HardCopy Series Handbook, Volume 1

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com

H5V1-4.1

Contents

Chapter 5. Quartus II Support for HardCopy II Devices

Chapter 6. Script-Based Design for HardCopy II Devices

Chapter 7. Timing Constraints for HardCopy II

Chapter 8. Migrating Stratix II Device Resources to HardCopy II Devices

Section II. HardCopy Stratix Device Family Data Sheet

Chapter 9. Introduction to HardCopy Stratix Devices

Chapter 10. Description, Architecture & Features

Chapter 11. Boundary-Scan Support

Chapter 12. Operating Conditions

Chapter 13. Quartus II Support for HardCopy Stratix Devices

Chapter 14. Design Guidelines for HardCopy Stratix Performance Improvement

Chapter 20. Power-Up Modes & Configuration Emulation in HardCopy Series Devices

Section V. HardCopy Design Center Migration Process

Chapter 21. Back-End Design Flow for HardCopy Series Devices

Chapter 22. Back-End Timing Closure for HardCopy Series Devices

Chapter Revision Dates

The chapters in this book, *HardCopy Series 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 to HardCopy II Devices
Revised: *March 2006*
Part number: *H51015-2.3*

Chapter 2. Description, Architecture & Features
Revised: *March 2006*
Part number: *H51016-2.2*

Chapter 3. Boundary-Scan Support
Revised: *October 2005*
Part number: *H51017-2.1*

Chapter 4. Operating Conditions
Revised: *October 2005*
Part number: *H51018-2.1*

Chapter 5. Quartus II Support for HardCopy II Devices
Revised: *May 2006*
Part number: *H51022-2.2*

Chapter 6. Script-Based Design for HardCopy II Devices
Revised: *October 2005*
Part number: *H51025-1.0*

Chapter 7. Timing Constraints for HardCopy II
Revised: *March 2006*
Part number: *H51028-1.0*

Chapter 8. Migrating Stratix II Device Resources to HardCopy II Devices
Revised: *October 2005*
Part number: *H51024-1.1*

Chapter 9. Introduction to HardCopy Stratix Devices
Revised: *October 2005*
Part number: *H51001-2.1*

Chapter 10. Description, Architecture & Features
Revised: *October 2005*
Part number: *H51002-3.1*

Chapter 11. Boundary-Scan Support
Revised: *October 2005*
Part number: *H51004-3.1*

Chapter 12. Operating Conditions
Revised: *October 2005*
Part number: *H51005-3.1*

Chapter 13. Quartus II Support for HardCopy Stratix Devices
Revised: *October 2005*
Part number: *H51014-3.1*

Chapter 14. Design Guidelines for HardCopy Stratix Performance Improvement
Revised: *October 2005*
Part number: *H51027-1.1*

Chapter 15. Introduction to HardCopy APEX Devices
Revised: *January 2005*
Part number: *H51006-2.0*

Chapter 16. Description, Architecture & Features
Revised: *January 2005*
Part number: *H51007-2.0*

Chapter 17. Boundary-Scan Support
Revised: *January 2005*
Part number: *H51009-2.0*

Chapter 18. Operating Conditions
Revised: *February 2006*
Part number: *H51010-2.0*

Chapter 19. Design Guidelines for HardCopy Series Devices
Revised: *October 2005*
Part number: *H51011-3.1*

Chapter 20. Power-Up Modes & Configuration Emulation in HardCopy Series Devices
Revised: *May 2006*
Part number: *H51012-2.2*

Chapter 21. Back-End Design Flow for HardCopy Series Devices
Revised: *October 2005*
Part number: *H51019-1.1*

Chapter 22. Back-End Timing Closure for HardCopy Series Devices
Revised: *October 2005*
Part number: *H51013-2.1*

About this Handbook

This handbook provides comprehensive information about the Altera® HardCopy® devices.

How to Contact Altera

For the most up-to-date information about Altera products, go to the Altera world-wide web site at www.altera.com. For technical support on this product, go to www.altera.com/mysupport. For additional information about Altera products, consult the sources shown below.

Information Type	USA & Canada	All Other Locations
Technical support	www.altera.com/mysupport/	www.altera.com/mysupport/
	(800) 800-EPLD (3753) (7:00 a.m. to 5:00 p.m. Pacific Time)	+1 408-544-8767 7:00 a.m. to 5:00 p.m. (GMT -8:00) Pacific Time
Product literature	www.altera.com	www.altera.com
Altera literature services	literature@altera.com	literature@altera.com
Non-technical customer service	(800) 767-3753	+ 1 408-544-7000 7:00 a.m. to 5:00 p.m. (GMT -8:00) Pacific Time
FTP site	ftp.altera.com	ftp.altera.com

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.*

Visual Cue	Meaning
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.
CAUTION	The caution indicates required information that needs special consideration and understanding and should be read prior to starting or continuing with the procedure or process.
	The warning indicates information that should be read prior to starting or continuing the procedure or processes
↵	The angled arrow indicates you should press the Enter key.
	The feet direct you to more information on a particular topic.

Section I. HardCopy II Device Family Data Sheet

This section provides designers with the data sheet specifications for HardCopy® II devices. These 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 HardCopy II devices.

This section contains the following:

- Chapter 1, Introduction to HardCopy II Devices
- Chapter 2, Description, Architecture & Features
- Chapter 3, Boundary-Scan Support
- Chapter 4, Operating Conditions
- Chapter 5, Quartus II Support for Hardcopy II Devices
- Chapter 6, Script-Based Design for HardCopy II Devices
- Chapter 7, Hardcopy II Timing Constraints
- Chapter 8, Migrating Stratix II Device Resources to HardCopy II Devices

Revision History

The table below shows the revision history for Chapters 1 through 8.

Chapter(s)	Date / Version	Changes Made
Chapter 1	March 2006, v2.3	• Updated Table 1–1 and Table 1–3. • Minor edits and clarifications throughout.
	October 2005, v2.2.	Updated graphics
	July 2005, v2.2.	Updated graphics
	May 2005, v2.0	• Updated Table 1–1. • Updated migration process time. • Updated "Features" section.
	January 2005 v1.0	Added document to the HardCopy Series Handbook.
Chapter 2	March 2006, v2.2	• Updated Table 2–1, Table 2–9, Table 2–13. • Updated Figure 2–5 and Figure 2–5.
	October 2005, v2.1	Updated graphics.
	May 2005, v2.0	• Added Table 2–1. • Updated HCell information for DSP functions in the Functional Description section. • Updated Table 2–9. • Updated Figures 2–4, 2–5, and 2–6.
	January 2005, v1.0	Added document to the HardCopy Series Handbook.
Chapter 3	October 2005, v2.1	Updated graphics.
	May 2005, v2.0	Updated Table 3–2.
	January 2005 v1.0	Added document to the HardCopy Series Handbook.
Chapter 4	October 2005, v2.1	Updated graphics.
	May 2005, v2.0	Updated various tables throughout chapter.
	January 2005 v1.0	Added document to the HardCopy Series Handbook.

Chapter(s)	Date / Version	Changes Made
Chapter 5	May 2006, v2.2	Added information on support for HardCopy II devices in version 6.0 of the Quartus II software.
	March 2006	Formerly chapter 18; no content change.
	October 2005 v2.1	• Moved *Chapter 17 Quartus II Support for HardCopy II Devices* to Chapter 18 in *Hardcopy Series Device Handbook* 3.2. • Updated Graphics. • Updated technical content for Quartus II 5.1 support of HardCopy II devices.
	May 2005 v2.0	Added information on support for HardCopy II devices in version 5.0 of the Quartus II software.
	January 2005 v1.0	Added document to the *HardCopy Series Handbook*.
Chapter 6	March 2006	Formerly chapter 15; no content change.
	October 2005 v1.0	Initial release of *Script-Based Design for Hardcopy II Devices*.
Chapter 7	March 2006, v1.0	Added document to the HardCopy Series Handbook.
Chapter 8	March 2006	Formerly chapter 19; no content change.
	October 2005 v1.1	Minor edits
	May 2005 v1.0	Added document to the *HardCopy Series Handbook*.

1. Introduction to HardCopy II Devices

H51015-2.3

Introduction

HardCopy® II devices are low-cost, high-performance 1.2 V, 90 nm structured ASICs with pin outs, densities, and architecture that complement Stratix® II FPGAs. HardCopy II device features, such as PLLs, memory, and I/O elements (IOEs), are functionally and electrically equivalent to the Stratix II FPGA features. The combination of Stratix II FPGAs for in-system prototype and design verification, HardCopy II devices for high volume production, and the Quartus® II design software, provide a complete, low-risk structured ASIC solution.

Altera® HardCopy II devices use the same base arrays across multiple designs for a given device density and are customized using only two metal layers. Designers can use the Quartus II software to design with Stratix II FPGAs before migrating their design to a corresponding HardCopy II device. For typical designs, HardCopy II devices offer over 350-MHz system performance, more than 50% power reduction (dynamic and static), and up to 90% cost reduction compared to Stratix II FPGA prototypes.

The Quartus II software provides a complete set of tools for designing HardCopy II devices. Additionally, HardCopy II devices are also supported through other front-end design tools, such as tools from Synopsys, Synplicity, and Mentor Graphics.

Stratix II FPGA designs can be seamlessly and quickly migrated to HardCopy II devices, a low-cost ASIC alternative. HardCopy II devices improve on the successful and proven methodology of the two previous generations of HardCopy series devices. The migration process is fully automated and takes approximately 8 to 10 weeks from design submission to receipt of fully tested HardCopy II prototypes.

The HardCopy II device family consists of five devices. Table 1–1 summarizes the features available in the HardCopy II devices.

Table 1–1. HardCopy II Device Family Features

Feature	HC210W *(1)*	HC210	HC220	HC230	HC240
ASIC gates *(2)*	1,000,000	1,000,000	1,600,000	2,200,000	2,200,000
Additional gates for digital signal processing (DSP) block *(3)*	0	0	300,000	700,000	1,400,000
M4K RAM blocks (4 Kbits plus parity)	190	190	408	614	768 *(4)*
M-RAM blocks (512 Kbits plus parity)	0	0	2	6	9
Total RAM bits (including parity bits)	875,520	875,520	3,059,712	6,368,256	8,847,360
Enhanced PLLs	2	2	2	4	4
Fast PLLs	2	2	2	4	8
Maximum user I/O pins *(5)*, *(6)*	308	334	494	698	951

Notes to Table 1–1:

(1) HC210W devices are in a wire-bond package. All other HardCopy II devices and Stratix II FPGAs use a flip-chip package. Devices in a wire-bond package offer different performance and signal integrity characteristics compared to devices in a flip-chip package.

(2) This is the number of ASIC gates available in the HardCopy II base array for both logic and DSP functions that can be implemented in a Stratix II FPGA prototype.

(3) This number includes additional ASIC gates available for Stratix II DSP functions.

(4) Total number of usable M4K blocks is 768, which allows migration compatibility when prototyping with an EP2S180 device. This may be different from the Quartus II software total physical M4K count of the HC240.

(5) The I/O pin counts include the dedicated `CLK` input pins, which can be used for clock signals or data inputs.

(6) The Quartus II I/O pin counts include an additional pin (`PLLENA`), which is not available as a general-purpose I/O pin. The `PLLENA` pin can only be used to enable the PLLs.

HardCopy II devices offer pin-to-pin compatibility to the Stratix II prototype, which makes them drop-in replacements for the FPGAs. Therefore, the same system board and software developed for prototyping and field trials can be retained, enabling the fastest time-to-market for high-volume production. When migrating a specific Stratix II

FPGA to a HardCopy II device, there are a number of FPGA prototype choices, as shown in Table 1–2. Depending on the design resource needs, designers can choose an appropriate HardCopy II device.

Table 1–2. HardCopy II Options

HardCopy II Device	Stratix II Device				
	EP2S30	EP2S60	EP2S90	EP2S130	EP2S180
HC210W 484-pin FineLine BGA® *(1)*	✓	✓	✓		
HC210 484-pin FineLine BGA	✓	✓	✓		
HC220 672-pin FineLine BGA		✓			
HC220 780-pin FineLine BGA			✓	✓	
HC230 1,020-pin FineLine BGA			✓	✓	✓
HC240 1,020-pin FineLine BGA					✓
HC240 1,508-pin FineLine BGA					✓

Note to Table 1–2:
(1) The HC210W device uses a wire-bond package while the Stratix II FPGA prototype device uses a pin-compatible flip-chip package.

For more information on the migration path from Stratix II FPGAs to HardCopy II devices, refer to the *Prototyping Strategy for HardCopy II Devices* chapter in the *HardCopy Series Handbook*.

Features

HardCopy II structured ASICs are manufactured on a 1.2 V, 90 nm all-layer-copper metal fabrication process (up to nine layers of metal). HardCopy II devices offer the following features:

- Fine-grained architecture resulting in a low-cost, high-performance, low-power structured ASIC
- Customized using only two metal layers for fast turn-around times and low non-recurring expenses (NRE)
- Preserves the design functionality of a Stratix II FPGA prototype
- Pin-compatible with Stratix II FPGA prototypes
- Typical system performance of more than 350 MHz
- Over 50% power reduction (dynamic and static) for typical designs compared to Stratix II FPGA prototypes
- 1,000,000 to 3,600,00 usable gates for both logic and DSP functions
- Up to 9,068,544 RAM bits available, including parity bits
- Up to 951 user I/O pins available
- Memory blocks to implement true dual-port memory and first-in-first-out (FIFO) buffers

- Up to 16 global clocks with 24 clocking resources per device region
- Clock control block supports dynamic clock network enable/disable and dynamic global clock network source selection
- Up to 12 PLLs (four enhanced PLLs and eight fast PLLs) per device which provide identical features as the FPGA counterparts, including spread spectrum, programmable bandwidth, clock switchover, real-time PLL reconfiguration, advanced multiplication, and phase shifting
- Support for numerous single-ended and differential I/O standards such as LVTTL, LVCMOS, PCI, PCI-X, SSTL, HSTL, and LVDS
- High-speed differential I/O support on up to 116 channels with dynamic phase alignment (DPA) circuitry for 1-Gigabits-per-second (Gbps) performance
- Support for high-speed networking and communications bus standards including Parallel RapidIO, SPI-4 Phase 2 (POS-PHY Level 4), HyperTransport™ technology, and SFI-4
- Support for high-speed external memory, including DDR and DDR2 SDRAM, RLDRAM II, QDRII SRAM, and SDR SDRAM
- Support for multiple intellectual property (IP) megafunctions from Altera MegaCore® functions, and Altera Megafunction Partners Program (AMPP^{SM}) megafunctions
- HardCopy II devices are available in wire-bond and flip-chip space-saving FineLine BGA packages (Tables 1–3 and 1–4)
- Support for instant on and instant on after 50 ms power-up modes

The actual performance and power consumption improvements mentioned in this data sheet are design-dependent.

Table 1–3. HardCopy II Package Options & I/O Pin Counts *Notes (1), (2)*

Device	484-Pin FineLine BGA (3)	484-Pin FineLine BGA (4)	672-Pin FineLine BGA (4)	780-Pin FineLine BGA (4)	1,020-Pin FineLine BGA (4)	1,508-Pin FineLine BGA (4)
HC210W	308					
HC210		334				
HC220			492			
				494		
HC230					698	
HC240					742	
						951

Notes to Table 1–3:
(1) The Quartus II I/O pin counts include an additional pin (PLLENA) which is not available as a general-purpose I/O pin. The PLLENA pin can only be used to enable the PLLs.
(2) The I/O pin counts include the dedicated CLK input pins, which can be used for clock signals or data inputs.
(3) This is a wire-bond package.
(4) This is a flip-chip package.

Table 1–4. HardCopy II FineLine BGA Package Sizes

Dimension	484 Pin (1)	672 Pin	780 Pin	1,020 Pin	1,508 Pin
Pitch (mm)	1.00	1.00	1.00	1.00	1.00
Area (mm^2)	529	729	841	1,089	1,600
Length × width (mm × mm)	23 × 23	27 × 27	29 × 29	33 × 33	40 × 40

Note to Table 1–4:
(1) The EP2S90 FPGA prototype uses a 484-pin hybrid FineLine BGA package. For more information, refer to the *Stratix II FPGA Handbook*.

2. Description, Architecture & Features

H51016-2.2

Introduction

Altera® HardCopy® II devices feature an architecture that provides high density, high performance, and low power consumption suitable for a variety of applications. HardCopy II devices are low-cost structured ASICs with pin-outs, densities, and architecture that complement Stratix® II FPGAs. HardCopy II devices make optimal use of die area and core resources while offering features that are functionally equivalent to the Stratix II FPGA. The combination of Stratix II FPGAs for in-system prototype and design verification, HardCopy II devices for high volume production, and the Quartus® II design software, provide a complete, seamless path from prototype to volume production. Table 2–1 provides an overview of the HardCopy II device features.

Table 2–1. Hardcopy II Family Overview (Part 1 of 2)

Feature	HC210W *(1)*	HC210	HC220	HC230	HC240
ASIC gates *(2)*	1,000,000	1,000,000	1,600,000	2,200,000	2,200,000
Additional gates for digital signal processing (DSP) blocks *(3)*	0	0	300,000	700,000	1,400,000
M4K RAM blocks (4k bits plus parity)	190	190	408	614	768 *(4)*
M-RAM blocks (512k bits plus parity)	0	0	2	6	9
Total RAM bits (including parity bits)	875,520	875,520	3,059,712	6,368,256	8,847,360
Enhanced PLLs	2	2	2	4	4
Fast PLLs	2	2	2	4	8
Package (maximum user I/O pins) *(5)*, *(6)*	308	484-pin FineLine BGA (334)	672-pin FineLine BGA (492) 780-pin FineLine BGA (494)	1,020-pin FineLine BGA (698)	1,020-pin FineLine BGA (742) 1,508-pin FineLine BGA (951)

Table 2–1. Hardcopy II Family Overview (Part 2 of 2)

Feature	HC210W *(1)*	HC210	HC220	HC230	HC240
FPGA prototype options	EP2S30 EP2S60 EP2S90	EP2S30 EP2S60 EP2S90	EP2S60 EP2S90 EP2S130	EP2S90 EP2S130 EP2S180	EP2S180

Notes to Table 2–1:

(1) HC210W devices use a wire-bond package. All other HardCopy II devices and Stratix II FPGAs use a flip-chip package. Devices in a wire-bond package offer different performance and signal integrity characteristics compared to devices in a flip-chip package.

(2) This is the number of ASIC gates available in the HardCopy II base array for both logic and DSP functions that can be implemented in a Stratix II FPGA prototype.

(3) This number includes additional ASIC gates available for Stratix II DSP functions.

(4) Total number of usable M4K blocks is 768, which allows migration compatibility when prototyping with an EP2S180 device. This may be different from the Quartus II software total physical M4K count of the HC240.

(5) The I/O pin counts include the dedicated clock input pins, which can be used for clock signals or data inputs.

(6) The Quartus II I/O pin counts include an additional pin (`PLLENA`), which is not available as a general-purpose I/O pin. The `PLLENA` pin can only be used to enable the PLLs.

Functional Description

The Hardcopy II device family provides greater flexibility to design with FPGA prototypes before moving to structured ASICs for production. Before seamlessly migrating to the HardCopy II structured ASIC, the designer can prototype and test their design functionality using a Stratix II FPGA. There are multiple options for the prototype FPGA, allowing designers to choose the right HardCopy II device for volume production and maximum cost savings. The Quartus II design software includes features such as the Device Resource Guide, to help select the optimal HardCopy II device based on the design requirements.

For more information on the Device Resource Guide, refer to the *Quartus II Support for HardCopy II Devices* chapter in the *HardCopy Series Handbook*.

HardCopy II devices require minimal involvement from the designer in the device migration process. Additionally, unlike ASICs, the designer is not required to generate test benches, test vectors, or timing and functional simulations since prototyping is performed using an FPGA.

For more information on the migration path from Stratix II FPGAs to HardCopy II devices, refer to the *Prototyping Strategy for HardCopy II Using Stratix II FPGA* chapter in the *HardCopy Series Handbook*.

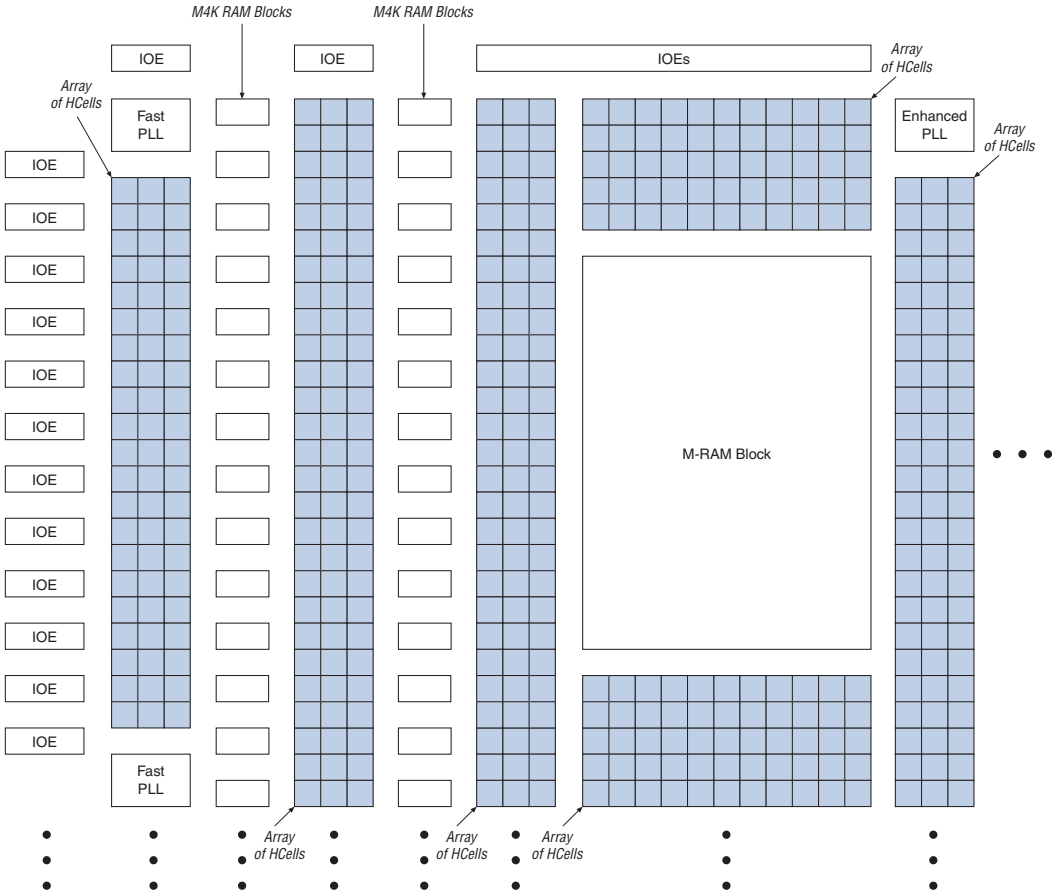
HardCopy II devices consist of base arrays that are common to all designs for a particular device density, with design-specific customization done using two metal layers. The reprogrammable FPGA logic, routing, memory, and FPGA configuration-related logic are stripped from

HardCopy II devices. Removing all programmable and configuration resources and replacing them with direct metal connections results in considerable die size reduction and cost savings. A fine-grain architecture consisting of an array of HCells extends the die reduction and cost savings, which results in low-cost structured ASICs with high performance and low power suitable for a wide variety of applications.

The SRAM configuration cells of the Stratix II FPGAs are replaced in HardCopy II devices with metal connections, which define the function of logic, memory, phase-locked loop (PLL), and I/O elements (IOEs) in the device. These resources are interconnected using metallization layers. Once a HardCopy II device is manufactured, the functionality of the device is fixed.

HardCopy II devices are manufactured using the same 90 nm process technology and operate using the same core voltage (1.2 V) as Stratix II FPGAs. Additionally, almost all architectural features in HardCopy II devices are functionally equivalent to features found in the Stratix II FPGA architecture. HardCopy II devices feature HCells, memory blocks, PLLs, and IOEs (Figure 2–1).

Figure 2–1. Example Block Diagram of HC230 Device *Note (1)*



Note to Figure 2–1:

(1) Figure 2–1 shows a graphical representation of the device floor plan. A detailed floor plan is available in the Quartus II software.

HardCopy II & Stratix II Similarities & Differences

HardCopy II devices preserve the functionality of Stratix II FPGAs. Implementation of these architectural features in HardCopy II structured ASICs matches Stratix II FPGA implementation, with a few exceptions. Table 2–2 shows a qualitative comparison of HardCopy II device feature

implementation versus Stratix II FPGA feature implementation. Other sections within this chapter provide details on similarities and differences of a particular HardCopy II feature.

Table 2–2. HardCopy II Device vs. Stratix II FPGA Feature Implementation

Feature	Equivalent	Different
Logic blocks		✓
DSP blocks		✓
Memory	✓	
Clock networks	✓	
PLLs	✓	
I/O features	✓	
Configuration *(1)*		✓

Note to Table 2–2:
(1) HardCopy II structured ASICs do not need to be configured upon power-up.

The major similarities and differences between Stratix II FPGAs and HardCopy II devices are highlighted below:

- HardCopy II devices consume less than 50% of the power of the equivalent Stratix II FPGAs operating at the same frequency. Power consumption is design dependent and is a direct result of design performance and resource utilization.
- HardCopy II devices offer up to 100% performance improvement when compared to Stratix II FPGA prototypes. The performance improvement is achieved by efficient use of logic blocks, metal interconnect optimization, die size reduction, and customized signal buffering.
- Logic blocks, known as HCells, are the basic building block of the core logic in HardCopy II devices and replace Stratix II adaptive logic modules (ALMs). HCells implement logic and DSP functions.
- DSP block functions are implemented using HCells, instead of dedicated DSP blocks.
- M4K and M-RAM memory blocks can implement various types of memory (the same as Stratix II FPGAs), with or without parity, including true dual-port, simple dual-port, and single-port RAM, ROM, and first-in first-out (FIFO) buffers.
- Unlike Stratix II FPGAs, the HardCopy II M4K block contents cannot be pre-loaded with a Memory Initialization File (**.mif**) when used as RAM. When used as ROM, HardCopy II M4K blocks are initialized to the ROM contents.

- When used as RAM, HardCopy II M4K and M-RAM blocks power up with outputs unknown. In Stratix II FPGAs, M4K blocks power up with outputs cleared, while M-RAM blocks power up with outputs unknown.
- All HardCopy II clock network features are the same as in Stratix II FPGAs.
- Enhanced PLL and fast PLL implementations in HardCopy II devices are the same as in Stratix II FPGAs.
- All Stratix II I/O features and supported I/O standards are offered in HardCopy II devices.
- The Joint Test Action Group (JTAG) boundary scan order and length in HardCopy II devices is different than that of the Stratix II FPGA. Use a HardCopy II boundary-scan description language (BSDL) file that describes the re-ordered and shortened boundary scan chain.
- Unlike Stratix II devices, HardCopy II devices are customized using two metal layers. Therefore, configuration circuitry is not required. FPGA configuration emulation and other configuration modes, including remote system upgrades and design security using configuration bitstream encryption, are not supported in HardCopy II devices.

Only supplementary information to highlight HardCopy II similarities and differences compared to the Stratix II FPGA architecture and functionality is provided in this chapter. Refer to the *Stratix II Device Handbook* for detailed explanations of architectural features and functions.

HCells

HardCopy II devices are built using an array of fine-grained architecture blocks called HCells. HCells are a collection of logic transistors based on 1.2 V, 90 nm process technology, similar to Stratix II devices. The construction of logic using HCells allows flexible functionality such that when HCells are combined, all viable logic combinations of Stratix II functionality are replicated. These HCells constitute the array of HCells area in Figure 2–1. Only HCells needed to implement the customer design are assembled together, which optimizes HCell utilization. The unused area of the HCell logic fabric is powered down, resulting in significant power savings compared with the Stratix II FPGA prototype.

The Quartus II software uses the library of pre-characterized HCell macros to place Stratix II ALM and DSP configurations into the HardCopy II HCell-based logic fabric. An HCell macro defines how a group of HCells are connected together within the array. HCell macros can construct all combinations of combinational logic, adder, and register functions that can be implemented by a Stratix II ALM. HCells not used for ALM configurations can be used to implement DSP block functions.

Based on design requirements, the Quartus II software will chose the appropriate HCell macros to implement the design functionality. For example, Stratix II ALMs offer flexible look-up table (LUT) blocks, registers, arithmetic blocks, and LAB-wide control signals. In HardCopy II devices, if the user's design requires these architectural elements, the Quartus II synthesis tool will map the design to the appropriate HCells, resulting in improved design performance compared to the Stratix II FPGA prototype.

Stratix II FPGAs have dedicated DSP blocks to implement various DSP functions. Stratix II DSP blocks consist of a multiplier block, an adder/subtractor/accumulator block, a summation block, input and output interfaces, and input and output registers. In HardCopy II devices, HCell macros implement Stratix II DSP block functionality with area efficiency and performance on par with the dedicated DSP blocks in Stratix II FPGAs.

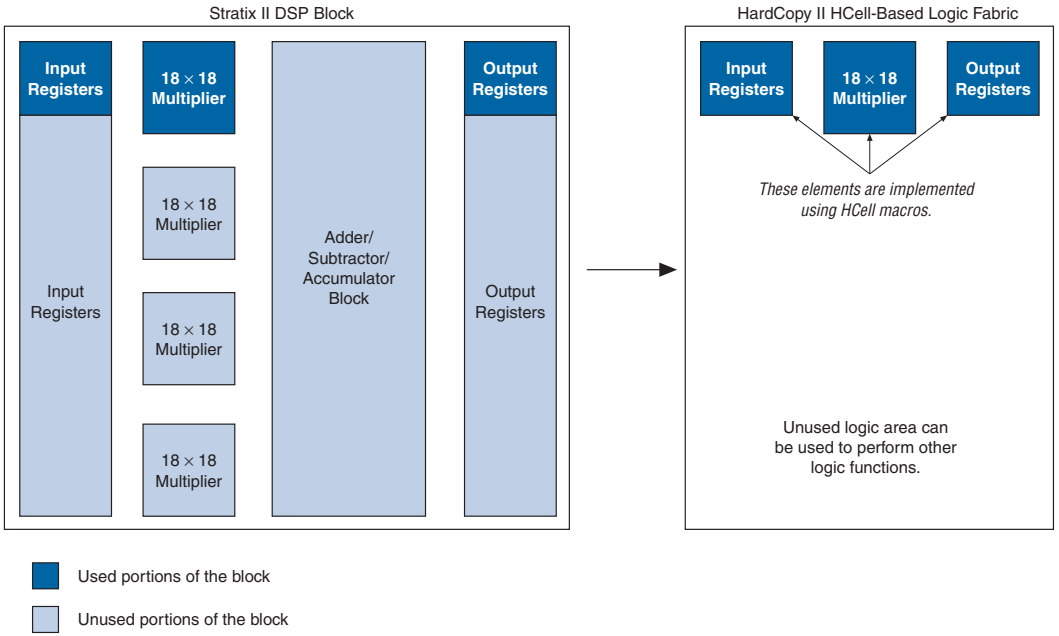
There are eight HCell macros which implement the eight supported modes of operation for the Stratix II DSP block:

- 9 × 9 multiplier
- 9 × 9 two-multiplier adder (9 × 9 complex multiply)
- 9 × 9 four-multiplier adder
- 18 × 18 multiplier
- 18 × 18 two-multiplier adder (18 × 18 complex multiply)
- 18 × 18 four-multiplier adder
- 52-bit (18 × 18) multiplier-accumulator
- 36 × 36 multiplier

Only HCells that are required to implement the design's DSP functions are enabled. HCells not needed for DSP functions can be used for ALM configurations, which results in efficient logic usage. In addition to area management, the placement of these HCell macros allows for optimized routing and performance.

An example of efficient logic area usage can be seen when comparing the 18 × 18 multiplier implementation in Stratix II FPGAs using the dedicated DSP block versus the implementation in HardCopy II devices using HCells. If the Stratix II DSP function only calls for one 18 × 18 multiplier, the other three 18 × 18 multipliers and the DSP block's adder output block are not used (Figure 2–2). In HardCopy II devices, the HCell-based logic fabric that is not used for DSP functions can be used to implement other combinational logic, adder, and register functions.

Figure 2–2. Stratix II DSP Block vs. HardCopy II HCell 18 × 18-Bit Multiplier Implementation



HardCopy II devices support all Stratix II DSP configurations (9 × 9, 18 × 18, and 36 × 36 multipliers) and all Stratix II DSP block features, such as dynamic sign controls, dynamic addition/subtraction, saturation, rounding, and dynamic input shift registers, except for dynamic mode switching.

Dynamic mode switching allows the designer to set up each Stratix II DSP block to dynamically switch between the following three modes:

- Up to four 18-bit independent multipliers
- Up to two 8-bit multiplier-accumulators
- One 36-bit multiplier

Each half of a Stratix II DSP block has separate mode control signals. Since DSP block functions are implemented in HardCopy II devices using HCells, HardCopy II devices do not support dynamic mode switching. If this feature is used, the Quartus II software flags the DSP implementation and does not allow you to migrate the design. The fitter reports that all HardCopy II devices are not compatible with the design. To migrate your Stratix II design to a HardCopy II companion device, disable dynamic switching in the DSP blocks.

For more information on the Stratix II DSP operational modes, refer to the *Stratix II Device Handbook*.

Embedded Memory

HardCopy II memory blocks 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. HardCopy II devices support the same memory functions and features as Stratix II FPGAs.

Functionally, the memory in both devices are identical. However, the number of available memory blocks differs based on density (Table 2–3).

Table 2–3. Hardcopy II Embedded Memory Resources

Feature	HC210W	HC210	HC220	HC230	HC240
M4K RAM blocks (4 Kbits)	190	190	408	614	768
M-RAM blocks (512 Kbits)	0	0	2	6	9
Total RAM bits (bits)	875,520	875,520	3,059,712	6,368,256	8,847,360

Since device functionality is fixed in HardCopy II devices, M4K block contents cannot be preloaded or initialized with a MIF when they are configured as RAM. When the M4K blocks are used as ROM, they will initialize to the design's ROM contents.

Unlike the Stratix II FPGA, the Hardcopy II M4K memory block power-up conditions behave like the M-RAM blocks. This means that all output registers of the memory blocks will have unknown output conditions. The designer must take this into consideration when designing logic that might evaluate the initial power-up values of the memory block.

HardCopy II embedded memory consists of M4K and M-RAM memory blocks and have a one-to-one mapping from Stratix II M4K and M-RAM resources. Table 2–4 shows the size and features of the different RAM blocks.

For more information on the Stratix II memory block features, refer to the *Stratix II Device Handbook*.

PLLs & Clock Networks

Both HardCopy II enhanced and fast PLLs are feature rich, supporting advanced capabilities such as clock switchover, reconfigurable phase shift, PLL reconfiguration, and reconfigurable bandwidth. PLLs are used for general-purpose clock management, supporting multiplication, division, phase shifting, and programmable duty cycle. In addition,

Table 2–4. HardCopy II Embedded Memory Features

Feature	M4K Blocks	M-RAM Blocks
Maximum performance *(1)*	350 MHz	350 MHz
Total RAM bits (including parity bits)	4,608	589,824
Configurations	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 8K × 64 8K × 72 4K × 128 4K × 144
Parity bits	✓	✓
Byte enable	✓	✓
Pack mode	✓	✓
Address clock enable	✓	✓
Single-port memory	✓	✓
Simple dual-port memory	✓	✓
True dual-port memory	✓	✓
Embedded shift register	✓	
ROM	✓	
FIFO buffer	✓	✓
Simple dual-port mixed width support	✓	✓
True dual-port mixed width support	✓	✓
Memory initialization file (**.mif**)		
Mixed-clock mode	✓	✓
Power-up condition	Outputs unknown	Outputs unknown
Register clears	Output registers only	Output registers only
Same-port read-during-write	New data available at positive clock edge	New data available at positive clock edge
Mixed-port read-during-write	Outputs set to unknown or old data	Unknown output

Note to Table 2–4:
(1) Maximum performance information is preliminary until device characterization.

enhanced PLLs support external clock feedback mode, spread-spectrum clocking, and counter cascading. Fast PLLs offer high speed outputs to manage the high-speed differential I/O interfaces.

All Stratix II PLL features are supported by HardCopy II PLLs.

Similar to Stratix II FPGAs, HardCopy II devices also support a power-down mode where unused clock networks can be disabled. HardCopy II and Stratix II clock control blocks support dynamic selection of the input clock from up to four possible sources, giving the designer the flexibility to choose from multiple (up to four) clock sources.

Enhanced & Fast PLLs

The number of PLLs available differs based on density (Table 2–5).

Table 2–5. HardCopy II PLLs

Feature	HC210W	HC210	HC220	HC230	HC240
Enhanced PLLs	2	2	2	4	4
Fast PLLs	2	2	2	4	8

The target HardCopy II device may not support the same number of enhanced PLLs as the prototyping Stratix II FPGA. However, since HardCopy II enhanced PLLs and fast PLLs offer a similar feature set (Table 2–7 on page 2–14), a fast PLL could be used in place of an enhanced PLL. The type of PLL used in the design should be chosen using the Quartus II software to accommodate the resources available in the HardCopy II device.

Table 2–6 shows which PLLs are available in each device density. Figure 2–3 shows the location of each PLL. During the prototyping stage using the FPGA, you must select the appropriate number of enhanced and fast PLLs that will be used in your HardCopy II device. Use Table 2–6 to ensure that the FPGA prototyping design uses the same PLL resources available in the HardCopy II device.

Table 2–6. HardCopy II PLLs Available *Note (1)*

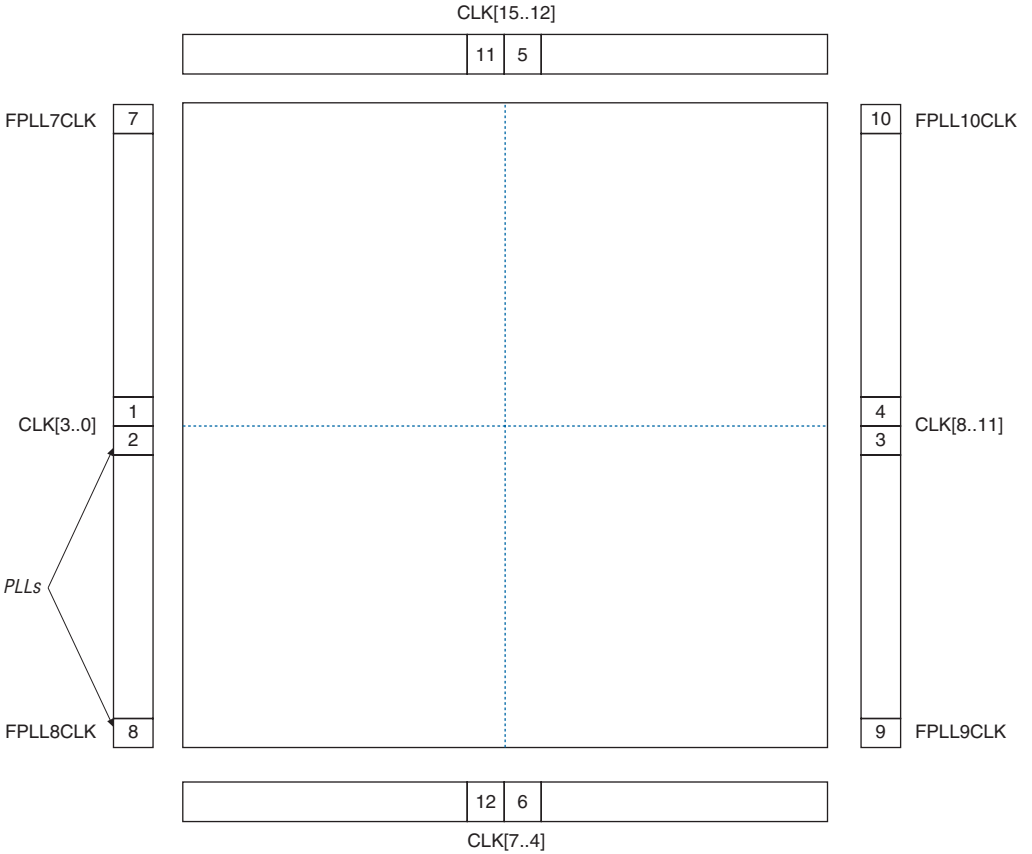
Device	Fast PLLs								Enhanced PLLs			
	1	2	3	4	7	8	9	10	5	6	11	12
HC210W	✓	✓							✓	✓		
HC210	✓	✓							✓	✓		

Table 2–6. HardCopy II PLLs Available *Note (1)*

Device	Fast PLLs								Enhanced PLLs			
	1	2	3	4	7	8	9	10	5	6	11	12
HC220	✓	✓							✓	✓		
HC230	✓	✓			✓	✓			✓	✓	✓	✓
HC240	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note to Table 2–6:
(1) PLL performance in the HC210W device may differ from the Stratix II FPGA prototype.

Figure 2–3. HardCopy II PLL Locations *Notes (1), (2)*



Notes to Figure 2–3:

(1) The PLLs may be located in the periphery or in the core of the device.

(2) This is the die-level top view of the device and is only a graphical representation of the PLL locations.

PLL functionality in HardCopy II devices remains the same as in Stratix II FPGA PLLs. Therefore, the Hardcopy II PLLs support PLL reconfiguration (the PLL can be dynamically configured in user mode).

HardCopy II enhanced and fast PLLs support a one-to-one mapping from Stratix II PLL resources. Table 2–7 shows the features of the different PLLs. For more information on the Stratix II PLL features, refer to the *Stratix II Device Handbook*.

Table 2–7. HardCopy II PLL Features

Feature	Enhanced PLL	Fast PLL
Clock multiplication and division	*m*/(*n* × post-scale counter) *(1)*	*m*/(*n* × post-scale counter) *(2)*
Phase shift	Down to 125-ps increments *(3)*	Down to 125-ps increments *(3)*
Clock switchover	✓	✓ *(4)*
PLL reconfiguration	✓	✓
Reconfigurable bandwidth	✓	✓
Spread-spectrum clocking	✓	
Programmable duty cycle	✓	✓
Number of clock outputs per PLL *(5)*	6	4
Number of dedicated external clock outputs per PLL	Three differential or six singled-ended	*(6)*
Number of feedback clock inputs per PLL	1 *(7)*	

Notes to Table 2–7:

(1) For enhanced PLLs, *m* and *n* range from 1 to 512 and post-scale counters range from 1 to 512 with 50% duty cycle. For non-50% duty-cycle clock outputs, post-scale counters range from 1 to 256.

(2) For fast PLLs, *n* can range from 1 to 4. The post-scale and *m* counters range from 1 to 32. For non-50% duty-cycle clock outputs, post-scale counters range from 1 to 16.

(3) The smallest phase shift is determined by the voltage controlled oscillator (VCO) period divided by eight. The supported phase shift range is from 125 to 250 ps. HardCopy II devices can shift all output frequencies in increments of at least 45° . Smaller degree increments are possible depending on the frequency and divide parameters. For non-50% duty cycle clock outputs post-scale counters range from 1 to 256.

(4) HardCopy II fast PLLs only support manual clock switchover.

(5) The clock outputs can be driven to internal clock networks or to a pin.

(6) The PLL clock outputs of the fast PLLs can drive to any I/O pin to be used as an external clock output. For high-speed differential I/O pins, the device uses a data channel to generate the transmitter output clock (`txclkout`).

(7) If the design uses external feedback input pins, you will lose one (or two, if f_{BIN} is differential) dedicated external clock output pin.

Clock Networks

There are 16 clock pins (`CLK[15..0]`) in HardCopy II devices that can drive either the global- or regional-clock networks. The `CLK` pins can drive clock ports or data inputs.

HardCopy II devices provide 16 dedicated global-clock networks and 32 regional-clock networks; the same as in Stratix II FPGAs. These clocks are organized to provide 24 unique clock sources per device quadrant

with low skew and delay. This clocking scheme provides up to 48 unique clock domains within the entire HardCopy II device. Table 2–8 lists the clock resources and features available in HardCopy II devices.

Table 2–8. Clock Network Resources & Features Available in HardCopy II Devices

Resources & Features	Availability
Number of global clock networks	16
Number of regional clock networks	32
Global clock input sources	Clock input pins, PLL outputs, logic array
Regional clock input sources	Clock input pins, PLL outputs, logic array
Number of unique clock sources in a quadrant	24 (16 global clocks and 8 regional clocks)
Number of unique clock sources in the entire device	48 (16 global clocks and 32 regional clocks)
Power-down mode	Global- and regional-clock networks, dual-regional-clock region
Clocking regions for high fan-out applications	Quadrant region, dual-regional, entire device via global- or regional-clock networks

Hardcopy II devices also support the same features as the Stratix II clock control block, which is available for each global- and regional-clock network. The control block has two functions:

- Clock source selection (dynamic selection for global clocks): The user can either dynamically select between two PLL outputs, between two clock pins (CLKp or CLKn), or a combination of the clock pins or PLL outputs.
- Clock power-down (dynamic clock enable or disable): In HardCopy II devices, the user can dynamically turn the clock off or on in user-mode.

I/O Structure & Features

The structure and features of the HardCopy II IOE remains the same as in Stratix II. Any feature implemented in Stratix II IOEs can be migrated to Hardcopy II IOEs.

The IOE feature set in HardCopy II devices can be classified in one of three categories:

- General purpose IOEs—The most commonly used I/O type in designs.
- Memory Interface IOEs—Includes features to interface with common external memory standards.
- High-speed IOEs—Supports high-speed data transmission and reception.

All I/O pins in Stratix II FPGAs support general-purpose I/O standards, which includes the LVTTL and LVCMOS I/O standards. In Stratix II FPGAs, the PCI clamping diode and memory interfaces are supported on the top and bottom I/O pins, while high-speed interfaces are supported on the left and right side I/O pins of the device.

The new general purpose IOEs in HardCopy II devices are a cost saving and area efficient advantage. The complex memory interface and the high-speed IOE circuitry is removed to save die area while still offering the more commonly-used features. The memory interface IOE supports all the features available in the general purpose IOE. The high-speed IOE also supports all the same features and I/O standards as the general purpose IOE, except for the PCI clamping diode (supported on the bottom general purpose IOEs in HC210 and HC220 devices).

In order to increase the I/O area efficiency of HardCopy II devices, the features available on any given IOE depends on the location.

Table 2–9 shows which I/O standards are supported by the different IOE types.

Table 2–9. Hardcopy II Supported I/O Standards (Part 1 of 3)

I/O Standard	Type	V_{CCIO} Level (V)		Memory Interface IOEs	General Purpose IOEs	High-Speed IOEs
		Input	Output			
3.3 V LVTTL/ LVCMOS	Single-ended	3.3/2.5	3.3	✓	✓	✓
2.5 V LVTTL/ LVCMOS	Single-ended	3.3/2.5	2.5	✓	✓	✓
1.8 V LVTTL/ LVCMOS	Single-ended	1.8/1.5	1.8	✓	✓	✓
1.5 V LVCMOS	Single-ended	1.8/1.5	1.5	✓	✓	✓
SSTL-2 class I	Voltage referenced	2.5	2.5	✓		
SSTL-2 class II	Voltage referenced	2.5	2.5	✓		
SSTL-18 class I	Voltage referenced	1.8	1.8	✓		
SSTL-18 class II	Voltage referenced	1.8	1.8	✓		
1.8 V HSTL class I	Voltage referenced	1.8	1.8	✓		

Table 2–9. Hardcopy II Supported I/O Standards (Part 2 of 3)

I/O Standard	Type	V_{CCIO} Level (V)		Memory Interface IOEs	General Purpose IOEs	High-Speed IOEs
		Input	Output			
1.8 V HSTL class II	Voltage referenced	1.8	1.8	✓		
1.5 V HSTL Class I	Voltage referenced	1.5	1.5	✓		
1.5 V HSTL Class II	Voltage referenced	1.5	1.5	✓		
PCI/PCI-X	Single-ended	3.3	3.3	✓ *(2)*	✓ *(2)*	
Differential SSTL-2 class I and II input	Pseudo differential *(1)*	3.3/2.5/ 1.8/1.5		*(3)*		
Differential SSTL-2 class I and II output	Pseudo differential *(1)*		2.5	*(3)*		
Differential SSTL-18 class I and II input	Pseudo differential *(1)*	3.3/2.5/ 1.8/1.5		*(3)*		
Differential SSTL-18 class I and II output	Pseudo differential *(1)*		1.8	*(3)*		
1.8 V differential HSTL class I and II input	Pseudo differential *(1)*	3.3/2.5/ 1.8/1.5		*(3)*		
1.8 V differential HSTL class I and II output	Pseudo Differential *(1)*		1.8	*(3)*		
1.5 V differential HSTL class I and II input	Pseudo differential *(1)*	3.3/2.5/ 1.8/1.5		*(3)*		
1.5 V differential HSTL class I and II output	Pseudo Differential *(1)*		1.5	*(3)*		
LVDS	Differential	2.5	2.5	*(5)*	*(4)*, *(6)*	✓
HyperTransport™ technology	Differential	2.5	2.5	*(5)*	*(4)*, *(6)*	✓

Table 2–9. Hardcopy II Supported I/O Standards (Part 3 of 3)

I/O Standard	Type	V_{CCIO} Level (V)		Memory Interface IOEs	General Purpose IOEs	High-Speed IOEs
		Input	Output			
LVPECL	Differential	3.3/2.5/ 1.8/1.5	*(8)*	*(8)*	*(8)*	

Notes to Table 2–9:

(1) Pseudo-differential HSTL and SSTL inputs only use the positive-polarity input in the speed path. The negative input is not connected internally. Pseudo-differential HSTL and SSTL outputs use two single-ended outputs with the second output programmed as inverted. This is similar to a Stratix II device implementation.
(2) The PCI clamping diode is only supported on the I/O pins on the top and bottom sides of the device.
(3) This I/O standard is only supported on the DQS, CLK and PLL_FB input pins or on the PLL_OUT output pins.
(4) This I/O standard is only supported on the bottom CLK and PLL_FB input pins or on the bottom PLL_OUT output pins.
(5) This I/O standard is only supported on the CLK and PLL_FB input pins or on the PLL_OUT output pins.
(6) Also supported on CLK9 and CLK11 pins.
(7) This I/O standard is only supported on CLK and PLL_FB input pins.
(8) LVPECL input I/O standard is supported on the top and bottom CLK and PLL_FB input pins. LVPECL output I/O standard is supported on the top and bottom PLL_OUT output pins. LVPECL support is similar to Stratix II devices.

The three types of IOEs are located in different areas of the device and are described in the following sections. HardCopy II devices have eight I/O banks, just as in Stratix II FPGAs. Figures 2–4 through 2–6 show which I/O type each bank supports.

Figure 2–4. I/O Type Support in HC210 & HC220 Devices *Notes (1), (2)*

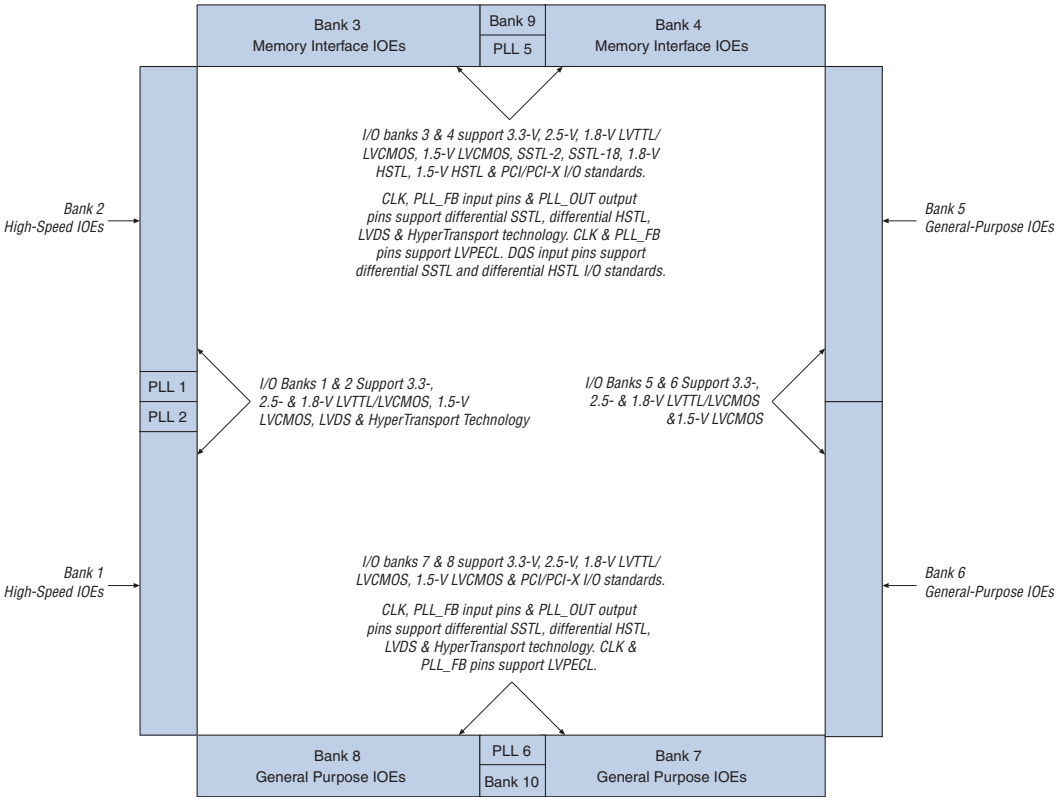


Figure 2–5. I/O Type Support in HC230 Devices *Notes (1), (2)*

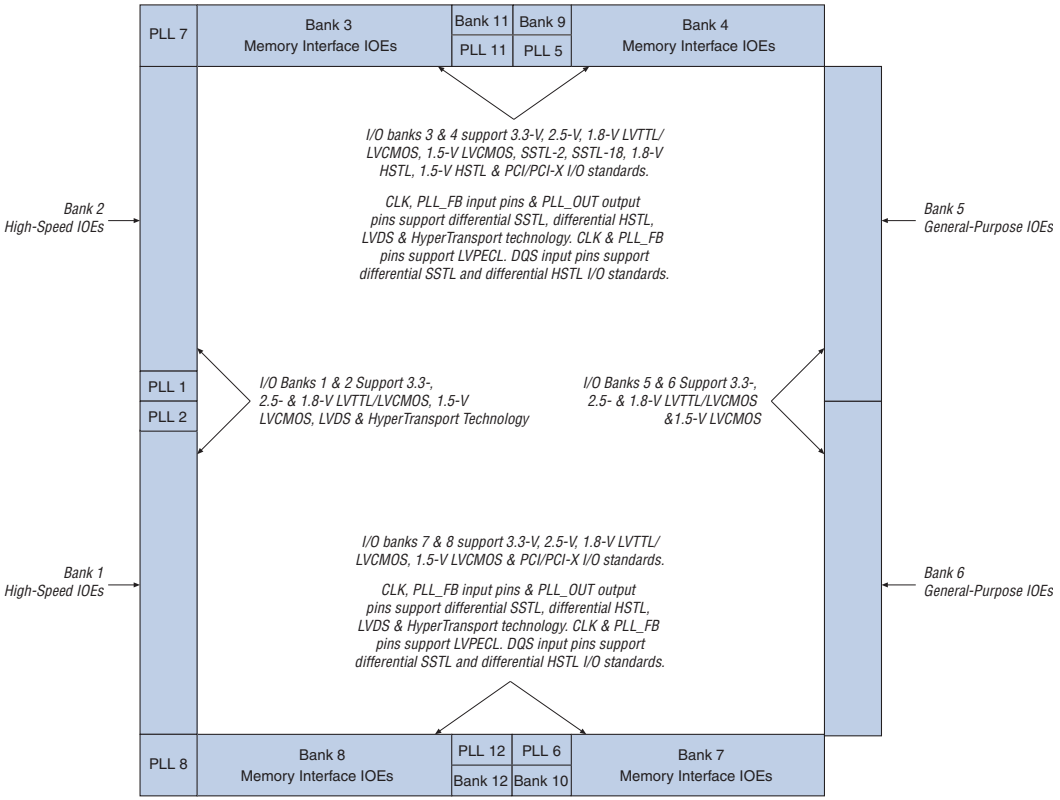
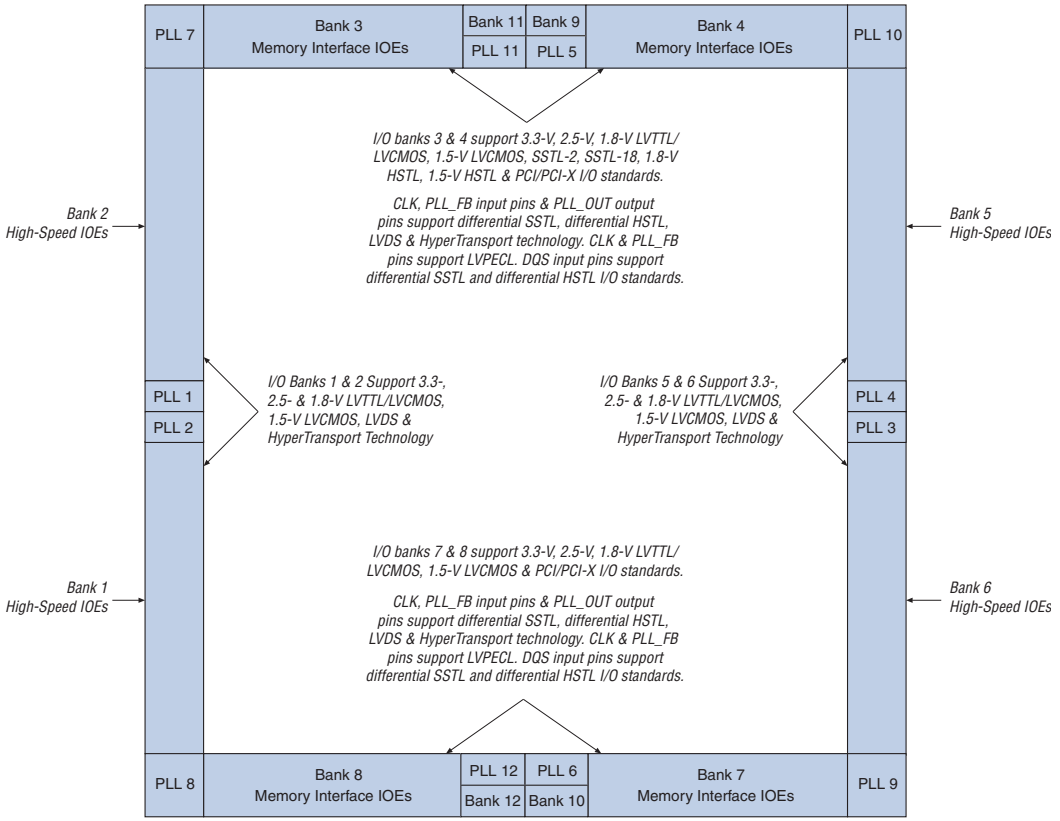


Figure 2–6. I/O Type Support in HC240 Devices *Notes (1), (2)*



Notes to Figures 2–4 through 2–6:

(1) In addition to supporting external memory interfaces, memory interface IOEs have the same features as general purpose IOEs. In addition to supporting high-speed I/O interfaces, high-speed IOEs have the same features as general purpose IOEs, except for the PCI clamping diode and LVPECL clock input support.

