on all Cylcone devices support PCI by using the LVTTL 16-mA setting and an external diode.

Cyclone devices support PCI input and output levels on I/O banks 1 and 3 only. See "Cyclone I/O Banks" for more details and the IP MegaStore™ website.

Table 8–2 lists the specific Cyclone devices that support 64- and 32-bit PCI at 66 MHz.

Table 8–2. Cyclone 66-MHz PCI Support

Device	Package	-6 and -7 Speed Grades	
		64 Bit	32 Bit
EP1C4	324-pin FineLine BGA	✓	✓
	400-pin FineLine BGA	✓	✓
EP1C6	240-pin PQFP	—	✓
	256-pin FineLine BGA	—	✓
EP1C12	324-pin FineLine BGA	✓	✓
EP1C20	324-pin FineLine BGA	✓	✓
	400-pin FineLine BGA	✓	✓

Table 8–3 lists the specific Cyclone devices that support 64- and 32-bit PCI at 33 MHz.

Table 8–3. Cyclone 33-MHz PCI Support

Device	Package	-6, -7 and -8 Speed Grades	
		64 Bit	32 Bit
EP1C4	324-pin FineLine BGA	✓	✓
	400-pin FineLine BGA	✓	✓
EP1C6	240-pin PQFP	—	✓
	256-pin FineLine BGA	—	✓
EP1C12	240-pin PQFP	—	✓
	256-pin FineLine BGA	—	✓
	324-pin FineLine BGA	✓	✓
EP1C20	324-pin FineLine BGA	✓	✓
	400-pin FineLine BGA	✓	✓

SSTL-3 Class I and II (EIA/JEDEC Standard JESD8-8)

The SSTL-3 I/O standard is a 3.3-V memory bus standard used for applications such as high-speed SDRAM interfaces. This standard defines the input and output specifications for devices that operate in the SSTL-3 logic switching range of 0.0 to 3.3 V. The SSTL-3 standard specifies an input voltage range of $-0.3\ V \le V_I \le V_{CCIO} + 0.3\text{-}V$.
SSTL-3 requires a 1.5-V V_{REF} and a 1.5-V V_{TT} to which the series and termination resistors are connected (see Figures 8–1 and 8–2). In typical applications, both the termination voltage and reference voltage track the output supply voltage.

Figure 8–1. SSTL-3 Class I Termination

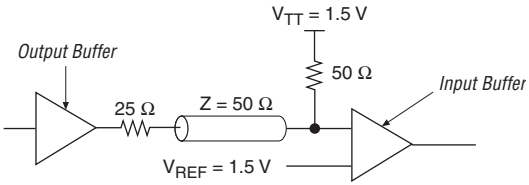
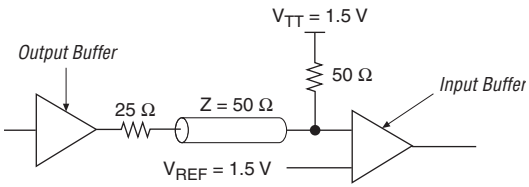


Figure 8–2. SSTL-3 Class II Termination



Cyclone devices support both input and output SSTL-3 Class I and II levels.

SSTL-2 Class I and II (EIA/JEDEC Standard JESD8-9A)

The SSTL-2 I/O standard is a 2.5-V memory bus standard used for applications such as high-speed double data rate (DDR) SDRAM interfaces. This standard defines the input and output specifications for devices that operate in the SSTL-2 logic switching range of 0.0-V to 2.5-V. This standard improves operation in conditions where a bus must be isolated from large stubs. The SSTL-2 standard specifies an input voltage range of $-0.3\ V \le V_I \le V_{CCIO} + 0.3\ V$. SSTL-2 requires a V_{REF} value of 1.25 V and a V_{TT} value of 1.25 V connected to the series and termination resistors (see Figures 8–3 and 8–4).

Figure 8–3. SSTL-2 Class I Termination

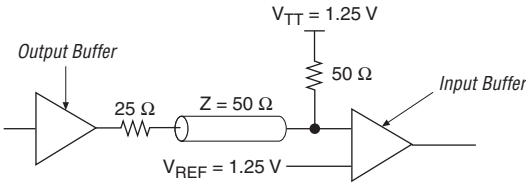
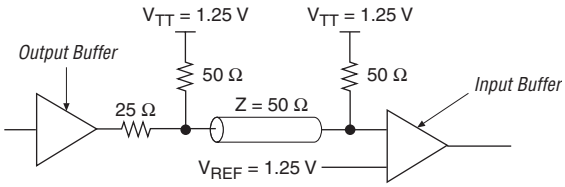


Figure 8–4. SSTL-2 Class II Termination



Cyclone devices support both input and output SSTL-2 Class I and II levels.

LVDS (ANSI/TIA/EIA Standard ANSI/TIA/EIA-644)

The LVDS I/O standard is a differential high-speed, low-voltage swing, low-power, general-purpose I/O interface standard. This standard is used in applications requiring high-bandwidth data transfer, backplane drivers, and clock distribution. The ANSI/TIA/EIA-644 standard specifies LVDS transmitters and receivers capable of operating at recommended maximum data signaling rates of 655 Mbps. Devices can operate at slower speeds if needed however, and there is a theoretical maximum of 1.923 Gbps. Due to the low-voltage swing of the LVDS I/O standard, the electromagnetic interference (EMI) effects are much smaller than CMOS, TTL, and PECL. This low EMI makes LVDS ideal for applications with low EMI requirements or noise immunity requirements. The LVDS standard specifies a differential output voltage range of 250 mV ≤ V_{OD} ≤ 550 mV.

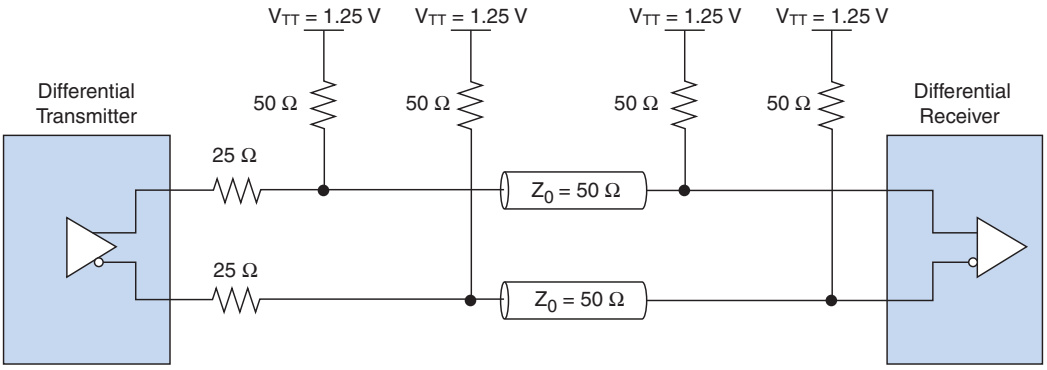
The Cyclone device family meets the ANSI/TIA/EIA-644 standard and is LVDS-compatible but, unlike previous products with LVDS support, Cyclone does not have dedicated SERDES or LVDS drivers. While external resistors are required for LVDS output support, Cyclone does have direct LVDS-compatible input support throughout the device. This

flexible approach to LVDS support allows LVDS compatibility on every bank of the Cyclone device at speeds up to 640 Mbps. (Contact Altera Applications for the latest LVDS specification).

Differential SSTL-2 - EIA/JEDEC Standard JESD8-9A

The differential SSTL-2 I/O standard is a 2.5-V standard used for applications such as high-speed DDR SDRAM clock interfaces. This standard supports differential signals in systems using the SSTL-2 standard and supplements the SSTL-2 standard for differential clocks. The differential SSTL-2 standard specifies an input voltage range of $-0.3\ V \le V_I \le V_{CCIO} + 0.3\text{-}V$. The differential SSTL-2 standard does not require an input reference voltage differential. See Figure 8–5 for details on differential SSTL-2 termination. Cyclone devices support output clock levels for differential SSTL-2 class II operation.

Figure 8–5. SSTL-2 Class II Differential Termination



For more details about the I/O standards discussed in this section, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

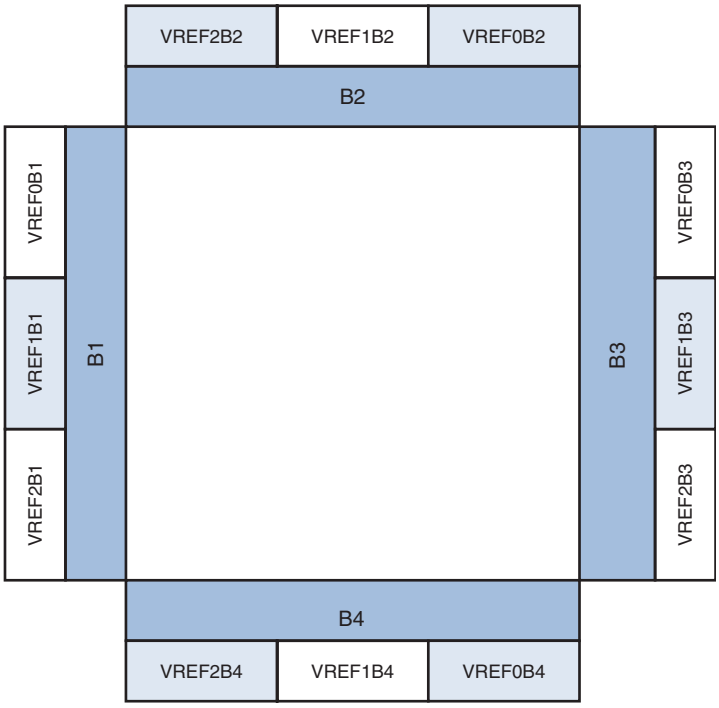
Cyclone I/O Banks

The I/O pins on Cyclone devices are grouped together into I/O banks and each bank has a separate power bus. This permits designers to select the preferred I/O standard for a given bank enabling tremendous flexibility in the Cyclone device's I/O support.

Each Cyclone device supports four I/O banks regardless of density. Similarly, each device I/O pin is associated with one of these specific, numbered I/O banks. To accommodate voltage-referenced I/O

standards, each Cyclone I/O bank supports three V_{REF} pins (see Figure 8–6). In the event these pins are not used as V_{REF} pins, they may be used as regular I/O pins.

Figure 8–6. Cyclone Power Bank and V_{REF} Arrangement



Additionally, each Cyclone I/O bank has its own VCCIO pins. Any single I/O bank must have only one V_{CCIO} setting from among 1.5-V, 1.8-V, 2.5-V or 3.3-V. Although there can only be one V_{CCIO} voltage, Cyclone devices permit additional input signaling capabilities as shown in Table 8–4.

Table 8–4. Acceptable Input Levels for LVTTL/LVCMOS *Note (1)* *(Part 1 of 2)*

Bank V_{CCIO}	Acceptable Input Levels			
	3.3-V	2.5-V	1.8-V	1.5-V
3.3-V	✓	✓	—	—
2.5-V	✓	✓	—	—
1.8-V	✓ *(2)*	✓ *(2)*	✓	✓

Table 8–4. Acceptable Input Levels for LVTTL/LVCMOS Note (1) (Part 2 of 2)				
Bank V_{CCIO}	Acceptable Input Levels			
	3.3-V	2.5-V	1.8-V	1.5-V
1.5-V	✓(2)	✓(2)	✓	✓

Notes to Table 8–4:

(1) For SSTL and LVDS I/O Standard, input buffers are powered by V_{CCINT} and not V_{CCIO}. Hence, input buffers can accept input levels of 3.3 V or 2.5 V regardless of V_{CCIO} level for both SSTL and LVDS I/O Standard.

(2) These input values overdrive the input buffer, so the pin leakage current is slightly higher than the default value. Check A**llow voltage overdrive for LVTTL/LVCMOS input pins** in Settings > Device > Device and Pin Options > Pin Placement tab to allow input pins with LVTTL or LVCMOS I/O standards to be placed by the Quartus II software inside an I/O bank with a lower V_{CCIO} voltage than the voltage specified by the pins.



For more information about acceptable input levels, refer to *Using Cyclone Devices in Multiple-Voltage Systems* chapter in the *Cyclone Device Handbook*.

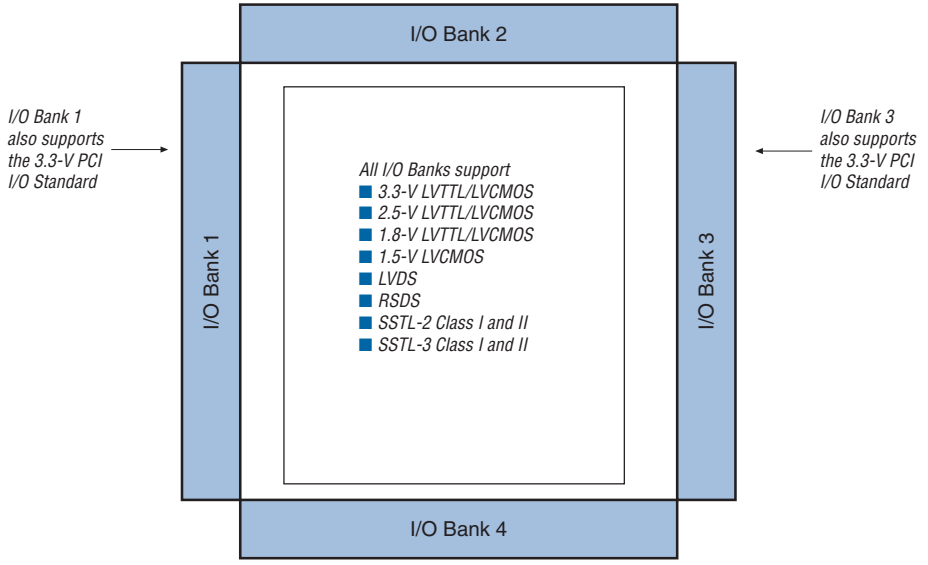
Any number of supported single-ended or differential standards can be simultaneously supported in a single I/O bank as long as they use compatible V_{CCIO} levels for input and output pins. For example, an I/O bank with a 2.5-V V_{CCIO} setting can support 2.5-V LVTTL inputs and outputs, 2.5-V LVDS-compatible inputs and outputs, and 3.3-V LVCMOS inputs only.

Voltage-referenced standards can be 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 a compatible V_{CCIO} value. For example, if you choose to implement both SSTL-3 and SSTL-2 in your Cyclone device, I/O pins using these standards—because they require different V_{REF} values—must be in different banks from each other. However, SSTL-3 and 3.3-V LVCMOS could be supported in the same bank with the V_{CCIO} set to 3.3-V and the V_{REF} set to 1.5-V.

See "Pad Placement and DC Guidelines" on page 8–14 for more information.

All four I/O banks support all of the I/O standards with the exception of PCI, which is only supported on banks 1 and 3 (see Figure 8–7).

Figure 8–7. I/O Standards Supported in Cyclone Devices *Notes (1), (2)*



Notes to Figure 8–7

(1) EP1C3 devices support PCI by using the LVTTL 16-mA I/O standard and drive strength assignments in the Quartus II software. The device requires an external diode for PCI compliance.

(2) The EP1C3 device in the 100-pin thin quad flat pack (TQFP) package does not have support for a PLL LVDS-compatible input or an external clock output.

Programmable Current Drive Strength

The Cyclone device I/O standards support various output current drive settings as shown in Table 8–5. These programmable drive-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 I_{OH} and I_{OL} of the corresponding I/O standard.

These drive-strength settings are programmable on a per-pin basis (for output and bidirectional pins only) using the Quartus II software. To modify the current strength of a particular pin, refer to "Programmable Drive Strength Settings".

Table 8–5. Programmable Drive Strength

I/O Standard *(1)*	I_{OH}/I_{OL} Current Strength Setting *(2)*
3.3-V LVTTL	24, 16, 12, 8, 4 mA
3.3-V LVCMOS	12, 8, 4, 2 mA
2.5-V LVTTL/LVCMOS	16, 12, 8, 2 mA
1.8-V LVTTL/LVCMOS	12, 8, 2 mA
1.5-V LVCMOS	8, 4, 2 mA

Notes to Table 8–5:

(1) The Quartus II software default current setting is the maximum setting for each I/O standard.

(2) SSTL 2 class I and II, SSTL 3 class I and II, and PCI do not support programmable drive strength.

Hot Socketing

Cyclone devices support any power-up or power-down sequence (V_{CCIO} and V_{CCINT}) to facilitate hot socketing. You can drive signals into the device before or during power-up or power-down without damaging the device. Cyclone devices will not drive out until the device is configured and has attained proper operating conditions.

You can power up or power down the VCCIO and VCCINT pins in any sequence. The power supply ramp rates can range from 100 ns to 100 ms. All V_{CC} supplies must power down within 100 ms of each other to prevent I/O pins from driving out. Additionally, during power-up, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20pF.

- The hot socketing DC specification is $| I_{IOPIN} | < 300$ µA.
- The hot socketing AC specification is $| I_{IOPIN} | < 8$ mA for 10 ns or less.

I/O Termination

The majority of the Cyclone I/O standards are single-ended, non-voltage-referenced I/O standards and, as such, the following I/O standards do not specify a recommended termination scheme:

- 3.3-V LVTTL / LVCMOS
- 2.5-V LVTTL / LVCMOS
- 1.8-V LVTTL / LVCMOS
- 1.5-V LVCMOS
- 3.3-V PCI

The Cyclone device family does not feature on-chip I/O termination resistors.

Voltage-Referenced I/O Standard Termination

Voltage-referenced I/O standards require both an input reference voltage, V_{REF}, and a termination voltage, V_{TT}. An external pull up to V_{TT} must be provided to the Cyclone device as the device does not have V_{TT} pins. The reference voltage of the receiving device tracks the termination voltage of the transmitting device.

For more information on termination for voltage-referenced I/O standards, refer to "Supported I/O Standards".

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.

LVDS and RSDS are the only differential I/O standards supported by Cyclone devices. For information on LVDS termination and RSDS termination, refer to the *LVDS Receiver and Transmitter Termination* and *RSDS I/O Standard Support in Cyclone Devices* sections, respectively, in the *High-Speed Differential Signaling in Cyclone Devices* chapter in the *Cyclone Device Handbook*.

Pad Placement and DC Guidelines

This section provides pad placement guidelines for the programmable I/O standards supported by Cyclone devices and includes essential information for designing systems using the devices' selectable I/O capabilities. This section also discusses the DC limitations and guidelines.

Differential Pad Placement Guidelines

In order to maintain an acceptable noise level on the V_{CCIO} supply, there are restrictions on placement of single-ended I/O pads in relation to differential pads. Use the following guidelines for placing single-ended pads with respect to differential pads in Cyclone devices.

- Single-ended inputs may be only be placed four or more pads away from a differential pad.
- Single-ended outputs and bidirectional pads may only be placed five or more pads away from a differential pad.



The Quartus II software generates an error message for illegally placed pads.

V_{REF} Pad Placement Guidelines

In order to maintain an acceptable noise level on the V_{CCIO} supply and to prevent output switching noise from shifting the V_{REF} rail, there are restrictions on the placement of single-ended voltage referenced I/Os with respect to V_{REF} pads and VCCIO/GND pairs. Please use the following guidelines for placing single-ended pads in Cyclone devices.

Input Pads

Each V_{REF} pad supports a maximum of 40 input pads with up to 20 on each side of the V_{REF} pad. This is irrespective of VCCIO/GND pairs.

Output Pads

When a voltage referenced input or bidirectional pad does not exist in a bank, there is no limit to the number of output pads that can be implemented in that bank. When a voltage referenced input exists, each VCCIO/GND pair supports 9 outputs for Fineline BGA® packages or 4 outputs for quad flat pack (QFP) packages. Any output pads must be placed greater than 1 pad away from your V_{REF} pad to maintain acceptable noise levels.

Bidirectional Pads

Bidirectional pads must satisfy input and output guidelines simultaneously. If the bidirectional pads are all controlled by the same OE and there are no other outputs or voltage referenced inputs in the bank, then there is no case where there is a voltage referenced input active at the same time as an output. Therefore, the output limitation does not apply. However, since the bidirectional pads are linked to the same OE, the bidirectional pads will all act as inputs at the same time. Therefore, the input limitation of 40 input pads (20 on each side of your V_{REF} pad) will apply.

If the bidirectional pads are all controlled by different output enables (OE) and there are no other outputs or voltage referenced inputs in the bank, then there may be a case where one group of bidirectional pads is acting as inputs while another group is acting as outputs. In such cases, apply the formulas shown in Table 8–6.

Table 8–6. Input-Only Bidirectional Pad Limitation Formulas

Package Type	Formula
FineLine BGA	(Total number of bidirectional pads) - (Total number of pads from the smallest group of pads controlled by an OE) ≤ 9 (per VCCIO/GND pair)
QFP	(Total number of bidirectional pads) - (Total number of pads from the smallest group of pads controlled by an OE) ≤ 4 (per VCCIO/GND pair).

Consider an FineLine BGA package with 4 bidirectional pads controlled by OE1, 4 bidirectional pads controlled by OE2, and 2 bidirectional pads controlled by OE3. If OE1 and OE2 are active and OE3 is inactive, there are 10 bidirectional pads, but it is safely allowable because there would be 8 or fewer outputs per VCCIO/GND pair.

When at least one additional voltage referenced input and no other outputs exist in the same V_{REF} bank, the bidirectional pad limitation applies in addition to the input and output limitations. See the following equation.

(Total number of bidirectional pads) + *(Total number of input pads)* ≤ 40
(20 on each side of your V_{REF} pad)



The bidirectional pad limitation applies to both Fineline BGA packages and QFP packages.

After applying the equation above, apply one of the equations in Table 8–7, depending on package type.

Table 8–7. Bidirectional Pad Limitation Formulas (Where V_{REF} Inputs Exist)

Package Type	Formula
FineLine BGA	(Total number of bidirectional pads) ≤ 9 (per VCCIO/GND pair)
QFP	(Total number of bidirectional pads) ≤ 4 (per VCCIO/GND pair)

When at least one additional output exists but no voltage referenced inputs exist, apply the appropriate formula from Table 8–8.

Table 8–8. Bidirectional Pad Limitation Formulas (Where V_{REF} Outputs Exist)

Package Type	Formula
FineLine BGA	(Total number of bidirectional pads) + (Total number of additional output pads) - (Total number of pads from the smallest group of pads controlled by an OE) ≤ 9 (per VCCIO/GND pair)
QFP	(Total number of bidirectional pads) + (Total number of additional output pads) - (Total number of pads from the smallest group of pads controlled by an OE) = 4 (per VCCIO/GND pair)

When additional voltage referenced inputs and other outputs exist in the same V_{REF} bank, then the bidirectional pad limitation must again simultaneously adhere to the input and output limitations. As such, the following rules apply:

Total number of bidirectional pads + *Total number of input pads* ≤40 (20 on each side of your V_{REF} pad).



The bidirectional pad limitation applies to both Fineline BGA packages and QFP packages.

After applying the equation above apply one of the equations in Table 8–9, depending on package type.

Table 8–9. Bidirectional Pad Limitation Formulas (Multiple V_{REF} Inputs and Outputs)

Package Type	Formula
FineLine BGA	(Total number of bidirectional pads) + (Total number of output pads) ≤ 9 (per VCCIO/GND pair)
QFP	(Total number of bidirectional pads) + (Total number of output pads) ≤ 4 (per VCCIO/GND pair)

Each I/O bank can only be set to a single V_{CCIO} voltage level and a single V_{REF} voltage level at a given time. Pins of different I/O standards can share the bank if they have compatible V_{CCIO} values (see Table 8–4 for more details).

In all cases listed above, the Quartus II software generates an error message for illegally placed pads.

DC Guidelines

There is a current limit of 320 mA per 16 consecutive output pins, as shown by the following equation:

$$\sum_{pin}^{pin+15} I_{pin} < 320\ mA$$

Table 8–10 shows the current allowed per pin by select I/O standards as measured under the standard's defined loading conditions. PCI, LVTTL, LVCMOS, and other supported I/O standards not shown in the table do not have standardized loading conditions. As such, the current allowed per pin in a series-loaded condition for these standards is considered negligible.

Table 8–10. I/O Standard DC Specification

Pin I/O Standard	I Pin (mA)	
	3.3-V V_{CCIO}	2.5-V V_{CCIO}
SSTL-3 Class I	8	N/A
SSTL-3 Class II	16	N/A
SSTL-2 Class I	N/A	8.1
SSTL-2 Class II	N/A	16.4
LVDS	N/A	

Quartus II Software Support

Use the Quartus II software to specify which programmable I/O standards to use for Cyclone devices. This section describes Quartus II implementation, placement, and assignment guidelines, including:

- Settings
- Device and pin options
- Assigning pins
- Programmable drive strength settings
- I/O banks in the floorplan view
- Auto placement and verification

Settings

The Settings dialog box (Assignments menu) includes options allowing you to set a default I/O standard, optimize for I/O placement, assign I/O pins, and numerous other I/O-related options. The most pertinent user features are described in detail below.

Device and Pin Options

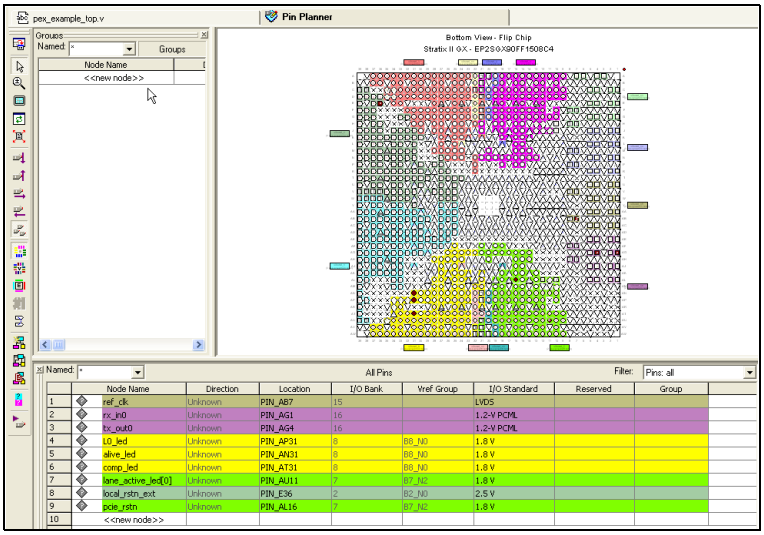
To access Device and Pin Options, choose **Settings** from Assignments menu. From **Settings** dialog box, click **Device and Pin Options**. There are numerous categories in the Device and Pin Options dialog box, including General, Configuration, Programming Files, Unused Pins, Dual-Purpose Pins, and Voltage. Similarly, each of these categories contains settings vital to the device operation such as the default I/O standard applied to the device (Voltage tab), how to reserve all unused pins (Unused Pins tab), specify the capacitive load (in picofarads (pF)) on output pins for each I/O standards (Capacitive Loading tab), and whether or not the device should enable a device-wide reset (General tab).

Assigning Pins

Assuming a specific device has been chosen in the available devices list in the Device Settings dialog box (Assignments menu), clicking **Pin Planner** provides the device's pin settings and pin assignments (see Figure 8–8). You can view, add, remove and update pin settings in the Pin Planner window. The information for each pin includes:

- Node Name
- Direction
- Location
- I/O Bank
- V_{ref} Group
- I/O Standard
- Reserved
- Group

Figure 8–8. Assign Pins



You can use **Filter** in the Pin Planner window to list assigned, unassigned, input, output, bidirectional or all pins.

When you assign an I/O standard that requires a reference voltage to an I/O pin, the Quartus II software automatically assigns VREF pins. Refer to Quartus II Help for instructions on how to use an I/O standard for a pin.

Programmable Drive Strength Settings

To specify programmable drive strength settings, perform the following steps:

1. Choose **Assignment Editor** (Assignments menu).
2. Under **To** field in the **Assignment Editor** box, right-click on a new row. Select **Node Finder**. Click **List** in the **Node Finder** window. Then select the output or bidirectional pin for which you will specify the current strength.
3. Set the **Assignment Name** field to **Current Strength** (accepts wildcards/groups), then enter the desired value in the **Value** field.
4. Select **Yes** under **Enabled** field to enable the selected current strength.

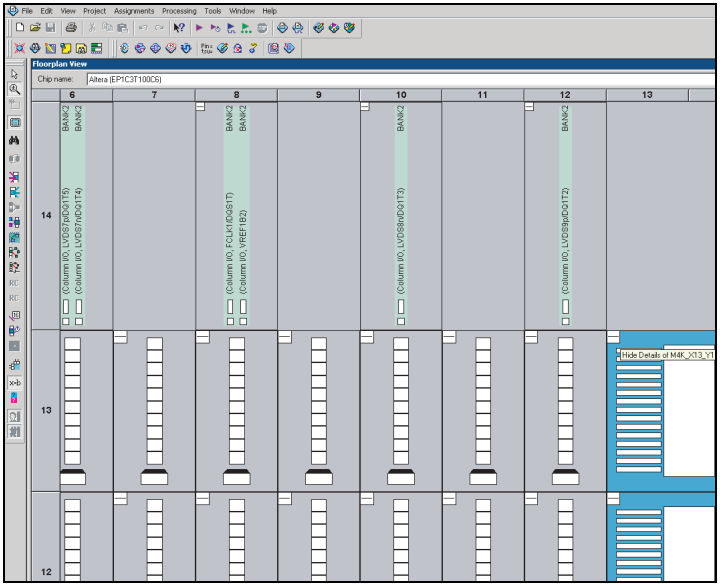
The Quartus II software displays the entire range of drive strength choices. While the Quartus II software does not prohibit you from specifying any of these for your I/O pin, not every setting is supported by every I/O standard. See Table 8–5 for supported combinations.

I/O Banks in the Floorplan View

View the arrangement of the device I/O banks by choosing **Timing Closure Floorplan** (Assignments View menu) with the Floorplan View displayed (see Figure 8–9). Pins that belong to the same I/O bank must use the same V_{CCIO} voltage. You can assign multiple I/O standards to the I/O pins in any given I/O bank as long as the V_{CCIO} voltage of the desired I/O standards is the same.

A given bank can have up to three V_{REF} signals, and each signal can support one voltage-referenced I/O standard. Each device I/O pin belongs to a specific, numbered I/O bank. By default, the **Show I/O Banks** option is enabled, allowing the I/O banks to be displayed as color coded (See Figure 8–9).

Figure 8–9. Floorplan View Window



Auto Placement and Verification of Selectable I/O Standards

The Quartus II software automatically verifies the placement for all I/O and VREF pins and performs the following actions:

- Automatically places I/O pins of different V_{REF} standards without pin assignments in separate I/O banks and enables the VREF pins of these I/O banks.
- Verifies that voltage-referenced I/O pins requiring different V_{REF} levels are not placed in the same bank.
- Reports an error message if the current limit is exceeded for a Cyclone power bank (See "DC Guidelines").
- Automatically assigns VREF pins and I/O pins such that the current requirements are met and I/O standards are placed properly.

Conclusion

Cyclone device I/O capabilities enable system designers to keep pace with increasing design complexity utilizing a low-cost FPGA device family. Support for I/O standards including SSTL and LVDS compatibility allow Cyclone devices to fit into a wide variety of applications. The Quartus II software makes it easy to use these I/O standards in Cyclone device designs. After design compilation, the software also provides clear, visual representations of pads and pins and the selected I/O standards. Taking advantage of the support of these I/O standards in Cyclone devices will allow you to lower your design costs without compromising design flexibility or complexity.

More Information

For more information about Cyclone devices refer to the following resources:

- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*
- *Using Cyclone Devices in Multiple-Voltage Systems* chapter in the *Cyclone Device Handbook*
- *AN 75: High-Speed Board Designs*

References

For more information on the I/O standards referred to in this document, see the following sources:

- Stub Series Terminated Logic for 2.5-V (SSTL-2), JESD8-9A, Electronic Industries Association, December 2000.
- 1.5-V +/- 0.1-V (Normal Range) and 0.9-V - 1.6-V (Wide Range) Power Supply Voltage and Interface Standard for Non-terminated Digital Integrated Circuits, JESD8-11, Electronic Industries Association, October 2000.

- 1.8-V +/- 0.15-V (Normal Range) and 1.2-V - 1.95-V (Wide Range) Power Supply Voltage and Interface Standard for Non-terminated Digital Integrated Circuits, JESD8-7, Electronic Industries Association, February 1997.
- 2.5-V +/- 0.2-V (Normal Range) and 1.8-V to 2.7-V (Wide Range) Power Supply Voltage and Interface Standard for Non-terminated Digital Integrated Circuits, JESD8-5, Electronic Industries Association, October 1995.
- Interface Standard for Nominal 3-V/ 3.3-V Supply Digital Integrated Circuits, JESD8-B, Electronic Industries Association, September 1999.
- PCI Local Bus Specification, Revision 2.2, PCI Special Interest Group, December 1998.
- Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits, ANSI/TIA/EIA-644, American National Standards Institute/Telecommunications Industry/Electronic Industries Association, October 1995.

Referenced Documents

This chapter references the following documents:

- *AN 75: High-Speed Board Designs*
- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*
- *High-Speed Differential Signaling in Cyclone Devices* chapter in the *Cyclone Device Handbook*
- *Using Cyclone Devices in Multiple-Voltage Systems* chapter in the *Cyclone Device Handbook*

Document Revision History

Table 8–11 shows the revision history for this chapter.

Table 8–11. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.6	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.5	• Added document revision history. • Removed references to "compiler" settings and updated information in "Quartus II Software Support" section. • Updated Figure 8–8 and the following handpara note. • Updated procedure in "Programmable Drive Strength Settings" section. • Minor update in "I/O Banks in the Floorplan View".	—

August 2005 v1.4	Minor updates.	—
February 2005 v1.3	• Updated information concerning hot socketing AC specifications. • Updated the notes to Figures 8-13 through 8-20. • Updated text in the Output Pads section. Changed 2 pads away to 1.	—
October 2003 v1.2	Updated the 3.3-V (PCI Special Interest Group (SIG) PCI Local Bus Specification Revision 2.2) section.	—
September 2003 v1.1	Updated LVDS data rates to 640 Mbps from 311 Mbps.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

9. High-Speed Differential Signaling in Cyclone Devices

C51009-1.6

Introduction

From high-speed backplane applications to high-end switch boxes, low-voltage differential signaling (LVDS) is the technology of choice. LVDS is a low-voltage differential signaling standard, allowing higher noise immunity than single-ended I/O technologies. Its low-voltage swing allows for high-speed data transfers, low power consumption, and less electromagnetic interference (EMI). LVDS I/O signaling is a data interface standard defined in the TIA/EIA-644 and IEEE Std. 1596.3 specifications.

The reduced swing differential signaling (RSDS) standard is a derivative of the LVDS standard. The RSDS I/O standard is similar in electrical characteristics to LVDS, but has a smaller voltage swing and therefore provides further power benefits and reduced EMI. National Semiconductor Corporation introduced the RSDS specification and now many vendors use it for flat panel display (FPD) links between the controller and the drivers that drive the display column drivers. Cyclone® devices support the RSDS I/O standard at speeds up to 311 megabits per second (Mbps).

Altera® Cyclone devices allow you to transmit and receive data through LVDS signals at a data rate up to 640 Mbps. For the LVDS transmitter and receiver, the Cyclone device's input and output pins support serialization and deserialization through internal logic.

This chapter describes how to use Cyclone I/O pins for LVDS and RSDS signaling and contains the following topics:

- Cyclone I/O Banks
- Cyclone High-Speed I/O Interface
- LVDS Receiver and Transmitter
- RSDS I/O Standard Support in Cyclone Devices
- Cyclone Receiver and Transmitter Termination
- Implementing Cyclone LVDS and RSDS I/O Pins in the Quartus® II Software
- Design Guidelines

Cyclone High-Speed I/O Banks

Cyclone devices offer four I/O banks, as shown in Figure 9–1. A subset of pins in each of the four I/O banks (on both rows and columns) support the high-speed I/O interface. Cyclone pin tables list the pins that support the high-speed I/O interface. The EP1C3 device in the 100-pin thin quad flat pack (TQFP) package does not support the high-speed I/O interface.

Figure 9–1. Cyclone I/O Banks

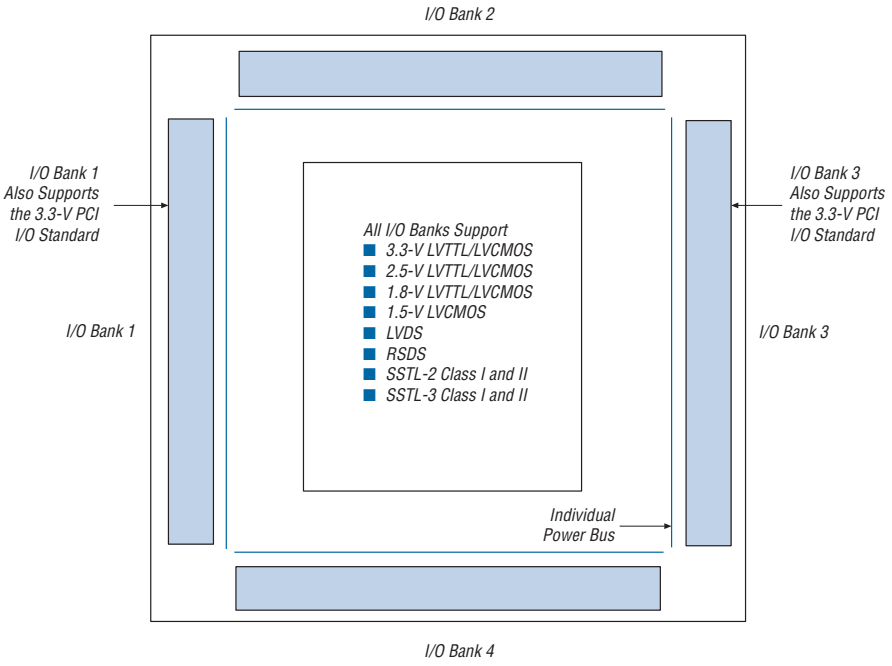


Table 9–1 shows the total number of supported high-speed I/O channels in each Cyclone device. You can use each channel as a receiver or transmitter.

Cyclone devices support different modes (× 1, × 2, × 4, × 7, × 8, and × 10) of operation with a maximum internal clock frequency of 405 MHz (-6 speed grade), 320 MHz (-7 speed grade), or 275 MHz (-8 speed grade), and a maximum data rate of 640 Mbps (-6 speed grade).

Table 9–1. Number of High-Speed I/O Channels Per Cyclone Device

Device	Pin Count	Total Number of High-Speed I/O Channels
EP1C3	144	34
EP1C4	324	103
	400	129
EP1C6	144	29
	240	72
	256	72
EP1C12	240	66
	256	72
	324	103
EP1C20	324	95
	400	129



For more information about I/O standards supported by Cyclone devices, refer to the *Using Selectable I/O Standards in Cyclone Devices* chapter in the *Cyclone Device Handbook*.

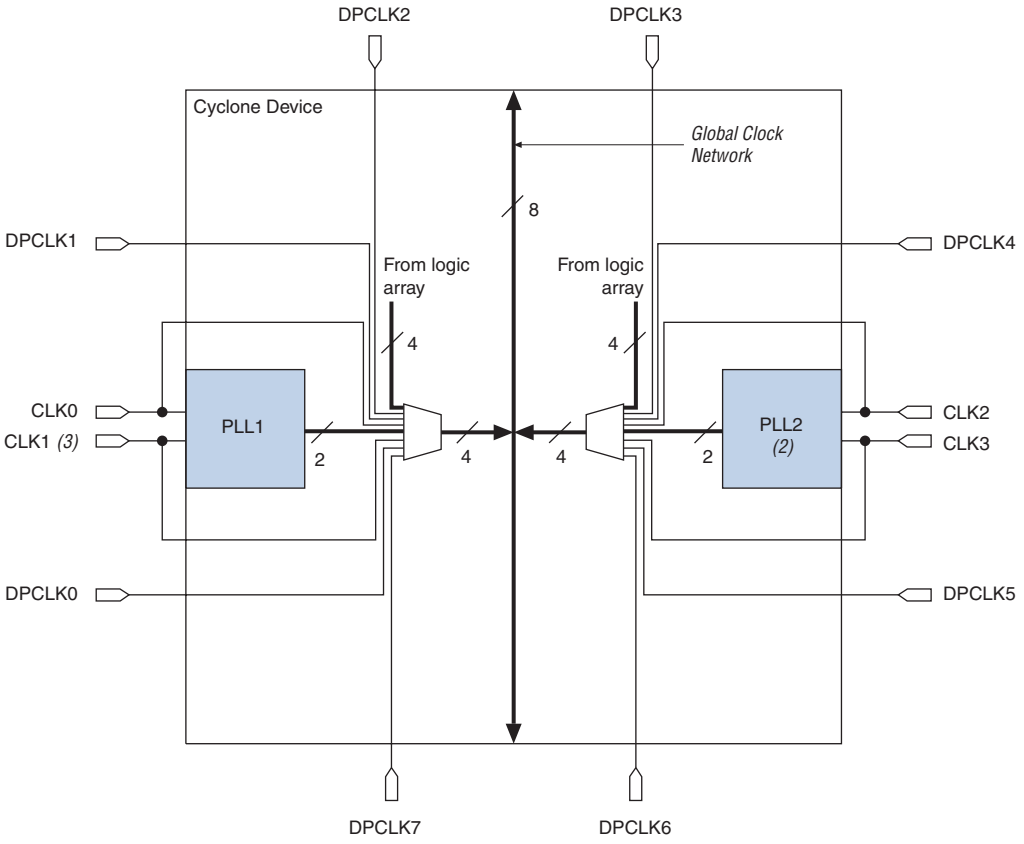
Cyclone High-Speed I/O Interface

You can use the I/O pins and internal logic to implement an high-speed I/O receiver and transmitter in Cyclone devices. Cyclone devices do not contain dedicated serialization or deserialization circuitry; therefore, shift registers, internal global 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.

Clock Domains

Cyclone devices provide a global clock network and two PLLs (the EP1C3 device only contains one PLL). The global clock network consists of eight global clock lines that drive through the entire device (see Figure 9–2). There are four dedicated clock pins that feed the PLL inputs (two dedicated clocks for each PLL). PLL pins can also act as LVDS input pins. Cyclone PLLs provide general-purpose clocking with clock multiplication and phase shifting as well as external outputs for high-speed differential I/O support. Altera recommends that designers use a data channel for the high-speed clock output for better balanced skew on the transmitter data pins with respect to the output clock.

Figure 9–2. Cyclone Global Clock Network *Note (1)*



Notes to Figure 9–2:

(1) The EP1C3 device in the 100-pin TQFP package has five DPCLK pins (DPCLK2, DPCLK3, DPCLK4, DPCLK6, and DPCLK7).