(2) This is a top view of the silicon die which corresponds to a reverse view for flip-chip packages. It is a graphical representation only.

When planning I/O placement for designs targeting HardCopy II devices, care should be taken to ensure the same I/O standards are supported in the same HardCopy II I/O banks as in the Stratix II I/O banks.

General Purpose IOE

The general purpose IOEs in HC210 and HC220 devices are located on the right side and at the bottom of the device. The general purpose IOEs in HC230 devices are located on the right side of the device. (Directions are

based on a top view of the silicon die.) HC240 devices do not have general purpose IOEs. The general purpose IOE functionality is supported in the memory interface IOEs for these devices. The high-speed IOEs also provide the same features as the general purpose IOEs except for the PCI clamping diode. In Stratix II FPGAs, all IOEs support the general purpose IOE features except the PCI diode, which is only supported on the top and bottom I/O pins.

The general purpose IOE has many features including:

- Dedicated single-ended I/O buffers
- 3.3 V, 64-bit, 66 MHz PCI compliance
- 3.3 V, 64-bit, 133 MHz PCI-X 1.0 compliance
- JTAG boundary-scan test (BST) support
- On-chip driver series termination (non-calibrated)
- Output drive strength control
- Tri-state buffers
- Bus-hold circuitry
- Programmable pull-up resistors
- Open-drain outputs
- PCI clamping diode (supported on the bottom I/O pins only)
- Double data rate (DDR) registers

General purpose 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
- 3.3 V PCI-X mode 1

The general purpose `CLK` and `PLL_FB` input pins and the `PLL_OUT` output pins support the following I/O standards:

- LVDS
- HyperTransport technology
- LVPECL (on input clocks and `PLL_OUT` only)

The programmable drive strengths available vary depending on the I/O standard being used and are listed in Table 2–10.

Table 2–10. Programmable Drive Strength Support for General-Purpose IOEs

I/O Standard	Programmable Drive Strength Options (mA)
3.3 V LVTTL	4, 8, 12
3.3 V LVCMOS	4, 8
2.5 V LVTTL/LVCMOS	4, 8, 12
1.8 V LVTTL/LVCMOS	2, 4, 6, 8
1.5 V LVCMOS	2, 4

General purpose IOEs support non-calibrated on-chip series termination. 50 and 25 Ω on-chip series termination is available for 3.3 V or 2.5 V I/O standards. 50-Ω on-chip series termination is available for 1.8 and 1.5 V I/O standards (pending characterization).

Memory Interface IOE

Memory interface IOEs in HC210 and HC220 devices are located on the top of the device. Memory interface IOEs in HC230 and HC240 devices are located on the top and the bottom of the device. In Stratix II FPGAs, the top and bottom IOEs support the memory interface IOE features.

The memory interface IOE has many features including:

- Dedicated single-ended I/O buffers
- 3.3 V, 64-bit, 66 MHz PCI compliance
- 3.3 V, 64-bit, 133 MHz PCI-X 1.0 compliance
- JTAG BST support
- On-chip driver series termination
- V_{REF} pins
- Output drive strength control
- Tri-state buffers
- Bus-hold circuitry
- Programmable pull-up resistors
- Open-drain outputs
- PCI clamping diode
- DQ and DQS I/O pins
- Double data rate (DDR) registers

The following I/O standards are supported when using the memory interface IOEs and can be used to interface to external memory, including DDR and DDR2 SDRAM, and QDRII, RLDRAM II, and SDR SRAM:

- 3.3 V LVTTL/LVCMOS
- 2.5 V LVTTL/LVCMOS
- 1.8 V LVTTL/LVCMOS
- 1.5 V LVCMOS
- 3.3 V PCI
- 3.3 V PCI-X mode 1
- SSTL-2 class I and II
- SSTL-18 class I and II
- 1.8 V HSTL class I and II
- 1.5 V HSTL class I and II

The memory interface DQS, `CLK`, and `PLL_FB` input pins and the `PLL_OUT` output pins support the following I/O standards:

- LVTTL/LVCMOS
- SSTL-2 class I and II
- SSTL-18 class I and II
- 1.8 V HSTL class I and II
- 1.5 V HSTL class I and II
- Differential SSTL-2 class I and II
- Differential SSTL-18 class I and II
- 1.8 V differential HSTL class I and II
- 1.5 V differential HSTL class I and II
- LVDS (not supported on DQS pins)
- HyperTransport technology (not supported on DQS pins)
- LVPECL on input clocks and `PLL_OUT` only (not supported on DQS pins)

Pseudo-differential HSTL and SSTL inputs are supported on clock and DQS pins, while outputs are supported on dedicated `PLL_OUT` and DQS pins. Pseudo-differential HSTL and SSTL I/O standards use two single-ended outputs with the second output programmed as inverted. Pseudo-differential HSTL and SSTL inputs treat differential inputs as two single-ended HSTL and SSTL inputs and only decode one of them. This I/O support is the same as in Stratix II FPGAs.

The functionality of all DQS circuitry in HardCopy II devices is the same as in Stratix II FPGAs. Table 2–11 shows the number of DQS/DQ groups supported in each HardCopy II device density and package.

Table 2–11. DQS & DQ Bus Mode Support

Device	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
HC210	484-pin FineLine BGA	4	2	0	0
HC220	672-pin FineLine BGA	9	4	2	0
	780-pin FineLine BGA	9	4	2	0
HC230	1,020-pin FineLine BGA	36	18	8	4
HC240	1,020-pin FineLine BGA	36	18	8	4
	1,508-pin FineLine BGA	36	18	8	4

The programmable drive strengths available vary depending on the I/O standard used. The options are listed in Table 2–12.

Table 2–12. Programmable Drive Strength Support for Memory Interface IOEs

I/O Standard	Programmable Drive Strength Options (mA)
3.3-V LVTTL	4, 8, 12, 16, 20, 24
3.3-V LVCMOS	4, 8, 12, 16, 20, 24
2.5-V LVTTL/LVCMOS	4, 8, 12, 16
1.8-V LVTTL/LVCMOS	2, 4, 6, 8, 10, 12
1.5-V LVCMOS	2, 4, 6, 8
SSTL-2 class I	8, 12
SSTL-2 class II	16, 20, 24
SSTL-18 class I	4, 6, 8, 10, 12
SSTL-18 class II	8, 16, 18, 20
1.8-V HSTL class I	4, 6, 8, 10, 12
1.8-V HSTL class II	16, 18, 20
1.5-V HSTL class I	4, 6, 8, 10, 12
1.5-V HSTL class II	16, 18, 20

Memory interface IOEs support both non-calibrated and calibrated on-chip series termination. 50 and 25 Ω on-chip series termination is available for 3.3, 2.5, or 1.8 V I/O standards. 50 Ω on-chip series termination is available for 1.5 or 1.2 V I/O standards (pending characterization).

If on-chip series termination is enabled, programmable drive strength support is not available.

High-Speed IOE

High-speed IOEs in HC210, HC220, and HC230 devices are located on the left side of the device. High-speed IOEs in HC240 devices are located on the left and right sides of the device. (Directions are based on a top view of the silicon die.) Unlike Stratix II left and right side I/O pins, Hardcopy II left and right side I/O pins do not support SSTL or HSTL I/O standards or the PCI clamping diode. In Stratix II FPGAs, the right and left IOEs support the high-speed IOE features.

The high-speed IOE has many features including:

- Dedicated single-ended I/O buffers
- Differential I/O buffer
- JTAG BST support
- On-chip driver series termination (non-calibrated)
- On-chip termination for differential I/O standards
- Output drive strength control
- Tri-state buffers
- Bus-hold circuitry
- Programmable pull-up resistors
- Open-drain outputs
- Transmit serializer
- Receive deserializer
- Dynamic phase alignment (DPA)
- Double data rate (DDR) registers

The following I/O standards are supported when using high-speed IOEs:

- 3.3 V LVTTL/LVCMOS
- 2.5 V LVTTL/LVCMOS
- 1.8 V LVTTL/LVCMOS
- 1.5- V LVCMOS
- LVDS
- HyperTransport technology

The SERDES and DPA circuitry and functionality is the same in Hardcopy II devices as in Stratix II FPGAs. HardCopy II devices support differential I/O standards at rates up to 1 Gbps when using DPA, and at rates up to 840 Mbps when not using DPA. Table 2–13 provides the number of differential channels per HardCopy II device.

Table 2–13. Number of Differential Channels in HardCopy II Devices *Notes (1), (2)*

Channel	HC210W	HC210	HC220		HC230	HC240	
	484-Pin FineLine BGA (Wire-Bond)	484-Pin FineLine BGA	672-Pin FineLine BGA	780-Pin FineLine BGA	1,020-Pin FineLine BGA	1,020-Pin FineLine BGA	1,508-Pin FineLine BGA
Transmitter channels	13	19	29	29	44	88	116
Receiver channels	17	21	31	31	46	92	116

Notes to Table 2–13:
(1) The pin count does not include dedicated PLL input and output pins.
(2) The total number of receiver channels includes the non-dedicated clock channels that can optionally be used as data channels.

Hardcopy II high-speed IOEs, which are on the left and/or right sides of the device, support fewer programmable drive strengths than Stratix II side IOEs. The programmable drive strengths available vary depending on the I/O standard being used. The options are listed in Table 2–14.

Table 2–14. Programmable Drive Strength Support for High-Speed IOEs

I/O Standard	Programmable Drive Strength Options (mA)
3.3 V LVTTL	4, 8, 12
3.3 V LVCMOS	4, 8
2.5 V LVTTL/LVCMOS	4, 8, 12
1.8 V LVTTL/LVCMOS	2, 4, 6, 8
1.5 V LVCMOS	2, 4

High-speed IOEs support non-calibrated on-chip series termination and differential termination on the receiver channels. 50 and 25 Ω on-chip series termination is available for 3.3 or 2.5 V I/O standards. 50 Ω on-chip series termination is available for 1.8- and 1.5 V I/O standards (pending characterization).

Power-Up Modes

The functionality of structured ASICs is determined before they are produced. Therefore, they do not require programmability. HardCopy II structured ASICs follow the same principle, enabling traditional ASIC-like power up. Although prototyping FPGAs require configuration upon power up, the HardCopy II structured ASICs do not need to be configured. HardCopy II devices do not support configuration and designers should take this into account in the prototyping to production development process. The HardCopy II device does not require a configuration device, but you must ensure that the *n*CE pin is low and that the *n*CONFIG and *n*STATUS pins are high after power up.

HardCopy II devices do not support FPGA configuration emulation, other configuration modes, including remote system upgrades and design security using configuration bitstream encryption.

HardCopy II devices support both instant on and instant on after 50 ms power-up modes. In the instant on power-up mode, the HardCopy II device is available for use shortly after the device powers up to a safe operating voltage. The on-chip power-on reset (POR) circuit will reset all registers. The *n*CE, *n*CONFIG, and *n*STATUS signals must be at the appropriate logic levels for the CONF_DONE output to be tristated once the POR has elapsed. This option is similar to an ASIC's functionality upon power up and is the most likely scenario in production.

In the instant on after 50 ms power-up mode, the HardCopy II device behaves similarly to the instant on mode, except that there is an additional delay of 50 ms, during which time the device will be held in reset. The CONF_DONE output is pulled low during this time, and then tri-stated after the 50 ms have elapsed.

For more information about which power-up modes HardCopy II devices support, refer to the *Power-Up Modes & Configuration Emulation in HardCopy Series Devices* chapter in the *HardCopy Series Handbook*.

3. Boundary-Scan Support

H51017-2.1

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All HardCopy® II structured ASICs provide Joint Test Action Group (JTAG) boundary-scan test (BST) circuitry that complies with the IEEE Std. 1149.1-1990 specification. The BST architecture offers the capability to efficiently test components on printed circuit boards (PCBs) with tight lead spacing by testing pin connections, without using physical test probes, and capturing functional data while a device is in normal operation. Boundary-scan cells in a device can force signals onto pins, or capture data from pin or core logic signals. Forced test data is serially shifted into the boundary-scan cells. Captured data is serially shifted out and externally compared to expected results.

A device using the JTAG interface uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS`, and `TRST` pins have weak internal pull-up resistors. The `TDO` output is powered by V_{CCIO} HardCopy II devices support the JTAG instructions shown in Table 3–1.

Table 3–1. HardCopy II 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.
EXTEST *(1)*	00 0000 1111	Allows the external circuitry and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	11 1111 1111	Places the 1-bit BYPASS register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.

Table 3–1. HardCopy II 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.

Note to Table 3–1:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

The BSDL files for HardCopy II devices are different from the corresponding Stratix® II FPGAs. Contact Altera Applications for HardCopy II BSDL files.

The HardCopy II device instruction register length is 10 bits and the USERCODE register length is 32 bits. The USERCODE registers are not reprogrammable and are mask-programmed. The designer can choose an appropriate 32 bit sequence which will be programmed into the USERCODE registers.

Tables 3–2 and 3–3 show the boundary-scan register length and device IDCODE information for HardCopy II devices.

Table 3–2. HardCopy II Boundary-Scan Register Length *Note (1)*

Device	Boundary-Scan Register Length
HC210W	*(2)*
HC210	1050
HC220	1530
HC230	2154
HC240	2910

Notes to Table 3–2:
(1) These values are preliminary.
(2) Contact Altera Applications for more information.

Table 3–3. 32 Bit HardCopy II Device IDCODE

Device	IDCODE (32 Bits) *(1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) *(2)*
HC210W	0000	0010 0000 0000 0011	000 0110 1110	1
HC210	0000	0010 0000 0000 0100	000 0110 1110	1
HC220	0000	0010 0000 0000 0101	000 0110 1110	1
HC230	0000	0010 0000 0000 0110	000 0110 1110	1
HC240	0000	0010 0000 0000 0111	000 0110 1110	1

Notes to Table 3–3:
(1) The most significant bit (MSB) is on the left.
(2) The least significant bit (LSB) of IDCODE is always 1.

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

Figure 3–1. HardCopy II JTAG Waveforms

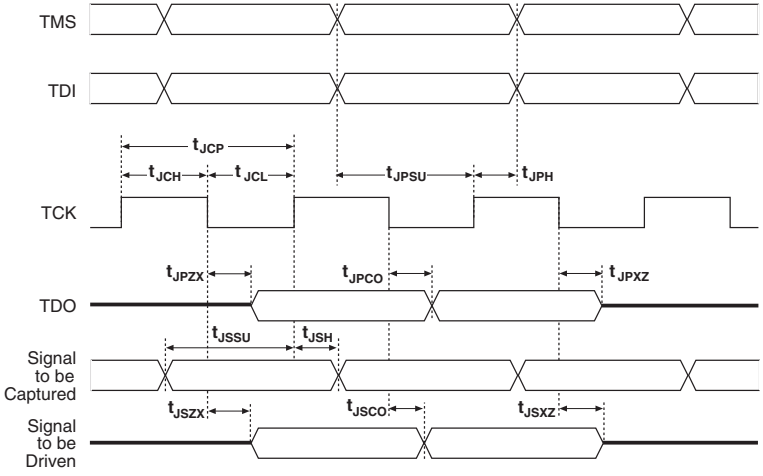


Table 3–4 shows the JTAG timing parameters and values for HardCopy II devices.

Table 3–4. HardCopy II JTAG Timing Parameters & Values (Part 1 of 2)

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

Table 3–4. HardCopy II JTAG Timing Parameters & Values (Part 2 of 2)

Symbol	Parameter	Min	Max	Unit
t_{JSZX}	Update register high impedance to valid output		35	ns
t_{JSXZ}	Update register valid output to high impedance		35	ns

For more information on JTAG or boundary-scan testing, refer to *AN 39: IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*.

Stratix II FPGAs support the SignalTap® II embedded logic analyzer, which monitors design operation over a period of time through the JTAG interface. The SignalTap II logic analyzer is a useful feature during the FPGA prototyping phase, but should be removed if not needed once the design has been migrated to a HardCopy II device.

4. Operating Conditions

H51018-2.1

Introduction

This chapter provides preliminary information on absolute maximum ratings, recommended operating conditions, DC electrical characteristics, and other specifications for HardCopy® II devices.

Absolute Maximum Ratings

Table 4–1 contains the absolute maximum ratings for the HardCopy II device family.

Table 4–1. HardCopy II Device Absolute Maximum Ratings *Notes (1), (2), (3)*

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage	With respect to ground	–0.5	1.8	V
V_{CCIO}	Supply voltage	With respect to ground	–0.5	4.6	V
V_{CCPD}	Supply voltage	With respect to ground	3.0	3.6	V
V_I	DC input voltage *(4)*		–0.5	4.6	V
I_{OUT}	DC output current, per pin		–25	40	mA
T_{STG}	Storage temperature	No bias	–65	150	°C
T_J	Junction temperature	Ball-grid array (BGA) packages under bias	–55	125	°C

Notes to Table 4–1:

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

(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) Supply voltage specifications apply to voltage readings taken at the device pins, not at the power supply.

(4) During transitions, the inputs may overshoot to the voltage shown in Table 4–2 based upon the input duty cycle. The DC case is equivalent to 100% duty cycle. During transitions, the inputs may undershoot to –2.0 V for input currents less than 100 mA and periods shorter than 20 ns.

Table 4–2. Maximum Duty Cycles in Voltage Transitions (Part 1 of 2)

V_{IN} (V)	Maximum Duty Cycles
4.0	100%
4.1	90%
4.2	50%

Table 4–2. Maximum Duty Cycles in Voltage Transitions (Part 2 of 2)	
V_{IN} (V)	Maximum Duty Cycles
4.3	30%
4.4	17%
4.5	10%

Recommended Operating Conditions

Table 4–3 contains the HardCopy II device family recommended operating conditions.

Table 4–3. HardCopy II Device Recommended Operating Conditions (Part 1 of 2) Note (1)					
Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	Maximum rise time = 100 ms (2)	1.15	1.25	V
V_{CCIO}	Supply voltage for output buffers, 3.3-V operation	Maximum rise time = 100 ms (2)	3.00	3.60	V
	Supply voltage for output buffers, 2.5-V operation	Maximum rise time = 100 ms (2)	2.375	2.625	V
	Supply voltage for output buffers, 1.8-V operation	Maximum rise time = 100 ms (2)	1.71	1.89	V
	Supply voltage for output buffers, 1.5-V operation	Maximum rise time = 100 ms (2)	1.425	1.575	V
V_{CCPD}	Supply voltage for pre-drivers as well as configuration and JTAG I/O buffers.	100 µs ≤rise time ≤100 ms (3)	3.135	3.465	V
V_I	Input voltage	(4), (5)	–0.5	4.0	V

Table 4–3. HardCopy II Device Recommended Operating Conditions (Part 2 of 2) Note (1)					
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

Notes to Table 4–3:
(1) Supply voltage specifications apply to voltage readings taken at the device pins, not at the power supply.
(2) Maximum V_{CC} rise time is 100 ms, and V_{CC} must rise monotonically.
(3) V_{CCPD} must ramp-up from 0 V to 3.3 V within 100 µs to 100 ms. If V_{CCPD} is not ramped up within this specified time, the HardCopy II device will not power up successfully.
(4) During transitions, the inputs may overshoot to the voltage shown in Table 4–2 based upon the input duty cycle. The DC case is equivalent to 100% duty cycle. During transitions, the inputs may undershoot to –2.0 V for input currents less than 100 mA and periods shorter than 20 ns.
(5) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT}, V_{CCPD}, and V_{CCIO} are powered.

DC Electrical Characteristics

Table 4–4 shows the HardCopy II device family DC electrical characteristics.

Table 4–4. HardCopy II Device DC Operating Conditions Note (1)						
Symbol	**Parameter**	**Conditions**	**Minimum**	**Typical**	**Maximum**	**Unit**
I_I	Input pin leakage current	$V_I = V_{CCIO}$ max to 0 V (2)	–10		10	µA
I_{OZ}	Tri-stated I/O pin leakage current	$V_O = V_{CCIO}$ max to 0 V (2)	–10		10	µA
I_{CC0}	V_{CC} supply current (standby) (all memory blocks in power-down mode)	V_I = ground, no load, no toggling inputs		(3)		mA
R_{CONF}	Value of I/O pin pull-up resistor before and during configuration	V_{CCIO} = 3.0 V (4)	20		50	kΩ
		V_{CCIO} = 2.375 V (4)	30		80	kΩ
		V_{CCIO} = 1.71 V (4)	60		150	kΩ

Notes to Table 4–4:
(1) Typical values are for T_A = 25 °C, V_{CCINT} = 1.2 V, and V_{CCIO} = 1.5, 1.8, 2.5, and 3.3 V.
(2) 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).
(3) This specification is pending device characterization.
(4) Pin pull-up resistance values will lower if an external source drives the pin higher than V_{CCIO}.

I/O Standard Specifications

Tables 4–5 through 4–26 show the HardCopy II device family I/O standard specifications.

Table 4–5. LVTTL Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output-supply voltage		3.135	3.465	V
V_{IH}	High-level input voltage		1.7	4.0	V
V_{IL}	Low-level input voltage		–0.3	0.8	V
V_{OH}	High-level output voltage	I_{OH} = –4 *(2)*, *(3)*	2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 4 *(2)*, *(3)*		0.45	V

Notes to Table 4–5:
(1) HardCopy II devices comply to the narrow range for the supply voltage as specified in the EIA/JEDEC Standard, JESD8-B.
(2) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.
(3) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–6. LVCMOS Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output-supply voltage		3.135	3.465	V
V_{IH}	High-level input voltage		1.7	4.0	V
V_{IL}	Low-level input voltage		–0.3	0.8	V
V_{OH}	High-level output voltage	V_{CCIO} = 3.0, I_{OH} = –0.1 mA *(2)*, *(3)*	V_{CCIO} – 0.2		V
V_{OL}	Low-level output voltage	V_{CCIO} = 3.0, I_{OL} = 0.1 mA *(2)*, *(3)*		0.2	V

Notes to Table 4–6:
(1) HardCopy II devices comply to the narrow range for the supply voltage as specified in the EIA/JEDEC Standard, JESD8-B.
(2) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.
(3) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–7. 2.5 V I/O Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output-supply voltage		2.375	2.625	V
V_{IH}	High-level input voltage		1.7	4.0	V

Table 4–7. 2.5 V I/O Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{IL}	Low-level input voltage		–0.3	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –1 mA *(2)*, *(3)*	2.0		V
V_{OL}	Low-level output voltage	I_{OL} = 1 mA *(2)*, *(3)*		0.4	V

Notes to Table 4–7:

(1) HardCopy II devices V_{CCIO} voltage-level support of 2.5± -5% is narrower than defined in the normal range of the EIA/JEDEC Standard.

(2) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(3) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–8. 1.8 V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output-supply voltage		1.71	1.89	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	2.25	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 *(2)*, *(3)*	V_{CCIO} – 0.45		V
V_{OL}	Low-level output voltage	I_{OL} = 2 to 8 mA *(2)*, *(3)*		0.45	V

Notes to Table 4–8:

(1) HardCopy II devices V_{CCIO} voltage-level support of 1.8± -5% is narrower than defined in the normal range of the EIA/JEDEC Standard.

(2) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(3) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, Volume 1* for more information.

Table 4–9. 1.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO} *(1)*	Output-supply voltage		1.425	1.575	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3	0.35 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 mA *(2)*, *(3)*	0.75 × V_{CCIO}		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(2)*, *(3)*		0.25 × V_{CCIO}	V

Notes to Table 4–9:

(1) HardCopy II devices V_{CCIO} voltage-level support of 1.5± -5% is narrower than defined in the normal range of the EIA/JEDEC Standard.

(2) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(3) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Figures 4–1 and 4–2 show receiver input and transmitter output waveforms, respectively, for all differential I/O standards (LVDS, LVPECL, and HyperTransport technology).

Figure 4–1. Receiver Input Waveforms for Differential I/O Standards

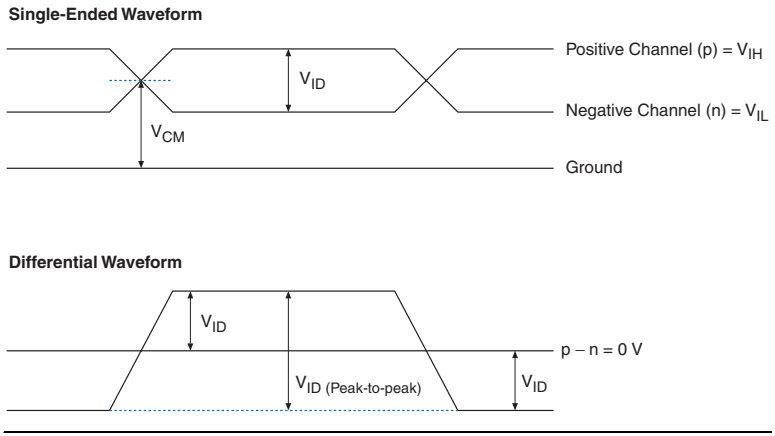


Figure 4–2. Transmitter Output Waveforms for Differential I/O Standards

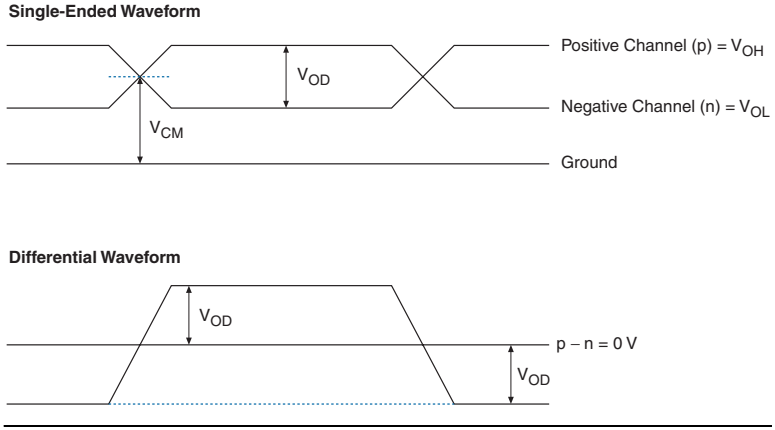


Table 4–10. 2.5-V LVDS I/O Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage for I/O banks that support high-speed IOEs *(1)*, *(2)*		2.375	2.5	2.625	V
	Output and feedback pins in PLL banks 9, 10, 11, and 12 *(3)*		3.135	3.3	3.465	V
V_{ID} (peak-to-peak)	Input differential voltage swing (single-ended)		300	600	1,000	mV
V_{ICM}	Input common mode voltage		200		1,800	V
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	250	375	550	mV
ΔV_{OD}	Change in V_{OD} between high and low	R_L = 100 Ω			50	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	1.125	1.25	1.375	V
ΔV_{OCM}	Change in V_{OCM} between high and low	R_L = 100 Ω			50	mV

Table 4–10. 2.5-V LVDS I/O Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
R_L	Receiver differential input discrete resistor (external to HardCopy II devices)		90	100	110	Ω

Notes to Table 4–10:
(1) IOEs = I/O elements.
(2) For information on which I/O banks support high-speed IOEs, refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1*.
(3) The top and bottom clock input differential buffers in I/O banks 3, 4, 7, and 8 are powered by V_{CCINT}, not V_{CCIO}. The PLL clock output and feedback differential buffers are powered by VCC_PLLOUT. For differential clock output and feedback operation, connect VCC_PLLOUT to 3.3V.

Table 4–11. LVPECL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO} *(1)*	I/O supply voltage		3.135	3.3	3.465	V
V_{ID} (peak-to-peak)	Input differential voltage swing (single-ended)		300	600	1,000	mV
V_{ICM}	Input common mode voltage		1.0		2.0	V
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	525	700	970	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	525	700	970	V
R_L	Receiver differential input resistor		90	100	110	Ω

Note to Table 4–11:
(1) The top and bottom clock input differential buffers in I/O banks 3, 4, 7, and 8 are powered by V_{CCINT}, not V_{CCIO}. The PLL clock output and feedback differential buffers are powered by VCC_PLLOUT. For differential clock output and feedback operation, connect VCC_PLLOUT to 3.3 V.

Table 4–12. HyperTransport Technology Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage for I/O banks that support high-speed IOEs *(1)*		2.375	2.5	2.625	V
	Output and feedback pins in PLL banks 9, 10, 11, and 12 *(2)*		3.135	3.3	3.465	V
V_{ID} (peak-to-peak)	Input differential voltage swing (single-ended)	R_L = 100 Ω	300	600	900	mV
V_{ICM}	Input common mode voltage	R_L = 100 Ω	385	600	845	mV

Table 4–12. HyperTransport Technology Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	400	600	820	mV
Δ V_{OD}	Change in V_{OD} between high and low	R_L = 100 Ω			75	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	440	600	780	mV
Δ V_{OCM}	Change in V_{OCM} between high and low	R_L = 100 Ω			50	mV
R_L	Receiver differential input resistor		90	100	110	Ω

Notes to Table 4–12:

(1) For information on which I/O banks support high-speed IOEs, refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1*.

(2) The top and bottom clock input differential buffers in I/O banks 3, 4, 7, and 8 are powered by V_{CCINT}, not V_{CCIO}. The PLL clock output and feedback differential buffers are powered by VCC_PLLOUT. For differential clock output and feedback operation, connect VCC_PLLOUT to 3.3V.

Table 4–13. 3.3-V PCI Specifications

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		–0.3		0.3 × V_{CCIO}	V
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 4–14. PCI-X Mode 1 Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		3.0		3.6	V
V_{IH}	High-level input voltage		0.5 × V_{CCIO}		V_{CCIO} + 0.5	V
V_{IL}	Low-level input voltage		–0.3		0.35 × V_{CCIO}	V
V_{IPU}	Input pull-up voltage		0.7 × V_{CCIO}			V
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 4–15. SSTL-18 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		1.71	1.8	1.89	V
V_{REF}	Reference voltage		0.855	0.9	0.945	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.125			V
$V_{IL(DC)}$	Low-level DC input voltage				V_{REF} – 0.125	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.25			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.25	V
V_{OH}	High-level output voltage	I_{OH} = –6.7 mA *(1)*, *(2)*	V_{TT} + 0.475			V
V_{OL}	Low-level output voltage	I_{OL} = 6.7 mA *(1)*, *(2)*			V_{TT} – 0.475	V

Notes to Table 4–15:

(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–16. SSTL-18 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		1.71	1.8	1.89	V
V_{REF}	Reference voltage		0.855	0.9	0.945	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.125			V
$V_{IL(DC)}$	Low-level DC input voltage				V_{REF} – 0.125	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.25			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.25	V
V_{OH}	High-level output voltage	I_{OH} = –13.4 mA *(1)*, *(2)*	V_{TT} – 0.28			V
V_{OL}	Low-level output voltage	I_{OL} = 13.4 mA *(1)*, *(2)*			0.28	V

Notes to Table 4–16:
(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.
(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–17. SSTL-18 Differential Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		1.71	1.8	1.89	V
$V_{SWING(DC)}$	DC differential input voltage		0.25			V
$V_{X(AC)}$	AC differential input cross point voltage		(V_{CCIO}/2) – 0.175		(V_{CCIO}/2) + 0.175	V
$V_{SWING(AC)}$	AC differential input voltage		0.5			V
V_{ISO}	Input clock signal offset voltage			0.5 × V_{CCIO}		V
ΔV_{ISO}	Input clock signal offset voltage variation			±200		V
$V_{OX(AC)}$	AC differential cross point voltage		(V_{CCIO}/2) – 0.125		(V_{CCIO}/2) + 0.125	V

Table 4–18. 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		1.188	1.25	1.313	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 *(1)*, *(2)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(1)*, *(2)*			V_{TT} – 0.57	V

Notes to Table 4–18:

(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–19. SSTL-2 Class II 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		1.188	1.25	1.313	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 *(1)*, *(2)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(1)*, *(2)*			V_{TT} – 0.76	V

Notes to Table 4–19:

(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–20. SSTL-2 Differential Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		2.375	2.5	2.625	V
$V_{SWING(DC)}$	DC differential input voltage		0.36			V

Table 4–20. SSTL-2 Differential Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
$V_{X(AC)}$	AC differential input cross point voltage		$(V_{CCIO}/2) - 0.2$		$(V_{CCIO}/2) + 0.2$	V
$V_{SWING(AC)}$	AC differential input voltage		0.7			V
V_{ISO}	Input clock signal offset voltage			$0.5 \times V_{CCIO}$		V
ΔV_{ISO}	Input clock signal offset voltage variation			±200		V
$V_{OX(AC)}$	AC differential output cross point voltage		$(V_{CCIO}/2) - 0.2$		$(V_{CCIO}/2) + 0.2$	V

Table 4–21. 1.5-V HSTL Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		1.425	1.5	1.575	V
V_{REF}	Input reference voltage		0.713	0.75	0.788	V
V_{TT}	Termination voltage		0.713	0.75	0.788	V
$V_{IH(DC)}$	DC high-level input voltage		$V_{REF} + 0.1$			V
$V_{IL(DC)}$	DC low-level input voltage		–0.3		$V_{REF} - 0.1$	V
$V_{IH(AC)}$	AC high-level input voltage		$V_{REF} + 0.2$			V
$V_{IL(AC)}$	AC low-level input voltage				$V_{REF} - 0.2$	V
V_{OH}	High-level output voltage	I_{OH} = 8 mA *(1)*, *(2)*	$V_{CCIO} - 0.4$			V
V_{OL}	Low-level output voltage	I_{OH} = –8 mA *(1)*, *(2)*			0.4	V

Notes to Table 4–21:

(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–22. 1.5-V HSTL Class II Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		1.425	1.50	1.575	V
V_{REF}	Input reference voltage		0.713	0.75	0.788	V
V_{TT}	Termination voltage		0.713	0.75	0.788	V

Table 4–22. 1.5-V HSTL Class II Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
$V_{IH(DC)}$	DC high-level input voltage		V_{REF} + 0.1			V
$V_{IL(DC)}$	DC low-level input voltage		–0.3		V_{REF} – 0.1	V
$V_{IH(AC)}$	AC high-level input voltage		V_{REF} + 0.2			V
$V_{IL(AC)}$	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 16 mA *(1)*, *(2)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(1)*, *(2)*			0.4	V

Notes to Table 4–22:

(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.

(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–23. 1.5-V Differential HSTL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		1.425	1.5	1.575	V
$V_{DIF(DC)}$	DC input differential voltage		0.2			V
$V_{CM(DC)}$	DC common mode input voltage		0.68		0.9	V
$V_{DIF(AC)}$	AC differential input voltage		0.4			V
$V_{OX(AC)}$	AC differential cross point voltage		0.68		0.9	V

Table 4–24. 1.8-V HSTL Class I Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		1.71	1.80	1.89	V
V_{REF}	Input reference voltage		0.85	0.90	0.95	V
V_{TT}	Termination voltage		0.85	0.90	0.95	V
$V_{IH(DC)}$	DC high-level input voltage		V_{REF} + 0.1			V
$V_{IL(DC)}$	DC low-level input voltage		–0.3		V_{REF} – 0.1	V

Table 4–24. 1.8-V HSTL Class I Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
$V_{IH(AC)}$	AC high-level input voltage		V_{REF} + 0.2			V
$V_{IL(AC)}$	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 8 mA *(1)*, *(2)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –8 mA *(1)*, *(2)*			0.4	V

Notes to Table 4–24:
(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.
(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–25. 1.8-V HSTL Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output-supply voltage		1.71	1.80	1.89	V
V_{REF}	Input reference voltage		0.85	0.90	0.95	V
V_{TT}	Termination voltage		0.85	0.90	0.95	V
$V_{IH(DC)}$	DC high-level input voltage		V_{REF} + 0.1			V
$V_{IL(DC)}$	DC low-level input voltage		–0.3		V_{REF} – 0.1	V
$V_{IH(AC)}$	AC high-level input voltage		V_{REF} + 0.2			V
$V_{IL(AC)}$	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 16 mA *(1)*, *(2)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(1)*, *(2)*			0.4	V

Notes to Table 4–25:
(1) Drive strength is programmable according to values in Tables 2–10, 2–12, and 2–14.
(2) Drive strength varies based on pin location. Refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1* for more information.

Table 4–26. 1.8-V Differential HSTL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		1.71	1.80	1.89	V
$V_{DIF(DC)}$	DC input differential voltage		0.2			V
$V_{CM(DC)}$	DC common mode input voltage		0.68		0.9	V
$V_{DIF(AC)}$	AC differential input voltage		0.4			V
$V_{OX(AC)}$	AC differential cross point voltage		0.68		0.9	V

Bus Hold Specifications

Table 4–27 shows the HardCopy II device family bus hold specifications.

Table 4–27. 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)	*(1)*		30		50		70		µA
High sustaining current	$V_{IN} < V_{IH}$ (minimum)	*(1)*		–30		–50		–70		µA
Low overdrive current	$0 V < V_{IN} < V_{CCIO}$		*(1)*		200		300		500	µA
High overdrive current	$0 V < V_{IN} < V_{CCIO}$		*(1)*		–200		–300		–500	µA

Note to Table 4–27:
(1) This specification is pending device characterization.

On-Chip Termination Specifications

Tables 4–28 and 4–29 define the specification for internal termination resistance tolerance when using series or differential on-chip termination.

Table 4–28. Series On-Chip Termination Specification for I/O Banks Supporting Memory Interface IOEs *Note (1)*

Symbol	Description	Conditions	Resistance Tolerance		
			Commercial Max	Industrial Max	Unit
25 Ω R_S 3.3/2.5	Internal series termination with calibration (25 Ω setting)	V_{CCIO} = 3.3/2.5 V	*(2)*	*(2)*	%
	Internal series termination without calibration (25 Ω setting)	V_{CCIO} = 3.3/2.5 V	±30	*(2)*	%
50 Ω R_S 3.3/2.5	Internal series termination with calibration (50 Ω setting)	V_{CCIO} = 3.3/2.5 V	*(2)*	*(2)*	%
	Internal series termination without calibration (50 Ω setting)	V_{CCIO} = 3.3/2.5 V	±30	*(2)*	%
25 Ω R_S 1.8	Internal series termination with calibration (25 Ω setting)	V_{CCIO} = 1.8 V	*(2)*	*(2)*	%
	Internal series termination without calibration (25 Ω setting)	V_{CCIO} = 1.8 V	±30	*(2)*	%
50 Ω R_S 1.8	Internal series termination with calibration (50 Ω setting)	V_{CCIO} = 1.8 V	*(2)*	*(2)*	%
	Internal series termination without calibration (50 Ω setting)	V_{CCIO} = 1.8 V	±30	*(2)*	%
50 Ω R_S 1.5	Internal series termination with calibration (50 Ω setting)	V_{CCIO} = 1.5 V	*(2)*	*(2)*	%
	Internal series termination without calibration (50 Ω setting)	V_{CCIO} = 1.5 V	±36	*(2)*	%
50 Ω R_S 1.2	Internal series termination with calibration (50 Ω setting)	V_{CCIO} = 1.2 V	*(2)*	*(2)*	%
	Internal series termination without calibration (50 Ω setting)				

Notes for Table 4–28:

(1) For information on which I/O banks support memory interface IOEs, refer to *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1*.

(2) This specification is pending device characterization.

Table 4–29. Series & Differential On-Chip Termination Specification for I/O Banks Supporting High-Speed & General Purpose IOEs *Note (1)*

Symbol	Description	Conditions	Resistance Tolerance		
			Commercial Max	Industrial Max	Unit
25 Ω R_S 3.3/2.5	Internal series termination without calibration (25 Ω setting)	V_{CCIO} = 3.3/2.5 V	±30	*(2)*	%
50 Ω R_S 3.3/2.5/1.8	Internal series termination without calibration (50 Ω setting)	V_{CCIO} = 3.3/2.5/1.8 V	±30	*(2)*	%
50 Ω R_S 1.5	Internal series termination without calibration (50 Ω setting)	V_{CCIO} = 1.5 V	*(2)*	*(2)*	%
R_D *(3)*	Internal differential termination for LVDS or HyperTransport technology		*(2)*	*(2)*	%

Notes to Table 4–29:
(1) For information on which I/O banks support high-speed IOEs, refer to the *Description, Architecture & Features* chapter in the *HardCopy II Device Family Data Sheet* section of the *HardCopy Series Handbook, volume 1*.
(2) This specification is pending device characterization.
(3) R_D is only supported on high-speed IOEs.

Pin Capacitance

Table 4–30 shows the HardCopy II device family pin capacitance.

Table 4–30. HardCopy II Device Capacitance (Part 1 of 2)

Symbol	Parameter	Minimum	Typical	Maximum	Unit
C_{GPIO}	Input capacitance on I/O pins in I/O banks supporting general-purpose IOEs.		5.0	*(1)*	pF
C_{MIIO}	Input capacitance on I/O pins in I/O banks supporting memory interface IOEs.		5.0	*(1)*	pF
C_{HSIO}	Input capacitance on I/O pins in I/O banks supporting high-speed IOEs.		6.1	*(1)*	pF
C_{CLKTB}	Input capacitance on top/bottom clock input pins `CLK[4..7]` and `CLK[12..15]`.		6.0	*(1)*	pF
C_{CLKLR}	Input capacitance on left/right clock inputs `CLK0`, `CLK2`, `CLK8`, `CLK10`.		6.1	*(1)*	pF
C_{CLKLR+}	Input capacitance on left/right clock inputs `CLK1`, `CLK3`, `CLK9`, and `CLK11`.		3.3	*(1)*	pF

Table 4–30. HardCopy II Device Capacitance (Part 2 of 2)					
Symbol	**Parameter**	**Minimum**	**Typical**	**Maximum**	**Unit**
C_{OUTFB}	Input capacitance on dual-purpose clock output/feedback pins in PLL banks 9, 10, 11, and 12.		6.7	*(1)*	pF

Note to Table 4–30:
(1) This specification is pending device characterization.

Maximum Input & Output Clock Rates

Refer to Tables 4–31 through 4–35 for maximum I/O clock rates.

Table 4–31. HardCopy II Maximum Input Clock Rate for Memory Interface IOEs (Part 1 of 2)		
I/O Standard	**Performance**	**Unit**
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
SSTL-2 class I	400	MHz
SSTL-2 class II	400	MHz
SSTL-18 class I	400	MHz
SSTL-18 class II	400	MHz
1.5-V HSTL class I	400	MHz
1.5-V HSTL class II	400	MHz
1.8-V HSTL class I	400	MHz
1.8-V HSTL class II	400	MHz
PCI *(1)*	420	MHz
PCI-X *(1)*	420	MHz
Differential SSTL-2 class I *(2)*	400	MHz
Differential SSTL-2 class II *(2)*	400	MHz
Differential SSTL-18 class I *(2)*	400	MHz
Differential SSTL-18 class II *(2)*	400	MHz
1.8 V differential HSTL class I *(2)*	400	MHz
1.8 V differential HSTL class II *(2)*	400	MHz
1.5 V differential HSTL class I *(2)*	400	MHz

Table 4–31. HardCopy II Maximum Input Clock Rate for Memory Interface IOEs (Part 2 of 2)

I/O Standard	Performance	Unit
1.5 V differential HSTL class II *(2)*	400	MHz

Notes to Table 4–31:
(1) The PCI clamping diode is only supported on the top and bottom I/O pins.
(2) This I/O standard is only supported on the DQS, CLK, and PLL_FB input pins.

Table 4–32. HardCopy II Maximum Input Clock Rate for General-Purpose IOEs

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
PCI *(1)*	422	MHz
PCI-X *(1)*	422	MHz
Differential SSTL-2 class I *(2)*	400	MHz
Differential SSTL-2 class II *(2)*	400	MHz
Differential SSTL-18 class I *(2)*	400	MHz
Differential SSTL-18 class II *(2)*	400	MHz
1.8 V differential HSTL class I *(2)*	400	MHz
1.8 V differential HSTL class II *(2)*	400	MHz
1.5 V differential HSTL class I *(2)*	400	MHz
1.5- V differential HSTL class II *(2)*	400	MHz
LVDS *(2)*	500	MHz
HyperTransport *(2)*	520	MHz

Notes to Table 4–32:
(1) The PCI clamping diode is only supported on the top and bottom I/O pins.
(2) This I/O standard is only supported on the bottom CLK and PLL_FB input pins.

Table 4–33. HardCopy II Maximum Input Clock Rate for High-Speed IOEs

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
LVDS	500	MHz
HyperTransport	520	MHz

Table 4–34. HardCopy II Maximum Output Clock Rate for Memory Interface IOEs (Part 1 of 4)

I/O Standard	Drive Strength	Performance	Unit
LVTTL	4 mA	350	MHz
	8 mA	350	MHz
	12 mA	350	MHz
	16 mA	350	MHz
	20 mA	350	MHz
	24 mA *(1)*	350	MHz
LVCMOS	4 mA	350	MHz
	8 mA	350	MHz
	12 mA	350	MHz
	16 mA	350	MHz
	20 mA	350	MHz
	24 mA *(1)*	350	MHz
2.5 V	4 mA	350	MHz
	8 mA	350	MHz
	12 mA	350	MHz
	16 mA *(1)*	350	MHz
1.8 V	2 mA	350	MHz
	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz

Table 4–34. HardCopy II Maximum Output Clock Rate for Memory Interface IOEs (Part 2 of 4)

I/O Standard	Drive Strength	Performance	Unit
1.5 V	2 mA	350	MHz
	4 mA	350	MHz
	6 mA	350	MHz
	8 mA *(1)*	350	MHz
SSTL-2 class I	8 mA	350	MHz
	12 mA *(1)*	350	MHz
SSTL-2 class II	16 mA	350	MHz
	20 mA	350	MHz
	24 mA *(1)*	350	MHz
SSTL-18 class I	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz
SSTL-18 class II	8 mA	350	MHz
	16 mA	350	MHz
	18 mA	350	MHz
	20 mA *(1)*	350	MHz
1.8-V HSTL class I	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz
1.8-V HSTL class II	16 mA	350	MHz
	18 mA	350	MHz
	20 mA *(1)*	350	MHz
1.5-V HSTL class I	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz

Table 4–34. HardCopy II Maximum Output Clock Rate for Memory Interface IOEs (Part 3 of 4)

I/O Standard	Drive Strength	Performance	Unit
1.5-V HSTL class II	16 mA	350	MHz
	18 mA	350	MHz
	20 mA *(1)*	350	MHz
PCI *(2)*		350	MHz
PCI-X *(2)*		350	MHz
Differential SSTL-2 class I *(3)*	8 mA	350	MHz
	12 mA *(1)*	350	MHz
Differential SSTL-2 class II *(3)*	16 mA	350	MHz
	20 mA	350	MHz
	24 mA *(1)*	350	MHz
Differential SSTL-18 class I *(3)*	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz
Differential SSTL-18 class II *(3)*	8 mA	350	MHz
	16 mA	350	MHz
	18 mA	350	MHz
	20 mA *(1)*	350	MHz
1.8 V differential HSTL class I *(3)*	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz
1.8 V differential HSTL class II *(3)*	16 mA	350	MHz
	18 mA	350	MHz
	20 mA *(1)*	350	MHz
1.5 V differential HSTL class I *(3)*	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz

Table 4–34. HardCopy II Maximum Output Clock Rate for Memory Interface IOEs (Part 4 of 4)

I/O Standard	Drive Strength	Performance	Unit
1.5 V differential HSTL class II *(3)*	16 mA	350	MHz
	18 mA	350	MHz
	20 mA *(1)*	350	MHz

Notes to Table 4–34:

(1) This is the default setting in Quartus II software if supported by the pin location.
(2) The PCI clamping diode is only supported on the top and bottom I/O pins.
(3) This I/O standard is only supported on the PLL_OUT pins.

Table 4–35. HardCopy II Maximum Output Clock Rate for General-Purpose IOEs (Part 1 of 2)

I/O Standard	Drive Strength	Performance	Unit
LVTTL	4 mA	350	MHz
	8 mA	350	MHz
	12 mA	350	MHz
LVCMOS	4 mA	350	MHz
	8 mA	350	MHz
2.5 V	4 mA	350	MHz
	8 mA	350	MHz
	12 mA	350	MHz
1.8 V	2 mA	350	MHz
	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
1.5 V	2 mA	350	MHz
	4mA	350	MHz
PCI *(2)*		350	MHz
PCI-X *(2)*		350	MHz
Differential SSTL-2 class I *(3)*	8 mA	350	MHz
	12 mA *(1)*	350	MHz
Differential SSTL-2 class II *(3)*	16 mA	350	MHz
	20 mA *(1)*	350	MHz

Table 4–35. HardCopy II Maximum Output Clock Rate for General-Purpose IOEs (Part 2 of 2)

I/O Standard	Drive Strength	Performance	Unit
Differential SSTL-18 class I *(3)*	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
	10 mA	350	MHz
	12 mA *(1)*	350	MHz
LVDS *(3)*		500	MHz
HyperTransport *(3)*		520	MHz

Notes to Table 4–35:
(1) This is the default setting in Quartus II software if supported by the pin location.
(2) The PCI clamping diode is only supported on the top and bottom I/O pins.
(3) This I/O standard is only supported on the bottom PLL_OUT output pins.

Table 4–36. HardCopy II Maximum Output Clock Rate for High-Speed IOEs

I/O Standard	Drive Strength	Performance	Unit
LVTTL	4 mA	350	MHz
	8 mA	350	MHz
	12 mA	350	MHz
LVCMOS	4 mA	350	MHz
	8 mA	350	MHz
2.5 V	4 mA	350	MHz
	8 mA	350	MHz
	12 mA	350	MHz
1.8 V	2 mA	350	MHz
	4 mA	350	MHz
	6 mA	350	MHz
	8 mA	350	MHz
1.5 V	2 mA	350	MHz
	4mA	350	MHz
LVDS		500	MHz
HyperTransport		520	MHz

5. Quartus II Support for HardCopy II Devices

H51022-2.2

HardCopy II Device Support

Altera® HardCopy II devices feature 1.2-V, 90 nm process technology, and provide a structured ASIC alternative to increasingly expensive multi-million gate ASIC designs. The HardCopy II design methodology offers a fast time-to-market schedule, providing ASIC designers with a solution to long ASIC development cycles. Using the Quartus II software, you can leverage a Stratix II FPGA as a prototype and seamlessly migrate your design to a HardCopy II device for production.

This document discusses the following topics:

- HardCopy II design development flow and companion devices
- HardCopy II Device Resource Guide
- Recommended Quartus II software settings
- HardCopy II Utilities menu options and functions

For more information about HardCopy II, HardCopy Stratix, and HardCopy APEX™ devices, refer to the respective device data sheets in the *HardCopy Series Handbook*.

HardCopy II Design Benefits

Designing with HardCopy II structured ASICs offers substantial benefits over other structured ASIC offerings:

- Prototyping using a Stratix II FPGA for functional verification and system development reduces total project development time
- Seamless migration from a Stratix II FPGA prototype to a HardCopy II device reduces time to market and risk
- Unified design methodology for Stratix II FPGA design and HardCopy II design reduces the need for ASIC development software
- Low up-front development cost of HardCopy II devices reduces the financial risk to your project

Quartus II Features for HardCopy II Planning

With the Quartus II software you can design a HardCopy II device using a Stratix II device as a prototype. The Quartus II software contains the following expanded features for HardCopy II device planning:

- **HardCopy II Companion Device Assignment**—Identifies compatible HardCopy II devices for migration with the Stratix II device currently selected.

  This feature constrains the pins of your Stratix II FPGA prototype making it compatible with your HardCopy II device. It also constrains the correct resources available for the HardCopy II device making sure that your Stratix II FPGA design does not become incompatible.

- **HardCopy II Utilities**—The HardCopy II Utilities functions create or overwrites HardCopy II companion revisions, change revisions to use, and compare revisions for equivalency.
- **HardCopy II Advisor**—The HardCopy II Advisor helps you follow the necessary steps to successfully submit a HardCopy II design to Altera's HardCopy Design Center.

  The HardCopy II Advisor is similar to the Resource Optimization Advisor and Timing Optimization Advisor. The HardCopy II Advisor provides guidelines you can follow during development, reporting the tasks completed as well as the tasks that you still need to complete during development.

- **HardCopy II Floorplan**—The Quartus II software can show a preliminary floorplan view of your HardCopy II design's Fitter placement results.
- **HardCopy II Design Archiving**—The Quartus II software archives the HardCopy II design project's files needed to handoff the design to the HardCopy Design Center.

  This feature is similar to the Quartus II software HardCopy Files Wizard used for HardCopy Stratix and HardCopy APEX families.

- **HardCopy II Device Preliminary Timing**—The Quartus II software performs a timing analysis of HardCopy II devices based on preliminary timing models and Fitter placements. Final timing results for HardCopy II devices are provided by the HardCopy Design Center.

- **HardCopy II Handoff Report**—The Quartus II software generates a handoff report containing information about the HardCopy II design used by the HardCopy Design Center in the design review process.
- **Formal Verification**—Cadence Encounter Conformal software can now perform formal verification between the source RTL design files and post-compile gate level netlist from a HardCopy II design.

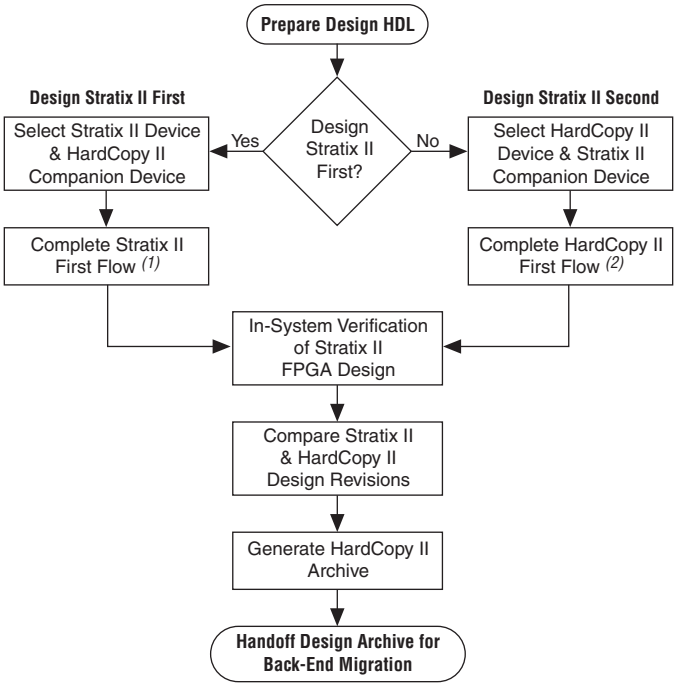
HardCopy II Development Flow

In the Quartus II software, you have two methods for designing your Stratix II FPGA and HardCopy II companion device together in one Quartus II project.

- Design the HardCopy II device first, and create the Stratix II FPGA companion device second and build your prototype for in-system verification
- Design the Stratix II FPGA first and create a HardCopy II companion device second

Both of these flows are illustrated at a high level in Figure 5–1. The added features in the HardCopy II Utilities menu assist you in completing your HardCopy II design for submission to Altera's HardCopy Design Center for back-end implementation.

Figure 5–1. HardCopy II Flow in Quartus II Software



Notes for Figure 5–1:
(1) Refer to Figure 5–2 for an expanded description of this process.
(2) Refer to Figure 5–3 for an expanded description of this process.

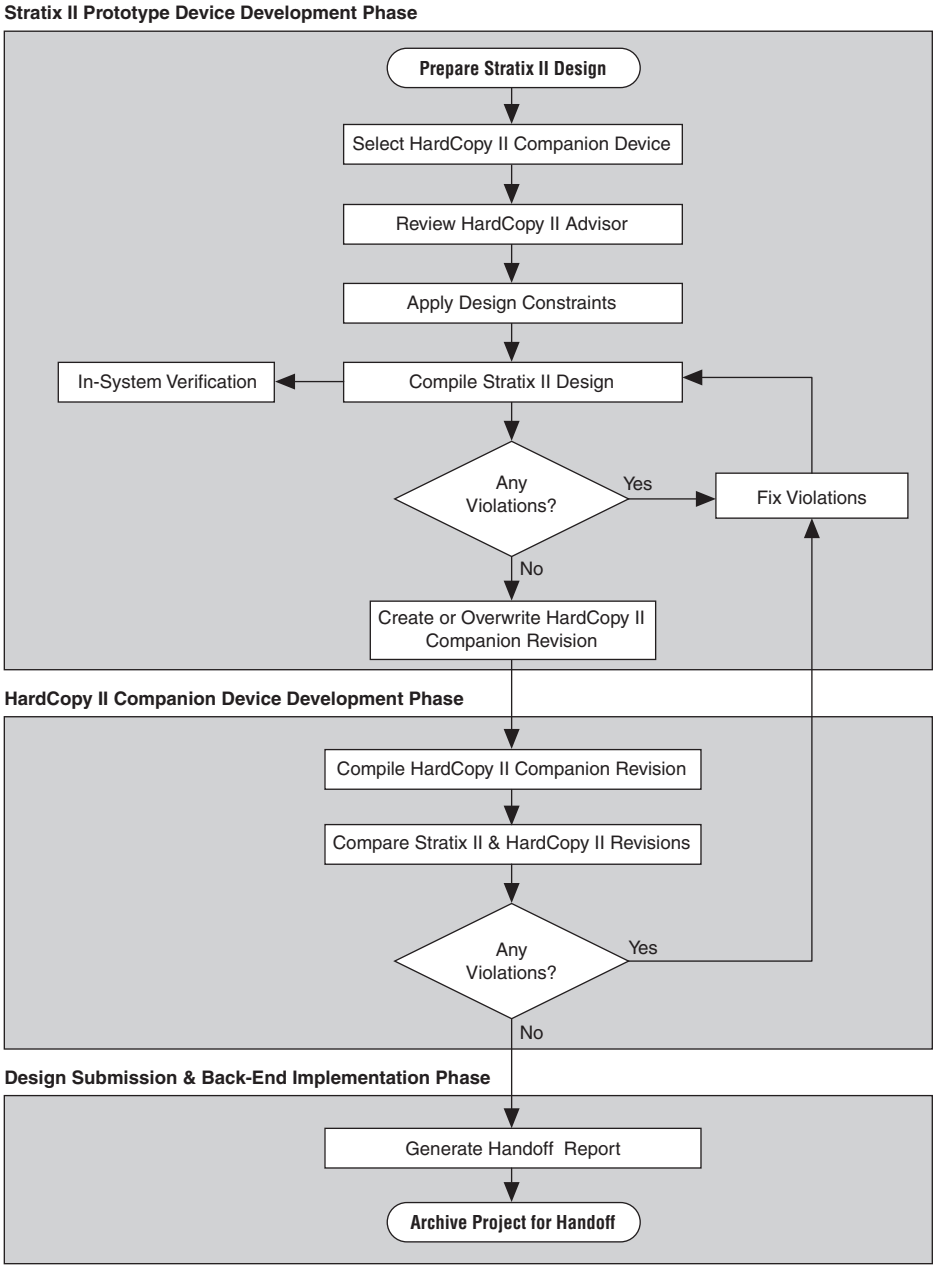
Designing the Stratix II FPGA First

The HardCopy II development flow beginning with the Stratix II FPGA prototype is very similar to a traditional Stratix II FPGA design flow, but requires a few additional tasks to be performed to migrate the design to the HardCopy II companion device. To design your HardCopy II device using the Stratix II FPGA as a prototype, complete the following tasks:

- Specify a HardCopy II device for migration
- Compile the Stratix II FPGA design
- Create and compile the HardCopy II companion revision
- Compare the HardCopy II companion revision compilation to the Stratix II device compilation

Figure 5–2 provides an overview highlighting the development process for designing with a Stratix II FPGA first and creating a HardCopy II companion device second.

Figure 5–2. Designing Stratix II Device First Flow



Prototype your HardCopy II design by selecting and then compiling a Stratix II device in the Quartus II software.

Once you compile the Stratix II design successfully, you can view the HardCopy II Device Resource Guide in the Quartus II software Fitter report to evaluate which HardCopy II devices meet your design's resource requirements. When you are satisfied with the compilation results and the choice of Stratix II and HardCopy II devices, on the Assignments menu, click **Settings**. In the **Category** list, select **Device**. In the **Device** page, select a HardCopy II companion device.

After you select your HardCopy II companion device, do the following:

- Review the HardCopy II Advisor for required and recommended tasks to perform
- Enable Design Assistant to run during compilation
- Add timing and location assignments
- Compile your Stratix II design
- Create your HardCopy II companion revision
- Compile your design for the HardCopy II companion device
- Use the HardCopy II Utilities to compare the HardCopy II companion device compilation with the Stratix II FPGA revision
- Generate a HardCopy II Handoff Report using the HardCopy II Utilities
- Generate a HardCopy II Handoff Archive using the HardCopy II Utilities
- Arrange for submission of your HardCopy II handoff archive to Altera's HardCopy Design Center for back-end implementation

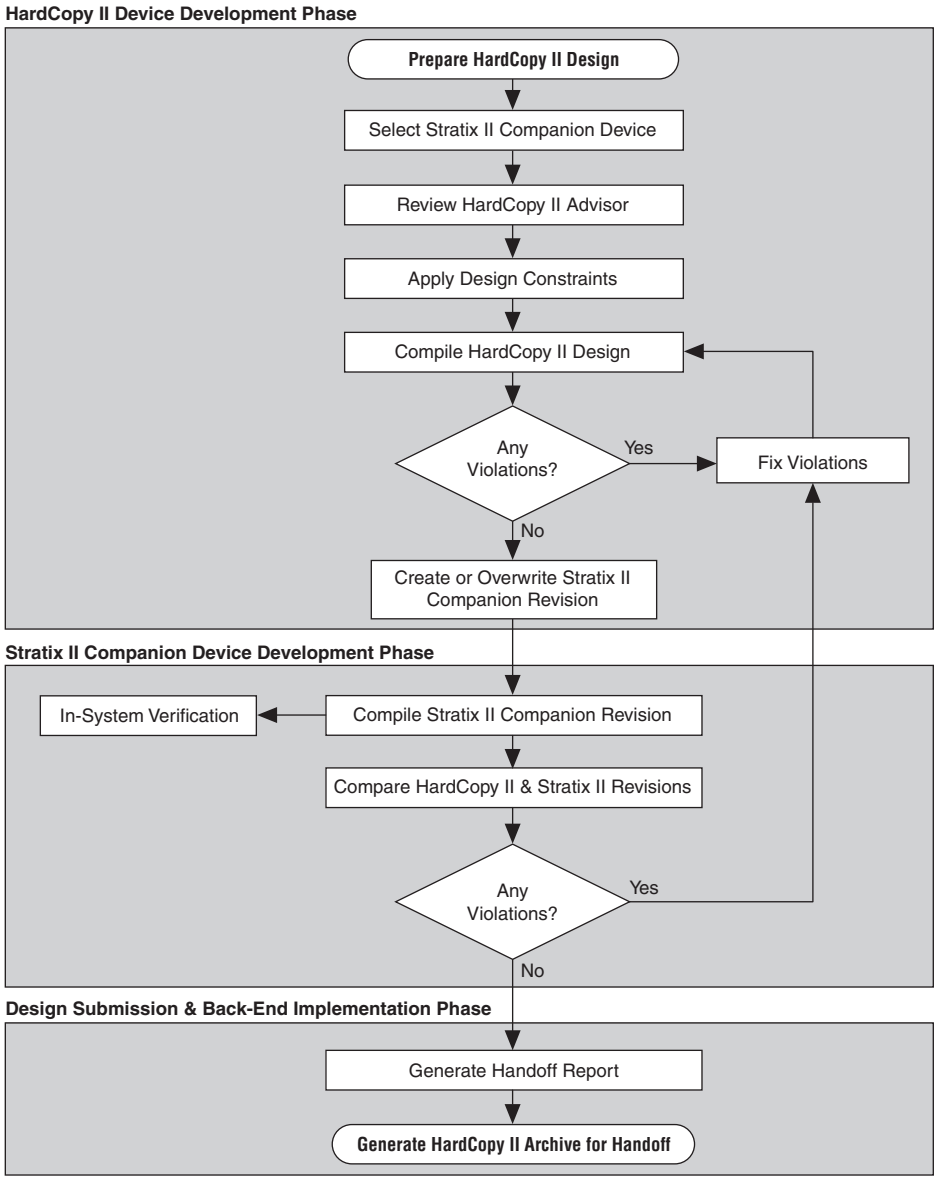
For more information about the overall design flow using the Quartus II software, refer to the *Introduction to Quartus II* manual on the Altera web site at www.altera.com.

Designing the HardCopy II Device First

The HardCopy II family presents a new option in designing unavailable in previous HardCopy families. You can design your HardCopy II device first and create your Stratix II FPGA prototype second in the Quartus II software. This allows you to see your potential maximum performance in the HardCopy II device immediately during development, and you can create a slower performing FPGA prototype of the design for in-system verification. This design process is similar to the traditional HardCopy II design flow where you build the FPGA first, but instead, you merely change the starting device family. The remaining tasks to complete your design for both Stratix II and HardCopy II devices roughly follow the

same process (Figure 5–3). The HardCopy II Advisor adjusts its list of tasks based on which device family you start with, Stratix II or HardCopy II, so that you can complete the process seamlessly.

Figure 5–3. Designing HardCopy II Device First Flow



HardCopy II Device Resource Guide

The HardCopy II Device Resource Guide compares the resources required to successfully compile a design with the resources available in the various HardCopy II devices. The report rates each HardCopy II device and each device resource for how well it fits the design. The Quartus II software generates the HardCopy II Device Resource Guide for all designs successfully compiled for Stratix II devices, and is found in the Fitter folder of the Compilation Report. Figure 5–4 shows an example of the HardCopy II Device Resource Guide. Refer to Table 5–1 for an explanation of the color codes in Figure 5–4.

Figure 5–4. HardCopy II Device Resource Guide

HardCopy II Device Resource Guide

Color Legend:
-- Green:
-- Package Resource: The HardCopy II package can be migrated from the Stratix II FPGA selected package, and the design has been fitted with the target device migration enabled.

	Resource	Stratix II EP2S130	HC210W*	HC210	HC220	HC220	HC230	HC240	HC240
1	Migration Compatibility		None	None	None	None	Medium	None	None
2	Primary Migration Constraint		Package	Package	Package	Package	Package	Package	Package
3	Package	FBGA - 1020	FBGA - 484	FBGA - 484	FBGA - 672	FBGA - 780	FBGA - 1020	FBGA - 1020	FBGA - 1508
4	⊟ Logic	--	19%	19%	10%	10%	6%	4%	4%
5	-- Logic cells	35572 ALUTs	--	--	--	--	--	--	--
6	-- DSP elements	0	--	--	--	--	--	--	--
7	⊟ Pins								
8	-- Total	515	515 / 302	515 / 335	515 / 493	515 / 495	515 / 699	515 / 743	515 / 952
9	-- Differential Input	0	0 / 66	0 / 70	0 / 90	0 / 90	0 / 128	0 / 224	0 / 272
10	-- Differential Output	0	0 / 44	0 / 50	0 / 70	0 / 70	0 / 112	0 / 200	0 / 256
11	-- PCI / PCI-X	0	0 / 153	0 / 167	0 / 245	0 / 247	0 / 359	0 / 367	0 / 472
12	-- DQ	0	0 / 20	0 / 20	0 / 50	0 / 50	0 / 204	0 / 204	0 / 204
13	-- DQS	0	0 / 8	0 / 8	0 / 18	0 / 18	0 / 72	0 / 72	0 / 72
14	⊟ Memory								
15	-- M-RAM	6	6 / 0	6 / 0	6 / 2	6 / 2	6 / 6	6 / 9	6 / 9
16	-- M4K blocks & M512 blocks**	44	44 / 190	44 / 190	44 / 408	44 / 408	44 / 614	44 / 816	44 / 816
17	⊟ PLLs								
18	-- Enhanced	2	2 / 2	2 / 2	2 / 2	2 / 2	2 / 4	2 / 4	2 / 4
19	-- Fast	0	0 / 2	0 / 2	0 / 2	0 / 2	0 / 4	0 / 8	0 / 8
20	DLLs	0	0 / 1	0 / 1	0 / 1	0 / 1	0 / 2	0 / 2	0 / 2
21	⊟ SERDES								
22	-- RX	0	0 / 17	0 / 21	0 / 31	0 / 31	0 / 46	0 / 92	0 / 116
23	-- TX	0	0 / 18	0 / 19	0 / 29	0 / 29	0 / 44	0 / 88	0 / 116
24	⊟ Configuration								
25	-- CRC	0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0
26	-- ASMI	0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0
27	-- Remote Update	0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0
28	-- JTAG	0	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1

* Device is preliminary. Overall performance is expected to be degraded.
** Design contains one or more M512 blocks, which cannot be migrated to HardCopy II devices.

Use this report to determine which HardCopy II device is a potential candidate for migration of your Stratix II design. The HardCopy II device package must be compatible with the Stratix II device package. A logic

resource usage greater than 100% or a ratio greater than 1/1 in any category indicates that the design does not fit in that particular HardCopy II device.

Table 5–1. HardCopy II Device Resource Guide Color Legend

Color	Package Resource *(1)*	Device Resources
Green (High)	The design can migrate to the Hardcopy II package and the design has been fit with target device migration enabled in the **HardCopy II Companion Device** dialog box.	The resource quantity is within the range of the HardCopy II device and the design can likely migrate if all other resources also fit.
Orange (Medium)	The design can migrate to the Hardcopy II package. However, the design has not been fit with target device migration enabled in the **HardCopy II Companion Device** dialog box.	The resource quantity is within the range of the HardCopy II device. However, the resource is at risk of exceeding the range for the HardCopy II package. Consult your Product Field Applications Engineer for a recommended course of action.
Red (None)	The design cannot migrate to the Hardcopy II package.	The resource quantity exceeds the range of the HardCopy II device. The design cannot migrate to this HardCopy II device.

Note to Table 5–1:

(1) The package resource is constrained by the Stratix II FPGA that the design was compiled for. Only vertical migration devices within the same package are able to migrate to HardCopy II devices.

The HardCopy II architecture consists of an array of fine-grained HCells, which are used to build logic equivalent to Stratix II adaptive logic modules (ALMs) and digital signal processing (DSP) blocks. The DSP blocks in HardCopy II devices match the functionality of the Stratix II DSP blocks, though timing of these blocks will be different than the FPGA since they are constructed of HCell Macros. The M4K and M-RAM memory blocks in HardCopy II devices are equivalent to the Stratix II memory blocks. Preliminary timing reports of the HardCopy II device are available in the Quartus II software. Final timing results of the HardCopy II device are provided by the HardCopy Design Center after back-end migration is complete.

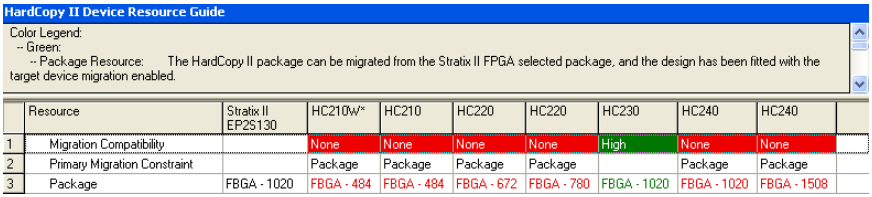
For more information about the HardCopy II device resources, refer to the *Introduction to HardCopy II Devices* and the *Description, Architecture & Features* chapters in the *HardCopy II Device Family Data Sheet* in the *HardCopy Series Handbook*.

The report example in Figure 5–4 shows the resource comparisons for a design compiled for a Stratix II EP2S130F1020 device. Based on the report, the HC230F1020 device in the 1,020-pin FineLine BGA® package is an appropriate HardCopy II device to migrate to. If the HC230F1020 device was not specified as a migration target during the compilation, its package and migration compatibility would be rated orange or Medium.

The migration compatibility of the other HardCopy II devices are rated red, or None, because the package types are incompatible with the Stratix II device. The 1,020-pin FBGA HC240 device is rated red because it is only compatible with the Stratix II EP2S180F1020 device.

Figure 5–5 shows the report after the (unchanged) design was recompiled with the HardCopy II HC230F1020 device specified as a migration target. Now the HC230F1020 device package and migration compatibility are rated green or High.

Figure 5–5. HardCopy II Device Resource Guide with Target Migration Enabled


HardCopy II Device Resource Guide

Color Legend:
-- Green:
-- Package Resource: The HardCopy II package can be migrated from the Stratix II FPGA selected package, and the design has been fitted with the target device migration enabled.

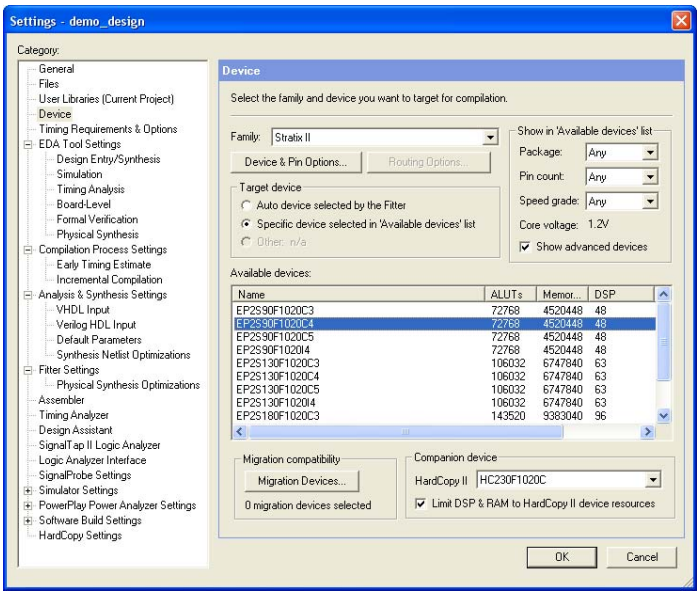
	Resource	Stratix II EP2S130	HC210W*	HC210	HC220	HC220	HC230	HC240	HC240
1	Migration Compatibility		None	None	None	None	High	None	None
2	Primary Migration Constraint		Package	Package	Package	Package		Package	Package
3	Package	FBGA - 1020	FBGA - 484	FBGA - 484	FBGA - 672	FBGA - 780	FBGA - 1020	FBGA - 1020	FBGA - 1508

HardCopy II Companion Device Selection

In the Quartus II software, you can select a HardCopy II companion device to help structure your design for migration from a Stratix II device to a HardCopy II device. To make your HardCopy II companion device selection, on the Assignments menu, click **Settings**. In the **Settings** dialog box in the **Category** list, select **Device** (Figure 5–6) and select your companion device from the **Available devices** list.

Selecting a HardCopy II Companion device to go with your Stratix II prototype constrains the memory blocks, DSP blocks, and pin assignments, so that your Stratix II and HardCopy II devices are migration-compatible. Pin assignments are constrained in the Stratix II design revision so that the HardCopy II device selected is pin-compatible. The Quartus II software also constrains the Stratix II design revision so it does not use M512 memory blocks or exceed the number of M-RAM blocks in the HardCopy II companion device.

Figure 5–6. Quartus II Settings Dialog Box



You can also specify your HardCopy II companion device using the following Tcl command:

```
set_global_assignment -name
DEVICE_TECHNOLOGY_MIGRATION_LIST <HardCopy II Device Part Number>
```

For example, to select the HC230F1020 device as your HardCopy II companion device for the EP2S130F1020C4 Stratix II FPGA, the Tcl command is:

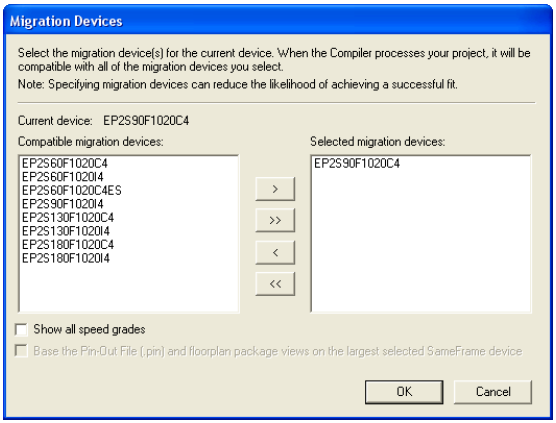
```
set_global_assignment -name
DEVICE_TECHNOLOGY_MIGRATION_LIST HC230F1020
```

Migration Compatibility Filtering

The **Migration Devices** dialog box displays which devices are vertically migratable within the same package and family for all Altera devices. When you are designing for HardCopy II devices with a Stratix II prototype device, the **Migration Devices** dialog box filters the compatible devices between Stratix II devices and HardCopy II devices within the same package.

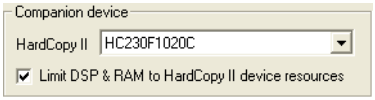
To view all Stratix II devices that are vertically migratable to a Stratix II device, on the Assignments menu, click **Settings**. In the **Category** list, select **Device** and on the **Device** page, in the **Family** list select **Stratix II**. In the **Available devices** list, select the desired device. Under **Companion device** in the **HardCopy II** list, select **<None>**, and click **Migration Devices**. The **Migration Devices** dialog box shows the Stratix II devices that are vertically migratable to the currently selected Stratix II device (Figure 5–7).

Figure 5–7. Available Migration Devices without Selecting a HardCopy II Device



Without HardCopy II companion device constraints, all Stratix II devices in the 1,020-pin FineLine BGA package are available for vertical migration. Selecting a HardCopy II companion device in the **Device** page, as shown in Figure 5–8, filters the list of migration devices to only those Stratix II devices that are vertically migratable within the same package and are usable as HardCopy II prototype devices.

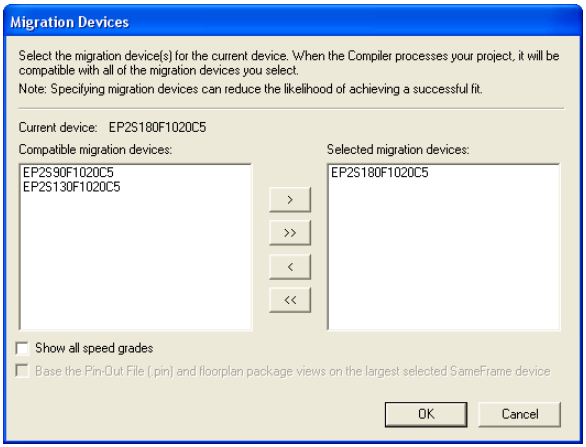
Figure 5–8. Setting a HardCopy II Companion Device



For example, if you select the HC230F1020 device as the companion device, the **Migration Devices** dialog box shows the EP2S90F1020C4 and EP2S180F1020C4 devices as possible companion devices to the EP2S130F1020C4 device currently selected (Figure 5–9). However, the

EP2S60F1020C4 device is not a compatible device to the HC230F1020 device, even though it is in the same package, so it is not listed in the **Migration Devices** dialog box.

Figure 5–9. Available Migration Devices after Selecting a HardCopy II Device



HardCopy II Recommended Settings in the Quartus II Software

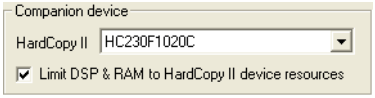
The HardCopy II development flow involves additional planning and preparation in the Quartus II software compared to a standard FPGA design. This is because you are developing your design to be implemented in two devices: a prototype of your design in a Stratix II prototype FPGA, and a companion revision in a HardCopy II device for production. You need additional settings and constraints to make the Stratix II design compatible with the HardCopy II device and, in some cases, you must remove certain settings in the design. This section explains the additional settings and constraints necessary for your design to be successful in both Stratix II FPGA and HardCopy II structured ASIC devices.

Limit DSP & RAM to HardCopy II Device Resources

On the Assignments menu, click **Settings** to view the **Settings** dialog box. In the **Category** list, select **Device**. In the **Family** list, select **Stratix II**. Under **Companion device**, **Limit DSP & RAM to HardCopy II device resources** is turned on by default (Figure 5–10). This maintains compatibility between the Stratix II and HardCopy II devices by ensuring your design does not use resources in the Stratix II device that are not available in the selected HardCopy II device.

If you require additional memory blocks or DSP blocks for debugging purposes using SignalTap® II, you can temporarily turn this setting off to compile and verify your design in your test environment. However, your final Stratix II and HardCopy II designs submitted to Altera for back-end migration must be compiled with this setting turned on.

Figure 5–10. Limit DSP & RAM to HardCopy II Device Resources Check Box



Enable Design Assistant to Run During Compile

You must use the Quartus II Design Assistant to check all HardCopy series designs for design rule violations before submitting the designs to the Altera HardCopy Design Center. Additionally, you must fix all critical and high-level errors.

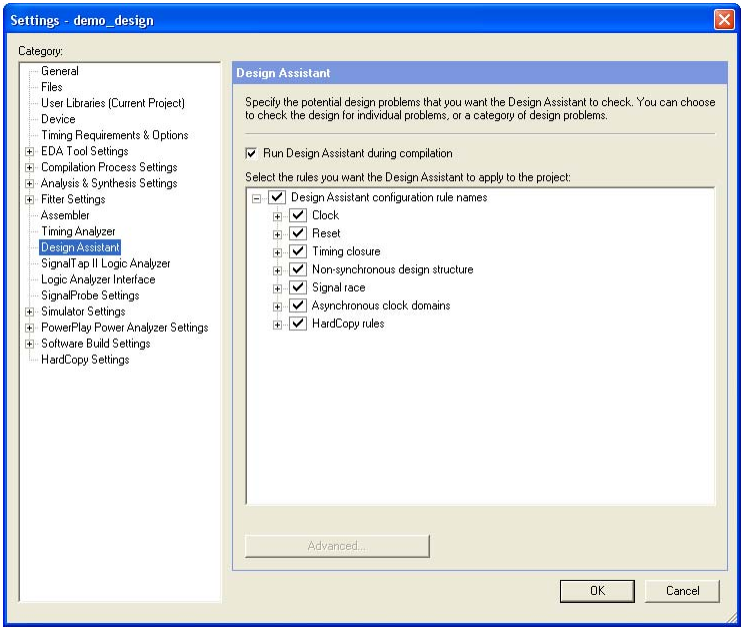
Altera recommends turning on the Design Assistant to run automatically during each compile, so that during development, you can see the violations you must fix.

For more information about the Design Assistant and the rules it uses, refer to the *Design Guidelines for HardCopy Series Devices* chapter of the *HardCopy Series Handbook*.

To enable the Design Assistant to run during compilation, on the Assignment menu, click **Settings**. In the **Category** list, select **Design Assistant** and turn on **Run Design Assistant during compilation** (Figure 5–11) or by entering the following Tcl command in the Tcl Console:

```
set_global_assignment -name ENABLE_DRC_SETTINGS ON
```

Figure 5–11. Enabling Design Assistant



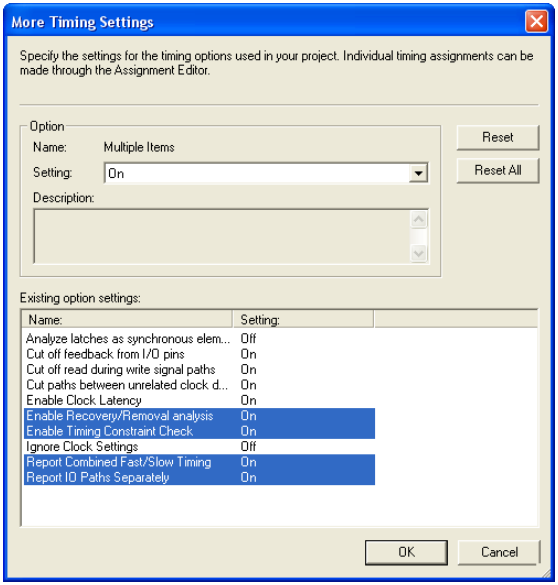
Timing Settings

In the **More Timing Settings** dialog box, you can specify optional timing settings, some of which are crucial to HardCopy II development. To specify these options, on the Assignments menu, click **Settings**. In the **Category** list, select **Timing Requirements & Options** and click **More Settings**. In the More Settings dialog box, set the desired timing settings (Figure 5–12).

For Stratix II and HardCopy II co-development, Altera recommends that you turn on the following settings:

- Enable Clock Latency
- Enable Recovery/Removal analysis
- Enable Timing Constraint Check
- Report Combined Fast/Slow Timing
- Report IO Paths Separately

Figure 5–12. More Timing Settings



Enable Clock Latency

Turning on the **Enable Clock Latency** option enables support for clock latency in the Timing Analyzer. Latency on a clock is a delay on the clock path and affects clock skew. This is different from an offset, which instead alters the setup relationship between two clocks.

When you enable clock latency, the design adjusts for early and late clock latency assignments. The phase-locked loop (PLL) compensation delay is analyzed as latency and does not affect the offset. For clock settings where you have not specified an offset, the design automatically treats computed offset as latency. By using latency for these automatically calculated clock offsets, the setup relationship for registers driven by these clocks does not vary with routing. This can potentially remove the need for multicycle assignments, as well as improve results by ensuring that timing results are more consistent for each Fitter iteration.

Once enabled, you might need to add, modify, or remove multicycle assignments for the PLL output clocks because of the potential change in the setup relationship for these clocks.

Use the following Tcl command to enable clock latency:

```
set_global_assignment -name ENABLE_CLOCK_LATENCY ON
```

Enable Recovery/Removal Analysis

This setting allows the Quartus II Timing Analysis tool to calculate recovery and removal times on control and reset signals. The recovery time is the minimum length of time that an asynchronous control input pin must be stable before the clock active edge. The removal time is the minimum length of time that an asynchronous control input pin must be stable after the clock active edge.

Altera recommends that you turn on register recovery/removal analysis in the Timing Analysis tool during development for more complete recovery/removal analysis of all logic paths in your design. However, if your design does not have a timing requirement for reset logic this option may be turned off.

Use the following Tcl command to enable recovery and removal analysis:

```
set_global_assignment -name \
ENABLE_RECOVERY_REMOVAL_ANALYSIS ON
```

Enable Timing Constraint Check

The **Enable Timing Constraint Check** setting enables the Timing Analysis tool to review your timing constraints for complete minimum and maximum timing coverage for all inputs, outputs, and bidirectional pins, as well as clock settings for all clock sources. Asynchronous pins such as resets and static control signals are also checked for minimum and maximum delay constraints. You must perform this check and review the results before handoff of the design to the HardCopy Design Center.

Use the following Tcl command to enable Timing Constraint Check:

```
set_global_assignment -name \
FLOW_ENABLE_TIMING_CONSTRAINT_CHECK ON
```

Report Combined Fast/Slow Timing

The Quartus II software can perform a separate timing analysis for worst-case and best-case conditions as independent reports. The **Report Combined Fast/Slow Timing** setting allows the Quartus II software to report slow corner delay case and fast corner delay case timing in one combined report. This setting provides a better timing report for your design by allowing you to see all hold-time issues as well as setup issues in one report. This report is required for HardCopy II device

development. Turning on the **Report Combined Fast/Slow Timing** setting requires the Quartus II software to run the Timing Analyzer twice, once for the fast corner delay model and once for the slow corner delay model.

Use the following Tcl command to enable the **Report Combined Fast/Slow Timing** setting:

```
set_global_assignment -name DO_COMBINED_ANALYSIS ON
```

Report IO Paths Separately

Turn on the **Report IO Paths Separately** setting to create separate report panels for I/O paths constrained by the `INPUT_MAX_DELAY`, `INPUT_MIN_DELAY`, `OUTPUT_MAX_DELAY`, or `OUTPUT_MIN_DELAY` parameters. To specify these constraints, on the Assignments menu, click **Assignment Editor**. By default, I/O paths are reported in the **Clock Setup** and **Clock Hold** sections of the **Timing Analyzer** compilation report.

Altera recommends that you turn on the **Report IO Paths Separately** setting to make it easier to view the I/O timing analysis reports for each device pin. This is optional in FPGA designs, but is helpful for HardCopy II development because the I/O timing requirements you specify must be met in both Stratix II I/O timing and HardCopy II I/O timing results. This setting helps to guarantee drop-in compatibility between your Stratix II FPGA prototype and your HardCopy II structured ASIC.

Use the following Tcl command to enable the **Report IO Paths Separately** setting:

```
set_global_assignment -name \
REPORT_IO_PATHS_SEPARATELY ON
```

Quartus II Software Version 6.0 Features Supported for HardCopy II Designs

The Quartus II software supports optimization features for HardCopy II prototype development including:

- Physical Synthesis Optimization
- LogicLock Regions
- PowerPlay Power Analyzer

Physical Synthesis Optimization

To enable the Physical Synthesis Optimizations for the Stratix II FPGA revision of the design, on the Assignments menu, click **Settings**. In the **Settings** dialog box, in the **Category** list, select **Fitter Settings**. These optimizations get migrated into the HardCopy II companion revision for placement and timing closure. When designing with a HardCopy II device first, physical synthesis optimizations can be enabled for the HardCopy II device, and these post-fit optimizations get migrated to the Stratix II FPGA revision.

LogicLock Regions

The use of LogicLock Regions in the Stratix II FPGA are supported for designs migrating to HardCopy II. However, the LogicLock Regions are not passed into the HardCopy II Companion Revision. You can use LogicLock in the HardCopy II design but you must create new LogicLock Regions in the HardCopy II companion revision. In addition, LogicLock Regions in HardCopy II devices can not have their properties set to Auto Size or Floating Location. HardCopy II LogicLock Regions must be manually sized and placed in the floorplan. When LogicLock Regions are created in a HardCopy II device, they start with width and height dimensions set to (1,1), and the origin coordinates for placement are at X1_Y1 in the lower left corner of the floorplan. You must adjust the size and location of your LogicLock Regions created in the HardCopy II device before compiling the design.

For information about using LogicLock Regions, refer to the *LogicLock Design Methodology*, chapter in volume 2 of the *Quartus II Handbook* on the Altera web site at www.altera.com.

PowerPlay Power Analyzer

You can perform power estimation and analysis of your HardCopy II and Stratix II devices using the PowerPlay Early Power Estimator and PowerPlay Power Analyzer for more accurate estimation of your device's power consumption. The PowerPlay Early Power Estimation is available in the Quartus II software version 5.1 and later. The PowerPlay Power Analyzer supports HardCopy II devices in version 6.0 and later of the Quartus II software.

For more information about using the PowerPlay Power Analyzer, refer to the *PowerPlay Power Analysis* chapter in volume 3 of the *Quartus II Handbook*.

Quartus II Features Not Presently Supported for HardCopy II Designs

The Quartus II software version 6.0 does not support HardCopy II devices with all of the advanced design features available for other Altera devices. Many of these features are scheduled for subsequent releases of the Quartus II software.

The Quartus II software version 6.0 does not support the following features for HardCopy II prototype development using the Stratix II FPGA:

- Incremental compilation (Synthesis and Fitter)
- Maximum fan-out assignments

Chip Editor for HardCopy II Devices

When using the Quartus II Chip Editor for your HardCopy II design, the Chip Editor changes are done in the following two ways:

- A Chip Editor change is applied to a compiled Stratix II design revision and a new HardCopy II Companion Revision is created afterwards, incorporating the Chip Editor modifications.
- A Chip Editor change is performed separately on compiled, existing Stratix II and HardCopy II design revisions. No new companion revisions are created.

If you want to use the Quartus II Chip Editor on a Stratix II design you want to migrate to a HardCopy II device, it is best if you start with a compiled Stratix II project and a new HardCopy II Companion Revision created or overwritten using the HardCopy II Utilities.

Using the Chip Editor on a compiled HardCopy II design revision, requires that you manually complete the changes in both HardCopy II and Stratix II revisions, and then use the HardCopy II Companion Comparison Utility and third-party formal verification software to determine if they are equivalent.

The Chip Editor for HardCopy II has the following enabled features:

- Add/Modify/Remove an HCell Macro of a Combinational Function, Register, or Adder/Subtractor and connect wires to them
- Create new wires in the design
- Edit IO Cell properties such as drive strength or programmable delay values
- Edit PLL settings such as M/N counter settings or phase shift of derived clocks

For more information about using the Quartus II Chip Editor, refer to the *Engineering Change Management* chapter in volume 1 of the *Quartus II Handbook*.

Formal Verification of Stratix II & HardCopy II Revisions

Third party formal verification software is available for your HardCopy II design. Cadence Encounter Conformal verification software is used for Stratix II and HardCopy II families, as well as several other Altera product families.

In order to use the Conformal software with the Quartus II software project for your Stratix II and HardCopy II design revisions, you must enable the **EDA Netlist Writer.** It is necessary to turn on the EDA Netlist Writer so it can generate the necessary netlists and command files needed to run the Conformal software. To automatically run the EDA Netlist Writer during the compile of your Stratix II and HardCopy II design revisions, perform the following steps:

1. On the Assignment menu, click **EDA Tool Settings**. The **Settings** dialog box displays.
2. In the **EDA Tool Settings** list, select **Formal Verification**, and in the **Tool name** list, select **Conformal LEC**.
3. Compile your Stratix II and Hardcopy II design revisions, with both the EDA Tool Settings and the Conformal LEC turned on so the EDA Netlist Writer automatically runs.

The Quartus II EDA Netlist Writer produces one netlist for Stratix II when it is run on that revision, and generates a second netlist when it runs on the HardCopy II revision. You can compare your Stratix II post-compile netlist to your RTL source code using the scripts generated by the EDA Netlist Writer. Similarly, you can compare your HardCopy II post-compile netlist to your RTL source code with scripts provided by the EDA Netlist Writer.

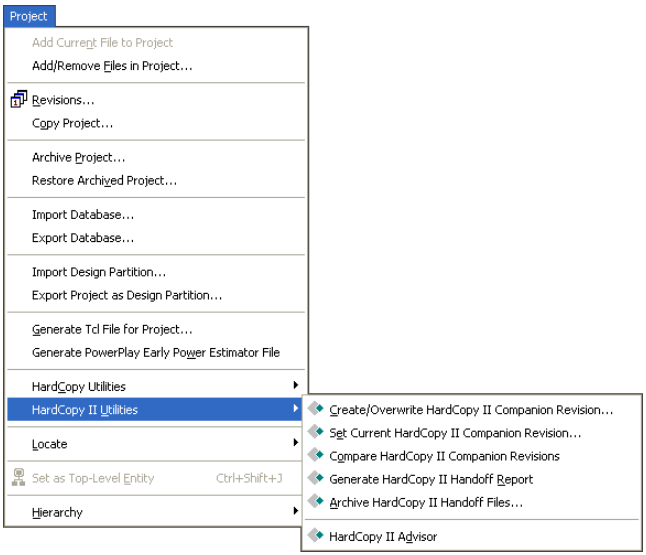
For more information about using the Cadence Encounter Conformal verification software, refer to the *Cadence Encounter Conformal Support* chapter in volume 3 of the *Quartus II Handbook*.

HardCopy II Utilities Menu

The **HardCopy II Utilities** menu is shown in the Quartus II software (Figure 5–13). To access this menu, on the Project menu, click **HardCopy II Utilities**. This menu contains the main functions you use to develop your HardCopy II design and Stratix II FPGA prototype companion revision. From the HardCopy II Utilities menu, you can:

- Create or update HardCopy II companion revisions
- Set which HardCopy II companion revision is the current revision
- Generate HardCopy II Handoff Report for design reviews
- Archive HardCopy II Handoff Files for submission to the HardCopy Design Center
- Compare the companion revisions for functional equivalence
- Track your design progress using the HardCopy II Advisor

Figure 5–13. HardCopy II Utilities Menu



Each of the features within the **HardCopy II Utilities** is summarized in Table 5–2. The process for using each of these features is explained in the following sections.

Table 5–2. HardCopy II Utilities Menu Options

Menu	Description	Applicable Design Revision	Restrictions
Create/Overwrite HardCopy II Companion Revision	Create a new companion revision or update an existing companion revision for your Stratix II and HardCopy II design.	Stratix II prototype design and HardCopy II Companion Revision	• Must disable Auto Device selection • Must set a Stratix II device and a HardCopy II companion device
Set Current HardCopy II Companion Revision	Specify which companion revision to associate with current design revision.	Stratix II prototype design and HardCopy II Companion Revision	Companion Revision must already exist
Compare HardCopy II Companion Revisions	Compares the Stratix II design revision with the HardCopy II companion design revision and generates a report.	Stratix II prototype design and HardCopy II Companion Revision	Compilation of both revisions must be complete
Generate HardCopy II Handoff Report	Generate a report containing important design information files and messages generated by the Quartus II compile	Stratix II prototype design and HardCopy II Companion Revision	• Compilation of both revisions must be complete • Compare HardCopy II Companion Revisions must have been executed
Archive HardCopy II Handoff Files	Generate a Quartus II Archive File specifically for submitting the design to the HardCopy Design Center. Similar to the HardCopy Files Wizard for HardCopy Stratix and APEX.	HardCopy II Companion Revision	• Compilation of both revisions must be completed • Compare HardCopy II Companion Revisions must have been executed • Generate HardCopy Handoff Report must have been executed
HardCopy II Advisor	Open an Advisor, similar to the Resource Optimization Advisor, helping you through the steps of creating a HardCopy II project.	Stratix II prototype design and HardCopy II Companion Revision	None

Companion Revisions

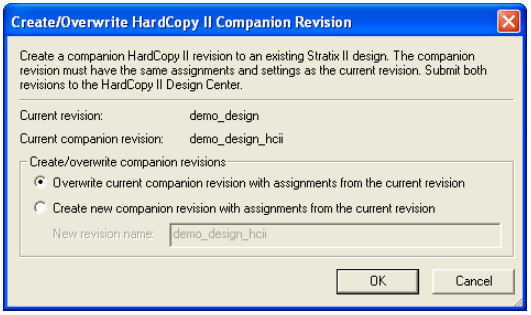
HardCopy II designs follow a different development flow in the Quartus II software compared with previous HardCopy families. You can create multiple revisions of your Stratix II prototype design, but you can also create separate revisions of your design for a HardCopy II device.

The Quartus II software creates specific HardCopy II design revisions of the project in conjunction to the regular project revisions. These parallel design revisions for HardCopy II devices are called companion revisions.

Although you can create multiple project revisions, Altera recommends that you maintain only one Stratix II FPGA revision once you have created the HardCopy II *companion revision*.

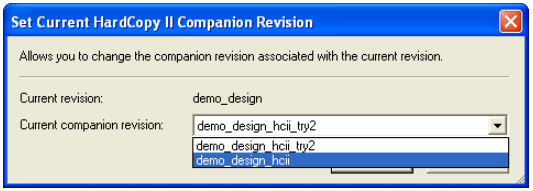
When you have successfully compiled your Stratix II prototype FPGA, you can create a HardCopy II companion revision of your design and proceed with compiling the HardCopy II companion revision. To create a companion revision, on the Project menu, point to HardCopy II Utilities and click **Create/Overwrite HardCopy II Companion Revision**. Use the dialog box to create a new companion revision or overwrite an existing companion revision (Figure 5–14).

Figure 5–14. Create or Overwrite HardCopy II Companion Revision



You can associate only one Stratix II revision to one HardCopy II companion revision. If you created more than one revision or more than one companion revision, set the current companion for the revision you are working on. On the Project menu, point to HardCopy II Utilities and click **Set Current HardCopy II Companion Revision** (Figure 5–15).

Figure 5–15. Set Current HardCopy II Companion Revision

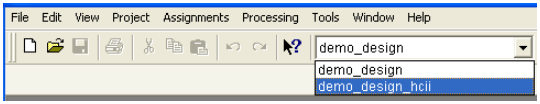


Compiling the HardCopy II Companion Revision

The Quartus II software enables you to compile your HardCopy II design with preliminary timing information. The timing constraints for the HardCopy II companion revision can be the same as the Stratix II design used to create the revision. The Quartus II software contains preliminary timing models for HardCopy II devices and you can gauge how much performance improvement you can achieve in the HardCopy II device compared to the Stratix II FPGA. Altera verifies that the HardCopy II Companion Device timing requirements are met in the HardCopy Design Center.

After you create your HardCopy II companion revision from your compiled Stratix II design, select the companion revision in the Quartus II software design revision drop-down box (Figure 5–16) or from the **Revisions** list. Compile the HardCopy II companion revision. After the Quartus II software compiles your design, you can perform a comparison check of the HardCopy II companion revision to the Stratix II prototype revision.

Figure 5–16. Changing Current Revision



Comparing HardCopy II & Stratix II Companion Revisions

Altera uses the companion revisions in a single Quartus II project to maintain the seamless migration of your design from a Stratix II FPGA to a HardCopy II structured ASIC. This methodology allows you to design with one set of Register Transfer Level (RTL) code to be used in both Stratix II FPGA and HardCopy II structured ASIC, guaranteeing functional equivalency.

When making changes to companion revisions, use the Compare HardCopy II Companion Revisions feature to ensure that your Stratix II design matches your HardCopy II design functionality and compilation settings. To compare companion revisions, on the Project menu, point to HardCopy II Utilities and click **Compare HardCopy II Companion Revisions**.

You must perform this comparison after both Stratix II and HardCopy II designs are compiled in order to hand off the design to Altera's HardCopy Design Center.

The Comparison Revision Summary is found in the Compilation Report and identifies where assignments were changed between revisions or if there is a change in the logic resource count due to different compilation settings.

Generate HardCopy II Handoff Report

In order to submit a design to the HardCopy Design Center, you must generate a HardCopy II Handoff Report providing important information about the design that you want the HardCopy Design Center to review. To generate the HardCopy II Handoff Report, you must:

- Successfully compile both Stratix II and HardCopy II revisions of your design
- Successfully run the Compare HardCopy II Companion Revisions utility

Once you generate the HardCopy II Handoff Report, you can archive the design using the Archive HardCopy II Handoff Files utility described in "Archive HardCopy II Handoff Files" on page 5–27.

Archive HardCopy II Handoff Files

The last step in the HardCopy II design methodology is to archive the HardCopy II project for submission to the HardCopy Design Center for back-end migration. The HardCopy II archive utility creates a different Quartus II Archive File than the standard Quartus II project archive utility generates. This archive contains only the necessary data from the Quartus II project needed to implement the design in the HardCopy Design Center.

In order to use the **Archive HardCopy II Handoff Files** utility, you must complete the following:

- Compile both the Stratix II and HardCopy II revisions of your design
- Run the Compare HardCopy II Revisions utility
- Generate the HardCopy II Handoff Report

To select this option, on the Project menu point to HardCopy II Utilities and click **Archive HardCopy II Handoff File** utility.

HardCopy II Advisor

The HardCopy II Advisor provides the list of tasks you should follow to develop your Stratix II prototype and your HardCopy II design. To run the HardCopy II Advisor, on the Project menu, point to HardCopy II Utilities and click **HardCopy II Advisor**. The following list highlights the

checkpoints that the HardCopy II Advisor reviews. This list includes the major check points in the design process; it does not show every step in the process for completing your Stratix II and HardCopy II designs:

1. Select a Stratix II device.
2. Select a HardCopy II device.
3. Turn on the **Design Assistant**.
4. Set up timing constraints.
5. Check for incompatible assignments.
6. Compile and check Stratix II design.
7. Create or overwrite companion revision.
8. Compile and check HardCopy II companion results.
9. Compare companion revisions.
10. Generate Handoff Report.
11. Archive Handoff Files and send to Altera.

The HardCopy II Advisor shows the necessary steps that pertain to your current selected device. The Advisor shows a slightly different view for a design with Stratix II selected as compared to a design with HardCopy II selected.

In the Quartus II software, you can start designing with the HardCopy II device selected first, and build a Stratix II companion revision second. When you use this approach, the HardCopy II Advisor task list adjusts automatically to guide you from HardCopy II development through Stratix II FPGA prototyping, it then completes the comparison archiving and handoff to Altera.

When your design uses the Stratix II FPGA as your starting point, Altera recommends following the Advisor guidelines for your Stratix II FPGA until you complete the prototype revision.

When the Stratix II FPGA design is complete, create and switch to your HardCopy II companion revision and follow the Advisor steps shown in that revision until you are finished with the HardCopy II revision and are ready to submit the design to Altera for back-end migration.

Each category in the HardCopy II Advisor list has an explanation of the recommended settings and constraints, as well as quick links to the features in the Quartus II software that are needed for each section. The HardCopy II Advisor displays:

- A green check box when you have successfully completed one of the steps
- A yellow caution sign for steps that must be completed before submitting your design to Altera for HardCopy development
- An information callout for items you must verify

Selecting an item within the HardCopy II flow menu provides a description of the task and recommended action. The view in the HardCopy II Advisor differs depending on the device you select.

Figure 5–17 shows the HardCopy II Advisor with the Stratix II device selected.

Figure 5–17. HardCopy II Advisor with Stratix II Selected

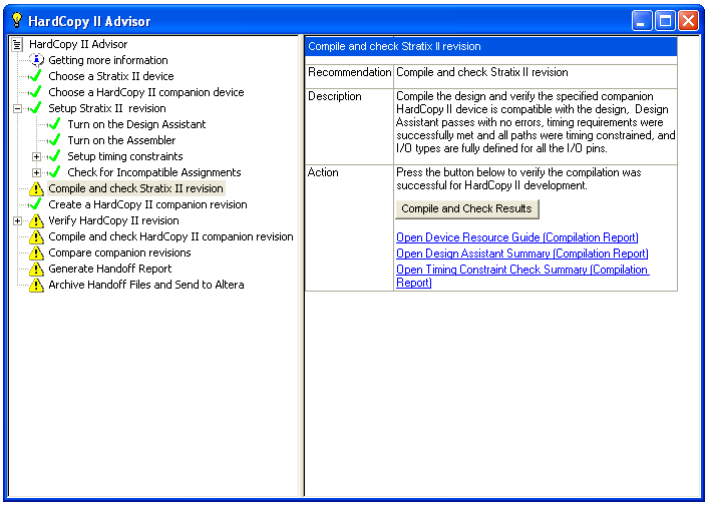
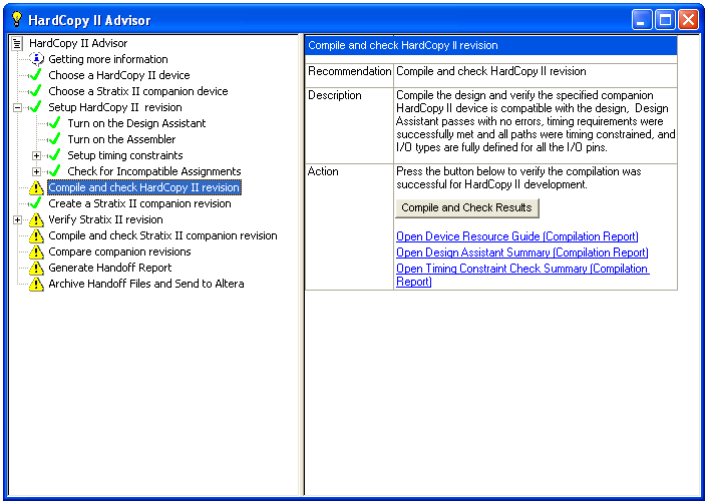


Figure 5–18 shows the HardCopy II Advisor with the HardCopy II device selected.

Figure 5–18. HardCopy II Advisor with HardCopy II Device Selected

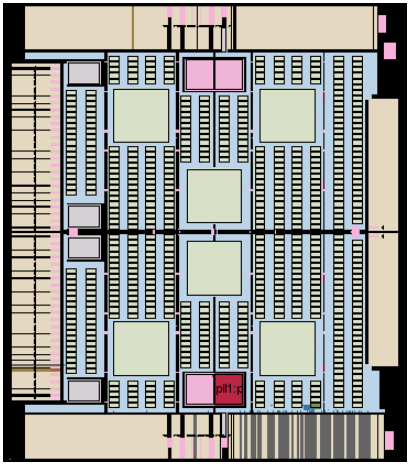


HardCopy II Floorplan View

The Quartus II software displays the preliminary timing closure floorplan and placement of your HardCopy II companion revision. This floorplan shows the preliminary placement and connectivity of all I/O pins, PLLs, memory blocks, HCell macros, and DSP HCell macros. Congestion mapping of routing connections can be viewed using the Bird's Eye viewer settings. This is useful in analyzing densely packed areas of your floorplan that could be reducing the peak performance of your design. The HardCopy Design Center verifies final HCell macro timing and placement to guarantee timing closure is achieved.