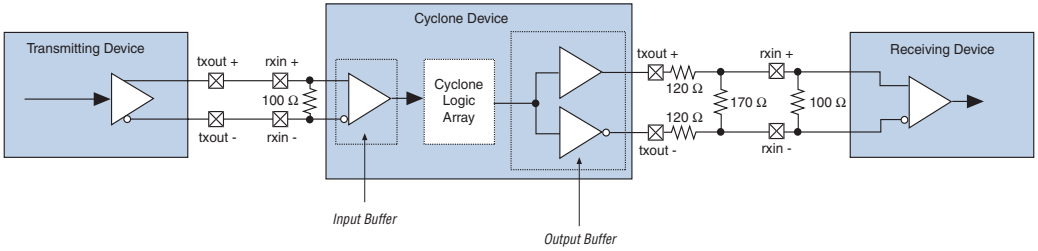
(2) EP1C3 devices only contain one PLL (PLL1).

(3) EP1C3 devices in the 100-pin TQFP package do not support differential clock inputs or outputs.

LVDS Receiver and Transmitter

Figure 9–3 shows a simple point-to-point LVDS application where the source of the data is a LVDS transmitter. These LVDS signals are typically transmitted over a pair of printed circuit board (PCB) traces, but a combination of a PCB trace, connectors, and cables is a common application setup.

Figure 9–3. Typical LVDS Application



The Cyclone LVDS I/O pins meet the IEEE 1596 LVDS specification. Figures 9–4 and 9–5 show the signaling levels for LVDS receiver inputs and transmitter outputs.

Figure 9–4. Receiver Input Waveform for the Differential I/O Standard

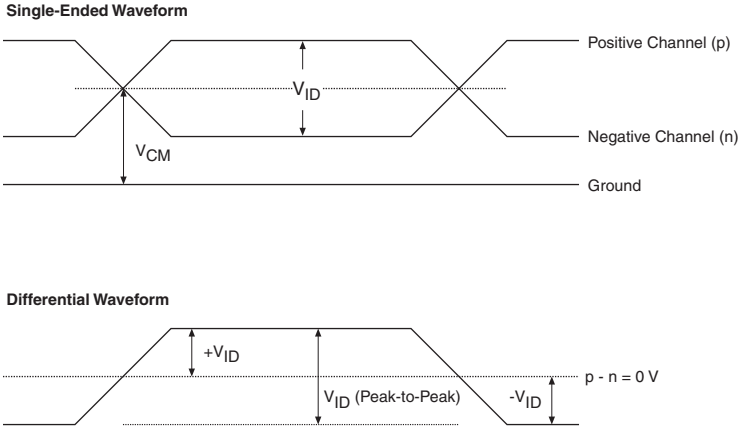


Figure 9–5. Transmitter Output Waveform for Differential I/O Standard

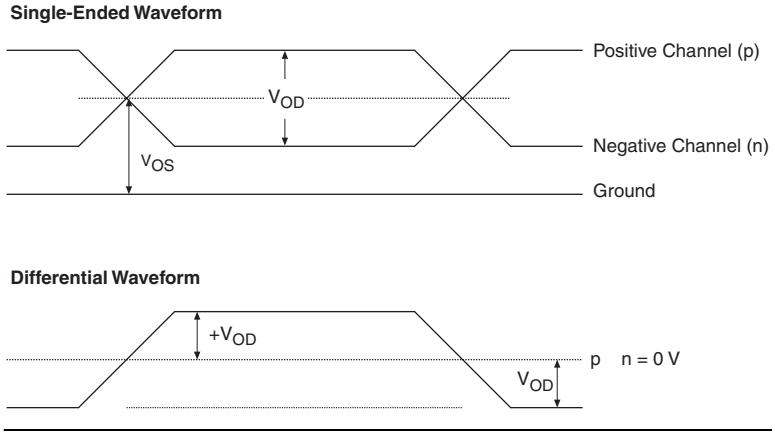


Table 9–2 lists the LVDS I/O specifications.

Table 9–2. LVDS I/O Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{CCINT}	Supply Voltage	—	1.425	1.5	1.575	V
V_{CCIO}	I/O Supply Voltage	—	2.375	2.5	2.625	V
V_{OD}	Differential Output Voltage	R_L = 100 Ω	250	350	550	mV
Δ V_{OD}	Change in V_{OD} between H and L	R_L = 100 Ω	—	—	50	mV
V_{OS}	Output Offset Voltage	R_L = 100 Ω	1.125	1.25	1.375	V
Δ V_{OS}	Change in V_{OS} between H and L	R_L = 100 Ω	—	—	50	mV
V_{ID}	Input differential voltage swing (single-ended)	0.1 V ≤V_{CM} ≤ 2.0 V	100	—	650	mV
V_{IN}	Receiver input voltage range	—	0	—	2.4	V

Table 9–2. LVDS I/O Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{CM}	Receiver input common mode voltage	100 mV ≤V_{ID} ≤ 650 mV	100	—	2,000	mV
R_L	Receiver Differential Input Resistor	—	90	100	110	W

RSDS I/O Standard Support in Cyclone Devices

The RSDS specification defines its use in chip-to-chip applications between the timing controller and the column drivers on display panels. The Cyclone characterization and simulations were performed to meet the National Semiconductor Corp. RSDS Interface Specification. Table 9–3 shows the RSDS electrical characteristics for Cyclone devices.

Table 9–3. RSDS Electrical Characteristics for Cyclone Devices

Symbol	Parameter	Min	Typ	Max	Unit
V_{CCIO}	I/O supply voltage	2.375	2.5	2.625	V
V_{OD}	Differential output voltage	100	200	600	mV
V_{OS}	Output offset voltage	0.5	1.2	1.5	V
V_{TH}	Differential threshold	—	—	±100	mV
V_{CM}	Input common mode voltage	0.3	—	1.5	V

Figures 9–6 and 9–7 show the RSDS receiver and transmitter signal waveforms.

Figure 9–6. Receiver Input Signal Level Waveforms for RSDS

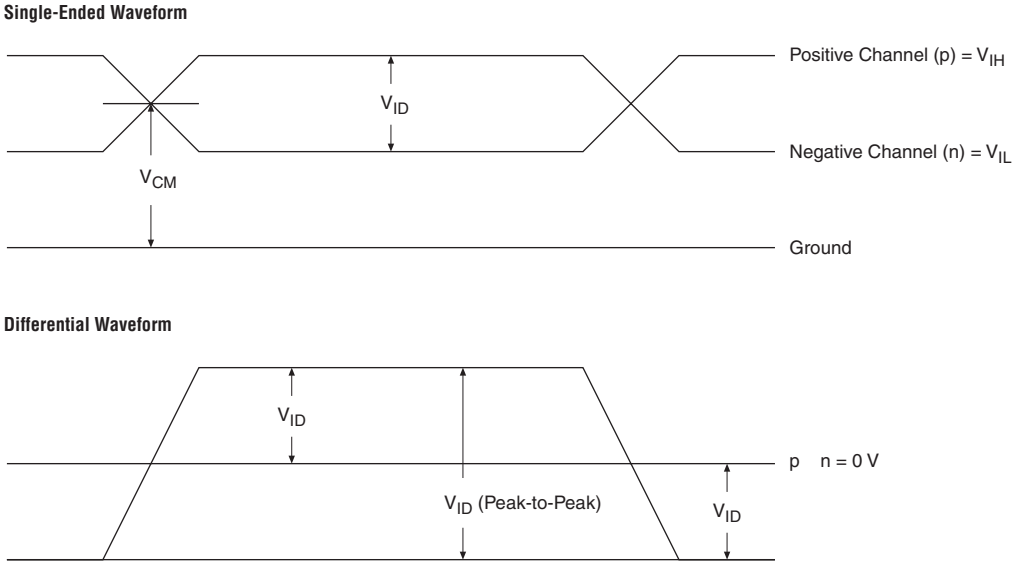
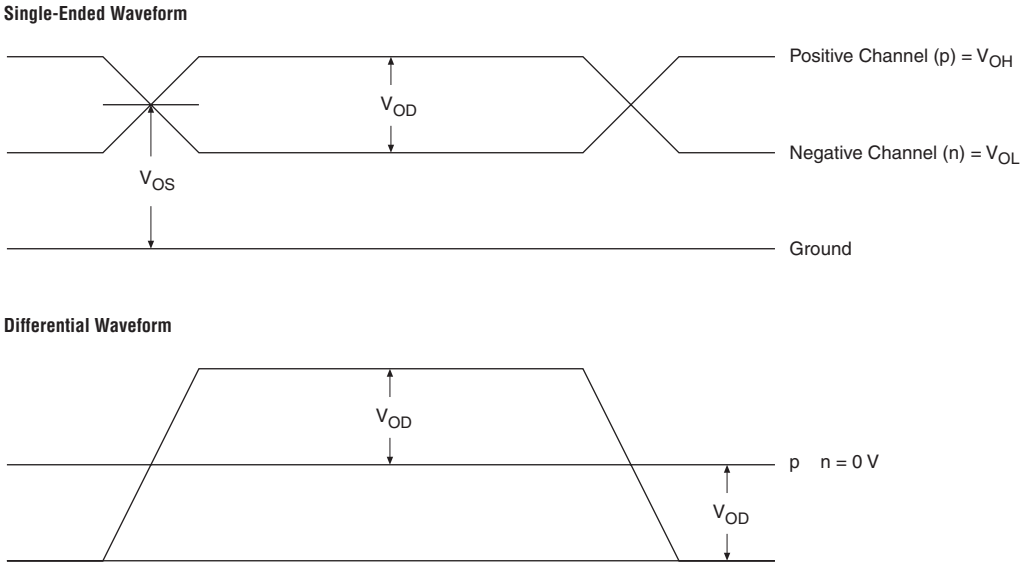


Figure 9–7. Transmitter Output Signal Level Waveforms for RSDS



Cyclone FPGA devices support all three bus configuration types as defined by the RSDS specification:

- Multi-drop bus with double termination
- Multi-drop bus with single end termination
- Double multi-drop bus with single termination

Designing with RSDS

Cyclone devices support the RSDS standard using the LVDS I/O buffer types. For receivers, the LVDS input buffer can be used without any changes. For transmitters, the LVDS output buffer can be used with the external resistor network shown in Figure 9–8.

Figure 9–8. RSDS Resistor Network

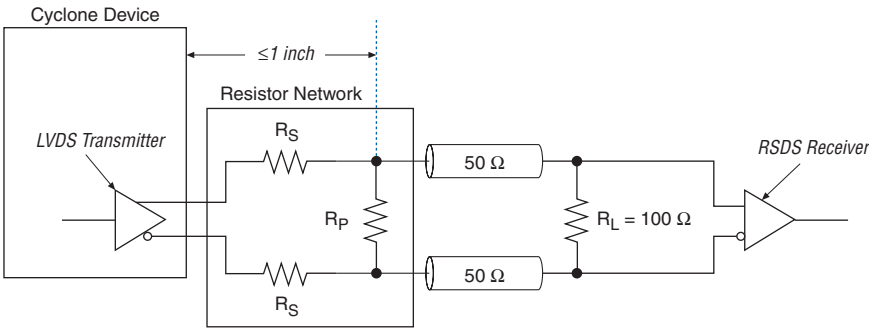


Table 9–4 shows the resistor values recommended for each RSDS bus configuration type.

Table 9–4. Recommended Resistor Values

Bus Configuration Type	R_S (Ω)	R_P (Ω)
Multi-drop bus with double termination	160	145
Multi-drop bus with single end termination	226	124
Double multi-drop bus with single termination	226	124

For more information about RSDS bus configuration types, refer to the RSDS specification from the National Semiconductor website (www.national.com).

A resistor network is required to attenuate the LVDS output voltage swing to meet the RSDS specifications. The resistor network values can be modified to reduce power or improve the noise margin. The resistor values chosen should satisfy the following equation:

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

For example, in the multi-drop bus with single end termination or double multi-drop bus with single termination bus configuration, the resistor values can be modified to $R_S = 200\ \Omega$ and $R_P = 130\ \Omega$ to increase the V_{OD} or voltage swing of the signal.

Additional simulations using the IBIS models should be performed to validate that custom resistor values meet the RSDS requirements.

RSDS Software Support

When designing for the RSDS I/O standard, assign the LVDS I/O standard to the I/O pins intended for RSDS in the Quartus II software. Contact Altera Applications for reference designs.

High-Speed I/O Timing in Cyclone Devices

Since LVDS and RSDS data communication is source synchronous, timing analysis is different than other I/O standards. You must understand how to analyze timing for the high-speed I/O signal, which is based on skew between the data and the clock signal.

You should also consider board skew, cable skew, and clock jitter in your calculation. This section provides details on high-speed I/O standards timing parameters in Cyclone devices.

Table 9–5 defines the parameters of the timing diagram shown in Figure 9–9.

Table 9–5. High-Speed I/O TIming Definitions *Note (1)*

Parameter	Symbol	Description
High-speed clock frequency	f_{HSCLK}	High-speed receiver/transmitter input clock frequency.
High-speed I/O data rate	HSIODR	High-speed receiver/transmitter input and output data rate.
High-speed external output clock	$f_{HSCLKOUT}$	High-speed transmitter external output clock frequency using an LVDS data channel.
Channel-to-channel skew	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 within the sampling window. SW = t_{SW} (max) – t_{SW} (min).
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 = (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.
Rise time	t_{RISE}	Low-to-high transmission time.
Fall time	t_{FALL}	High-to-low transmission time.
Duty cycle	t_{DUTY}	Duty cycle on LVDS transmitter output clock.
PLL lock time	t_{LOCK}	Lock time for the PLL

Note to Table 9–5:

(1) The TCCS specification applies to the whole bank of LVDS as long as the SERDES logic is placed within the LAB adjacent to the output pins.

Figure 9–9. High-Speed I/O Timing Diagram

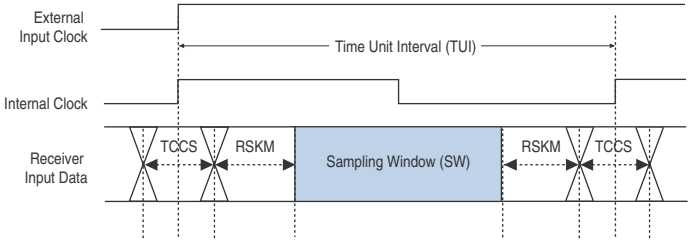
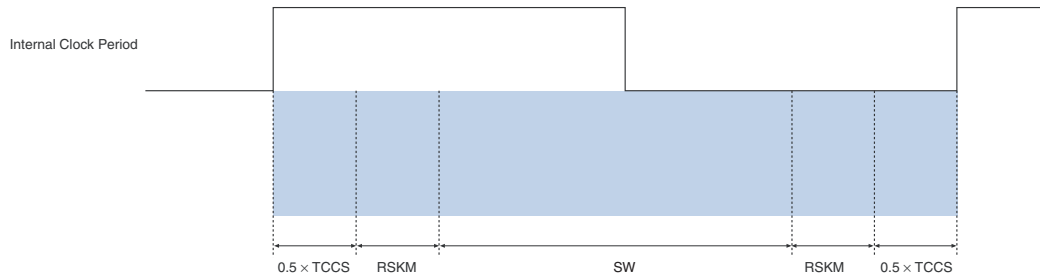


Figure 9–10 shows the high-speed I/O timing budget.

Figure 9–10. Cyclone High-Speed I/O Timing Budget *Note (1)*



Note to Figure 9–10:

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

Table 9–6 shows the RSDS timing budget for Cyclone devices at 311 Mbps.

Table 9–6. RSDS Timing Specification for Cyclone Devices (Part 1 of 2)

Symbol	Conditions	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSCLK}	×10	15.625	NA	31.1	15.625	NA	31.1	15.625	NA	31.1	MHz
	×8	15.625	NA	38.875	15.625	NA	38.875	15.625	NA	38.875	MHz
	×7	17.857	NA	44.429	17.857	NA	44.429	17.857	NA	44.429	MHz
	×4	15.625	NA	77.75	15.625	NA	77.75	15.625	NA	77.75	MHz
	×2	15.625	NA	155.5	15.625	NA	155.5	15.625	NA	155.5	MHz
	×1 *(1)*	15.625	NA	275	15.625	NA	275	15.625	NA	275	MHz

Table 9–6. RSDS Timing Specification for Cyclone Devices (Part 2 of 2)

Symbol	Conditions	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
HSIODR	×10	156.25	NA	311	156.25	NA	311	156.25	NA	311	Mbps
	×8	125	NA	311	125	NA	311	125	NA	311	Mbps
	×7	125	NA	311	125	NA	311	125	NA	311	Mbps
	×4	62.5	NA	311	62.5	NA	311	62.5	NA	311	Mbps
	×2	31.25	NA	311	31.25	NA	311	31.25	NA	311	Mbps
	×1 *(1)*	15.625	NA	275	15.625	NA	275	15.625	NA	275	Mbps
$f_{HSCLKOUT}$	—	15.625	NA	275	15.625	NA	275	15.625	NA	275	MHz
TCCS	—	NA	NA	±150	NA	NA	±150	NA	NA	±150	ps
SW	—	NA	NA	500	NA	NA	550	NA	NA	550	ps
Input jitter tolerance (peak-to-peak)	—	NA	NA	400	NA	NA	400	NA	NA	400	ps
Output jitter (peak-to-peak)	—	NA	NA	400	NA	NA	400	NA	NA	400	ps
t_{RISE}	—	150	200	250	150	200	250	150	200	250	ps
t_{FALL}	—	150	200	250	150	200	250	150	200	250	ps
t_{DUTY}	—	45	50	55	45	50	55	45	50	55	%
t_{LOCK}	—	NA	NA	100	NA	NA	100	NA	NA	100	µs

Note to Table 9–6:

(1) The PLL must divide down the input clock frequency to have the internal clock frequency meet the specification shown in the *DC and Switching Characteristics* chapter in the *Cyclone Device Handbook*.

Table 9–7 shows the LVDS timing budget for Cyclone devices at 640 Mbps.

Table 9–7. LVDS Timing Specification for Cyclone Devices

Symbol	Conditions	-6 Speed Grade			-7 Speed Grade			-8 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSCLK}	×10	15.625	NA	64	15.625	NA	64	15.625	NA	55	MHz
	×8	15.625	NA	80	15.625	NA	80	15.625	NA	68.75	MHz
	×7	17.857	NA	91.429	17.857	NA	91.429	17.857	NA	78.571	MHz
	×4	15.625	NA	160	15.625	NA	160	15.625	NA	137.5	MHz
	×2	15.625	NA	320	15.625	NA	320	15.625	NA	275	MHz
	×1 *(1)*	15.625	NA	567	15.625	NA	549	15.625	NA	531	MHz
HSIODR	×10	156.25	NA	640	156.25	NA	640	156.25	NA	550	Mbps
	×8	125	NA	640	125	NA	640	125	NA	550	Mbps
	×7	125	NA	640	125	NA	640	125	NA	550	Mbps
	×4	62.5	NA	640	62.5	NA	640	62.5	NA	550	Mbps
	×2	31.25	NA	640	31.25	NA	640	31.25	NA	550	Mbps
	×1 *(1)*	15.625	NA	320	15.625	NA	320	15.625	NA	275	Mbps
$f_{HSCLKOUT}$		15.625	NA	320	15.625	NA	320	15.625	NA	275	MHz
TCCS		NA	NA	±150	NA	NA	±150	NA	NA	±150	ps
SW		NA	NA	500	NA	NA	500	NA	NA	550	ps
Input jitter tolerance (peak-to-peak)		NA	NA	400	NA	NA	400	NA	NA	400	ps
Output jitter (peak-to-peak)		NA	NA	400	NA	NA	400	NA	NA	400	ps
t_{RISE}		150	200	250	150	200	250	150	200	250	ps
t_{FALL}		150	200	250	150	200	250	150	200	250	ps
t_{DUTY}		45	50	55	45	50	55	45	50	55	%
t_{LOCK}		NA	NA	100	NA	NA	100	NA	NA	100	µs

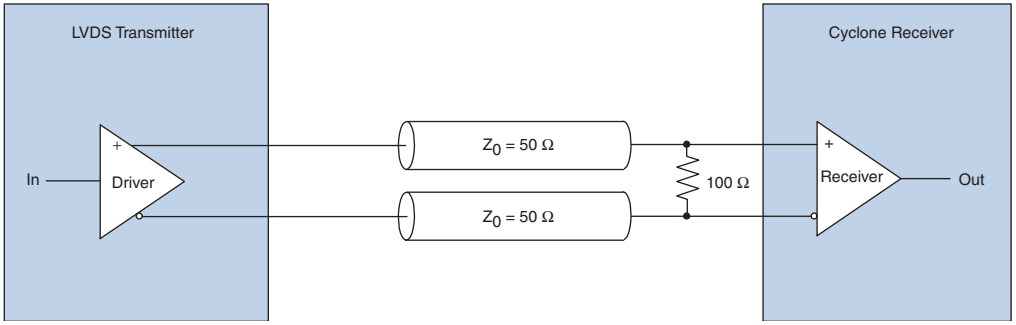
Note to Table 9–7:

(1) The PLL must divide down the input clock frequency to have the internal clock frequency meet the specification shown in the *DC and Switching Characteristics* chapter in the *Cyclone Device Handbook*.

LVDS Receiver and Transmitter Termination

Receiving LVDS signals on Cyclone I/O pins is straightforward, and can be done by assigning LVDS to desired pins in the Quartus II software. A 100-Ω parallel terminator is required at the receiver input pin, as shown in Figure 9–11.

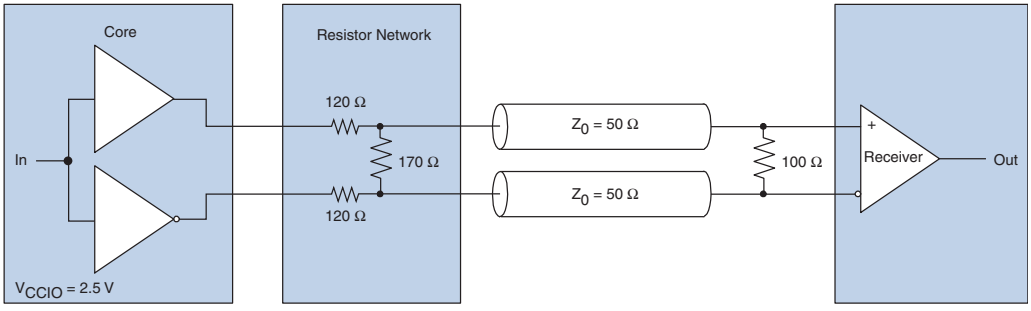
Figure 9–11. Termination Scheme on Cyclone LVDS Receiver



For PCB layout guidelines, refer to *AN 224: High-Speed Board Layout Guidelines*.

Cyclone LVDS transmitter signals are generated using a resistor network, as shown in Figure 9–12 (with R_S = 120 Ω and R_{DIV} = 170 Ω). The resistor network attenuates the driver outputs to levels similar to the LVDS signaling, which is recognized by LVDS receivers with minimal effect on 50-Ω trace impedance.

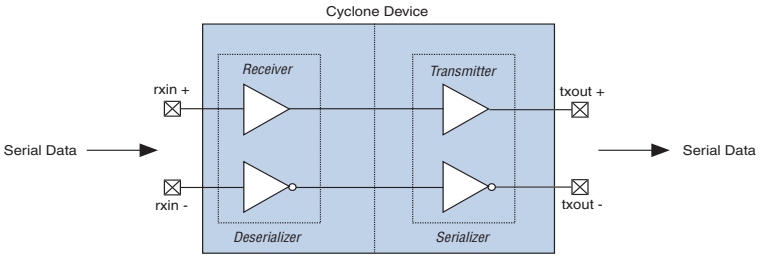
Figure 9–12. Termination Scheme on Cyclone LVDS Transmitter



Implementing Cyclone LVDS and RSDS I/O Pins in the Quartus II Software

For differential signaling, the receiver must deserialize the incoming data and send it to the internal logic as a parallel signal. Accordingly, the transmitter must serialize the parallel data coming from the internal logic to send it off-chip (see Figure 9–13).

Figure 9–13. Deserialization and Serialization at Receiver and Transmitter



Although Cyclone devices do not incorporate a dedicated serializer/deserializer (SERDES), you can incorporate these functions in your design using the Quartus II software. The device implements the SERDES in logic elements (LEs) and requires a PLL.

LVDS in Cyclone devices is implememented using megafunctions in Quartus II software. The `altlvds_rx` megafunction implements a deserialization receiver. The `altlvds_tx` megafunction implements a serialization transmitter.

The placement of the LE registers is handled by the LVDS MegaWizard® in the Quartus II software. The Cyclone device DDIO logic placer in the Quartus II software only places the DDIO output registers according to Altera's recommendation and does not check if it meets the TCCS specification. There is no timing analysis done in the Quartus II software to report the TCCS. Verify timing anaysis by running the Timing Analyzer in the Quartus II software.

Refer to the Quartus II software documenation and the Quartus II Help for more information on these megafunctions. Follow the recommendations in Tables 9–8 and 9–9 for PLL phase shift settings. The operation of these settings are guaranteed by operation.

The required receiver PLL phase settings for top and bottom I/O banks (I/O banks 2 and 4) based on high-speed I/O data rate and operating mode are shown in Table 9–8.

Table 9–8. Receiver PLL Phase Settings for Top and Bottom I/O Banks

Device	Phase Shift (Degree)			Unit
	0	22.5	45	
EP1C3	—	—	300 to 640	Mbps
EP1C4	—	601 to 640	300 to 600	Mbps
EP1C6	—	601 to 640	300 to 600	Mbps
EP1C12	—	451 to 640	300 to 450	Mbps
EP1C20	551 to 640	300 to 550	—	Mbps

The required receiver PLL phase settings for right and left I/O banks (I/O Bank 1 and 3) based on high-speed I/O data rate and operating mode are shown in Table 9–9.

Table 9–9. Receiver PLL Phase Settingsfor Right and Left I/O Banks

Device	Phase Shift (Degree)				Unit
	–22.5	0	22.5	45	
EP1C3	—	—	451 to 640	300 to 450	Mbps
EP1C4	—	551 to 640	300 to 550	—	Mbps
EP1C6	—	—	451 to 640	300 to 450	Mbps
EP1C12	601 to 640	451 to 600	300 to 450	—	Mbps
EP1C20	501 to 640	300 to 500	—	—	Mbps

Design Guidelines

To implement LVDS in Cyclone devices, adhere to the following design guidelines in the Quartus II software.

- Route LVDS `CLKOUT` to pins through regular user LVDS pins. This routing provides better TCCS margin.
- To meet the t_{SU} and t_{CO} timing requirement between serial and parallel registers, use the I/O registers of the input and output pins.
- f_{MAX} is limited by the delay between the IOE and the next logic element (LE) register. To achieve an f_{MAX} of 320 MHz, the delay between the IOE and the next LE register at the receiver and transmitter side must not be more than 3.125 ns.
- The best location to implement the shift registers is within the LAB adjacent to the input or output pin.
- LVDS data and clock should be aligned at the output pin. If these signals are not aligned, use a phase shift to align them.

Differential Pad Placement Guidelines

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



For placing single-ended pads with respect to differential pads in Cyclone devices, refer to the guidelines in the *Using Selectable I/O Standards in Cyclone Devices* chapter in the *Cyclone Device Handbook*.

Board Design Considerations

This section explains how to get the optimal performance from the Cyclone I/O block and ensure first-time success in implementing a functional design with optimal signal quality. The critical issues of controlled impedance of traces and connectors, differential routing, and termination techniques must all be considered to get the best performance from the integrated circuit (IC). Use this chapter together with the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

The Cyclone device generates signals that travel over the media at frequencies as high as 640 Mbps. Use the following general guidelines:

- 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 LVDS pairs, as much as possible. Routing the pair of traces close to each other will maximize the common-mode rejection ratio (CMRR)

- Longer traces have more inductance and capacitance. These traces should 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° or 45° corners.
- Use high-performance connectors.
- Design backplane and card traces so that trace impedance matches the connector's and/or the termination's impedance.
- Keep equal number of vias for both signal traces.
- Create equal trace lengths to avoid skew between signals. Unequal trace lengths result in misplaced crossing points and decrease system margins as the TCCS value increases.
- Limit vias because they cause discontinuities.
- Use the common bypass capacitor values such as 0.001 µF, 0.01 µF, and 0.1 µF to decouple the high-speed PLL power and ground planes.
- Keep switching 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.

Conclusion

Cyclone LVDS I/O capabilities enable you to keep pace with increasing design complexity while offering the lowest-cost FPGA on the market. Support for I/O standards including LVDS allows Cyclone devices to fit into a wide variety of applications. Taking advantage of these I/O standards and Cyclone pricing allows you to lower your design costs while remaining on the cutting edge of technology.

Referenced Documents

This chapter references the following documents:

- *AN 224: High-Speed Board Layout Guidelines*
- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*
- *DC and Switching Characteristics* chapter in the *Cyclone Device Handbook*
- *Using Selectable I/O Standards in Cyclone Devices* chapter in the *Cyclone Device Handbook*

Document Revision History

Table 9–10 shows the revision history for this chapter.

Table 9–10. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.6	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.5	Added document revision history.	—
August 2005 v1.4	Updated minimum LVDS LOD value listed in Table 9-2.	—
February 2005 v1.3	Minor updates.	—
October 2003 v1.2	• Added RSDS information. • Removed V_{SS} from Figure 9–5. • Added RSDS and LVDS timing information in Tables 9–6 and 9–7, respectively. • Updated Implementing Cyclone LVDS and RSDS I/O Pins in the Quartus II Software section, including addition of the PLL Circuit section.	—
September 2003 v1.1	Updated LVDS data rates to 640 Mbps from 311 Mbps.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

Section V. Design Considerations

This section provides documentation on design considerations when utilizing Cyclone devices. In addition to these design considerations, refer to the Intellectual Property section of the Altera web site for a complete offering of IP cores for Cyclone devices.

This section contains the following chapters:

- Chapter 10, Implementing Double Data Rate I/O Signaling in Cyclone Devices
- Chapter 11. Using Cyclone Devices in Multiple-Voltage Systems
- Chapter 12. Designing with 1.5-V Devices

Revision History

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

10. Implementing Double Data Rate I/O Signaling in Cyclone Devices

C51010-1.2

Introduction

Double data rate (DDR) transmission is used in many applications where fast data transmission is needed, such as memory access and first-in first-out (FIFO) memory structures. DDR uses both edges of a clock to transmit data, which facilitates data transmission at twice the rate of a single data rate (SDR) architecture using the same clock speed. This method also reduces the number of I/O pins required to transmit data.

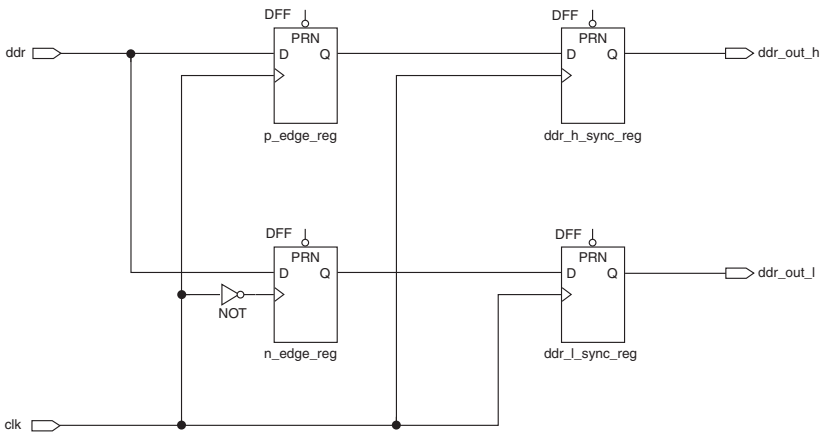
This chapter shows implementations of a double data rate I/O interface using Cyclone® devices. Cyclone devices support DDR input, DDR output, and bidirectional DDR signaling.

For more information on using Cyclone devices in applications with DDR SDRAM and FCRAM memory devices, refer to "DDR Memory Support" on page 10–4.

Double Data Rate Input

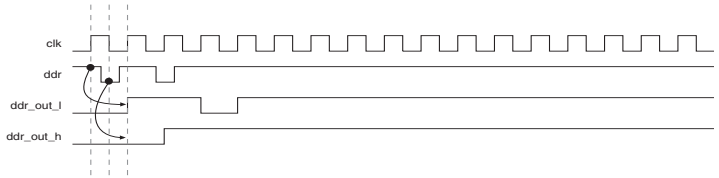
The DDR input implementation shown in Figure 10–1 uses four internal logic element (LE) registers located in the logic array block (LAB) adjacent to the DDR input pin. The DDR data is fed to the first two of four registers. One register captures the DDR data present during the rising edge of the clock. The second register captures the DDR data present during the falling edge of the clock.

Figure 10–1. Double Data Rate Input Implementation



The third and fourth registers synchronize the two data streams to the rising edge of the clock. Figure 10–2 shows examples of functional waveforms from a double data rate input implementation.

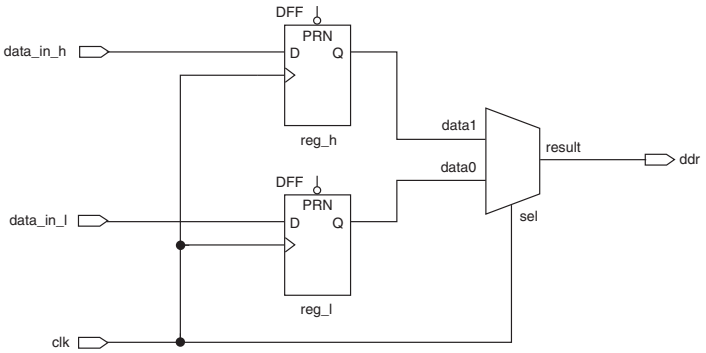
Figure 10–2. Double Data Rate Input Functional Waveforms



Double Data Rate Output

Figure 10–3 shows a schematic representation of double data rate output implemented in a Cyclone device. The DDR output logic is implemented using LEs in the LAB adjacent to the output pin. Two registers are used to synchronize two serial data streams. The registered outputs are then multiplexed by the common clock to drive the DDR output pin at two times the data rate.

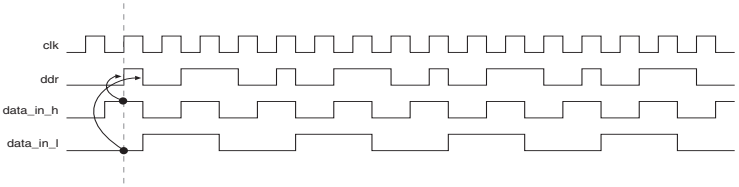
Figure 10–3. Double Data Rate Output Implementation



While the clock signal is logic-high, the output from `reg_h` is driven onto the DDR output pin. While the clock signal is logic-low, the output from `reg_l` is driven onto the DDR output pin. The DDR output pin can be any available user I/O pin.

Figure 10–4 shows examples of functional waveforms from a double data rate output implementation.

Figure 10–4. Double Data Rate Output Waveforms



Bidirectional Double Data Rate

Figure 10–5 shows a bidirectional DDR interface, constructed using the DDR input and DDR output examples described in the previous two sections. As with the DDR input and DDR output examples, the bidirectional DDR pin can be any available user I/O pin, and the registers used to implement DDR bidirectional logic are LEs in the LAB adjacent to that pin. The tri-state buffer (TRI) controls when the device drives data onto the bidirectional DDR pin.

Figure 10–5. Bidirectional Double Data Rate Implementation

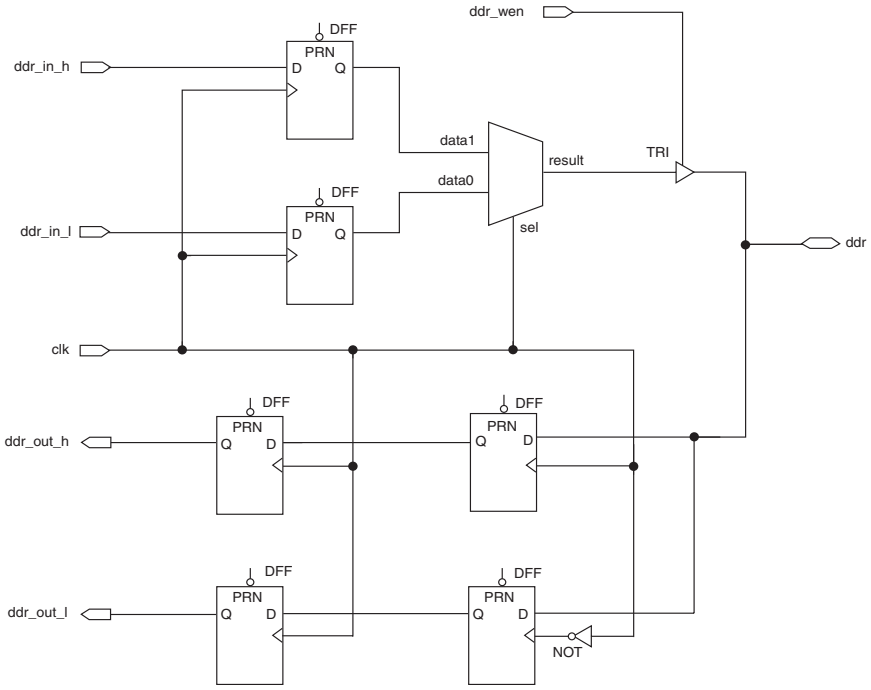
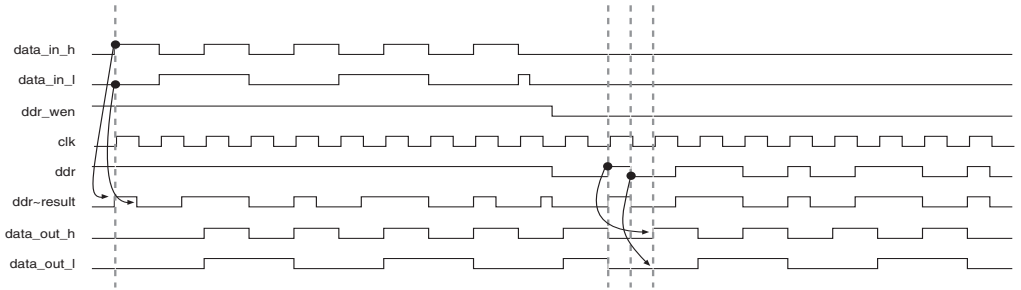


Figure 10–6 shows example waveforms from a bidirectional double data rate implementation.

Figure 10–6. Double Data Rate Bidirectional Waveforms



DDR Memory Support

The Cyclone device family supports both DDR SDRAM and FCRAM memory interfaces up to 133 MHz.

For more information about extended DDR memory support in Cyclone devices, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

Conclusion

Utilizing both the rising and falling edges of a clock signal, double data rate transmission is a popular strategy for increasing the speed of data transmission while reducing the required number of I/O pins. Cyclone devices can be used to implement this strategy for use in applications such as FIFO structures, SDRAM/FCRAM interfaces, as well as other time-sensitive memory access and data-transmission situations.

Referenced Documents

This chapter references the following document:

- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*

Document Revision History

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

Table 10–1. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.2	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.1	Added document revision history.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

11. Using Cyclone Devices in Multiple-Voltage Systems

C51011-1.2

Introduction

To meet the demand for higher system speed in data communications, semiconductor vendors use increasingly advanced processing technologies requiring lower operating voltages. As a result, printed circuit boards (PCBs) often incorporate devices conforming to one of several voltage level I/O standards, such as 3.3-V, 2.5-V, 1.8-V and 1.5-V. A mixture of components with various voltage level I/O standards on a single PCB is inevitable.

In order to accommodate this mixture of devices on a single PCB, a device that can act as a bridge or interface between these devices is needed. The Cyclone® device family's MultiVolt™ I/O operation capability meets the increasing demand for compatibility with devices of different voltages. MultiVolt I/O operation separates the power supply voltage from the output voltage, enabling Cyclone devices to interoperate with other devices using different voltage levels on the same PCB.

In addition to MultiVolt I/O operation, this chapter discusses several other features that allow you to use Cyclone devices in multiple-voltage systems without damaging the device or the system, including:

- Hot-Socketing—add and remove Cyclone devices to and from a powered-up system without affecting the device or system operation
- Power-Up Sequence flexibility—Cyclone devices can accommodate any possible power-up sequence
- Power-On Reset—Cyclone devices maintain a reset state until voltage is within operating range

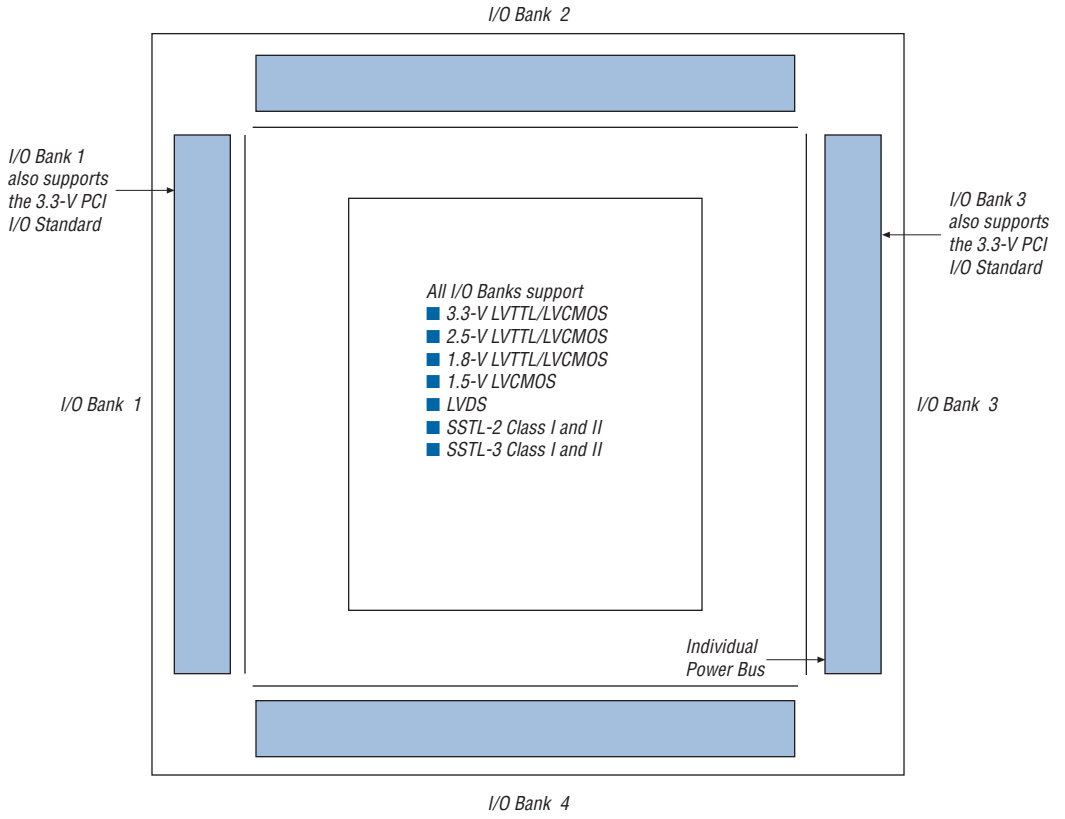
I/O Standards

The I/O buffer of a Cyclone device is programmable and supports a wide range of I/O voltage standards. Each I/O bank in a Cyclone device can be programmed to comply with a different I/O standard. All I/O banks can be configured with the following I/O standards:

- 3.3-V LVTTL/LVCMOS
- 2.5-V LVTTL/LVCMOS
- 1.8-V LVTTL/LVCMOS
- 1.5-V LVCMOS
- LVDS
- SSTL-2 Class I and II
- SSTL-3 Class I and II

I/O banks 1 and 3 also include 3.3-V PCI I/O standard interface capability. See Figure 11–1.