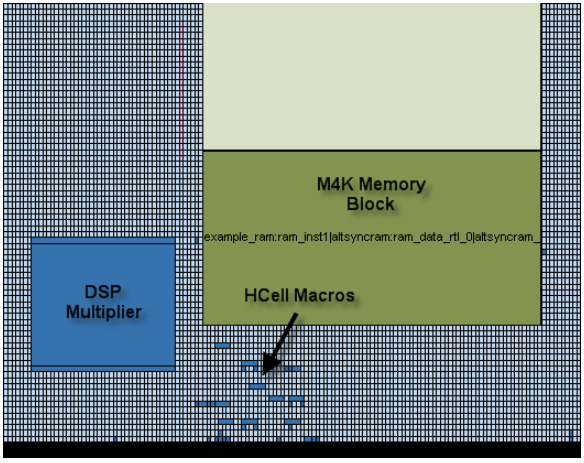
Figure 5–19 shows an example of the HC230F1020 device floorplan.

Figure 5–19. HC230F1020 Device Floorplan



In this small example design, the logic is placed near the bottom edge. You can see the placement of a DSP block constructed of HCell Macros, various logic HCell Macros, and an M4K memory block. A labeled close-up view of this region is shown in Figure 5–20.

Figure 5–20. Close-Up View of Floorplan



The HardCopy Design Center performs final placement and timing closure on your HardCopy II design based on the timing constraints provided in the Stratix II design.

For more information about the HardCopy Design Center's process, refer to the *Back-End Design Flow for HardCopy Series Devices* chapter in volume 1 of the *HardCopy Series Device Handbook*.

Conclusion

You can use the Quartus II software to design HardCopy II devices and to develop prototypes using Stratix II FPGAs. This is done using the standard FPGA development process with the addition of the HardCopy II Device Resource Guide, HardCopy II Companion Devices assignment HardCopy II Utilities, and the HardCopy II Advisor.

The addition of the HardCopy II Advisor to the Quartus II software provides an instrumental development guide for you to complete your HardCopy II and Stratix II device designs. The HardCopy II Utilities included in the Quartus II software provide you with the tools necessary to complete your Stratix II FPGA prototype and HardCopy II structured ASIC design. The addition of the HardCopy II companion revisions feature to the process allows for rapid development and verification that your HardCopy II design is functionally equivalent to your Stratix II FPGA prototype.

6. Script-Based Design for HardCopy II Devices

H51025-1.0

Introduction

The Quartus II software includes a set of command-line executables, many of which support an interactive Tcl shell. Using the Tcl shell, you can perform FPGA or HardCopy design operations without using the Quartus II window-based GUI.

This chapter provides an introduction to Tcl operations for script-based HardCopy II design using the interactive Tcl shell. Topics covered in this chapter include:

- Overview of Tcl scripting features in the Quartus II software
- HardCopy II design flow
- Applying location and timing constraints
- Synthesis, place and route for HardCopy II designs, and Stratix II prototypes
- Design verification and analysis

Tcl Support in the Quartus II Software

The Quartus II software provides different ways to execute Tcl commands and scripts including:

- A Tcl Console Window
- A Tcl Scripts Dialogue Box
- Command-Line Processing
- An Interactive Tcl Shell

The Tcl Console window and **Tcl Scripts** dialogue box both run within the Quartus II GUI and are not described here. Instead, this chapter focuses on the Interactive Tcl shell that you can use with the Quartus II command-line executables.

For more information about command-line processing and the use of Quartus II command-line executables in batchfiles, makefiles, and scripts, refer to the *Command-Line Scripting* chapter in volume 2 of the *Quartus II Handbook*.

For more information on the Quartus II Tcl implementation, refer to the *Tcl Reference Manual* and the *Tcl Scripting* chapter of the *Quartus II Handbook*.

Interactive Tcl Shell

A number of the Quartus II executables can be run with an interactive Tcl shell as the user interface. These executables are identified in Table 6–1. The interactive Tcl shell supports Tcl version 8.4.

Table 6–1. Quartus II Command-Line Executables with Interactive Tcl Support

Executable Name	Description
quartus_sh	A basic Tcl interpreter shell. Supports assignment specification, compile operations, and native operating system commands. For more information refer to quartus_sh in the *Command-Line Executables* section of the *Quartus II Scripting Reference Manual*.
quartus_tan	The Quartus II timing analyzer engine supports building the timing graph for the design and timing analysis Tcl commands. For more information refer to quartus_tan in the *Command-Line Executables* section of the *Quartus II Scripting Reference Manual*.
quartus_cdb	The Quartus II database interface executable. Supports operations related to the design database such as LogicLock, back-annotation and FPGA-HardCopy comparison for HardCopy II designs. For more information refer to quartus_cdb in the *Command-Line Executables* section of the *Quartus II Scripting Reference Manual*.
quartus_sim	The Quartus II Simulator. For more information refer to quartus_sim in the *Command-Line Executables* section of the *Quartus II Scripting Reference Manual*.

The interactive Tcl shell for command-line executables is invoked using the -s command-line switch. For example, to run the basic Quartus shell, type quartus_sh -s at the command prompt:

```
% quartus_sh -s
Info:
*******************************************************************
Info: Running Quartus II Shell
Info:
*******************************************************************
Info: The Quartus II Shell supports all TCL commands in addition
Info: to Quartus II Tcl commands. All unrecognized commands are
Info: assumed to be external and are run using Tcl's "exec"
Info: command.
Info: - Type "exit" to exit.
Info: - Type "help" to view a list of Quartus II Tcl packages.
Info: - Type "help -pkg <package name>" to view a list of Tcl commands
Info:   available for the specified Quartus II Tcl package.
Info: - Type "help -tcl" to get an overview on Quartus II Tcl usages.
Info:
*******************************************************************
tcl>
```

The Quartus II Tcl implementation provides custom Tcl procedures to perform Quartus II operations. These procedures are organized into Tcl packages based on their functionality. Table 6–2 lists these Tcl packages and their availability. Some packages are loaded by default when the executable is invoked. Others must be explicitly loaded before their Tcl procedures are used. To load a particular package, use the load_package Tcl procedure. For example, to load the flow package in the quartus_sh shell, the following Tcl statement is executed:

```
tcl> load_package flow
```

It is important to note that not all executables support all Tcl packages.

Table 6–2. Tcl Package Support in Quartus II Executables (Part 1 of 2)

Executable Name	Supported Tcl Package	Loaded by Default?
quartus_sh	device	Loaded
	flow	Not Loaded
	misc	Loaded
	project	Loaded
	report	Not Loaded

Table 6–2. Tcl Package Support in Quartus II Executables (Part 2 of 2)

Executable Name	Supported Tcl Package	Loaded by Default?
quartus_tan	advanced_timing	Not Loaded
	device	Not Loaded
	flow	Not Loaded
	logiclock	Not Loaded
	Misc	Loaded
	project	Loaded
	report	Not Loaded
	timing	Loaded
	timing_report	Not Loaded
quartus_cdb	backannotate	Not Loaded
	chip_editor	Not Loaded
	device	Loaded
	flow	Not Loaded
	logiclocs	Not Loaded
	misc	Loaded
	project	Loaded
	report	Not Loaded
quartus_sim	device	Loaded
	flow	Not Loaded
	misc	Loaded
	project	Loaded
	report	Loaded
	simulator	Loaded

A brief description of each of the Tcl packages referenced in Table 6–2 is given in Table 6–3.

The Quartus II command-line executables and Tcl shells are supported on all Quartus II operating systems including Microsoft Windows, Linux, and Unix platforms.

For more information on Quartus II Tcl packages and their available Tcl procedures, refer to the *Tcl Packages and Commands* chapter in the *Quartus II Scripting Reference Manual*.

Table 6–3. Quartus II Tcl Package Descriptions

Tcl Package	Description
advanced_timing	Traverse the timing netlist and get information about timing modes
backannotate	Back annotate assignments
chip_editor	Identify and modify resource usage and routing with the Chip Editor
database_manager	Manage version-comparable database files
device	Get device and family information from the device database
flow	Compile a project, run command-line executables and other common flows
logiclock	Create and manage LogicLock regions
misc	Perform miscellaneous tasks
project	Create and manage projects and revisions, make any project assignments including timing assignments
report	Get information from report tables, create custom reports
simulator	Configure and perform simulations
stp	Operate the SignalTap II Analyzer
timing	Annotate timing netlist with delay information, compute and report timing paths
timing_report	List timing paths

Command-Line Processing

In addition to the interactive Tcl shell, the Quartus II command-line executables support command-line switches for executing Tcl scripts and commands. When used with these switches, a command-line executable quits when complete. The command-line executables also provide switches for performing specific Quartus II operations. For example, the following c-shell script takes as its argument the top-level design file and entity name and runs it through the entire HardCopy II design flow.

```
!#/bin/csh
quartus_sh --flow compile %1
quartus_cdb %1 --create_companion=%1_hcii
quartus_sh --flow compile %1 -c %1_hcii
quartus_cdb --compare=%1_hcii %1 -c %1
```

This example shows what is, perhaps, the simplest way to execute the HardCopy II design flow. If you have developed and applied the design I/O, location and timing constraints for the project, these constraints are included during script execution.

For more information on the Quartus II executables and command-line options, refer to the *Command-Line Executables* chapter in the *Quartus II Scripting Reference Manual* and the *Command-Line Scripting* section in volume 2 of the *Quartus II Handbook*.

The HardCopy II Design Flow

The Quartus II software supports both HardCopy II first and Stratix II first design flows. The Stratix II first flow involves the following:

- Compiling for the Stratix II FPGA prototype
- Verifying the Stratix II FPGA prototype
- Migrating the prototype design to a HardCopy II design
- Compiling the HardCopy II design
- Transferring your HardCopy II files to the Altera Design Center

The Hardcopy II first flow is similar, but starts with compiling the HardCopy II target device. Once the HardCopy II compile completes successfully, the design is migrated to the Stratix II target.

The HardCopy II design flow in the Quartus II software is shown in Figure 6–1. To begin a design, create a new project and revision for the Stratix II FPGA prototype. Apply Quartus II settings together with I/O assignments and timing constraints. Compile the Stratix II prototype revision (synthesis, place and route, and assembly) to produce a complete layout, with timing closure and free from errors. You can now perform any additional functional and timing verification necessary and then implement and verify the prototype in hardware.

Once the FPGA prototype is verified, you can compile the HardCopy II design. Begin by creating a HardCopy II companion revision for the FPGA prototype:

1. Create a HardCopy II companion revision for the FPGA prototype. All design settings and constraints are automatically migrated to the new companion revision.

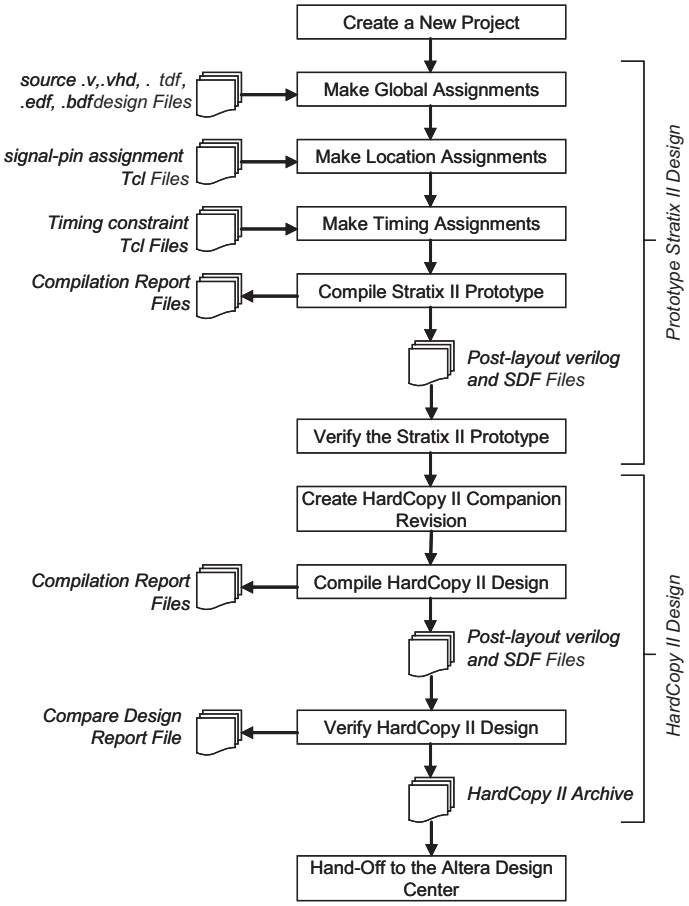
2. Compile the HardCopy II revision. As the compile runs, the Design Assistant checks for errors. When the compile completes, you should correct errors and resolve failures that appear in the Quartus II reports.

3. Run the HardCopy II Companion Revision Comparison tool to compare the HardCopy II design against the FPGA prototype. The comparison tool checks for structural equivalency and consistency between the two revisions.

4. If there are no mismatches, you can prepare the HardCopy II design files for transfer to the Altera Design Center.

In addition to design verification in the Quartus II software, the flow can generate files required to perform Static Timing Analysis (STA) in Synopsys' Primetime.

Figure 6–1. The HardCopy II Design Flow



The design flow of Figure 6–1 begins with a Stratix II FPGA prototype design and migrates this design to a HardCopy II device target. Beginning in the Quartus II version 5.1 software, you have the option of starting with a HardCopy II target and migrating this design to a Stratix II target for FPGA prototyping. The design flow in this case is the same as the one shown in Figure 6–1 with the Stratix II and HardCopy II design steps switched.

For more information on the HardCopy II design flow and alternative methods to complete HardCopy II designs using the Quartus II GUI refer to *Quartus II Support for HardCopy II Devices* chapter in the *Quartus II Handbook* or the *HardCopy II Design Considerations* chapter in volume 1 of the *HardCopy Series Handbook*.

The following sections describe each step of the flow shown in Figure 6–1 and explains how each step is completed using the interactive Tcl shell.

Creating a New Project

Both FPGA and HardCopy design in Quartus II revolve around the use of projects. You must create a project before you begin working with a new design. A project includes source design files (RTL and schematics), Quartus II tool settings, and a set of pin locations and timing constraints. Although a project can contain many different revisions for a design, each revision can have a unique set of design constraints, target device settings, and Quartus II settings. You must explicitly open a project before you can perform other operations on the project. You must close the current project to switch to a different project or revision.

This section details the different operations relating to project management using Tcl commands.

Creating a Stratix II Prototype Project

To create a new Stratix II prototype project, use the **project_new** Tcl command. The syntax for this command is:

`tcl> project_new`[`-family` *<family>*][`-overwrite`][`-part` *<part>*] [`-revision` *<revision_name>*] *<project_name>*

The only required argument for this command is the project name, *<project name>*, although the target device family, part code, and revision name can be specified at this time also. By default, the revision name is the same as the project name. The device family and part code can be set later using the **set_global_assignment** command. For example, to open a project called `demo_design` with the default revision name of `demo_design` and an unspecified target device family or part, the following Tcl command is executed:

```
tcl> project_new demo_design
```

Creating a new project creates a quartus settings file (QSF) and a Quartus II Project file (QPF) in the current directory. In addition, a db subdirectory is created that is used to store Quartus II database files. In the case of the `demo_design` project example, the following files are created in the project directory:

```
demo_design.qpf
demo_design.qsf
db/
  demo_design.db_info
```

Opening a Project

The project created automatically opens when you use the **project_ne**w command. In future Quartus II sessions, or if you close the project, you must open the project with the Tcl command: **project_open**. The syntax for the **project_open** command is:

```
tcl> project_open [-current_revision] [-revision <revision_name>] <project_name>
```

For example, to open the default revision of project `demo_design`, execute the following Tcl command:

```
tcl> project_open demo_design -revision demo_design_fpga
```

It is a good practice to have consistent names for the Stratix II and HardCopy II revisions of your project. This makes it easy to identify which revision is which. For example, naming your revisions *projectname_fpga* and *projectname_hcii* would help you easily identify which revision is the Stratix II revision, and which is the HardCopy II revision.

Closing a Project

Before ending a Quartus II project session, it is good practice to close the Quartus II project using the **project_close** command. This ensures that any changes you have made to your project are written to the Quartus II QSF file. The syntax for the **project_close** command is:

```
tcl> project_close [-dont_export_assignments]
```

New Project Example Script

The following script shows the use of Tcl commands for opening and closing a project called `demo_design` with the revision name, `demo_design_fpga`. If the project does not already exist, it is created. This script makes use of the **project_exists** and **project_open** Tcl commands.

```
## Example Tcl Script for opening and closing a project

## Open Project demo_design.  If the Project does not Already
## Exist, Create it
if [is_project_open] project_close
if [project_exists demo_design] {
```

```
  project_open demo_design -revision demo_design_fpga
} else {
  project_new demo_design -revision demo_design_fpga
}

## Include Other Tcl Commands Here …

## Close project demo_design and write any changes to settings to
## demo_design.qsf
project_close

## End of script
```

For more information on these and other useful project-related commands, refer to the *Project* section in the *Tcl Packages and Commands* chapter in the *Quartus II Scripting Reference Manual*.

Making Global Assignments

Initializing a HardCopy II Design

For a HardCopy II design, the following key operations are required after a Quartus II project is created:

- Specify design source files (Verilog, VHDL, AHDL, EDIF, and BDF files)
- Specify the Stratix II prototype target family and device name
- Specify the HardCopy II companion revision and migration device
- Enable the Design Assistant
- Make recommended HardCopy II specific Quartus II tool settings

In addition to these, other project settings affecting downstream tools, such as synthesis and place-and-route, can be made at this time.

The operations listed above are performed using the **set_global_assignment** command. The syntax for this command is:

```
tcl> set_global_assignment [-comment<comment>] [-disable]
[-entity <entity_name>] -name <name> [-remove] [-section_id
<section_id>] <value>
```

The most important parameters for the **set_global_assignment** command are <*name*> and <*value*>. The <*name*> argument specifies the Quartus II global variable to be set and <*value*> is the new value assigned to that variable.

One of the steps in initializing a HardCopy II design is to turn on the Design Assistant. When run in the GUI, the Design Assistant provides a visual checklist for running both the Stratix II and HardCopy II phases of the design. For first-time users, this can provide a powerful guide for successfully completing your HardCopy II project.

The key global variables for a HardCopy II project are listed in Table 6–4.

Table 6–4. Key HardCopy II Design Settings

Global Variable Name <*name*>	Value Description <*value*>
VERILOG_FILE	Verilog file name
VHDL_FILE	VHDL file name
AHDL_FILE	Altera HDL file name
EDIF_FILE	EDIF file name
BDF_FILE	Altera schematic file name
FAMILY	Device Family Name, for example, Stratix II
DEVICE	Prototype FPGA target device name
TOP_LEVEL_ENTITY	Top-level design entity or module name
DEVICE_TECHNOLOGY_MIGRATION_LIST	HardCopy II target device name
COMPANION_REVISION	HardCopy II design revision name
ENABLE_DRC_SETTINGS	Turn on the Design Assistant
REPORT_IO_PATHS_SEPARATELY	Creates a separate report panel for input and output min and max timing results
FLOW_ENABLE_TIMING_CONSTRAINT_CHECK	Timing constraints are checked for completeness (all clock domains constraints and minimum and maximum constraints are set for all I/O paths)
DO_COMBINED_ANALYSIS	Timing analysis are run for fast and slow operating conditions and for best and worst-case timing analysis, respectively
IGNORE_CLOCK_SETTINGS	This must be turned off
ENABLE_RECOVERY_REMOVAL_ANALYSIS	Verify recovery and removal times on asynchronous control and reset signals
ENABLE_CLOCK_LATENCY	Clock latency is included in timing analysis to asses clock-insertion timing and clock skew

The DEVICE and DEVICE_TECHNOLOGY_MIGRATION_LIST variables are the parts used for the Stratix II prototype design and the HardCopy II design. The selected Stratix II prototype device must be compatible with the selected HardCopy II device to make migration possible. Valid pairings for these devices are listed in Table 6–5.

For the DEVICE_TECHNOLOGY_MIGRATION_LIST variable, the HardCopy II part names listed in Table 6–5 are used. For the DEVICE variables, the Stratix II part names include the speed grade for the part. The speed grade is a two character code indicating industrial (I) or commercial (C) and the speed indicator (number 3, 4, or 5). For example,

a -4 commercial part is denoted using the two character speed grade C4. The two-character speed grade is appended to the Stratix II part name to form the value string for the DEVICE variable.

Table 6–5. Stratix II Prototype Options for HardCopy II

HardCopy II Part	Stratix II Prototype Part
HC210F484C	EP2S30F484C3 EP2S30F484C4 EP2S30F484C5 EP2S30F484I4
	EP2S60F484C3 EP2S60F484C4 EP2S60F484C5 EP2S60F484I4
	EP2S90H484C4 EP2S90H484C5
HC220F672C	EP2S60F672C4 EP2S60F672C4 EP2S60F672C5 EP2S60F672I4
HC220F780C	EP2S90F780C4 EP2S90F780C5
	EP2S130F780C4 EP2S130F780C5
HC230F1020C	EP2S90F1020C3 EP2S90F1020C4 EP2S90F1020C5 EP2S90F1020I4
	EP2S130F1020C3 EP2S130F1020C4 EP2S130F1020C5 EP2S130F1020I4
	EP2S180F1020C3 EP2S180F1020C4 EP2S180F1020C5 EP2S180F1020I4
HC2401020C	EP2S180F1020C3 EP2S180F1020C4 EP2S180F1020C5 EP2S180F1020I4
HC240F1508C	EP2S180F1508C3 EP2S180F1508C4 EP2S180F1508C5 EP2S180F1508I4

The following two Tcl commands demonstrate setting the DEVICE and DEVICE_TECHNOLOGY_MIGRATION_LIST variables.

```
tcl> set_global_assignment -name DEVICE EP2S90F1020C4
tcl> set_global_assignment -name \
DEVICE_TECHNOLOGY_MIGRATION_LIST HC230F1020
```

The Design Assistant

You should turn on the Design Assistant at the beginning of the design process by turning on the ENABLE_DRC_SETTINGS global variable.

```
tcl> set_global_assignment -name ENABLE_DRC_SETTINGS ON
```

The Design Assistant runs concurrently with every step of both the prototype Stratix II and HardCopy II design flows. When the Design Assistant is turned on, the Quartus II software checks to ensure that the project fully complies with all HardCopy II design rules and requirements.

For more information on the Design Assistant, refer to the *Design Guidelines for HardCopy II Devices* chapter in volume 1 of the *HardCopy Series Handbook*. and the *Quartus Support for HardCopy II Devices* chapter in the *Quartus II Handbook*.

Example Tcl Script for Making Global Assignments

The example Tcl script below illustrates the application of global constraints for a HardCopy II project.

```
## Example Global Assignments Script for a HardCopy II Design
## This Script Applies Settings for a EP2S90 Stratix II
## prototype FPGA target and a HC230 HardCopy II target

## Source Design File Settings
## ==========================
set_global_assignment -name VERILOG_FILE demo_design.v
set_global_assignment -name VERILOG_FILE example_ram.v

## Stratix II Prototype FPGA Target Settings
## =========================================
set_global_assignment -name FAMILY "Stratix II"
set_global_assignment -name DEVICE EP2S90F1020C4
set_global_assignment -name TOP_LEVEL_ENTITY demo_design

## HardCopy II Companion Revision and Target Settings
## ================================================
set_global_assignment -name COMPANION_REVISION_NAME \
                                      demo_design_hardcopyii
set_global_assignment -name DEVICE_TECHNOLOGY_MIGRATION_LIST HC230F1020
## Design Assistant Assignments and Settings Required for HardCopy II

##============================================================
set_global_assignment -name ENABLE_DRC_SETTINGS ON

set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 1

set_global_assignment -name REPORT_IO_PATHS_SEPARATELY ON
set_global_assignment -name FLOW_ENABLE_TIMING_CONSTRAINT_CHECK ON
set_global_assignment -name DO_COMBINED_ANALYSIS ON
set_global_assignment -name IGNORE_CLOCK_SETTINGS OFF

set_global_assignment -name ENABLE_RECOVERY_REMOVAL_ANALYSIS ON
set_global_assignment -name ENABLE_CLOCK_LATENCY ON

## End of Script
```

Making I/O Assignments

Because of the complex rules governing the use of programmable I/O cells and their availability for specific pins and packages, Altera highly recommends that I/O assignments are completed using the Pin Planning tool and the Assignment Editor in the Quartus II GUI. These tools ensure that all of the rules regarding each pin and I/O cell are applied correctly. The Quartus II GUI can export a Tcl script containing all I/O assignments and specifications. I/O assignments are described here for information only.

For more information on I/O location and type assignments using the Quartus II Assignment Editor and Pin Planner tools, refer to the *Assignment Editor* chapter in volume 2 of the *Quartus II Handbook*.

In this section, I/O specification is considered in two parts:

- Pin assignments
- I/O Type assignments

Pin Assignments

Design I/O signals are assigned to package balls using the s**et_location_assignment** command. The syntax for this command is given below:

`tcl> set_location_assignment` [`-comment` <*comment*>] [`-disable`] [`-remove`] `-to` <*destination*> <*value*>

Here, <*destination*> is the package ball name and <*value*> is the design I/O signal name. For BGA and FBGA packages, the ball name follows the form `PIN_`<*coordinate*>. For example, to assign design I/O signal `data_out[15]` to package ball AL17:

```
tcl> set_location_assignment -to PIN_AL17 data_out[15]
```

Setting I/O Type & Parameters

For I/O type and parameter specification the **set_instance_assignment** command is used. The syntax for this command is:

`tcl> set_instance_assignment` [`-comment <comment>`] [`-disable`] [`-entity` <*entity_name*>] [`-from` <*source*>] `-name` <*name*> [`-remove`] [`-section_id` <*section_id*>] [`-to` <*destination*>] <*value*>

The assignment name, <*name*>, should be set to `IO_STANDARD` to indicate that an I/O specification is being applied. The related I/O signal is specified as `-to` <*destination*>. The destination argument is a string

providing details on the I/O type, such as levels and standards. Table 6–6 lists the strings corresponding to the I/O standards supported in HardCopy II devices.

Table 6–6. Tcl I/O Standard Strings

I/O Type or <*name*>	Description
LVTTL	LVTTL I/O
LVCMOS	LVCMOS I/O
“3.3-V PCI”	3.3 V PCI I/O
“3.3-V PCI-X”	3.3 V PCI X I/O
“1.5 V“	1.5 V I/O
“1.8 V”	1.8 V I/O
“2.5 V”	2.5 V I/O
“1.5-V HSTL CLASS I”	QDRII SRAM 1.5 V I/O
“1.5-V HSTL CLASS II”	QDRII SRAM 1.5 V I/O
“1.8-V HSTL CLASS I”	QDRII SRAM/RLDRAM II 1.8 V I/O
“1.8-V HSTL CLASS II”	QDRII SRAM/RLDRAM II 1.8 V I/O
“DIFFERENTIAL 1.5-V HSTL CLASS I”	Memory clock interface
“DIFFERENTIAL 1.5-V HSTL CLASS II”	Memory clock interface
“DIFFERENTIAL 1.8-V HSTL CLASS I”	Memory clock interface
“DIFFERENTIAL 1.8-V HSTL CLASS II”	Memory clock interface
“DIFFERENTIAL 1.8-V SSTL CLASS I”	DDR2 SDRAM
“DIFFERENTIAL 1.8-V SSTL CLASS II”	DDR2 SDRAM
“DIFFERENTIAL SSTL-2”	DDR SDRAM
“DIFFERENTIAL 2.5-V SSTL CLASS II”	DDR SDRAM
“SSTL-18 CLASS I”	DDR2 SDRAM
“SSTL-18 CLASS II”	DDR2 SDRAM
“SSTL-2 CLASS I”	DDR SDRAM
“SSTL-2 CLASS II”	DDR SDRAM
LVDS	2.5 V differential signaling
HYPERTRANSPORT	2.5 V differential signaling
LVPCL	Differential

You can specify a number of other I/O parameters by using the **set_instance_assignment** command. Some of the more common parameters are listed in Table 6–7.

Table 6–7. Tcl Common I/O Parameter Settings

<*name*> setting	<*value*> setting	Description
weak_pull_up_resistor	on	Implement a weak pull-up resistor on the pin
output_pin_load	integer	Capacitive load for an output or bidirectional pin. Units of pF.
fast_output_register	on	Implements a fast output register in the I/O cell or adjacent LAB
fast_output_enable_register	on	Implement a fast output enable register in the I/O cell or/and adjacent LAB
fast_input_register	on	Implements a fast input register in the I/O cell or adjacent LAB
current_strength_new	2 mA 4 mA 8 mA 10 mA 12 mA 16 mA 18 mA 20 mA 24 mA minimum_current or maximum_current	Drive strength for an output or bidi pin
stratixii_termination	differential “series 25 ohms with calibration” “series 25 ohms without calibration” “series 50 ohms with calibration” “series 50 ohms without calibration”	On-chip termination (or impedance matching) for an I/O pin

For more information on I/O availability in HardCopy II devices, refer to the *I/O Structures & Features* section in volume 1 of the *HardCopy Series Handbook*.

I/O Assignment Example Script

The following Tcl script example specifies several different I/O constraints.

```
## Signal-Ball Assignments
set_location_assignment PIN_AH5 -to addr_out[0]
set_location_assignment PIN_AH6 -to addr_out[1]
set_location_assignment PIN_AJ5 -to data_in[0]
set_location_assignment PIN_AJ6 -to data_in[1]
set_location_assignment PIN_AJ32 -to resetn
set_location_assignment PIN_AM17 -to ref_clk

# I/O Type and Parameter Assignments
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to addr_out[0]
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to addr_out[1]
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to data_in[0]
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to data_in[1]
set_instance_assignment -name IO_STANDARD LVDS -to resetn
set_instance_assignment -name IO_STANDARD LVCMOS -to ref_clk

set_instance_assignment -name fast_input_register on -to data_in[0]
set_instance_assignment -name fast_input_register on -to data_in[1]
set_instance_assignment -name fast_output_register on -to addr_out[0]
set_instance_assignment -name fast_output_register on -to addr_out[1]

set_instance_assignment -name output_pin_load 10 -to addr_out[0]
set_instance_assignment -name output_pin_load 10 -to addr_out[1]
```

Assigning Timing Constraints

Planning Design Timing Constraints

Timing constraints ensure that a design compiled in the Quartus II software meets specific timing requirements. When you target an FPGA, you may decide not to apply a complete set of timing constraints, choosing instead to fix any timing problems in your prototype system if and when they arise. HardCopy devices, however, cannot be modified using reconfiguration to fix timing problems, so it is critically important that a design is fully constrained. Designs not fully constrained would result in significantly different timing characteristics between the prototype Stratix II FPGA and the HardCopy II device. By fully constraining a design, Altera can guarantee that both the Stratix II FPGA and the HardCopy II device fully complies with your timing specifications.

The minimum set of timing constraints for a HardCopy II design are:

- Clock settings (FMax) for each and every clock domain
- Minimum and maximum delays for all I/O paths, including asynchronous reset and control I/O signals

In addition, it is good design practice to develop timing constraints to cover:

- Specific cross-clock domain timing requirements
- False paths
- Multicycle paths

In the Quartus II software, timing constraints are applied using dedicated Tcl commands and by assigning timing-specific attributes using the **set_instance_assignment** command.

This section provides an overview of timing constraint development using Tcl commands.

For more information on timing constraints, refer to the *Timing Analysis* section in volume 3 of the *Quartus II Handbook*.

Specifying System Clocks

The most basic constraints that should be applied describe the clock for each clock domain. Parameters usually specified for each clock are:

- Maximum frequency (FMAX_REQUIREMENT global assignment)
- Latency (LATE_CLOCK_LATENCY/EARLY_CLOCK_LATENCY assignments)
- Uncertainty (**set_clock_uncertainty** command)

Clock uncertainty specified with the **set_clock_uncertainty** command models any uncertainty in the clock period, including jitter, and is often used to introduce some margin into the target clock frequency. The following example script illustrates clock definition for a design with two clock domains, clk_a and clk_b. In this case, both clocks run at 100 MHz, but with different clock latency and skew.

```
## Example Tcl Script Defining Clocks clk_a and clk_b
create_base_clock -fmax 10.0ns -target clk_a clk_a
set_instance_assignment -name LATE_CLOCK_LATENCY 3ns -to clk_a
set_instance_assignment -name EARLY_CLOCK_LATENCY 2ns -to clk_a
set_clock_uncertainty 0.25ns -to clk_a

create_base_clock -fmax 10.0ns -target clk_b clk_b
set_instance_assignment -name LATE_CLOCK_LATENCY 4ns -to clk_b
set_instance_assignment -name EARLY_CLOCK_LATENCY 3ns -to clk_b
set_clock_uncertainty 0.25ns -to clk_b
```

Input/Output Timing

System clock parameters define the setup and hold timing for register to register paths within each clock domain. I/O timing parameters are used to describe I/O to register, and register to I/O timing.

The **set_input_delay** command can be used to specify the delay from a source external to the chip to an input pin, relative to a defined clock. The syntax for this command is given below.

```
tcl> set_input_delay [-h | -help] [-long_help] [-clk_ref <clock>] -to <input_pin> [-min]
[-max] [-clock_fall] [-remove] [-disable] [-comment <comment>] [<value>]
```

The <*clock*> argument specifies the reference clock for the delay. The <*input_pin*> argument is the top-level input signal for the design, and <*value*> is the external delay. The external delay is measured from the positive (rising) edge of <*clock*> unless the `-clock_fall` argument is specified. The `-min` and `-max` arguments are used to specify whether <`value`> is the minimum or maximum external delay, respectively.

The **set_output_delay** command is similar to the **set_input_delay** command except that is specifies the delay from an output pin to its external destination relative to a clock.

```
tcl> set_output_delay [-h | -help] [-long_help] [-clk_ref <clock>] -to <output_pin> [-min]
[-max] [-clock_fall] [-remove] [-disable] [-comment <comment>] [<value>]
```

As an example, the following Tcl script specifies input and output min and max delays for two I/O signals. Input `data_in[0]` has minimum and maximum external delays of 3 ns and 7 ns respectively. Output `data_out[0]` has minimum and maximum external delays of 4 ns and 8 ns respectively. The external input delays for `data_in[0]` are relative to the positive edge of clock `ref_clk` and the external output delays for `data_out[0]` are relative to the negative edge of clock `ref_clk`.

```
# Tcl Script Setting I/O Timing Using set_input_delay and set_output_delay
set_input_delay -clk_ref ref_clk -max -to data_in[0] 7ns
set_input_delay -clk_ref ref_clk -min -to data_in[0] 3ns
set_output_delay -clk_ref ref_clk -max -clock_fall -to data_out[0] 8ns
set_output_delay -clk_ref ref_clk -min -clock_fall -to data_out[0] 4ns
```

Creating Timing Exceptions

Timing exceptions are used to correct timing constraints not covered by clock settings and I/O timing settings. The most common of these are multicycle paths and false paths.

Multicycle paths are described using the **set_multicycle_assignment** command. The syntax for this command is:

```
tcl> set_multicycle_assignment [-comment <comment>] [-disable]
[-end] [-from <from_list>] [-hold] [-remove] [-setup] [-start]
[-to <to_list>] <path_multiplier>
```

Multicycle assignments are made with the `-setup` argument, to specify the maximum number of cycles, or with the `-hold` argument to specify the minimum number of cycles for a path.

False paths describe paths that should not be included in timing optimization or analysis operations. In the Quartus II software, there are a number of ways to describe false paths. By default, feedback from the output to input side of bidirectional I/O, read-while-write paths through memories, and cross-clock domain paths are not timed during optimization or timing analysis. To change these default settings, refer to the *Timing Settings* section in the *Quartus II Support of HardCopy Series Devices* chapter in volume 1 of the *Quartus II Handbook*.

The most common command for controlling false paths is the **set_timing_cut_assignment** command. The syntax for this command is:

```
tcl> set_timing_cut_assignment [-comment <comment>] [-disable]
[-from <from_pin_list>] [-remove] [-to <to_pin_list>]
```

All paths between nodes in the <*from_pin_list*> to nodes in the <*to_pin_list*> are excluded from timing optimization and analysis operations.

Example Tcl Script for Timing Constraints

```
# Timing Assignments
# ==================
create_base_clock –fmax 10.0ns –target ref_clk ref_clk

set_instance_assignment -name LATE_CLOCK_LATENCY 3ns -to ref_clk
set_instance_assignment -name EARLY_CLOCK_LATENCY 2ns -to ref_clk
set_clock_uncertainty –hold –to ref_clk 0.250ns
set_clock_uncertainty –setup –to ref_clk 0.250ns

# Input delay of 6ns (max) & 2ns (min) for bus data_in[1:0]
set_input_delay –clk_ref ref_clk –max –to data_in 6.0ns
set_input_delay –clk_ref ref_clk –min –to data_in 2.0ns
# Output delay of 6ns (max) & 2ns (min) for bus data_out[1:0]
set_output_delay –clk_ref ref_clk –max –to data_out 6.0ns
set_output_delay –clk_ref ref_clk –min –to data_out 2.0ns

# Don’t care about timing on the resetn net. Set as false path
set_timing_cut_assignment -from resetn
```

This section has provided an overview of Tcl commands for applying timing constraints.

For more information on the application of timing constraints using Tcl commands, refer to the *Tcl Packages and Commands* chapter in the *Quartus II Scripting Reference Manual*.

Compiling the Stratix II Prototype Design

Once all global assignments, resource assignments, and timing assignments have been specified, the next step in the design process is to compile the Stratix II FPGA prototype design. The `execute_flow` command is provided for this purpose and supports various arguments affecting the compilation process. The syntax for this command is:

```
tcl> execute_flow [-analysis_and_elaboration]
[-attempt_similar_placement] [-check_ios] [-check_netlist]
[-compile] [-compile_and_simulate] [-early_timing_estimate]
[-eco] [-export_database] [-fast_model]
[-generate_functional_sim_netlist]
[-import_database]
[-signalprobe]
```

The switches relevant to prototype Stratix II and HardCopy II design are listed in Table 6–8.

Table 6–8. `execute_flow` *Tcl Command Switches*

Switch	Description
`analysis_and_elaboration`	Perform synthesis and mapping to the target Altera technology
`attempt_similar_placement`	Runs Attempt Similar Placement
`check_ios`	Verify I/O assignments
`check_netlist`	Perform syntax checks on the netlist
`compile`	Execute the Quartus II compilation flow
`compile_and_simulate`	As for compile, but also run simulation
`early_timing_estimate`	Runs the early timing estimator
`eco`	Executes a Fitter ECO compilation
`export_database`	Exports a Version-Compatible Database
`fast_model`	Runs Timing Analysis (fast mode analysis
`generate_functional_sim_netlist`	Generate a Simulation Netlist
`import_database`	Imports a Version-Compatible Database

It is important to note that the HardCopy switches for the `execute_flow` command are for HardCopy Stratix designs, not HardCopy II designs.

The simplest way to run the `execute_flow` command is to use the `-compile` switch.

```
tcl> execute_flow -compile
```

Running the **execute_flow** command in this way executes the five stages of the Quartus II compilation flow with default settings for each stage:

- Analysis and Synthesis
- Fitter
- Timing Analysis
- Assembler

You should check I/O assignments to avoid problems in downstream compile operations. To do this, the `execute_flow` compilation is broken into three steps:

1. `tcl> execute_flow -analysis_and_elaboration`

2. `tcl> execute_flow -check_ios`

3. `tcl> execute_flow -compile`

It should be noted that, in the interests of clarity and brevity, the Tcl fragments given here do not incorporate any error checking. However, it is good practice to include code in your Tcl scripts that checks for success as your design proceeds. In the case of the `execute_flow` procedure, the return value can be used with the Tcl catch command to handle success or failure. The example below shows one option for doing this.

```
# Determine if compilation was successful and print out a
personalized message.
if {[catch {execute_flow -compile} result]} {
   puts "\nResult: $result\n"
   puts "ERROR: Compilation failed. See report files.\n"
} else {
 puts "\nINFO: Compilation was successful.\n"
}
```

For more information on the **`execute_flow`** command, refer to the command description in the *Tcl Packages and Commands* chapter in the *Quartus II Scripting Reference Manual*.

Compiling the HardCopy II Design

Once the Stratix II FPGA prototype design is compiled and verified, you can compile the HardCopy II revision of the design. This is a two-step process:

1. Create the HardCopy II companion revision.

2. Compile the HardCopy II companion revision.

To create the HardCopy II version of the design, run the `execute_hardcopyii` Tcl command with the `-create_companion` option:

```
tcl> execute_hardcopyii -create_companion demo_design_hcii
```

This command initializes the database for the HardCopy II revision and creates a new QSF file (in this example, **demo_design_hcii.qsf**), ensuring that all constraints for the Stratix II FPGA revision are ported over.

Next, the current working revision for the Quartus II project is changed to the HardCopy II revision and the design is compiled for the HardCopy II device target:

```
tcl> set_current_revision demo_design_hcii
tcl> execute_flow -compile
```

As with the prototype Stratix II revision, report files are generated in the project directory for each of the tools that are executed.

Understanding Report Files

The **execute_flow** command generates a number of report files in the project directory. These files summarize messages displayed on the console during compilation and provide additional information about the design. The name of each report file follows the format *<revision><tool short name>***.summary** and *<revision><tool short name>***.rpt**, where *<revision>* is the revision name of the current design. The **.summary** file contains a brief summary of messages and results from the tool while the **.rpt** file contains more detailed messages and information. For a HardCopy II project, two sets of report files are generated: one for the Stratix II prototype FPGA revision and one for the HardCopy II revision. Table 6–9 describes the different report files.

The Tcl report package provides a powerful collection of procedures for customizing and managing report files related to the Quartus II fitter and timing analysis engines.

For more information on customizing and managing report files, refer to the T*cl Packages & Command*s report section of the *Quartus II Tcl Reference Manual*.

Table 6–9. Stratix II Compile Report File Descriptions

Switch	Tool	Description
*<revision>***.map.rpt**	Analysis & Synthesis	Synthesis settings, source files, messages and resource usage
*<revision>***.map.eqn**	Analysis & Synthesis	Implementation equations and device resource instantiations
*<revision>***.fit.rpt**	Fitter	Fitter settings, layout optimizations, resources, pin-out and messages
*<revision>***.fit.eqn**	Fitter	Implemented equations and device resource instantiations after fitting
*<revision>***.drc.rpt**	Design Assistant	Design rule settings, violations, and messages
*<revision>***.upc.rpt**	Timing Constraint Checker	Constraint coverage information
*<revision>***.asm.rpt**	Assembler	Assembler settings, **.pof** and **.sof** output file options, and messages
*<revision>***.rec.rpt**	Companion Revision Comparison	A status report on the structural comparison between the HardCopy II revision and the Stratix II Prototype design
*<revision>***.flow.rpt**	Flow	Resource summary and execution time for each tool in the flow. This report is updated as different tools in the flow complete

Comparing the FPGA & HardCopy Revisions

Before submitting the HardCopy II project to the Altera Design Center, it should be checked against the Stratix II prototype FPGA revision. To do this, run the `execute_hardcopyii` Tcl command with the `-compare` option from the `quartus_sh` shell:

```
tcl> execute_hardcopyii -compare
```

Running this command generates a report file and summary file in the project directory. These files are called *<revision_name>*.**rec.rpt** and *<revision_name>*.**rec.summary**. The command checks to verify that the following items conform to HardCopy II design rules and are consistent between the HardCopy II and Stratix II revisions:

- Source design files and device netlist files
- User clock assignments
- Timing constraints (assignments)
- I/O location and type assignments
- PLL parameters
- Memory implantation parameters
- DSP implementation parameters
- Global resource properties
- Properties of all other device resources used

Any errors or failures in comparison are reported in the **.rec** report files. An example **.rec** file is given below. Note that for this example, the design comparison checks in the HardCopy II Companion Revision Comparison Summary table are all marked passed, indicating that the HardCopy II design in the Quartus II software is finished and ready for hand-off to the back-end engineering team in the Altera Design Center.

You must resolve any failures that show up in the Comparison Summary before you proceed any further with your design.

```
HardCopy II Companion Revision Comparison report for demo_design_hardcopyii
---------------------
; Table of Contents ;
---------------------
  1. Legal Notice
  2. HardCopy II Companion Revision Comparison Summary
  3. HardCopy II Companion Revision Comparison Messages
+-------------------------------------------------------------------------+
; HardCopy II Companion Revision Comparison Summary                       ;
+-----------------------------------------------+---------------------+
; HardCopyII Companion Revision Comparison Status; Successful - Mon   ;
; Jun 13 16:53:42 2005                          ;                     ;
; Quartus II Version                            ; 5.0 Build 144       ;
; 03/31/2005 SJ Full Version                    ;                     ;
; Revision Name                                 ;                     ;
; demo_design_hardcopyii                        ;                     ;
; Top-level Entity Name                    ; demo_design       ;
```

```
; Family                                        ; HardCopy II        ;
; Source Files Compared                         ; Passed (23/23)     ;
; User Clocks Compared                          ; Passed (1/1)       ;
; Assignments Compared                          ; Passed             ;
; I/O Structure Compared                        ; Passed (51/51)     ;
; Package Pins Compared                         ; Passed (1020/1020) ;
; PLL Structure Compared                        ; Passed (1/1)       ;
; PLL Clocks Compared                           ; Passed (1/1)       ;
; RAM Information Compared                      ; Passed (1/1)       ;
; DSP Information Compared                      ; Passed (1/1)       ;
; Global Resources Compared                     ; Passed (1/1)       ;
; Resource Counts Compared                      ; Passed (5/5)       ;
+-----------------------------------------------+--------------------+

+-------------------------------------------------+
; HardCopy II Companion Revision Comparison Messages ;
+-------------------------------------------------+

Info: *******************************************************************
Info: Running Quartus II HardCopy II Companion Revision Comparison
Info: Command: quartus_cdb demo_design -c demo_design_hardcopyii --
compare=demo_design
```

Performing Static Timing Analysis

Static Timing Analysis in the Quartus II Software

The global assignments made for the Stratix II prototype and HardCopy II revisions ensure that Static Timing Analysis (STA) is run for both fast and slow operating conditions and both best and worst case timing is verified.

You can run the timing analysis independent of the compile process in one of two ways:

1. Use the **qexec** Tcl command to run a timing analysis Tcl script in `quartus_tan` from within the basic quartus shell, `quartus_sh`
2. Run the `quartus_tan` interactive Tcl shell independently and execute Tcl commands and scripts at the Tcl prompt

For more information on running static timing analysis in the Quartus II software, refer to the *Timing Analysis* section in the *Quartus II Handbook*.

For Tcl commands related to static timing analysis, refer to the Timing section of the *Tcl Packages & Commands* in the *Quartus II Scripting Reference Manual*.

Static Timing Analysis in Primetime

The Quartus II software can also generate files required to run STA in Synopsys' PrimeTime. To do this, you must set the EDA_TIMING_ANALYSIS_TOOL and EDA_OUTPUT_DATA_FORMAT variables before compiling the design. The following example Tcl commands direct the Quartus II software to generate PrimeTime files for STA.

```
## Tcl Script to Set up for PrimeTime STA File Output
set_global_assignment -name EDA_TIMING_ANALYSIS_TOOL "PrimeTime (Verilog)"
set_global_assignment -name EDA_OUTPUT_DATA_FORMAT VERILOG -section_id eda_timing_analysis
```

The files generated by the Quartus II software are organized in a subdirectory within the project directory. For example, after compiling a Stratix II prototype design (demo_design), the following verilog (**.vo**) SDF (**.sdo**) and PrimeTime Tcl script (**.tcl**) are created in the project directory.

```
timing\
      primetime\
          demo_design_min.vo
          demo_design_v_min.sdo
          demo_design_pt_v_min.tcl
```

The Tcl script includes all timing constraints applied during the Quartus II software compilation.

It is important to note that the PrimeTime files generated by default, for a Stratix II Prototype design, are for fast-corner (best-case timing) analysis. This is an artifact of the DO_COMBINED_ANALYSIS global variable set at the start of the flow.

To generate PrimeTime files for slow-corner (worst-case timing) analysis:

1. Disable the DO_COMBINED_ANALYSIS global variable
2. Re-run the timing analysis.
3. Run the EDA netlist writer.

The following example script shows one way to do this. The script first turns off combined fast/slow analysis, runs the timing analyzer to create a timing netlist for slow-corner analysis, runs the EDA netlist writer to generate slow-corner analysis PrimeTime files, and then re-sets combined fast/slow analysis.

```
## Tcl Script to Generate slow-corner PrimeTime Files After Compiling
## the Startix II FPGA Prototype revision
set_global_assignment -name DO_COMBINED_ANALYSIS OFF
```

```
execute_module -tool tan
qexec "quartus_eda demo_design"
set_global_assignment -name DO_COMBINED_ANALYSIS ON
```

HardCopy II Example Tcl Script

The following script draws together the Tcl ideas discussed thus far into a top-level Tcl script for the `quartus_sh` Tcl shell. This script implements a HardCopy II design called demo_design. It begins by creating a new project, called `demo_design`, compiling the Stratix II FPGA prototype, creating a HardCopy II companion revision and then compiling the companion revision. Finally, the revision comparison tool is run to verify that both revisions are consistent.

In this example, global, pin, and timing assignment scripts are read into the top-level script using the Tcl **`source`** command. The sourced scripts are listed after the top-level script listing.

Top-Level Example Script demo_design.tcl

```
## demo_design.tcl
## Top-level script for executing a HardCopy II design in quartus_sh -s
load_package flow

## Open of create the Stratix II FPGA prototype revision
if [is_project_open] close_project
if {[project_exists demo_design]} {
  project_open demo_design

} else {
  project_new demo_design
}

## Apply global design settings
source global_assignments.tcl

## Apply I/O assignments
source pin_assignments.tcl

## Apply FPGA timing constraints
source timing_assignments.tcl

## Compile the Stratix II FPGA prototype design
execute_flow -compile

# #Create and switch to the HardCopy II target revision
execute_hardcopyii -create_companion demo_design_hcii
project_close
project_open demo_design -revision demo_design_hcii

## Compile the HardCopy II design revision
execute_flow -compile

## Check the HardCopy II revision and make sure it matches the FPGA
## design
execute_hardcopyii -compare
## Quit quartus_sh -s
qexit

## End of demo_design.tcl
```

Global Assignments Script global_assignments.tcl

The `global_assignments.tcl` script source in the top-level script, `demo_design.tcl` prepares global variables, target devices, and revision names for the HardCopy II project:

```
## global_assignments.tcl

## Source Design File Settings
## ==========================
set_global_assignment -name VERILOG_FILE demo_design.v
set_global_assignment -name VERILOG_FILE example_ram.v

## Stratix II Prototype FPGA Target Settings
## =========================================
set_global_assignment -name FAMILY "Stratix II"
set_global_assignment -name DEVICE EP2S90F1020C4
set_global_assignment -name TOP_LEVEL_ENTITY demo_design

## HardCopy II Companion Revision and Target Settings
## ================================================
set_global_assignment -name COMPANION_REVISION_NAME \
demo_design_hardcopyii
set_global_assignment -name DEVICE_TECHNOLOGY_MIGRATION_LIST HC230F1020

## Design Assistant Assignments and Settings Required for HardCopy II
## ==================================================================
set_global_assignment -name ENABLE_DRC_SETTINGS ON
set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 1
set_global_assignment -name REPORT_IO_PATHS_SEPARATELY ON
set_global_assignment -name FLOW_ENABLE_TIMING_CONSTRAINT_CHECK ON
set_global_assignment -name DO_COMBINED_ANALYSIS ON
set_global_assignment -name IGNORE_CLOCK_SETTINGS OFF
set_global_assignment -name ENABLE_RECOVERY_REMOVAL_ANALYSIS ON
set_global_assignment -name ENABLE_CLOCK_LATENCY ON

## End of global_assignments.tcl
```

Pin Assignments Script pin_assignments.tcl

The `pin_assignments.tcl` script run from the top-level script, `demo_design.tcl`, specifies top-level design signal to package ball assignments and I/O parameters:

```
## pin_assignments.tcl
set_location_assignment PIN_AH5 -to addr_out[0]
set_location_assignment PIN_AH6 -to addr_out[1]
set_location_assignment PIN_AJ5 -to data_in[0]
set_location_assignment PIN_AJ6 -to data_in[1]
set_location_assignment PIN_AJ32 -to resetn
set_location_assignment PIN_AM17 -to ref_clk

## I/O Type and Parameter Assignments
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to addr_out[0]
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to addr_out[1]
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to data_in[0]
set_instance_assignment -name IO_STANDARD "1.5-V HSTL CLASS II" -to data_in[1]
set_instance_assignment -name IO_STANDARD LVDS -to resetn
set_instance_assignment -name IO_STANDARD LVCMOS -to ref_clk

set_instance_assignment -name fast_input_register on -to data_in[0]
set_instance_assignment -name fast_input_register on -to data_in[1]
set_instance_assignment -name fast_output_register on -to addr_out[0]
set_instance_assignment -name fast_output_register on -to addr_out[1]

set_instance_assignment -name output_pin_load 10 -to addr_out[0]
set_instance_assignment -name output_pin_load 10 -to addr_out[1]

## End of pin_assignments.tcl
```

Timing Assignments Script timing_assignments.tcl

The `timing_assignments.tcl` script is run from the top-level script, `demo_design.tcl`, it applies timing constraints for the system clock, ref_clk, and I/O to core timing specifications.

```
## timing_assignments.tcl
create_base_clock –fmax 10.0ns –target ref_clk ref_clk

set_instance_assignment -name LATE_CLOCK_LATENCY 3ns -to ref_clk
set_instance_assignment -name EARLY_CLOCK_LATENCY 2ns -to ref_clk
set_clock_uncertainty –hold –to ref_clk 0.250ns
set_clock_uncertainty –setup –to ref_clk 0.250ns

# Input delay of 6ns (max) & 2ns (min) for bus data_in[1:0]
set_input_delay –clk_ref ref_clk –max –to data_in 6.0ns
set_input_delay –clk_ref ref_clk –min –to data_in 2.0ns
# Output delay of 6ns (max) & 2ns (min) for bus data_out[1:0]
set_output_delay –clk_ref ref_clk –max –to data_out 6.0ns
set_output_delay –clk_ref ref_clk –min –to data_out 2.0ns

# Don't care about timing on the resetn net. Set as false path
set_timing_cut_assignment -from resetn

## End of timing_assignments.tcl
```

Summary

This chapter introduced script-based design for HardCopy II devices using the Quartus II interactive Tcl shell. This approach provides you with an alternative to GUI-based design for certain situations such as remote-terminal Quartus II execution, design flow automation, or even if you are simply more comfortable operating in a scripting environment.

7. Timing Constraints for HardCopy II

H51028-1.0

Introduction

In a Stratix® II FPGA design, a complete and accurate set of timing constraints is often not critical to achieving a fully functioning product. The reconfigurability of the FPGA means that if a timing-related problem occurs during hardware test and verification the device can be reprogrammed to correct it. No ASIC re-spin or board-level work-around is necessary and the fix can be implemented in a timely and cost-effective way.

In contrast, a HardCopy® II design results in a mask-programmed, structured ASIC device. If timing problems are evident when the device is tested in hardware, then either a board-level workaround is required or, more likely, a re-spin. This can result in design-change turn-around times of the order of months and high NRE (non-recurring engineering expense for design services and mask generation) costs.

To reduce the risk of a re-spin and ensure smooth transition through the Quartus® II software and back-end design in the Altera® HCDC (HardCopy Design Center) Altera recommends that you follow the timing considerations and timing constraint recommendations given in this chapter.