Figure 11–1. I/O Standards Supported by Cyclone Devices *Notes (1), (2), (3)*



Notes to Figure 11–1

(1) Figure 1 is a top view of the silicon die.

(2) Figure 1 is a graphical representation only. Refer to the pin list and the Quartus® II software for exact pin locations.

(3) The EP1C3 device in the 100-pin thin quad flat pack (TQFP) package does not have support for a PLL LVDS input or an external clock output.

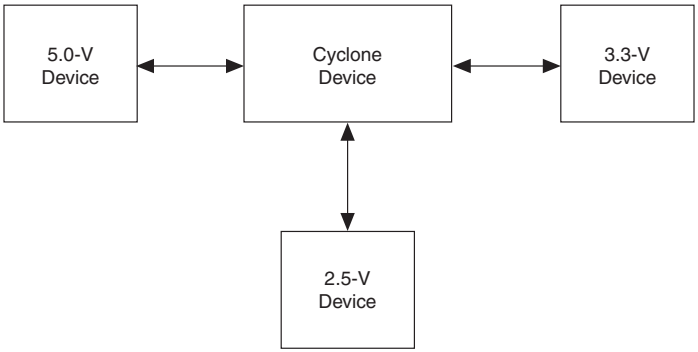
MultiVolt I/O Operation

Cyclone devices include MultiVolt I/O operation capability, allowing the core and I/O blocks of the device to be powered-up with separate supply voltages. The VCCINT pins supply power to the device core and the VCCIO pins supply power the device's I/O buffers.



Supply all device VCCIO pins that have MultiVolt I/O capability at the same voltage level (e.g., 3.3-V, 2.5-V, 1.8-V, or 1.5-V). See Figure 11–2.

Figure 11–2. Implementing a Multiple-Voltage System with a Cyclone Device



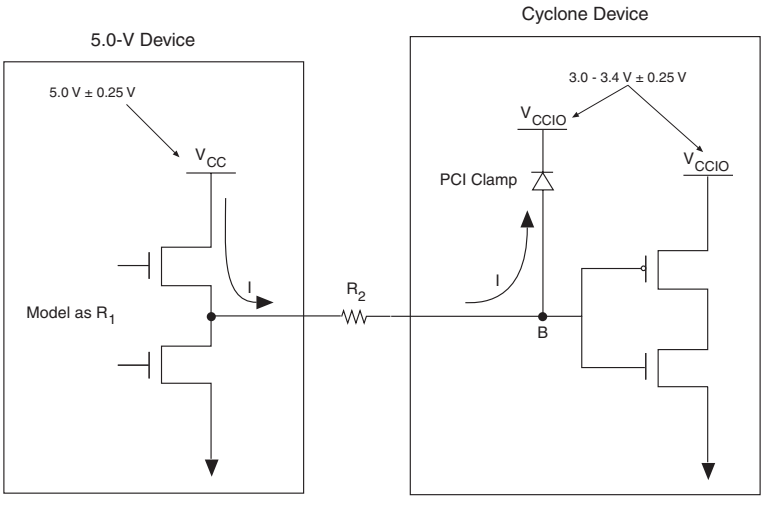
5.0-V Device Compatibility

A Cyclone device may not correctly interoperate with a 5.0-V device if the output of the Cyclone device is connected directly to the input of the 5.0-V device. If V_{OUT} of the Cyclone device is greater than V_{CCIO}, the PMOS pull-up transistor still conducts if the pin is driving high, preventing an external pull-up resistor from pulling the signal to 5.0-V.

A Cyclone device can drive a 5.0-V LVTTL device by connecting the VCCIO pins of the Cyclone device to 3.3 V. This is because the output high voltage (V_{OH}) of a 3.3-V interface meets the minimum high-level voltage of 2.4-V of a 5.0-V LVTTL device. (A Cyclone device cannot drive a 5.0-V LVCMOS device.)

Because the Cyclone devices are 3.3-V, 64- and 32-bit, 66- and 33-MHz PCI compliant the input circuitry accepts a maximum high-level input voltage (V_{IH}) of 4.1-V. To drive a Cyclone device with a 5.0-V device, you must connect a resistor (R_2) between the Cyclone device and the 5.0-V device. See Figure 11–3.

Figure 11–3. Driving a Cyclone Device with a 5.0-Volt Device

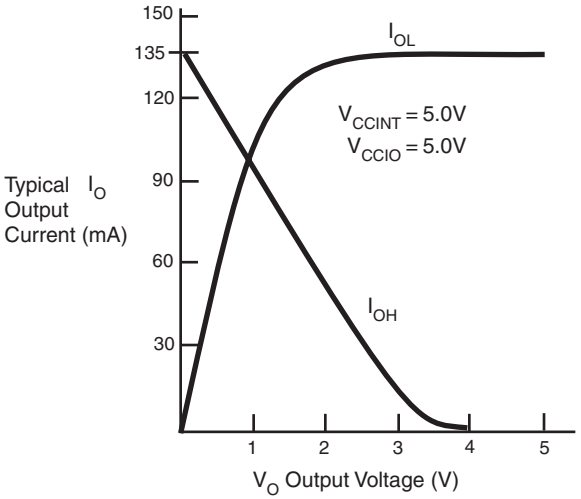


If V_{CCIO} is between 3.0-V and 3.6-V and the PCI clamping diode (not available on EP1C3 devices) is enabled, the voltage at point B in Figure 11–3 is 4.3-V or less. To limit large current draw from the 5.0-V device, R_2 should be small enough for a fast signal rise time and large enough so that it does not violate the high-level output current (I_{OH}) specifications of the devices driving the trace. The PCI clamping diode in the Cyclone device can support 25mA of current.

To compute the required value of R_2, first calculate the model of the pull-up transistors on the 5.0-V device. This output resistor (R_1) can be modeled by dividing the 5.0-V device supply voltage (V_{CC}) by the I_{OH}: $R_1 = V_{CC}/I_{OH}$.

Figure 11–4 shows an example of typical output drive characteristics of a 5.0-V device.

Figure 11–4. Output Drive Characteristics of a 5.0-V Device



As shown above, R_1 = 5.0-V/135 mA.



The values usually shown in data sheets reflect typical operating conditions. Subtract 20% from the data sheet value for guard band. This subtraction applied to the above example gives R_1 a value of 30 Ω

R_2 should be selected to not violate the driving device's IOH specification. For example, if the above device has a maximum IOH of 8 mA, given the PCI clamping diode, $V_{IN} = V_{CCIO} + 0.7\text{-V} = 3.7\text{-V}$. Given that the maximum supply load of a 5.0-V device (V_{CC}) will be 5.25-V, the value of R_2 can be calculated as follows:

$$R_2 = \frac{(5.25\text{V} - 3.7\ \text{V}) - (8\ \text{mA} \times 30\ \Omega)}{8\ \text{mA}} = 164\ \Omega$$

This analysis assumes worst-case conditions. If your system will not see a wide variation in voltage-supply levels, you can adjust these calculations accordingly.



Because 5.0-V device tolerance in Cyclone devices requires use of the PCI clamp (not available on EP1C3 devices), and this clamp is activated during configuration, 5.0-V signals may not be driven into the device until it is configured.

Hot-Socketing

Hot-socketing, also known as hot-swapping, refers to inserting or removing a board or device into or out of a system board while system power is on. For a system to support hot-socketing, plug-in or removal of the subsystem or device must not damage the system or interrupt system operation.

All devices in the Cyclone family are designed to support hot-socketing without special design requirements. The following features have been implemented in Cyclone devices to facilitate hot-socketing:

- Devices can be driven before power-up with no damage to the device.
- I/O pins remain tri-stated during power-up.
- Signal pins do not drive the V_{CCIO} or V_{CCINT} power supplies.



Because 5.0-V tolerance in Cyclone devices require the use of the PCI clamping diode, and the clamping diode is only available after configuration has finished, be careful not to connect 5.0-V signals to the device.

Devices Can Be Driven before Power-Up

The device I/O pins, dedicated input pins, and dedicated clock pins of Cyclone devices can be driven before or during power-up without damaging the devices.

I/O Pins Remain Tri-Stated during Power-Up

A device that does not support hot-socketing may interrupt system operation or cause contention by driving out before or during power-up. For Cyclone devices, I/O pins are tri-stated before and during power-up and configuration, and will not drive out.

Signal Pins Do Not Drive the V_{CCIO} or V_{CCINT} Power Supplies

A device that does not support hot-socketing will short power supplies together when powered-up through its signal pins. This irregular power-up can damage both the driving and driven devices and can disrupt card power-up.

In Cyclone devices, there is no current path from I/O pins, dedicated input pins, or dedicated clock pins to the `VCCIO` or `VCCINT` pins before or during power-up. A Cyclone device may be inserted into (or removed from) a powered-up system board without damaging or interfering with system-board operation. When hot-socketing, Cyclone devices have a minimal effect on the signal integrity of the backplane.



The maximum DC current when hot-socketing Cyclone devices is less than 300 µA, whereas the maximum AC current during hot-socketing is less than 8 mA for a period of 10ns or less.

During hot-socketing, the signal pins of a device may be connected and driven by the active system before the power supply can provide current to the device V_{CC} and ground planes. Known as latch-up, this condition can cause parasitic diodes to turn on within the device, causing the device to consume a large amount of current, and possibly causing electrical damage. This operation can also cause parasitic diodes to turn on inside of the driven device. Cyclone devices are immune to latch-up when hot-socketing.

Power-Up Sequence

Because Cyclone devices can be used in a multi-voltage environment, they are designed to tolerate any possible power-up sequence. Either V_{CCINT} or V_{CCIO} can initially supply power to the device, and 3.3-V, 2.5-V, 1.8-V, or 1.5-V input signals can drive the devices without special precautions before V_{CCINT} or V_{CCIO} is applied. Cyclone devices can operate with a V_{CCIO} voltage level that is higher than the V_{CCINT} level. You can also change the V_{CCIO} supply voltage while the board is powered-up. However, you must ensure that the V_{CCINT} and V_{CCIO} power supplies stay within the correct device operating conditions.

When V_{CCIO} and V_{CCINT} are supplied from different power sources to a Cyclone device, a delay between V_{CCIO} and V_{CCINT} may occur. Normal operation does not occur until both power supplies are in their recommended operating range. When V_{CCINT} is powered-up, the IEEE Std. 1149.1 Joint Test Action Group (JTAG) circuitry is active. If `TMS` and `TCK` are connected to V_{CCIO} and V_{CCIO} is not powered-up, the JTAG signals are left floating. Thus, any transition on `TCK` can cause the state machine to transition to an unknown JTAG state, leading to incorrect operation when V_{CCIO} is finally powered-up. To disable the JTAG state during the power-up sequence, `TCK` should be pulled low to ensure that an inadvertent rising edge does not occur on `TCK`.

Power-On Reset

When designing a circuit, it is important to consider system state at power-up. Cyclone devices maintain a reset state during power-up. When power is applied to a Cyclone device, a power-on-reset event occurs if V_{CC} reaches the recommended operating range within a certain period of time (specified as a maximum V_{CC} rise time). A POR event does not occur if these conditions are not met because slower rise times can cause incorrect device initialization and functional failure. The V_{CCIO} level of the I/O banks that contains configuration pins must also reach an acceptable level to trigger POR event.



If V_{CCINT} does not remain in the specified operating range, operation is not assured until V_{CCINT} re-enters the range.

Conclusion

PCBs often contain a mix of 5.0-V, 3.3-V, 2.5-V, 1.8-V, and 1.5-V devices. The Cyclone device family's MultiVolt I/O operation capability allows you to incorporate newer-generation devices with devices of varying voltage levels. This capability also enables the device core to run at its core voltage, V_{CCINT}, while maintaining I/O pin compatibility with other logic levels. Altera has taken further steps to make system design easier by designing devices that allow V_{CCINT} and V_{CCIO} to power-up in any sequence and by incorporating support for hot-socketing.

Document Revision History

Table 11–1 shows the revision history for this chapter.

Table 11–1. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.3	Minor textual and style changes.	—
January 2007 v1.2	Updated "Power-On Reset" section.	—
October 2003 v1.1	Added 64-bit PCI support information.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

12. Designing with 1.5-V Devices

C51012-1.4

Introduction

The Cyclone® FPGA family provides the best solution for high-volume, cost-sensitive applications. A Cyclone device is fabricated on a leading-edge 1.5-V, 0.13-µm, all-layer copper SRAM process.

Using a 1.5-V operating voltage provides the following advantages:

- Lower power consumption compared to 2.5-V or 3.3-V devices.
- Lower operating temperature.
- Less need for fans and other temperature-control elements.

Since many existing designs are based on 5.0-V, 3.3-V and 2.5-V power supplies, a voltage regulator may be required to lower the voltage supply level to 1.5-V. This document provides guidelines for designing with Cyclone devices in mixed-voltage and single-voltage systems and provides examples using voltage regulators. This document also includes information about:

- "Power Sequencing and Hot Socketing" on page 12–1
- "Using MultiVolt I/O Pins" on page 12–2
- "Voltage Regulators" on page 12–3
- "1.5-V Regulator Application Examples" on page 12–19
- "Board Layout" on page 12–21
- "Power Sequencing and Hot Socketing" on page 12–1

Power Sequencing and Hot Socketing

Because 1.5-V Cyclone FPGAs can be used in a mixed-voltage environment, they have been designed specifically to tolerate any possible power-up sequence. Therefore, the V_{CCIO} and V_{CCINT} power supplies may be powered in any order.

You can drive signals into Cyclone FPGAs before and during power up without damaging the device. In addition, Cyclone FPGAs do not drive out during power up since they are tri-stated during power up. Once the device reaches operating conditions and is configured, Cyclone FPGAs operate as specified by the user.



For more information, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

Using MultiVolt I/O Pins

Cyclone FPGAs require a 1.5-V V_{CCINT} and a 3.3-V, 2.5-V, 1.8-V, or 1.5-V I/O supply voltage level (V_{CCIO}). All pins, including dedicated inputs, clock, I/O, and JTAG pins, are 3.3-V tolerant before and after V_{CCINT} and V_{CCIO} are powered.

When V_{CCIO} is connected to 1.5-V, the output is compatible with 1.5-V logic levels. The output pins can be made 1.8-V, 2.5-V, or 3.3-V compatible by using open-drain outputs pulled up with external resistors. You can use external resistors to pull open-drain outputs up with a 1.8-V, 2.5-V, or 3.3-V V_{CCIO}. Table 12–1 summarizes Cyclone MultiVolt I/O support.

Table 12–1. Cyclone MultiVolt I/O Support *Note (1)*

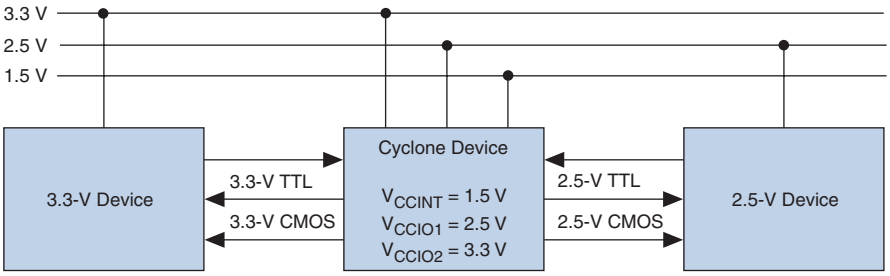
V_{CCIO} (V)	Input Signal					Output Signal				
	1.5-V	1.8-V	2.5-V	3.3-V	5.0-V	1.5-V	1.8-V	2.5-V	3.3-V	5.0-V
1.5-V	✓	✓	✓ (2)	✓ (2)	—	✓	—	—	—	—
1.8-V	✓	✓	✓	✓	—	✓ (3)	✓	—	—	—
2.5-V	—	—	✓	✓	—	✓ (5)	✓ (5)	✓	—	—
3.3-V	—	—	✓ (4)	✓	✓ (6)	✓ (7)	✓ (7)	✓ (7)	✓	✓ (8)

Notes to Table 12–1:

(1) The PCI clamping diode must be disabled to drive an input with voltages higher than V_{CCIO}.
(2) When V_{CCIO} = 1.5-V and a 2.5-V or 3.3-V input signal feeds an input pin, higher pin leakage current is expected.
(3) When V_{CCIO} = 1.8-V, a Cyclone device can drive a 1.5-V device with 1.8-V tolerant inputs.
(4) When V_{CCIO} = 3.3-V and a 2.5-V input signal feeds an input pin, or when V_{CCIO} = 1.8-V and a 1.5-V input signal feeds an input pin, the V_{CCIO} supply current is slightly larger than expected. The reason for this increase is that the input signal level does not drive to the V_{CCIO} rail, which causes the input buffer to not completely shut off.
(5) When V_{CCIO} = 2.5-V, a Cyclone device can drive a 1.5-V or 1.8-V device with 2.5-V tolerant inputs.
(6) Cyclone devices can be 5.0-V tolerant with the use of an external resistor and the internal PCI clamp diode.
(7) When V_{CCIO} = 3.3-V, a Cyclone device can drive a 1.5-V, 1.8-V, or 2.5-V device with 3.3-V tolerant inputs.
(8) When V_{CCIO} = 3.3-V, a Cyclone device can drive a device with 5.0-V LVTTL inputs but not 5.0-V LVCMOS inputs.

Figure 12–1 shows how Cyclone FPGAs interface with 3.3--V and 2.5-V devices while operating with a 1.5-V V_{CCINT} to increase performance and save power.

Figure 12–1. Cyclone FPGAs Interface with 3.3-V and 2.5-V Devices



Voltage Regulators

This section explains how to generate a 1.5-V supply from another system supply. Supplying power to the 1.5-V logic array and/or I/O pins requires a 5.0-V- or 3.3-V-to-1.5-V voltage regulator. A linear regulator is ideal for low-power applications because it minimizes device count and has acceptable efficiency for most applications. A switching voltage regulator provides optimal efficiency. Switching regulators are ideal for high-power applications because of their high efficiency.

This section will help you decide which regulator to use in your system, and how to implement the regulator in your design. There are several companies that provide voltage regulators for low-voltage devices, such as Linear Technology Corporation, Maxim Integrated Products, Intersil Corporation (Elantec), and National Semiconductor Corporation.

Table 12–2 shows the terminology and specifications commonly encountered with voltage regulators. Symbols are shown in parentheses. If the symbols are different for linear and switching regulators, the linear regulator symbol is listed first.

Table 12–2. Voltage Regulator Specifications and Terminology (Part 1 of 2)

Specification/Terminology	Description
Input voltage range (V_{IN},V_{CC})	Minimum and maximum input voltages define the input voltage range, which is determined by the regulator process voltage capabilities.
Line regulation (line regulation, V_{OUT})	Line regulation is the variation of the output voltage (V_{OUT}) with changes in the input voltage (V_{IN}). Error amplifier gain, pass transistor gain, and output impedance all influence line regulation. Higher gain results in better regulation. Board layout and regulator pin-outs are also important because stray resistance can introduce errors.

Table 12–2. Voltage Regulator Specifications and Terminology (Part 2 of 2)

Specification/Terminology	Description
Load regulation (load regulation, V_{OUT})	Load regulation is a variation in the output voltage caused by changes in the input supply current. Linear Technology regulators are designed to minimize load regulation, which is affected by error amplifier gain, pass transistor gain, and output impedance.
Output voltage selection	Output voltage selection is adjustable by resistor voltage divider networks, connected to the error amplifier input, that control the output voltage. There are multiple output regulators that create 5.0-, 3.3-, 2.5-, 1.8- and 1.5-V supplies.
Quiescent current	Quiescent current is the supply current during no-load or quiescent state. This current is sometimes used as a general term for a supply current used by the regulator.
Dropout voltage	Dropout voltage is the difference between the input and output voltages when the input is low enough to cause the output to drop out of regulation. The dropout voltage should be as low as possible for better efficiency.
Current limiting	Voltage regulators are designed to limit the amount of output current in the event of a failing load. A short in the load causes the output current and voltage to decrease. This event cuts power dissipation in the regulator during a short circuit.
Thermal overload protection	This feature limits power dissipation if the regulator overheats. When a specified temperature is reached, the regulator turns off the output drive transistors, allowing the regulator to cool. Normal operation resumes once the regulator reaches a normal operating temperature.
Reverse current protection	If the input power supply fails, large output capacitors can cause a substantial reverse current to flow backward through the regulator, potentially causing damage. To prevent damage, protection diodes in the regulator create a path for the current to flow from V_{OUT} to V_{IN}.
Stability	The dominant pole placed by the output capacitor influences stability. Voltage regulator vendors can assist you in output capacitor selection for regulator designs that differ from what is offered.
Minimum load requirements	A minimum load from the voltage divider network is required for good regulation, which also serves as the ground for the regulator's current path.
Efficiency	Efficiency is the division of the output power by the input power. Each regulator model has a specific efficiency value. The higher the efficiency value, the better the regulator.

Linear Voltage Regulators

Linear voltage regulators generate a regulated output from a larger input voltage using current pass elements in a linear mode. There are two types of linear regulators available: one using a series pass element and another using a shunt element (e.g., a zener diode). Altera recommends using series linear regulators because shunt regulators are less efficient.

Series linear regulators use a series pass element (i.e., a bipolar transistor or MOSFET) controlled by a feedback error amplifier (see Figure 12–2) to regulate the output voltage by comparing the output to a reference voltage. The error amplifier drives the transistor further on or off continuously to control the flow of current needed to sustain a steady voltage level across the load.

Figure 12–2. Series Linear Regulator

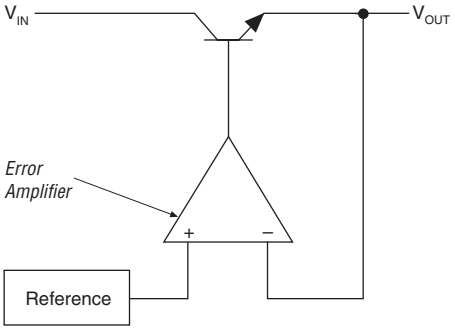


Table 12–3 shows the advantages and disadvantages of linear regulators compared to switching regulators.

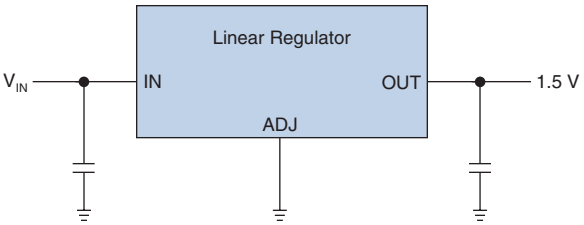
Table 12–3. Linear Regulator Advantages and Disadvantages

Advantages	Disadvantages
Requires few supporting components Low cost Requires less board space Quick transient response Better noise and drift characteristics No electromagnetic interference (EMI) radiation from the switching components Tighter regulation	Less efficient (typically 60%) Higher power dissipation Larger heat sink requirements

You can minimize the difference between the input and output voltages to improve the efficiency of linear regulators. The dropout voltage is the minimum allowable difference between the regulator's input and output voltage.

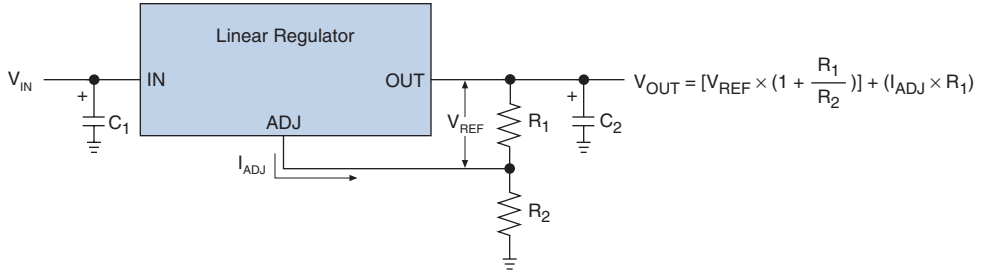
Linear regulators are available with fixed, variable, single, or multiple outputs. Multiple-output regulators can generate multiple outputs (e.g., 1.5- and 3.3-V outputs). If the board only has a 5.0-V power voltage supply, you should use multiple-output regulators. The logic array requires a 1.5-V power supply, and a 3.3-V power supply is required to interface with 3.3- and 5.0-V devices. However, fixed-output regulators have fewer supporting components, reducing board space and cost. Figure 12–3 shows an example of a three-terminal, fixed-output linear regulator.

Figure 12–3. Three-Terminal, Fixed-Output Linear Regulator



Adjustable-output regulators contain a voltage divider network that controls the regulator's output. Figure 12–4 shows how you can also use a three-terminal linear regulator in an adjustable-output configuration.

Figure 12–4. Adjustable-Output Linear Regulator



Switching Voltage Regulators

Step-down switching regulators can provide 3.3-V-to-1.5-V conversion with up to 95% efficiencies. This high efficiency comes from minimizing quiescent current, using a low-resistance power MOSFET switch, and, in higher-current applications, using a synchronous switch to reduce diode losses.

Switching regulators supply power by pulsing the output voltage and current to the load. Table 12–4 shows the advantages and disadvantages of switching regulators compared to linear regulators.

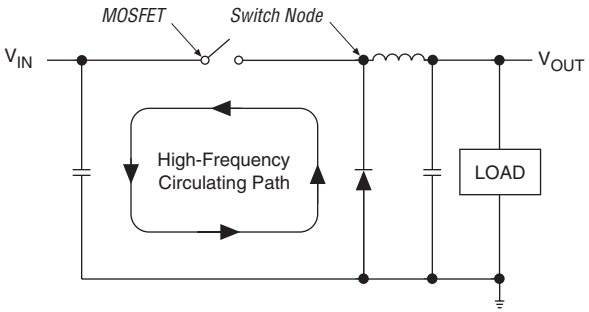
For more information about switching regulators, refer to Linear Technology's application note, *AN35: Step Down Switching Regulators,* at www.linear.com/designtools/app_notes.jsp.

Table 12–4. Switching Regulator Advantages and Disadvantages

Advantages	Disadvantages
Highly efficient (typically >80%) Reduced power dissipation Smaller heat sink requirements Wider input voltage range High power density	Generates EMI Complex to design Requires 15 or more supporting components Higher cost Requires more board space

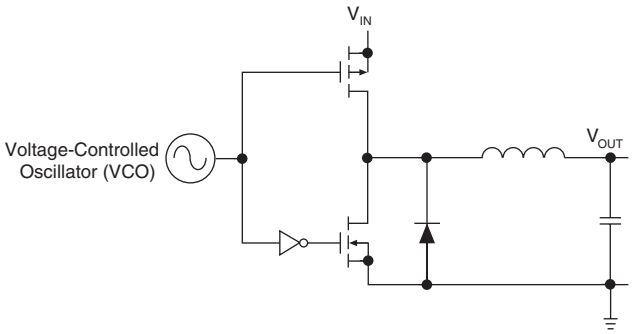
There are two types of switching regulators, asynchronous and synchronous. Asynchronous switching regulators have one field effect transistor (FET) and a diode to provide the current path while the FET is off (see Figure 12–5).

Figure 12–5. Asynchronous Switching Regulator



Synchronous switching regulators have a voltage- or current-controlled oscillator that controls the on and off time of the two MOSFET devices that supply the current to the circuit (see Figure 12–6).

Figure 12–6. Voltage-Controlled Synchronous Switching Regulator



Maximum Output Current

Select an external MOSFET switching transistor (optional) based on the maximum output current that it can supply. Use a MOSFET with a low on-resistance and a voltage rating high enough to avoid avalanche breakdown. For gate-drive voltages less than 9-V, use a logic-level MOSFET. A logic-level MOSFET is only required for topologies with a controller IC and an external MOSFET.

Selecting Voltage Regulators

Your design requirements determine which voltage regulator you need. The key to selecting a voltage regulator is understanding the regulator parameters and how they relate to the design.

The following checklist can help you select the proper regulator for your design:

- Do you require a 3.3-V, 2.5-V, and 1.5-V output (V_{OUT})?
- What precision is required on the regulated 1.5-V supplies (line and load regulation)?
- What supply voltages (V_{IN} or V_{CC}) are available on the board?
- What voltage variance (input voltage range) is expected on V_{IN} or V_{CC}?
- What is the maximum I_{CC} (I_{OUT}) required by your Altera® device?
- What is the maximum current surge ($I_{OUT(MAX)}$) that the regulator will need to supply instantaneously?

Choose a Regulator Type

If required, select either a linear, asynchronous switching, or synchronous switching regulator based on your output current, regulator efficiency, cost, and board-space requirements. DC-to-DC converters have output current capabilities from 1 to 8 A. You can use a controller with an external MOSFET rated for higher current for higher-output-current applications.

Calculate the Maximum Input Current

Use the following equation to estimate the maximum input current based on the output power requirements at the maximum input voltage:

$$I_{IN,DC(MAX)} = \frac{V_{OUT} \times I_{OUT(MAX)}}{\eta \times V_{IN(MAX)}}$$

Where η is nominal efficiency: typically 90% for switching regulators, 60% for linear 2.5-V-to-1.5-V conversion, 45% for linear 3.3-V-to-1.5-V conversion, and 30% for linear 5.0-V-to-1.5-V conversion.

Once you identify the design requirements, select the voltage regulator that is best for your design. Tables 12–5 and 12–6 list a few Linear Technology and Elantec regulators available at the time this document was published. There may be more regulators to choose from depending on your design specification. Contact a regulator manufacturer for availability.

Table 12–5. Linear Technology 1.5-V Output Voltage Regulators

Voltage Regulator	Regulator Type	Total Number of Components	V_{IN} (V)	I_{OUT} (A)	Special Features
LT1573	Linear	10	2.5 or 3.3 *(1)*	6	—
LT1083	Linear	5	5.0	7.5	—
LT1084	Linear	5	5.0	5	—
LT1085	Linear	5	5.0	3	Inexpensive solution
LTC1649	Switching	22	3.3	15	Selectable output
LTC1775	Switching	17	5.0	5	—

Note to Table 12–5:
(1) A 3.3-V V_{IN} requires a 3.3-V supply to the regulator's input and 2.5-V supply to bias the transistors.

Table 12–6. Elantec 1.5-V Output Voltage Regulators

Voltage Regulator	Regulator Type	Total Number of Components	V_{IN} (V)	I_{OUT} (A)	Special Features
EL7551C	Switching	11	5.0	1	—
EL7564CM	Switching	13	5.0	4	—
EL7556BC	Switching	21	5.0	6	—
EL7562CM	Switching	17	3.3 or 5.5	2	—
EL7563CM	Switching	19	3.3	4	—

Voltage Divider Network

Design a voltage divider network if you are using an adjustable output regulator. Follow the controller or converter IC's instructions to adjust the output voltage.

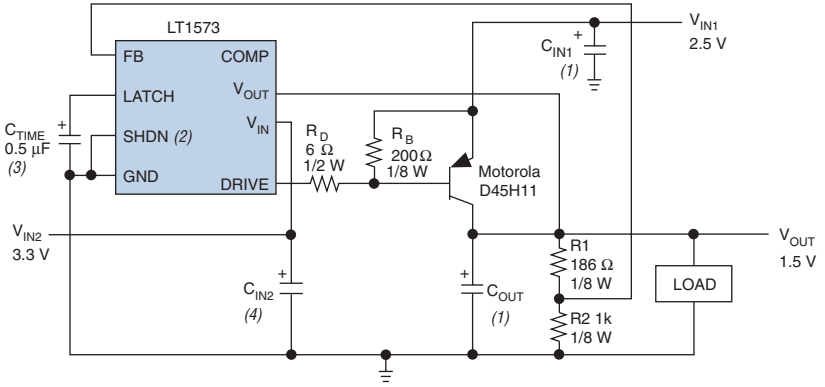
1.5-V Regulator Circuits

This section contains the circuit diagrams for the voltage regulators discussed in this chapter. You can use the voltage regulators in this section to generate a 1.5-V power supply. Refer to the voltage regulator data sheet to find detailed specifications. If you require further information that is not shown in the data sheet, contact the regulator's vendor.

Figures 12–7 through 12–12 show the circuit diagrams of Linear Technology voltage regulators listed in Table 12–5.

The LT1573 linear voltage regulator converts 2.5-V to 1.5-V with an output current of 6A (see Figure 12–7).

Figure 12–7. LT1573: 2.5-V-to-1.5-V/6.0-A Linear Voltage Regulator

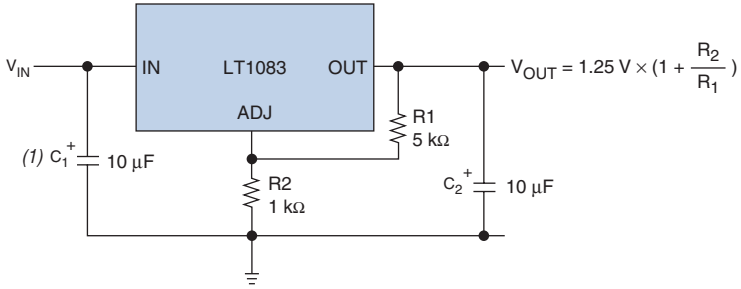


Notes to Figure 12–7:

(1) C_{IN1} and C_{OUT} are AVX 100-µF/10-V surface-mount tantalum capacitors.
(2) Use SHDN (active high) to shut down the regulator.
(3) C_{TIME} is a 0.5-µF capacitor for 100-ms time out at room temperature.
(4) C_{IN2} is an AVX 15-µF/10-V surface-mount tantalum capacitor.

Use adjustable 5.0- to 1.5-V regulators (shown in Figures 12–8 through 12–10) for 3.0- to 7.5-A low-cost, low-device-count, board-space-efficient solutions.

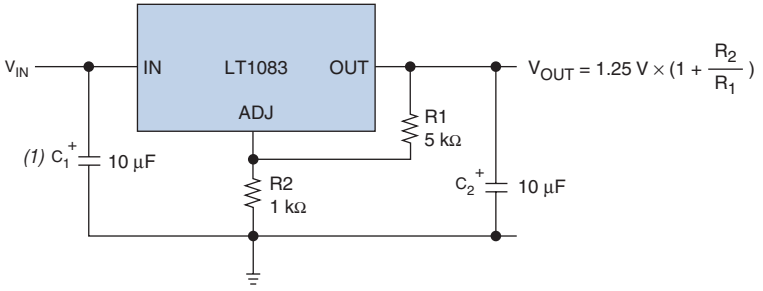
Figure 12–8. LT1083: 5.0-V-to-1.5-V/7.5-A Linear Voltage Regulator



Note to Figure 12–8:

(1) This capacitor is necessary to maintain the voltage level at the input regulator. There could be a voltage drop at the input if the voltage supply is too far away.

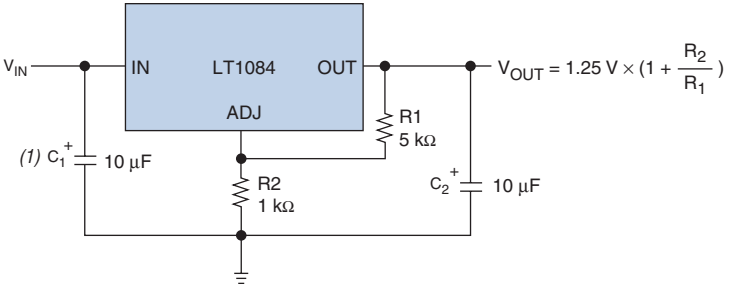
Figure 12–9. LT1084: 5.0-V-to-1.5-V/5.0-A Linear Voltage Regulator



Note to Figure 12–9:

(1) This capacitor is necessary to maintain the voltage level at the input regulator. There could be a voltage drop at the input if the voltage supply is too far away.

Figure 12–10. LT1085: 5.0-V-to-1.5-V/3-A Linear Voltage Regulator

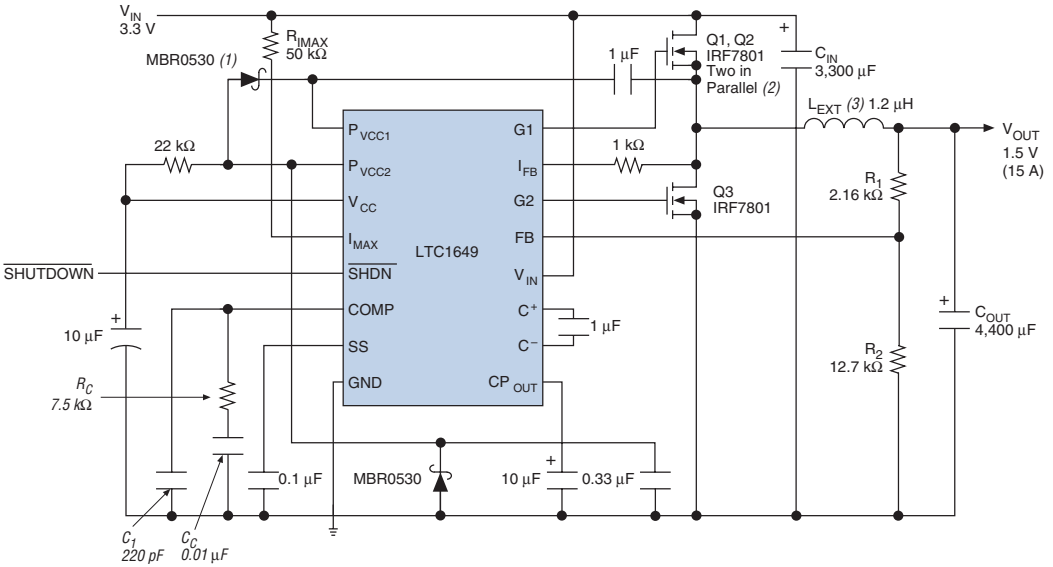


Note to Figure 12–10:

(1) This capacitor is necessary to maintain the voltage level at the input regulator. There could be a voltage drop at the input if the voltage supply is too far away.

Figure 12–11 shows a high-efficiency switching regulator circuit diagram. A selectable resistor network controls the output voltage. The resistor values in Figure 12–11 are selected for 1.5-V output operation.

Figure 12–11. LT1649: 3.3-V-to-1.5-V/15-A Asynchronous Switching Regulator

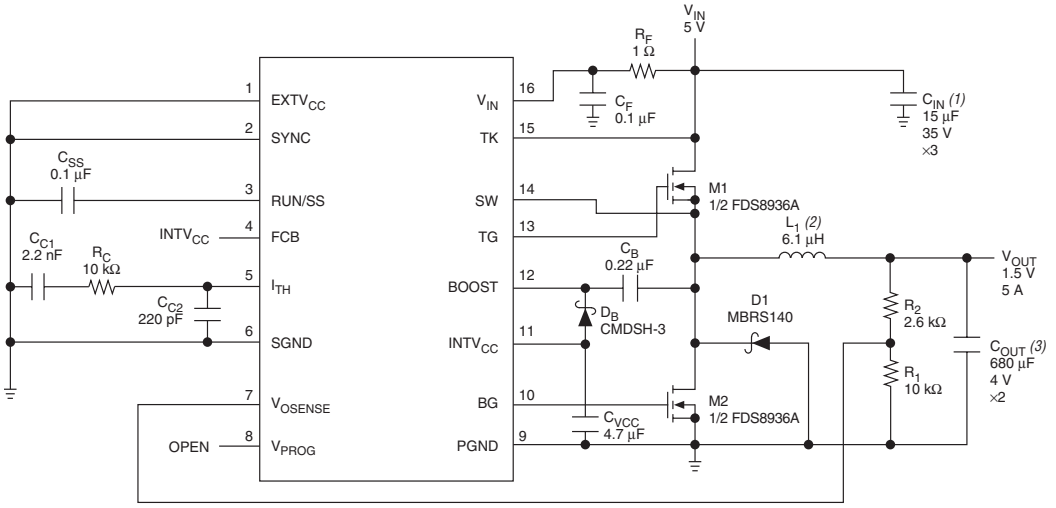


Notes to Figure 12–11:

(1) MBR0530 is a Motorola device.
(2) IRF7801 is a International Rectifier device.
(3) Refer to the Panasonic 12TS-1R2HL device.

Figure 12–12 shows synchronous switching regulator with adjustable outputs.

Figure 12–12. LTC1775: 5.0-V-to-1.5-V/5-A Synchronous Switching Regulator



Notes to Figure 12–12:

(1) This is a KEMETT495X156M035AS capacitor.
(2) This is a Sumida CDRH127-6R1 inductor.
(3) This is a KEMETT510X687K004AS capacitor.

Figures 12–13 through 12–17 show the circuit diagrams of Elantec voltage regulators listed in Table 12–6.

Figures 12–13 through 12–15 show the switching regulator that converts 5.0-V to 1.5-V with different output current.

Figure 12–13. EL7551C: 5.0-V-to-1.5-V/1-A Synchronous Switching Regulator

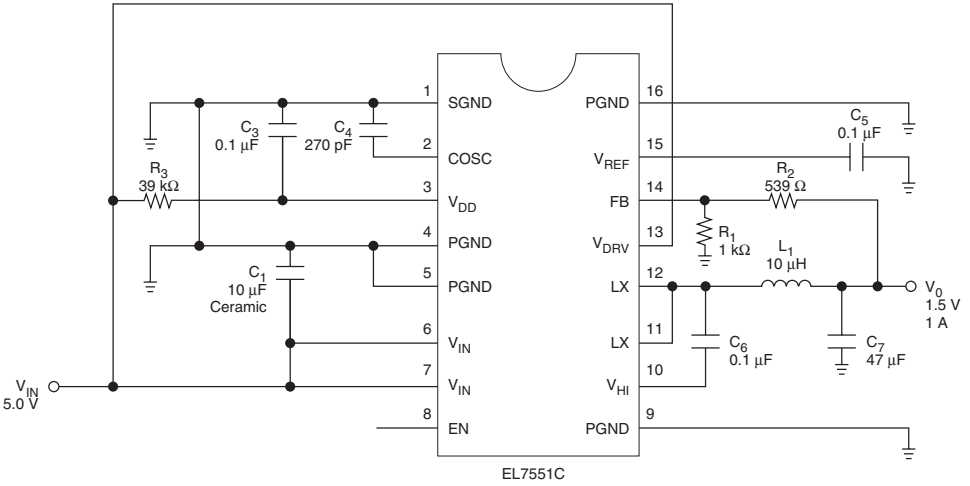


Figure 12–14. EL7564CM: 5.0-V-to-1.5-V/4-A Synchronous Switching Regulator

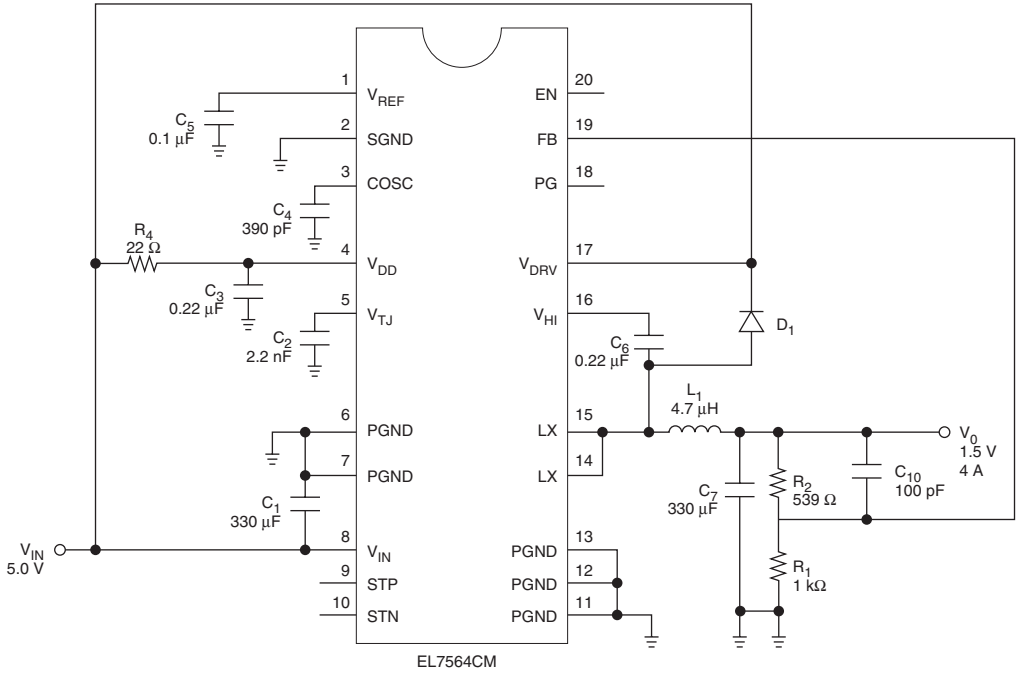
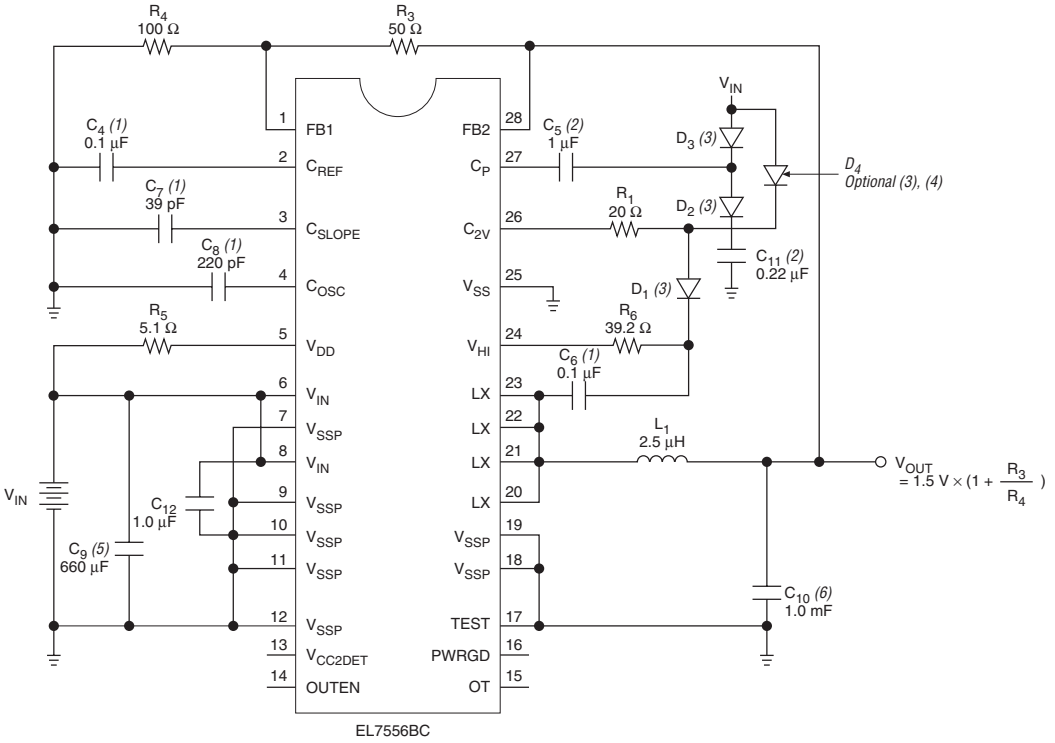


Figure 12–15. EL7556BC: 5.0-V-to-1.5-V/6-A Synchronous Switching Regulator



Notes to Figures 12–13 –12–15:

(1) These capacitors are ceramic capacitors.
(2) These capacitors are ceramic or tantalum capacitor.
(3) These are BAT54S fast diodes.
(4) D4 is only required for EL7556ACM.
(5) This is a Sprague 293D337X96R3 2X330µF capacitor.
(6) This is a Sprague 293D337X96R3 3X330µF capacitor.

Figures 12–16 and 12–17 show the switching regulator that converts 3.3 V to 1.5 V with different output currents.

Figure 12–16. EL7562CM: 3.3-V to 1.5-V/2-A Synchronous Switching Regulator

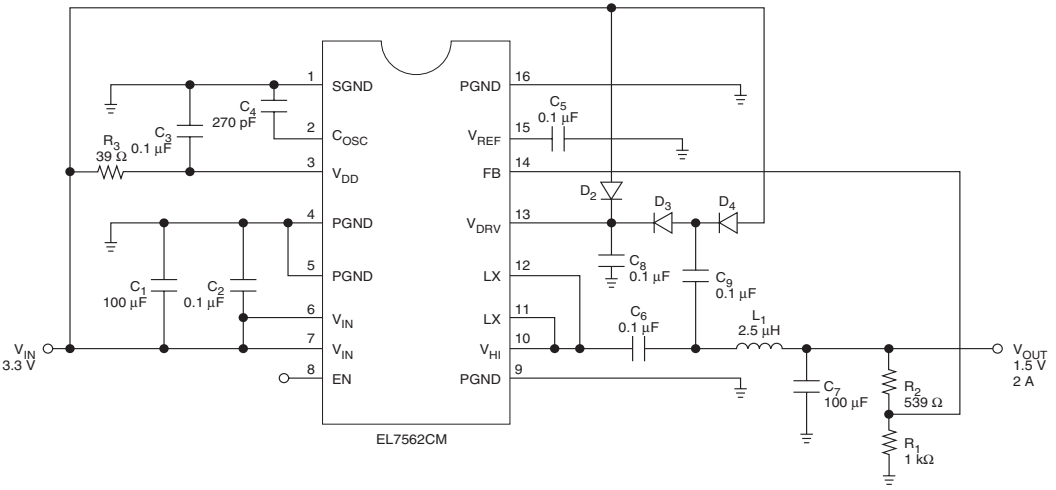
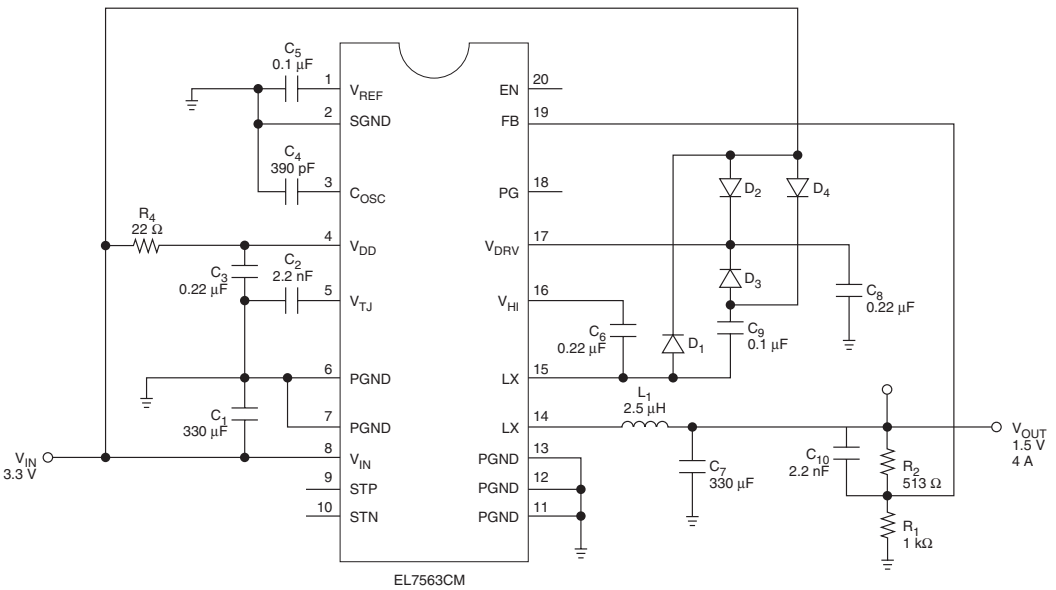


Figure 12–17. EL7563CM: 3.3-V to 1.5-V/4-A Synchronous Switching Regulator



1.5-V Regulator Application Examples

The following sections show the process used to select a voltage regulator for three sample designs. The regulator selection is based on the amount of power that the Cyclone device consumes. There are 14 variables to consider when selecting a voltage regulator. The following variables apply to Cyclone device power consumption:

- f_{MAX}
- Output and bidirectional pins
- Average toggle rate for I/O pins (tog_{IO})
- Average toggle rate for logic elements (LEs) (tog_{LC})
- User-mode I_{CC} consumption
- Maximum power-up I_{CCINT} requirement
- Utilization
- V_{CCIO} supply level
- V_{CCINT} supply level

The following variables apply to the voltage regulator:

- Output voltage precision requirement
- Supply voltage on the board
- Voltage supply output current
- Variance of board supply
- Efficiency

Different designs have different power consumptions based on the variables listed. Once you calculate the Cyclone device's power consumption, you must consider how much current the Cyclone device needs. You can use the Cyclone power calculator (available at www.altera.com) or the PowerGauge™ tool in the Quartus II software to determine the current needs. Also check the maximum power-up current requirement listed in the Power Consumption section of the Cyclone FPGA Family Data Sheet because the power-up current requirement may exceed the user-mode current consumption for a specific design.

Once you determine the minimum current the Cyclone device requires, you must select a voltage regulator that can generate the desired output current with the voltage and current supply that is available on the board using the variables listed in this section. An example is shown to illustrate the voltage regulator selection process.

Synchronous Switching Regulator Example

This example shows a worst-case scenario for power consumption where the design uses all the LEs and RAM. Table 12–7 shows the design requirements for 1.5-V design using a Cyclone EP1C12 FPGA.

Table 12–7. Design Requirements for the Example EP1C12F324C

Design Requirement	Value
Output voltage precision requirement	±5%
Supply voltages available on the board	3.3 V
Voltage supply output current available for this section ($I_{IN,\ DC(MAX)}$)	2 A
Variance of board supply (V_{IN})	±5%
f_{MAX}	150 MHz
Average tog_{IO}	12.5%
Average tog_{LC}	12.5%
Utilization	100%
Output and bidirectional pins	125
V_{CCIO} supply level	3.3 V
V_{CCINT} supply level	1.5 V
Efficiency	≥90%

Table 12–8 uses the checklist on page 12–8 to help select the appropriate voltage regulator.

Table 12–8. Voltage Regulator Selection Process for EP1C12F324C Design (Part 1 of 2)

Output voltage requirements	V_{OUT} = 1.5 V
Supply voltages	V_{IN} OR V_{CC} = 3.3 V
Supply variance from Linear Technology data sheet	Supply variance = ±5%
Estimated I_{CCINT} Use Cyclone Power Calculator	I_{CCINT} = 620 mA
Estimated I_{CCIO} if regulator powers V_{CCIO} Use Cyclone Power Calculator (not applicable in this example because V_{CCIO} = 3.3 V)	I_{CCIO} = N/A
Total user-mode current consumption $I_{CC} = I_{CCINT} + I_{CCIO}$	I_{CC} = 620 mA

Table 12–8. Voltage Regulator Selection Process for EP1C12F324C Design (Part 2 of 2)	
EP1C12 maximum power-up current requirement See Power Consumption section of the Cyclone FPGA Family Data Sheet for other densities	$I_{PUC(MAX)}$ = 900 mA
Maximum output current required Compare I_{CC} with $I_{PUC(MAX)}$	$I_{OUT(MAX)}$ = 900 mA
Voltage regulator selection See *Linear Technology LTC 1649 data sheet* See *Intersil (Elantec) EL7562C data sheet*	LTC1649 $I_{OUT(MAX)}$ = 15 A EL7562C $I_{OUT(MAX)}$ = 2 A
LTC1649	
Nominal efficiency (η)	Nominal efficiency (η) = > 90%
Line and load regulation Line regulation + load regulation = (0.17 mV + 7 mV)/ 1.5 V × 100%	Line and Load Regulation = 0.478% < 5%
Minimum input voltage ($V_{IN(MIN)}$) ($V_{IN(MIN)}$) = $V_{IN}(1 - \Delta V_{IN})$ = 3.3V(1 – 0.05)	($V_{IN(MIN)}$) = 3.135 V
Maximum input current $I_{IN, DC(MAX)} = (V_{OUT} \times I_{OUT(MAX)})/(\eta \times V_{IN(MIN)})$	$I_{IN, DC(MAX)}$ = 478 mA < 2 A
EL7562C	
Nominal efficiency (η)	Nominal efficiency (η) = > 95%
Line and load regulation Line regulation + load regulation = (0.17 mV + 7 mV)/ 1.5 V × 100%	Line and Load Regulation = 0.5% < 5%
Minimum input voltage ($V_{IN(MIN)}$) ($V_{IN(MIN)}$) = $V_{IN}(1 - \Delta V_{IN})$ = 3.3V(1 – 0.05)	($V_{IN(MIN)}$) = 3.135 V
Maximum input current $I_{IN, DC(MAX)} = (V_{OUT} \times I_{OUT(MAX)})/(\eta \times V_{IN(MIN)})$	$I_{IN, DC(MAX)}$ = 453 mA < 2 A

Board Layout

Laying out a printed circuit board (PCB) properly is extremely important in high-frequency (≥100 kHz) switching regulator designs. A poor PCB layout results in increased EMI and ground bounce, which affects the reliability of the voltage regulator by obscuring important voltage and current feedback signals. Altera recommends using Gerber files —pre-designed layout files—supplied by the regulator vendor for your board layout.

If you cannot use the supplied layout files, contact the voltage regulator vendor for help on re-designing the board to fit your design requirements while maintaining the proper functionality.

Altera recommends that you use separate layers for signals, the ground plane, and voltage supply planes. You can support separate layers by using multi-layer PCBs, assuming you are using two signal layers.

Figure 12–18 shows how to use regulators to generate 1.5-V and 2.5-V power supplies if the system needs two power supply systems. One regulator is used for each power supply.

Figure 12–18. Two Regulator Solution for Systems that Require 5.0-V, 2.5-V and 1.5-V Supply Levels

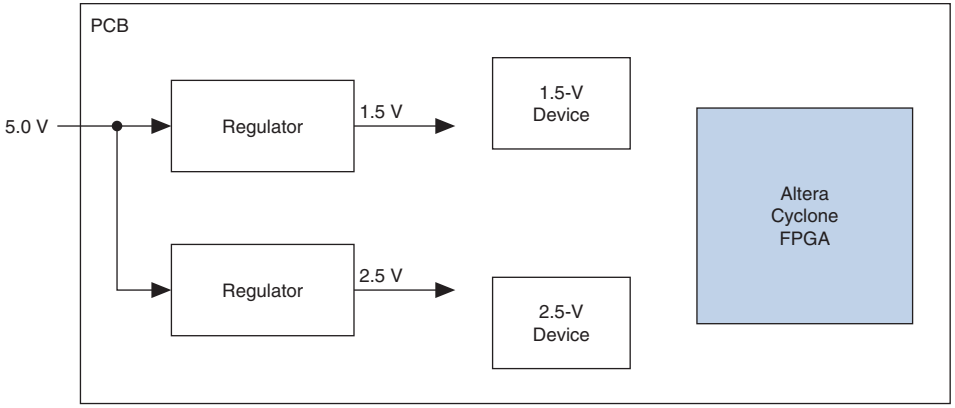
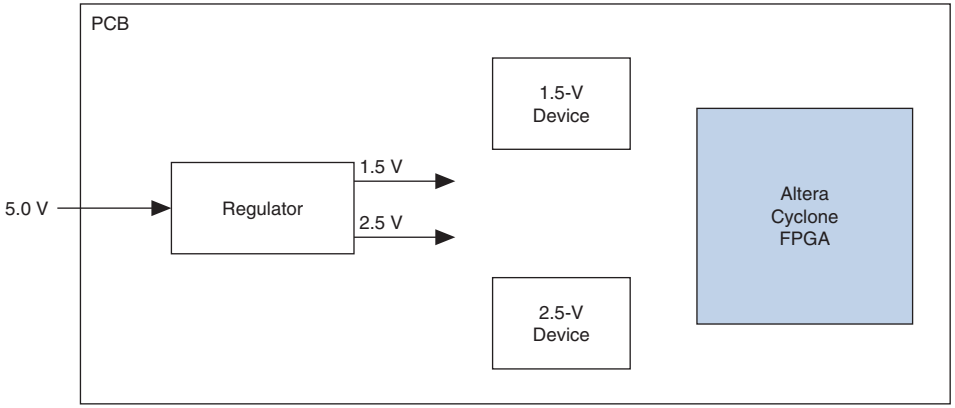


Figure 12–19 shows how to use a single regulator to generate two different power supplies (1.5-V and 2.5-V). The use of a single regulator to generate 1.5-V and 2.5-V supplies from the 5.0-V power supply can minimize the board size and thus save cost.

Figure 12–19. Single Regulator Solution for Systems that Require 5.0-V, 2.5-V and 1.5-V Supply Levels



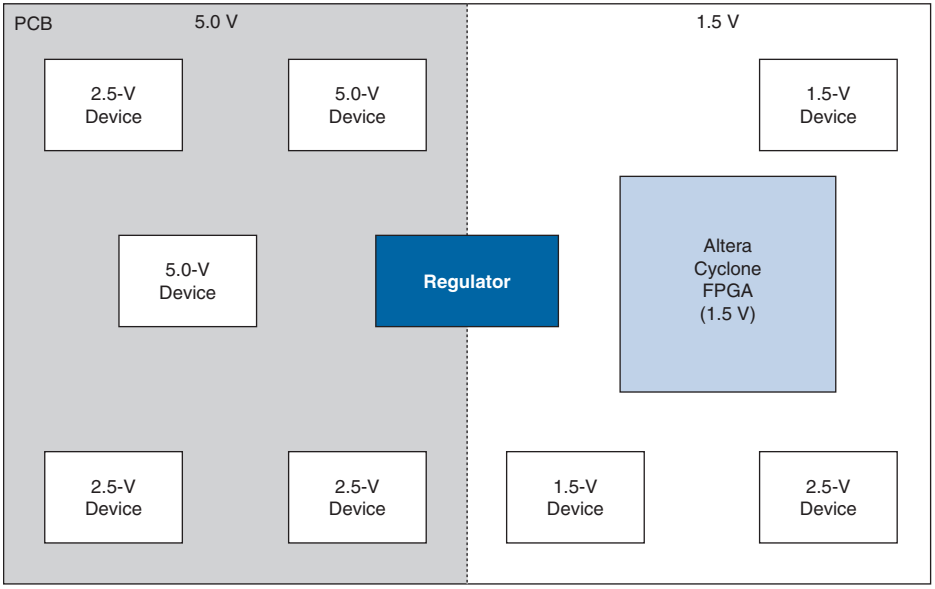
Split-Plane Method

The split-plane design method reduces the number of planes required by placing two power supply planes in one plane (see Figure 12–20). For example, the layout for this method can be structured as follows:

- One 2.5-V plane, covering the entire board
- One plane split between 5.0-V and 1.5-V

This technique assumes that the majority of devices are 2.5-V. To support MultiVolt I/O, Altera devices must have access to 1.5-V and 2.5-V planes.

Figure 12–20. Split Board Layout for 2.5-V Systems With 5.0-V and 1.5-V Devices



Conclusion

With the proliferation of multiple voltage levels in systems, it is important to design a voltage system that can support a low-power device like Cyclone devices. Designers must consider key elements of the PCB, such as power supplies, regulators, power consumption, and board layout when successfully designing a system that incorporates the low-voltage Cyclone family of devices.

References

Linear Technology Corporation. *Application Note 35 (Step-Down Switching Regulators)*. Milpitas: Linear Technology Corporation, 1989.

Linear Technology Corporation. *LT1573 Data Sheet (Low Dropout Regulator Driver)*. Milpitas: Linear Technology Corporation, 1997.

Linear Technology Corporation. *LT1083/LT1084/LT1085 Data Sheet (7.5 A, 5 A, 3 A Low Dropout Positive Adjustable Regulators)*. Milpitas: Linear Technology Corporation, 1994.

Linear Technology Corporation. *LTC1649 Data Sheet (3.3V Input High Power Step-Down Switching Regulator Controller)*. Milpitas: Linear Technology Corporation, 1998.

Linear Technology Corporation. *LTC1775 Data Sheet (High Power No Rsense Current Mode Synchronous Step-Down Switching Regulator)*. Milpitas: Linear Technology Corporation, 1999.

Intersil Corporation. *EL7551C Data Sheet (Monolithic 1 Amp DC:DC Step-Down Regulator)*. Milpitas: Intersil Corporation, 2002.

Intersil Corporation. *EL7564C Data Sheet (Monolithic 4 Amp DC:DC Step-Down Regulator)*. Milpitas: Intersil Corporation, 2002.

Intersil Corporation. *EL7556BC Data Sheet (Integrated Adjustable 6 Amp Synchronous Switcher)*. Milpitas: Intersil Corporation, 2001.

Intersil Corporation. *EL7562C Data Sheet (Monolithic 2 Amp DC:DC Step-Down Regulator)*. Milpitas: Intersil Corporation, 2002.

Intersil Corporation. *EL7563C Data Sheet (Monolithic 4 Amp DC:DC Step-Down Regulator)*. Milpitas: Intersil Corporation, 2002.

Referenced Documents

This chapter references the following document:

- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*

Document Revision History

Table 12–9 shows the revision history for this chapter.

Table 12–9. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.4	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.3	• Added document revision history. • Removed references to Stratix in "Introduction" and "Power Sequencing and Hot Socketing" sections.	—
August 2005 v1.1	Minor updates.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—

Section VI. Configuration

This section provides information for all of the supported configuration schemes for Cyclone devices. The last chapter provides information on EPCS1 and EPCS4 serial configuration devices.

This section contains the following chapters:

- Chapter 13. Configuring Cyclone FPGAs
- Chapter 14. Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet

Revision History

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

13. Configuring Cyclone FPGAs

C51013-1.8

Introduction

You can configure Cyclone® FPGAs using one of several configuration schemes, including the active serial (AS) configuration scheme. This scheme is used with the low cost serial configuration devices. Passive serial (PS) and Joint Test Action Group (JTAG)-based configuration schemes are also supported by Cyclone FPGAs. Additionally, Cyclone FPGAs can receive a compressed configuration bit stream and decompress this data in real-time, reducing storage requirements and configuration time.

This chapter describes how to configure Cyclone devices using each of the three supported configuration schemes.

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

Device Configuration Overview

Cyclone FPGAs use SRAM cells to store configuration data. Since SRAM memory is volatile, configuration data must be downloaded to Cyclone FPGAs each time the device powers up. You can download configuration data to Cyclone FPGAs using the AS, PS, or JTAG interfaces (see Table 13–1).

Table 13–1. Cyclone FPGA Configuration Schemes

Configuration Scheme	Description
Active serial (AS) configuration	Configuration using: • Serial configuration devices (EPCS1, EPCS4, and EPCS16)
Passive serial (PS) configuration	Configuration using: • Enhanced configuration devices (EPC4, EPC8, and EPC16) • EPC2, EPC1 configuration devices • Intelligent host (microprocessor) • Download cable
JTAG-based configuration	Configuration via JTAG pins using: • Download cable • Intelligent host (microprocessor) • Jam™ Standard Test and Programming Language (STAPL) • Ability to use SignalTap® II Embedded Logic Analyzer.

You can select a Cyclone FPGA configuration scheme by driving its MSEL1 and MSEL0 pins either high (1) or low (0), as shown in Table 13–2. If your application only requires a single configuration mode, the MSEL pins can be connected to V_{CC} (the I/O bank's V_{CCIO} voltage where the MSEL pin resides) or to ground. If your application requires more than one configuration mode, the MSEL pins can be switched after the FPGA has been configured successfully. Toggling these pins during user mode does not affect the device operation. However, the MSEL pins must be valid before initiating reconfiguration.

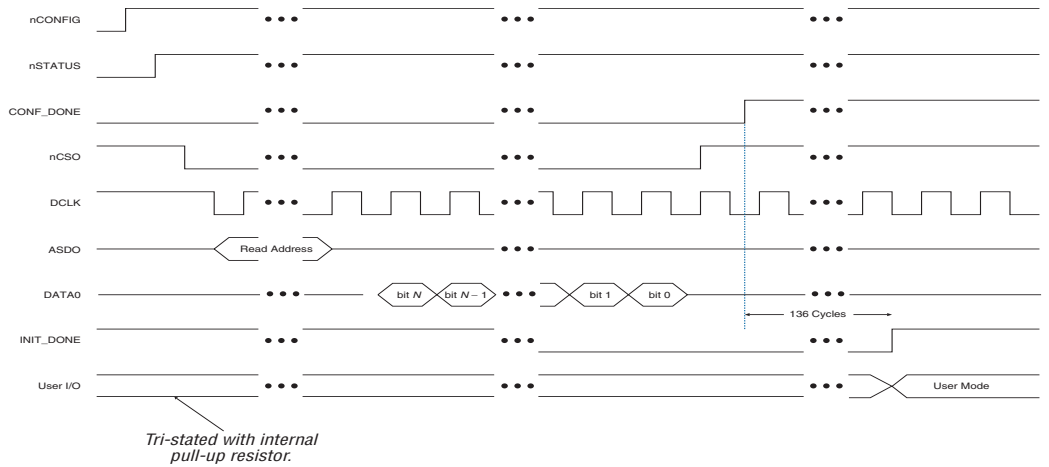
Table 13–2. Selecting Cyclone Configuration Schemes

MSEL1	MSEL0	Configuration Scheme
0	0	AS
0	1	PS
0	1	JTAG-based *(1)*

Note to Table 13–2:
(1) JTAG-based configuration takes precedence over other schemes, which means that MSEL pin settings are ignored.

After configuration, Cyclone FPGAs will initialize registers and I/O pins, then enter user mode and function as per the user design. Figure 13–1 shows an AS configuration waveform.

Figure 13–1. AS Configuration Waveform



You can configure Cyclone FPGAs using the 3.3-, 2.5-, 1.8-, or 1.5-V LVTTL I/O standard on configuration and JTAG input pins. These devices do not feature a VCCSEL pin; therefore, you should connect the VCCIO pins of the I/O banks containing configuration or JTAG pins according to the I/O standard specifications.

Table 13–3 summarizes the approximate uncompressed configuration file size for each Cyclone FPGA. To calculate the amount of storage space required for multi-device configurations, add the file size of each device together.

Table 13–3. Cyclone Raw Binary File (.rbf) Sizes

Device	Data Size (Bits)	Data Size (Bytes)
EP1C3	627,376	78,422
EP1C4	924,512	115,564
EP1C6	1,167,216	145,902
EP1C12	2,323,240	290,405
EP1C20	3,559,608	435,000

You should only use the numbers in Table 13–3 to estimate the configuration file size before design compilation. Different file formats, such as **.hex** or **.ttf** files, have different file sizes. For any specific version of the Quartus® II software, any design targeted for the same device has the same uncompressed configuration file size. If compression is used, the file size can vary after each compilation.

Data Compression

Cyclone FPGAs are the first FPGAs to support decompression of configuration data. This feature allows you to store compressed configuration data in configuration devices or other memory, and transmit this compressed bit stream to Cyclone FPGAs. During configuration, the Cyclone FPGA decompresses the bit stream in real time and programs its SRAM cells.

Cyclone FPGAs support compression in the AS and PS configuration schemes. Compression is not supported for JTAG-based configuration.



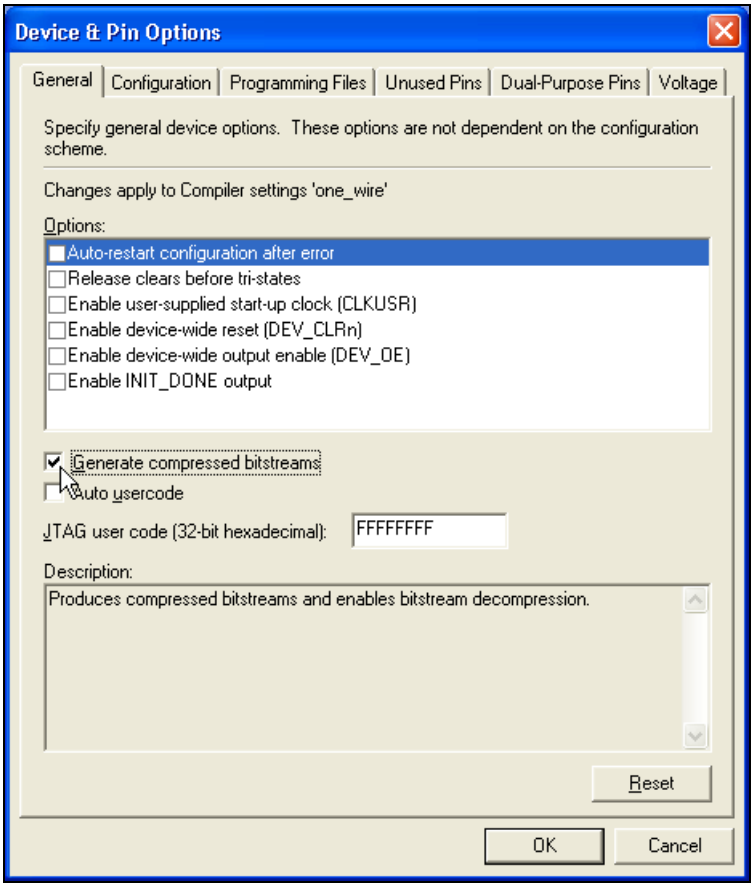
Preliminary data indicates that compression reduces configuration bit stream size by 35 to 60%.

When you enable compression, the Quartus II software generates configuration files with compressed configuration data. This compression reduces the storage requirements in the configuration device or flash, and decreases the time needed to transmit the bit stream to the Cyclone FPGA.

There are two methods to enable compression for Cyclone bitstreams: before design compilation (in the Compiler Settings menu) and after design compilation (in the **Convert Programming Files** window).

To enable compression in the project's compiler settings, select **Device** under the Assignments menu to bring up the settings window. After selecting your Cyclone device open the **Device and Pin Options** window, and in the **General** settings tab enable the check box for **Generate compressed bitstreams** (as shown in Figure 13–2).

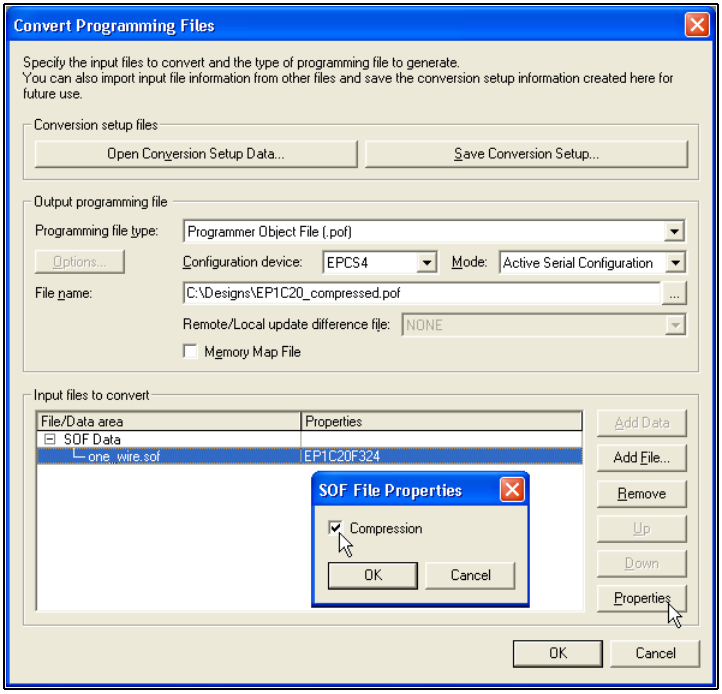
Figure 13–2. Enabling Compression for Cyclone Bitstreams in Compiler Settings



Compression can also be enabled when creating programming files from the **Convert Programming Files** window. See Figure 13–3.

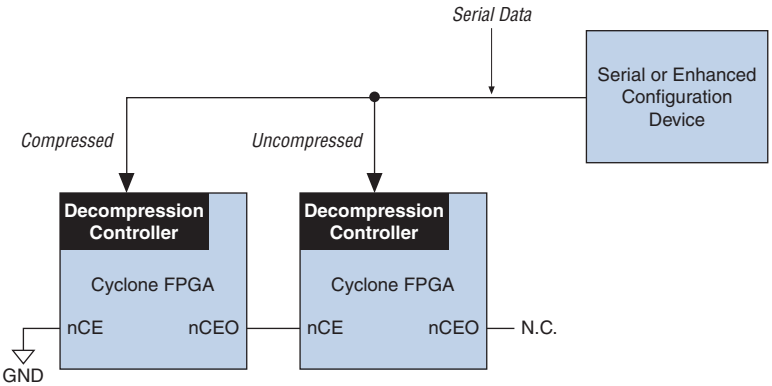
1. Click **Convert Programming Files** (File menu).
2. Select the programming file type (POF, SRAM HEXOUT, RBF, or TTF).
3. For POF output files, select a configuration device.
4. Select **Add File** and add a Cyclone SOF file(s).
5. Select the name of the file you added to the **SOF Data** area and click **Properties**.
6. Turn on **Compression**.

Figure 13–3. Enabling Compression for Cyclone Bitstreams in Convert Programming Files



When multiple Cyclone devices are cascaded, the compression feature can be selectively enabled for each device in the chain. Figure 13–4 depicts a chain of two Cyclone FPGAs. The first Cyclone FPGA has the compression feature enabled and therefore receives a compressed bit stream from the configuration device. The second Cyclone FPGA has the compression feature disabled and receives uncompressed data.

Figure 13–4. Compressed and Uncompressed Configuration Data in the Same Programming File *Note (1)*



Note to Figure 13–4:

(1) The first device in the chain should be set up in AS configuration mode (`MSEL[1..0]="00"`). The remaining devices in the chain must be set up in PS configuration mode (`MSEL[1..0]="01"`).

You can generate programming files for this setup from the **Convert Programming Files** window (File menu) in the Quartus II software.

The decompression feature supported by Cyclone FPGAs is separate from the decompression feature in enhanced configuration devices (EPC16, EPC8, and EPC4 devices). The data compression feature in the enhanced configuration devices allows them to store compressed data and decompress the bit stream before transmitting to the target devices. When using Cyclone FPGAs with enhanced configuration devices, Altera recommends using compression on one of the devices, not both (preferably the Cyclone FPGA since transmitting compressed data reduces configuration time).

Configuration Schemes

This section describes the various configuration schemes you can use to configure Cyclone FPGAs. Descriptions include an overview of the protocol, pin connections, and timing information. The schemes discussed are:

- AS configuration (serial configuration devices)
- PS configuration
- JTAG-based configuration

Active Serial Configuration (Serial Configuration Devices)

In the AS configuration scheme, Cyclone FPGAs are configured using the new serial configuration devices. These configuration devices are low cost devices with non-volatile memory that feature a simple four-pin interface and a small form factor. These features make serial configuration devices an ideal solution for configuring the low-cost Cyclone FPGAs.

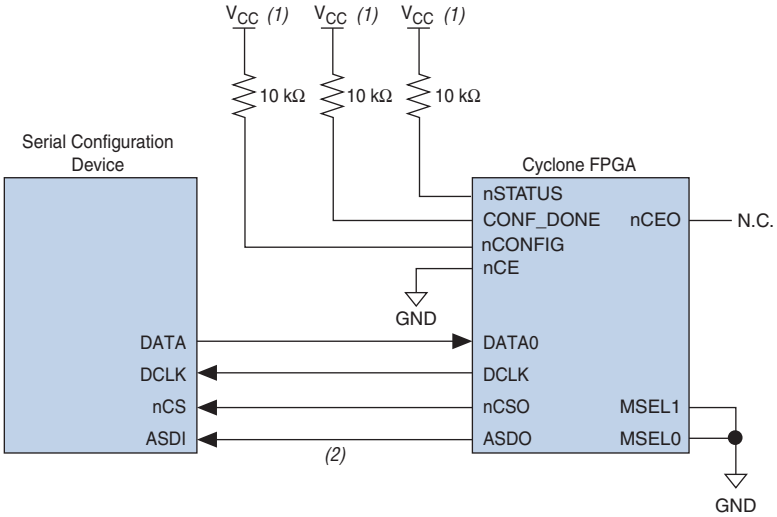


For more information on programming serial configuration devices, refer to the Cyclone Literature web page at www.altera.com and the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet*.

Serial configuration devices provide a serial interface to access configuration data. During device configuration, Cyclone FPGAs read configuration data via the serial interface, decompress data if necessary, and configure their SRAM cells. This scheme is referred to as an AS configuration scheme because the FPGA controls the configuration interface. This scheme is in contrast to the PS configuration scheme where the configuration device controls the interface.

Serial configuration devices have a four-pin interface: serial clock input (`DCLK`), serial data output (`DATA`), AS data input (`ASDI`), and an active-low chip select (`nCS`). This four-pin interface connects to Cyclone FPGA pins as shown in Figure 13–5.

Figure 13–5. AS Configuration of a Single Cyclone FPGA



Notes to Figure 13–5:
(1) Connect the pull-up resistors to a 3.3-V supply.
(2) Cyclone FPGAs use the ASDO to ASDI path to control the configuration device.

Connecting the MSEL[1..0] pins to 00 selects the AS configuration scheme. The Cyclone chip enable signal, nCE, must also be connected to ground or driven low for successful configuration.

During system power up, both the Cyclone FPGA and serial configuration device enter a power-on reset (POR) period. As soon as the Cyclone FPGA enters POR, it drives nSTATUS low to indicate it is busy and drives CONF_DONE low to indicate that it has not been configured. After POR, which typically lasts 100 ms, the Cyclone FPGA releases nSTATUS and enters configuration mode when this signal is pulled high by the external 10-kΩ resistor. Once the FPGA successfully exits POR, all user I/O pins are tri-stated. Cyclone devices have weak pull-up resistors on the user I/O pins which are on before and during configuration.

The value of the weak pull-up resistors on the I/O pins that are on before and during configuration can be found in the *DC and Switching Characteristics* chapter in the *Cyclone Device Handbook*.

The serial clock (DCLK) generated by the Cyclone FPGA controls the entire configuration cycle (see Figure 13–1 on page 13–2) and this clock signal provides the timing for the serial interface. Cyclone FPGAs use an

internal oscillator to generate DCLK. After configuration, this internal oscillator is turned off. Table 13–4 shows the active serial DCLK output frequencies.

Table 13–4. Active Serial DCLK Output Frequency

Minimum	Typical	Maximum	Units
14	17	20	MHz

The serial configuration device latches input/control signals on the rising edge of DCLK and drives out configuration data on the falling edge. Cyclone FPGAs drive out control signals on the falling edge of DCLK and latch configuration data on the falling edge of DCLK.

In configuration mode, the Cyclone FPGA enables the serial configuration device by driving its nCSO output pin low that is connected to the chip select (nCS) pin of the configuration device. The Cyclone FPGA's serial clock (DCLK) and serial data output (ASDO) pins send operation commands and read-address signals to the serial configuration device. The configuration device provides data on its serial data output (DATA) pin that is connected to the DATA0 input on Cyclone FPGAs.

After the Cyclone FPGA receives all configuration bits, it releases the open-drain CONF_DONE pin allowing the external 10-kΩ resistor to pull this signal to a high level. Initialization begins only after the CONF_DONE line reaches a high level. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

You can select the clock used for initialization by using the **User Supplied Start-Up Clock** option in the Quartus II software. The Quartus II software uses the 10-MHz (typical) internal oscillator (separate from the AS internal oscillator) by default to initialize the Cyclone FPGA. After initialization, the internal oscillator is turned off. When you enable the **User Supplied Start-Up Clock** option, the software uses the CLKUSR pin as the initialization clock. Supplying a clock on the CLKUSR pin does not affect the configuration process. After all configuration data is accepted and the CONF_DONE signal goes high, Cyclone devices require 136 clock cycles to initialize properly.

An optional INIT_DONE pin is available. This pin 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. If the INIT_DONE pin is used, it is high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. Once 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 FPGA has entered user mode. In user mode, the user I/O pins do not have weak pull-ups and functions as assigned in your design.

If an error occurs during configuration, the Cyclone FPGA asserts the nSTATUS signal low indicating a data frame error, and the CONF_DONE signal stays low. With the **Auto-Restart Configuration on Frame Error** option enabled in the Quartus II software, the Cyclone FPGA resets the configuration device by pulsing nCSO, releases nSTATUS after a reset time-out period (about 30 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 40 µs to restart configuration. After successful configuration, the CONF_DONE signal is tri-stated by the target device and then pulled high by the pull-up resistor.

All AS configuration pins, DATA0, DCLK, nCSO, and ASDO, have weak internal pull-up resistors. These pull-up resistors are always active.