This chapter includes the following information:

- A description of timing-related differences between Stratix II FPGAs and HardCopy II structured ASICs.
- An explanation on the use of timing constraints in the Quartus II software, including some of the important timing-related checks reported by the HardCopy II Advisor and Design Assistant.
- Timing constraint recommendations for your HardCopy II project are given together with recommendations for handling legacy designs that use timing constraints not supported in the HardCopy II design flow.

Stratix II vs. HardCopy II Timing

The back-end design of your HardCopy II structured ASIC includes timing closure in accordance with the timing specification achieved in the Quartus II software for the Stratix II FPGA prototype and HardCopy II device. However, you should be aware that this does not mean that actual path timing in the Stratix II FPGA is duplicated in the HardCopy II device. In fact, because of the architectural differences between Stratix II

and HardCopy II devices, you should expect that while internal and I/O path timing are within whatever timing constraints you applied, actual path delays are different.

The key factors that impact timing differences between Stratix II and HardCopy II devices are listed below.

- The HardCopy II die is significantly smaller than its Stratix II counterpart
- Course-grain ALMs in Stratix II devices are mapped to fine-grain HCell macros in HardCopy II devices
- Design connections are implemented using custom metal routing in HardCopy II devices
- HardCopy II devices contain no SRAM-configurable programmable connection points
- Leaf sub-trees in HardCopy II global clock networks are custom routed

The following sections briefly describe the affect of these factors on HardCopy II timing characteristics.

Internal Register-to-Register Timing

Internal timing is the timing of paths from register to register within core logic. Internal timing is dependent on the transport delays of logic elements on register-to-register paths and the overall effects of parasitic capacitance, parasitic resistance, and crosstalk on routing connections between those logic elements.

User-logic implementation in HardCopy II devices is more area efficient and has improved timing when compared with the Stratix II FPGA. This is a result of the re-mapping of coarse-grain, programmable ALMs in Stratix II devices to fine-grain HCell macros in HardCopy II devices. All ALM and DSP functions are re-mapped to HCells in HardCopy II devices. Using fine-grain HCells eliminates the need for the programmable routing MUXs found inside the Stratix II ALM and DSP blocks. This reduces the number of levels of logic required to implement ALM functions from the Stratix II device. Consequently, the transport, or propagation delays, associated with logic elements in register-to-register paths are smaller.

The HardCopy II device does not require configuration SRAM, so programmable routing resources and SRAM-configurable ALMs, die size is significantly smaller than Stratix II counterpart devices. One effect of reduced die size is that overall routing length is shorter. In addition, HardCopy II devices use personalization of metal layers 5 and 6 to implement user-logic connections. This eliminates the need for

SRAM-configurable routing switches and programmable connection points, all of which adversely affect timing. This means that, overall, parasitic capacitance and resistance, and crosstalk are minimized, leading to faster connections than those found in the Stratix II FPGA.

Faster logic element implementation and faster routing in HardCopy II devices generally result in faster register-to-register paths and higher overall clock frequencies. Software place-and-route tools have a significant impact on timing results, however, so there are cases where Stratix II register-to-register paths are faster than the corresponding paths in the HardCopy II device.

I/O Path Timing

The actual timing and parametric characteristics of I/O cells in HardCopy II devices are very similar to those in Stratix II devices. You should expect, however, to see differences in I/O signal path timing. These differences are primarily because of timing differences in core-to-I/O and clock distribution.

For core-to-I/O timing, one of the largest influencing factors is the timing behavior of signal paths, as described in the "Internal Register-to-Register Timing" section. In general, core-to-I/O and I/O-to-core timing are different between HardCopy II and Stratix II devices.

The other major influence on I/O timing is clock distribution differences between HardCopy II and Stratix II. Shorter, faster clock trees, custom clock tree buffering and custom routing of leaf sub-trees in HardCopy II mean that insertion delays, latencies, skew characteristics, jitter, and PLL compensation are different from the Stratix II FPGA. The effect of this is described in the "Clock Distribution Effects" section.

Clock Distribution Effects

The HardCopy II structured ASIC has a clock distribution scheme that is similar to that in Stratix II FPGAs with some notable differences:

- There are no SRAM-programmable switches and routing connections
- Reduced die-size means shorter overall clock tree routing length
- Leaf sub-trees of clock networks are custom routed using metal 5 and 6

These physical differences impact clock distribution characteristics across the device. Timing characteristics most affected are:

- Clock tree latency and clock insertion delay

- Clock skew
- Clock jitter
- PLL compensation delays

In general, clock tree latencies are smaller in the HardCopy II device because of shorter routing length and the absence of SRAM-programmable switches. As a result, you should expect that any clock insertion delays that are modeled will also be shorter.

The most significant impact of reduced clock tree latency is that core-to-I/O and I/O-to-core timing changes. For example, if an I/O register is clocked earlier because of reduced clock latency, then the arrival time of the register output at the device pin will be reduced. Similarly, if an input register is clocked earlier, then the setup time for that register will also be earlier, and the hold time requirement will be relaxed.

The Quartus II software accommodates these differences to ensure that your timing requirements are satisfied. However, you should be aware that reduced clock insertion delay causes I/O timing differences between your Stratix II FPGA prototype and a HardCopy II-structured ASIC.

PLL Characteristics

Many of the effects described in the "Clock Distribution Effects" section also apply to the clock outputs from PLLs between Stratix II and HardCopy II devices. The Quartus II software implements compensation delays for PLLs in your HardCopy II device to account for differences in PLL clock distribution. This ensures that the compensation modes used in the Stratix II FPGA are also used in the HardCopy II-structured ASIC.

HardCopy II Timing Closure Methodology

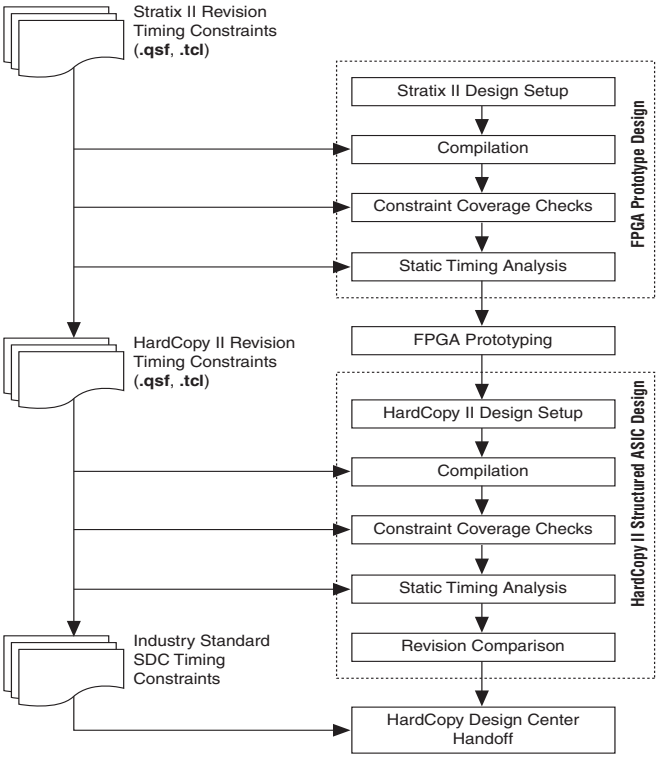
To achieve timing closure for your HardCopy II-structured ASIC, it is imperative that you use a complete set of accurate timing constraints throughout the flow. For the Stratix II FPGA prototype, although you may verify timing and functionality in hardware, it is essential that the design is compiled and verified in the Quartus II software using a complete set of timing constraints. These constraints feed-forward to the HardCopy II revision of the project, and ultimately to the HCDC. The back-end design of your structured ASIC in the HCDC ensures that it conforms to whatever timing constraints are satisfied in the Quartus II software. It is important to remember that while the Quartus II timing constraints are respected, the actual Stratix II FPGA prototype timing you observe in hardware is not duplicated in the HardCopy II-structured ASIC.

HardCopy II Timing Closure Flow

HardCopy II timing closure methodology is comprehensive and includes a powerful Static Timing Analysis (STA) engine in the Quartus II software, an interface to a third-party static timing analyzer, and FPGA-prototype timing verification in hardware.

The timing closure methodology used in the Quartus II software for a HardCopy II design is shown in Figure 7–1. This diagram shows the FPGA-first flow. For the HardCopy II-first flow, the methodology is the same except that the HardCopy II compilation is performed before the Stratix II compilation.

Figure 7–1. Stratix II First Timing Closure Flow



As you can see from Figure 7–1, timing constraints are used very early in the Quartus II design flow. During the Stratix II FPGA prototype compilation, these constraints are used as the timing target for timing-driven compilation. When the compilation is complete, the Quartus II STA engine reports timing results for your design. Any failed

timing reports mean that you must either modify your timing constraints, change your compile settings and recompile, or both. In addition, by using the timing constraint checker and HardCopy II Advisor you can access check reports for unconstrained paths (the UCP report) and unsupported timing assignments. For timing verification in third-party tools, the Quartus II software can generate static timing analysis scripts for use in Synopsys PrimeTime tools. In addition, timing can be further verified in third-party, timing-driven simulation tools.

When software timing verification of the Stratix II prototype FPGA is complete, you can verify your prototype in hardware. It is a requirement of the HardCopy II design flow that you fully verify the Stratix II FPGA prototype timing over the range of operating conditions that your design is exposed to.

The next step is to create and compile your HardCopy II design revision. By default, your HardCopy II compilation is run with the same timing constraints used during the compilation and verification of your Stratix II FPGA. If you wish to change the target timing specifications for the HardCopy II revision, you can do so by changing the HardCopy II timing constraints before compiling. When the HardCopy II compilation is complete, like the Stratix II compilation, run the Quartus II STA tool to check timing results. You should review and resolve any timing failures that are reported.

One of the final steps in the HardCopy II design flow in the Quartus II software is the revision comparison check. Part of this check compares timing constraints and settings between the Stratix II and HardCopy II revisions of the project. Any differences between the two are reported.

When your Quartus II design is transferred to the HCDC it includes an industry-standard (SDC) version of the HardCopy II timing constraints. These timing constraints are automatically converted from Quartus II timing constraints and are used as the basis for timing closure using industry-standard design and verification tools.

Quartus II Timing Related Checks & Settings

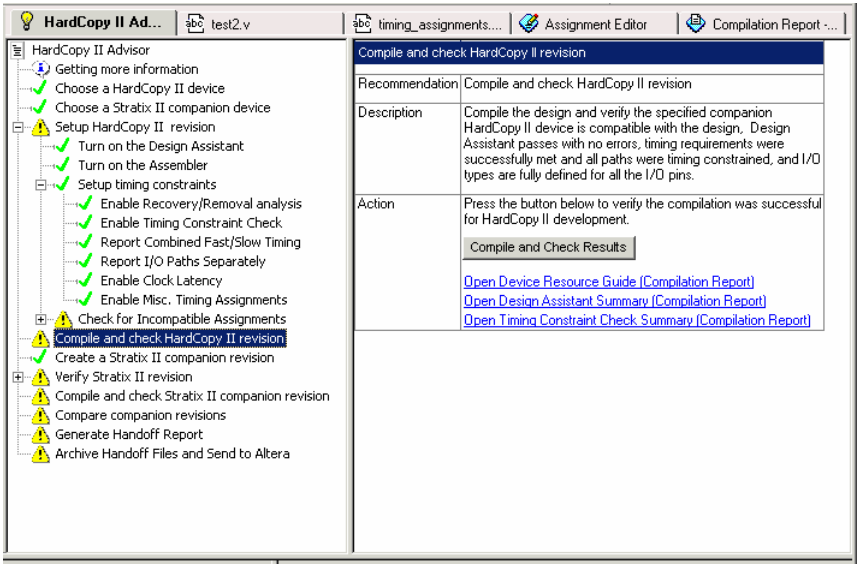
The Quartus II software provides a number of timing related checks as you go through a HardCopy II design flow. The HardCopy II Advisor can guide you through these checks and ensure that you perform all steps required to successfully complete a HardCopy II design.

For more information on the HardCopy II Advisor and the checks performed by the Design Assistant, refer to the *Design Guidelines for HardCopy Series Devices* chapter in the *Hardware Design Considerations* section of the *HardCopy Series Handbook*.

Some of the first timing-related issues addressed as you go through the HardCopy II Advisor are settings for timing analysis (Figure 7–2). These settings are necessary to make sure you generate accurate and complete timing reports and include the following:

- Enable Recovery/Removal Analysis
- Enable Timing Constraints Check
- Report Combined Fast/Slow Timing
- Report I/O Paths Separately
- Enable Clock Latency
- Enable Misc. Timing Assignments

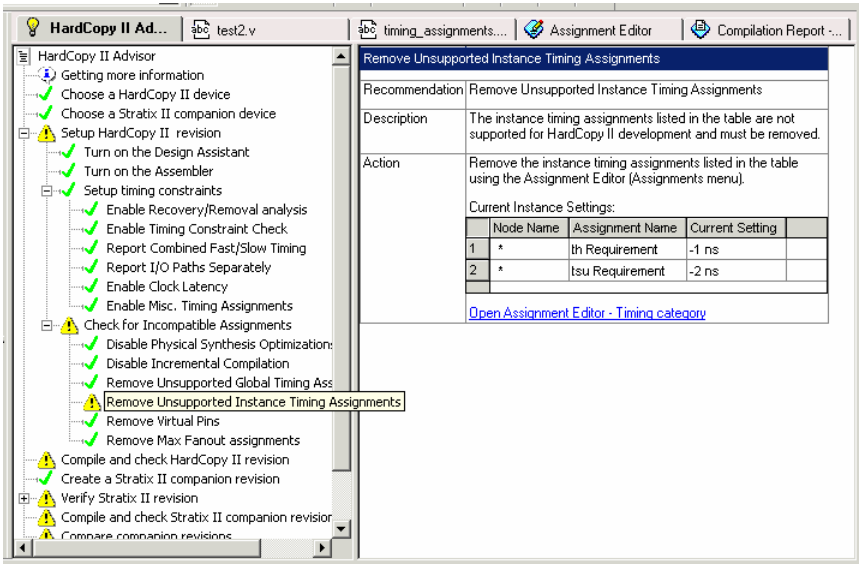
Figure 7–2. Timing Related Settings in the HardCopy II Advisor



In the HardCopy II Advisor design flow, the **Check for Incompatible Assignments** section (Figure 7–3) contains two subsections, Remove Unsupported Global Timing Assignments and Remove Unsupported Instance Timing Assignments. Both of these sections list timing constraints that are incompatible with the HardCopy II design flow. These constraints are explained in "Unsupported HardCopy II Timing Constraints" on page 7–19.

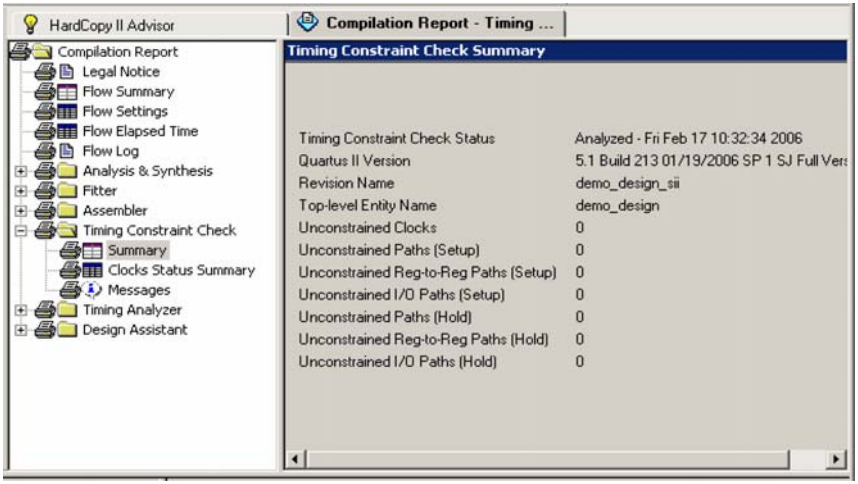
While timing analysis in the Quartus II software completes successfully with timing constraints, it is very important that all unsupported timing assignments are corrected before you transfer the HardCopy II design to the HCDC. Failure to do so results in delays during back-end timing closure.

Figure 7–3. Unsupported Timing Assignments in the HardCopy II Advisor



In both Stratix II and HardCopy II revisions of your project, after compilation is complete, the Compilation Report includes a **Timing Constraints Check** section. This section checks the coverage of the timing constraints used in the design and reports any unconstrained paths that may exist. You should endeavor to make sure that all paths, internal and I/O, and all clock domains are constrained for both setup and hold checks.

Figure 7–4. Timing Constraint Check in the Compilation Report



In addition to these timing-related checks, you should review the Quartus II timing report sections in the **Compilation Report** and resolve any timing violations that may be reported.

Recommended HardCopy II Timing Constraints

To ensure a smooth transition of your design through the Quartus II software and through the back-end design process in the HCDC, you should use only recommended timing constraints. Recommended constraints, detailed in the following sections, are fully supported in the industry-standard SDC format constraints translation that is performed as part of the final design step in the Quartus II software.

The following sections use the TCL commands for timing constraints.

To see the equivalent assignments using the Assignment Editor, refer to the *Timing Analysis* section in the *Verification* volume of the *Quartus II Handbook*.

For more information on Tcl support in the Quartus II software and Tcl commands for timing constraints, refer to the *Tcl Scripting Reference Manual* and the *Tcl Scripting* chapter in the *Design Implementation & Optimization* volume of the *Quartus II Handbook*.

Core Clock Domains

Every clock domain in your design should be fully constrained to represent clock period and duty cycle. You should also specify clock uncertainty to model jitter and provide some performance margin.

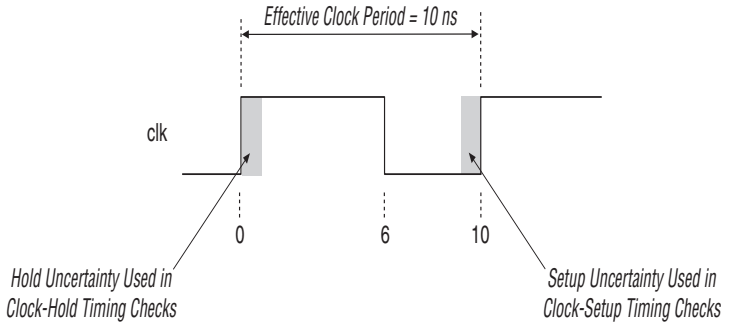
With clock latency enabled in the timing settings in the Quartus II software, use the following Tcl commands to constrain each clock domain in your design:

- `create_base_clock`
- `set_clock_uncertainty -setup`
- `set_clock_uncertainty -hold`

The `create_base_clock` Tcl command specifies the period of a clock, the source pin or port for the clock, and the duty cycle of the clock waveform. The `set_clock_uncertainty` Tcl commands model any jitter and margin associated with the clocks. Different uncertainty settings can be used for clock setup and clock hold timing checks. The following example specifies a 100 MHz clock with a 50-50 duty cycle, setup uncertainty of 150 ps, and hold uncertainty of 250 ps. The clock is called `clk` and the source is pin `clk_pin`. Figure 7–5 shows the waveform associated with this clock.

```
Tcl> create_base_clock -fmax 10ns -duty_cycle 50 -target clk_pin clk
Tcl> set_clock_uncertainty 0.15ns -to clk -setup
Tcl> set_clock_uncertainty 0.25ns -to clk -hold
```

Figure 7–5. Clock Waveform Specification



PLL Clocks

The clock settings for PLL clocks are derived automatically based on the PLL settings and reference clock characteristics. You can also override the default PLL clock settings for timing analysis by specifying clock settings for the input clock port on the PLL.

Clock uncertainty in PLL clock outputs is not modeled by default. You should use the `set_clock_uncertainty` command to model jitter and any other uncertainty and margin in your PLL clocks.

Please consult with your Altera FAE (Field Applications Engineer), or use MySupport regarding PLL jitter specifications for your design.

I/O Input Timing

For I/O to core paths the recommended Tcl command is `set_input_delay`. The syntax for this command is:

```
set_input_delay [-clk_ref <clock>] [-clock_fall]
[-max] [-min] -to <input_pin> <value>
```

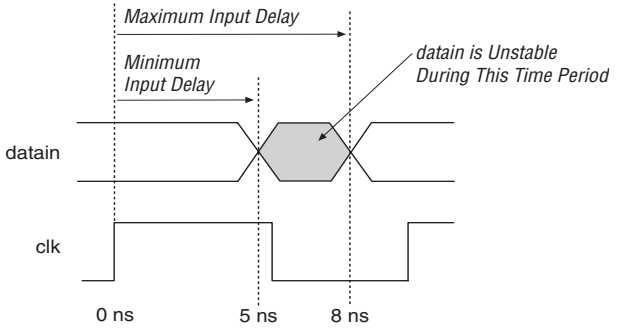
This Tcl command specifies the signal delay, relative to a clock, <`clock`>, from an external device, to the input pin, <`input_pin`>, on the device. Use both the -max and -min options for maximum and minimum external delays and to ensure setup and hold checks, respectively, during timing analysis in the Quartus II software.

The following example shows the `set_input_delay` command with an off-chip delay of 8 ns (max.) and 5 ns (min.) to the input pin, `datain`, on an Altera device, which is relative to the rising edge of clock, `clk`.

```
Tcl> set_input_delay -clk_ref clk -max -to datain 8ns
Tcl> set_input_delay -clk_ref clk -min -to datain 5ns
```

The input delay timing constraints are illustrated in Figure 7–6.

Figure 7–6. Input Delay Settings



There are scenarios where using `set_input_delay` does not accurately model the delays on an input path. For these scenarios, the recommended timing constraints that you should use are `setup_relationship` and `hold_relationship`. These constraints allow you to specify specific timing requirements from an input pin to specific registers.

For example, Figure 7–7 shows a DDIO input arrangement where two registers are fed by input pin `din`. Register r1 is clocked by the positive edge of clock, `clk`, and register r2 is clocked by the negative edge. Using `set_input_delay` in this example does not correctly model the timing requirement to the negative edge triggered register. A better solution is to use `setup_relationship` and `hold_relationship`, as shown in the following Tcl example. Figure 7–8 shows the waveforms associated with these settings.

```
Tcl> set_input_delay -max -clock_ref clk -to din 0ns
Tcl> set_instance_assignment -name setup_relationship -from din -to r1 4ns
Tcl> set_instance_assignment -name setup_relationship -from din -to r2 4ns
Tcl> set_instance_assignment -name hold_relationship -from din -to r1 2ns
Tcl> set_instance_assignment -name hold_relationship -from din -to r2 2ns
```

In this Tcl example, the first command is the `set_input_delay` command. The `setup_relationship` and `hold_relationship` timing constraints generally support only register to register paths, so `set_input_delay` is necessary. It changes the way the `din` input pin is handled so that, for the purposes of timing analysis, it can be treated as a register.

Figure 7–7. Timing Constraint Check in the Compilation Report

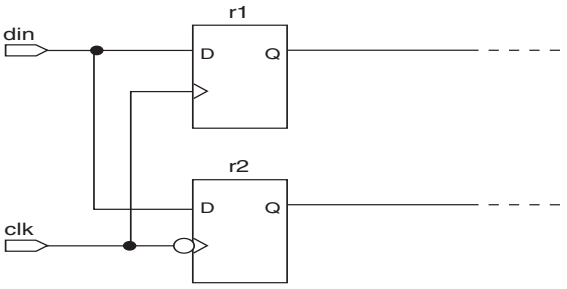
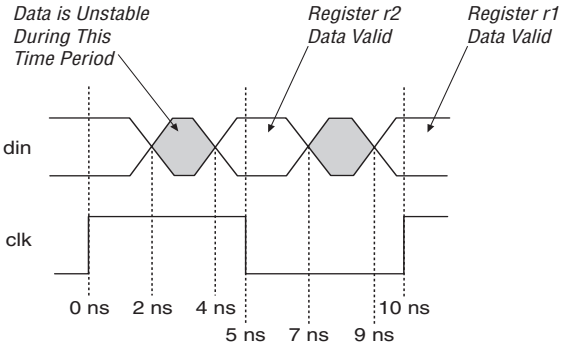


Figure 7–8. Double Data Rate Timing Waveforms



I/O Output Timing

The recommended Tcl command for core to I/O paths is `set_output_delay`. The syntax for this command is:

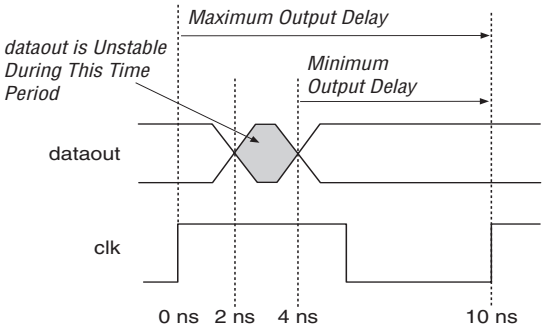
```
set_output_delay [-clk_ref <clock>] [-clock_fall] [-max] [-min]
-to <output_pin> <value>
```

This Tcl command specifies the signal delay, relative to a clock, <`clock`>, from an output pin on the Altera device, <`output_pin`>, to an external device. Use both the `-max` and `-min` options for maximum and minimum external delays and to ensure setup and hold checks, respectively, during timing analysis in the Quartus II software.

The following example shows the `set_output_delay` command with an off-chip delay of 8 ns (max.) and 6 ns (min.) from the output pin, `dataout`, relative to the rising edge of clock, `clk`. Associated waveforms are shown in Figure 7–9.

```
Tcl> set_output_delay -clk_ref clk -max -to dataout 8ns
Tcl> set_output_delay -clk_ref clk -min -to dataout 6ns
```

Figure 7–9. Output Delay Timing Settings



You should use the `set_output_delay` Tcl command to model timing delays from an output pin on your Altera device to external, off-chip destinations. In some cases, however, you may need to constrain several different path delays from registers inside your Altera device to an output pin. In these cases, you should use the `setup_relationship` and `hold_relationship` timing constraints as described in the "I/O Input Timing" section.

The following example shows how these commands are used to perform timing analysis on the DDIO output arrangement (Figure 7–10).

As for the DDIO input arrangement, as described in the "I/O Output Timing", the `set_output_delay` Tcl command is first used so that the `setup_relationship` and `hold_relationship` assignments take effect. The timing waveforms shown in Figure 7–11 show the setup and hold relationships relative to the clock, `clk`.

```
Tcl> set_output_delay -max -clock_ref clk -to dout 0ns
Tcl> set_instance_assignment -name setup_relationship -from r1 -to dout 4ns
Tcl> set_instance_assignment -name setup_relationship -from r2 -to dout 4ns
Tcl> set_instance_assignment -name hold_relationship -from r1 -to dout 2ns
Tcl> set_instance_assignment -name hold_relationship -from r2 -to dout 2ns
```

Figure 7–10. DDIO Output Arrangement

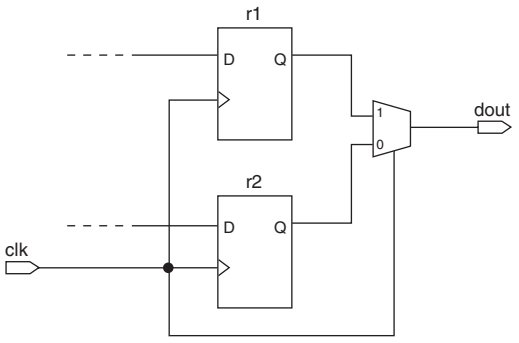
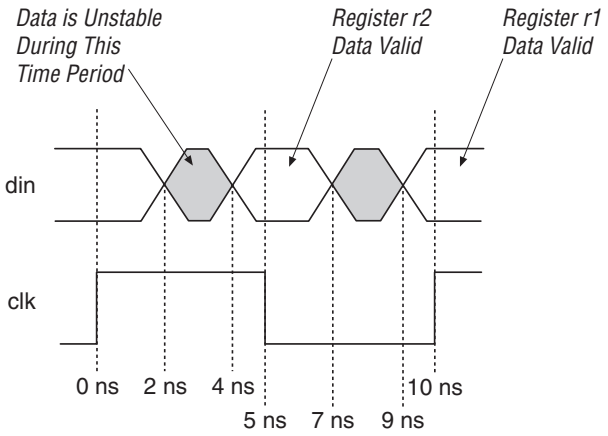


Figure 7–11. Waveforms for the DDIO Output Arrangement



False Paths

False paths describe paths in your design that are not considered during timing analysis. Actual timing achieved on these paths is irrelevant to timing closure and are not be reported by the Quartus II Timing Analyzer.

The following Tcl command specifies a false path.

```
set_timing_cut_assignment [-comment <comment>] [-from from_pin_list>]
[-to <to_pin_list>]
```

This Tcl command specifies false paths from one or more source pins, or time groups, to one or more destination pins, or time groups. By specifying clock pins as the source and destinations, all signal paths between the two clock domains are disabled for timing analysis.

The following example of a Tcl fragment shows the use of the `set_timing_cut_assignment` command in disabling timing analysis for the two busses, `data` and `addr`, from clock domain `clk1` to clock domain `clk2` (Figure 7–12).

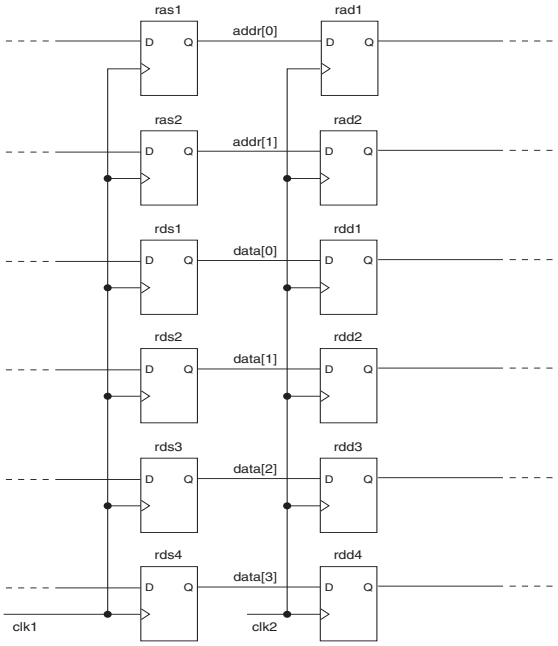
```
Tcl> timegroup "src_group" -add_member "ras*"
Tcl> timegroup "src_group" -add_member "rds*"
Tcl> timegroup "dst_group" -add_member "rad*"
Tcl> timegroup "dst_group" -add_member "rdd*"
Tcl> set_timing_cut_assignment -from "src_group" -to "dst_group"
```

Alternatively, if you want to disable timing for all paths from clock domain `clk1` to clock domain `clk2`, you can use the following example command.

```
Tcl> set_timing_cut_assignment -from "clk1" -to "clk2"
```

This command only disables paths for signals originating in domain `clk1` to destinations in domain `clk2`. It does not disable timing for signals originating in domain `clk2` to destinations in domain `clk2`.

Figure 7–12. Signals addr & data From Domain clk1 to Domain clk2



Multicycle Paths

Multicycle paths are paths within a single clock domain or between different clock domains whose delay should be within some multiple of the source or destination clock periods. In the Quartus II Timing Analyzer, a multicycle constraint on a path may be used for setup time analysis, hold time analysis, or both. The syntax for the Tcl multicycle constraint command is given below.

```
set_multicycle_assignment [-comment <comment>] [-disable] [-end]
[-from <from_list>][-hold] [-remove] [-setup] [-start] [-to <to_list>]
<path_multiplier>
```

Multicycle timing constraints are useful in a variety of situations. You can use them to accurately control timing analysis for functions in a design that may have long combinatorial paths and take multiple clock cycles to complete, such as large, combinatorial multiply or divide functions. You can also use them to impose a timing constraint on cross clock domain transfers. Another common use of multicycle paths is to impose a timing

constraint on infrequently changing control signals where, because of the sequence of operations in the design, the impact of a change in those signals does not need to resolve in a single cycle.

The following example shows the use of the multicycle path Tcl command in controlling the operation of the combinatorial multiplier in Figure 7–13. By using both setup and hold multicycle constraints, the operation of the multiplier is timed to complete in exactly three clock cycles. This enables new input operands to be applied to the multiplier on each clock cycle, with results available on successive clock cycles at the output register, with a latency of three clock cycles. The waveforms shown in Figure 7–14 show the timing relationship between source operands and the result.

```
Tcl> timegroup "src_group" -add_member "r_opa"
Tcl> timegroup "src_group" -add_member "r_opb"
Tcl> set_multicycle_assignment -setup -from "src_group" -to "r_res" 3
Tcl> set_multicycle_assignment -hold -from "src_group" -to "r_res" 3
```

Figure 7–13. Example Multicycle Path Application

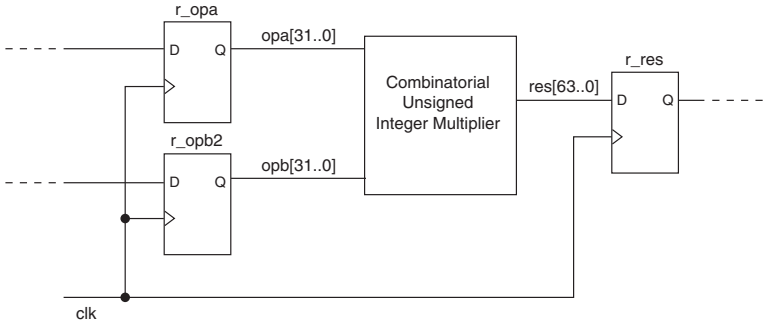
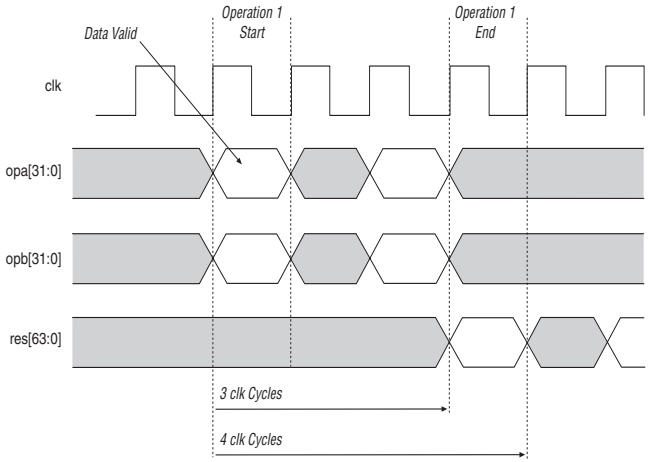


Figure 7–14. Waveforms for 3-Cycle Multiplier



Unsupported HardCopy II Timing Constraints

The Quartus II software supports a wide variety of complex timing constraints. In a HardCopy II design flow, however, some of these constraints are not translated to SDC format constraints when the design is transferred to the HCDC. The unsupported timing constraints for HardCopy II are listed below:

- Clock enable multicycle paths
- Inverted clocks
- TSU, Th, TCO, and Min TCO
- Internal TPD
- Virtual clocks
- Maximum clock and data skew
- Maximum and minimum delay

If these constraints are used, you can still perform timing analysis in the Quartus II software and produce the correct results. When a HardCopy II archive for handoff is created, however, they will be ignored. The translation of Quartus II timing constraints to SDC constraints simply drops unsupported constraints and they do not feed forward to the HCDC. Any unsupported constraints in a design are listed under the **Incompatible Assignments** section in the HardCopy II Advisor (refer to Figure 7–3).

While it is possible to translate unsupported constraints to constraints that are supported, the process is difficult and error-prone often requiring detailed analysis of the particular context in which the constraint is used. For this reason, Altera recommends that you use only supported timing constraints from the start of your HardCopy II project. This avoids any translation or constraint coverage issues that may occur later in a project and the inevitable delay and risk that results.

In some cases, a HardCopy II project in the Quartus II software may already be using the unsupported constraints, and you may choose either to translate the existing, unsupported constraints, or replace them with a new set of constraints that use only the recommended HardCopy II timing assignments. In many cases, you may find it easier to rebuild the constraints rather than translate existing constraints. This is because of the ambiguous nature of many unsupported timing constraints, which often require additional information outside of the Quartus II software before the translation can be properly resolved. Verifying that the translations produce the same timing constraint coverage and the same timing analysis results can also be a time-consuming and error-prone exercise.

If you do wish to translate existing, unsupported timing constraints to recommended constraints, use Table 7–1 as a rough guide. It shows how values used in TCO, Th, TSU, and Min TCO assignments normally convert to values used in recommended HardCopy II assignments. In the table, unsupported constraints are listed in the left hand column. Recommended constraints are listed along the top row. To use the table, cross-reference the unsupported constraints you wish to translate against a recommended constraint. The cross reference cell contains the conversion of the original, unsupported constraint value that should be used with the new, recommended constraint. It is very important to note that these translations are not valid in every design scenario.

Table 7–1. TSU, TH, TCO, & Minimum TCO Timing Constraint Conversion Notes (1), (2), (3), (4), (5)

	setup_relationship	set_input_delay	hold_relationship	set_output_delay
TSU Req	TSU	-max <TCK-TSU>		
Th Req		-min Th	-Th	
TCO Req	TCO			-max <TCK-TCO>
Min T_{CO} Req			Min T_{CO}	-min <- Min T_{CO}>

(1) TSU = value used in the TSU requirement assignment.
(2) TCO = value used in the TCO requirement assignment.
(3) Th = value used in the Th requirement assignment.
(4) Min TCO = value used in Min TCO requirement assignment.
(5) TCK = period of the clock for registers associated with the TSU and TCO requirements.

Conclusion

This chapter described timing considerations and timing constraint recommendations for HardCopy II projects in the Quartus II software. By understanding these considerations and following the recommendations in your design, you ensure a smooth transition through the Quartus II software and subsequent transfer to the Altera HardCopy Design Center for the back-end design of your structured ASIC. Following the recommendations in this chapter will help ensure success in your HardCopy II project.

8. Migrating Stratix II Device Resources to HardCopy II Devices

H51024-1.1

Introduction

Altera® HardCopy® II devices and Stratix® II devices are both manufactured on a 1.2-V, 90-nm process technology and offer many similar features. Designers can use the Quartus® II software to migrate their Stratix II design to a HardCopy II device. The Quartus II software will ensure that the design revision targeting a HardCopy II device retains the same functionality as the original Stratix II design.

Beginning with version 5.0 of the Quartus II software, you can select a HardCopy II companion device from the **Device Settings** dialog box (Device menu). Selecting a HardCopy II device as a companion device is similar to adding another Stratix II device in the migration device chain. The Quartus II software will compile the design to use the common resources available in all of the selected Stratix II devices and the selected HardCopy II devices. The HardCopy II companion device becomes the target device when you switch to the HardCopy II flow from this Stratix II flow later in the Quartus II project compilation.

For more information on compiling with Stratix II and HardCopy II companion revisions using Quartus II software, refer to the *Quartus II Support for HardCopy II Devices* chapter of the *HardCopy Series Devices Handbook*.

When you select a HardCopy II companion device, you can set the Quartus II Compiler to limit the design to the minimum resource availability of memory blocks and available logic for digital signal processing (DSP) from either the targeted Stratix II or HardCopy II companion device. Additional limitations also include I/O pin assignments and phase-locked loops (PLLs). This document is a guide for designers migrating Stratix II designs into HardCopy II devices. This document highlights resources that are not supported by the selected Stratix II and HardCopy II companion device pair or any resource differences between Stratix II devices and the HardCopy II device.

This document includes the following topics:

- Stratix II & HardCopy II Migration Options
- I/O Support & Planning
- External Memory Interface Support
- On-Chip Termination
- Stratix II & HardCopy II Companion Memory Blocks
- PLL Planning & Utilization

- Global & Local Signals
- Stratix II ALM Adaptation into HardCopy II Logic
- HardCopy II DSP Implementation from Stratix II DSP Blocks
- JTAG BST & Extended Functions
- Power Up & Configuration Compatibility

Stratix II & HardCopy II Migration Options

The Quartus II software allows you to migrate between different Stratix II devices in the same package. When compiling Stratix II designs in the Quartus II software, you can specify one Stratix II target device and one or more Stratix II migration devices. When you specify at least one migration device, the Quartus II Compiler constrains the overall design's I/O pins and other resource assignments to the minimum resources available in any of the selected migration devices. This feature allows vertical migration between devices using the same package footprint. To create the proper configuration file for one of the Stratix II devices selected in the migration devices menu, select that device as a target device.

The introduction of HardCopy II provides an additional seamless migration path for Stratix II devices. After you select a particular Stratix II device, the Quartus II software provides migration options in the Settings dialog box. For example, if your design targets the EP2S130 device in the 1,020-pin FineLine BGA® package, the Quartus II software provides the EP2S90 and EP2S180 devices in the 1,020-pin FineLine BGA package as migration options as well as the HC230 device in the 1,020-pin FineLine BGA package.

Conversely, the HardCopy II architecture allows you to design a structured ASIC and then prototype with a wide range of Stratix II devices. If the target device is a HardCopy II HC220 device in the 780-pin

FineLine BGA package, you can select the Stratix II EP2S90 or EP2S130 device in the 780-pin FineLine BGA package as prototype devices. Table 8–1 shows vertical migration options by package.

Table 8–1. Stratix II & HardCopy II Migration Options *Note (1)*

Device	FineLine BGA Package					
	484 Pins	672 Pins	780 Pins	1,020 Pins	1,020 Pins	1,508 Pins
HardCopy II	HC210	HC220	HC220	HC230	HC240	HC240
Stratix II	EP2S30 EP2S60 EP2S90*(2)*	EP2S60	EP2S90 EP2S130	EP2S90 EP2S130 EP2S180	EP2S180	EP2S180

Note to Table 8–1:

(1) Table 8–1 does not include the HC210W device. For information on the HC210W device, contact Altera Applications.

(2) This is a Hybrid FineLine BGA package. For more details, refer to the *Package Information for Stratix II Devices* chapter in the *Stratix Device Handbook*, Volume 2.

Beginning with version 5.0 of the Quartus II software, when you compile a design targeting a HardCopy II device, you will need to select a target Stratix II device and a HardCopy II companion device for compilation. Table 8–2 lists the available HardCopy II and Stratix II companion pairs. These pairs are retained in most resource availability tables in this chapter to show the maximum resources available that are supported by either device of the companion pair.

Table 8–2. Stratix II & HardCopy II Companion Devices (Part 1 of 2) *Note (1)*

Package	Companion Pair	
	HardCopy II Device	Stratix II Device
484-pin FineLine BGA	HC210	EP2S30
484-pin FineLine BGA	HC210	EP2S60
484-pin Hybrid FineLine BGA	HC210	EP2S90*(2)*
672-pin FineLine BGA	HC220	EP2S60
780-pin FineLine BGA	HC220	EP2S90
780-pin FineLine BGA	HC220	EP2S130
1,020-pin FineLine BGA	HC230	EP2S90
1,020-pin FineLine BGA	HC230	EP2S130

Table 8–2. Stratix II & HardCopy II Companion Devices (Part 2 of 2) Note (1)		
Package	Companion Pair	
	HardCopy II Device	Stratix II Device
1,020-pin FineLine BGA	HC230	EP2S180
1,020-pin FineLine BGA	HC240	EP2S180
1,508-pin FineLine BGA	HC240	EP2S180

Note to Table 8–2:

(1) Table 8–2 does not include the HC210W device. For information on the HC210W device, contact Altera Applications.

(2) This is a Hybrid FineLine BGA package. For more details, refer to the *Package Information for Stratix II Devices* chapter in the *Stratix Device Handbook*, Volume 2.

When the Quartus II software successfully compiles a design, the HardCopy II Device Resource Guide in the fitter compilation report contains information on migration compatibility to a HardCopy II device. Use this information to select the optimal HardCopy II device for the prototype Stratix II device based on resource requirements and package preference.

For more information on the HardCopy II resource guide in the Quartus II software, refer to the *Prototyping Strategy for HardCopy II Devices* chapter of the *HardCopy Series Device Handbook*.

Table 8–3 shows the available resources for prototyping on a Stratix II device when choosing a HardCopy II device. This chapter examines each resource availability in greater detail.

Table 8–3. Stratix II & HardCopy II Companion Devices Resource Availability Guide (Part 1 of 2) Note (1)									
Stratix II & HardCopy II Companion Devices	Package	Stratix II ALMs (2)	HardCopy II Prototyping Resources						
			ASIC Gates for Logic	User I/O Pins (3)	M4K Blocks	M-RAM Blocks	Total RAM Bits	18 × 18 Multipliers	PLLs
EP2S30 HC210	484-pin FineLine BGA	13,552	360K	334	144	0	663,552	64	4
EP2S60 HC210	484-pin FineLine BGA	24,176	720K	334	190	0	875,520	144	4
EP2S90 HC210	484-pin FineLine BGA	36,384	1 M	308	190	0	875,520	192	4
EP2S60 HC220	672-pin FineLine BGA	24,176	720K	492	255	2	2,354,688	144	4

Table 8–3. Stratix II & HardCopy II Companion Devices Resource Availability Guide (Part 2 of 2) Note (1)

Stratix II & HardCopy II Companion Devices	Package	Stratix II ALMs (2)	HardCopy II Prototyping Resources						
			ASIC Gates for Logic	User I/O Pins (3)	M4K Blocks	M-RAM Blocks	Total RAM Bits	18 × 18 Multipliers	PLLs
EP2S90 HC220	780-pin FineLine BGA	36,384	1 M	494	408	2	3,059,712	192	4
EP2S130 HC220	780-pin FineLine BGA	53,016	1.6 M	494	408	2	3,059,712	252	4
EP2S90 HC230	1,020-pin FineLine BGA	36,384	1 M	698	408	4	4,239,360	192	8
EP2S130 HC230	1,020-pin FineLine BGA	53,016	1.6 M	698	609	6	6,345,216	252	8
EP2S180 HC230	1,020-pin FineLine BGA	71,760	2.2 M	698	614	6	6,368,256	384	8
EP2S180 HC240	1,020-pin FineLine BGA	71,760	2.2 M	742	768 (4)	9	8,847,360	384	12
EP2S180 HC240	1,508-pin FineLine BGA	71,760	2.2 M	951	768 (4)	9	8,847,360	384	12

Notes to Table 8–3:

(1) Table 8–3 does not include the HC210W device. For information on the HC210W device, contact Altera Applications.

(2) ALM: adaptive logic module.

(3) User I/O pin counts are preliminary. The Quartus II software I/O pin counts include one additional pin, PLL_ENA, which is not included in this pin count.

(4) Total number of usable M4K blocks is limited to 768 to allow migration compatibility when prototyping with an EP2S180 device.

I/O Support & Planning

HardCopy II companion devices offer pin-to-pin compatibility with the Stratix II prototype device, which makes them drop-in replacements for the FPGAs. Therefore, you can use HardCopy II devices with the same system board and software developed for prototyping and field trials, enabling the fastest time-to market for high-volume production.

HardCopy II devices offer up to 951 user I/O pins. Table 8–4 lists all available I/O pin counts when assigning a Stratix II device while selecting a HardCopy II companion device. If a Stratix II design uses I/O pins that are not available in both the Stratix II device and the

HardCopy II companion device, the Quartus II software issues a no fit error. Therefore it is important to monitor pin assignments based on the Stratix II device and the HardCopy II companion device.

Table 8–4. Package Options & I/O Pin Counts for Stratix II & HardCopy II Companion Devices Notes (1), (2)

Stratix II Device	HC210	HC220		HC230 (3)	HC240 (4)	
	484-Pin FineLine BGA	672-Pin FineLine BGA	780-Pin FineLine BGA	1,020-Pin FineLine BGA	1,020-Pin FineLine BGA	1,508-Pin FineLine BGA
EP2S30	334					
EP2S60	334	492				
EP2S90	308		494	698		
EP2S130			494	698		
EP2S180				698	742	951

Notes to Table 8–4:

(1) User I/O pin counts are preliminary. The Quartus II software I/O pin counts include one additional pin, `PLL_ENA`, which is not included in this pin count. The `PLL_ENA` pin is not available as a general purpose I/O pin and can only be used to enable the PLLs in this device.

(2) All I/O pin counts include eight dedicated clock input pins (`clk1p`, `clk1n`, `clk3p`, `clk3n`, `clk9p`, `clk9n`, `clk11p`, and `clk11n`) that can be used for data inputs.

(3) The I/O pin counts for all HC230 combinations include four dedicated fast PLL clock inputs (`FPLL7CLKp/n`, `FPLL8CLKp/n`) that can be used for data inputs.

(4) The I/O pin counts for HC240 combinations include eight dedicated fast PLL clock inputs (`FPLL7CLKp/n`, `FPLL8CLKp/n`, `FPLL9CLKp/n`, and `FPLL10CLKp/n`) that can be used for data inputs.

HardCopy II devices offer three distinct types of I/O elements (IOE) which support a variety of I/O features to match Stratix II IOEs. These are memory interface IOEs, high-speed IOEs, and general purpose IOEs.

Memory interface IOEs support popular I/O standards used by external memory devices, including single-ended standards from LVTTL, LVCMOS to SSTL and HSTL voltage referenced (V_{REF}) type I/O standards. Memory interface IOEs also have PCI clamp circuitry for PCI support.

High-speed IOEs support differential applications utilizing LVDS and HyperTransport technology. High-speed IOEs also support single-ended LVTTL and LVCMOS I/O standards, but do not support V_{REF} I/O standards.

General purpose IOEs support LVTTL and LVCMOS I/O standards. General purpose IOEs on the bottom I/O banks (banks 7 and 8) also have PCI clamping circuitry to support the PCI interface on HardCopy II devices.

For more information on HardCopy II IOEs, refer to the *HardCopy II Description, Architecture & Features* chapter of the *HardCopy Series Handbook*.

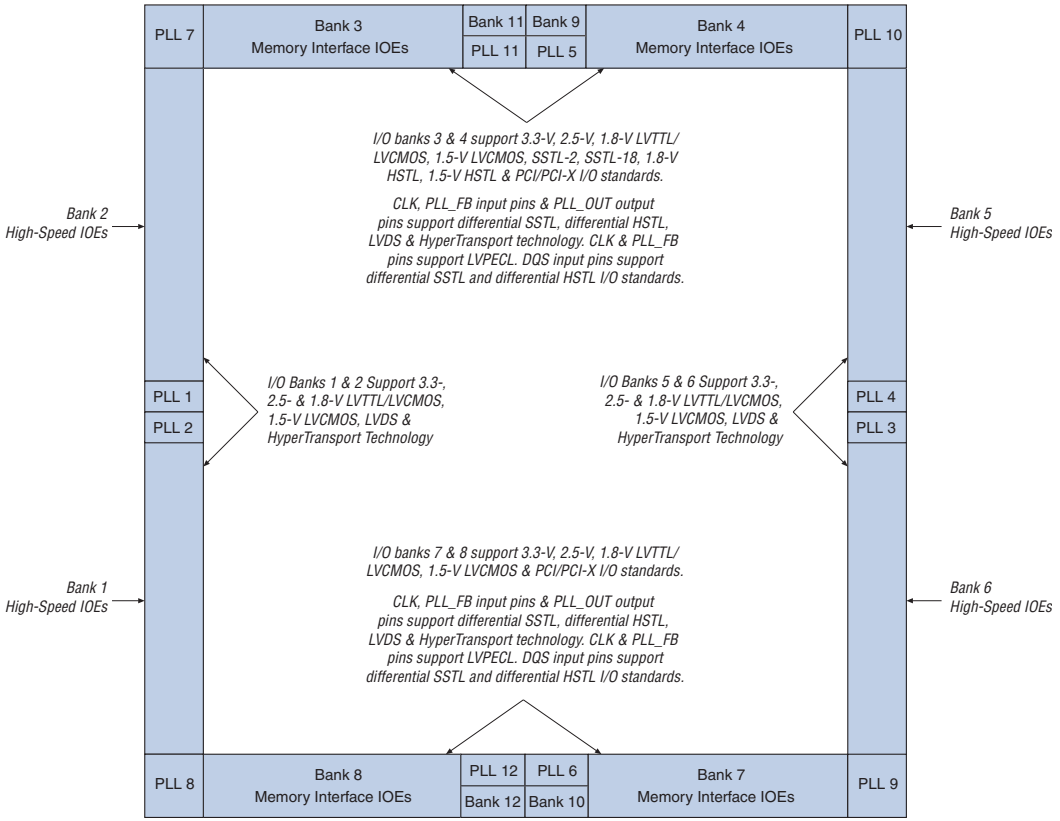
HardCopy II I/O Banks

HardCopy II devices have eight general I/O banks and up to four enhanced PLL external clock output banks (banks 9, 10, 11, 12). HC210 and HC220 devices only have PLL output banks 9 and 10. Figure 8–1 shows the HardCopy II I/O banks and the relative PLL positions.

The left side I/O banks 1 and 2 are high speed IOE banks on all HardCopy II devices. The right side I/O banks 5 and 6 are general purpose IOEs on HC210, HC220, and HC230 devices, but high speed IOEs on HC240 devices.

The top I/O banks 3 and 4 are memory interface IOEs on all HardCopy II devices. The bottom I/O banks 7 and 8 are general purpose IOEs on HC210 and HC220 but memory interface IOEs on HC230 and HC240 devices. The general purpose IOEs on the bottom of the device support PCI clamping, but the general purpose IOEs on the right side do not.

Figure 8–1. HardCopy II HC240 I/O Banks *Notes (1), (2), (3), (4)*



Notes to Figure 8–1:

(1) Figure 8–1 is a top view of the silicon die that corresponds to a reverse view for flip-chip packages. It is a graphical representation only. Refer to the pin list and Quartus II software for exact locations.

(2) Differential HSTL and differential SSTL standards are available for bidirectional operations on DQS pin and input only operations on PLL clock input pins; LVDS, LVPECL, and HyperTransport standards are available for input only operations on PLL clock input pins. Refer to the Differential I/O Termination section for more details.

(3) HardCopy II devices and the Quartus II software does not support differential SSTL and differential HSTL standards at left and right I/O banks. Side I/O banks do not have V_{REF} pins.

(4) Figure 8–1 shows the HC240 device. Other HardCopy II devices have fewer PLL blocks.

User I/O Count Per IOE Type & Bank Location

Table 8–5 lists the maximum I/O count per IOE type. This helps you select a HardCopy II device based on the I/O standard support requirement.

Table 8–5. HardCopy II Maximum User I/O Count Per IOE Type *Notes (1), (2)*

Device	Package	Memory Interface IOEs		General Purpose IOEs		High-Speed IOEs	
		Top	Bottom	Right	Bottom	Left	Right
HC210	484-pin FineLine BGA	87		84	79	84	
HC220	672-pin FineLine BGA	126		124	118	124	
HC220	780-pin FineLine BGA	126		124	120	124	
HC230 *(3)*	1,020-pin FineLine BGA	180	178	152		188	
HC240 *(4)*	1,020-pin FineLine BGA	184	182			188	188
HC240 *(4)*	1,508-pin FineLine BGA	238	233			240	240

Notes to Table 8–5:

(1) User I/O pin counts are preliminary. The Quartus II software I/O pin counts include one additional pin, `PLL_ENA`, which is not included in this pin count. The `PLL_ENA` pin is not available as a general purpose I/O pin and can only be used to enable the PLLs in this device.

(2) All I/O pin counts include eight dedicated clock input pins (`clk1p`, `clk1n`, `clk3p`, `clk3n`, `clk9p`, `clk9n`, `clk11p`, and `clk11n`) that can be used for data inputs.

(3) The I/O pin counts for all HC230 combinations include four dedicated fast PLL clock inputs (`FPLL7CLKp/n`, `FPLL8CLKp/n`) that can be used for data inputs.

(4) The I/O pin counts for HC240 combinations include eight dedicated fast PLL clock inputs (`FPLL7CLKp/n`, `FPLL8CLKp/n`, `FPLL9CLKp/n`, and `FPLL10CLKp/n`) that can be used for data inputs.