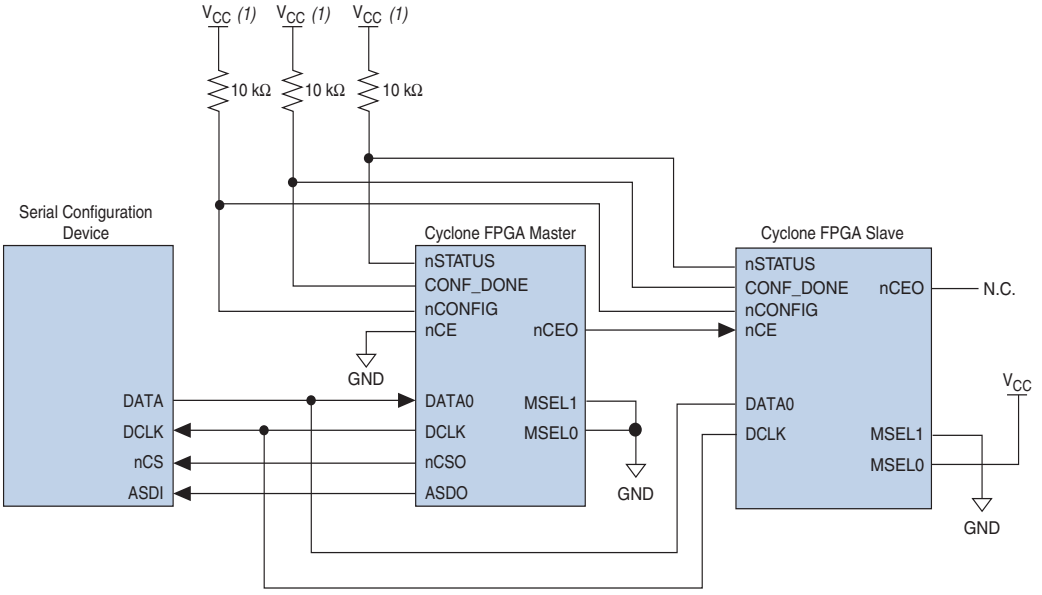
When the Cyclone FPGA is in user mode, you can initiate reconfiguration by pulling the nCONFIG pin low. The nCONFIG pin should be low for at least 40 µs. When nCONFIG is pulled low, the FPGA also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Once nCONFIG returns to a logic high level and nSTATUS is released by the Cyclone FPGA, reconfiguration begins.

Configuring Multiple Devices (Cascading)

You can configure multiple Cyclone FPGAs using a single serial configuration device. You can cascade multiple Cyclone FPGAs using the chip-enable (nCE) and chip-enable-out (nCEO) pins. The first device in the chain must have its nCE pin connected to ground. You must connect its nCEO pin to the nCE pin of the next device in the chain. When the first device captures all of its configuration data from the bit stream, it drives the nCEO pin low enabling the next device in the chain. You must leave the nCEO pin of the last device unconnected. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA0 pins of each device in the chain are connected (see Figure 13–6).

This first Cyclone FPGA in the chain is the configuration master and controls configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Cyclone FPGAs are configuration slaves and you must connect their MSEL pins to select the PS configuration scheme. Figure 13–6 shows the pin connections for this setup.

Figure 13–6. Configuring Multiple Devices Using a Serial Configuration Device (AS)



Note to Figure 13–6:
(1) Connect the pull-up resistors to a 3.3-V supply.

As shown in Figure 13–6, the nSTATUS and CONF_DONE pins on all target FPGAs are connected together with external pull-up resistors. These pins are open-drain bidirectional pins on the FPGAs. When the first device asserts nCEO (after receiving all of its configuration data), it releases its CONF_DONE pin. But the subsequent devices in the chain keep this shared CONF_DONE line low until they have received their configuration data. When all target FPGAs in the chain have received their configuration data and have released CONF_DONE, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode. If an error occurs at any point during configuration, the nSTATUS line is driven low by the failing FPGA. If you enable the **Auto Restart Configuration on Frame Error** option, reconfiguration of the entire chain begins after a reset time-out period (a maximum of 40 µs). If the option is turned off, the external system must monitor nSTATUS for errors and then pulse nCONFIG low to restart configuration. The external system can pulse nCONFIG if it is under system control rather than tied to V_{CC}.



While you can cascade Cyclone FPGAs, serial configuration devices cannot be cascaded or chained together.

If the configuration bit stream size exceeds the capacity of a serial configuration device, you must select a larger configuration device and/or enable the compression feature. While configuring multiple devices, the size of the bit stream is the sum of the individual devices' configuration bit streams.

Configuring Multiple Devices with the Same Data

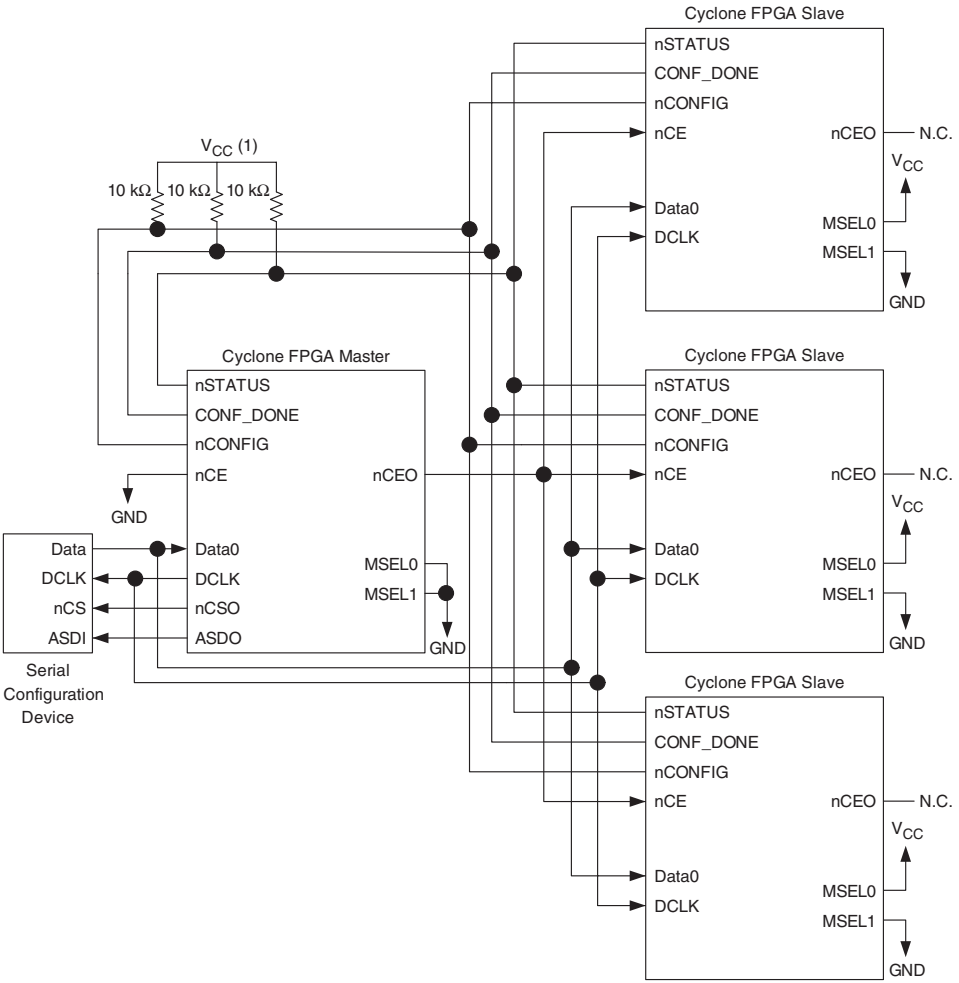
Certain applications require the configuration of multiple Cyclone devices with the same design through a configuration bit stream or SOF file. This can actually be done by two methods and they are shown below. For both methods, the serial configuration devices cannot be cascaded or chained together.

Method 1

For method 1, the serial configuration device stores two copies of the SOF file. The first copy configures the master Cyclone device, and the second copy configures all the remaining slave devices concurrently. The setup is similar to Figure 13–7 where the master is setup in AS mode (MSEL=00) and the slave devices are setup in PS mode (MSEL01).

To configure four identical Cyclone devices with the same SOF file, you could setup the chain similar to the example shown in Figure 13–6, except connect the three slave devices for concurrent configuration. The nCEO pin from the master device drives the nCE input pins on all three slave devices, and the DATA and DCLK pins 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 drives nCE low and transmits the second copy of the configuration data to all three slave devices, configuring them simultaneously. The advantage of using the setup in Figure 13–7 is you can have a different SOF file for the Cyclone master device. However, all the Cyclone slave devices must be configured with the same SOF file.

Figure 13–7. Configuring Multiple Devices with the Same Design Using a Serial Configuration Device



Note to Figure 13–7:

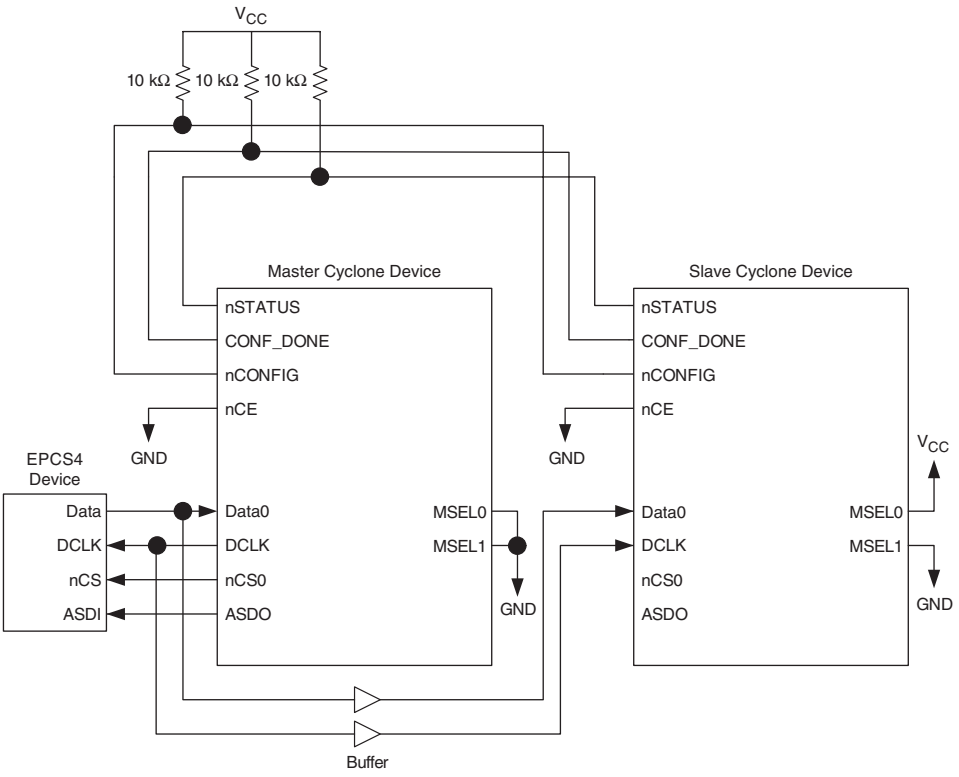
(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

Method 2

Method 2 configures multiple Cyclone devices with the same SOFs by storing only one copy of the SOF in the serial configuration device. This saves memory space in the serial configuration device for general-purpose use and may reduce costs. This method is shown in Figure 13–8 where the master device is set up in AS mode (MSLE=00), and

the slave devices are set up in PS mode (MSEL=01). You could set up one or more slave devices in the chain and all the slave devices are set up in the same way as the design shown in Figure 13–8.

Figure 13–8. Configuring Multiple Devices with the Same Design Using a Serial Configuration Device



In this setup, all the Cyclone devices in the chain are connected for concurrent configuration. This reduces the active serial configuration time because all the Cyclone devices are configured in only one configuration cycle. To achieve this, the nCE input pins on all the Cyclone devices are connected to ground and the nCEO output pins on all the Cyclone devices are left unconnected. The DATA and DCLK pins connect in parallel to all the Cyclone devices.

It is recommended to add a buffer before the DATA and DCLK output from the master Cyclone to avoid signal strength and signal integrity issues. The buffer should not significantly change the DATA-to-DCLK relationships or delay them with respect to other ASMI signals, which are

ASDI and nCS signals. Also, the buffer should only drive the slave Cyclone devices, so that the timing between the master Cyclone device and serial configuration device is unaffected.

This setup can support both compressed and uncompressed SOFs. Therefore, if the configuration bit stream size exceeds the capacity of a serial configuration device, you can enable the compression feature on the SOF used or you can select a larger serial configuration device.

Estimating Active Serial Configuration Time

Active serial configuration time is dominated by the time it takes to transfer data from the serial configuration device to the Cyclone FPGA. This serial interface is clocked by the Cyclone DCLK output (generated from an internal oscillator). As listed in Table 13–4, the DCLK minimum frequency is 14 MHz (71 ns). Therefore, the maximum configuration time estimate for an EP1C3 device (0.628 MBits of uncompressed data) is:

(0.628 MBits × 71 ns) = 47 ms.

The typical configuration time is 33 ms.

Enabling compression reduces the amount of configuration data that is transmitted to the Cyclone device, reducing 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 ByteBlaster™ II download cable. Alternatively, you can program them using the Altera Programming Unit (APU) or supported third-party programmers.

You can perform in-system programming of serial configuration devices via the AS programming interface. During in-system programming, the download cable disables FPGA access to the AS interface by driving the nCE pin high. Cyclone FPGAs are also held in reset by a low level on nCONFIG. After programming is complete, the download cable releases nCE and nCONFIG, allowing the pull-down and pull-up resistor to drive GND and VCC, respectively. Figure 13–9 shows the download cable connections to the serial configuration device.

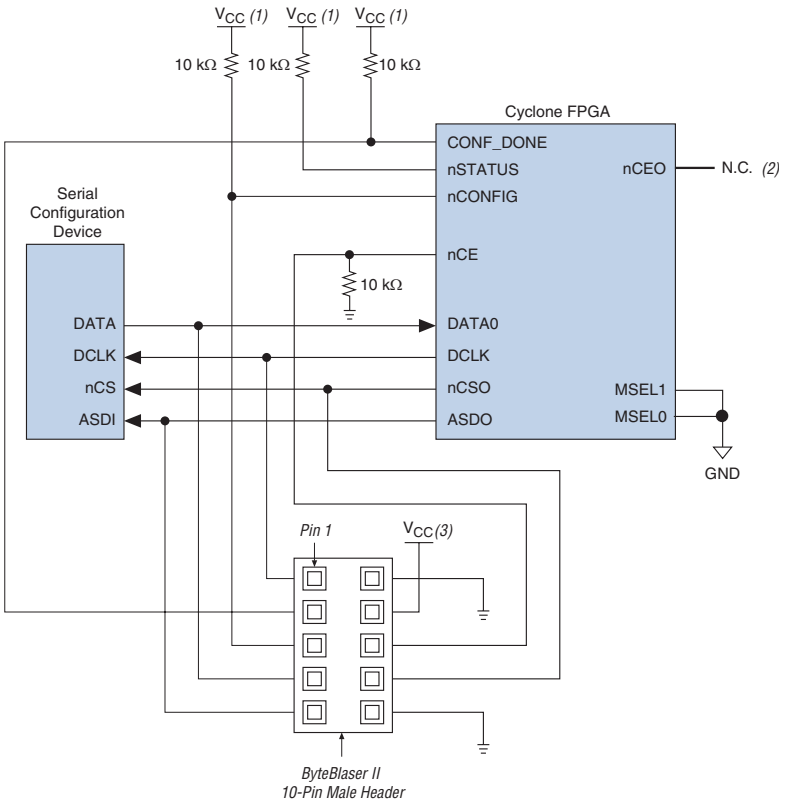


For more information about the ByteBlaster II cable, refer to the *ByteBlaster II Download Cable User Guide*.

The serial configuration devices can be programmed in-system by an external microprocessor using SRunner. SRunner is a software driver developed for embedded serial configuration device programming that can be customized to fit in different embedded systems. The SRunner can read a Raw Programming Data file (.**rpd**) and write to the serial configuration devices. The programming time is comparable to the Quartus II software programming time.

For more information about SRunner, refer tothe *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming* and the source code on the Altera website (www.altera.com).

Figure 13–9. In-System Programming of Serial Configuration Devices



Notes to Figure 13–9:

(1) Connect these pull-up resistors to 3.3-V supply.

(2) The nCEO pin is left unconnected.

(3) Power up the ByteBlaster II cable's V_{CC} with a 3.3-V supply.

You can program serial configuration devices by using the Quartus II software with the APU and the appropriate configuration device programming adapter. All serial configuration devices are offered in an eight-pin small outline integrated circuit (SOIC) package and can be programmed using the PLMSEPC-8 adapter.

In production environments, serial configuration devices can be programmed using multiple methods. Altera programming hardware (APU) or other third-party programming hardware can be 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 using C-based software drivers provided by Altera.



For more information on programming serial configuration devices, refer to the Cyclone Literature web page at www.altera.com and the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet*.

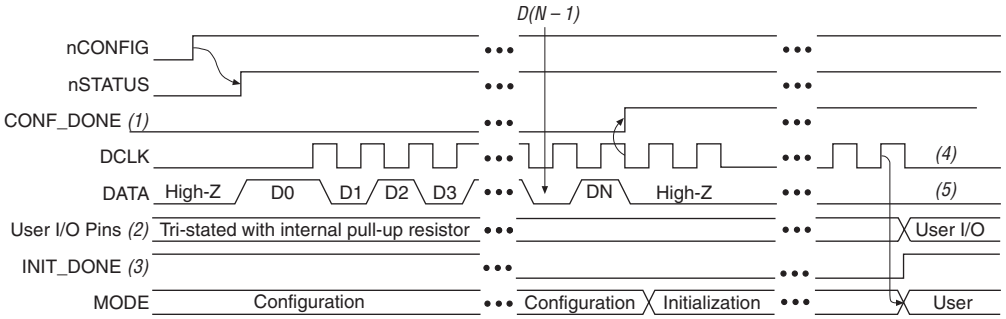


Device configuration options and how to create configuration files are discussed further in the *Software Settings* section in volume 2 of the *Configuration Handbook*.

Passive Serial Configuration

Cyclone FPGAs also feature the PS configuration scheme supported by all Altera FPGAs. In the PS scheme, an external host (configuration device, embedded processor, or host PC) controls configuration. Configuration data is clocked into the target Cyclone FPGAs via the DATA0 pin at each rising edge of DCLK. The configuration waveforms for this scheme are shown in Figure 13–10.

Figure 13–10. PS Configuration Cycle Waveform



Notes to Figure 13–10:

(1) During initial power up and configuration, CONF_DONE is low. After configuration, CONF_DONE goes high to indicate successful configuration. If the device is reconfigured, CONF_DONE goes low after nCONFIG is driven low.

(2) User I/O pins are tri-stated during configuration. Cyclone FPGAs also have a weak pull-up resistor on I/O pins during configuration. After initialization, the user I/O pins perform the function assigned in the user's design.

(3) When used, the optional INIT_DONE signal is high when nCONFIG is low before configuration and during the first 136 clock cycles of configuration.

(4) In user mode, DCLK should be driven high or low when using the PS configuration scheme. When using the AS configuration scheme, DCLK is a Cyclone output pin and should not be driven externally.

(5) In user mode, DATA0 should be driven high or low.

PS Configuration Using Configuration Device

In the PS configuration device scheme, nCONFIG is usually tied to V_{CC} (when using EPC16, EPC8, EPC4, or EPC2 devices, you can connect nCONFIG to nINIT_CONF). Upon device power-up, the target Cyclone FPGA senses the low-to-high transition on nCONFIG and initiates configuration. The target device then drives the open-drain CONF_DONE pin low, which in-turn drives the configuration device's nCS pin low. When exiting POR, both the target and configuration device release the open-drain nSTATUS pin (typically Cyclone POR lasts 100 ms).

Before configuration begins, the configuration device goes through a POR delay of up to 100 ms (maximum) to allow the power supply to stabilize. You must power the Cyclone FPGA before or during the POR time of the enhanced configuration device. During POR, the configuration device drives its OE pin low. This low signal delays configuration because the OE pin is connected to the target device's nSTATUS pin. When the target and configuration devices complete POR, they both release the nSTATUS to OE line, which is then pulled high by a pull-up resistor.

When configuring multiple devices, configuration does not begin until all devices release their OE or nSTATUS pins. When all devices are ready, the configuration device clocks out DATA and DCLK to the target devices using an internal oscillator.

After successful configuration, the Cyclone FPGA starts initialization using the 10-MHz internal oscillator as the reference clock. After initialization, this internal oscillator is turned off. The CONF_DONE pin is released by the target device and then pulled high by a pull-up resistor. When initialization is complete, the target Cyclone FPGA enters user mode. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

If an error occurs during configuration, the target device drives its nSTATUS pin low, resetting itself internally and resetting the configuration device. If you turn on the **Auto-Restart Configuration on Frame Error** option, the device reconfigures automatically if an error occurs. To set this option, select **Compiler Settings** (Processing menu), and click on the **Chips & Devices** tab. Select **Device and Pin Options**, and click on the **Configuration** tab.

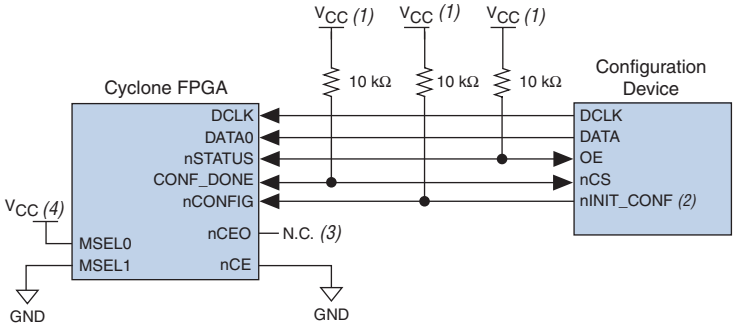
If the **Auto-Restart Configuration on Frame Error** option is turned off, the external system (configuration device or microprocessor) must monitor nSTATUS for errors and then pulse nCONFIG low to restart configuration. The external system can pulse nCONFIG if it is under system control rather than tied to V_{CC}. When configuration is complete, the target device releases CONF_DONE, which disables the configuration device by driving nCS high. The configuration device drives DCLK low before and after configuration.

In addition, if the configuration device sends all of its data and then detects that CONF_DONE has not gone high, it recognizes that the target device has not configured successfully. (For CONF_DONE to reach a high state, enhanced configuration devices wait for 64 DCLK cycles after the last configuration bit. EPC2 devices wait for 16 DCLK cycles.) In this case, the configuration device pulses its OE pin low for a few microseconds, driving the target device's nSTATUS pin low. If the **Auto-Restart Configuration on Frame Error** option is set in the Quartus II software, the target device resets and then releases its nSTATUS pin after a reset time-out period. When nSTATUS returns high, the configuration device reconfigures the target device.

You should not pull CONF_DONE low to delay initialization. Instead, use the Quartus II software's **User-Supplied Start-Up Clock** option to synchronize the initialization of multiple devices that are not in the same configuration chain. Devices in the same configuration chain initialize together since their CONF_DONE pins are tied together.

CONF_DONE goes high during the first few clock cycles of initialization. Hence, when using the CLKUSR feature you would not see the CONF_DONE signal high until you start clocking CLKUSR. However, the device does retain configuration data and waits for these initialization clocks to release CONF_DONE and go into user mode. Figure 13–11 shows how to configure one Cyclone FPGA with one configuration device.

Figure 13–11. Single Device Configuration Circuit



Notes to Figure 13–11:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device. This pull-up resistor is 10 kΩ The EPC16, EPC8, EPC4, and EPC2 devices' OE and nCS pins have internal, user-configurable pull-up resistors. If you use internal pull-up resistors, do not use external pull-up resistors on these pins.

(2) The nINIT_CONF pin is available on EPC16, EPC8, EPC4, and EPC2 devices and has an internal pull-up resistor that is always active. If nINIT_CONF is not used, nCONFIG can be pulled to V_{CC} directly or through a resistor.

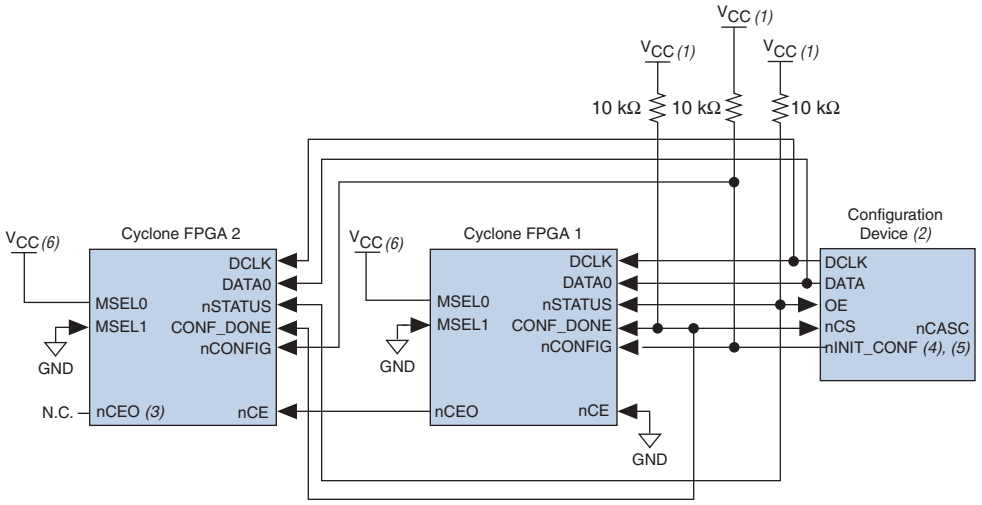
(3) The nCEO pin is left unconnected for the last device in the chain.

(4) Connect MSEL0 to the V_{CC} supply voltage of the I/O bank it resides in.

Configuring Multiple Cyclone FPGAs

You can use a single configuration device to configure multiple Cyclone FPGAs. In this setup, the nCEO pin of the first device is connected to the nCE pin of the second device in the chain. If there are additional devices, connect the nCE pin of the next device to the nCEO pin of the previous device. You should leave the nCEO pin on the last device in the chain unconnected. To configure properly, all of the target device CONF_DONE and nSTATUS pins must be tied together. Figure 13–12 shows an example of configuring multiple Cyclone FPGAs using a single configuration device.

Figure 13–12. Configuring Multiple Cyclone FPGAs with a Single Configuration Device



Notes to Figure 13–12:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device. The EPC16, EPC8, EPC4, and EPC2 devices' OE and nCS pins have internal, user-configurable pull-up resistors. If you use internal pull-up resistors, do not use external pull-up resistors on these pins.

(2) EPC16, EPC8, and EPC4 configuration devices cannot be cascaded.

(3) The nCEO pin is left unconnected for the last device in the chain.

(4) The nINIT_CONF pin is available on EPC16, EPC8, EPC4, and EPC2 devices. If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} directly or through a resistor.

(5) The nINIT_CONF pin has an internal pull-up resistor that is always active in EPC16, EPC8, EPC4, and EPC2 devices. These devices do not need an external pull-up resistor on the nINIT_CONF pin.

(6) Connect MSEL0 to the V_{CC} supply voltage of the I/O bank it resides in.

When performing multi-device PS configuration, you must generate the configuration device programming file (**.sof**) from each project. Then you must combine multiple **.sof** files using the Quartus II software through the **Convert Programming Files** dialog box.

After the first Cyclone FPGA completes configuration during multi-device configuration, its nCEO pin activates the second device's nCE pin, prompting the second device to begin configuration. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

In addition, all nSTATUS pins are tied together; therefore, if any device (including the configuration device) detects an error, configuration stops for the entire chain. Also, if the configuration device does not detect CONF_DONE going high at the end of configuration, it resets the chain by

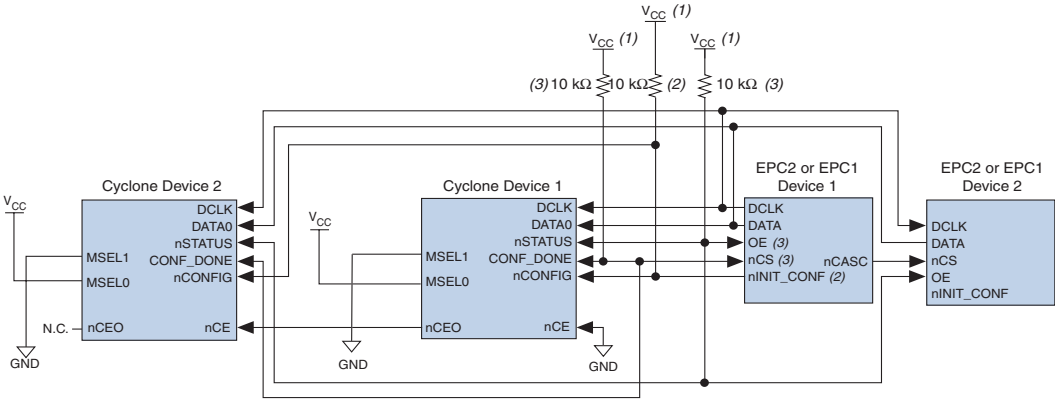
pulsing its OE pin low for a few microseconds. For CONF_DONE to reach a high state, enhanced configuration devices wait for 64 DCLK cycles after the last configuration bit. EPC2 devices wait for 16 DCLK cycles.

If the **Auto-Restart Configuration on Frame Error** option is turned on in the Quartus II software, the Cyclone FPGA releases its nSTATUS pins after a reset time-out period (about 30 µs). When the nSTATUS pins are released and pulled high, the configuration device reconfigures the chain. If the **Auto-Restart Configuration on Frame Error** option is not turned on, the devices drive nSTATUS low until they are reset with a low pulse on nCONFIG.

You can also cascade several EPC2 or EPC1 configuration devices to configure multiple Cyclone FPGAs. When all data from the first configuration device is sent, it drives nCASC low, which in turn drives nCS on the subsequent EPC2 or EPC1 device. Because a configuration device requires less than one clock cycle to activate a subsequent configuration device, the data stream is uninterrupted. You cannot cascade EPC16, EPC8, and EPC4 configuration devices.

Figure 13–13 shows how to configure multiple devices using cascaded EPC2 or EPC1 devices.

Figure 13–13. Multi-Device PS Configuration Using Cascaded EPC2 or EPC1 Devices



Notes to Figure 13–13:

(1) The pull-up resistor should be connected to the same supply voltage as the configuration device.

(2) The nINIT_CONF pin (available on enhanced configuration devices and EPC2 devices only) has an internal pull-up resistor that is always active, meaning an external pull-up resistor should not be used on the nINIT_CONF-nCONFIG line. The nINIT_CONF pin does not need to be connected if its function is not used. If nINIT_CONF is not used or not available (such as on EPC1 devices), nCONFIG must be pulled to V_{CC} either directly or through a resistor.

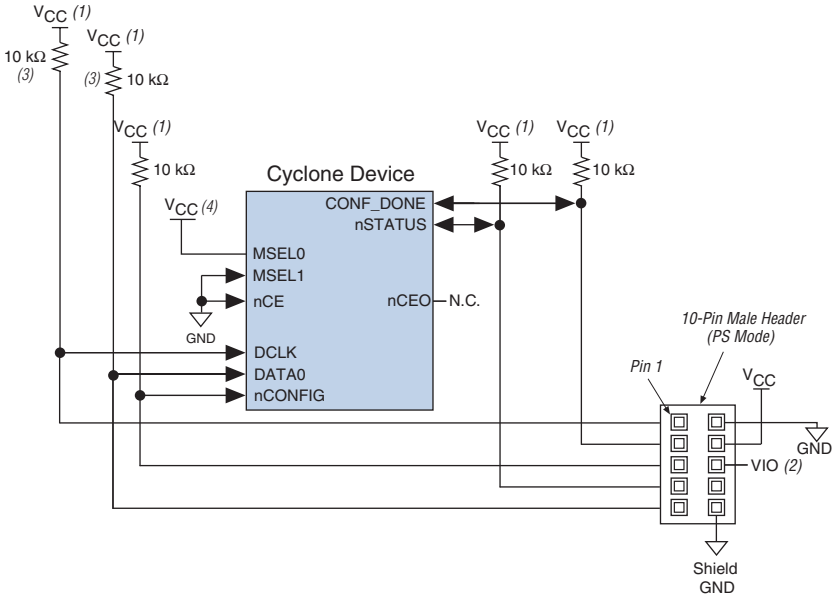
(3) The enhanced configuration devices' and EPC2 devices' OE and nCS pins have internal programmable pull-up resistors. External 10-kΩ pull-up resistors should be used. To turn off the internal pull-up resistors, check the **Disable nCS and OE pull-ups on configuration device** option when generating programming files.

PS Configuration Using a Download Cable

Using a download cable in PS configuration, an intelligent host (for example, your PC) transfers data from a storage device (for example, your hard drive) to the Cyclone FPGA through a USB Blaster, ByteBlaster II, MasterBlaster, or ByteBlasterMV cable. To initiate configuration in this scheme, the download cable generates a low-to-high transition on the nCONFIG pin. The programming hardware then sends the configuration data one bit at a time on the device's DATA0 pin. The data is clocked into the target device using DCLK until the CONF_DONE goes high.

When using programming hardware for the Cyclone FPGA, turning on the **Auto-Restart Configuration on Frame Error** option does not affect the configuration cycle because the Quartus II software must restart configuration when an error occurs. Figure 13–14 shows the PS configuration setup for the Cyclone FPGA using a USB Blaster, ByteBlaster II, MasterBlaster, or ByteBlasterMV cable.

Figure 13–14. PS Configuration Circuit with a Download Cable



Notes to Figure 13–14:

(1) You should connect the pull-up resistor to the same supply voltage as the MasterBlaster (VIO pin) or ByteBlasterMV cable.

(2) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. This pin is a no-connect pin for the ByteBlasterMV header.

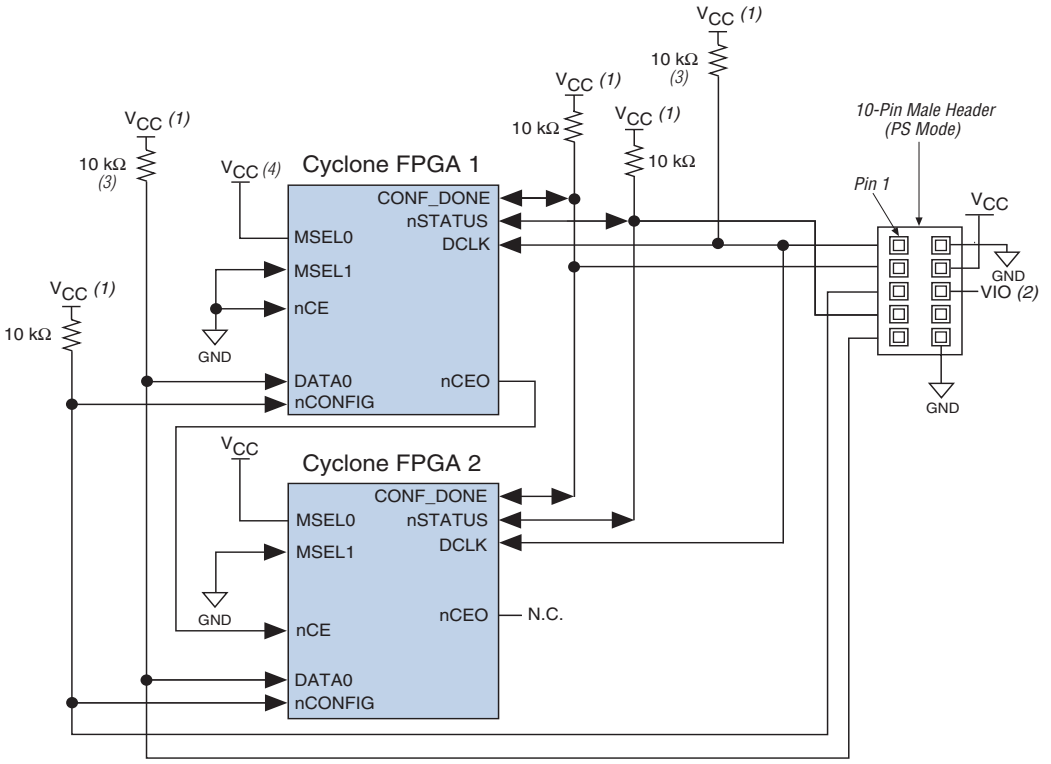
(3) The pull-up resistors on DATA0 and DCLK are only needed if the download cable is the only configuration scheme used on your board. This is to ensure that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, the pull-up resistors on DATA0 and DCLK are not needed.

(4) Connect MSEL0 to the V_{CC} supply voltage of the I/O bank it resides in.

You can use the download cable to configure multiple Cyclone FPGAs by connecting each device's nCEO pin to the subsequent device's nCE pin. All other configuration pins are connected to each 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, because the nSTATUS pins are tied together, the entire chain halts configuration if any device detects an error. In this situation, the Quartus II software must restart configuration; the **Auto-Restart Configuration on Frame Error** option does not affect the configuration cycle. Figure 13–15 shows how to configure multiple Cyclone FPGAs with a ByteBlaster II, MasterBlaster, or ByteBlasterMV cable.

Figure 13–15. Multi-Device PS Configuration with a Download Cable



Notes to Figure 13–15:

(1) You should connect the pull-up resistor to the same supply voltage as the MasterBlaster (VIO pin) or ByteBlasterMV cable.

(2) V_{IO} is a reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value.

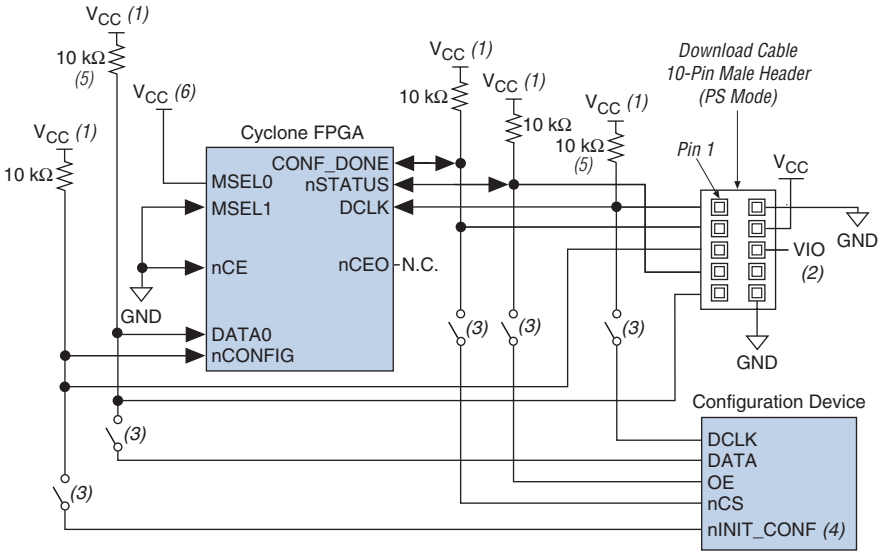
(3) The pull-up resistors on DATA0 and DCLK are only needed if the download cable is the only configuration scheme used on your board. This is to ensure that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, the pull-up resistors on DATA0 and DCLK are not needed.

(4) Connect MSEL0 to the V_{CC} supply voltage of the I/O bank it resides in.

If you are using a ByteBlaster II, MasterBlaster, or ByteBlasterMV cable to configure device(s) on a board that also is populated with configuration devices, you should electrically isolate the configuration devices from the target device(s) and cable. One way to isolate the configuration devices is to add logic, such as a multiplexer, that can select between the configuration devices and the cable. The multiplexer allows bidirectional transfers on the nSTATUS and CONF_DONE signals. Another option is to add switches to the five common signals (CONF_DONE, nSTATUS, DCLK,

nCONFIG, and DATA0) between the cable and the configuration devices. The last option is to remove the configuration devices from the board when configuring with the cable. Figure 13–16 shows a combination of a configuration device and a ByteBlaster II, MasterBlaster, or ByteBlasterMV cable to configure a Cyclone FPGA.

Figure 13–16. Configuring with a Combined PS and Configuration Device Scheme



Notes to Figure 13–16:

(1) You should connect the pull-up resistor to the same supply voltage as the configuration device.

(2) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} should match the target device's V_{CCIO}. This is a no-connect pin for the ByteBlasterMV header.

(3) You should not attempt configuration with a ByteBlaster II, MasterBlaster, or ByteBlasterMV cable while a configuration device is connected to a Cyclone FPGA. Instead, you should either remove the configuration device from its socket when using the download cable or place a switch on the five common signals between the download cable and the configuration device. Remove the ByteBlaster II, MasterBlaster, or ByteBlasterMV cable when configuring with a configuration device.

(4) If nINIT_CONF is not used, nCONFIG must be pulled to V_{CC} either directly or through a resistor.

(5) The pull-up resistors on DATA0 and DCLK are only needed if the download cable is the only configuration scheme used on your board. This is to ensure that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, the pull-up resistors on DATA0 and DCLK are not needed.

(6) Connect MSEL0 to the V_{CC} supply voltage of the I/O bank it resides in.

For more information on how to use the ByteBlaster II, MasterBlaster, or ByteBlasterMV cables, see the following documents:

- *ByteBlaster II Download Cable User Guide*
- *ByteBlasterMV Download Cable User Guide*
- *MasterBlaster Serial/USB Communications Cable User Guide*

PS Configuration from a Microprocessor

In PS configuration with a microprocessor, a microprocessor transfers data from a storage device to the target Cyclone FPGA. To initiate configuration in this scheme, the microprocessor must generate a low-to-high transition on the nCONFIG pin and the target device must release nSTATUS. The microprocessor then places the configuration data one bit at a time on the DATA0 pin of the Cyclone FPGA. The least significant bit (LSB) of each data byte must be presented first. Data is clocked continuously into the target device using DCLK until the CONF_DONE signal goes high.

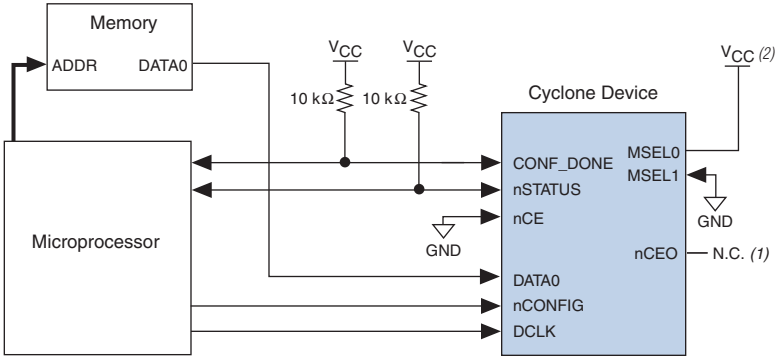
The Cyclone FPGA starts initialization using the internal oscillator after all configuration data is transferred. After initialization, this internal oscillator is turned off. The device's CONF_DONE pin goes high to show successful configuration and the start of initialization. During configuration and initialization and before the device enters user ode the microprocessor must not drive CONF_DONE low. Driving DCLK to the device after configuration does not affect device operation.

Since the PS configuration scheme is a synchronous scheme, the configuration clock speed must be below the specified maximum frequency to ensure successful configuration. Maximum DCLK frequency supported by Cyclone FPGAs is 100 MHz (see Table 13–5 on page 13–30). No maximum DCLK period (i.e., minimum DCLK frequency) exists. You can pause configuration by halting DCLK for an indefinite amount of time.

If the target device detects an error during configuration, it drives its nSTATUS pin low to alert the microprocessor. The microprocessor can then pulse nCONFIG low to restart the configuration process. Alternatively, if the **Auto-Restart Configuration on Frame Error** option is turned on in the Quartus II software, the target device releases nSTATUS after a reset time-out period. After nSTATUS is released, the microprocessor can reconfigure the target device without needing to pulse nCONFIG low.

The microprocessor can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration and initialization. If the microprocessor sends all data, but CONF_DONE and INIT_DONE has not gone high, it must reconfigure the target device. Figure 13–17 shows the circuit for PS configuration with a microprocessor.

Figure 13–17. PS Configuration Circuit with a Microprocessor



Notes to Figure 13–17:
(1) The nCEO pin is left unconnected.
(2) Connect MSEL0 to the V_{CC} supply voltage of the I/O bank it resides in.

Configuring Cyclone FPGAs with the MicroBlaster Software

The MicroBlaster™ software driver allows you to configure Altera FPGAs, including Cyclone FPGAs, through the ByteBlaster II or ByteBlasterMV cable in PS mode. The MicroBlaster software driver supports a Raw Binary File (**.rbf**) programming input file and is targeted for embedded PS configuration. The source code is developed for the Windows NT operating system, although you can customize it to run on other operating systems.

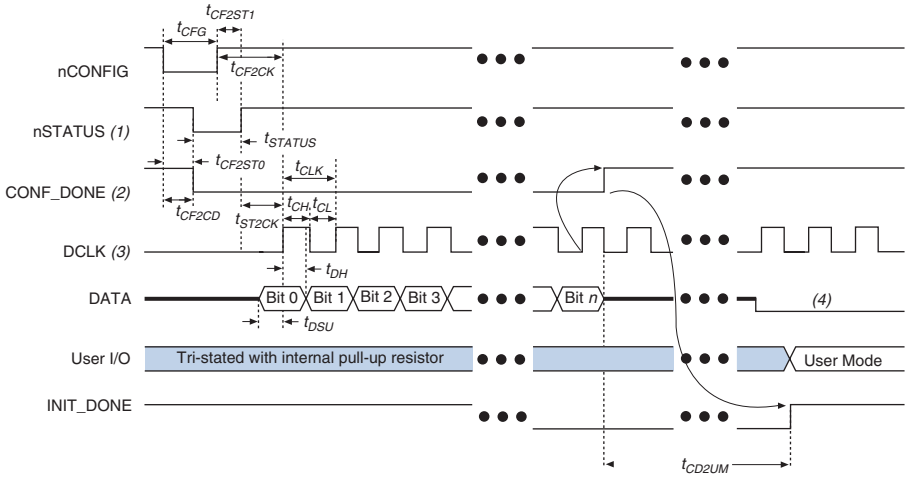
For more information about the MicroBlaster software driver, refer to the *AN 423: Configuring the MicroBlaster Passive Serial Software Driver* and source files on the Altera website at www.altera.com.

Passive Serial Timing

For successful configuration using the PS scheme, several timing parameters such as setup, hold, and maximum clock frequency must be satisfied. The enhanced configuration and EPC2 devices are designed to meet these interface timing specifications. If you use a microprocessor or another intelligent host to control the PS interface, ensure that you meet these timing requirements.

Figure 13–18 shows the PS timing waveform for Cyclone FPGAs.

Figure 13–18. PS Timing Waveform for Cyclone FPGAs



Notes to Figure 13–18:

(1) Upon power-up, the Cyclone FPGA holds nSTATUS low for about 100 ms.
(2) Upon power-up and before configuration, CONF_DONE is low.
(3) In user mode, DCLK should be driven high or low when using the PS configuration scheme. When using the AS configuration scheme, DCLK is a Cyclone output pin and should not be driven externally.
(4) DATA should not be left floating after configuration. It should be driven high or low, whichever is more convenient.

Table 13–5 contains the PS timing information for Cyclone FPGAs.

Table 13–5. PS Timing Parameters for Cyclone Devices *Note (1)* ***(Part 1 of 2)***

Symbol	Parameter	Min	Max	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low		800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low		800	ns
t_{CF2ST1}	nCONFIG high to nSTATUS high		40 *(4)*	µs
t_{CFG}	nCONFIG low pulse width *(2)*	40		µs
t_{STATUS}	nSTATUS low pulse width	10	40 *(4)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	40		µs
t_{ST2CK}	nSTATUS high to first rising edge on DCLK	1		µs
t_{DSU}	Data setup time before rising edge on DCLK	7		ns
t_{DH}	Data hold time after rising edge on DCLK	0		ns
t_{CH}	DCLK high time	7		ns

Table 13–5. PS Timing Parameters for Cyclone Devices Note (1) (Part 2 of 2)

Symbol	Parameter	Min	Max	Units
t_{CL}	DCLK low time	7		ns
t_{CLK}	DCLK period	15		ns
f_{MAX}	DCLK maximum frequency		66	MHz
t_{CD2UM}	CONF_DONE high to user mode *(3)*	6	20	µs

Notes to Table 13–5:

(1) This information is preliminary.
(2) This value applies only if the internal oscillator is selected as the clock source for device initialization. If the clock source is CLKUSR, multiply the clock period by 270 to obtain this value. CLKUSR must be running during this period to reset the device.
(3) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for device initialization. If the clock source is CLKUSR, multiply the clock period by 140 to obtain this value.
(4) You can obtain this value if you do not delay configuration by extending the nSTATUS low-pulse width.



Device configuration options and how to create configuration files are discussed further in the *Software Settings* section in volume 2 of the *Configuration Handbook*.

JTAG-Based Configuration

JTAG has developed a specification for boundary-scan testing. This boundary-scan test (BST) architecture offers the capability to efficiently test components on printed circuit boards (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 Cyclone FPGAs. The Quartus II software automatically generates **.sof** files that can be used for JTAG configuration.



For more information about JTAG boundary-scan testing, refer to *AN 39: IEEE 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*.

To use the SignalTap II Embedded Logic Analyzer, you need to connect the JTAG pins of your Cyclone device to a download cableheader on your PCB.



For more information about SignalTap II, refer to the *Design Debugging Using the SignalTap II Embedded Logic Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

Cyclone devices are designed such that JTAG instructions have precedence over any device operating modes. So JTAG configuration can take place without waiting for other configuration to complete (e.g.,

configuration with serial or enhanced configuration devices). If you attempt JTAG configuration in Cyclone FPGAs during non-JTAG configuration, non-JTAG configuration is terminated and JTAG configuration is initiated.



The Cyclone configuration data decompression feature is not supported in JTAG-based configuration.

A device operating in JTAG mode uses four required pins: TDI, TDO, TMS, and TCK. Cyclone FPGAs do not support the optional TRST pin. The three JTAG input pins, TCK, TDI, and TMS, have weak internal pull-up resistors, whose values are approximately 20 to 40 kΩ All user I/O pins are tri-stated during JTAG configuration.

Table 13–6 shows each JTAG pin's function.

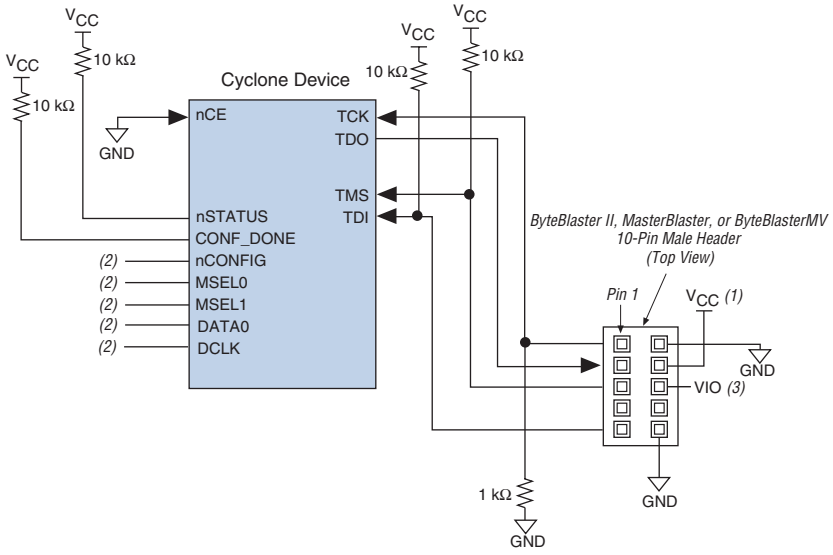
Table 13–6. JTAG Pin Descriptions

Pin	Description	Function
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry can be 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 is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by leaving this pin unconnected.
TMS	Test mode select	Input pin that provides the control signal to determine the transitions of the Test Access Port (TAP) controller state machine. Transitions within the state machine occur on the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to V_{CC}.
TCK	Test clock input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled, by connecting this pin to GND.

JTAG Configuration Using a Download Cable

During JTAG configuration, data is downloaded to the device on the board through a USB Blaster, ByteBlaster II, ByteBlasterMV, or MasterBlaster download cable. Configuring devices through a cable is similar to programming devices in-system. See Figure 13–19 for pin connection information.

Figure 13–19. JTAG Configuration of Single Cyclone FPGA



Notes to Figure 13–19:

(1) You should connect the pull-up resistor to the same supply voltage as the download cable.

(2) You should connect the nCONFIG, MSEL0, and MSEL1 pins to support a non-JTAG configuration scheme. If you only use JTAG configuration, connect nCONFIG and MSEL0 to V_{CC}, and MSEL1 to ground. Pull DATA0 and DCLK to high or low.

(3) V_{IO} is a reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value. In the ByteBlaster MV, this pin is a no connect. In the USB Blaster and ByteBlaster II, this pin is connected to nCE when it is used for Active Serial programming; otherwise it is a no connect.

(4) nCE must be connected to GND or driven low for successful configuration.

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. The software checks the state of CONF_DONE through the JTAG port. 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 bit stream is transmitted serially via the JTAG TDI port, the TCK port is clocked an additional 134 cycles to perform device initialization.



If V_{CCIO} is tied to 3.3-V, both the I/O pins and the JTAG TDO port drive at 3.3-V levels.

Cyclone FPGAs have dedicated JTAG pins. Not only can you perform JTAG testing on Cyclone FPGAs before and after, but also during configuration. While other device families do not support JTAG testing during configuration, Cyclone FPGAs support the BYPASS, IDCODE, and SAMPLE instructions during configuration without interrupting configuration. All other JTAG instructions may only be issued by first interrupting configuration and reprogramming I/O pins using the CONFIG_IO instruction.

The CONFIG_IO instruction allows I/O buffers to be configured via the JTAG port, and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Cyclone FPGA or waiting for a configuration device to complete configuration. Once configuration has been interrupted and JTAG testing is complete, the part must be reconfigured via JTAG (PULSE_CONFIG instruction) or by pulsing nCONFIG low.

The chip-wide reset and output enable pins on Cyclone FPGAs 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 of Cyclone FPGAs, you should consider the dedicated configuration pins. Table 13–7 shows how you should connect these pins during JTAG configuration.

Table 13–7. Dedicated Configuration Pin Connections During JTAG Configuration

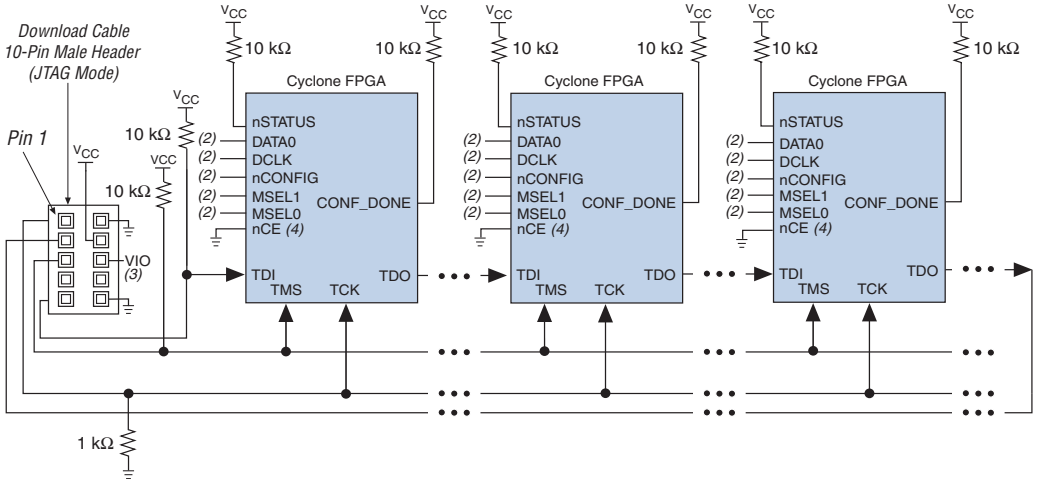
Signal	Description
nCE	Drive all Cyclone devices in the chain low by connecting nCE to ground, pulling it down via a resistor, or driving it low by some control circuitry. For devices in a multi-device PS and AS configuration chains, connect the nCE pins to ground during JTAG configuration or configure them via JTAG in the same order as the configuration chain.
nCEO	For all Cyclone devices in a chain, the nCEO pin can be left floating or connected to the nCE pin of the next device. See nCE description above.
nSTATUS	Pulled to V_{CC} through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, pull up each nSTATUS pin to V_{CC} individually.
CONF_DONE	Pulled to V_{CC} through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, pull up each CONF_DONE pin to V_{CC} individually. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.
nCONFIG	Driven high by connecting to V_{CC}, pulling up through a resistor, or driving it high by some control circuitry.
MSEL0, MSEL1	Do not leave these pins floating. These pins support whichever non-JTAG configuration is used in production. If only JTAG configuration is used, you should tie these pins to ground.
DCLK	Do not leave these pins floating. Drive low or high, whichever is more convenient.
DATA0	Do not leave these pins floating. Drive low or high, whichever is more convenient.

JTAG Configuration of Multiple Devices

When programming a JTAG device chain, one JTAG-compatible header, such as the ByteBlaster II header, is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capacity of the download cable. However, 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 configuration is ideal when the system contains multiple devices, or when testing your system using JTAG BST circuitry. Figure 13–20 shows multi-device JTAG configuration.

Figure 13–20. Multi-Device JTAG Configuration *Note (1)*



Notes to Figure 13–20:

(1) Cyclone, Stratix, Stratix GX, APEX™ II, APEX 20K, Mercury™, ACEX® 1K, and FLEX® 10K devices can be placed within the same JTAG chain for device programming and configuration.

(2) Connect the nCONFIG, MSEL0, and MSEL1 pins to support a non-JTAG configuration scheme. If only JTAG configuration is used, connect nCONFIG to V_{CC}, and MSEL0 and MSEL1 to ground. Pull DATA0 and DCLK to either high or low.

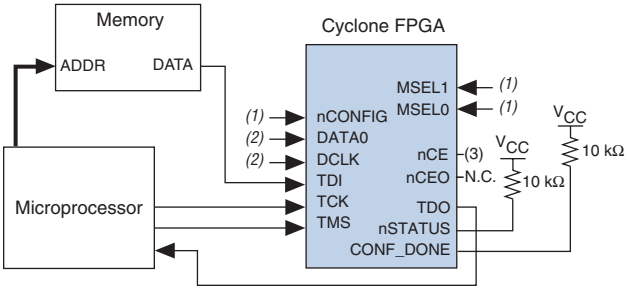
(3) V_{IO} is a reference voltage for the MasterBlaster output driver. V_{IO} should match the device's V_{CCIO}. Refer to the *MasterBlaster Serial/USB Communications Cable User Guide* for this value. In the ByteBlaster MV, this pin is a no connect. In the USB Blaster and ByteBlaster, this pin is connected to nCE when it is used for Active Serial programming; otherwise it is a no connect.

(4) nCE must be connected to GND or driven low for successful configuration.

Connect the nCE pin to ground or drive it low during JTAG configuration. In multi-device PS and AS configuration chains, connect the first device's nCE pin to ground and connect the nCEO pin to the nCE pin of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, it's nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, you should make sure the nCE pins are connected to ground during JTAG configuration or that the devices are configured via JTAG in the same order as the configuration chain. As long as the devices are 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 configured.

Figure 13–21 shows the JTAG configuration of a Cyclone FPGA with a microprocessor.

Figure 13–21. JTAG Configuration of Cyclone FPGAs with a Microprocessor



Notes to Figure 13–21:

(1) Connect the nCONFIG, MSEL1, and MSEL0 pins to support a non-JTAG configuration scheme. If your design only uses JTAG configuration, connect the nCONFIG pin to V_{CC} and the MSEL1 and MSEL0 pins to ground.

(2) Pull DATA0 and DCLK to either high or low.

(3) nCE must be connected to GND or driver low for succesful JTAG configuration.

For more information about JTAG programming in an embedded environment, refer to *AN 122: Using JamSTAPL for ISP &ICR via an Embedded Processor.*

Configuring Cyclone FPGAs with JRunner

JRunner is a software driver that allows you to configure Altera FPGAs, including Cyclone FPGAs, through the ByteBlaster II or ByteBlasterMV cables in JTAG mode. The programming input file supported is in **.rbf** format. JRunner also requires a Chain Description File (**.cdf**) generated by the Quartus II software. JRunner is targeted for embedded JTAG configuration. The source code has been developed for the Windows NT operating system (OS). You can customize the code to make it run on other platforms. For more information on the JRunner software driver, see JRunner Software Driver: An Embedded Solution to the JTAG Configuration and the source files on the Altera website.

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.

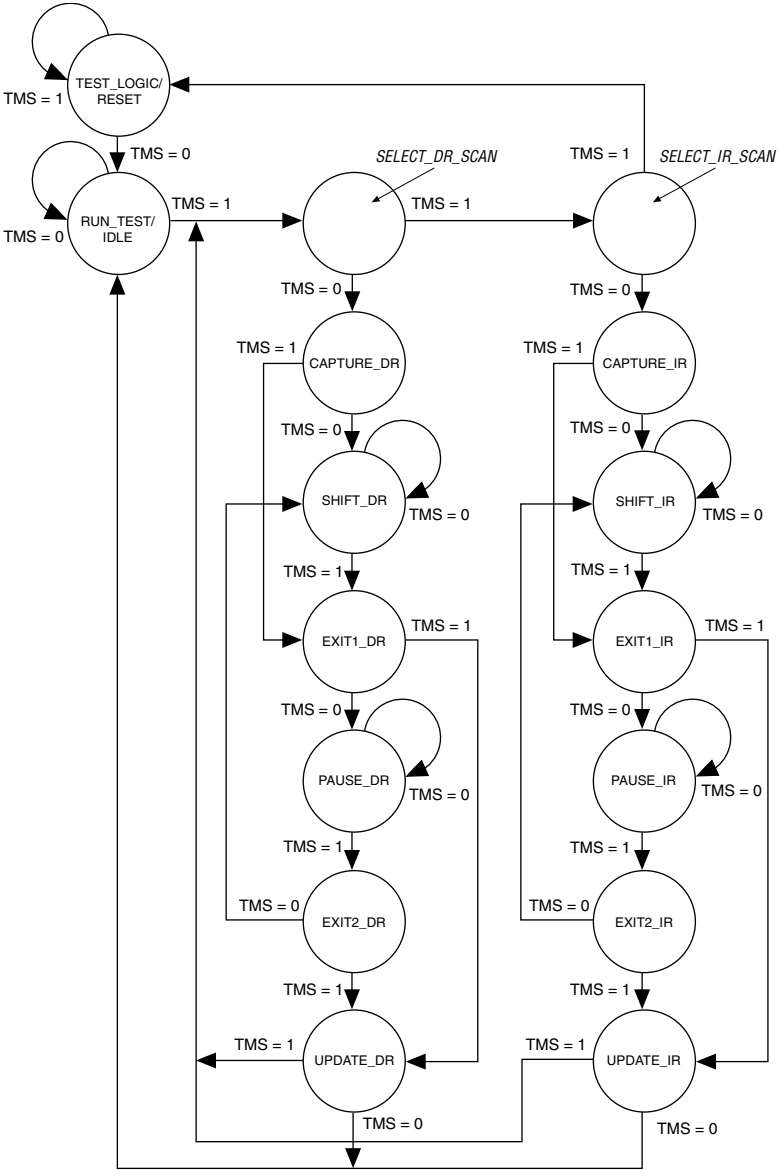


Both JTAG connection methods should include space for the MasterBlaster or ByteBlasterMV header connection. The header is useful during prototyping because it allows you to verify or modify the Cyclone FPGA's contents. During production, you can remove the header to save cost.

Program Flow

The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine. The TAP controller is a 16-state, state machine that is clocked on the rising edge of `TCK`, and uses the `TMS` pin to control JTAG operation in a device. Figure 13–22 shows the flow of an IEEE Std. 1149.1 TAP controller state machine.

Figure 13–22. JTAG TAP Controller State Machine



While the Jam Player provides a driver that manipulates the TAP controller, the Jam Byte-Code File (**.jbc**) provides the high-level intelligence needed to program a given device. All Jam instructions that

send JTAG data to the device involve moving the TAP controller through either the data register leg or the instruction register leg of the state machine. For example, loading a JTAG instruction involves moving the TAP controller to the SHIFT_IR state and shifting the instruction into the instruction register through the TDI pin. Next, the TAP controller is moved to the RUN_TEST/IDLE state where a delay is implemented to allow the instruction time to be latched. This process is identical for data register scans, except that the data register leg of the state machine is traversed.

The high-level Jam instructions are the DRSCAN instruction for scanning the JTAG data register, the IRSCAN instruction for scanning the instruction register, and the WAIT command that causes the state machine to sit idle for a specified period of time. Each leg of the TAP controller is scanned repeatedly, according to instructions in the **.jbc** file, until all of the target devices are programmed.

Figure 13–23 shows the functional behavior of the Jam Player when it parses the **.jbc** file. When the Jam Player encounters a DRSCAN, IRSCAN, or WAIT instruction, it generates the proper data on TCK, TMS, and TDI to complete the instruction. The flow diagram shows branches for the DRSCAN, IRSCAN, and WAIT instructions. Although the Jam Player supports other instructions, they are omitted from the flow diagram for simplicity.

Figure 13–23. Jam Player Flow Diagram (Part 1 of 2)

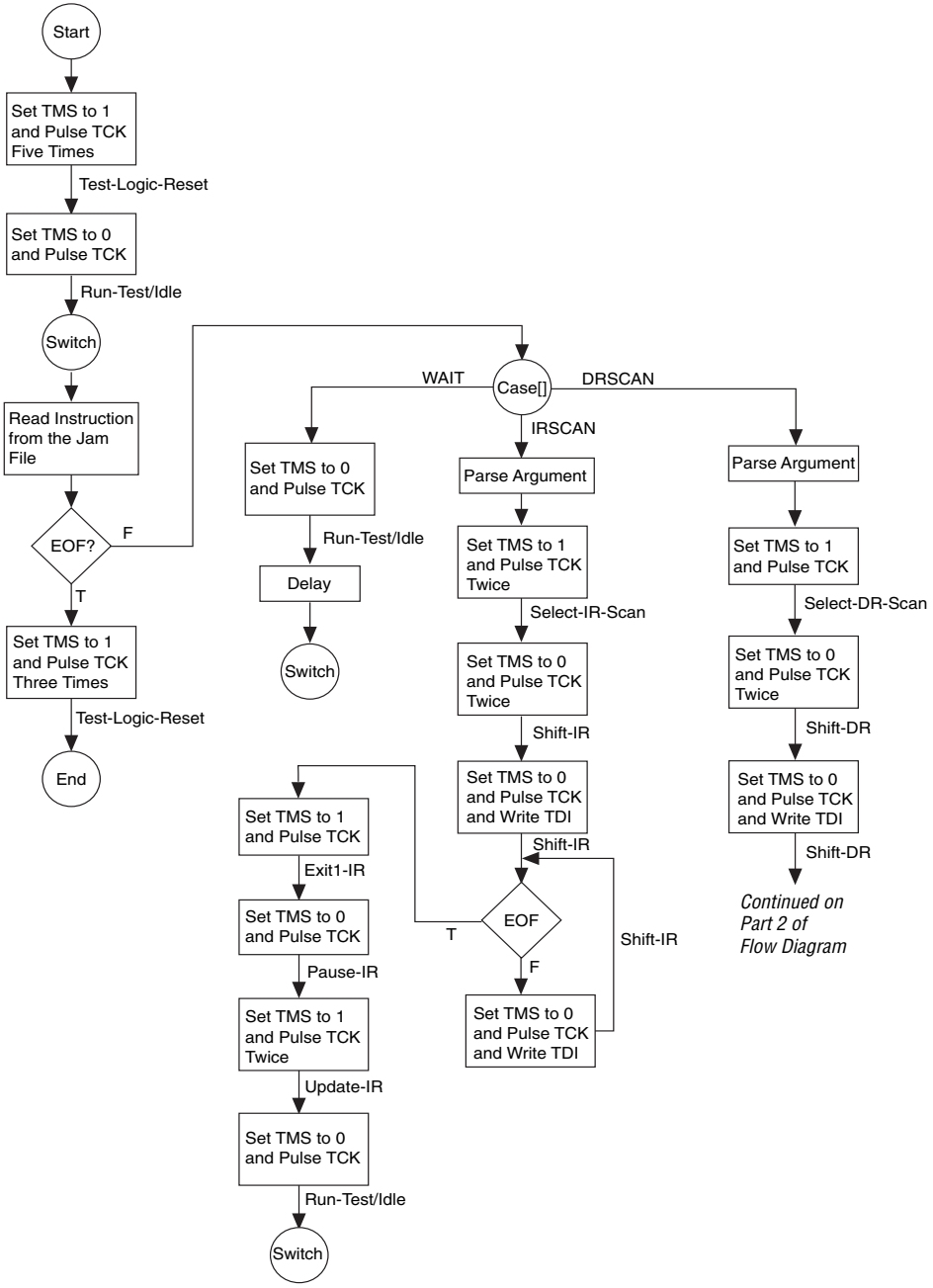
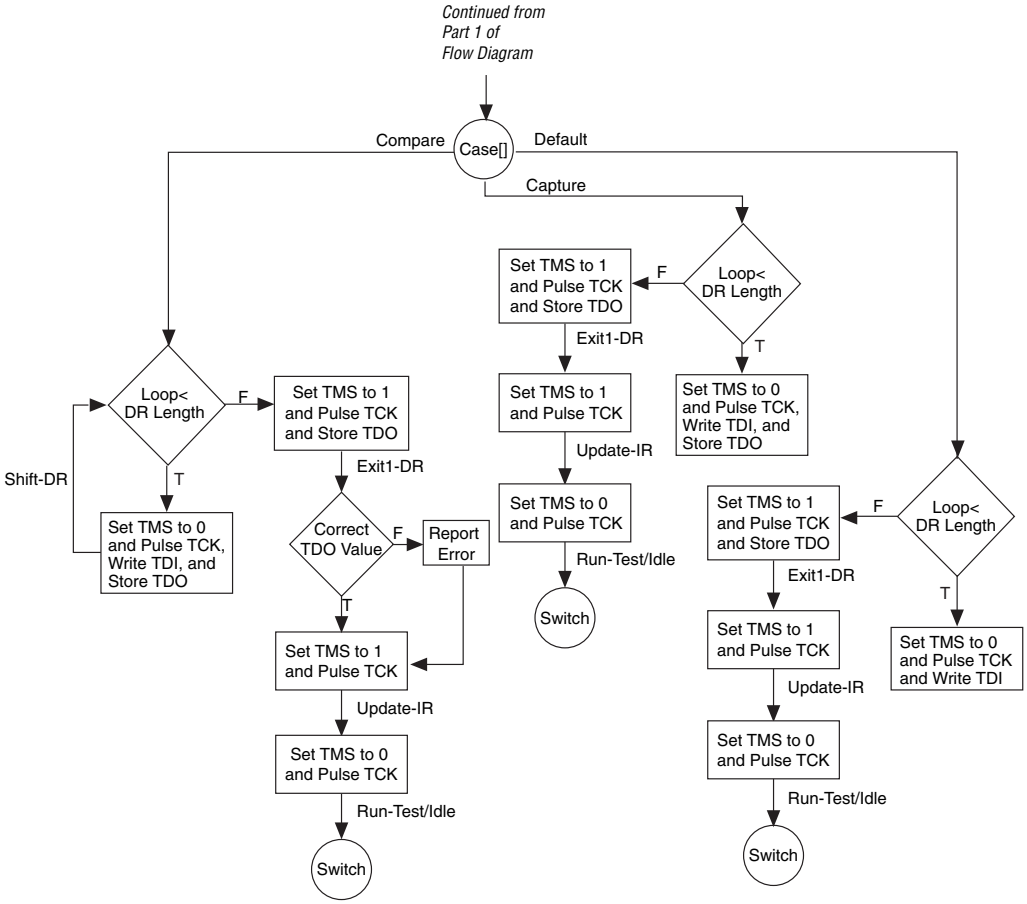


Figure 13–24. Jam Player Flow Diagram (Part 2 of 2)



Execution of a Jam program starts at the beginning of the program. The program flow is controlled using `GOTO`, `CALL/RETURN`, and `FOR/NEXT` structures. The `GOTO` and `CALL` statements refer to labels that are symbolic names for program statements located elsewhere in the Jam program. The language itself enforces almost no constraints on the organizational structure or control flow of a program.



The Jam language does not support linking multiple Jam programs together or including the contents of another file into a Jam program.

Jam Instructions

Each Jam statement begins with one of the instruction names listed in Table 13–8. The instruction names, including the names of the optional instructions, are reserved keywords that you cannot use as variable or label identifiers in a Jam program.

Table 13–8. Instruction Names

BOOLEAN	INTEGER	PREIR
CALL	IRSCAN	PRINT
CRC	IRSTOP	PUSH
DRSCAN	LET	RETURN
DRSTOP	NEXT	STATE
EXIT	NOTE	WAIT
EXPORT	POP	VECTOR *(1)*
FOR	POSTDR	VMAP *(1)*
GOTO	POSTIR	—
IF	PREDR	—

Note to Table 13–8:
(1) This instruction name is an optional language extension.

Table 13–9 shows the state names that are reserved keywords in the Jam language. These keywords correspond to the state names specified in the IEEE Std. 1149.1 JTAG specification.

Table 13–9. Reserved Keywords (Part 1 of 2)

IEEE Std. 1149.1 JTAG State Names	Jam Reserved State Names
Test-Logic-Reset	RESET
Run-Test-Idle	IDLE
Select-DR-Scan	DRSELECT
Capture-DR	DRCAPTURE
Shift-DR	DRSHIFT
Exit1-DR	DREXIT1
Pause-DR	DRPAUSE
Exit2-DR	DREXIT2
Update-DR	DRUPDATE
Select-IR-Scan	IRSELECT
Capture-IR	IRCAPTURE

Table 13–9. Reserved Keywords (Part 2 of 2)	
IEEE Std. 1149.1 JTAG State Names	**Jam Reserved State Names**
Shift-IR	IRSHIFT
Exit1-IR	IREXIT1
Pause-IR	IRPAUSE
Exit2-IR	IREXIT2
Update-IR	IRUPDATE

Example Jam File that Reads the IDCODE

The following illustrates the flexibility and utility of the Jam STAPL. The example code reads the IDCODE out of a single device in a JTAG chain.



The array variable, I_IDCODE, is initialized with the IDCODE instruction bits ordered the LSB first (on the left) to most significant bit (MSB) (on the right). This order is important because the array field in the IRSCAN instruction is always interpreted and sent, MSB to LSB.

Example Jam File Reading IDCODE

```
BOOLEAN read_data[32];
BOOLEAN I_IDCODE[10] = BIN 1001101000; 'assumed
BOOLEAN ONES_DATA[32] = HEX FFFFFFFF;
INTEGER i;
'Set up stop state for IRSCAN
IRSTOP IRPAUSE;
'Initialize device
STATE RESET;
IRSCAN 10, I_IDCODE[0..9]; 'LOAD IDCODE INSTRUCTION
STATE IDLE;
WAIT 5 USEC, 3 CYCLES;
DRSCAN 32, ONES_DATA[0..31], CAPTURE read_data[0..31];
'CAPTURE IDCODE
PRINT "IDCODE:";
FOR i=0 to 31;
PRINT read_data[i];
NEXT i;
EXIT 0;
```

Combining Configuration Schemes

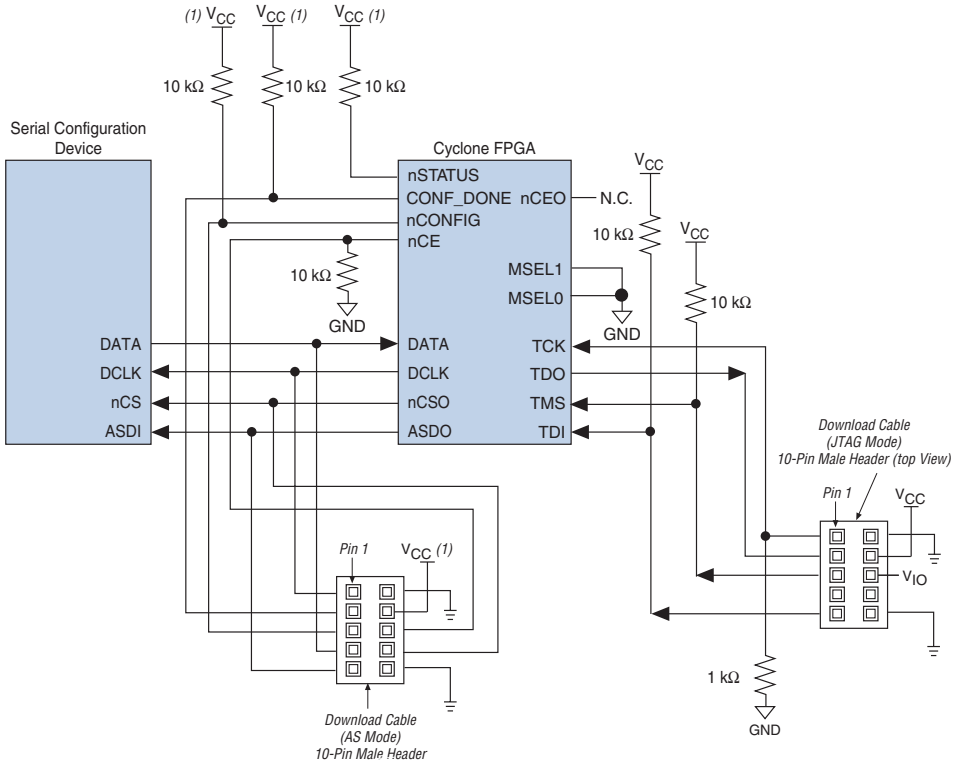
This section shows you how to configure Cyclone FPGAs using multiple configuration schemes on the same board.

Active Serial and JTAG

You can combine the AS configuration scheme with JTAG-based configuration. Set the `MSEL[1..0]` pins to `00` in this setup, as shown in Figure 13–25. This setup uses two 10-pin download cable headers on the board. The first header programs the serial configuration device in-system via the AS programming interface, and the second header configures the Cyclone FPGA directly via the JTAG interface.

If you try configuring the device using both schemes simultaneously, JTAG configuration takes precedence and AS configuration is terminated.

Figure 13–25. Combining AS and JTAG Configuration



Note to Figure 13–25:
(1) Connect these pull-up resistors to 3.3 V.

Device Configuration Pins

Tables 13–10 through 13–12 describe the connections and functionality of all the configuration related pins on the Cyclone device. Table 13–10 describes the dedicated configuration pins. These pins are required to be connected properly on your board for successful configuration. Some of these pins may not be required for your configuration schemes.

Table 13–10. Dedicated Cyclone Device Configuration Pins (Part 1 of 3)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
MSEL1 MSEL0	–	All	Input	Two-bit configuration input that set the Cyclone device configuration scheme (see Table 13–2). Use these pins to select the Cyclone configuration schemes for the appropriate connections. These pins must remain at a valid state during power-up before nCONFIG is pulled low to initiate a reconfiguration and during configuration. This pin uses Schmitt trigger input buffers.
nCONFIG	–	All	Input	Configuration control input. Pulling this pin low during user-mode causes the FPGA to lose its configuration data, enter a reset state, and tri-state all I/O pins. Returning this pin to a logic high initiates a reconfiguration. If the configuration scheme uses an enhanced configuration device or EPC2 device, the nCONFIG pin can be tied directly to V_{CC} or to the configuration device's nINIT_CONF pin. This pin uses Schmitt trigger input buffers

Table 13–10. Dedicated Cyclone Device Configuration Pins (Part 2 of 3)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nSTATUS	–	All	Bidirectional open-drain	The device drives nSTATUS low immediately after power-up and releases it within 5 µs. (When using a configuration device, the configuration device holds nSTATUS low for up to 200 ms.) Status output. If an error occurs during configuration, nSTATUS is pulled low by the target device. Status input. If an external source 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 the design uses a configuration device, driving nSTATUS low causes the configuration device to attempt to configure the FPGA, but since the FPGA ignores transitions on nSTATUS in user-mode, the FPGA does not reconfigure. To initiate a reconfiguration, nCONFIG must be pulled low. The OE and nCS pins in the enhanced configuration devices and EPC2 devices have optional internal programmable pull-up resistors. If the design uses internal pull-up resistors, do not use external 10-kΩ pull-up resistors on these pins. This pin uses Schmitt trigger input buffers
CONF_DONE	–	All	Bidirectional open-drain	Status output. The target device drives the CONF_DONE pin low before and during configuration. Once all configuration data is received without error and the initialization clock 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. Driving CONF_DONE low after configuration and initialization does not affect the configured device. The OE and nCS pins in the enhanced configuration devices and EPC2 devices have optional internal programmable pull-up resistors. If the design uses internal pull-up resistors, do not use external 10-kΩ pull-up resistors on these pins. This pin uses Schmitt trigger input buffers

Table 13–10. Dedicated Cyclone Device Configuration Pins (Part 3 of 3)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DCLK	–	PS AS	Input (PS) Output (AS)	In PS configuration, the clock input clocks data from an external source into the target device. Data is latched into the FPGA on the rising edge of DCLK. In AS configuration, DCLK is an output from the Cyclone FPGA that provides timing for the configuration interface. After configuration, the logic levels on this pin do not affect the Cyclone FPGA. This pin uses Schmitt trigger input buffers
ASDO	I/O in PS mode, N/A in AS mode	AS	Output	Control signal from the Cyclone FPGA to the serial configuration device in AS mode used to read out configuration data.
nCSO	I/O in PS mode, N/A in AS mode	AS	Output	Output control signal from the Cyclone FPGA to the serial configuration device in AS mode that enables the configuration device.
nCE	–	All	Input	Active-low chip enable. The nCE pin activates the device with a low signal to allow configuration. The nCE pin must be held low during configuration, initialization, and user mode. In single device configuration, tie the nCE pin low. In multi-device configuration, the first device's nCE pin is tied low while its nCEO pin is connected to nCE of the next device in the chain. Hold the nCE pin low for programming the FPGA via JTAG. This pin uses Schmitt trigger input buffers
nCEO	–	All	Output	Output that drives low when device configuration is complete. In single device configuration, this pin is left floating. In multi-device configuration, this pin feeds the next device's nCE pin. The nCEO of the last device in the chain is left floating.
DATA0	–	All	Input	Data input. In serial configuration mode, bit-wide configuration data is presented to the target device on the DATA0 pin. Toggling DATA0 after configuration does not affect the configured device. This pin uses Schmitt trigger input buffers

Table 13–11 describes 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-ups.

Table 13–11. Optional Cyclone Device 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. Can be used to indicate when the device has initialized and is in user mode. The INIT_DONE pin must be pulled to V_{CC} with a 10-kΩ resistor. The INIT_DONE pin drives low during configuration. Before and after configuration, the INIT_DONE pin is released and is pulled to V_{CC} by an external pull-up resistor. Because INIT_DONE is tri-stated before configuration, it is pulled high by the external pull-up resistor. 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.
DEV_OE	N/A if the option is on, I/O if the option is off.	Input	Optional pin that allows the user 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 the option is on, I/O if the 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.

Table 13–12 describes the dedicated JTAG pins. JTAG pins must be kept stable before and during configuration to prevent accidental loading of JTAG instructions.

Table 13–12. Dedicated JTAG Pins

Pin Name	User Mode	Pin Type	Description
TDI	N/A	Input	Serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to V_{CC}. This pin uses Schmitt trigger input buffers
TDO	N/A	Output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by leaving this pin unconnected.
TMS	N/A	Input	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions within the state machine occur on the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to V_{CC}. This pin uses Schmitt trigger input buffers
TCK	N/A	Input	The clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. If the JTAG interface is not required on the board, the JTAG circuitry can be disabled by connecting this pin to ground. This pin uses Schmitt trigger input buffers

Referenced Documents

This chapter references the following documents:

- *AN 39: IEEE 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*
- *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming*
- *AN 423: Configuring the MicroBlaster Passive Serial Software Driver*
- *ByteBlaster II Download Cable User Guide*
- *ByteBlasterMV Download Cable User Guide*
- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*
- *DC and Switching Characteristics* chapter in the *Cyclone Device Handbook*
- *Design Debugging Using the SignalTap II Embedded Logic Analyzer* chapter in volume 3 of the *Quartus II Handbook*
- *MasterBlaster Serial/USB Communications Cable User Guide*
- *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet*
- *Software Settings* section in volume 2 of the *Configuration Handbook*

Document Revision History

Table 13–13 shows the revision history for this chapter.

Table 13–13. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.8	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.7	• Added document revision history. • Removed a note from Table 13–2. • Updated Figure 13–1. • Updated Table 13–3. • Updated feetpara note in "Active Serial Configuration (Serial Configuration Devices)" section. • Updated feetpara note on page 13–18. • Updated *Note (2)* in Figure 13–11. • Updated *Note (4)* in Figure 13–12. • Updated *Note (2)* in Figure 13–19.	—
July 2006 v1.6	Updated Figure 13–19.	—
August 2005 v1.5	• Updated tables. • Minor text updates.	—
March 2005 v1.4	• Updated Figure 13–1. • Updated Figure 13–10.	—
February 2005 v1.3	Updated Figure 13–13.	—
August 2004 v1.2	• Deleted sections: Programming Configuration Devices, Connecting the JTAG Chain, Passive Serial and JTAG, Device Options, Device Configuration Files, Configuration Reliability, and Board Layout Tips. • Deleted figures: Embedded System Block Diagram, Combining PS & JTAG Configuration, Configuration Options Dialog Box. • Deleted table: Cyclone Configuration Option Bits. • Added: USB Blaster to cable list; new Figure 13–13; text on pages 13-14, 13-29, and 13-30, and information to Table 13–6. • Changes to Figures 13–14 to 13–16, 13–19, 13–20, 13–25; numbers changed in EP1C4 row of Table 13–3. • Added extensive descriptions of configuration methods under the "Configuring Multiple Devices with the Same Data" section.	—
July 2003 v1.1	Updated **.rbf** sizes. Minor updates throughout the document.	—
May 2003 v1.0	Added document to *Cyclone Device Handbook*.	—

14. Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet

C51014-3.1

Introduction

The serial configuration devices provide the following features:

- 1-, 4-, 16-, 64-, and 128-Mbit flash memory devices that serially configure Stratix® III, Stratix II GX, and Stratix II FPGAs, Arria™ GX FPGAs, and the Cyclone® series FPGAs using the active serial (AS) configuration scheme
- Easy-to-use four-pin interface
- Low cost, low-pin count, and non-volatile memory
- Low current during configuration and near-zero standby mode current
- 3.3-V operation
- Available in 8-pin and 16-pin small outline integrated circuit (SOIC) package
- Enables the Nios® processor to access unused flash memory through AS memory interface
- Re-programmable memory with more than 100,000 erase/program cycles
- Write protection support for memory sectors using status register bits
- In-system programming support with SRunner software driver
- In-system programming support with USB Blaster™, EthernetBlaster™, or ByteBlaster™ II download cables
- Additional programming support with the Altera® Programming Unit (APU) and programming hardware from BP Microsystems, System General, and other vendors
- Software design support with the Altera Quartus® II development system for Windows-based PCs as well as Sun SPARC station and HP 9000 Series 700/800
- Delivered with the memory array erased (all the bits set to 1)

The term "serial configuration devices" used in this document refers to Altera EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128.