HardCopy II Supported I/O Standards

Table 8–6 lists I/O standards that HardCopy II devices supports, separated by IOE type. This list only focuses on user I/O pins.

Table 8–6. Hardcopy II Supported I/O Standards on User I/O Pins (Part 1 of 2)

I/O Standard	Type	V_{CCIO} Level (V)		Memory Interface IOEs	General Purpose IOEs	High-Speed IOEs
		Input	Output			
3.3-V LVTTL/LVCMOS	Single-ended	3.3/2.5	3.3	✓	✓	✓
2.5-V LVTTL/LVCMOS	Single-ended	3.3/2.5	2.5	✓	✓	✓
1.8-V LVTTL/LVCMOS	Single-ended	1.8/1.5	1.8	✓	✓	✓
1.5-V LVCMOS	Single-ended	1.8/1.5	1.5	✓	✓	✓

Table 8–6. Hardcopy II Supported I/O Standards on User I/O Pins (Part 2 of 2)

I/O Standard	Type	V_{CCIO} Level (V)		Memory Interface IOEs	General Purpose IOEs	High-Speed IOEs
		Input	Output			
SSTL-2 class I and II	Voltage referenced	2.5	2.5	✓		
SSTL-18 class I and II	Voltage referenced	1.8	1.8	✓		
1.8-V HSTL class I and II	Voltage referenced	1.8	1.8	✓		
1.5-V HSTL class I and II	Voltage referenced	1.5	1.5	✓		
PCI / PCI-X	Single-ended	3.3	3.3	✓	*(1)*	
Differential SSTL-2 class I and II input	Pseudo differential *(3)*	3.3/2.5/ 1.8/1.5		*(2)*		
Differential SSTL-2 class I and II output	Pseudo differential *(3)*		2.5	*(2)*		
Differential SSTL-18 class I and II input	Pseudo differential *(3)*	3.3/2.5/ 1.8/1.5		*(2)*		
Differential SSTL-18 class I and II output	Pseudo differential *(3)*		1.8	*(2)*		
1.8-V differential HSTL class I and II input	Pseudo differential *(3)*	3.3/2.5/ 1.8/1.5			1.8/1.5	*(2)*
1.8-V differential HSTL class I and II output	Pseudo differential *(3)*		1.8	*(2)*		
1.5-V differential HSTL class I and II input	Pseudo differential *(3)*	3.3/2.5/ 1.8/1.5		*(2)*		
1.5-V differential HSTL class I and II output	Pseudo differential *(3)*		1.5	*(2)*		
LVDS	Differential	2.5	2.5			✓
HyperTransport™ technology	Differential	2.5	2.5			✓

Notes to Table 8–6:

(1) Like Stratix II devices, the optional PCI clamp is only available on column I/O pins. General purpose IOEs on the right row I/O pins do not support the PCI clamp.

(2) Similar to Stratix II devices, these I/O standards are only available on input clock pins, output clock pins in I/O banks 9,10,11, 12, and DQS pins in top I/O banks 3, 4 for all HardCopy II devices, and DQS pins in bottom I/O banks 7, 8 for HC230 and HC240 devices.

(3) Pseudo-differential HSTL and SSTL inputs only use the positive polarity input in the speed path. The negative input is not connected internally. Pseudo-differential HSTL and SSTL outputs use two single-ended outputs with the second output programmed as inverted. This is similar to a Stratix II device implementation.

Table 8–7 lists the I/O standards that HardCopy II devices support. Table 8–7 is organized by clock input, clock output, and PLL feedback pins.

Table 8–7. Hardcopy II Supported I/O Standards of Input Clocks, Clock Out & PLL Feedback (Part 1 of 2)

I/O Standard	Type	V_{CCIO} Level (V)		CLK[0..3, 8..11] *(1)*	CLK[4..7, 12..15] *(2)*	FPLL_CLK *(3)*	PLL_OUT *(4)*	PLL_FB *(5)*
		Input	Output					
3.3-V LVTTL / LVCMOS	Single-ended	3.3/2.5	3.3	✓	✓	✓	✓	✓
2.5-V LVTTL / LVCMOS	Single-ended	3.3/2.5	2.5	✓	✓	✓	✓	✓
1.8-V LVTTL / LVCMOS	Single-ended	1.8/1.5	1.8	✓	✓	✓	✓	✓
1.5-V LVCMOS	Single-ended	1.8/1.5	1.5	✓	✓	✓	✓	✓
SSTL-2 class I	Voltage referenced	2.5	2.5		✓		✓	✓
SSTL-2 class II	Voltage referenced	2.5	2.5		✓		✓	✓
SSTL-18 class I	Voltage referenced	1.8	1.8		✓		✓	✓
SSTL-18 class II	Voltage referenced	1.8	1.8		✓		✓	✓
1.8-V HSTL class I	Voltage referenced	1.8	1.8		✓		✓	✓
1.8-V HSTL class II	Voltage referenced	1.8	1.8		✓		✓	✓
1.5-V HSTL class I	Voltage referenced	1.5	1.5		✓		✓	✓
1.5-V HSTL class II	Voltage referenced	1.5	1.5		✓		✓	✓
PCI / PCI-X	Single-ended	3.3	3.3		✓		✓	✓
Differential SSTL-2 class I and II input	Pseudo differential *(6)*	3.3/2.5/ 1.8/1.5			✓			✓
Differential SSTL-2 class I and II output	Pseudo differential *(6)*		2.5				✓	✓
Differential SSTL-18 class I and II input	Pseudo differential *(6)*	3.3/2.5/ 1.8/1.5			✓			✓

Table 8–7. Hardcopy II Supported I/O Standards of Input Clocks, Clock Out & PLL Feedback (Part 2 of 2)

I/O Standard	Type	V_{CCIO} Level (V)		CLK[0..3, 8..11] *(1)*	CLK[4..7, 12..15] *(2)*	FPLL_CLK *(3)*	PLL_OUT *(4)*	PLL_FB *(5)*
		Input	Output					
Differential SSTL-18 class I and II output	Pseudo differential *(6)*		1.8				✓	✓
1.8-V differential HSTL class I and II input	Pseudo differential *(6)*	3.3/2.5/ 1.8/1.5			✓			✓
1.8-V differential HSTL class I and II output	Pseudo differential *(6)*		1.8				✓	✓
1.5-V differential HSTL class I and II input	Pseudo differential *(6)*	3.3/2.5/ 1.8/1.5			✓			✓
1.5-V differential HSTL class I and II output	Pseudo differential *(6)*		1.5				✓	✓
LVDS input	Differential	2.5		✓	✓	✓		✓
LVDS output	Differential		2.5				✓	✓
HyperTransport technology input	Differential	2.5		✓	✓	✓		✓
HyperTransport technology output	Differential		2.5V				✓	✓
LVPECL input	Differential	3.3/2.5/ 1.8/1.5	*(7)*		✓		✓	✓

Notes to Table 8–7:

(1) `CLK8` and `CLK10` pins on HC210, HC220 and HC230 devices do not support differential standards LVDS and HyperTransport technology. Only LVTTL is supported on these CLK pins for these devices.

(2) `CLK[4..7]` pins on HC210 and HC220 devices do not support SSTL, HSTL and differential SSTL, HSTL input or output.

(3) HC230 only has two fast PLL clocks, `FPLL[7..8]CLK`. HC240 has four FPLL clocks, `FPLL[7..10]CLK`.

(4) HC210 and HC220 `PLL6_OUT` pins do not support SSTL, HSTL and differential SSTL, HSTL input or output

(5) HC210 and HC220 `PLL6_FB` pins do not support SSTL, HSTL and differential SSTL, HSTL input or output

(6) Pseudo-differential HSTL and SSTL inputs only use the positive polarity input in the speed path. The negative input is not connected internally. Pseudo-differential HSTL and SSTL outputs use two single-ended outputs with the second output programmed as inverted. This is similar to a Stratix II device implementation.

(7) This is not supported.

External Memory Interface Support

Like Stratix II devices, HardCopy II I/O pins have dedicated phase-shift circuitry for interfacing with external memory, including DDR and DDR2 SDRAM, QDR II SRAM, RLDRAM II, and SDR SDRAM. A compensated delay element on each DQS pin automatically aligns input DQS synchronization signals with the data window of their corresponding DQ data signals.

For all HardCopy I devices, the top I/O banks (3 and 4) support DQ and DQS signals with DQ bus modes that vary from ×4, ×8/×9, ×16/×18 and up to ×32/×36. The top bank has a phase-shifting reference circuit that controls the compensated delay elements for all DQS pins on the top bank.

For the HC230 and HC240 HardCopy II devices, the bottom I/O banks (7 and 8) also support DQ and DQS signals with DQ bus modes from ×4, ×8/×9, ×16/×18 and ×32/×36. Similar to the top banks, the bottom I/O banks of these devices also have a phase-shifting reference circuit to control the delay elements at the bottom DQS pins.

Table 8–8 shows the number of DQ and DQS buses supported per companion device pair. *(3)*

Table 8–8. DQ & DQS Bus Mode support for Stratix II & HardCopy II Companion Devices (Part 1 of 2) *Note (1)*

Stratix II & HardCopy II Companion Devices	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2S30 HC210 *(2)*	484-pin FineLine BGA	4	2		
EP2S60 HC210 *(2)*	484-pin FineLine BGA	4	2		
EP2S90 HC210 *(2)*	484-pin FineLine BGA	4	2		
EP2S60 HC220 *(2)*	672-pin FineLine BGA	9	4	2	
EP2S90 HC220 *(2)*	780-pin FineLine BGA	9	4	2	
EP2S130 HC220 *(2)*	780-pin FineLine BGA	9	4	2	
EP2S90 HC230 *(3)*	1,020-pin FineLine BGA	36	18	8	4
EP2S130 HC230 *(3)*	1,020-pin FineLine BGA	36	18	8	4

Table 8–8. DQ & DQS Bus Mode support for Stratix II & HardCopy II Companion Devices (Part 2 of 2) *Note (1)*

Stratix II & HardCopy II Companion Devices	Package	Number of ×4 Groups	Number of ×8/×9 Groups	Number of ×16/×18 Groups	Number of ×32/×36 Groups
EP2S180 HC230 *(3)*	1,020-pin FineLine BGA	36	18	8	4
EP2S180 HC240	1,020-pin FineLine BGA	36	18	8	4
EP2S180 HC240	1,508-pin FineLine BGA	36	18	8	4

Notes to Table 8–8:

(1) The DQ and DQS numbers are preliminary.

(2) HardCopy II devices HC210 and HC220 support memory interface in the top I/O banks only. Unlike their Stratix II companions, these devices cannot support DIMMs.

(3) Similar to their Stratix II companions, these device and package combinations can support two 64- or 72-bit DIMMs in ×4 and ×8/×9 modes.

LVDS, SERDES & DPA Compatibility

HardCopy II devices offer up to 116 transmitter and receiver pairs. Similar to Stratix II devices, these differential I/O pins are located on row I/O pins. The HC240 device's left and right banks are high-speed IOEs which support differential transmission. The HC210, HC220, and HC230 devices only support differential transmission on the left banks. The LVDS and HyperTransport technology interface functionality, including the SERDES and DPA, is the same as Stratix II devices.

Table 8–9 shows the maximum differential channel supported by each HardCopy II and Stratix II companion pair.

Table 8–9. Differential Channels with Stratix II & HardCopy II Companion Devices (Part 1 of 2) *Note (1)*

Stratix II & HardCopy II Companion Devices	Package	Transmitters	Receivers
EP2S30 HC210 *(2)*	484-pin FineLine BGA	19	21
EP2S60 HC210 *(2)*	484-pin FineLine BGA	19	21
EP2S90 HC210 *(2)*	484-pin FineLine BGA	19	21

Table 8–9. Differential Channels with Stratix II & HardCopy II Companion Devices (Part 2 of 2) *Note (1)*

Stratix II & HardCopy II Companion Devices	Package	Transmitters	Receivers
EP2S60 HC220 *(2)*	672-pin FineLine BGA	29	31
EP2S90 HC220 *(2)*	780-pin FineLine BGA	29	31
EP2S130 HC220 *(2)*	780-pin FineLine BGA	29	31
EP2S90 HC230 *(2)*	1,020-pin FineLine BGA	44	46
EP2S130 HC230 *(2)*	1,020-pin FineLine BGA	44	46
EP2S180 HC230 *(2)*	1,020-pin FineLine BGA	44	46
EP2S180 HC240 *(3)*	1,020-pin FineLine BGA	88	92
EP2S180 HC240 *(3)*	1,508-pin FineLine BGA	116	116

Notes to Table 8–9:

(1) Pin count does not include dedicated PLL input and output pins.

(2) The total number of receiver channels for HC210, HC220 and HC230 devices include two non-dedicated clock channels that can optionally be used as data channels.

(3) The total number of receiver channels for HC240 devices include four non-dedicated clock channels that can optionally be used as data channels.

Programmable Drive Strength Support

The maximum current strength setting is the default setting in the Quartus II software and achieves maximum I/O performance. Stratix II device output buffers for each I/O pin have a programmable drive strength control for certain I/O standards.

HardCopy II support for these settings differs from that found in Stratix II devices. For compatibility with HardCopy II HC210 and HC220 devices, you must restrict the I/O drive settings of Stratix II companion devices as shown in Table 8–10.

Table 8–10. HC210 & HC220 Device Programmable Drive Strengths

I/O Standard	I_{OH} & I_{OL} Current Strength Setting (mA) for Top Column I/O Pins	I_{OH} & I_{OL} Current Strength Setting (mA) for Bottom Column I/O Pins	I_{OH} & I_{OL} Current Strength Setting (mA) for Left Row I/O Pins	I_{OH} & I_{OL} Current Strength Setting (mA) for Right Row I/O Pins
3.3-V LVTTL	24, 20, 12, 8, 4 *(1)*	12, 8, 4 *(1)*	12, 8, 4	12, 8, 4
3.3-V LVCMOS	24, 20, 12, 8, 4 *(1)*	8, 4 *(1)*	8, 4	8, 4
2.5-V LVTTL/LVCMOS	16, 12, 8, 4	12, 8, 4 *(1)*	12, 8, 4	12, 8, 4
1.8-V LVTTL/LVCMOS	12, 10, 8, 6, 4, 2	8, 6, 4, 2 *(1)*	8, 6, 4, 2	8, 6, 4, 2
1.5-V LVCMOS	8, 6, 4, 2	4, 2 *(1)*	4, 2	4, 2
SSTL-2 class I	12, 8	*(2)*	*(3)*	*(3)*
SSTL-2 class II	24, 20, 16	*(2)*	*(3)*	*(3)*
SSTL-18 class I	12, 10, 8, 6, 4	*(2)*	*(3)*	*(3)*
SSTL-18 class II	20, 18, 16, 8	*(2)*	-	-
HSTL-18 class I	12, 10, 8, 6, 4	*(2)*	-	-
HSTL-18 class II	20, 18, 16	*(2)*	-	-
HSTL-15 class I	12, 10, 8, 6, 4	*(2)*	-	-
HSTL-15 class II	20, 18, 16	*(2)*	-	-

Notes to Table 8–10:

(1) HardCopy II devices do not support some of the settings available in the Stratix II prototype device. For more information, refer to the *Stratix II Device Family Data Sheet* in Volume 1 of the *Stratix II Device Handbook*.

(2) HC220 and HC210 devices do not support memory interface standards on bottom I/O pins.

(3) Row I/O pins do not support SSTL I/O standards.

Similarly, when using HardCopy II HC230 and HC240 devices as companion devices, you must restrict the I/O drive settings as shown in Table 8–11.

Table 8–11. HC230 & HC240 Device Programmable Drive Strengths

I/O Standard	I_{OH} & I_{OL} Current Strength Setting (mA) for Column I/O Pins	I_{OH} & I_{OL} Current Strength Setting (mA) for Row I/O Pins
3.3-V LVTTL	24, 20, 16, 12, 8, 4 *(1)*	12, 8, 4
3.3-V LVCMOS	24, 20, 16, 12, 8, 4 *(1)*	8, 4
2.5-V LVTTL/LVCMOS	16, 12, 8, 4	12, 8, 4
1.8-V LVTTL/LVCMOS	12, 10, 8, 6, 4, 2	8, 6, 4, 2
1.5-V LVCMOS	8, 6, 4, 2	4, 2
SSTL-2 class I	12, 8	*(2)*
SSTL-2 class II	24, 20, 16	*(2)*
SSTL-18 class I	12, 10, 8, 6, 4	*(2)*
SSTL-18 class II	20, 18, 16, 8	-
HSTL-18 class I	12, 10, 8, 6, 4	-
HSTL-18 class II	20, 18, 16	-
HSTL-15 class I	12, 10, 8, 6, 4	-
HSTL-15 class II	20, 18, 16	-

Notes to Table 8–11:

(1) HardCopy II devices do not support some of the settings available in the Stratix II prototype device. For more information, refer to the *Stratix II Device Family Data Sheet* in volume 1 of the *Stratix II Device Handbook*.

(2) Row I/O pins do not support SSTL I/O standards.

On-Chip Termination

Like Stratix II devices, HardCopy II devices feature on-chip termination to provide I/O impedance matching and termination capabilities. To maintain compatibility with Stratix II prototype devices, HardCopy II devices support on-chip series termination (RS) for single-ended I/O standards and on-chip differential termination (RD) for differential I/O standards. However, some HardCopy II pins do not support the on-chip termination that may be available on the same Stratix II pin. This section highlights the termination schemes that HardCopy II devices support.

On-Chip Series Termination

Stratix II and HardCopy II devices support I/O driver on-chip series termination (RS) through drive-strength control for single-ended I/O standards. There are two ways to implement the RS in Stratix II and Hardcopy II devices:

- RS without calibration for both row and column I/O pins
- RS with calibration only for column I/O pins

On-Chip Series Termination without Calibration

HardCopy II devices support output-driver impedance matching to closely match the impedance of the transmission line. If you select matching impedance, you cannot select programmable-current drive strength. Table 8–12 lists the HardCopy II HC230 and HC240 output standards that support on-chip series termination without calibration.

Table 8–12. HC230 & HC240 Selectable I/O Drivers with On-Chip Series Termination without Calibration *Note (1)*

I/O Standard	Column I/O Pins	Row I/O Pins
3.3-V LVTTL	25 or 50 Ω	25 or 50 Ω
3.3-V LVCMOS	25 or 50 Ω	25 or 50 Ω
2.5-V LVTTL	25 or 50 Ω	25 or 50 Ω
2.5-V LVCMOS	25 or 50 Ω	25 or 50 Ω
1.8-V LVTTL	25 or 50 Ω	50 Ω
1.8-V LVCMOS	25 or 50 Ω	50 Ω
1.5-V LVTTL	50 Ω	
1.5-V LVCMOS	50 Ω	
2.5-V SSTL class I	50 Ω	*(2)*
2.5-V SSTL class II	25 Ω	*(2)*
1.8-V SSTL class I	50 Ω	*(2)*
1.8-V SSTL class II	25 Ω	
1.8-V HSTL class I	50 Ω	*(2)*
1.8-V HSTL class II	25 Ω	
1.5-V HSTL class I	*(3)*	

Notes to Table 8–12:

(1) These numbers are preliminary and pending silicon characterization.
(2) HardCopy II HC230 and HC240 devices do not support on-chip series termination with this I/O standard on these pins.
(3) Support pending HardCopy II characterization.

Table 8–13 lists the HardCopy II HC210 and HC220 output standards that support on-chip series termination without calibration.

Table 8–13. HC210 & HC220 Selectable I/O Drivers with On-Chip Series Termination without Calibration Note (1)

I/O Standard	Top Column I/O Pins	Bottom Column I/O Pins	Left Row I/O Pins	Right Row I/O Pins
3.3-V LVTTL	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω
3.3-V LVCMOS	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω
2.5-V LVTTL	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω
2.5-V LVCMOS	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω	25 or 50 Ω
1.8-V LVTTL	25 or 50 Ω	50 Ω	50 Ω	50 Ω
1.8-V LVCMOS	25 or 50 Ω	50 Ω	50 Ω	50 Ω
1.5-V LVTTL	(3)	(2)		
1.5-V LVCMOS	(3)	(2)		
2.5-V SSTL class I	50 Ω	(2)	(2)	(2)
2.5-V SSTL class II	25 Ω	(2)	(2)	(2)
1.8-V SSTL class I	50 Ω	(2)	(2)	(2)
1.8-V SSTL class II	25 Ω	(2)		
1.8-V HSTL class I	50 Ω	(2)	(2)	(2)
1.8-V HSTL class II	25 Ω	(2)		
1.5-V HSTL class I	(3)	(2)		

Notes to Table 8–13:

(1) All these numbers are preliminary and pending silicon characterization.

(2) HardCopy II HC210 and HC220 devices do not support on-chip series termination with this I/O standard on these pins.

(3) Support pending HardCopy II characterization.

On-Chip Series Termination with Calibration

Stratix II devices support on-chip series termination with calibration in column I/O pins in top and bottom banks. HC230 and HC240 devices also support on-chip series termination with calibration in column I/O pins in top and bottom banks, but HC220 and HC210 devices only

support this feature on the top I/O banks. Table 8–14 lists available I/O standards on the HardCopy II devices that support calibrated-series termination.

Table 8–14. HardCopy II Selectable I/O Drivers with On-Chip Series Termination with Calibration *Note (1)*

I/O Standard	HC230, HC240 Column I/O Pins	HC210, HC220 Top Column I/O Pins *(2)*
3.3-V LVTTL	25 or 50 Ω	25 or 50 Ω
3.3-V LVCMOS	25 or 50 Ω	25 or 50 Ω
2.5-V LVTTL	25 or 50 Ω	25 or 50 Ω
2.5-V LVCMOS	25 or 50 Ω	25 or 50 Ω
1.8-V LVTTL	25 or 50 Ω	25 or 50 Ω
1.8-V LVCMOS	25 or 50 Ω	25 or 50 Ω
1.5-V LVTTL	*(3)*	50 Ω
1.5-V LVCMOS	*(3)*	50 Ω
2.5-V SSTL class I	50 Ω	50 Ω
2.5-V SSTL class II	25 Ω	50 Ω
1.8-V SSTL class I	50 Ω	50 Ω
1.8-V SSTL class II	25 Ω	25 Ω
1.8-V HSTL class I	50 Ω	50 Ω
1.8-V HSTL class II	25 Ω	25 Ω
1.5-V HSTL class I	*(3)*	50 Ω

Notes to Table 8–14:

(1) These numbers are preliminary and pending silicon characterization

(2) HardCopy II HC210 and HC220 devices do not support on-chip series termination with calibration on bottom I/O pins.

(3) Support pending HardCopy II characterization.

Differential I/O Termination

Similar to the FPGA, HardCopy II devices provide an on-chip 100-Ω differential termination option on each differential receiver channel for LVDS and HyperTransport technology standards. When using an HC240 device as a companion device, differential termination is supported on all row I/O pins that support LVDS and HyperTransport technology standards.

When using HC230, HC220, and HC210 devices, only the left row I/O pins support differential termination. The right row I/O pins do not support LVDS and HyperTransport technology standards.

Table 8–15 shows the differential termination support.

Table 8–15. HardCopy II I/O Banks Supporting 100-Ω Differential Termination *Notes (1), (2)*

I/O Standard	HC240 Left & Right Banks (1, 2, 5 & 6)	HC240 Top & Bottom Banks (3, 4, 7 through 12)	HC230, HC210, HC220 Left Banks (1 & 2)	HC230, HC210, HC220 Other Banks (3 to 12)
LVDS	✓		✓	
HyperTransport technology	✓		✓	
Clock Inputs *(3)*	✓		✓	

Notes to Table 8–15:
(1) HC230, HC220 and HC210 device left clock pins `CLK0` and `CLK2` support differential on-chip termination.
(2) All other clock pins, including `FPLL[7..10]CLK` do not support differential on-chip termination.
(3) HardCopy II HC240 device clock pins `CLK0`, `CLK2`, `CLK8`, and `CLK10` support differential on-chip termination, similar to Stratix II devices.

Stratix II & HardCopy II Companion Memory Blocks

HardCopy II device RAM bit offerings range from 663 kbits to 8.8 Mbits. HardCopy II memory blocks are functionally equivalent to the Stratix II memory blocks. HardCopy II memory blocks can implement various Stratix II device memory configurations, including simple and true dual port modes, FIFO, parity bits, ROM modes, and all other features as listed in the *HardCopy II Description, Architecture & Features* chapter of the *HardCopy Series Handbook*. One difference between HardCopy II and Stratix II devices is that HardCopy II devices do not support M512 blocks. Additionally, you cannot pre-load HardCopy II M4K blocks with a Memory Initialization File (**.mif**) when used as RAM.

Table 8–16 shows all the memory block offerings when compiling for a Stratix II FPGA in conjunction with a HardCopy II companion device. Use Table 8–16 as a guide when optimizing memory requirements for selected Stratix II and HardCopy II pairs.

Table 8–16. Total RAM Blocks for Stratix II & HardCopy II Companion Devices (Part 1 of 2)

Stratix II & HardCopy II Companion Devices	Package	M4K Blocks	M-RAM Blocks	Total RAM Bits
EP2S30 HC210	484-pin FineLine BGA	144	0	663,552
EP2S60 HC210	484-pin FineLine BGA	190	0	875,520

Table 8–16. Total RAM Blocks for Stratix II & HardCopy II Companion Devices (Part 2 of 2)

Stratix II & HardCopy II Companion Devices	Package	M4K Blocks	M-RAM Blocks	Total RAM Bits
EP2S90 HC210	484-pin FineLine BGA	190	0	875,520
EP2S60 HC220	672-pin FineLine BGA	255	2	2,354,688
EP2S90 HC220	780-pin FineLine BGA	408	2	3,059,712
EP2S130 HC220	780-pin FineLine BGA	408	2	3,059,712
EP2S90 HC230	1,020-pin FineLine BGA	408	4	4,239,360
EP2S130 HC230	1,020-pin FineLine BGA	609	6	6,345,216
EP2S180 HC230	1,020-pin FineLine BGA	614	6	6,368,256
EP2S180 HC240	1,020-pin FineLine BGA	768*(1)*	9	8,847,360
EP2S180 HC240	1,508-pin FineLine BGA	768*(1)*	9	8,847,360

Notes to Table 8–16

(1) Total number of usable M4K blocks is limited to 768 to allow migration compatibility when prototyping with an EP2S180 device.

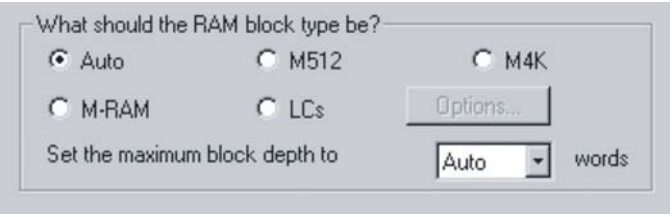
Table 8–16 does not list M512 blocks because they are not supported in HardCopy II devices. Also, the HC210 devices do not offer M-RAM blocks. Some compatibility guidelines are discussed in the next sections.

M512 Options

HardCopy II devices do not support M512 blocks. When compiling Stratix II designs with Hardcopy II companions devices in the Quartus II software, you must check the **Limit DSP and RAM to HardCopy II Resources** box in the Device Settings dialog box (Assignments Menu). This automatically places all memory blocks in the available HardCopy II resources. If you do not check this box, the Quartus II software may use memory resources not available in the HardCopy II device but available in the Stratix II device, such as M512 blocks. However, migration into HardCopy II devices is not allowed and this is indicated in the Quartus II fitter report.

Your HardCopy II design can use M4K memory blocks to implement memory designs instead of M512 blocks. Quartus II megafunctions offer various memory implementations that use M4K blocks. When using the Quartus II MegaWizard® Plug-In Manager to configure the megafunction, Altera recommends selecting the Auto option to allow the Quartus II software to determine how the design is implemented in memory blocks (Figure 8–2). This allows the Quartus II software to optimize memory selection based on memory size and placement requirements into the available memory blocks of the selected HardCopy II and Stratix II companion pair.

Figure 8–2. Quartus II MegaFunction RAM selection



You can select logic cells in the megafunction to implement small-memory blocks in your design. This implements the memory design in Stratix II ALMs or HardCopy II HCells. However, there may be power and performance trade-offs when choosing between an M4K or M-RAM block or using the ALMs (or HCells). HardCopy II devices power down unused M4K blocks, M-RAM blocks and HCells.

Implementing memory blocks using logic cells, as seen in Figure 8–2, allows you to select a memory implementation functionally equivalent to M512 blocks or a non-equivalent option to save resources. Altera recommends setting the option to a functionally equivalent version with the M512 blocks.

For very small memory implementations such as a 8 × 16 single port RAM, the M4K or M-RAM blocks will be under-utilized, and may be less power efficient than a small number of HCells. If you select the logic cell option, only a fraction of ALMs are required in the Stratix II device, which translates into a small number HCells used in the HardCopy II device. However, when performance is a key factor, or your design requires ALMs to implement other logic, it may be more efficient to use M4K blocks. Altera recommends using the Quartus II software to analyze performance trade-offs between the given options.

M4K Utilization

HardCopy II M4K block functionality is similar to Stratix II M4K blocks. You cannot pre-load HardCopy II M4K blocks with a Memory Initialization File (**.mif**) when used as RAM. Also, unlike Stratix II devices, the HardCopy II M4K RAM contents and their output registers are unknown after power up. However, if the HardCopy II M4K block is designated as ROM, then it powers up with the ROM contents. When designing M4K blocks as RAM, Altera recommends writing to the block before reading from it to avoid reading unknown initial power-up data conditions. One advantage over Stratix II RAM blocks is unused M4K blocks are disconnected from the power rails, optimizing overall power consumption.

M-RAM Compatibility

HardCopy II M-RAM blocks share the same functionality as Stratix II M-RAM blocks. One key feature with HardCopy II M-RAM blocks is power optimization when the M-RAM block is not used. Unused M-RAM blocks are disconnected from the power rails, optimizing overall power consumption.

Some Stratix II devices (engineering sample devices and Revision A production devices) have M-RAM functionality that differs slightly from current Stratix II production devices. HardCopy II M-RAM functionality only matches that of current Stratix II devices. Hence, in order to maintain proper compatibility, compiling only for current production Stratix II devices is supported. More information on the Stratix II M-RAM errata can be found in the *Stratix II FPGA Family Errata Sheet* available on the Altera web site (**www.altera.com**).

Table 8–17 lists the M4K and M-RAM block supported features. This information can also be found in the *HardCopy II Description, Architecture & Features* chapter of the *HardCopy Series Handbook*.

Table 8–17. HardCopy II Embedded Memory Features (Part 1 of 2)

Feature	M4K Blocks	M-RAM Blocks
Total RAM bits (including parity bits)	4,608	589,824
Configurations	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 8K × 64 8K × 72 4K × 128 4K × 144
Parity bits	✓	✓
Byte enable	✓	✓
Pack mode	✓	✓
Address clock enable	✓	✓
Single-port memory	✓	✓
Simple dual-port memory	✓	✓
True dual-port memory	✓	✓
Embedded shift register	✓	
ROM	✓	
FIFO buffer	✓	✓
Simple dual-port mixed width support	✓	✓
True dual-port mixed width support	✓	✓
Memory initialization file (**.mif**)	*(1)*	
Mixed-clock mode	✓	✓
Power-up condition	Outputs unknown	Outputs unknown
Register clears	Output registers only	Output registers only
Same-port read-during-write	New data available at positive clock edge	New data available at positive clock edge
Mixed-port read-during-write	Outputs set to unknown or old data	Unknown output

Table 8–17. HardCopy II Embedded Memory Features (Part 2 of 2)		
Feature	**M4K Blocks**	**M-RAM Blocks**
Power down of unused RAM blocks *(2)*	✓	✓

Notes to Table 8–17:
(1) Stratix II M4K blocks support **.mif** file loading.
(2) Stratix II memory blocks remain powered up even when not used.

PLL Planning & Utilization

Stratix II devices support enhanced PLLs and fast PLLs. HardCopy II devices also support enhanced PLLs and fast PLLs, but with two variations:

- HardCopy II devices have a different number of PLLs than Stratix II devices.
- HardCopy II devices may support fewer I/O standards for clock inputs and outputs. This is explained in the I/O standards support section later in this chapter.

Table 8–18 shows which PLLs each HardCopy II and Stratix II device supports. The Stratix II reference columns are divided based on package, and not density. Figures 8–3 to 8–5 show PLL number designations. The Stratix II devices support 6 or 12 PLLs depending on the package offering, and not the device density. The HardCopy II PLLs are not removed symmetrically from all four sides. In general, fast PLLs are removed from sides that do not support high speed IOEs since the primary use of the fast PLL on the sides is for high speed I/O interface functions

Table 8–18. Stratix II / HardCopy II Companion Device PLL Availability Guide													
Stratix II & HardCopy II Companion Devices	**Package**	**Fast PLLs**								**Enhanced PLLs**			
		1	**2**	**3**	**4**	**7**	**8**	**9**	**10**	**5**	**6**	**11**	**12**
EP2S30 HC210 *(1)*	484-pin FineLine BGA	✓	✓							✓	✓		
EP2S60 HC210 *(1)*	484-pin FineLine BGA	✓	✓							✓	✓		
EP2S90 HC210 *(1)*	484-pin FineLine BGA	✓	✓							✓	✓		
EP2S60 HC220 *(1)*	672-pin FineLine BGA	✓	✓							✓	✓		
EP2S90 HC220 *(1)*	780-pin FineLine BGA	✓	✓							✓	✓		

Table 8–18. Stratix II / HardCopy II Companion Device PLL Availability Guide

Stratix II & HardCopy II Companion Devices	Package	Fast PLLs								Enhanced PLLs			
		1	2	3	4	7	8	9	10	5	6	11	12
EP2S130 HC220 *(1)*	780-pin FineLine BGA	✓	✓							✓	✓		
EP2S90 HC230 *(2)*	1,020-pin FineLine BGA	✓	✓			✓	✓			✓	✓	✓	✓
EP2S130 HC230 *(2)*	1,020-pin FineLine BGA	✓	✓			✓	✓			✓	✓	✓	✓
EP2S180 HC230 *(2)*	1,020-pin FineLine BGA	✓	✓			✓	✓			✓	✓	✓	✓
EP2S180 HC240	1,020-pin FineLine BGA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP2S180 HC240	1,508-pin FineLine BGA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Notes to Table 8–18:

(1) HC210 and HC220 devices do not support fast PLLs 3, 4, 9, and 10, unlike Stratix II devices.

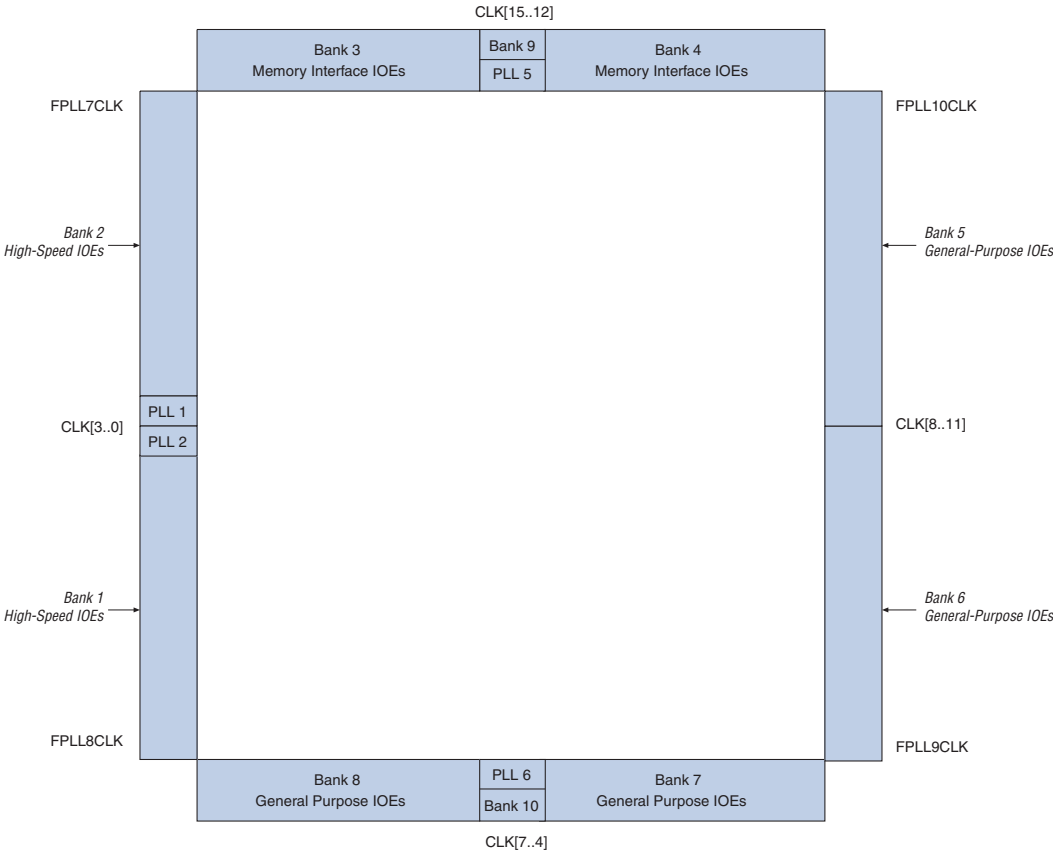
(2) HC230 devices do not support fast PLLs 3 and 4, unlike Stratix II devices.

HardCopy II PLLs are functionally identical to the Stratix II PLLs. The HardCopy II enhanced and fast PLLs support reconfiguration and are also reconfigurable for bandwidth and phase shift.

Figures 8–3 to 8–5 show the PLL locations for each HardCopy II device.

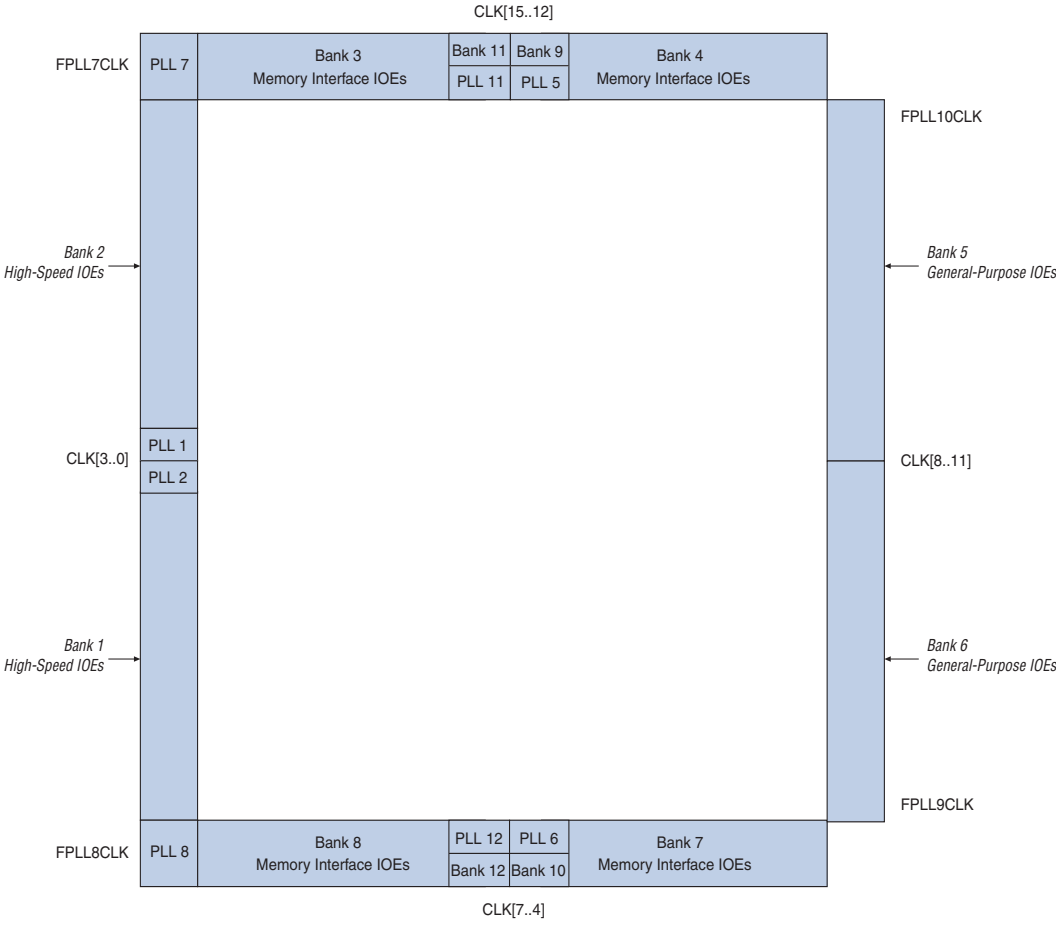
For HC210 and HC220 devices, fast PLLs 1 and 2 are located in the logic array of the device and enhanced PLLs 5 and 6 are located in the periphery next to the device's top and bottom I/O banks.

Figure 8–3. HC210 & HC220 PLL Locations



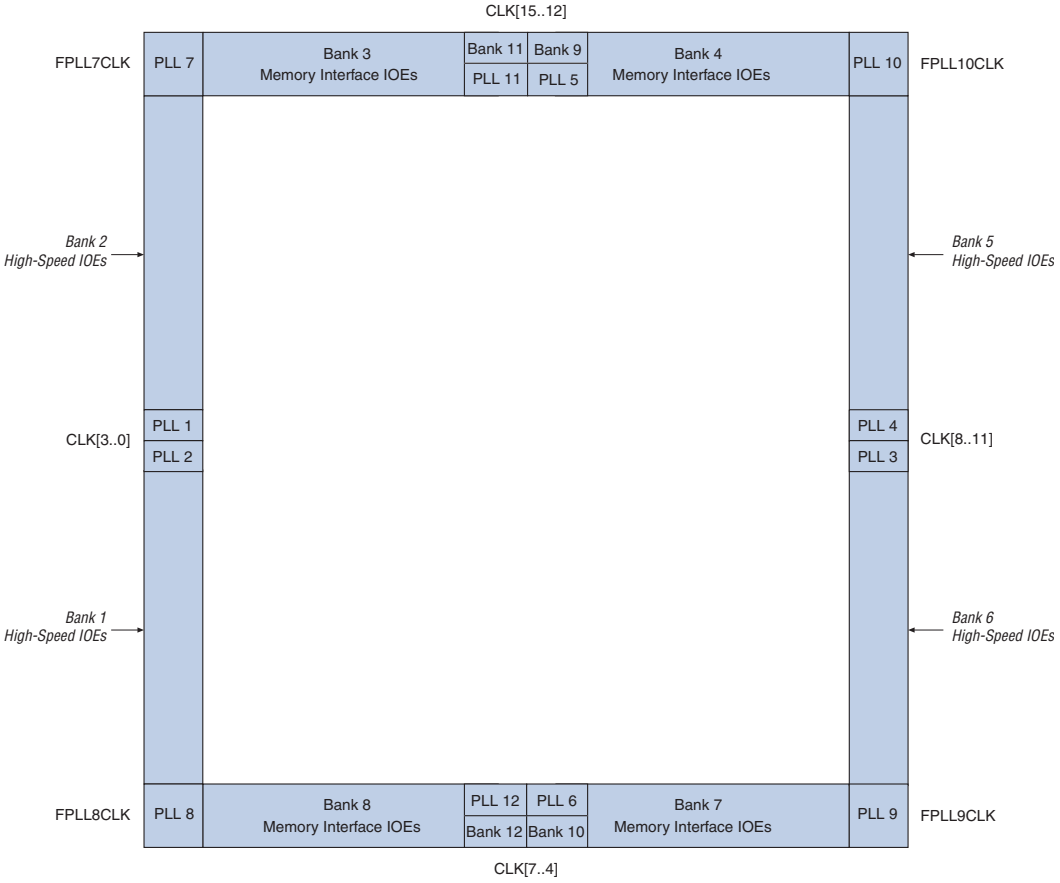
HC230 device fast PLLs 1, 2, 7, and 8 are located in the logic array, next to the device's left high-speed IOEs. HC230 device enhanced PLLs 5, 6, 11, and 12 are also located in the logic array, next to the top and bottom memory interface IOEs.

Figure 8–4. HC230 PLL Locations



HC240 device fast PLLs 1, 2, 7, and 8 are located in the logic array, next to the left high speed IOEs of the device. HC240 device fast PLLs 3, 4, 9, and 10 are located in the logic array, next to the right High Speed IOEs. HC240 device enhanced PLLs 5, 6, 11, and 12 are located in the logic array, next to the top and bottom memory interface IOEs.

Figure 8–5. HC240 PLL Locations



Global & Local Signals

HardCopy II devices have 16 clock pins (`CLK[15..0]`) to drive either the global or local clock networks. Four clock pins drive each side of the device. This is similar to Stratix II devices, therefore there are no limitations when compiling designs for Stratix II devices and HardCopy II companion devices.

Internal logic and enhanced and fast PLL outputs can also drive the global and regional clock networks. Each global and regional clock network has a clock control block, which controls the selection of the

clock source and allows you to dynamically enable or disable the clock network to reduce power consumption. Table 8–19 lists the clock resources available in HardCopy II devices.

Table 8–19. Clock Network Resources & Features Available in HardCopy II Devices

Resources & Features	Availability
Number of global clock networks	16
Number of regional clock networks	32
Global clock input sources	Clock input pins, PLL outputs, logic array
Regional clock input sources	Clock input pins, PLL outputs, logic array
Number of unique clock sources in a quadrant	24 (16 global clocks and 8 regional clocks)
Number of unique clock sources in the entire device	48 (16 global clocks and 32 regional clocks)
Power-down mode	Global and regional clock networks, dual-regional clock region
Clocking regions for high fan-out applications	Quadrant region, dual-regional, entire device via global or regional clock networks

Stratix II ALM Adaptation into HardCopy II Logic

The basic logic building block in the Stratix II architecture is the ALM. Each ALM contains a variety of look-up table- (LUT-) based resources, two programmable registers, two dedicated full adders, and various routing resources to and from the ALM.

HardCopy II devices do not have ALM blocks, but use a fine-grain architecture called HCells. HCells can implement all combinations of Stratix II ALM and DSP logic. Each HardCopy II companion device contains an abundance of HCells to implement a Stratix II design utilizing all available ALMs. Therefore, there are no compatibility constraints when compiling for HardCopy II devices.

When compiling a Stratix II design into a HardCopy II companion device, the Quartus II software replaces ALM blocks used in Stratix II with predefined HCell macros. Unused ALM resources are not implemented in HardCopy II devices. This allows for optimal placement of the HardCopy II floor plan and significant power savings.

Figure 8–6 shows an example of a Stratix II ALM block implementation using only one of the registers. When compiling this Stratix II design for a HardCopy II companion device, the Quartus II compiler replaces the ALM block with a predetermined HCell macro that implements a register from its HardCopy II library of HCell macros. This macro entry has predetermined timing.

Figure 8–6. Stratix II ALM Simple Registered Input & Output

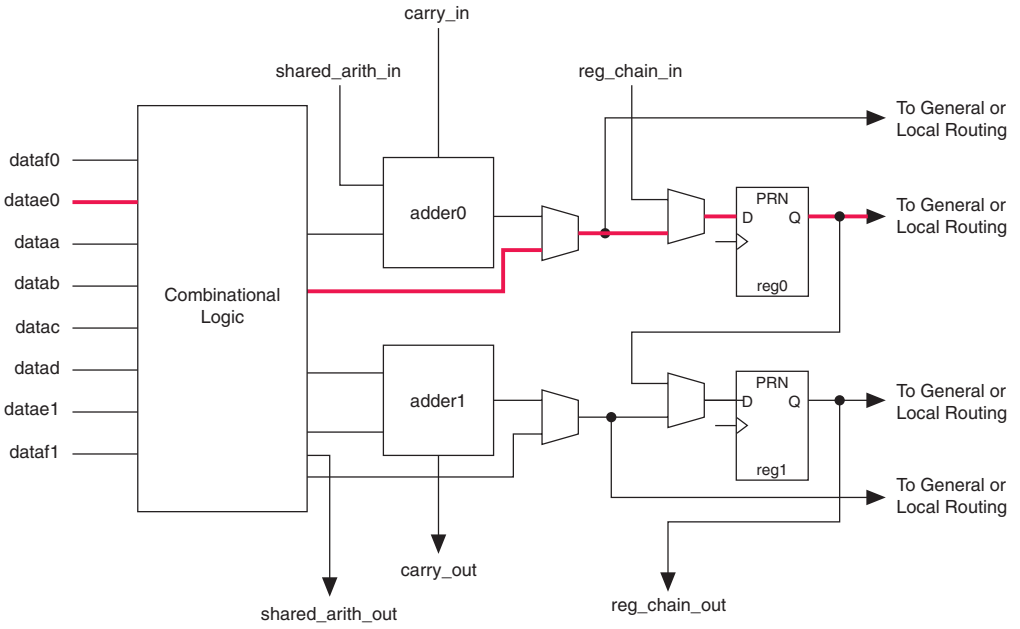
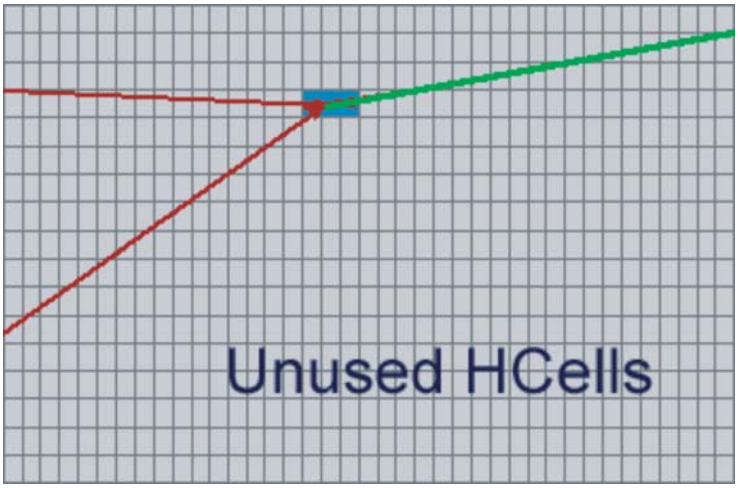


Figure 8–7 shows a HardCopy II ALM register implementation showing Clock, Data In, and Data Out originating from a small cluster of HCells. Unused HCells are reserved for other logic implementation or powered down.

Figure 8–7. HardCopy II Unused HCells



HardCopy II DSP Implementation from Stratix II DSP Blocks

Stratix II FPGAs have dedicated DSP blocks to implement various DSP functions. Stratix II DSP blocks consist of multipliers, an adder/subtractor/accumulator and a summation block, input and output interfaces, and input and output registers. The Quartus II software implements DSP functions in HardCopy II devices with HCells using predetermined logic implementations from its library of HCell macros, all of which have predetermined timing.

DSP blocks that are not used in the Stratix II design are not implemented in Hardcopy II devices. This preserves the HardCopy II logic for other implementations, saving resources and power. Furthermore, the HardCopy II DSP block placement can be optimized to meet the timing constraint requirements placed on the HardCopy II designs.

The HardCopy II DSP implementation is functionally equivalent to Stratix II DSP blocks and all features are supported except for dynamic-mode switching. You can set up Stratix II DSP blocks to dynamically switch between the following three modes:

- Up to four 18-bit independent multipliers
- Up to two 18-bit multiplier-accumulators
- One 36-bit multiplier

HardCopy II DSP implementation does not support dynamic switching. If this feature is used, the Quartus II software flags the DSP implementation and does not allow you to migrate the design. The fitter reports that all HardCopy II devices are not compatible with the design. To migrate your Stratix II design to a HardCopy II companion device, disable dynamic switching in DSP blocks.

The total number of DSP blocks is dependent on the Stratix II device selected. HardCopy II devices will match the available DSP block resources in the Stratix II device. Table 8–20 lists available DSP implementations based on the selected Stratix II device.

Table 8–20. DSP Multiplier Availability for Stratix II & HardCopy II Companion Devices

Stratix II Device	HC210			HC220			HC230			HC240		
	9 × 9	18 × 18	36 × 36	9 × 9	18 × 18	36 × 36	9 × 9	18 × 18	36 × 36	9 × 9	18 × 18	36 × 36
EP2S30	128	64	16									
EP2S60	288	144	36	288	144	36						
EP2S90 *(1)*	384	192	48	384	192	48	384	192	48			
EP2S130 *(1)*				504	252	63	504	252	63			
EP2S180 *(1)*							768	384	96	768	384	96

Note to Table 8–20:

(1) If these Stratix II devices are selected with smaller HardCopy II companion devices, all Stratix II DSP resources may not be available if all the Stratix II ALM blocks are used and fully utilized. Quartus II will determine available resources for DSP and ALM implementation when compiling with HardCopy II devices.

Figure 8–8 shows an example of a Stratix II DSP block that uses only 1 of 8 available 9 × 9 multiplier blocks and an accumulator block to implement an 8 × 8 bit multiplication function with clock latency. When this DSP block is implemented in the HardCopy II design, the Quartus II Compiler chooses the appropriate entry from the macro library to implement the 9 × 9 multiplier and accumulator block which results in an optimized logic utilization and placement flexibility.

Figure 8–8. HardCopy II Floorplan of 8 × 8 DSP Block

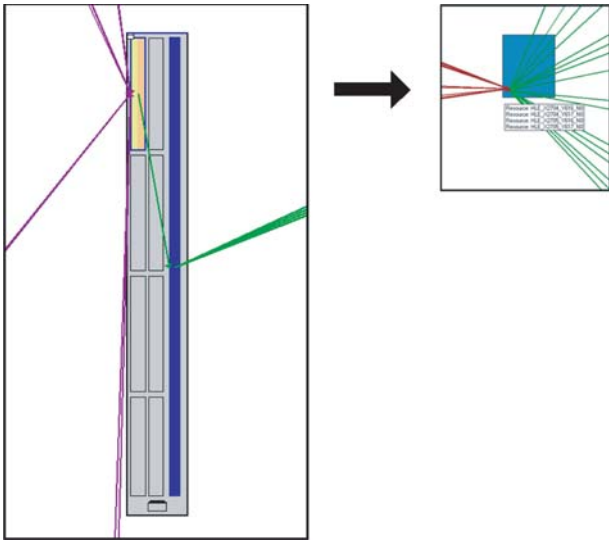
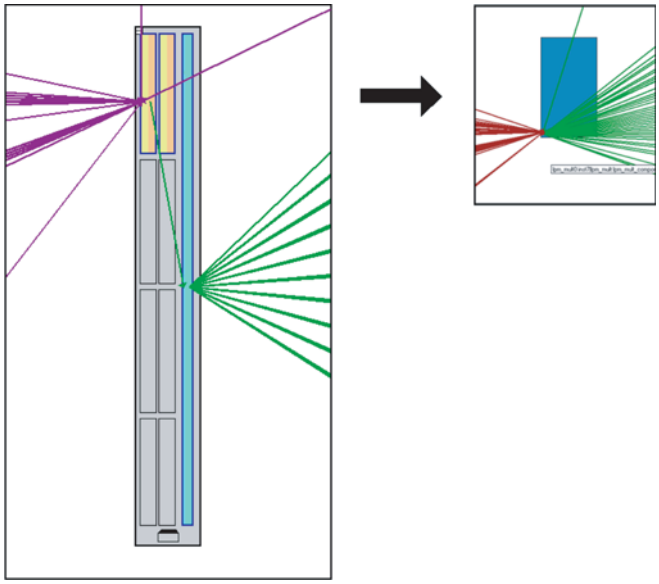


Figure 8–9 shows Quartus II floor plans of a Stratix II DSP block on the left and a HardCopy II DSP implementation on the right, both configured with an 18 × 18 multiply with accumulate function. In the HardCopy II implementation, the Quartus II software selected the appropriate DSP logic implementation from the macro library which results in an optimal utilization of the HardCopy II device's HCells. The unused sections of the Stratix II DSP block remain powered up, but these are not implemented in the HardCopy II device. Unused logic in HardCopy II devices are powered down.