Functional Description

With SRAM-based devices that support active serial configuration, configuration data must be reloaded each time the device powers up, the system reconfigures, or when new configuration data is required. Serial configuration devices are flash memory devices with a serial interface that can store configuration data for FPGA devices that support active serial configuration and reload the data to the device upon power-up or reconfiguration. Table 14–1 lists the serial configuration devices.

Table 14–1. Serial Configuration Devices (3.3-V Operation)

Device	Memory Size (Bits)
EPCS1	1,048,576
EPCS4	4,194,304
EPCS16	16,777,216
EPCS64	67,108,864
EPCS128	134,217,728

For an 8-pin SOIC package, you can migrate vertically from the EPCS1 to the EPCS4 or EPCS16 since the EPCS devices are offered in the same device package. Similarly, for a 16-pin SOIC package, you can migrate vertically from the EPCS16 to the EPCS64 or EPCS128.



EPCS16 is available in 8-pin and 16-pin SOIC packages.

Table 14–2 lists the serial configuration device used with each Stratix III FPGA and the configuration file size. Stratix III devices can be used with EPCS16, EPCS64, or EPCS128.

Table 14–2. Serial Configuration Device Support for Stratix III Devices (Part 1 of 2)

Stratix III Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device				
		EPCS1	EPCS4	EPCS16	EPCS64	EPCS128
EP3SL50	22,178,792	—	—	✓ *(2)*	✓	✓
EP3SL70	22,178,792	—	—	✓ *(2)*	✓	✓
EP3SL110	47,413,312	—	—	—	✓	✓
EP3SL150	47,413,312	—	—	—	✓	✓
EP3SL200	93,324,656	—	—	—	✓ *(2)*	✓
EP3SL340	117,384,664	—	—	—	—	✓
EP3SE50	25,891,968	—	—	—	✓	✓
EP3SE80	48,225,392	—	—	—	✓	✓

Table 14–2. Serial Configuration Device Support for Stratix III Devices (Part 2 of 2)

Stratix III Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device				
		EPCS1	EPCS4	EPCS16	EPCS64	EPCS128
EP3SE110	48,225,392	—	—	—	✓	✓
EP3SE260	93,324,656	—	—	—	✓ *(2)*	✓

Notes to Table 14–2:
(1) These are uncompressed file sizes.
(2) This is with the Stratix III compression feature enabled.

Table 14–3 lists the serial configuration device used with each Stratix II GX FPGA and the configuration file size. Stratix II GX devices can be used with EPCS16, EPCS64, or EPCS128.

Table 14–3. Serial Configuration Device Support for Stratix II GX Devices

Stratix II GX Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device				
		EPCS1	EPCS4	EPCS16	EPCS64	EPCS128
EP2SGX30C EP2SGX30D	9,640,672	—	—	✓	✓	✓
EP2SGX60C EP2SGX60D EP2SGX60E	16,951,824	—	—	✓ *(2)*	✓	✓
EP2SGX90E EP2SGX90F	25,699,104	—	—	—	✓	✓
EP2SGX130G	37,325,760	—	—	—	✓	✓

Notes to Table 14–3:
(1) These are uncompressed file sizes.
(2) This is with the Stratix II GX compression feature enabled.

Table 14–4 lists the serial configuration device used with each Stratix II FPGA and the configuration file size. Stratix II devices can be used with EPCS4, EPCS16, EPCS64, or EPCS128.

Table 14–4. Serial Configuration Device Support for Stratix II Devices

Stratix II Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device			
		EPCS4	EPCS16	EPCS64	EPCS128
EP2S15	4,721,544	✓ *(2)*	✓	✓	✓
EP2S30	9,640,672	—	✓	✓	✓
EP2S60	16,951,824	—	✓ *(2)*	✓	✓
EP2S90	25,699,104	—	✓ *(2)*	✓	✓
EP2S130	37,325,760	—	—	✓	✓
EP2S180	49,814,760	—	—	✓	✓

Notes to Table 14–4:
(1) These are uncompressed file sizes.
(2) This is with the Stratix II compression feature enabled.

Table 14–5 lists the serial configuration device used with each Arria GX FPGA and the configuration file size. Arria GX devices can be used with EPCS16, EPCS64, or EPCS128.

Table 14–5. Serial Configuration Device Support for Arria GX Devices

Arria GX Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device				
		EPCS1	EPCS4	EPCS16	EPCS64	EPCS128
EP1AGX20C	9,640,672	—	—	✓	✓	✓
EP1AGX35C EP1AGX35D	9,640,672	—	—	✓	✓	✓
EP1AGX50C EP1AGX50D	16,951,824	—	—	✓ *(2)*	✓	✓
EP1AGX60C EP1AGX60D EP1AGX60E	16,951,824	—	—	✓ *(2)*	✓	✓
EP1AGX90E	25,699,104	—	—	—	✓	✓

Notes to Table 14–5:
(1) These are uncompressed file sizes.
(2) This is with the Arria GX compression feature enabled.

Table 14–6 lists the serial configuration device used with each Cyclone III FPGA and the configuration file size. Cyclone III devices can be used with EPCS4, EPCS16, EPCS64, or EPCS128.

Table 14–6. Serial Configuration Device for Cyclone III Devices

Cyclone III Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device				
		EPCS1	EPCS4	EPCS16	EPCS64	EPCS128
EP3C5	2,944,088	—	✓	✓	✓	✓
EP3C10	2,944,088	—	✓	✓	✓	✓
EP3C16	4,086,848	—	✓	✓	✓	✓
EP3C25	5,748,552	—	—	✓	✓	✓
EP3C40	9,534,304	—	—	✓	✓	✓
EP3C55	14,889,560	—	—	✓	✓	✓
EP3C80	19,965,752	—	—	✓ *(2)*	✓	✓
EP3C120	28,571,696	—	—	—	✓	✓

Notes to Table 14–6:
(1) These are uncompressed file sizes.
(2) This is with the Cyclone III compression feature enabled.

Table 14–7 lists the serial configuration device used with each Cyclone II FPGA and the configuration file size. Cyclone II devices can be used with EPCS1, EPCS4, EPCS16, EPCS64, or EPCS128.

Table 14–7. Serial Configuration Device for Cyclone II Devices

Cyclone II Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device				
		EPCS1	EPCS4	EPCS16	EPCS64	EPCS128
EP2C5	1,265,792	✓ *(2)*	✓	✓	✓	✓
EP2C8	1,983,536	—	✓	✓	✓	✓
EP2C20	3,892,496	—	✓	✓	✓	✓
EP2C35	6,848,608	—	—	✓	✓	✓
EP2C50	9,951,104	—	—	✓	✓	✓
EP2C70	14,319,216	—	—	✓	✓	✓

Notes to Table 14–7:
(1) These are uncompressed file sizes.
(2) This is with the Cyclone II compression feature enabled.

Table 14–8 lists the serial configuration device used with each Cyclone FPGA and the configuration file size. Cyclone devices can be used with EPCS1, EPCS4, EPCS16, EPCS64, or EPCS128.

Table 14–8. Serial Configuration Device Support for Cyclone Devices

Cyclone Device	Raw Binary File Size (Bits) *(1)*	Serial Configuration Device				
		EPCS1	EPCS4	EPCS16	EPCS64	EPCS128
EP1C3	627,376	✓	✓	✓	✓	✓
EP1C4	924,512	✓	✓	✓	✓	✓
EP1C6	1,167,216	✓ *(2)*	✓	✓	✓	✓
EP1C12	2,323,240	—	✓	✓	✓	✓
EP1C20	3,559,608	—	✓	✓	✓	✓

Notes to Table 14–8:
(1) These are uncompressed file sizes.
(2) This is with the Cyclone compression feature enabled.

With the new data-decompression feature in the Stratix III, Stratix II GX, and Stratix II FPGAs, Arria GX FPGAs, and Cyclone FPGA families, you can use smaller serial configuration devices to configure larger FPGAs.

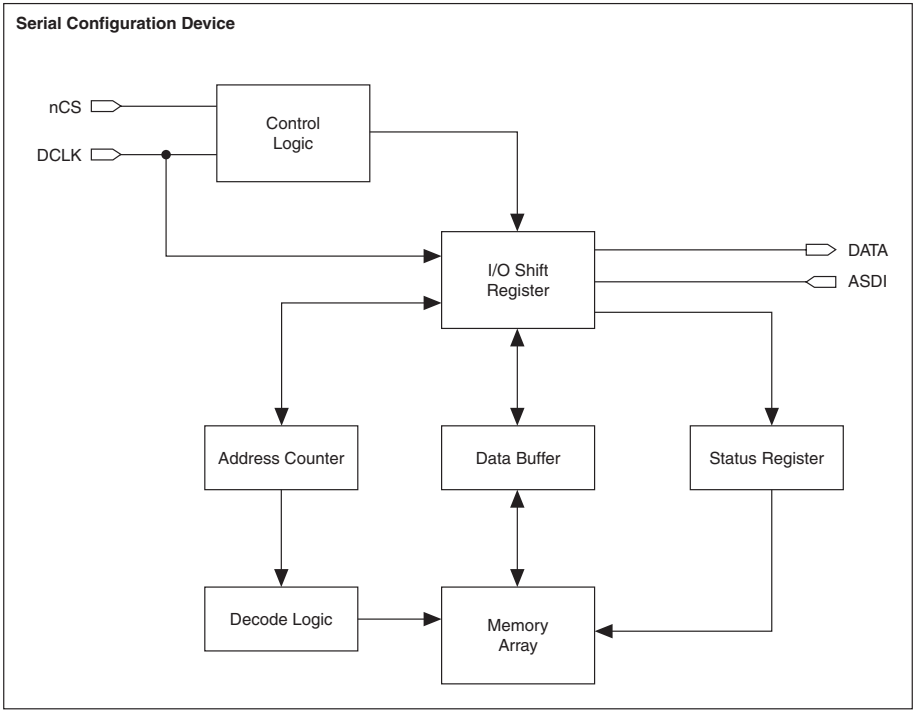
Serial configuration devices cannot be cascaded.

For more information about the FPGA decompression feature, refer to the configuration chapter in the appropriate device handbook.

The serial configuration devices are designed to configure Stratix III, Stratix II GX, and Stratix II FPGAs and the Cyclone series FPGAs and cannot configure other existing Altera FPGA device families.

Figure 14–1 shows the serial configuration device block diagram.

Figure 14–1. Serial Configuration Device Block Diagram



Accessing Memory in Serial Configuration Devices

You can access the unused memory locations of the serial configuration device to store or retrieve data through the Nios processor and SOPC Builder. SOPC Builder is an Altera tool for creating bus-based (especially microprocessor-based) systems in Altera devices. SOPC Builder assembles library components such as processors and memories into custom microprocessor systems.

SOPC Builder includes the EPCS device controller core, which is an interface core specifically designed to work with the serial configuration device. With this core, you can create a system with a Nios embedded processor that allows software access to any memory location within the serial configuration device.

For more information about accessing memory within the serial configuration device, refer to the *Active Serial Memory Interface Data Sheet*.

Active Serial FPGA Configuration

The following Altera FPGAs support Active Serial (AS) configuration scheme with serial configuration devices:

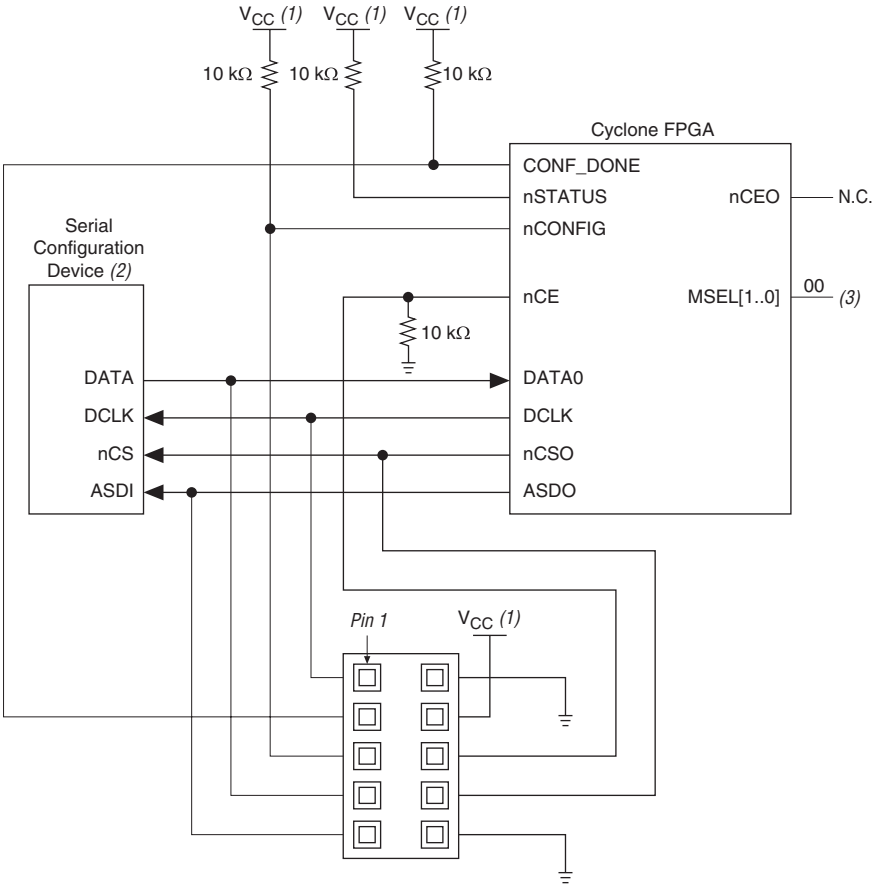
- Stratix III
- Stratix II GX
- Stratix II
- Arria GX
- Cyclone series FPGAs



This section is only relevant for FPGAs that support the AS configuration scheme.

There are four signals on the serial configuration device that interface directly with the FPGA's control signals. The serial configuration device signals DATA, DCLK, ASDI, and nCS interface with DATA0, DCLK, ASDO, and nCSO control signals on the FPGA, respectively. Figure 14–2 shows a serial configuration device programmed via a download cable, which configures an FPGA in AS mode. Figure 14–3 shows a serial configuration device programmed using the APU or a third-party programmer configuring an FPGA in AS configuration mode.

Figure 14–2. Cyclone FPGA Configuration in AS Mode (Serial Configuration Device Programmed Using Download Cable) *Note (4)*



Notes to Figure 14–2:

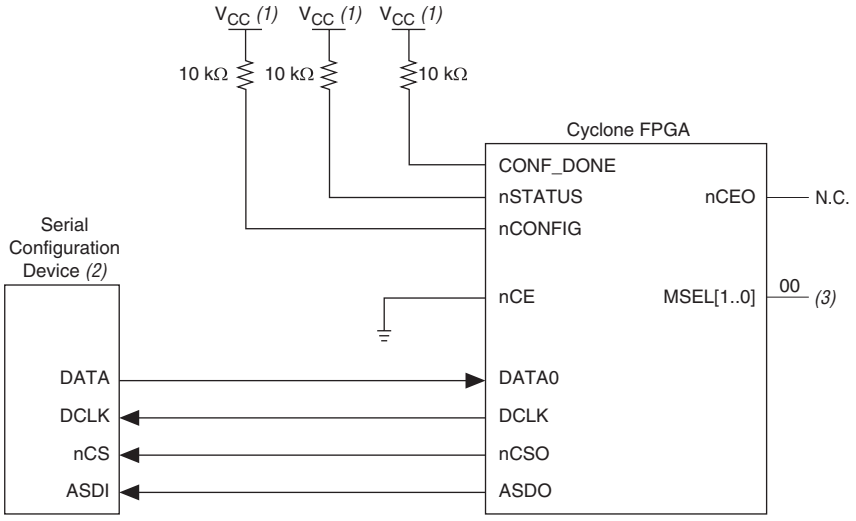
(1) V_{CC} = 3.3 V.

(2) Serial configuration devices cannot be cascaded.

(3) Connect the FPGA `MSEL[]` input pins to select the AS configuration mode. For details, refer to the appropriate FPGA family chapter in the *Configuration Handbook*.

(4) For more information about configuration pin I/O requirements in an AS scheme for a Cyclone III FPGA, refer to the *Configuring Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*.

Figure 14–3. Cyclone FPGA Configuration in AS Mode (Serial Configuration Device Programmed by APU or Third-Party Programmer) *Note (4)*



Notes to Figure 14–3:

(1) V_{CC} = 3.3 V.

(2) Serial configuration devices cannot be cascaded.

(3) Connect the FPGA MSEL[] input pins to select the AS configuration mode. For details, refer to the appropriate FPGA family chapter in the *Configuration Handbook*.

(4) For more information about configuration pin I/O requirements in an AS scheme for a Cyclone III FPGA, refer to the *Configuring Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*.

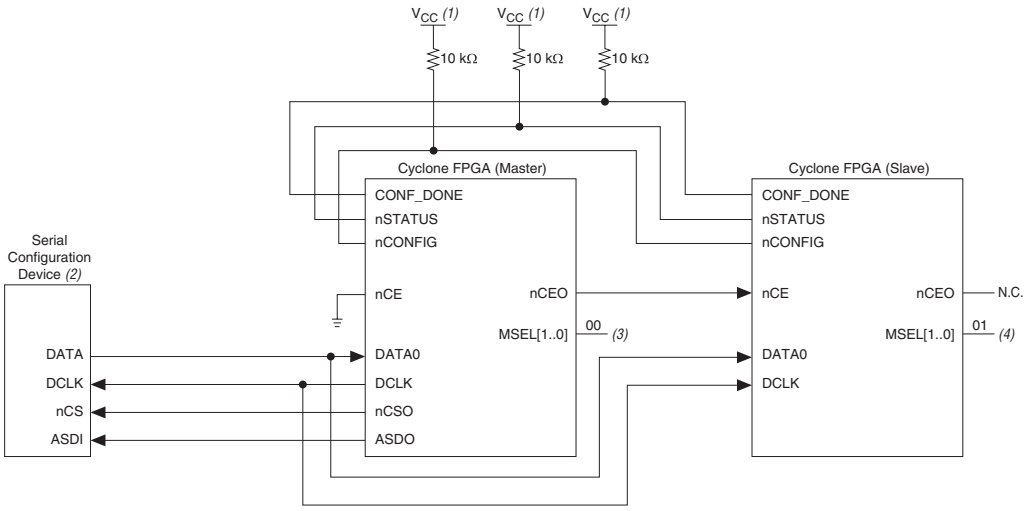
The FPGA acts as the configuration master in the configuration flow and provides the clock to the serial configuration device. The FPGA enables the serial configuration device by pulling the nCS signal low via the nCSO signal (refer to Figures 14–2 and 14–3). Subsequently, the FPGA sends the instructions and addresses to the serial configuration device via the ASDO signal. The serial configuration device responds to the instructions by sending the configuration data to the FPGA's DATA0 pin on the falling edge of DCLK. The data is latched into the FPGA on the DCLK signal's falling edge.

The FPGA controls the nSTATUS and CONF_DONE pins during configuration in AS mode. If the CONF_DONE signal does not go high at the end of configuration or if the signal goes high too early, the FPGA will pulse its nSTATUS pin low to start reconfiguration. Upon successful configuration, the FPGA releases the CONF_DONE pin, allowing the external 10-kΩ resistor to pull this signal high. Initialization begins after the CONF_DONE goes high. After initialization, the FPGA enters user mode.

Refer to the configuration chapter in the appropriate device handbook for more information about configuring the FPGAs in AS mode or other configuration modes.

Multiple devices can be configured by a single EPCS device. However, serial configuration devices cannot be cascaded. Refer to Table 14–1 to ensure the programming file size of the cascaded FPGAs does not exceed the capacity of a serial configuration device. Figure 14–4 shows the AS configuration scheme with multiple FPGAs in the chain. The first FPGA is the configuration master and has its MSEL[] pins set to AS mode. The following FPGAs are configuration slave devices and have their MSEL[] pins set to PS mode.

Figure 14–4. Multiple Devices in AS Mode *Note (5)*



Notes to Figure 14–4:

(1) V_{CC} = 3.3 V.

(2) Serial configuration devices cannot be cascaded.

(3) Connect the FPGA MSEL[] input pins to select the AS configuration mode. For details, refer to the appropriate FPGA family chapter in the *Configuration Handbook*.

(4) Connect the FPGA MSEL[] input pins to select the PS configuration mode. For details, refer to the appropriate FPGA family chapter in the *Configuration Handbook*.

(5) For more information about configuration pin I/O requirements in an AS scheme for a Cyclone III FPGA, refer to the *Configuring Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*.

Serial Configuration Device Memory Access

This section describes the serial configuration device's memory array organization and operation codes. Timing specifications for the memory are provided in the "Timing Information" section.

Memory Array Organization

Table 14–9 provides details about the memory array organization in EPCS128, EPCS64, EPCS16, EPCS4, and EPCS1.

Table 14–9. Memory Array Organization in Serial Configuration Devices

Details	EPCS128	EPCS64	EPCS16	EPCS4	EPCS1
Bytes (bits)	16,777,216 bytes (128 Mbits)	8,388,608 bytes (64 Mbits)	2,097,152 bytes (16 Mbits)	524,288 bytes (4 Mbits)	131,072 bytes (1 Mbit)
Number of sectors	64	128	32	8	4
Bytes (bits) per sector	262,144 (2 Mbits)	65,536 bytes (512 Kbits)	65,536 bytes (512 Kbits)	65,536 bytes (512 Kbits)	32,768 bytes (256 Kbits)
Pages per sector	1,024	256	256	256	128
Total number of pages	65,536	32,768	8,192	2,048	512
Bytes per page	256 bytes	256 bytes	256 bytes	256 bytes	256 bytes

Tables 14–10 through 14–14 show the address range for each sector in EPCS128, EPCS64, EPCS16, EPCS4, and EPCS1.

Table 14–10. Address Range for Sectors in EPCS128 (Part 1 of 3)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
63	H'FC0000	H'FFFFFF
62	H'F80000	H'FBFFFF
61	H'F40000	H'F7FFFF
60	H'F00000	H'F3FFFF
59	H'EC0000	H'EFFFFF
58	H'E80000	H'EBFFFF
57	H'E40000	H'E7FFFF
56	H'E00000	H'E3FFFF
55	H'DC0000	H'DFFFFF
54	H'D80000	H'DBFFFF

Table 14–10. Address Range for Sectors in EPCS128 (Part 2 of 3)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
53	H'D40000	H'D7FFFF
52	H'D00000	H'D3FFFF
51	H'CC0000	H'CFFFFF
50	H'C80000	H'CBFFFF
49	H'C40000	H'C7FFFF
48	H'C00000	H'C3FFFF
47	H'BC0000	H'BFFFFF
46	H'B80000	H'BBFFFF
45	H'B40000	H'B7FFFF
44	H'B00000	H'B3FFFF
43	H'AC0000	H'AFFFFF
42	H'A80000	H'ABFFFF
41	H'A40000	H'A7FFFF
40	H'A00000	H'A3FFFF
39	H'9C0000	H'9FFFFF
38	H'980000	H'9BFFFF
37	H'940000	H'97FFFF
36	H'900000	H'93FFFF
35	H'8C0000	H'8FFFFF
34	H'880000	H'8BFFFF
33	H'840000	H'87FFFF
32	H'800000	H'83FFFF
31	H'7C0000	H'7FFFFF
30	H'780000	H'7BFFFF
29	H'740000	H'77FFFF
28	H'700000	H'73FFFF
27	H'6C0000	H'6FFFFF
26	H'680000	H'6BFFFF
25	H'640000	H'67FFFF
24	H'600000	H'63FFFF
23	H'5C0000	H'5FFFFF
22	H'580000	H'5BFFFF

Table 14–10. Address Range for Sectors in EPCS128 (Part 3 of 3)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
21	H'540000	H'57FFFF
20	H'500000	H'53FFFF
19	H'4C0000	H'4FFFFF
18	H'480000	H'4BFFFF
17	H'440000	H'47FFFF
16	H'400000	H'43FFFF
15	H'3C0000	H'3FFFFF
14	H'380000	H'3BFFFF
13	H'340000	H'37FFFF
12	H'300000	H'33FFFF
11	H'2C0000	H'2FFFFF
10	H'280000	H'2BFFFF
9	H'240000	H'27FFFF
8	H'200000	H'23FFFF
7	H'1C0000	H'1FFFFF
6	H'180000	H'1BFFFF
5	H'140000	H'17FFFF
4	H'100000	H'13FFFF
3	H'0C0000	H'0FFFFF
2	H'080000	H'0BFFFF
1	H'040000	H'07FFFF
0	H'000000	H'03FFFF

Table 14–11. Address Range for Sectors in EPCS64 (Part 1 of 5)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
127	H'7F0000	H'7FFFFF
126	H'7E0000	H'7EFFFF
125	H'7D0000	H'7DFFFF
124	H'7C0000	H'7CFFFF
123	H'7B0000	H'7BFFFF

Table 14–11. Address Range for Sectors in EPCS64 (Part 2 of 5)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
122	H'7A0000	H'7AFFFF
121	H'790000	H'79FFFF
120	H'780000	H'78FFFF
119	H'770000	H'77FFFF
118	H'760000	H'76FFFF
117	H'750000	H'75FFFF
116	H'740000	H'74FFFF
115	H'730000	H'73FFFF
114	H'720000	H'72FFFF
113	H'710000	H'71FFFF
112	H'700000	H'70FFFF
111	H'6F0000	H'6FFFFF
110	H'6E0000	H'6EFFFF
109	H'6D0000	H'6DFFFF
108	H'6C0000	H'6CFFFF
107	H'6B0000	H'6BFFFF
106	H'6A0000	H'6AFFFF
105	H'690000	H'69FFFF
104	H'680000	H'68FFFF
103	H'670000	H'67FFFF
102	H'660000	H'66FFFF
101	H'650000	H'65FFFF
100	H'640000	H'64FFFF
99	H'630000	H'63FFFF
98	H'620000	H'62FFFF
97	H'610000	H'61FFFF
96	H'600000	H'60FFFF
95	H'5F0000	H'5FFFFF
94	H'5E0000	H'5EFFFF
93	H'5D0000	H'5DFFFF
92	H'5C0000	H'5CFFFF
91	H'5B0000	H'5BFFFF

Table 14–11. Address Range for Sectors in EPCS64 (Part 3 of 5)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
90	H'5A0000	H'5AFFFF
89	H'590000	H'59FFFF
88	H'580000	H'58FFFF
87	H'570000	H'57FFFF
86	H'560000	H'56FFFF
85	H'550000	H'55FFFF
84	H'540000	H'54FFFF
83	H'530000	H'53FFFF
82	H'520000	H'52FFFF
81	H'510000	H'51FFFF
80	H'500000	H'50FFFF
79	H'4F0000	H'4FFFFF
78	H'4E0000	H'4EFFFF
77	H'4D0000	H'4DFFFF
76	H'4C0000	H'4CFFFF
75	H'4B0000	H'4BFFFF
74	H'4A0000	H'4AFFFF
73	H'490000	H'49FFFF
72	H'480000	H'48FFFF
71	H'470000	H'47FFFF
70	H'460000	H'46FFFF
69	H'450000	H'45FFFF
68	H'440000	H'44FFFF
67	H'430000	H'43FFFF
66	H'420000	H'42FFFF
65	H'410000	H'41FFFF
64	H'400000	H'40FFFF
63	H'3F0000	H'3FFFFF
62	H'3E0000	H'3EFFFF
61	H'3D0000	H'3DFFFF
60	H'3C0000	H'3CFFFF
59	H'3B0000	H'3BFFFF

Table 14–11. Address Range for Sectors in EPCS64 (Part 4 of 5)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
58	H'3A0000	H'3AFFFF
57	H'390000	H'39FFFF
56	H'380000	H'38FFFF
55	H'370000	H'37FFFF
54	H'360000	H'36FFFF
53	H'350000	H'35FFFF
52	H'340000	H'34FFFF
51	H'330000	H'33FFFF
50	H'320000	H'32FFFF
49	H'310000	H'31FFFF
48	H'300000	H'30FFFF
47	H'2F0000	H'2FFFFF
46	H'2E0000	H'2EFFFF
45	H'2D0000	H'2DFFFF
44	H'2C0000	H'2CFFFF
43	H'2B0000	H'2BFFFF
42	H'2A0000	H'2AFFFF
41	H'290000	H'29FFFF
40	H'280000	H'28FFFF
39	H'270000	H'27FFFF
38	H'260000	H'26FFFF
37	H'250000	H'25FFFF
36	H'240000	H'24FFFF
35	H'230000	H'23FFFF
34	H'220000	H'22FFFF
33	H'210000	H'21FFFF
32	H'200000	H'20FFFF
31	H'1F0000	H'1FFFFF
30	H'1E0000	H'1EFFFF
29	H'1D0000	H'1DFFFF
28	H'1C0000	H'1CFFFF
27	H'1B0000	H'1BFFFF

Table 14–11. Address Range for Sectors in EPCS64 (Part 5 of 5)

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
26	H'1A0000	H'1AFFFF
25	H'190000	H'19FFFF
24	H'180000	H'18FFFF
23	H'170000	H'17FFFF
22	H'160000	H'16FFFF
21	H'150000	H'15FFFF
20	H'140000	H'14FFFF
19	H'130000	H'13FFFF
18	H'120000	H'12FFFF
17	H'110000	H'11FFFF
16	H'100000	H'10FFFF
15	H'0F0000	H'0FFFFF
14	H'0E0000	H'0EFFFF
13	H'0D0000	H'0DFFFF
12	H'0C0000	H'0CFFFF
11	H'0B0000	H'0BFFFF
10	H'0A0000	H'0AFFFF
9	H'090000	H'09FFFF
8	H'080000	H'08FFFF
7	H'070000	H'07FFFF
6	H'060000	H'06FFFF
5	H'050000	H'05FFFF
4	H'040000	H'04FFFF
3	H'030000	H'03FFFF
2	H'020000	H'02FFFF
1	H'010000	H'01FFFF
0	H'000000	H'00FFFF

Table 14–12. Address Range for Sectors in EPCS16

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
31	H'1F0000	H'1FFFFF
30	H'1E0000	H'1EFFFF
29	H'1D0000	H'1DFFFF
28	H'1C0000	H'1CFFFF
27	H'1B0000	H'1BFFFF
26	H'1A0000	H'1AFFFF
25	H'190000	H'19FFFF
24	H'180000	H'18FFFF
23	H'170000	H'17FFFF
22	H'160000	H'16FFFF
21	H'150000	H'15FFFF
20	H'140000	H'14FFFF
19	H'130000	H'13FFFF
18	H'120000	H'12FFFF
17	H'110000	H'11FFFF
16	H'100000	H'10FFFF
15	H'0F0000	H'0FFFFF
14	H'0E0000	H'0EFFFF
13	H'0D0000	H'0DFFFF
12	H'0C0000	H'0CFFFF
11	H'0B0000	H'0BFFFF
10	H'0A0000	H'0AFFFF
9	H'090000	H'09FFFF
8	H'080000	H'08FFFF
7	H'070000	H'07FFFF
6	H'060000	H'06FFFF
5	H'050000	H'05FFFF
4	H'040000	H'04FFFF
3	H'030000	H'03FFFF
2	H'020000	H'02FFFF
1	H'010000	H'01FFFF
0	H'000000	H'00FFFF

Table 14–13. Address Range for Sectors in EPCS4

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
7	H'70000	H'7FFFF
6	H'60000	H'6FFFF
5	H'50000	H'5FFFF
4	H'40000	H'4FFFF
3	H'30000	H'3FFFF
2	H'20000	H'2FFFF
1	H'10000	H'1FFFF
0	H'00000	H'0FFFF

Table 14–14. Address Range for Sectors in EPCS1

Sector	Address Range (Byte Addresses in HEX)	
	Start	End
3	H'18000	H'1FFFF
2	H'10000	H'17FFF
1	H'08000	H'0FFFF
0	H'00000	H'07FFF

Operation Codes

This section describes the operations that can be used to access the memory in serial configuration devices. The DATA, DCLK, ASDI, and nCS signals access the memory in serial configuration devices. All serial configuration device operation codes, addresses and data are shifted in and out of the device serially, with the most significant bit (MSB) first.

The device samples the active serial data input on the first rising edge of the DCLK after the active low chip select (nCS) input signal is driven low. Shift the operation code (MSB first) serially into the serial configuration device through the active serial data input pin. Each operation code bit is latched into the serial configuration device on the rising edge of the DCLK.

Different operations require a different sequence of inputs. While executing an operation, you must shift in the desired operation code, followed by the address bytes, data bytes, both, or neither. The device

must drive nCS high after the last bit of the operation sequence is shifted in. Table 14–15 shows the operation sequence for every operation supported by the serial configuration devices.

For the read byte, read status, and read silicon ID operations, the shifted-in operation sequence is followed by data shifted out on the DATA pin. You can drive the nCS pin high after any bit of the data-out sequence is shifted out.

For the write byte, erase bulk, erase sector, write enable, write disable, and write status operations, drive the nCS pin high exactly at a byte boundary (drive the nCS pin high a multiple of eight clock pulses after the nCS pin is driven low); otherwise, the operation is rejected and is not executed.

All attempts to access the memory contents while a write or erase cycle is in progress will not be granted, and the write or erase cycle will continue unaffected.

Table 14–15. Operation Codes for Serial Configuration Devices

Operation	Operation Code *(1)*	Address Bytes	Dummy Bytes	Data Bytes	DCLK f_{MAX} (MHz)
Write enable	0000 0110	0	0	0	25
Write disable	0000 0100	0	0	0	25
Read status	0000 0101	0	0	1 to infinite *(2)*	25
Read bytes	0000 0011	3	0	1 to infinite *(2)*	20
Read silicon ID *(4)*	1010 1011	0	3	1 to infinite *(2)*	25
Write status	0000 0001	0	0	1	25
Write bytes	0000 0010	3	0	1 to 256 *(3)*	25
Erase bulk	1100 0111	0	0	0	25
Erase sector	1101 1000	3	0	0	25
Read Device Identification *(5)*	1001 1111	0	2	1 to infinite *(2)*	25

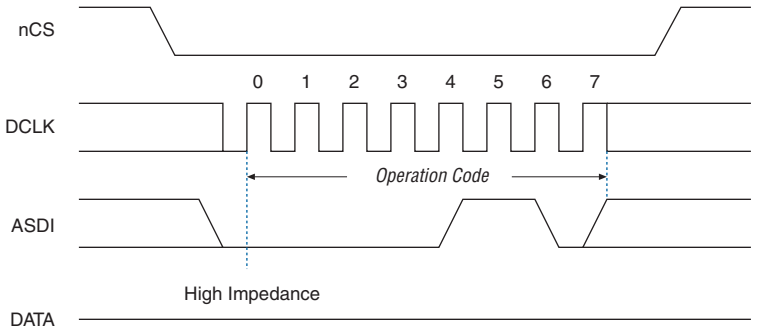
Notes to Table 14–15:

(1) The MSB is listed first and the least significant bit (LSB) is listed last.
(2) The status register, data or silicon ID are read out at least once on the DATA pin and will continuously be read out until nCS is driven high.
(3) Write bytes operation requires at least one data byte on the DATA pin. If more than 256 bytes are sent to the device, only the last 256 bytes are written to the memory.
(4) Read silicon ID operation is available only for EPCS1, EPCS4, EPCS16, and EPCS64.
(5) Read Device Identification operation is available only for EPCS128.

Write Enable Operation

The write enable operation code is b'0000 0110, and the MSB is listed first. The write enable operation sets the write enable latch bit, which is bit 1 in the status register. Always set the write enable latch bit before write bytes, write status, erase bulk, and erase sector operations. Figure 14–5 shows the timing diagram for the write enable operation. Figures 14–7 and 14–8 show the status register bit definitions.

Figure 14–5. Write Enable Operation Timing Diagram



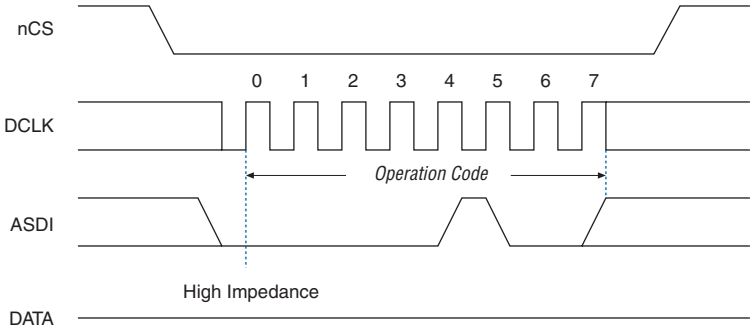
Write Disable Operation

The write disable operation code is b'0000 0100, with the MSB listed first. The write disable operation resets the write enable latch bit, which is bit 1 in the status register. To prevent the memory from being written unintentionally, the write enable latch bit is automatically reset when implementing the write disable operation as well as under the following conditions:

- Power up
- Write bytes operation completion
- Write status operation completion
- Erase bulk operation completion
- Erase sector operation completion

Figure 14–6 shows the timing diagram for the write disable operation.

Figure 14–6. Write Disable Operation Timing Diagram



Read Status Operation

The read status operation code is `b'0000 0101`, with the MSB listed first. You can use the read status operation to read the status register. Figures 14–7 and 14–8 show the status bits in the status register of both serial configuration devices.

Figure 14–7. EPCS4, EPCS16, EPCS64, and EPCS128 Status Register Status Bits

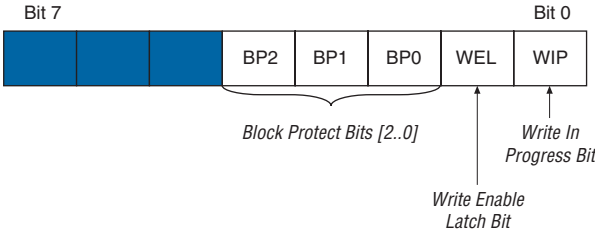
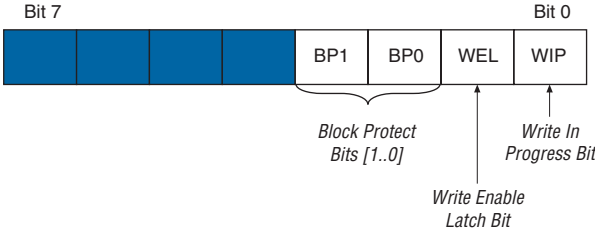


Figure 14–8. EPCS1 Status Register Status Bits



Setting the write in progress bit to 1 indicates that the serial configuration device is busy with a write or erase cycle. Resetting the write in progress bit to 0 means no write or erase cycle is in progress.

Resetting the write enable latch bit to 0 indicates that no write or erase cycle will be accepted. Set the write enable latch bit to 1 before every write bytes, write status, erase bulk, and erase sector operation.

The non-volatile block protect bits determine the area of the memory protected from being written or erased unintentionally. Table 14–16 through Table 14–20 show the protected area in the serial configuration devices with reference to the block protect bits. The erase bulk operation is only available when all the block protect bits are 0. When any of the block protect bits are set to 1, the relevant area is protected from being written by write bytes operations or erased by erase sector operations.

Table 14–16. Block Protection Bits in EPCS1

Status Register Content		Memory Content	
BP1 Bit	**BP0 Bit**	**Protected Area**	**Unprotected Area**
0	0	None	All four sectors: 0 to 3
0	1	Sector 3	Three sectors: 0 to 2
1	0	Two sectors: 2 and 3	Two sectors: 0 and 1
1	1	All sectors	None

Table 14–17. Block Protection Bits in EPCS4

Status Register Content			Memory Content	
BP2 Bit	**BP1 Bit**	**BP0 Bit**	**Protected Area**	**Unprotected Area**
0	0	0	None	All eight sectors: 0 to 7
0	0	1	Sector 7	Seven sectors: 0 to 6
0	1	0	Sectors 6 and 7	Six sectors: 0 to 5
0	1	1	Four sectors: 4 to 7	Four sectors: 0 to 3
1	0	0	All sectors	None
1	0	1	All sectors	None
1	1	0	All sectors	None
1	1	1	All sectors	None

Table 14–18. Block Protection Bits in EPCS16

Status Register Content			Memory Content	
BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	0	0	None	All sectors (32 sectors 0 to 31)
0	0	1	Upper 32nd (Sector 31)	Lower 31/32nds (31 sectors: 0 to 30)
0	1	0	Upper sixteenth (two sectors: 30 and 31)	Lower 15/16ths (30 sectors: 0 to 29)
0	1	1	Upper eighth (four sectors: 28 to 31)	Lower seven-eighths (28 sectors: 0 to 27)
1	0	0	Upper quarter (eight sectors: 24 to 31)	Lower three-quarters (24 sectors: 0 to 23)
1	0	1	Upper half (sixteen sectors: 16 to 31)	Lower half (16 sectors: 0 to 15)
1	1	0	All sectors (32 sectors: 0 to 31)	None
1	1	1	All sectors (32 sectors: 0 to 31)	None

Table 14–19. Block Protection Bits in EPCS64

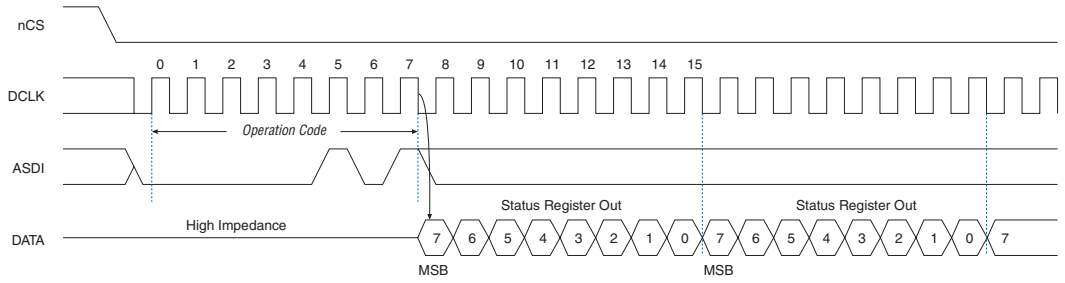
Status Register Content			Memory Content	
BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	0	0	None	All sectors (128 sectors: 0 to 127)
0	0	1	Upper 64th (2 sectors: 126 and 127)	Lower 63/64ths (126 sectors: 0 to 125)
0	1	0	Upper 32nd (4 sectors: 124 to 127)	Lower 31/32nds (124 sectors: 0 to 123)
0	1	1	Upper sixteenth (8 sectors: 120 to 127)	Lower 15/16ths (120 sectors: 0 to 119)
1	0	0	Upper eighth (16 sectors: 112 to 127)	Lower seven-eighths (112 sectors: 0 to 111)
1	0	1	Upper quarter (32 sectors: 96 to 127)	Lower three-quarters (96 sectors: 0 to 95)
1	1	0	Upper half (64 sectors: 64 to 127)	Lower half (64 sectors: 0 to 63)
1	1	1	All sectors (128 sectors: 0 to 127)	None

Table 14–20. Block Protection Bits in EPCS128

Status Register Content			Memory Content	
BP2 Bit	BP1 Bit	BP0 Bit	Protected Area	Unprotected Area
0	0	0	None	All sectors (64 sectors: 0 to 63)
0	0	1	Upper 64th (1 sector: 63)	Lower 63/64ths (63 sectors: 0 to 62)
0	1	0	Upper 32nd (2 sectors: 62 to 63)	Lower 31/32nds (62 sectors: 0 to 61)
0	1	1	Upper 16th (4 sectors: 60 to 63)	Lower 15/16ths (60 sectors: 0 to 59)
1	0	0	Upper 8th (8 sectors: 56 to 63)	Lower seven-eighths (56 sectors: 0 to 55)
1	0	1	Upper quarter (16 sectors: 48 to 63)	Lower three-quarters (48 sectors: 0 to 47)
1	1	0	Upper half (32 sectors: 32 to 63)	Lower half (32 sectors: 0 to 31)
1	1	1	All sectors (64 sectors: 0 to 63)	None

You can read the status register at any time, even while a write or erase cycle is in progress. When one of these cycles is in progress, you can check the write in progress bit (bit 0 of the status register) before sending a new operation to the device. The device can also read the status register continuously, as shown in Figure 14–9.