Figure 8–9. HardCopy II Floorplan of 18 × 18 DSP Block



JTAG BST & Extended Functions

HardCopy II devices support the same boundary-scan test (BST) functionality as the Stratix II devices. However, since HardCopy II devices are mask-programmed, no reconfiguration is possible. Therefore, HardCopy II devices do not support instructions to reconfigure device through the JTAG pins. For a list of supported features and instruction codes, refer to the *Boundary-Scan Support* chapter of the *HardCopy Series Handbook*.

One Stratix II feature utilizing JTAG pins is the Signal Tap II embedded logic analyzer (ELA). HardCopy II devices support the JTAG ELA feature. However, designing with this feature will use additional resources and may reduce peak performance in Stratix II and HardCopy II devices. Unlike Stratix II devices, where this feature can be eliminated prior to compiling a final version of the design, HardCopy II devices are masked programmed and this feature will remain permanent in the HardCopy II device. Therefore, if the design requires optimal performance and resource utilization, Altera recommends using this feature on the Stratix II prototype device, but eliminating it prior to recompiling the design for a HardCopy II device.

Power Up & Configuration Compatibility

When designing a board with a Stratix II prototype device and its companion HardCopy II device, most configuration pins required by the Stratix II device are not required by the HardCopy II device. To maximize I/O pin counts with HardCopy II device utilization, Altera recommends minimizing power up and configuration pins that will not carry over from a Stratix II device into a HardCopy II device. Table 8–21 lists the dedicated and optional configuration pins that a Stratix II device can use and if their optional functionality is used on a HardCopy II device.

If the HardCopy II device can use the pin's optional function found in Stratix II devices, the Quartus II software allows you to set these pins as dual purpose pins. As dual purpose pins, they have I/O functionality after power up, reconfiguration and initialization. These pins will only switch to their I/O designation when the device enters user mode (when INIT_DONE is asserted). The design may require that some signals be present when the device transitions into user mode, so you should not use dual purpose pins because it may result in unstable operation after power up for both the HardCopy II and the Stratix II devices.

Table 8–21. Power Up & Configuration Pin Compatibility (Part 1 of 2)

Stratix II Pin Name		I/O Bank	HardCopy II Use	
Main Function	Optional Function		Main Function	Optional Function
MSEL3		B4		
MSEL2		B4		
MSEL1		B4		
MSEL0		B4		
VCCSEL		B8	✓	✓
*n*CONFIG		B8	✓	✓
*n*STATUS		B3	✓	✓
CONF_DONE		B3	✓	✓
*n*CE		B3	✓	✓
*n*CEO		B7	✓	✓
PORSEL		B7	✓	✓
*n*IO_PULLUP		B7	✓	✓
PLL_ENA		B7	✓	✓
I/O pin	CLKUSR	B8	✓	
I/O pin	DEV_OE	B8	✓	✓

Table 8–21. Power Up & Configuration Pin Compatibility (Part 2 of 2)

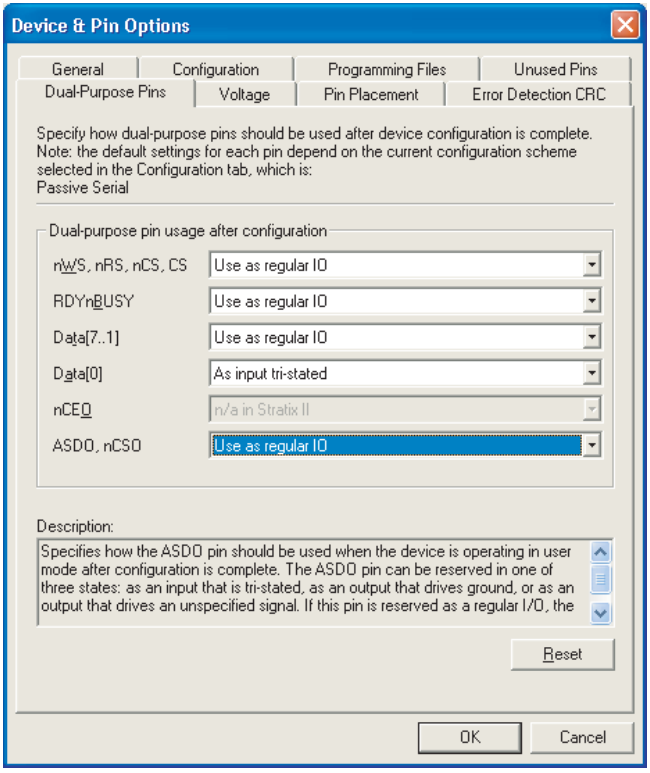
Stratix II Pin Name		I/O Bank	HardCopy II Use	
Main Function	Optional Function		Main Function	Optional Function
I/O pin	DEV_CLRn	B8	✓	✓
I/O pin	INIT_DONE	B3	✓	✓
DCLK		B3	✓	
I/O pin	DATA0	B3	✓	
I/O pin	DATA1	B3	✓	
I/O pin	DATA2	B3	✓	
I/O pin	DATA3	B3	✓	
I/O pin	DATA4	B3	✓	
I/O pin	DATA5	B3	✓	
I/O pin	DATA6	B3	✓	
I/O pin	DATA7	B3	✓	
I/O pin	RDYnBSY	B3	✓	
I/O pin	CRC_ERROR	B3	✓	
I/O pin	CS	B8	✓	
I/O pin	*n*CS	B8	✓	
I/O pin	*n*RS	B8	✓	
I/O pin	*n*WS	B8	✓	
I/O pin	RUnLU	B8	✓	
I/O pin	PGM2	B3	✓	
I/O pin	PGM1	B3	✓	
I/O pin	PGM0	B3	✓	
I/O pin	ASDO	B3	✓	
I/O pin	*n*CSO	B3	✓	

Most optional configuration pins listed in Table 8–21 support the various configuration schemes available in Stratix II FPGAs. Parallel programming and remote update configuration modes utilize most of the pins in Table 8–21. HardCopy II devices are not configurable and do not support the Configuration Emulation mode. Therefore, Altera

recommends that you minimize the configuration pin requirements of the Stratix II design, for example by using the Passive Serial configuration mode.

If some of these dual-purpose pins are needed to configure the Stratix II FPGA, but will be unused after configuration, then these pins will be completely unused on the HardCopy II device. Therefore, when migrating from the Stratix II device to the HardCopy II device, care must be taken when designing these pins on board. The removal of the Stratix II device and its corresponding configuration device may leave these pins floating on the HardCopy II device if such pins are assigned as inputs by the user, without any external means of driving them to a stable level. When selecting a Stratix II device and its device options, consider the after-configuration requirements of these pins and set them appropriately in the Quartus II software (Figure 8–10).

Figure 8–10. Device & Pin Options



For more information about HardCopy II power-up modes, refer to the *Power-Up Modes & Configuration Emulation in HardCopy Series Devices* chapter of the *HardCopy Series Handbook*.

Conclusion

HardCopy II devices provide a seamless migration path for Stratix II devices. HardCopy II devices support the PLL, memory, logic, and I/O features offered on a Stratix II device. HardCopy II device architecture also allows you to use a wide range of Stratix II devices for prototyping. HardCopy II devices offer pin to pin compatibility with Stratix II FPGAs, making HardCopy II devices drop-in replacements on systems designed with the Quartus II software and using Stratix II and HardCopy II companion devices. Use the Quartus II software to compile designs and determine available resources to guarantee fit and feature compatibility of for Stratix II and HardCopy II companion devices.

More Information

For more information on migrating Stratix II designs to HardCopy II devices, Refer to the following sources for more information:

- *HardCopy II Device Family Data Sheet* in the *HardCopy Series Handbook*
- *Quartus II Support for HardCopy II Devices*
- *Power-Up Modes & Configuration Emulation in HardCopy Series Devices* chapter in the *HardCopy Series Handbook*
- *Prototyping Strategy for HardCopy II Devices* chapter in the *HardCopy Series Handbook*

Section II. HardCopy Stratix Device Family Data Sheet

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

This section contains the following:

- Chapter 9, Introduction to HardCopy Stratix Devices
- Chapter 10, Description, Architecture & Features
- Chapter 11, Boundary-Scan Support
- Chapter 12, Operating Conditions
- Chapter 13, Quartus II Support for HardCopy Stratix Devices
- Chapter 14, Design Guidelines for HardCopy Stratix Performance Improvement

Revision History

The table below shows the revision history for Chapters 9 through 14.

Chapter(s)	Date / Version	Changes Made
Chapter 9	March 2006	Formerly chapter 5; no content change.
	October 2005 v2.1	Minor edits
	January 2005 v2.0	Minor edits
	June 2003 v1.0	Initial release of Chapter 5, *Introduction to HardCopy Stratix Devices*, in the *HardCopy Device Handbook*.
Chapter 10	March 2006	Formerly chapter 6; no content change.
	October 2005 v3.1	• Minor edits • Updated graphics

Chapter(s)	Date / Version	Changes Made
	May 2005 v3.0	• Added Table 6-1 • Added the Logic Elements section • Added the Embedded Memory section • Added the DSP Blocks section • Added the PLLs & Clock Networks section • Added the I/O Structure & Features section
	January 2005 v2.0	• Added summary of I/O and timing differences between Stratix FPGAs and HardCopy Stratix devices • Removed section on Quartus II support of HardCopy Stratix devices • Added “Hot Socketing” section
	August 2003 v1.1	Edited section headings’ hierarchy.
	June 2003 v1.0	Initial release of Chapter 6, Description, Architecture & Features, in the *HardCopy Device Handbook*
Chapter 11	March 2006	Formerly chapter 7; no content change.
	October 2005 v3.1	• Minor edits • Graphic updates
	May 2005 v3.0	Updated “IEEE Std. 1149.1 (JTAG) Boundary-Scan Support” section
	January 2005 v2.0	Added information about USERCODE registers
	June 2003 v1.0	Initial release of Chapter 7, Boundary-Scan Support, in the *HardCopy Device Handbook*
Chapter 12	March 2006	Formerly chapter 8; no content change.
	October 2005 v3.1	• Minor edits • Graphic updates
	May 2005 v3.0	• Updated SSTL-2 and SSTL-3 specifications in Tables 8–19 through 8–22 • Updated CTT I/O specifications in Table 8–30 • Updated bus hold parameters in Table 8–31. • Added the External Timing Parameters, HardCopy Stratix External I/O Timing, and Maximum Input & Output Clock Rates sections • Added the High-Speed I/O Specification, and PLL Specifications sections
	January 2005 v2.0	Removed recommended maximum rise and fall times (t_R and t_F) for input signals
	June 2003 v1.0	Initial release of Chapter 8, Operating Conditions, in the *HardCopy Device Handbook*

Chapter(s)	Date / Version	Changes Made
Chapter 13	March 2006	Formerly chapter 20; no content change.
	October 2005 v3.1	• Updated for technical contents for Quartus II 5.1 release • Minor edits
	May 2005 v3.0	Added PowerPlay early Power estimator information.
	January 2005 v2.0	This revision was previously the *Quartus® II Support for HardCopy Devices* chapter in the *Quartus II Development Software Handbook, v4.1.*
	August 2003 v1.1	Overall edit; added Tcl script appendix.
	June 2003 v1.0	Initial release of Chapter 20, Quartus II Support for HardCopy Stratix Devices.
Chapter 14	March 2006	Formerly chapter 21; no content change.
	October 2005 v1.1	• Updated graphics • Minor edits
	July 2005 v1.0	Initial release of Chapter 21, Design Guidelines for HardCopy Stratix Performance Improvement.

9. Introduction to HardCopy Stratix Devices

H51001-2.1

Introduction

HardCopy® Stratix® structured ASICs, Altera's second-generation HardCopy structured ASICs, are low-cost, high-performance devices with the same architecture as the high-density Stratix FPGAs. The combination of Stratix FPGAs for prototyping and design verification, HardCopy Stratix devices for high volume production, and the Quartus® II design software beginning with version 3.0, provide a complete and powerful alternative to ASIC design and development.

HardCopy Stratix devices are architecturally equivalent and have the same features as the corresponding Stratix FPGA. They offer pin-to-pin compatibility using the same package as the corresponding Stratix FPGA prototype. Designers can prototype their design to verify functionality with Stratix FPGAs before seamlessly migrating the proven design to a HardCopy Stratix structured ASIC.

The Quartus II software provides a complete set of inexpensive and easy-to-use tools for designing HardCopy Stratix devices. Using the successful and proven methodology from HardCopy APEX™ devices, Stratix FPGA designs can be seamlessly and quickly migrated to a low-cost ASIC alternative. Designers can use the Quartus II software to design HardCopy Stratix devices to obtain an average of 50% higher performance and 40% lower power consumption than can be achieved in the corresponding Stratix FPGAs. The migration process is fully automated, requires minimal customer involvement, and takes approximately eight weeks to deliver fully tested HardCopy Stratix prototypes.

The HardCopy Stratix devices use the same base arrays across multiple designs for a given device density and are customized using the top two metal layers. The HardCopy Stratix family consists of the HC1S25, HC1S30, HC1S40, HC1S60, and HC1S80 devices. Table 9–1 provides the details of the HardCopy Stratix devices.

Table 9–1. HardCopy Stratix Devices & Features

Device	LEs *(1)*	M512 Blocks	M4K Blocks	M-RAM Blocks	DSP Blocks *(2)*	PLLs *(3)*
HC1S25	25,660	224	138	2	10	6
HC1S30	32,470	295	171	2 *(4)*	12	6
HC1S40	41,250	384	183	2 *(4)*	14	6
HC1S60	57,120	574	292	6	18	12
HC1S80	79,040	767	364	6 *(4)*	22	12

Notes to Table 9–1:
(1) LE: logic elements.
(2) DSP: digital signal processing.
(3) PLLs: phase-locked loops.
(4) In HC1S30, HC1S40, and HC1S80 devices, there are fewer M-RAM blocks than in the equivalent Stratix FPGA. All other resources are identical to the Stratix counterpart.

Features

HardCopy Stratix devices are manufactured on the same 1.5 V, 0.13 µm all-layer-copper metal fabrication process (up to eight layers of metal) as the Stratix FPGAs.

- Preserves the functionality of a configured Stratix device
- Pin-compatible with the Stratix counterparts
- On average, 50% faster than their Stratix equivalents
- On average, 40% less power consumption than their Stratix equivalents
- 25,660 to 79,040 LEs
- Up to 5,658,408 RAM bits available
- TriMatrix™ memory architecture consisting of three RAM block sizes to implement true dual-port memory and first-in-first-out (FIFO) buffers
- Embedded high-speed DSP blocks provide dedicated implementation of multipliers, multiply-accumulate functions, and finite impulse response (FIR) filters
- Up to 12 PLLs (four enhanced PLLs and eight fast PLLs) per device which provide identical features as the FPGA counterparts, including spread spectrum, programmable bandwidth, clock switchover, real-time PLL reconfiguration, advanced multiplication and phase shifting
- Supports numerous single-ended and differential I/O standards
- Supports high-speed networking and communications bus standards including RapidIO™, UTOPIA IV, CSIX, HyperTransport™ technology, 10G Ethernet XSBI, SPI-4 Phase 2 (POS-PHY Level 4), and SFI-4
- Differential on-chip termination support for LVDS

- Supports high-speed external memory, including zero bus turnaround (ZBT) SRAM, quad data rate (QDR and QDRII) SRAM, double data rate (DDR) SDRAM, DDR fast-cycle RAM (FCRAM), and single data rate (SDR) SDRAM
- Support for multiple intellectual property (IP) megafunctions from Altera® MegaCore® functions, and Altera Megafunction Partners Program (AMPP^{SM}) megafunctions
- Available in space-saving flip-chip FineLine BGA® and wire-bond packages (Tables 9–2 and 9–3)
- Optional emulation of original FPGA configuration sequence
- Optional instant-on power-up

The actual performance and power consumption improvements over the Stratix equivalents mentioned in this data sheet are design-dependent.

Table 9–2. HardCopy Stratix Device Package Options & I/O Pin Counts *Note (1)*

Device	672-Pin FineLine BGA *(2)*	780-Pin FineLine BGA *(3)*	1,020-Pin FineLine BGA *(3)*
HC1S25	473		
HC1S30		597	
HC1S40		613 *(4)*	
HC1S60			773
HC1S80			773

Notes to Table 9–2:

(1) Quartus II I/O pin counts include one additional pin, `PLLENA`, which is not a general-purpose I/O pin. `PLLENA` can only be used to enable the PLLs.
(2) This device uses a wire-bond package.
(3) This device uses a flip-chip package.
(4) In the Stratix EP1S40F780 FPGA, the I/O pins `U12` and `U18` are general-purpose I/O pins. In the FPGA prototype, EP1S40F780_HARDCOPY_FPGA_PROTOTYPE, and in the HardCopy Stratix HC1S40F780 device, `U12` and `U18` must be connected to ground. The EP1S40F780_HARDCOPY_FPGA_PROTOTYPE and HC1S40F780 pin-outs are identical.

Table 9–3. HardCopy Stratix Device Package Sizes

Device	672-Pin FineLine BGA	780-Pin FineLine BGA	1,020-Pin FineLine BGA
Pitch (mm)	1.00	1.00	1.00
Area (mm^2)	729	841	1,089
Length × width (mm × mm)	27 × 27	29 × 29	33 × 33

10. Description, Architecture & Features

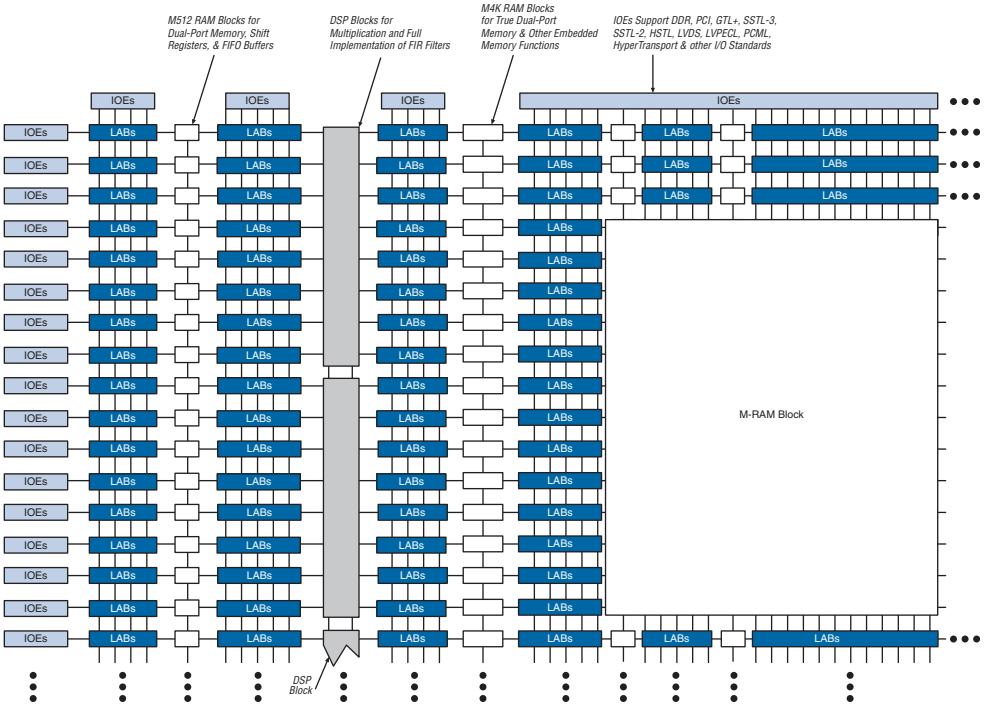
H51002-3.1

Functional Description

HardCopy® Stratix® structured ASICs provide a comprehensive alternative to ASICs. The HardCopy Stratix device family is fully supported by the Quartus® II design software, and, combined with a vast intellectual property (IP) portfolio, provides a complete path from prototype to volume production. Designers can now procure devices, tools, and Altera® IP for their high-volume applications.

As shown in Figure 10–1, HardCopy Stratix devices preserve their Stratix FPGA counterpart's architecture, but the programmability for logic, memory, and interconnect is removed. HardCopy Stratix devices are also manufactured in the same process technology and process voltage as Stratix FPGAs. Removing all configuration and programmable routing resources and replacing it with direct metal interconnect results in considerable die size reduction and the ensuing cost savings.

Figure 10–1. HardCopy Stratix Device Architecture



The HardCopy Stratix family consists of base arrays that are common to all designs for a particular device density. Design-specific customization is done within the top two metal layers. The base arrays use an area-efficient sea-of-logic-elements (SOLE) core and extend the flexibility of high-density Stratix FPGAs to a cost-effective, high-volume production solution. With a seamless migration process employed in numerous successful designs, functionality-verified Stratix FPGA designs can be migrated to fixed-function HardCopy Stratix devices with minimal risk and guaranteed first-time success.

The SRAM configuration cells of the original Stratix devices are replaced in HardCopy Stratix devices by metal connects, which define the function of each logic element (LE), digital signal processing (DSP) block, phase-locked loop (PLL), embedded memory, and I/O cell in the device. These resources are interconnected using metallization layers. Once a HardCopy Stratix device has been manufactured, the functionality of the device is fixed and no re-programming is possible. However, as is the case with Stratix FPGAs, the PLLs can be dynamically configured in HardCopy Stratix devices.

HardCopy Stratix & Stratix FPGA Differences

To ensure HardCopy Stratix device functionality and performance, designers should thoroughly test the original Stratix FPGA-based design for satisfactory results before committing the design for migration to a HardCopy Stratix device. Unlike Stratix FPGAs, HardCopy Stratix devices are customized at the time of manufacturing and therefore do not have programmability support.

Since HardCopy Stratix devices are customized within the top two metal layers, no configuration circuitry is required. Refer to "Power-Up Modes in HardCopy Stratix Devices" on page 10–8 for more information on this topic.

Depending on the design, HardCopy Stratix devices can provide, on average, a 50% performance improvement over equivalent Stratix FPGAs. The performance improvement is achieved by die size reduction, metal interconnect optimization, and customized signal buffering. HardCopy Stratix devices consume, on average, 40% less power than their equivalent Stratix FPGAs.

Designers can use the Quartus II software to design HardCopy Stratix devices, estimate performance and power consumption, and maximize system throughput.

Table 10–1 illustrates the differences between HardCopy Stratix and Stratix devices.

Table 10–1. HardCopy Stratix & Stratix Device Comparison (Part 1 of 2)

HardCopy Stratix	Stratix
Customized device. All reprogrammability support is removed and no configuration is required.	Re-programmable with configuration is required upon power-up.
Average of 50% performance improvement over corresponding FPGA.	High-performance FPGA.
Average of 40% less power consumption compared to corresponding FPGA.	Standard FPGA power consumption.
Contact Altera for information regarding specific IP support.	IP support for all devices is available.
Double data rate (DDR) SDRAM maximum operating frequency is pending characterization.	DDR SDRAM can operate at 200 MHz for -5 speed grade devices.
All routing connections are direct and all unused routing is removed.	MultiTrack™ routing stitches together routing resources to provide a path.

Table 10–1. HardCopy Stratix & Stratix Device Comparison (Part 2 of 2)

HardCopy Stratix	Stratix
HC1S30 and HC1S40 devices have two M-RAM blocks. HC1S80 devices have six M-RAM blocks.	EP1S30 and EP1S40 devices have four M-RAM blocks. EP1S80 devices have nine M-RAM blocks.
It is not possible to initialize M512 and M4K RAM contents during power-up.	The contents of M512 and M4K RAM blocks can be preloaded during configuration with data specified in a memory initialization file (**.mif**).
The contents of memory output registers are unknown after power-on reset (POR).	The contents of memory output registers are initialized to '0' after POR.
HC1S30 and HC1S40 devices have six PLLs.	HC1S30 devices have 10 PLLs. HC1S40 devices have 12 PLLs.
PLL dynamic reconfiguration uses ROM for information. This reconfiguration is performed in the back-end and does not affect the migration flow.	PLL dynamic reconfiguration uses a MIF to initialize a RAM resource with information.
The I/O elements (IOEs) are equivalent but not identical to FPGA IOEs due to slight design optimizations for HardCopy devices.	The IOEs are optimized for the FPGA architecture.
The I/O drive strength for single-ended I/O pins are slightly different and is modeled in the HardCopy Stratix IBIS models.	The I/O drive strength for single-ended I/O pins are found in Stratix IBIS models.
In the HC1S40 780-pin FineLine BGA® device, the I/O pins U12 and U18 must be connected to ground.	In the HC1S40 780-pin FineLine BGA device, the I/O pins U12 and U18 are available as general-purpose I/O pins.
The BSDL file describes re-ordered Joint Test Action Group (JTAG) boundary-scan chains.	The JTAG boundary-scan chain is defined in the BSDL file.

Logic Elements

Logic is implemented in HardCopy Stratix devices using the same architectural units as the Stratix device family. The basic unit is the logic element (LE) with logic array blocks (LAB) consisting of 10 LEs. The implementation of LEs and LABs is identical to the Stratix device family.

In the HardCopy Stratix device family, all extraneous routing resources not essential to the specific design are removed for performance and die size efficiency. Therefore, the MultiTrack interconnect for routing

implementation between LABs and other device resources in the Stratix device family is no longer necessary in the HardCopy Stratix device family.

Table 10–2 illustrates the differences between HardCopy Stratix and Stratix logic.

Table 10–2. HardCopy Stratix & Stratix Logic Comparison

HardCopy Stratix	Stratix
All routing connections are direct and all unused routing is removed.	MultiTrack routing stitches routing resources together to provide a path.

Embedded Memory

TriMatrix™ memory blocks from Stratix devices, including M512, M4K, and M-RAM memory blocks are available in HardCopy Stratix devices. Embedded memory is seamlessly implemented in the equivalent resource.

Although memory resource implementation is equivalent, the number of specific M-RAM blocks are not necessarily the same between corresponding Stratix and HardCopy Stratix devices. Table 10–3 shows the number of M-RAM blocks available in each device.

Table 10–3. HardCopy Stratix & Stratix M-RAM Block Comparison

HardCopy Stratix		Stratix	
Device	M-RAM Blocks	Device	M-RAM Blocks
HC1S25	2	EP1S25	2
HC1S30	2	EP1S30	4
HC1S40	2	EP1S40	4
HC1S60	6	EP1S60	6
HC1S830	6	EP1S830	9

In HardCopy Stratix devices, it is not possible to preload RAM contents using a MIF after powering up; the output registers of memory blocks will have unknown values. This occurs because there is no configuration process that is executed.

Table 10–4 illustrates the differences between HardCopy Stratix and Stratix memory.

Table 10–4. HardCopy Stratix & Stratix Memory Comparison

HardCopy Stratix	Stratix
HC1S30 and HC1S40 devices have two M-RAM blocks. HC1S80 devices have six M-RAM blocks.	EP1S30 and EP1S40 devices have four M-RAM blocks. EP1S80 devices have nine M-RAM blocks.
It is not possible to initialize M512 and M4k RAM contents during power-up.	The contents of M512 and M4K RAM blocks can be preloaded during configuration with data specified in a MIF.
The contents of memory output registers are unknown after POR.	The contents of memory output registers are initialized to '0' after POR.

DSP Blocks

DSP blocks in HardCopy Stratix devices are architecturally identical to those in Stratix devices. The number of DSP blocks available in HardCopy Stratix devices matches the number of DSP blocks available in the corresponding Stratix device.

PLLs & Clock Networks

The PLLs in HardCopy Stratix devices are identical to those in Stratix devices. The clock networks are also implemented exactly as they are in Stratix devices. The number of PLLs can vary between corresponding Stratix and HardCopy Stratix devices. Table 10–5 shows the number of PLLs available in each device.

Table 10–5. HardCopy Stratix & Stratix PLL Comparison

HardCopy Stratix		Stratix	
Device	PLLs	Device	PLLs
HC1S25	6	EP1S25	6
HC1S30	6	EP1S30	10
HC1S40	6	EP1S40	12
HC1S60	12	EP1S60	12
EP1S830	12	EP1S830	12

Table 10–6 illustrates the differences between HardCopy Stratix and Stratix PLLs.

Table 10–6. HardCopy Stratix & Stratix PLL Differences

HardCopy Stratix	Stratix
HC1S30 and HC1S40 devices have six PLLs.	HC1S30 devices have 10 PLLs. HC1S40 devices have12 PLLs.
PLL dynamic reconfiguration uses ROM for information. This reconfiguration is performed in the back-end and does not affect the migration flow.	PLL dynamic reconfiguration uses a MIF to initialize a RAM resource with information.

I/O Structure & Features

The HardCopy Stratix IOEs are equivalent but not identical to the Stratix FPGA IOEs. This is due to the reduced die size, layout difference, and metal customization of the HardCopy Stratix device. The differences are minor but may be relevant to customers designing with tight DC and switching characteristics. However, no signal integrity concerns are introduced with HardCopy Stratix IOEs.

When designing with very tight timing constraints (for example, DDR or quad data rate (QDR)), or if using the programmable drive strength option, Altera recommends verifying final drive strength using updated IBIS models located on the Altera web site at **www.altera.com**. Differential I/O standards are unaffected.

I/O pin placement and VREF pin placement rules are identical between HardCopy Stratix and Stratix devices. Unused pin settings will carry over from Stratix device settings and are implemented as tri-stated outputs driving ground or outputs driving V_{CC}.

In Stratix EP1S40 780-pin FineLine BGA FPGAs, the I/O pins `U12` and `U18` are available as general-purpose I/O pins. In the FPGA prototype, EP1S40F780_HARDCOPY_FPGA_PROTOTYPE, and in the Hardcopy Stratix HC1S40 780-pin FineLine BGA device, the I/O pins `U12` and `U18` must be connected to ground. HC1S40 780-pin FineLine BGA and EP1S40F780_HARDCOPY_FPGA_PROTOTYPE pin-outs are identical.

Table 10–7 illustrates the differences between HardCopy Stratix and Stratix I/O pins.

Table 10–7. HardCopy Stratix & Stratix I/O Pin Comparison

HardCopy Stratix	Stratix
The IOEs are equivalent but not identical to the FPGA IOEs due to slight design optimizations for HardCopy devices.	IOEs are optimized for the FPGA architecture.
The I/O drive strength for single-ended I/O pins are slightly different and are found in the HardCopy Stratix IBIS models.	The I/O drive strength for single-ended I/O pins are found in Stratix IBIS models.
In the HC1S40 780-pin FineLine BGA device, the I/O pins U12 and U18 must be connected to ground.	In the EP1S40 780-pin FineLine BGA device, the I/O pins U12 and U18 are available as general-purpose I/O pins.

Power-Up Modes in HardCopy Stratix Devices

Designers do not need to configure HardCopy Stratix devices, unlike their FPGA counterparts. However, to facilitate seamless migration, configuration can be emulated in HardCopy Stratix devices.

The modes in which a HardCopy Stratix device can be made ready for operation after power-up are: instant on, instant on after 50 ms, and configuration emulation. These modes are briefly described below.

- In instant on mode, the HardCopy Stratix device is available for use shortly after the device receives power. The on-chip POR circuit resets all registers. The CONF_DONE output is tri-stated once the POR has elapsed. No configuration device or configuration data is necessary.
- In instant on after 50 ms mode, the HardCopy Stratix device will perform in a fashion similar to the instant on mode, except that there is an additional delay of 50 ms, during which time the device is held in reset stage. The CONF_DONE output is pulled low during this time, and then tri-stated after the 50 ms have elapsed. No configuration device or configuration data is necessary for this option.

- In configuration emulation mode, the HardCopy series device emulates the behavior of an APEX or Stratix FPGA during its configuration phase. When this mode is used, the HardCopy device uses a configuration emulation circuit to receive configuration bit streams. When all the configuration data is received, the HardCopy series device transitions into an initialization phase and releases the CONF_DONE pin to be pulled high. Pulling the CONF_DONE pin

high signals that the HardCopy series device is ready for normal operation. If the optional open-drain INIT_DONE output is used, the normal operation is delayed until this signal is released by the HardCopy series device.

HardCopy II and some HardCopy Stratix devices do not support configuration emulation mode.

Instant on and instant on after 50 ms modes are the recommended power-up modes because these modes are similar to an ASIC's functionality upon power-up. No changes to the existing board design or the configuration software are required.

All three of these modes provide significant benefits to system designers. They enable seamless migration of the design from the FPGA device to the HardCopy device with no changes to the existing board design or the configuration software. The pull-up resistors on *n*CONFIG, *n*STATUS, and CONF_DONE should be left on the printed circuit board.

For more information, refer to the *HardCopy Series Configuration Emulation* chapter in the *HardCopy Series Handbook*.

Hot Socketing

HardCopy Stratix devices support hot socketing without any external components. In a hot socketing situation, a device's output buffers are turned off during system power-up or power-down. To simplify board design, HardCopy Stratix devices support any power-up or power-down sequence (V_{CCIO} and V_{CCINT}). For mixed-voltage environments, you can drive signals into the device before or during power-up or power-down without damaging the device. HardCopy Stratix devices do not drive out until they have attained proper operating conditions.

You can power up or power down the V_{CCIO} and V_{CCINT} pins in any sequence. The power supply ramp rates can range from 100 ns to 100 ms. During hot socketing, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

- The hot socketing DC specification is $| I_{IOPIN} | < 300$ µA.
- The hot socketing AC specification is dependent on the signaling voltage and board capacitance: $| I_{IOPIN} | (\Delta v / \Delta t) \times$ capacitance where capacitance is the sum of the I/O, trace, and connector capacitance.

The DC specification applies when all V_{CC} supplies to the device are stable in the powered-up or powered-down conditions. The AC specification applies when the device is being powered up or powered down in any of the conditions mentioned above.

HARDCOPY_ FPGA_ PROTOTYPE Devices

HARDCOPY_FPGA_PROTOTYPE devices are Stratix FPGAs available for designers to prototype their HardCopy Stratix designs and perform in-system verification before migration to a HardCopy Stratix device. The HARDCOPY_FPGA_PROTOTYPE devices have the same available resources as in the final HardCopy Stratix devices.

The Quartus II software version 4.1 and later contains the latest timing models. For designs with tight timing constraints, Altera strongly recommends compiling the design with the Quartus II software version 4.1 or later. To properly verify I/O features, it is important to design with the HARDCOPY_FPGA_PROTOTYPE device option prior to migrating to a HardCopy Stratix device.

Some HARDCOPY_FPGA_PROTOTYPE devices, as indicated in Table 10–8, have fewer M-RAM blocks compared to the equivalent Stratix FPGAs. The selective removal of these resources provides a significant price benefit to designers using HardCopy Stratix devices.

Table 10–8. M-RAM Block Comparison Between Various Devices

Number of LEs	HARDCOPY_FPGA_PROTOTYPE Devices		HardCopy Stratix Devices		Stratix Devices	
	Device	M-RAM Blocks	Device	M-RAM Blocks	Device	M-RAM Blocks
25,660	EP1S25	2	HC1S25	2	EP1S25	2
32,470	EP1S30	2	HC1S30	2	EP1S30	4
41,250	EP1S40	2	HC1S40	2	EP1S40	4
57,120	EP1S60	6	HC1S60	6	EP1S60	6
79,040	EP1S830	6	HC1S830	6	EP1S830	9

For more information about how the various features in the Quartus II software can be used for designing HardCopy Stratix devices, refer to the *Quartus II Support for HardCopy Stratix Devices* chapter of the *HardCopy Series Handbook*.

HARDCOPY_FPGA_PROTOTYPE FPGA devices have the identical speed grade as the equivalent Stratix FPGAs. However, HardCopy Stratix devices are customized and do not have any speed grading. HardCopy Stratix devices, on an average, can be 50% faster than their equivalent HARDCOPY_FPGA_PROTOTYPE devices. The actual improvement is design-dependent.

11. Boundary-Scan Support

H51004-3.1

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All HardCopy® Stratix® structured ASICs provide JTAG BST (Boundary-Scan Test) circuitry that complies with the IEEE Std. 1149.1-1990 specification. The BST architecture offers the capability to efficiently test components on printed circuit boards (PCBs) with tight lead spacing by testing pin connections, without using physical test probes, and capturing functional data while a device is in normal operation. Boundary-scan cells in a device can force signals onto pins, or capture data from pin or core logic signals. Forced test data is serially shifted into the boundary-scan cells. Captured data is serially shifted out and externally compared to expected results.

A device using the JTAG interface uses four required pins, TDI, TDO, TMS, and TCK, and one optional pin, TRST. HardCopy Stratix devices support the JTAG instructions shown in Table 11–1.

Table 11–1. HardCopy Stratix 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.
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.
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.

Table 11–1. HardCopy Stratix JTAG Instructions (Part 2 of 2)

JTAG Instruction	Instruction Code	Description
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.

Note to Table 11–1:
(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

The Boundary-Scan Description Language (BSDL) files for HardCopy Stratix devices are different from the corresponding Stratix FPGAs. The BSDL files for HardCopy Stratix devices are available for download from **www.altera.com**.

The HardCopy Stratix device instruction register length is 10 bits; the USERCODE register length is 32 bits. The USERCODE registers are mask-programmed, so they are not re-programmable. The designer can choose an appropriate 32-bit sequence to program into the USERCODE registers.

Tables 11–2 and 11–3 show the boundary-scan register length and device IDCODE information for HardCopy Stratix devices.

Table 11–2. HardCopy Stratix Boundary-Scan Register Length

Device	Maximum Boundary-Scan Register Length
HC1S25 672-pin FineLine® BGA	1,458
HC1S30 780-pin FineLine BGA	1,878
HC1S40 780-pin FineLine BGA	1,878
HC1S60 1,020-pin FineLine BGA	2,382
HC1S80 1,020-pin FineLine BGA	2,382

Table 11–3. 32-Bit HardCopy Stratix Device IDCODE

Device	IDCODE (32 Bits) *(1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) *(2)*
HC1S25	0000	0010 0000 0000 0011	000 0110 1110	1
HC1S30	0000	0010 0000 0000 0100	000 0110 1110	1
HC1S40	0000	0010 0000 0000 0101	000 0110 1110	1
HC1S60	0000	0010 0000 0000 0110	000 0110 1110	1
HC1S80	0000	0010 0000 0000 0111	000 0110 1110	1

Notes to Table 11–3:
(1) The most significant bit (MSB) is on the left.
(2) The IDCODE's least significant bit (LSB) is always 1.

Figure 11–1 shows the timing requirements for the JTAG signals.

Figure 11–1. HardCopy Stratix JTAG Waveforms

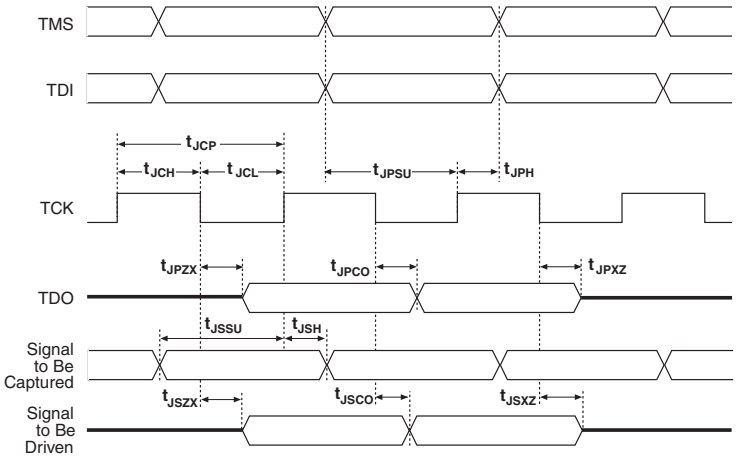


Table 11–4 shows the JTAG timing parameters and values for HardCopy Stratix devices.

Table 11–4. HardCopy Stratix JTAG Timing Parameters & 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

For more information on JTAG, refer to *AN 39: IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*.

12. Operating Conditions

H51005-3.1

Recommended Operating Conditions

Tables 12–1 through 12–3 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 1.5-V HardCopy® Stratix® devices.

Table 12–1. HardCopy Stratix Device Absolute Maximum Ratings *Notes (1), (2)*

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCINT}	Supply voltage	With respect to ground	–0.5	2.4	V
V_{CCIO}			–0.5	4.6	V
V_I	DC input voltage *(3)*		–0.5	4.6	V
I_{OUT}	DC output current, per pin		–25	40	mA
T_{STG}	Storage temperature	No bias	–65	150	°C
T_J	Junction temperature	BGA packages under bias		135	°C

Table 12–2. HardCopy Stratix Device Recommended Operating Conditions

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)*, *(5)*	3.00 (3.135)	3.60 (3.465)	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)*, *(6)*	–0.5	4.1	V
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

Table 12–3. HardCopy Stratix Device DC Operating Conditions *Note (7)*

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 memory blocks in power-down mode)	V_I = ground, no load, no toggling inputs				mA
R_{CONF}	Value of I/O pin pull-up resistor before and during configuration	V_{CCIO} = 3.0 V *(9)*	20		50	kΩ
		V_{CCIO} = 2.375 V *(9)*	30		80	kΩ
		V_{CCIO} = 1.71 V *(9)*	60		150	kΩ

Notes to Tables 12–1 through 12–3:

(1) Refer to the *Operating Requirements for Altera Devices Data Sheet*.
(2) Conditions beyond those listed in Table 12–1 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 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) V_{CCIO} maximum and minimum conditions for LVPECL, LVDS, and 3.3-V PCML are shown in parentheses.
(6) All pins, including dedicated inputs, clock, I/O, and JTAG pins, may be driven before V_{CCINT} and V_{CCIO} are powered.
(7) 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.
(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) Pin pull-up resistance values will lower if an external source drives the pin higher than V_{CCIO}.

Tables 12–4 through 12–31 list the DC operating specifications for the supported I/O standards. These tables list minimal specifications only; HardCopy Stratix devices may exceed these specifications. Table 12–32 provides information on capacitance for 1.5-V HardCopy Stratix devices.

Table 12–4. 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 *(1)*	2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 4 to 24 mA *(1)*		0.45	V

Table 12–5. 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 12–6. 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 *(1)*	1.7		V
V_{OL}	Low-level output voltage	I_{OL} = 0.1 mA		0.2	V
		I_{OL} = 1 mA		0.4	V
		I_{OL} = 2 to 16 mA *(1)*		0.7	V

Table 12–7. 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	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 *(1)*	V_{CCIO} – 0.45		V
V_{OL}	Low-level output voltage	I_{OL} = 2 to 8 mA *(1)*		0.45	V

Table 12–8. 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	V
V_{IL}	Low-level input voltage		–0.3	0.35 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OH} = –2 mA *(1)*	0.75 × V_{CCIO}		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(1)*		0.25 × V_{CCIO}	V

Table 12–9. 3.3-V LVDS I/O Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{ID}	Input differential voltage swing	0.1 V < V_{CM} < 1.1 V *J* = 1 through 10	300		1,000	mV
		1.1 V ≤ V_{CM} ≤ 1.6 V J = 1	200		1,000	mV
		1.1 V ≤ V_{CM} ≤ 1.6 V *J* = 2 through10	100		1,000	mV
		1.6 V < V_{CM} < 1.8 V *J* = 1 through 10	300		1,000	mV
V_{ICM}	Input common mode voltage	LVDS 0.3 V < V_{ID} < 1.0 V *J* = 1 through 10	100		1,100	mV
		LVDS 0.3 V < V_{ID} < 1.0 V *J* = 1 through 10	1,600		1,800	mV
		LVDS 0.2 V < V_{ID} < 1.0 V *J* = 1	1,100		1,600	mV
		LVDS 0.1 V < V_{ID} < 1.0 V *J* = 2 through 10	1,100		1,600	mV
V_{OD} *(2)*	Output differential voltage	R_L = 100 Ω	250	375	550	mV
Δ V_{OD}	Change in V_{OD} between high and low	R_L = 100 Ω			50	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	1,125	1,200	1,375	mV

Table 12–9. 3.3-V LVDS I/O Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
ΔV_{OCM}	Change in V_{OCM} between high and low	R_L = 100 Ω			50	mV
R_L	Receiver differential input resistor		90	100	110	Ω

Table 12–10. 3.3-V PCML Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{ID}	Input differential voltage swing		300		600	mV
V_{ICM}	Input common mode voltage		1.5		3.465	V
V_{OD}	Output differential voltage		300	370	500	mV
ΔV_{OD}	Change in V_{OD} between high and low				50	mV
V_{OCM}	Output common mode voltage		2.5	2.85	3.3	V
ΔV_{OCM}	Change in V_{OCM} between high and low				50	mV
V_T	Output termination voltage			V_{CCIO}		V
R_1	Output external pull-up resistors		45	50	55	Ω
R_2	Output external pull-up resistors		45	50	55	Ω

Table 12–11. LVPECL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{ID}	Input differential voltage swing		300		1,000	mV
V_{ICM}	Input common mode voltage		1		2	V
V_{OD}	Output differential voltage	R_L = 100 Ω	525	700	970	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	1.5	1.7	1.9	V
R_L	Receiver differential input resistor		90	100	110	Ω

Table 12–12. HyperTransport Technology Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		2.375	2.5	2.625	V
V_{ID}	Input differential voltage swing		300		900	mV
V_{ICM}	Input common mode voltage		300		900	mV
V_{OD}	Output differential voltage	R_L = 100 Ω	380	485	820	mV
ΔV_{OD}	Change in V_{OD} between high and low	R_L = 100 Ω			50	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	440	650	780	mV
ΔV_{OCM}	Change in V_{OCM} between high and low	R_L = 100 Ω			50	mV
R_L	Receiver differential input resistor		90	100	110	Ω

Table 12–13. 3.3-V PCI Specifications

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
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 12–14. PCI-X 1.0 Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.0		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		0.35 × V_{CCIO}	V
V_{IPU}	Input pull-up voltage		0.7 × V_{CCIO}			V
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 12–15. GTL+ I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{TT}	Termination voltage		1.35	1.5	1.65	V
V_{REF}	Reference voltage		0.88	1.0	1.12	V
V_{IH}	High-level input voltage		V_{REF} + 0.1			V
V_{IL}	Low-level input voltage				V_{REF} – 0.1	V
V_{OL}	Low-level output voltage	I_{OL} = 34 mA *(1)*			0.65	V

Table 12–16. GTL I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{TT}	Termination voltage		1.14	1.2	1.26	V
V_{REF}	Reference voltage		0.74	0.8	0.86	V
V_{IH}	High-level input voltage		V_{REF} + 0.05			V
V_{IL}	Low-level input voltage				V_{REF} – 0.05	V
V_{OL}	Low-level output voltage	I_{OL} = 40 mA *(1)*			0.4	V

Table 12–17. SSTL-18 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.65	1.8	1.95	V
V_{REF}	Reference voltage		0.8	0.9	1.0	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.125			V
$V_{IL(DC)}$	Low-level DC input voltage				V_{REF} – 0.125	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.275			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.275	V
V_{OH}	High-level output voltage	I_{OH} = –6.7 mA *(1)*	V_{TT} + 0.475			V
V_{OL}	Low-level output voltage	I_{OL} = 6.7 mA *(1)*			V_{TT} – 0.475	V

Table 12–18. SSTL-18 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.65	1.8	1.95	V
V_{REF}	Reference voltage		0.8	0.9	1.0	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.125			V
$V_{IL(DC)}$	Low-level DC input voltage				V_{REF} – 0.125	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.275			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.275	V
V_{OH}	High-level output voltage	I_{OH} = –13.4 mA *(1)*	V_{TT} + 0.630			V
V_{OL}	Low-level output voltage	I_{OL} = 13.4 mA *(1)*			V_{TT} – 0.630	V

Table 12–19. 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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.18		3.0	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.35			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –8.1 mA *(1)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(1)*			V_{TT} – 0.57	V

Table 12–20. SSTL-2 Class II 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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.18		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.35			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –16.4 mA *(1)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(1)*			V_{TT} – 0.76	V

Table 12–21. 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 12–21. 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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.2	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.4			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.4	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA *(1)*	V_{TT} + 0.6			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA *(1)*			V_{TT} – 0.6	V

Table 12–22. 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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.2	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.4			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.4	V
V_{OH}	High-level output voltage	I_{OH} = –16 mA *(1)*	V_{TT} + 0.8			V
V_{OL}	Low-level output voltage	I_{OL} = 16 mA *(1)*			V_{TT} – 0.8	V

Table 12–23. 3.3-V AGP 2× Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.15	3.3	3.45	V
V_{REF}	Reference voltage		0.39 × V_{CCIO}		0.41 × V_{CCIO}	V
V_{IH}	High-level input voltage *(4)*		0.5 × V_{CCIO}		V_{CCIO} + 0.5	V

Table 12–23. 3.3-V AGP 2× Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{IL}	Low-level input voltage *(4)*				$0.3 \times V_{CCIO}$	V
V_{OH}	High-level output voltage	$I_{OUT} = –0.5$ mA	$0.9 \times V_{CCIO}$		3.6	V
V_{OL}	Low-level output voltage	$I_{OUT} = 1.5$ mA			$0.1 \times V_{CCIO}$	V

Table 12–24. 3.3-V AGP 1× Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.15	3.3	3.45	V
V_{IH}	High-level input voltage *(4)*		$0.5 \times V_{CCIO}$		$V_{CCIO} + 0.5$	V
V_{IL}	Low-level input voltage *(4)*				$0.3 \times V_{CCIO}$	V
V_{OH}	High-level output voltage	$I_{OUT} = –0.5$ mA	$0.9 \times V_{CCIO}$		3.6	V
V_{OL}	Low-level output voltage	$I_{OUT} = 1.5$ mA			$0.1 \times V_{CCIO}$	V

Table 12–25. 1.5-V HSTL Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.4	1.5	1.6	V
V_{REF}	Input reference voltage		0.68	0.75	0.9	V
V_{TT}	Termination voltage		0.7	0.75	0.8	V
V_{IH} (DC)	DC high-level input voltage		$V_{REF} + 0.1$			V
V_{IL} (DC)	DC low-level input voltage		–0.3		$V_{REF} – 0.1$	V
V_{IH} (AC)	AC high-level input voltage		$V_{REF} + 0.2$			V
V_{IL} (AC)	AC low-level input voltage				$V_{REF} – 0.2$	V
V_{OH}	High-level output voltage	$I_{OH} = 8$ mA *(1)*	$V_{CCIO} – 0.4$			V
V_{OL}	Low-level output voltage	$I_{OH} = –8$ mA *(1)*			0.4	V

Table 12–26. 1.5-V HSTL Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.4	1.5	1.6	V
V_{REF}	Input reference voltage		0.68	0.75	0.9	V
V_{TT}	Termination voltage		0.7	0.75	0.8	V
V_{IH} (DC)	DC high-level input voltage		V_{REF} + 0.1			V
V_{IL} (DC)	DC low-level input voltage		–0.3		V_{REF} – 0.1	V
V_{IH} (AC)	AC high-level input voltage		V_{REF} + 0.2			V
V_{IL} (AC)	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 16 mA *(1)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(1)*			0.4	V

Table 12–27. 1.8-V HSTL Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.65	1.80	1.95	V
V_{REF}	Input reference voltage		0.70	0.90	0.95	V
V_{TT}	Termination voltage			$V_{CCIO} \times 0.5$		V
V_{IH} (DC)	DC high-level input voltage		V_{REF} + 0.1			V
V_{IL} (DC)	DC low-level input voltage		–0.5		V_{REF} – 0.1	V
V_{IH} (AC)	AC high-level input voltage		V_{REF} + 0.2			V
V_{IL} (AC)	AC low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = 8 mA *(1)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OH} = –8 mA *(1)*			0.4	V

Table 12–28. 1.8-V HSTL Class II Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		1.65	1.80	1.95	V
V_{REF}	Input reference voltage		0.70	0.90	0.95	V

Table 12–28. 1.8-V HSTL Class II Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{TT}	Termination voltage			$V_{CCIO} \times 0.5$		V
V_{IH} (DC)	DC high-level input voltage		$V_{REF} + 0.1$			V
V_{IL} (DC)	DC low-level input voltage		–0.5		$V_{REF} – 0.1$	V
V_{IH} (AC)	AC high-level input voltage		$V_{REF} + 0.2$			V
V_{IL} (AC)	AC low-level input voltage				$V_{REF} – 0.2$	V
V_{OH}	High-level output voltage	I_{OH} = 16 mA *(1)*	$V_{CCIO} – 0.4$			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(1)*			0.4	V

Table 12–29. 1.5-V Differential HSTL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		1.4	1.5	1.6	V
V_{DIF} (DC)	DC input differential voltage		0.2			V
V_{CM} (DC)	DC common mode input voltage		0.68		0.9	V
V_{DIF} (AC)	AC differential input voltage		0.4			V

Table 12–30. CTT I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		2.05	3.3	3.6	V
V_{TT}/V_{REF}	Termination and input reference voltage		1.35	1.5	1.65	V
V_{IH}	High-level input voltage		$V_{REF} + 0.2$			V
V_{IL}	Low-level input voltage				$V_{REF} – 0.2$	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA	$V_{REF} + 0.4$			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA			$V_{REF} – 0.4$	V
I_O	Output leakage current (when output is high *Z*)	$GND \le V_{OUT} \le V_{CCIO}$	–10		10	µA

Table 12–31. 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)	25		30		50		70		µA
High sustaining current	$V_{IN} < V_{IH}$ (minimum)	–25		–30		–50		–70		µA
Low overdrive current	$0 V < V_{IN} < V_{CCIO}$		160		200		300		500	µA
High overdrive current	$0 V < V_{IN} < V_{CCIO}$		–160		–200		–300		–500	µA
Bus hold trip point		0.5	1.0	0.68	1.07	0.7	1.7	0.8	2.0	V

Table 12–32. Stratix Device Capacitance *Note (5)*

Symbol	Parameter	Minimum	Typical	Maximum	Unit
C_{IOTB}	Input capacitance on I/O pins in I/O banks 3, 4, 7, and 8.		11.5		pF
C_{IOLR}	Input capacitance on I/O pins in I/O banks 1, 2, 5, and 6, including high-speed differential receiver and transmitter pins.		8.2		pF
C_{CLKTB}	Input capacitance on top/bottom clock input pins: `CLK[4..7]` and `CLK[12..15]`.		11.5		pF
C_{CLKLR}	Input capacitance on left/right clock inputs: `CLK1`, `CLK3`, `CLK8`, `CLK10`.		7.8		pF
C_{CLKLR+}	Input capacitance on left/right clock inputs: `CLK0`, `CLK2`, `CLK9`, and `CLK11`.		4.4		pF

Notes to Tables 12–4 through 12–32:

(1) Drive strength is programmable according to values in the *Stratix Architecture* chapter of the *Stratix Device Handbook*.

(2) When `tx_outclock` port of `altlvds_tx` megafunction is 717 MHz, $V_{OD(min)}$ = 235 mV on the output clock pin.

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

(4) V_{REF} specifies the center point of the switching range.

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

Power Consumption

Altera offers two ways to calculate power for a design, the Altera® web power calculator and the power estimation feature in the Quartus® II software.

The interactive power calculator on the Altera web site is typically used prior to designing the FPGA in order to get a magnitude estimate of the device power. The Quartus II software power estimation feature allows designers to apply test vectors against their design for more accurate power consumption modeling.

In both cases, these calculations should only be used as an estimation of power, not as a specification.

Timing Closure

The timing numbers in Tables 12–34 to 12–43 are only provided as an indication of allowable timing for HardCopy Stratix devices. The Quartus II software provides preliminary timing information for HardCopy Stratix designs, which can be used as an estimation of the device performance.