Figure 14–9. Read Status Operation Timing Diagram



Write Status Operation

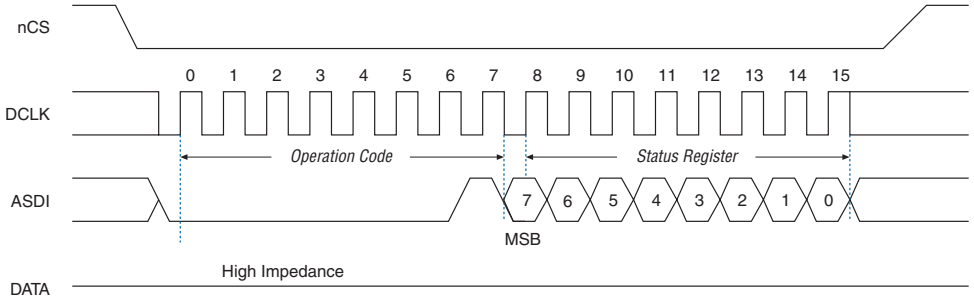
The write status operation code is b'0000 0001, with the MSB listed first. Use the write status operation to set the status register block protection bits. The write status operation has no effect on the other bits. Therefore, you can implement this operation to protect certain memory sectors, as defined in Table 14–16 through Table 14–20. After setting the block protect bits, the protected memory sectors are treated as read-only

memory. You must execute the write enable operation before the write status operation so the device sets the status register's write enable latch bit to 1.

The write status operation is implemented by driving nCS low, followed by shifting in the write status operation code and one data byte for the status register on the ASDI pin. Figure 14–10 shows the timing diagram for the write status operation. nCS must be driven high after the eighth bit of the data byte has been latched in, otherwise, the write status operation is not executed.

Immediately after nCS drives high, the device initiates the self-timed write status cycle. The self-timed write status cycle usually takes 5 ms for all serial configuration devices and is guaranteed to be less than 15 ms (refer to t_{WS} in Table 14–23). You must account for this delay to ensure that the status register is written with desired block protect bits. Alternatively, you can check the write in progress bit in the status register by executing the read status operation while the self-timed write status cycle is in progress. The write in progress bit is 1 during the self-timed write status cycle, and 0 when it is complete.

Figure 14–10. Write Status Operation Timing Diagram



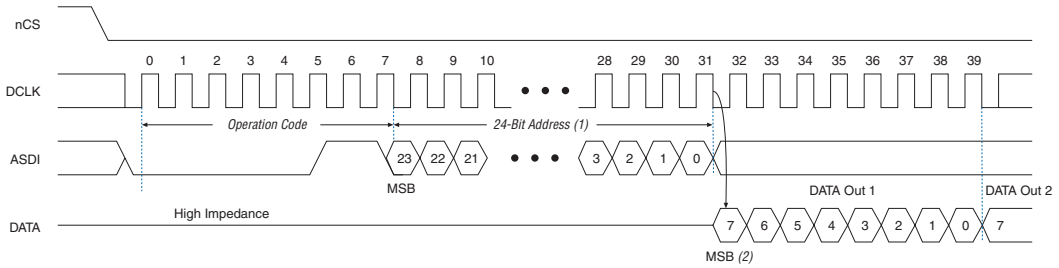
Read Bytes Operation

The read bytes operation code is b'0000 0011, with the MSB listed first. To read the memory contents of the serial configuration device, the device is first selected by driving nCS low. Then, the read bytes operation code is shifted in followed by a 3-byte address (A[23..0]). Each address bit must be latched in on the rising edge of the DCLK. After the address is latched in, the memory contents of the specified address are shifted out serially on the DATA pin, beginning with the MSB. For reading Raw Programming Data files (**.rpd**), the content is shifted out serially beginning with the LSB. Each data bit is shifted out on the falling edge of

DCLK. The maximum DCLK frequency during the read bytes operation is 20 MHz. Figure 14–11 shows the timing diagram for the read bytes operation.

The first byte address can be at any location. The device automatically increments the address to the next higher address after shifting out each byte of data. Therefore, the device can read the whole memory with a single read bytes operation. When the device reaches the highest address, the address counter restarts at 0x000000, allowing the memory contents to be read out indefinitely until the read bytes operation is terminated by driving nCS high. The device can drive nCS high any time after data is shifted out. If the read bytes operation is shifted in while a write or erase cycle is in progress, the operation is not executed and has no effect on the write or erase cycle in progress.

Figure 14–11. Read Bytes Operation Timing Diagram



Notes to Figure 14–11:

(1) Address bit A[23] is a don't-care bit in EPCS64. Address bits A[23..21] are don't-care bits in EPCS16. Address bits A[23..19] are don't-care bits in EPCS4. Address bits A[23..17] are don't-care bits in EPCS1.

(2) For RPD files, the read sequence shifts out the LSB of the data byte first.

Read Silicon ID Operation

The read silicon ID operation code is b'1010 1011, with the MSB listed first. Only EPCS1, EPCS4, EPCS16, and EPCS64 support this operation. It reads the serial configuration device's 8-bit silicon ID from the DATA output pin. If this operation is shifted in during an erase or write cycle, it is ignored and has no effect on the cycle that is in progress.

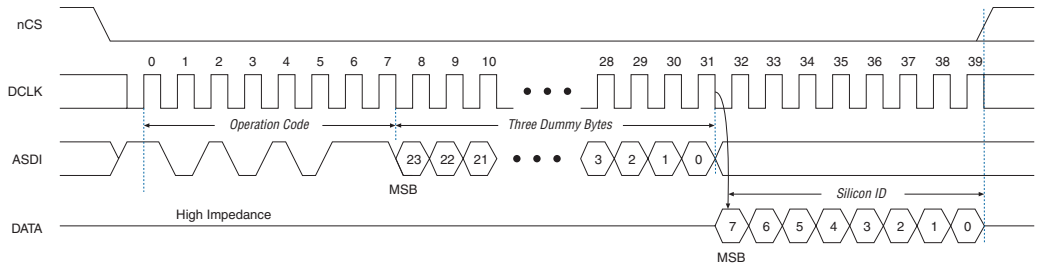
Table 14–21 shows the serial configuration device silicon IDs.

Table 14–21. Serial Configuration Device Silicon ID

Serial Configuration Device	Silicon ID (Binary Value)
EPCS1	b'0001 0000
EPCS4	b'0001 0010
EPCS16	b'0001 0100
EPCS64	b'0001 0110

The device implements the read silicon ID operation by driving nCS low then shifting in the read silicon ID operation code followed by three dummy bytes on ASDI. The serial configuration device's 8-bit silicon ID is then shifted out on the DATA pin on the falling edge of DCLK, as shown in Figure 14–12. The device can terminate the read silicon ID operation by driving nCS high after the silicon ID has been read at least once. Sending additional clock cycles on DCLK while nCS is driven low can cause the silicon ID to be shifted out repeatedly.

Figure 14–12. Read Silicon ID Operation Timing Diagram *Note (1)*



Note to Figure 14–12:
(1) Only EPCS1, EPCS4, EPCS16, and EPCS64 support Read Silicon ID operation.

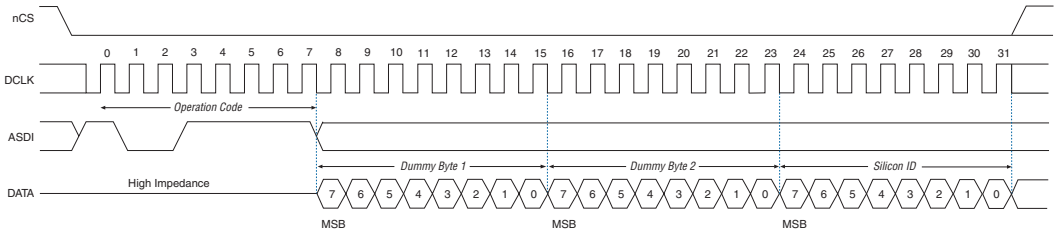
Read Device Identification Operation

The read device identification operation code is b'1001 1111, with the MSB listed first. Only EPCS128 supports this operation. It reads the serial configuration device's 8-bit device identification from the DATA output pin. If this operation is shifted in during an erase or write cycle, it is ignored and has no effect on the cycle that is in progress. Table 14–22 shows the serial configuration device identification.

Table 14–22. Serial Configuration Device Identification	
Serial Configuration Device	**Silicon ID (Binary Value)**
EPCS128	b'0001 1000

The device implements the read device identification operation by driving nCS low then shifting in the read device identification operation code followed by one dummy byte on ASDI. The serial configuration device's 16-bit device identification is then shifted out on the DATA pin on the falling edge of DCLK, as shown in Figure 14–13. The device can terminate the read device identification operation by driving nCS high after reading the device identification at least once.

Figure 14–13. Read Device Identification Operation Timing Diagram *Note (1)*



Note to Figure 14–13:
(1) Only EPCS128 supports read device identification operation.

Write Bytes Operation

The write bytes operation code is b'0000 0010, with the MSB listed first. The write bytes operation allows bytes to be written to the memory. The write enable operation must be executed prior to the write bytes operation to set the write enable latch bit in the status register to 1.

The write bytes operation is implemented by driving nCS low, followed by the write bytes operation code, three address bytes and a minimum one data byte on ASDI. If the eight least significant address bits (A[7..0]) are not all 0, all sent data that goes beyond the end of the current page is not written into the next page. Instead, this data is written at the start address of the same page (from the address whose eight LSBs are all 0). Drive nCS low during the entire write bytes operation sequence, as shown in Figure 14–14.

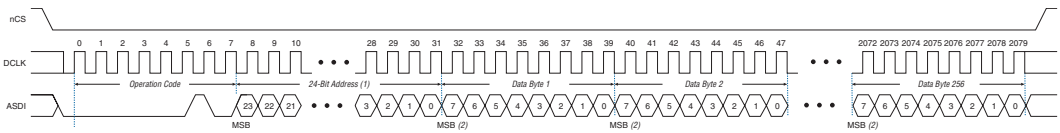
If more than 256 data bytes are shifted into the serial configuration device with a write bytes operation, the previously latched data is discarded and the last 256 bytes are written to the page. However, if less than 256 data bytes are shifted into the serial configuration device, they are guaranteed to be written at the specified addresses and the other bytes of the same page are unaffected.

If the design must write more than 256 data bytes to the memory, it needs more than one page of memory. Send the write enable and write bytes operation codes followed by three new targeted address bytes and 256 data bytes before a new page is written.

nCS must be driven high after the eighth bit of the last data byte has been latched in. Otherwise, the device will not execute the write bytes operation. The write enable latch bit in the status register is reset to 0 before the completion of each write bytes operation. Therefore, the write enable operation must be carried out before the next write bytes operation.

The device initiates the self-timed write cycle immediately after nCS is driven high. Refer to t_{WB} in Table 14–23 for the self-timed write cycle time for the respective EPCS devices. Therefore, you must account for this amount of delay before another page of memory is written. Alternatively, you can check the status register's write in progress bit by executing the read status operation while the self-timed write cycle is in progress. The write in progress bit is set to 1 during the self-timed write cycle, and 0 when it is complete.

Figure 14–14. Write Bytes Operation Timing Diagram



Notes to Figure 14–14:

(1) Address bit A[23] is a don't-care bit in EPCS64. Address bits A[23..21] are don't-care bits in EPCS16. Address bits A[23..19] are don't-care bits in EPCS4. Address bits A[23..17] are don't-care bits in EPCS1.

(2) For RPD files, write the LSB of the data byte first.

Erase Bulk Operation

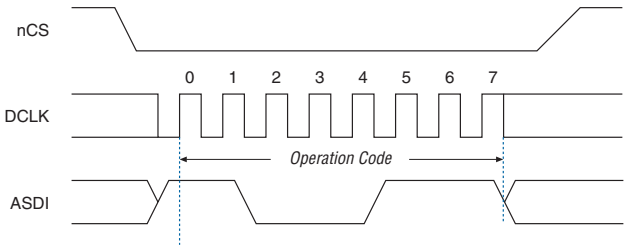
The erase bulk operation code is b'1100 0111, with the MSB listed first. The erase bulk operation sets all memory bits to 1 or 0xFF. Similar to the write bytes operation, the write enable operation must be executed prior to the erase bulk operation so that the write enable latch bit in the status register is set to 1.

You can implement the erase bulk operation by driving nCS low and then shifting in the erase bulk operation code on the ASDI pin. nCS must be driven high after the eighth bit of the erase bulk operation code has been latched in. Figure 14–15 shows the timing diagram.

The device initiates the self-timed erase bulk cycle immediately after nCS is driven high. Refer to t_{EB} in Table 14–23 for the self-timed erase bulk cycle time for the respective EPCS devices.

You must account for this delay before accessing the memory contents. Alternatively, you can check the write in progress bit in the status register by executing the read status operation while the self-timed erase cycle is in progress. The write in progress bit is 1 during the self-timed erase cycle and 0 when it is complete. The write enable latch bit in the status register is reset to 0 before the erase cycle is complete.

Figure 14–15. Erase Bulk Operation Timing Diagram



Erase Sector Operation

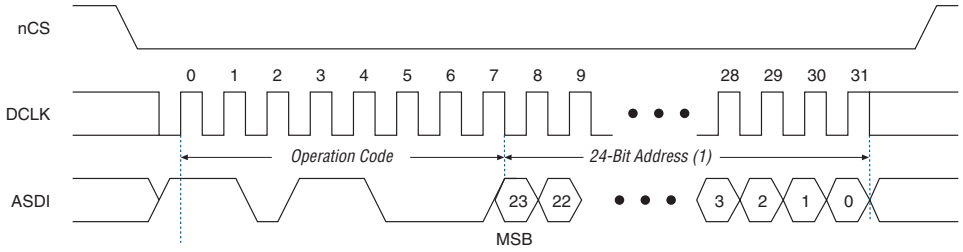
The erase sector operation code is b'1101 1000, with the MSB listed first. The erase sector operation allows the user to erase a certain sector in the serial configuration device by setting all bits inside the sector to 1 or 0xFF. This operation is useful for users who access the unused sectors as general purpose memory in their applications.

The write enable operation must be executed prior to the erase sector operation so that the write enable latch bit in the status register is set to 1.

The erase sector operation is implemented by first driving nCS low, then shifting in the erase sector operation code and the three address bytes of the chosen sector on the ASDI pin. The three address bytes for the erase sector operation can be any address inside the specified sector. (Refer to Tables 14–10 through 14–14 for sector address range information.) Drive nCS high after the eighth bit of the erase sector operation code has been latched in. Figure 14–16 shows the timing diagram.

Immediately after the device drives nCS high, the self-timed erase sector cycle is initiated. Refer to t_{ES} in Table 14–23 for the self-timed erase sector cycle time for the respective EPCS devices. You must account for this amount of delay before the memory contents can be accessed. Alternatively, you can check the write in progress bit in the status register by executing the read status operation while the erase cycle is in progress. The write in progress bit is 1 during the self-timed erase cycle and 0 when it is complete. The write enable latch bit in the status register resets to 0 before the erase cycle is complete.

Figure 14–16. Erase Sector Operation Timing Diagram



Note to Figure 14–16:

(1) Address bit A[23] is a don't-care bit in EPCS64. Address bits A[23..21] are don't-care bits in EPCS16. Address bits A[23..19] are don't-care bits in EPCS4. Address bits A[23..17] are don't-care bits in EPCS1.

Power and Operation

This section describes the power modes, power-on reset (POR) delay, error detection, and initial programming state of serial configuration devices.

Power Mode

Serial configuration devices support active power and standby power modes. When nCS is low, the device is enabled and is in active power mode. The FPGA is configured while in active power mode. When nCS is high, the device is disabled but could remain in active power mode until all internal cycles have completed (such as write or erase operations). The serial configuration device then goes into stand-by power mode. The I_{CC1} parameter specifies the V_{CC} supply current when the device is in active power mode and the I_{CC0} parameter specifies the current when the device is in stand-by power mode (refer to Table 14–29).

Power-On Reset

During initial power-up, a POR delay occurs to ensure the system voltage levels have stabilized. During AS configuration, the FPGA controls the configuration and has a longer POR delay than the serial configuration device.



For the POR delay time, refer to the configuration chapter in the appropriate device handbook.

Error Detection

During AS configuration with the serial configuration device, the FPGA monitors the configuration status through the nSTATUS and CONF_DONE pins. If an error condition occurs (nSTATUS drives low) or if the CONF_DONE pin does not go high, the FPGA will initiate reconfiguration by pulsing the nSTATUS and nCSO signals, which controls the chip select pin on the serial configuration device (nCS).

After an error, configuration automatically restarts if the **Auto-Restart Upon Frame Error** option is turned on in the Quartus II software. If the option is turned off, the system must monitor the nSTATUS signal for errors and then pulse the nCONFIG signal low to restart configuration.

Timing Information

Figure 14–17 shows the timing waveform for write operation to the serial configuration device.

Figure 14–17. Write Operation Timing

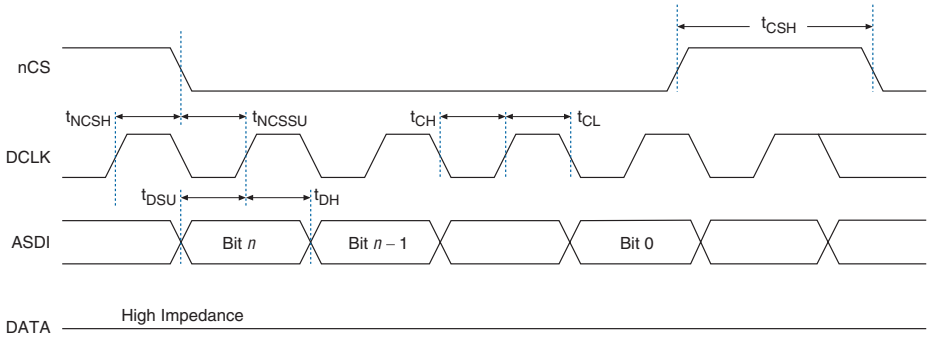


Table 14–23 defines the serial configuration device timing parameters for write operation.

Table 14–23. Write Operation Parameters (Part 1 of 2)

Symbol	Parameter	Min	Typ	Max	Unit
f_{WCLK}	Write clock frequency (from FPGA, download cable, or embedded processor) for write enable, write disable, read status, read silicon ID, write bytes, erase bulk, and erase sector operations	—	—	25	MHz
t_{CH}	DCLK high time	20	—	—	ns
t_{CL}	DCLK low time	20	—	—	ns
t_{NCSSU}	Chip select (nCS) setup time	10	—	—	ns
t_{NCSH}	Chip select (nCS) hold time	10	—	—	ns
t_{DSU}	Data (ASDI) in setup time before rising edge on DCLK	5	—	—	ns
t_{DH}	Data (ASDI) hold time after rising edge on DCLK	5	—	—	ns
t_{CSH}	Chip select high time	100	—	—	ns
t_{WB} *(1)*	Write bytes cycle time for EPCS1, EPCS4, EPCS16, and EPCS64	—	1.5	5	ms
	Write bytes cycle time for EPCS128	—	2.5	7	ms
t_{WS} *(1)*	Write status cycle time	—	5	15	ms

Table 14–23. Write Operation Parameters (Part 2 of 2)					
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Unit**
t_{EB} *(1)*	Erase bulk cycle time for EPCS1	—	3	6	s
	Erase bulk cycle time for EPCS4	—	5	10	s
	Erase bulk cycle time for EPCS16	—	17	40	s
	Erase bulk cycle time for EPCS64	—	68	160	s
	Erase bulk cycle time for EPCS128	—	105	250	s
t_{ES} *(1)*	Erase sector cycle time for EPCS1, EPCS4, EPCS16, and EPCS64	—	2	3	s
	Erase sector cycle time for EPCS128	—	2	6	s

Note to Table 14–23:
(1) These parameters are not shown in Figure 14–17.

Figure 14–18 shows the timing waveform for the serial configuration device's read operation.

Figure 14–18. Read Operation Timing

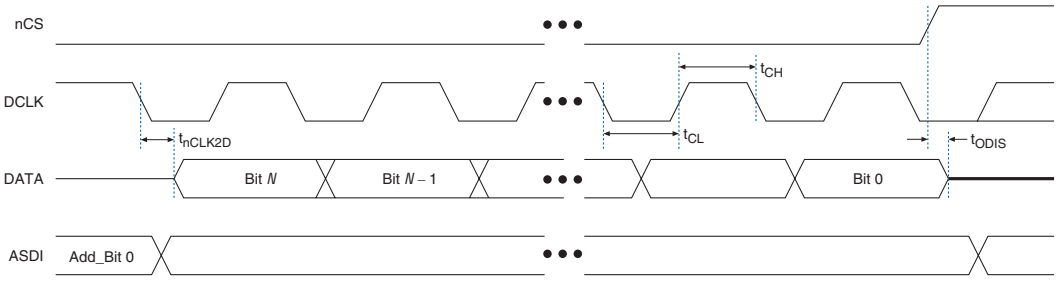


Table 14–24 defines the serial configuration device timing parameters for read operation.

Table 14–24. Read Operation Parameters (Part 1 of 2)				
Symbol	**Parameter**	**Min**	**Max**	**Unit**
f_{RCLK}	Read clock frequency (from FPGA or embedded processor) for read bytes operation	—	20	MHz
t_{CH}	DCLK high time	25	—	ns
t_{CL}	DCLK low time	25	—	ns

Table 14–24. Read Operation Parameters (Part 2 of 2)				
Symbol	**Parameter**	**Min**	**Max**	**Unit**
t_{ODIS}	Output disable time after read	—	15	ns
t_{nCLK2D}	Clock falling edge to data	—	15	ns

Figure 14–19 shows the timing waveform for FPGA AS configuration scheme using a serial configuration device.

Figure 14–19. AS Configuration Timing

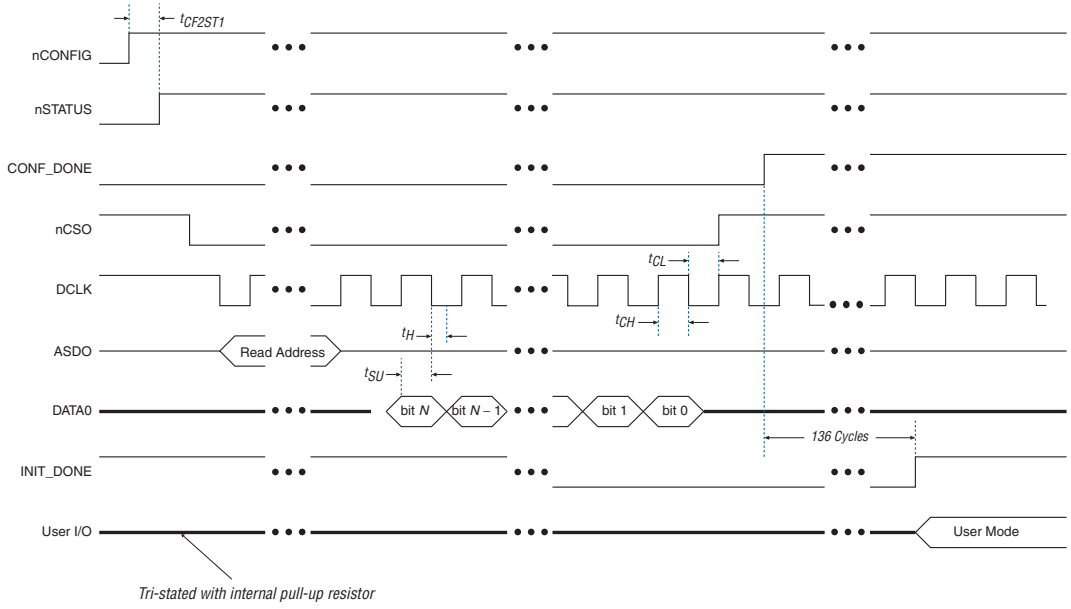


Table 14–25 shows the timing parameters for AS configuration mode.

Table 14–25. Timing Parameters for AS Configuration

Symbol	Parameter	Min	Typ	Max	Unit
f_{CLK}	DCLK frequency from Cyclone FPGA	14	17	20	MHz
	DCLK frequency from Stratix II or Cyclone II FPGA (40 MHz) *(1)*	20	26	40	MHz
	DCLK frequency from Stratix II or Cyclone II FPGA (20 MHz)	10	13	20	MHz
	DCLK frequency from Cyclone III FPGA *(1)*	20	30	40	MHz
	DCLK frequency from Stratix III FPGA *(1)*	15	25	40	MHz
t_H	Data hold time after rising edge on DCLK	0	—	—	ns
t_{SU}	Data set up time before rising edge on DCLK	5	—	—	ns

Note to Table 14–25:

(1) Existing batches of EPCS1 and EPCS4 manufactured on 0.15 µm process geometry supports AS configuration up to 40 MHz. However, batches of EPCS1 and EPCS4 manufactured on 0.18 µm process geometry support only up to 20 MHz. EPCS16, EPCS64, and EPCS128 are not affected. For information about product traceability and transition date to differentiate between 0.15 µm process geometry and 0.18 µm process geometry EPCS1 and EPCS4, refer to PCN 0514 Manufacturing Changes on EPCS Family process change notification on the Altera website at www.altera.com.

Programming and Configuration File Support

The Quartus II design software provides programming support for serial configuration devices. After selecting the serial configuration device, the Quartus II software automatically generates the Programmer Object File (**.pof**) to program the device. The software allows users to select the appropriate serial configuration device density that most efficiently stores the configuration data for a selected FPGA.

The serial configuration device can be programmed in-system by an external microprocessor using SRunner. SRunner is a software driver developed for embedded serial configuration device programming that designers can customize to fit in different embedded systems. The SRunner can read RPD file and write to the serial configuration devices. The programming time is comparable to the Quartus II software programming time. Note that writing and reading the RPD file to the EPCS is different from other data and address bytes. The LSB of RPD bytes must be shifted out first during the read bytes instruction and the LSB of RPD bytes must be shifted in first during the write bytes instruction. This is because the FPGA reads the LSB of the RPD data first during the configuration process.



For more information about SRunner, refer to the *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming User Guide* and the source code on the Altera website (www.altera.com).

Serial configuration devices can be programmed using the APU with the appropriate programming adapter (PLMSEPC-8) via the Quartus II software, USB Blaster, EthernetBlaster, or the ByteBlaster II download cable via the Quartus II software. In addition, many third-party programmers, such as BP Microsystems and System General, offer programming hardware that supports serial configuration devices.

During in-system programming of a serial configuration device via the USB Blaster, EthernetBlaster, or ByteBlaster II download cable, the cable pulls nCONFIG low to reset the FPGA and overrides the 10-kΩ pull-down resistor on the FPGA's nCE pin (refer to Figure 14–2). The download cable then uses the four interface pins (DATA, nCS, ASDI, and DCLK) to program the serial configuration device. Once the programming is complete, the download cable releases the serial configuration device's four interface pins and the FPGA's nCE pin, and pulses nCONFIG to start configuration.

The FPGA can program the serial configuration device in-system using the JTAG interface with the Serial FlashLoader. This solution allows you to indirectly program the serial configuration device using the same JTAG interface that is used to configure the FPGA.

For more information about the Serial FlashLoader, refer to *AN 370: Using the Serial FlashLoader with the Quartus II Software*.

For more information on programming and configuration support, refer to the following documents:

- *Altera Programming Hardware Data Sheet*
- *Programming Hardware Manufacturers*
- *USB-Blaster Download Cable User Guide*
- *ByteBlaster II Download Cable User Guide*
- *EthernetBlaster Communications Cable User Guide*

Operating Conditions

Tables 14–26 through 14–30 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for serial configuration devices.

Table 14–26. Absolute Maximum Ratings *Note (1)* *(Part 1 of 2)*

Symbol	Parameter	Condition	Min	Max	Unit
V_{CC}	Supply voltage for EPCS1, EPCS4, and EPCS16	With respect to ground	–0.6	4.0	V
	Supply voltage for EPCS64 and EPCS128	With respect to ground	–0.2	4.0	V

Table 14–26. Absolute Maximum Ratings *Note (1)* *(Part 2 of 2)*

Symbol	Parameter	Condition	Min	Max	Unit
V_I	DC input voltage for EPCS1, EPCS4, and EPCS16	With respect to ground	–0.6	4.0	V
	DC input voltage for EPCS64 and EPCS128	With respect to ground	–0.5	4.0	V
I_{MAX}	DC V_{CC} or GND current	—	—	15	mA
I_{OUT}	DC output current per pin	—	–25	25	mA
P_D	Power dissipation	—	—	54	mW
T_{STG}	Storage temperature	No bias	–65	150	°C
T_{AMB}	Ambient temperature	Under bias	–65	135	°C
T_J	Junction temperature	Under bias	—	135	°C

Table 14–27. Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CC}	Supply voltage	*(2)*	2.7	3.6	V
V_I	Input voltage	Respect to GND	–0.3	0.3 + V_{CC}	V
V_O	Output voltage	—	0	V_{CC}	V
T_A	Operating temperature	For commercial use	0	70	°C
		For industrial use	–40	85	°C
t_R	Input rise time	—	—	5	ns
t_F	Input fall time	—	—	5	ns

Table 14–28. DC Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{IH}	High-level input voltage for EPCS1, EPCS4, and EPCS16	—	0.6 × V_{CC}	V_{CC} + 0.4	V
	High-level input voltage for EPCS64 and EPCS128	—	0.6 × V_{CC}	V_{CC} + 0.2	V
V_{IL}	Low-level input voltage	—	–0.5	0.3 × V_{CC}	V
V_{OH}	High-level output voltage	I_{OH} = –100 µA *(3)*	V_{CC} – 0.2	—	V
V_{OL}	Low-level output voltage	I_{OL} = 1.6 mA *(3)*	—	0.4	V
I_I	Input leakage current	V_I = V_{CC} or GND	–10	10	µA
I_{OZ}	Tri-state output off-state current	V_O = V_{CC} or GND	–10	10	µA

Table 14–29. I_{CC} Supply Current

Symbol	Parameter	Conditions	Min	Max	Unit
I_{CC0}	V_{CC} supply current (standby) for EPCS1, EPCS4, and EPCS16	—	—	50	µA
	V_{CC} supply current (standby) for EPCS64 and EPCS128	—	—	100	µA
I_{CC1}	V_{CC} supply current (during active power mode) for EPCS1, EPCS4, and EPCS16	—	5	15	mA
	V_{CC} supply current (during active power mode) for EPCS64 and EPCS128	—	5	20	mA

Table 14–30. Capacitance *Note (4)*

Symbol	Parameter	Conditions	Min	Max	Unit
C_{IN}	Input pin capacitance	V_{IN} = 0 V	—	6	pF
C_{OUT}	Output pin capacitance	V_{OUT} = 0 V	—	8	pF

Notes to Table 14–26 through 14–30:

(1) Refer to the *Operating Requirements for Altera Devices Data Sheet*.

(2) Maximum V_{CC} rise time is 100 ms.

(3) The I_{OH} parameter refers to high-level TTL or CMOS output current; the I_{OL} parameter refers to low-level TTL or CMOS output current.

(4) Capacitance is sample-tested only at T_A = 25 °C and at a 20-MHz frequency.

Pin Information

As shown in Figures 14–20 and 14–21, the serial configuration device is an 8-pin or 16-pin device. The control pins on the serial configuration device are: serial data output (DATA), active serial data input (ASDI), serial clock (DCLK), and chip select (nCS). Table 14–31 shows the serial configuration device's pin descriptions.

Figure 14–20 shows the Altera serial configuration device 8-pin SOIC package and its pin-out diagram.

Figure 14–20. Altera Serial Configuration Device 8-Pin SOIC Package Pin-Out Diagram

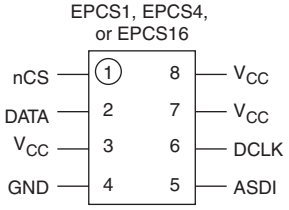
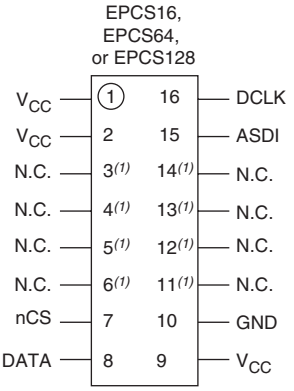


Figure 14–21 shows the Altera serial configuration device 16-pin SOIC package and its pin-out diagram.

Figure 14–21. Altera Serial Configuration Device 16-Pin SOIC Package Pin-Out Diagram



Note to Figure 14–21:

(1) These pins can be left floating or connected to V_{CC} or GND, whichever is more convenient on the board.

Table 14–31. Serial Configuration Device Pin Description

Pin Name	Pin Number in 8-Pin SOIC Package	Pin Number in 16-Pin SOIC Package	Pin Type	Description
DATA	2	8	Output	The DATA output signal transfers data serially out of the serial configuration device to the FPGA during read/configuration operation. During a read/configuration operations, the serial configuration device is enabled by pulling nCS low. The DATA signal transitions on the falling edge of DCLK.
ASDI	5	15	Input	The AS data input signal is used to transfer data serially into the serial configuration device. It receives the data that should be programmed into the serial configuration device. Data is latched on the rising edge of DCLK.
nCS	1	7	Input	The active low chip select input signal toggles at the beginning and end of a valid instruction. When this signal is high, the device is deselected and the DATA pin is tri-stated. When this signal is low, it enables the device and puts the device in an active mode. After power up, the serial configuration device requires a falling edge on the nCS signal before beginning any operation.
DCLK	6	16	Input	DCLK is provided by the FPGA. This signal provides the timing of the serial interface. The data presented on ASDI is latched to the serial configuration device on the falling edge of DCLK. Data on the DATA pin changes after the falling edge of DCLK and is latched into the FPGA on the falling edge.
V_{CC}	3, 7, 8	1, 2, 9	Power	Power pins connect to 3.3 V.
GND	4	10	Ground	Ground pin.

Package

All serial configuration devices are available in 8-pin or 16-pin plastic SOIC package.

For more information on Altera device packaging including mechanical drawing and specifications for this package, refer to the *Altera Device Package Information Data Sheet*.

Ordering Code

Table 14–32 shows the ordering codes for serial configuration devices.

Table 14–32. Serial Configuration Device Ordering Codes

Device	Ordering Code *(1)*
EPCS1	EPCS1SI8 EPCS1SI8N
EPCS4	EPCS4SI8 EPCS4SI8N
EPCS16	EPCS16SI16N EPCS16SI8N
EPCS64	EPCS64SI16N
EPCS128	EPCS128SI16N

Note to Table 14–32:
(1) N: Lead free.

Referenced Documents

This chapter references the following documents:

- *Active Serial Memory Interface Data Sheet*
- *Altera Device Package Information Data Sheet*
- *Altera Programming Hardware Data Sheet*
- *AN 370: Using the Serial FlashLoader with the Quartus II Software*
- *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming User Guide*
- *ByteBlaster II Download Cable User Guide*
- *Configuring Cyclone III Devices* chapter in volume 1 of the *Cyclone III Device Handbook*
- *EthernetBlaster Communications Cable User Guide*
- *Operating Requirements for Altera Devices Data Sheet*
- *Programming Hardware Manufacturers*
- *USB-Blaster Download Cable User Guide*

Document Revision History

Table 14–33 shows the revision history for this chapter.

Table 14–33. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
May 2008 v3.1	• Updated Tables 14–2, 14–5, 14–6, 14–27, and 14–28. • Deleted Note 5 to Table 14–30. • Added "Referenced Documents" section.	—
August 2007 v3.0	• Updated "Introduction" section. • Updated "Functional Description" section. • Updated Tables 14–1 through 14–3 and Tables 14–6 through 14–8 to with EPCS128 information. • Added Table 14–5 on Arria GX. • Added *Note (4)* to Figure 14–3. • Added *Note (5)* to Figure 14–4. • Updated Table 14–9 with EPCS128 information. • Added new Table 14–10 on address range for sectors in EPCS128. • Updated Table 14–15 with information on "Read Device Identification" and added *Note (5)*. • Added new Table 14–20 on block protection bits in EPCS128. • Added *Note (1)* to Figure 14–12. • Added new section "Read Device Identification Operation" with Table 14–22 and Figure 14–13. • Updated "Write Bytes Operation", "Erase Bulk Operation" and "Erase Sector Operation" sections. • Updated Table 14–23 to include EPCS128 information. • Updated *Note (1)* to Table 14–25. • Updated V_{CC} and V_I information to include EPCS128 in Table 14–26. • Updated V_{IH} information to include EPCS128 in Table 14–28. • Updated I_{CC0} and I_{CC1} information to include EPCS128 in Table 14–29. • Updated Figure 14–21 and Table 14–32 with EPCS128 information.	• Updated document to include EPCS128. • Updated document to include Arria GX.
April 2007 v2.0	• Updated "Introduction" section. • Updated "Functional Description" section and added handpara note. • Added Tables 14–3, 14–5, and 14–6. • Updated "Active Serial FPGA Configuration" section and its handpara note. • Added *Note (4)* to Figure 14–2. • Updated Table 14–25 and added *Note (1)*. • Updated Figure 14–20. • Updated Table 14–32.	• Updated chapter to include Stratix II GX, Stratix III, and Cyclone III support for EPCS devices. • Added information about EPCS16SI8N.
January 2007 v1.7	• Removed reference to PLMSEPC-16 in "Programming and Configuration File Support". • Updated DCLK pin information in Table 14–31.	—

Table 14–33. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2006 v1.6	• Updated Figure 14–19. • Updated Table 14–29 and Table 14–31.	—
August 2005 v1.5	Updated table 4-4 to include EPCS64 support for Cyclone devices.	—
August 2005 v1.4	• Updated tables. • Minor text updates.	—
February 2005 v1.3	Updated hot socketing AC specifications.	—
October 2003 v1.2	• Added Serial Configuration Device Memory Access section. • Updated timing information in Tables 4–10 and 4–11.section. • Updated timing information in Tables 4-16 and 4-17.	—
July 2003 v1.1	Minor updates.	—
May 2003 v1.0	Added document to the *Cyclone Device Handbook.*	—

Section VII. Cyclone Device Package Information

This section provides information for board layout designers to successfully layout their boards for Cyclone devices. It contains the required PCB layout guidelines, device pin tables, and package specifications.

This section includes the following chapter:

- Chapter 15. Package Information for Cyclone Devices

Revision History

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

15. Package Information for Cyclone Devices

C52006-1.3

Introduction

This data sheet provides package information for Altera® devices. It includes the following sections:

- "Device and Package Cross Reference" on page 15–1
- "Thermal Resistance" on page 15–2
- "Package Outlines" on page 15–2

In this data sheet, packages are listed in the order of ascending pin count.

Device and Package Cross Reference

Table 15–1 shows which Altera Cyclone® devices are available in FineLine BGA packages.

Table 15–1. Cyclone Devices in FineLine BGA Packages

Device	Package	Pins
EP1C4	Non-Thermally Enhanced FineLine BGA	324
	Non-Thermally Enhanced FineLine BGA	400
EP1C6	Non-Thermally Enhanced FineLine BGA	256
EP1C12	Non-Thermally Enhanced FineLine BGA	256
	Non-Thermally Enhanced FineLine BGA	324
EP1C20	Non-Thermally Enhanced FineLine BGA	324
	Non-Thermally Enhanced FineLine BGA	400

Thermal Resistance

Table 15–2 provides θ_{JA} (junction-to-ambient thermal resistance) and θ_{JC} (junction-to-case thermal resistance) values for Altera Cyclone devices.

Table 15–2. Thermal Resistance of Cyclone Devices *Notes (1), (2)*

Device	Pin Count	Package	θ_{JC} (° C/W)	θ_{JA} (° C/W) Still Air	θ_{JA} (° C/W) 100 ft./min.	θ_{JA} (° C/W) 200 ft./min.	θ_{JA} (° C/W) 400 ft./min.
EP1C3	100	TQFP	11.0	37.5	35.4	33.4	29.8
	144	TQFP	10.0	31.1	29.4	27.9	25.5
EP1C4	324	FineLine BGA	8.3	28.5	24.4	22.1	20.3
	400	FineLine BGA	7.9	20.7	17.5	15.5	13.9
EP1C6	144	TQFP	9.8	29.4	28.0	26.7	24.7
	240	PQFP	4.3	27.2	24.7	22.1	17.8
	256	FineLine BGA	8.8	28.7	24.5	22.3	20.5
EP1C12	240	PQFP	4.0	26.0	23.4	20.8	17.1
	256	FineLine BGA	6.6	24.3	20.2	18.1	16.4
	324	FineLine BGA	6.1	23.0	19.8	17.7	16.1
EP1C20	324	FineLine BGA	5.0	21.0	17.7	15.6	14.1
	400	FineLine BGA	4.7	20.7	17.5	15.5	13.9

Notes to Table 15–2:
(1) TQFP: thin quad flat pack
(2) PQFP: plastic quad flat pack

Package Outlines

The package outlines on the following pages are listed in order of ascending pin count. Altera package outlines meet the requirements of *JEDEC Publication No. 95*.

Document Revision History

Table 15–3 shows the revision history for this chapter.

Table 15–3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.3	Minor changes to format.	—
January 2007 v1.2	Added document revision history.	—