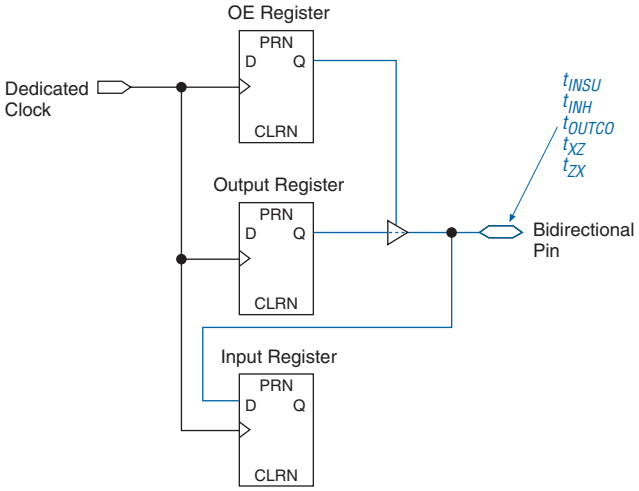
The final timing numbers and actual performance for each HardCopy Stratix design is available when the design migration is complete and are subject to verification and approval by Altera and the designer during the HardCopy Design review process.

For more information, refer to the *HardCopy Series Back-End Timing Closure* chapter in the *HardCopy Series Handbook*.

External Timing Parameters

External timing parameters are specified by device density and speed grade. Figure 12–1 shows the pin-to-pin timing model for bidirectional IOE pin timing. All registers are within the IOE.

Figure 12–1. External Timing in HardCopy Stratix Devices



All external timing parameters reported in this section are defined with respect to the dedicated clock pin as the starting point. All external I/O timing parameters shown are for 3.3-V LVTTL I/O standard with the 4-mA 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 the *Stratix Device Handbook*.

Table 12–33 shows the external I/O timing parameters when using global clock networks.

Table 12–33. HardCopy Stratix Global Clock External I/O Timing Parameters *Notes (1), (2)*

Symbol	Parameter
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
$t_{INSUPLL}$	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
$t_{OUTCOPLL}$	Clock-to-output delay output or bidirectional pin using IOE output register with global clock Enhanced PLL with default phase setting
t_{XZPLL}	Synchronous IOE output enable register to output pin disable delay using global clock fed by Enhanced PLL with default phase setting
t_{ZXPLL}	Synchronous IOE output enable register to output pin enable delay using global clock fed by Enhanced PLL with default phase setting

Notes to Table 12–33:
(1) These timing parameters are sample-tested only.
(2) These timing parameters are for column and row IOE pins. Designers should use the Quartus II software to verify the external timing for any pin.

HardCopy Stratix External I/O Timing

These timing parameters are for both column IOE and row IOE pins. In HC1S30 devices and above, designers can decrease the t_{SU} time by using the FPLLCLK, but may get positive hold time in HC1S60 and HC1S80 devices. Designers should use the Quartus II software to verify the external devices for any pin.

Tables 12–34 through 12–35 show the external timing parameters on column and row pins for HC1S25 devices.

Table 12–34. HC1S25 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.371		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.809	7.155	ns
t_{XZ}	2.749	7.040	ns
t_{ZX}	2.749	7.040	ns
$t_{INSUPLL}$	1.271		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.124	2.602	ns
t_{XZPLL}	1.064	2.487	ns
t_{ZXPLL}	1.064	2.487	ns

Table 12–35. HC1S25 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.665		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.834	7.194	ns
t_{XZ}	2.861	7.276	ns
t_{ZX}	2.861	7.276	ns
$t_{INSUPLL}$	1.538		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.164	2.653	ns
t_{XZPLL}	1.191	2.735	ns
t_{ZXPLL}	1.191	2.735	ns

Tables 12–36 through 12–37 show the external timing parameters on column and row pins for HC1S30 devices.

Table 12–36. HC1S30 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.935		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.814	7.274	ns
t_{XZ}	2.754	7.159	ns
t_{ZX}	2.754	7.159	ns
$t_{INSUPLL}$	1.265		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.068	2.423	ns
t_{XZPLL}	1.008	2.308	ns
t_{ZXPLL}	1.008	2.308	ns

Table 12–37. HC1S30 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.995		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.917	7.548	ns
t_{XZ}	2.944	7.630	ns
t_{ZX}	2.944	7.630	ns
$t_{INSUPLL}$	1.337		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.164	2.672	ns
t_{XZPLL}	1.191	2.754	ns
t_{ZXPLL}	1.191	2.754	ns

Tables 12–38 through 12–39 show the external timing parameters on column and row pins for HC1S40 devices.

Table 12–38. HC1S40 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.126		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.856	7.253	ns
t_{XZ}	2.796	7.138	ns
t_{ZX}	2.796	7.138	ns
$t_{INSUPLL}$	1.466		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.092	2.473	ns
t_{XZPLL}	1.032	2.358	ns
t_{ZXPLL}	1.032	2.358	ns

Table 12–39. HC1S40 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.020		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.912	7.480	ns
t_{XZ}	2.939	7.562	ns
t_{ZX}	2.939	7.562	ns
$t_{INSUPLL}$	1.370		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.144	2.693	ns
t_{XZPLL}	1.171	2.775	ns
t_{ZXPLL}	1.171	2.775	ns

Tables 12–40 through 12–41 show the external timing parameters on column and row pins for HC1S60 devices.

Table 12–40. HC1S60 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.000		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.051	6.977	ns
t_{XZ}	2.991	6.853	ns
t_{ZX}	2.991	6.853	ns
$t_{INSUPLL}$	1.315		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.029	2.323	ns
t_{XZPLL}	0.969	2.199	ns
t_{ZXPLL}	0.969	2.199	ns

Table 12–41. HC1S60 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.232		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.182	7.286	ns
t_{XZ}	3.209	7.354	ns
t_{ZX}	3.209	7.354	ns
$t_{INSUPLL}$	1.651		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.154	2.622	ns
t_{XZPLL}	1.181	2.690	ns
t_{ZXPLL}	1.181	2.690	ns

Tables 12–42 through 12–43 show the external timing parameters on column and row pins for HC1S80 devices.

Table 12–42. HC1S80 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	0.884		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.267	7.415	ns
t_{XZ}	3.207	7.291	ns
t_{ZX}	3.207	7.291	ns
$t_{INSUPLL}$	0.506		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.635	2.828	ns
t_{XZPLL}	1.575	2.704	ns
t_{ZXPLL}	1.575	2.704	ns

Table 12–43. HC1S80 External I/O Timing on Rows Using Pin Global Clock Networks

Symbol	Performance		Unit
	Min	Max	
t_{INSU}	1.362		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.457	7.859	ns
t_{XZ}	3.484	7.927	ns
t_{ZX}	3.484	7.927	ns
$t_{INSUPLL}$	0.994		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.821	3.254	ns
t_{XZPLL}	1.848	3.322	ns
t_{ZXPLL}	1.848	3.322	ns

Maximum Input & Output Clock Rates

Tables 12–44 through 12–46 show the maximum input clock rate for column and row pins in HardCopy Stratix devices.

Table 12–44. HardCopy Stratix Maximum Input Clock Rate for CLK[7..4] & CLK[15..12] Pins

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
GTL	300	MHz
GTL+	300	MHz
SSTL-3 class I	400	MHz
SSTL-3 class II	400	MHz
SSTL-2 class I	400	MHz
SSTL-2 class II	400	MHz
SSTL-18 class I	400	MHz
SSTL-18 class II	400	MHz
1.5-V HSTL class I	400	MHz
1.5-V HSTL class II	400	MHz
1.8-V HSTL class I	400	MHz
1.8-V HSTL class II	400	MHz
3.3-V PCI	422	MHz
3.3-V PCI-X 1.0	422	MHz
Compact PCI	422	MHz
AGP 1×	422	MHz
AGP 2×	422	MHz
CTT	300	MHz
Differential HSTL	400	MHz
LVPECL *(1)*	645	MHz
PCML *(1)*	300	MHz
LVDS *(1)*	645	MHz
HyperTransport technology *(1)*	500	MHz

Table 12–45. HardCopy Stratix Maximum Input Clock Rate for CLK[0, 2, 9, 11] Pins & FPLL[10..7]CLK Pins

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
GTL	300	MHz
GTL+	300	MHz
SSTL-3 class I	400	MHz
SSTL-3 class II	400	MHz
SSTL-2 class I	400	MHz
SSTL-2 class II	400	MHz
SSTL-18 class I	400	MHz
SSTL-18 class II	400	MHz
1.5-V HSTL class I	400	MHz
1.5-V HSTL class II	400	MHz
1.8-V HSTL class I	400	MHz
1.8-V HSTL class II	400	MHz
3.3-V PCI	422	MHz
3.3-V PCI-X 1.0	422	MHz
Compact PCI	422	MHz
AGP 1×	422	MHz
AGP 2×	422	MHz
CTT	300	MHz
Differential HSTL	400	MHz
LVPECL *(1)*	717	MHz
PCML *(1)*	400	MHz
LVDS *(1)*	717	MHz
HyperTransport technology *(1)*	717	MHz

Table 12–46. HardCopy Stratix Maximum Input Clock Rate for CLK[1, 3, 8, 10] Pins

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
GTL	300	MHz
GTL+	300	MHz
SSTL-3 class I	400	MHz
SSTL-3 class II	400	MHz
SSTL-2 class I	400	MHz
SSTL-2 class II	400	MHz
SSTL-18 class I	400	MHz
SSTL-18 class II	400	MHz
1.5-V HSTL class I	400	MHz
1.5-V HSTL class II	400	MHz
1.8-V HSTL class I	400	MHz
1.8-V HSTL class II	400	MHz
3.3-V PCI	422	MHz
3.3-V PCI-X 1.0	422	MHz
Compact PCI	422	MHz
AGP 1×	422	MHz
AGP 2×	422	MHz
CTT	300	MHz
Differential HSTL	400	MHz
LVPECL *(1)*	645	MHz
PCML *(1)*	300	MHz
LVDS *(1)*	645	MHz
HyperTransport technology *(1)*	500	MHz

Note to Tables 12–44 through 12–46:

(1) These parameters are only available on row I/O pins.

Tables 12–47 through 12–48 show the maximum output clock rate for column and row pins in HardCopy Stratix devices.

Table 12–47. HardCopy Stratix Maximum Output Clock Rate for PLL[5, 6, 11, 12] Pins (Part 1 of 2)

I/O Standard	Performance	Unit
LVTTL	350	MHz
2.5 V	350	MHz
1.8 V	250	MHz
1.5 V	225	MHz
LVCMOS	350	MHz
GTL	200	MHz
GTL+	200	MHz
SSTL-3 class I	200	MHz
SSTL-3 class II	200	MHz
SSTL-2 class I *(3)*	200	MHz
SSTL-2 class I *(4)*	200	MHz
SSTL-2 class I *(5)*	150	MHz
SSTL-2 class II *(3)*	200	MHz
SSTL-2 class II *(4)*	200	MHz
SSTL-2 class II *(5)*	150	MHz
SSTL-18 class I	150	MHz
SSTL-18 class II	150	MHz
1.5-V HSTL class I	250	MHz
1.5-V HSTL class II	225	MHz
1.8-V HSTL class I	250	MHz
1.8-V HSTL class II	225	MHz
3.3-V PCI	350	MHz
3.3-V PCI-X 1.0	350	MHz
Compact PCI	350	MHz
AGP 1×	350	MHz
AGP 2×	350	MHz
CTT	200	MHz
Differential HSTL	225	MHz
Differential SSTL-2 *(6)*	200	MHz
LVPECL *(2)*	500	MHz
PCML *(2)*	350	MHz

Table 12–47. HardCopy Stratix Maximum Output Clock Rate for PLL[5, 6, 11, 12] Pins (Part 2 of 2)

I/O Standard	Performance	Unit
LVDS *(2)*	500	MHz
HyperTransport technology *(2)*	350	MHz

Table 12–48. HardCopy Stratix Maximum Output Clock Rate (Using I/O Pins) for PLL[1, 2, 3, 4] Pins (Part 1 of 2)

I/O Standard	Performance	Unit
LVTTL	400	MHz
2.5 V	400	MHz
1.8 V	400	MHz
1.5 V	350	MHz
LVCMOS	400	MHz
GTL	200	MHz
GTL+	200	MHz
SSTL-3 class I	167	MHz
SSTL-3 class II	167	MHz
SSTL-2 class I	150	MHz
SSTL-2 class II	150	MHz
SSTL-18 class I	150	MHz
SSTL-18 class II	150	MHz
1.5-V HSTL class I	250	MHz
1.5-V HSTL class II	225	MHz
1.8-V HSTL class I	250	MHz
1.8-V HSTL class II	225	MHz
3.3-V PCI	250	MHz
3.3-V PCI-X 1.0	225	MHz
Compact PCI	400	MHz
AGP 1×	400	MHz
AGP 2×	400	MHz
CTT	300	MHz
Differential HSTL	225	MHz
LVPECL *(2)*	717	MHz
PCML *(2)*	420	MHz

Table 12–48. HardCopy Stratix Maximum Output Clock Rate (Using I/O Pins) for PLL[1, 2, 3, 4] Pins (Part 2 of 2)

I/O Standard	Performance	Unit
LVDS *(2)*	717	MHz
HyperTransport technology *(2)*	420	MHz

Notes to Tables 12–47 through 12–48:
(1) Differential SSTL-2 outputs are only available on column clock pins.
(2) These parameters are only available on row I/O pins.
(3) SSTL-2 in maximum drive strength condition.
(4) SSTL-2 in minimum drive strength with ≤10pF output load condition.
(5) SSTL-2 in minimum drive strength with > 10pF output load condition.
(6) Differential SSTL-2 outputs are only supported on column clock pins.

High-Speed I/O Specification

Table 12–49 provides high-speed timing specifications definitions.

Table 12–49. High-Speed Timing Specifications & Terminology

High-Speed Timing Specification	Terminology
t_C	High-speed receiver/transmitter input and output clock period.
f_{HSCLK}	High-speed receiver/transmitter input and output clock frequency.
t_{RISE}	Low-to-high transmission time.
t_{FALL}	High-to-low transmission time.
Timing unit interval (TUI)	The timing budget allowed for skew, propagation delays, and data sampling window. (TUI = 1/(Receiver Input Clock Frequency × Multiplication Factor) = t_C/w).
f_{HSDR}	Maximum LVDS data transfer rate (f_{HSDR} = 1/TUI).
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 to be captured correctly. The setup and hold times determine the ideal strobe position within the sampling window. SW = t_{SW} (max) – t_{SW} (min).
Input jitter (peak-to-peak)	Peak-to-peak input jitter on high-speed PLLs.
Output jitter (peak-to-peak)	Peak-to-peak output jitter on high-speed PLLs.
t_{DUTY}	Duty cycle on high-speed transmitter output clock.
t_{LOCK}	Lock time for high-speed transmitter and receiver PLLs.

Table 12–50 shows the high-speed I/O timing for HardCopy Stratix devices.

Table 12–50. High-Speed I/O Specifications (Part 1 of 2) *Notes (1), (2)*

Symbol	Conditions	Performance			Unit
		Min	Typ	Max	
f_{HSCLK} (Clock frequency) (LVDS, LVPECL, HyperTransport technology) $f_{HSCLK} = f_{HSDR} / W$	$W = 4$ to 30 (Serdes used)	10		210	MHz
	$W = 2$ (Serdes bypass)	50		231	MHz
	$W = 2$ (Serdes used)	150		420	MHz
	$W = 1$ (Serdes bypass)	100		462	MHz
	$W = 1$ (Serdes used)	300		717	MHz
f_{HSDR} Device operation (LVDS, LVPECL, HyperTransport technology)	$J = 10$	300		840	Mbps
	$J = 8$	300		840	Mbps
	$J = 7$	300		840	Mbps
	$J = 4$	300		840	Mbps
	$J = 2$	100		462	Mbps
	$J = 1$ (LVDS and LVPECL only)	100		462	Mbps
f_{HSCLK} (Clock frequency) (PCML) $f_{HSCLK} = f_{HSDR} / W$	$W = 4$ to 30 (Serdes used)	10		100	MHz
	$W = 2$ (Serdes bypass)	50		200	MHz
	$W = 2$ (Serdes used)	150		200	MHz
	$W = 1$ (Serdes bypass)	100		250	MHz
	$W = 1$ (Serdes used)	300		400	MHz
f_{HSDR} Device operation (PCML)	$J = 10$	300		400	Mbps
	$J = 8$	300		400	Mbps
	$J = 7$	300		400	Mbps
	$J = 4$	300		400	Mbps
	$J = 2$	100		400	Mbps
	$J = 1$	100		250	Mbps
TCCS	All			200	ps
SW	PCML ($J = 4, 7, 8, 10$)	750			ps
	PCML ($J = 2$)	900			ps
	PCML ($J = 1$)	1,500			ps
	LVDS and LVPECL ($J = 1$)	500			ps
	LVDS, LVPECL, HyperTransport technology ($J = 2$ through 10)	440			ps

Table 12–50. High-Speed I/O Specifications (Part 2 of 2) Notes (1), (2)

Symbol	Conditions	Performance			Unit
		Min	Typ	Max	
Input jitter tolerance (peak-to-peak)	All			250	ps
Output jitter (peak-to-peak)	All			160	ps
Output t_{RISE}	LVDS	80	110	120	ps
	HyperTransport technology	110	170	200	ps
	LVPECL	90	130	150	ps
	PCML	80	110	135	ps
Output t_{FALL}	LVDS	80	110	120	ps
	HyperTransport technology	110	170	200	ps
	LVPECL	90	130	160	ps
	PCML	105	140	175	ps
t_{DUTY}	LVDS (J = 2 through 10)	47.5	50	52.5	%
	LVDS (J =1) and LVPECL, PCML, HyperTransport technology	45	50	55	%
t_{LOCK}	All			100	µs

Notes to Table 12–50:
(1) When J = 4, 7, 8, and 10, the SERDES block is used.
(2) When J = 2 or J = 1, the SERDES is bypassed.

PLL Specifications

Table 12–51 describes the HardCopy Stratix device enhanced PLL specifications.

Table 12–51. Enhanced PLL Specifications (Part 1 of 3)

Symbol	Parameter	Min	Typ	Max	Unit
f_{IN}	Input clock frequency	3 (1)		684	MHz
f_{INDUTY}	Input clock duty cycle	40		60	%
$f_{EINDUTY}$	External feedback clock input duty cycle	40		60	%
$t_{INJITTER}$	Input clock period jitter			±200 (2)	ps
$t_{EINJITTER}$	External feedback clock period jitter			±200 (2)	ps
t_{FCOMP}	External feedback clock compensation time (3)			6	ns

Table 12–51. Enhanced PLL Specifications (Part 2 of 3)

Symbol	Parameter	Min	Typ	Max	Unit
f_{OUT}	Output frequency for internal global or regional clock	0.3		500	MHz
f_{OUT_EXT}	Output frequency for external clock *(2)*	0.3		526	MHz
$t_{OUTDUTY}$	Duty cycle for external clock output (when set to 50%)	45		55	%
t_{JITTER}	Period jitter for external clock output *(5)*			±100 ps for >200 MHz `outclk` ±20 mUI for <200 MHz `outclk`	ps or mUI
$t_{CONFIG5,6}$	Time required to reconfigure the scan chains for PLLs 5 and 6			$289/f_{SCANCLK}$	
$t_{CONFIG11,12}$	Time required to reconfigure the scan chains for PLLs 11 and 12			$193/f_{SCANCLK}$	
$t_{SCANCLK}$	`scanclk` frequency *(4)*			22	MHz
t_{DLOCK}	Time required to lock dynamically (after switchover or reconfiguring any non-post-scale counters/delays) *(6)*	*(8)*		100	µs
t_{LOCK}	Time required to lock from end of device configuration	10		400	µs
f_{VCO}	PLL internal VCO operating range	300		800 *(7)*	MHz
t_{LSKEW}	Clock skew between two external clock outputs driven by the same counter		±50		ps
t_{SKEW}	Clock skew between two external clock outputs driven by the different counters with the same settings		±75		ps
f_{SS}	Spread spectrum modulation frequency	30		150	kHz
% spread	Percentage spread for spread spectrum frequency *(9)*	0.4	0.5	0.6	%

Table 12–51. Enhanced PLL Specifications (Part 3 of 3)

Symbol	Parameter	Min	Typ	Max	Unit
t_{ARESET}	Minimum pulse width on `areset` signal	10			ns

Notes to Table 12–51:

(1) The minimum input clock frequency to the PFD (f_{IN}/N) must be at least 3 MHz for HardCopy Stratix device enhanced PLLs.
(2) Refer to "Maximum Input & Output Clock Rates".
(3) t_{FCOMP} can also equal 50% of the input clock period multiplied by the pre-scale divider *n* (whichever is less).
(4) This parameter is timing analyzed by the Quartus II software because the `scanclk` and `scandata` ports can be driven by the logic array.
(5) Actual jitter performance may vary based on the system configuration.
(6) Total required time to reconfigure and lock is equal to $t_{DLOCK} + t_{CONFIG}$. If only post-scale counters and delays are changed, then t_{DLOCK} is equal to 0.
(7) The VCO range is limited to 500 to 800 MHz when the spread spectrum feature is selected.
(8) Lock time is a function of PLL configuration and may be significantly faster depending on bandwidth settings or feedback counter change increment.
(9) Exact, user-controllable value depends on the PLL settings.
(10) The LOCK circuit on HardCopy Stratix PLLs does not work for industrial devices below –20 C unless the PFD frequency > 200 MHz. Refer to the *Stratix FPGA Errata Sheet* for more information on the PLL.

Table 12–52 describes the HardCopy Stratix device fast PLL specifications.

Table 12–52. Fast PLL Specifications

Symbol	Parameter	Min	Max	Unit
f_{IN}	CLKIN frequency (for *m* = 1) *(1)*, *(2)*	300	717	MHz
	CLKIN frequency (for *m* = 2 to 19)	300/*m*	1,000/*m*	MHz
	CLKIN frequency (for *m* = 20 to 32)	10	1,000/*m*	MHz
f_{OUT}	Output frequency for internal global or regional clock *(3)*	9.4	420	MHz
f_{OUT_EXT}	Output frequency for external clock *(2)*	9.375	717	MHz
f_{VCO}	VCO operating frequency	300	1,000	MHz
t_{INDUTY}	CLKIN duty cycle	40	60	%
$t_{INJITTER}$	Period jitter for CLKIN pin		±200	ps
t_{DUTY}	Duty cycle for DFFIO 1× CLKOUT pin *(4)*	45	55	%
t_{JITTER}	Period jitter for DFFIO clock out *(4)*		±80	ps
	Period jitter for internal global or regional clock		±100 ps for >200-MHz outclk ±20 mUI for <200-MHz outclk	ps or mUI
t_{LOCK}	Time required for PLL to acquire lock	10	100	µs
m	Multiplication factors for *m* counter *(4)*	1	32	Integer
*l*0, *l*1, *g*0	Multiplication factors for *l*0, *l*1, and *g*0 counter *(5)*, *(6)*	1	32	Integer
t_{ARESET}	Minimum pulse width on areset signal	10		ns

Notes to Table 12–52:

(1) Refer to "Maximum Input & Output Clock Rates" on page 12–23.
(2) PLLs 7, 8, 9, and 10 in the HC1S80 device support up to 717-MHz input and output.
(3) When using the SERDES, high-speed differential I/O mode supports a maximum output frequency of 210 MHz to the global or regional clocks (i.e., the maximum data rate 840 Mbps divided by the smallest SERDES J factor of 4).
(4) This parameter is for high-speed differential I/O mode only.
(5) These counters have a maximum of 32 if programmed for 50/50 duty cycle. Otherwise, they have a maximum of 16.
(6) High-speed differential I/O mode supports *W* = 1 to 16 and *J* = 4, 7, 8, or 10.

13. Quartus II Support for HardCopy Stratix Devices

H51014-3.1

Introduction

Altera® HardCopy® devices provide a comprehensive alternative to ASICs. HardCopy structured ASICs offer a complete solution from prototype to high-volume production, and maintain the powerful features and high-performance architecture of their equivalent FPGAs with the programmability removed. You can use the Quartus® II design software to design HardCopy devices in a manner similar to the traditional ASIC design flow and you can prototype with Altera's high density Stratix®, APEX™ 20KC, and APEX 20KE FPGAs before seamlessly migrating to the corresponding HardCopy device for high-volume production.

HardCopy structured ASICs provide the following key benefits:

- Improves performance, on the average, by 40% over the corresponding -6 speed grade FPGA device
- Lowers power consumption, on the average, by 40% over the corresponding FPGA
- Preserves the FPGA architecture and features, and minimizes risk
- Guarantees first-silicon success through a proven, seamless migration process from the FPGA to the equivalent HardCopy device
- Offers a quick turnaround of the FPGA design to a structured ASIC device—samples are available in about eight weeks

Altera's Quartus II software has built-in support for HardCopy Stratix devices. The HardCopy design flow in Quartus II software offers the following advantages:

- Unified design flow from prototype to production
- Performance estimation of the HardCopy Stratix device allows you to design systems for maximum throughput
- Easy-to-use and inexpensive design tools from a single vendor
- An integrated design methodology that enables system-on-a-chip designs

This chapter discusses the following areas:

- How to design HardCopy Stratix and HardCopy APEX structured ASICs using the Quartus II software
- An explanation of what the HARDCOPY_FPGA_PROTOTYPE devices are and how to target designs to these devices
- Performance and power estimation of HardCopy Stratix devices
- How to generate the HardCopy design database for submitting HardCopy Stratix and HardCopy APEX designs to the HardCopy Design Center

Features

Beginning with version 4.2, the Quartus II software contains several powerful features that facilitate design of HardCopy Stratix and HardCopy APEX devices:

- **HARDCOPY_FPGA_PROTOTYPE Devices**
  These are virtual Stratix FPGA devices with features identical to HardCopy Stratix devices. You must use these FPGA devices to prototype your designs and verify the functionality in silicon.

- **HardCopy Timing Optimization Wizard**
  Using this feature, you can target your design to HardCopy Stratix devices, providing an estimate of the design's performance in a HardCopy Stratix device.

- **HardCopy Stratix Floorplans and Timing Models**
  The Quartus II software supports post-migration HardCopy Stratix device floorplans and timing models and facilitates design optimization for design performance.

- **Placement Constraints**
  Location and LogicLock™ constraints are supported at the HardCopy Stratix floorplan level to improve overall performance.

- **Improved Timing Estimation**
  Beginning with version 4.2, the Quartus II software determines routing and associated buffer insertion for HardCopy Stratix designs, and provides the Timing Analyzer with more accurate information on the delays than was possible in previous versions of the Quartus II software. The Quartus II Archive File (**.qar**) automatically receives buffer insertion information, which greatly enhances the timing closure process in the back-end migration of your HardCopy Stratix device.

- **Design Assistant**
  This feature checks your design for compliance with all HardCopy device design rules and establishes a seamless migration path in the quickest time.

- **HardCopy Files Wizard**
  This wizard enables you to deliver to Altera the design database and all the deliverables required for migration. This feature is used for HardCopy Stratix and HardCopy APEX devices.

The HardCopy Stratix and HardCopy APEX PowerPlay Early Power Estimator is available on the Altera web site at **www.altera.com**.

HARDCOPY_FPGA_PROTOTYPE, HardCopy Stratix & Stratix Devices

You must use the HARDCOPY_FPGA_PROTOTYPE virtual devices available in Quartus II software to target your designs to the actual resources and package options available in the equivalent post-migration HardCopy Stratix device. The programming file generated for the HARDCOPY_FPGA_PROTOTYPE can be used in the corresponding Stratix FPGA device.

The purpose of the HARDCOPY_FPGA_PROTOTYPE is to guarantee seamless migration to HardCopy by making sure that your design only uses resources in the FPGA that can be used in the HardCopy device after migration. You can use the equivalent Stratix FPGAs to verify the design's functionality in-system, then generate the design database necessary to migrate to a HardCopy device. This process ensures the seamless migration of the design from a prototyping device to a production device in high volume. It also minimizes risk, assures samples in about eight weeks, and guarantees first-silicon success.

HARDCOPY_FPGA_PROTOTYPE devices are only available for HardCopy Stratix devices and are not available for the HardCopy II or HardCopy APEX device families.

Table 13–1 compares HARDCOPY_FPGA_PROTOTYPE devices, Stratix devices, and HardCopy Stratix devices.

Table 13–1. Qualitative Comparison of HARDCOPY_FPGA_PROTOTYPE to Stratix & HardCopy Stratix Devices (Part 1 of 2)

Stratix Device	HARDCOPY_FPGA_ PROTOTYPE Device	HardCopy Stratix Device
FPGA	Virtual FPGA	Structured ASIC
FPGA	Architecture identical to Stratix FPGA	Architecture identical to Stratix FPGA

Table 13–1. Qualitative Comparison of HARDCOPY_FPGA_PROTOTYPE to Stratix & HardCopy Stratix Devices (Part 2 of 2)

Stratix Device	HARDCOPY_FPGA_ PROTOTYPE Device	HardCopy Stratix Device
FPGA	Resources identical to HardCopy Stratix device	M-RAM resources different than Stratix FPGA in some devices
Ordered through Altera part number	Cannot be ordered, use the Altera Stratix FPGA part number	Ordered by Altera part number

Table 13–2 lists the resources available in each of the HardCopy Stratix devices.

Table 13–2. HardCopy Stratix Device Physical Resources

Device	LEs	ASIC Equivalent Gates (K) *(1)*	M512 Blocks	M4K Blocks	M-RAM Blocks	DSP Blocks	PLLs	Maximum User I/O Pins
HC1S25F672	25,660	250	224	138	2	10	6	473
HC1S30F780	32,470	325	295	171	2 *(2)*	12	6	597
HC1S40F780	41,250	410	384	183	2 *(2)*	14	6	615
HC1S60F1020	57,120	570	574	292	6	18	12	773
HC1S80F1020	79,040	800	767	364	6 *(2)*	22	12	773

Notes to Table 13–2:

(1) Combinational and registered logic do not include digital signal processing (DSP) blocks, on-chip RAM, or phase-locked loops (PLLs).

(2) The M-RAM resources for these HardCopy devices differ from the corresponding Stratix FPGA.

For a given device, the number of available M-RAM blocks in HardCopy Stratix devices is identical with the corresponding HARDCOPY_FPGA_PROTOTYPE devices, but may be different from the corresponding Stratix devices. Maintaining the identical resources between HARDCOPY_FPGA_PROTOTYPE and HardCopy Stratix devices facilitates seamless migration from the FPGA to the structured ASIC device.

For more information on HardCopy Stratix devices, refer to the *HardCopy Stratix Device Family Data Sheet* section in volume 1 of the *HardCopy Series Handbook.*

The three devices, Stratix FPGA, HARDCOPY_FPGA_PROTOTYPE, and HardCopy device, are distinct devices in the Quartus II software. The HARDCOPY_FPGA_PROTOTYPE programming files are used in the

Stratix FPGA for your design. The three devices are tied together with the same netlist, thus a single SRAM Object File (**.sof)** can be used to achieve the various goals at each stage. The same SRAM Object File is generated in the HARDCOPY_FPGA_PROTOTYPE design, and is used to program the Stratix FPGA device, the same way that it is used to generate the HardCopy Stratix device, guaranteeing a seamless migration.

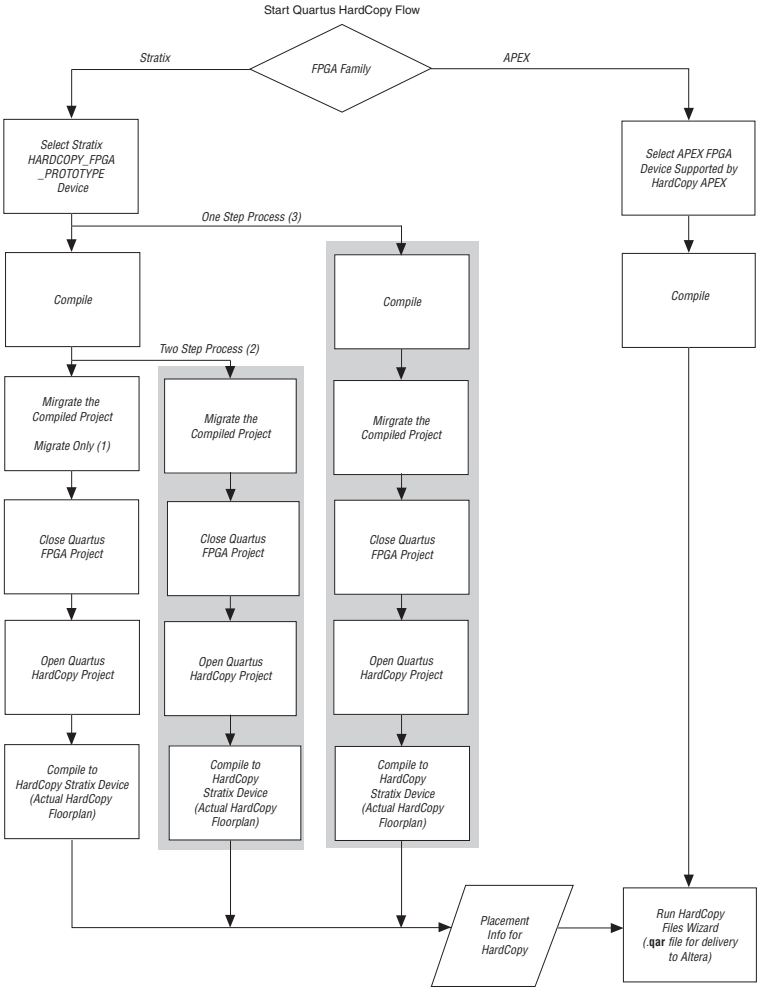
For more information on the **.sof** file and programming Stratix FPGA devices, refer to the *Programming and Configuration* chapter of the *Introduction to Quartus II Manual*.

HardCopy Design Flow

Figure 13–1 shows a HardCopy design flow diagram. The design steps are explained in detail in the following sections of this chapter. The HardCopy Stratix design flow utilizes the HardCopy Timing Optimization Wizard to automate the migration process into a one-step process. The remainder of this section explains the tasks performed by this automated process.

For a detailed description of the HardCopy Timing Optimization Wizard and HardCopy Files Wizard, refer to "HardCopy Timing Optimization Wizard Summary" and "Generating the HardCopy Design Database".

Figure 13–1. HardCopy Stratix & HardCopy APEX Design Flow Diagram



Notes to Figure 13–1:
(1) Migrate Only Process: The displayed flow is completed manually.
(2) Two Step Process: Migration and Compilation are done automatically (shaded area).
(3) One Step Process: Full HardCopy Compilation. The entire process is completed automatically (shaded area).

The Design Flow Steps of the One Step Process

The following sections describe each step of the full HardCopy compilation (the One Step Process), Figure 13–1.

Compile the Design for an FPGA

This step compiles the design for a HARDCOPY_FPGA_PROTOTYPE device and gives you the resource utilization and performance of the FPGA.

Migrate the Compiled Project

This step generates the Quartus II Project File (**.qpf**) and the other files required for HardCopy implementation. The Quartus II software also assigns the appropriate HardCopy Stratix device for the design migration.

Close the Quartus FPGA Project

Because you must compile the project for a HardCopy Stratix device, you must close the existing project which you have targeted to a HARDCOPY_FPGA_PROTOTYPE device.

Open the Quartus HardCopy Project

Open the Quartus II project that you created in the "Migrate the Compiled Project" step. The selected device is one of the devices from the HardCopy Stratix family that was assigned during that step.

Compile for HardCopy Stratix Device

Compile the design for a HardCopy Stratix device. After successful compilation, the Timing Analysis section of the compilation report shows the performance of the design implemented in the HardCopy device.

How to Design HardCopy Stratix Devices

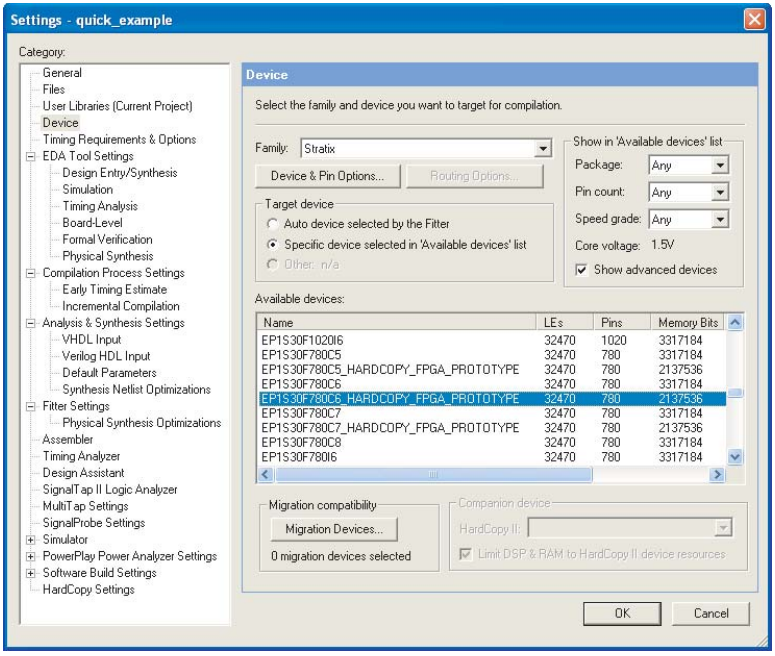
This section describes the process of designing for a HardCopy Stratix device using the HARDCOPY_FPGA_PROTOTYPE as your initial selected device. In order to use the HardCopy Timing Optimization Wizard, you must first design with the HARDCOPY_FPGA_PROTOTYPE in order for the design to migrate to a HardCopy Stratix device.

To target a design to a HardCopy Stratix device in the Quartus II software, follow these steps:

1. If you have not yet done so, create a new project or open an existing project.
2. Select **Device** (Assignments menu), then select **Stratix** in the **Family** list. Select the desired HARDCOPY_FPGA_PROTOTYPE device in the **Available Devices** list, as shown in Figure 13–2 on page 13–8.

By choosing the HARDCOPY_FPGA_PROTOTYPE device, all the design information, available resources, package option, and pin assignments are constrained to guarantee a seamless migration of your project to the HardCopy Stratix device. The netlist resulting from the HARDCOPY_FPGA_PROTOTYPE device compilation contains information about the electrical connectivity, resources used, I/O placements, and the unused resources in the FPGA device.

Figure 13–2. Selecting a HARDCOPY_FPGA_PROTOTYPE Device



3. Choose **Settings** (Assignments menu). In the **Category** list select **HardCopy Settings** and specify the input transition timing to be modeled for both clock and data input pins. These transition times are used in static timing analysis during back-end timing closure of the HardCopy device.

4. Add constraints to your HARDCOPY_FPGA_PROTOTYPE device and compile the design by choosing **Start Compilation** (Processing menu).

HardCopy Timing Optimization Wizard

After you have successfully compiled your design in the HARDCOPY_FPGA_PROTOTYPE, you must migrate the design to the HardCopy Stratix device to get a performance estimation of the HardCopy Stratix device. This migration is required before submitting the design to Altera for the HardCopy Stratix device implementation. To perform the required migration, choose the HardCopy Timing Optimization Wizard in the **HardCopy Utilities** (Project menu).

At this point, you are presented with the following three choices to target the designs to HardCopy Stratix devices, as shown in Figure 13–3.

- **Migration Only**—You can select this option after compiling the HARDCOPY_FPGA_PROTOTYPE project to migrate the project to a HardCopy Stratix project.

  You can now perform the following tasks manually to target the design to a HardCopy Stratix device. "Performance Estimation" on page 13–13 contains more information on how to perform these tasks.

  - Close the existing project
  - Open the migrated HardCopy Stratix project
  - Compile the HardCopy Stratix project for a HardCopy Stratix device

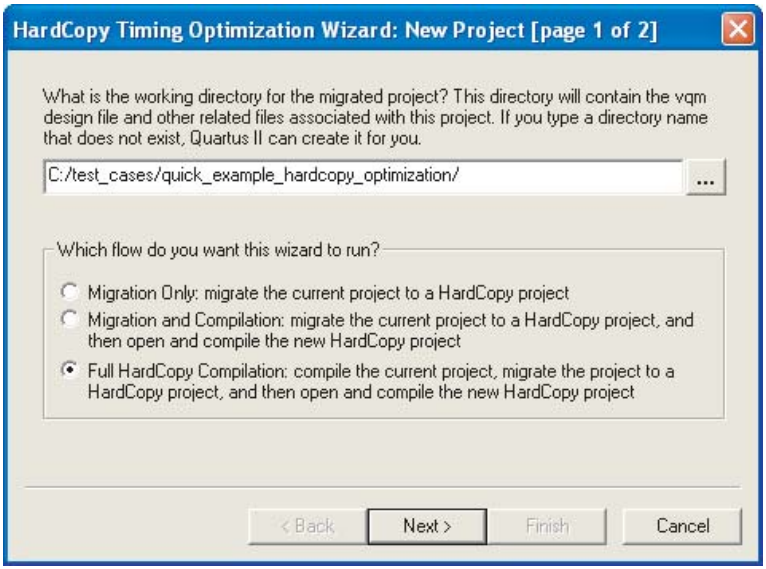
- **Migration and Compilation**—You can select this option after compiling the project. This option results in the following actions:

  - Migrating the project to a HardCopy Stratix project
  - Opening the migrated HardCopy Stratix project and compiling the project for a HardCopy Stratix device

- **Full HardCopy Compilation**—Selecting this option results in the following actions:

  - Compiling the existing HARDCOPY_FPGA_PROTOTYPE project
  - Migrating the project to a HardCopy Stratix project
  - Opening the migrated HardCopy Stratix project and compiling it for a HardCopy Stratix device

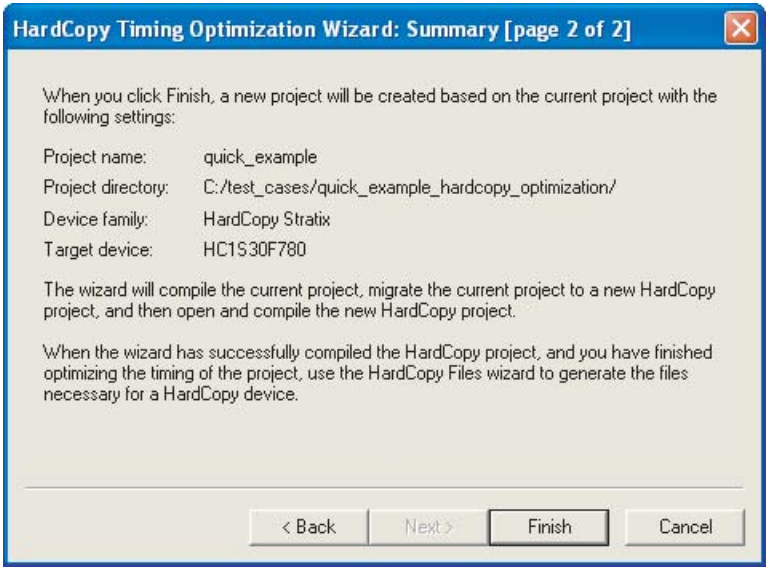
Figure 13–3. HardCopy Timing Optimization Wizard Options



The main benefit of the HardCopy Timing Wizard's three options is flexibility of the conversion process automation. The first time you migrate your HARDCOPY_FPGA_PROTOTYPE project to a HardCopy Stratix device, you may want to use Migration Only, and then work on the HardCopy Stratix project in the Quartus II software. As your prototype FPGA project and HardCopy Stratix project constraints stabilize and you have fewer changes, the Full HardCopy Compilation is ideal for one-click compiling of your HARDCOPY_FPGA_PROTOTYPE and HardCopy Stratix projects.

After selecting the wizard you want to run, the "HardCopy Timing Optimization Wizard: Summary" page shows you details about the settings you made in the Wizard (Figure 13–4).

Figure 13–4. HardCopy Timing Optimization Wizard Summary Page



When either of the second two options (Figure 13–3) are selected (**Migration and Compilation** or **Full HardCopy Compilation**), designs are targeted to HardCopy Stratix devices and optimized using the HardCopy Stratix placement and timing analysis to estimate performance. For details on the performance optimization and estimation steps, refer to "Performance Estimation". If the performance requirement is not met, you can modify your RTL source, optimize the FPGA design, and estimate timing until you reach timing closure.

Tcl Support for HardCopy Migration

To complement the GUI features for HardCopy migration, the Quartus II software provides the following command-line executables (which provide the tool command language (Tcl) shell to run the `--flow` Tcl command) to migrate the HARDCOPY_FPGA_PROTOTYPE project to HardCopy Stratix devices:

- `quartus_sh --flow migrate_to_hardcopy` <*project_name*> [`-c` <*revision*>] ↵

  This command migrates the project compiled for the HARDCOPY_FPGA_PROTOTYPE device to a HardCopy Stratix device.

- `quartus_sh --flow hardcopy_full_compile` <*project_name*> [`-c` <*revision*>] ↵

  This command performs the following tasks:

  - Compiles the exsisting project for a HARDCOPY_FPGA_PROTOTYPE device
  - Migrates the project to a HardCopy Stratix project
  - Opens the migrated HardCopy Stratix project and compiles it for a HardCopy Stratix device

Design Optimization & Performance Estimation

The HardCopy Timing Optimization Wizard creates the HardCopy Stratix project in the Quartus II software, where you can perform design optimization and performance estimation of your HardCopy Stratix device.

Design Optimization

Beginning with version 4.2, the Quartus II software supports HardCopy Stratix design optimization by providing floorplans for placement optimization and HardCopy Stratix timing models. These features enable you to refine placement of logic array blocks (LAB) and optimize the HardCopy design further than the FPGA performance. Customized routing and buffer insertion done in the Quartus II software are then used to estimate the design's performance in the migrated device. The HardCopy device floorplan, routing, and timing estimates in the Quartus II software reflect the actual placement of the design in the HardCopy Stratix device, and can be used to refer to the available resources, and the location of the resources in the actual device.

Performance Estimation

Figure 13–5 illustrates the design flow for estimating performance and optimizing your design. You can target your designs to HARDCOPY_FPGA_PROTOTYPE devices, migrate the design to the HardCopy Stratix device, and get placement optimization and timing estimation of your HardCopy Stratix device. In the event that the required performance is not met, you can:

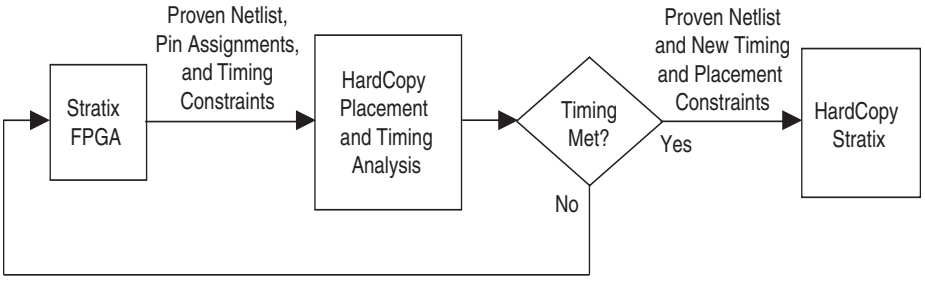
- Work to improve LAB placement in the HardCopy Stratix project.

or

- Go back to the HARDCOPY_FPGA_PROTOTYPE project and optimize that design, modify your RTL source code, repeat the migration to the HardCopy Stratix device, and perform the optimization and timing estimation steps.

On average, HardCopy Stratix devices are 40% faster than the equivalent -6 speed grade Stratix FPGA device. These performance numbers are highly design dependent, and you must obtain final performance numbers from Altera.

Figure 13–5. Obtaining a HardCopy Performance Estimation



To perform Timing Analysis for a HardCopy Stratix device, follow these steps:

1. Open an existing project compiled for a HARDCOPY_FPGA_PROTOYPE device.
2. Choose **HardCopy Utilities** > **HardCopy Timing Optimization Wizard** (Project menu).
3. Select a destination directory for the migrated project and complete the HardCopy Timing Optimization Wizard process.

On completion of the HardCopy Timing Optimization Wizard, the destination directory created contains the Quartus II project file, and all files required for HardCopy Stratix implementation. At this stage, the design is copied from the HARDCOPY_FPGA_PROTOTYPE project directory to a new directory to perform the timing analysis. This two-project directory structure enables you to move back and forth between the HARDCOPY_FPGA_PROTOTYPE design database and the HardCopy Stratix design database. The Quartus II software creates the <*project name*>**_hardcopy_optimization** directory.

You do not have to select the HardCopy Stratix device while performing a performance estimation. When you run the HardCopy Timing Optimization Wizard, the Quartus II software selects the HardCopy Stratix device corresponding to the specified HARDCOPY_FPGA_PROTOTYPE FPGA. Thus, the information necessary for the HardCopy Stratix device is available from the earlier HARDCOPY_FPGA_PROTOTYPE device selection.

All constraints related to the design are also transferred to the new project directory. You can modify these constraints, if necessary, in your optimized design environment to achieve the necessary timing closure. However, if the design is optimized at the HARDCOPY_FPGA_PROTOTYPE device level by modifying the RTL code or the device constraints, you must migrate the project with the HardCopy Timing Optimization Wizard.

If an existing project directory is selected when the HardCopy Timing Optimization Wizard is run, the existing information is overwritten with the new compile results.

The project directory is the directory that you chose for the migrated project. A snapshot of the files inside the <*project name*>**_hardcopy_optimization** directory is shown in Table 13–3.

Table 13–3. Directory Structure Generated by the HardCopy Timing Optimization Wizard

<*project name*>**_hardcopy_optimization**
- <*project name*>**.qsf**
- <*project name*>**.qpf**
- <*project name*>**.sof**
- <*project name*>**.macr**
- <*project name*>**.gclk**
- **db**
- **hardcopy_fpga_prototype**
  - fpga_<*project name*>**_violations.datasheet**
  - fpga_<*project name*>**_target.datasheet**
  - fpga_<*project name*>**_rba_pt_hcpy_v.tcl**
  - fpga_<*project name*>**_pt_hcpy_v.tcl**
  - fpga_<*project name*>**_hcpy_v.sdo**
  - fpga_<*project name*>**_hcpy.vo**
  - fpga_<*project name*>**_cpld.datasheet**
  - fpga_<*project name*>**_cksum.datasheet**
  - fpga_<*project name*>**.tan.rpt**
  - fpga_<*project name*>**.map.rpt**
  - fpga_<*project name*>**.map.atm**
  - fpga_<*project name*>**.fit.rpt**
  - fpga_<*project name*>**.db_info**
  - fpga_<*project name*>**.cmp.xml**
  - fpga_<*project name*>**.cmp.rcf**
  - fpga_<*project name*>**.cmp.atm**
  - fpga_<*project name*>**.asm.rpt**
  - fpga_<*project name*>**.qarlog**
  - fpga_<*project name*>**.qar**
  - fpga_<*project name*>**.qsf**
  - fpga_<*project name*>**.pin**
  - fpga_<*project name*>**.qpf**
- **db_export**
  - <*project name*>**.map.atm**
  - <*project name*>**.map.hdbx**
  - <*project name*>**.db_info**

4. Open the migrated Quartus II project created in Step 3.

5. Perform a full compilation.

   After successful compilation, the Timing Analysis section of the Compilation Report shows the performance of the design.

Performance estimation is not supported for HardCopy APEX devices in the Quartus II software. Your design can be optimized by modifying the RTL code or the FPGA design and the constraints. You should contact Altera to discuss any desired performance improvements with HardCopy APEX devices.

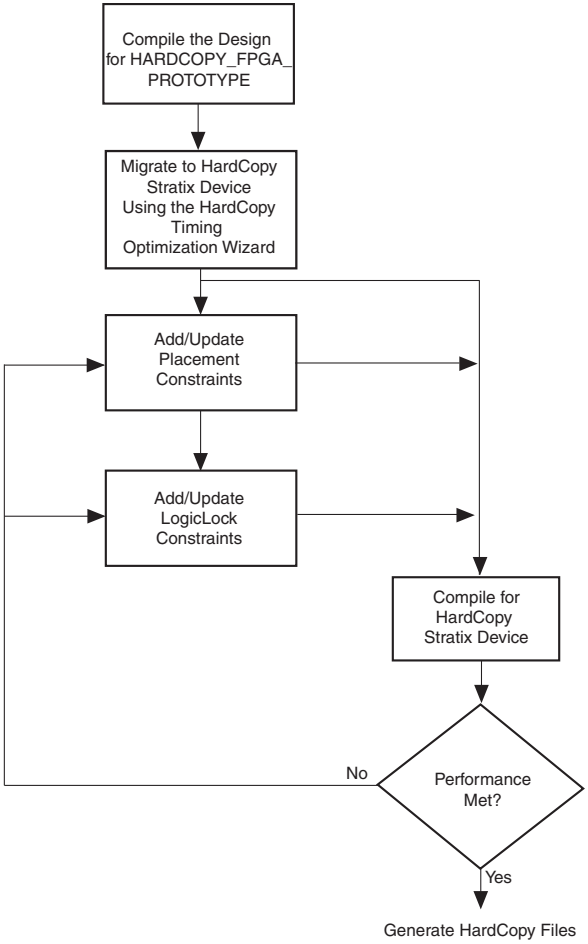
Buffer Insertion

Beginning with version 4.2, the Quartus II software provides improved HardCopy Stratix device timing closure and estimation, to more accurately reflect the results expected after back-end migration. The Quartus II software performs the necessary buffer insertion in your HardCopy Stratix device during the Fitter process, and stores the location of these buffers and necessary routing information in the Quartus II Archive File (**.qar)**. This buffer insertion improves the estimation of the Quartus II Timing Analyzer for the HardCopy Stratix device.

Placement Constraints

Beginning with version 4.2, the Quartus II software supports placement constraints and LogicLock regions for HardCopy Stratix devices. Figure 13–6 shows an iterative process to modify the placement constraints until the best placement for the HardCopy Stratix device is achieved.

Figure 13–6. Placement Constraints Flow for HardCopy Stratix Devices



Location Constraints

This section provides information on HardCopy Stratix logic location constraints.

LAB Assignments

Logic placement in HardCopy Stratix is limited to LAB placement and optimization of the interconnecting signals between them. In a Stratix FPGA, individual logic elements (LE) are placed by the Quartus II Fitter into LABs. The HardCopy Stratix migration process requires that LAB contents cannot change after the Timing Optimization Wizard task is

done. Therefore you can only make LAB-level placement optimization and location assignments after migrating the HARDCOPY_FPGA_PROTOTYPE project to the HardCopy Stratix device.

The Quartus II software supports these LAB location constraints for HardCopy Stratix devices. The entire contents of a LAB is moved to an empty LAB when using LAB location assignments. If you want to move the logic contents of LAB A to LAB B, the entire contents of LAB A are moved to an empty LAB B. For example, the logic contents of LAB_X33_Y65 can be moved to an empty LAB at LAB_X43_Y56 but individual logic cell LC_X33_Y65_N1 can not be moved by itself in the HardCopy Stratix Timing Closure Floorplan.

LogicLock Assignments

The LogicLock feature of the Quartus II software provides a block-based design approach. Using this technique you can partition your design and create each block of logic independently, optimize placement and area, and integrate all blocks into the top level design.

To learn more about this methodology, refer to the *LogicLock Design Methodology* chapter in volume 2 of the *Quartus II Handbook*.

LogicLock constraints are supported when you migrate the project from a HARDCOPY_FPGA_PROTOTYPE project to a HardCopy Stratix project. If the LogicLock region was specified as "Size=Fixed" and "Location=Locked" in the HARDCOPY_FPGA_PROTOTYPE project, it is converted to have "Size=Auto" and "Location=Floating" as shown in the following LogicLock examples. This modification is necessary because the floorplan of a HardCopy Stratix device is different from that of the Stratix device, and the assigned coordinates in the HARDCOPY_FPGA_PROTOTYPE do not match the HardCopy Stratix floorplan. If this modification did not occur, LogicLock assignments would lead to incorrect placement in the Quartus II Fitter. Making the regions auto-size and floating, maintains your LogicLock assignments, allowing you to easily adjust the LogicLock regions as required and lock their locations again after HardCopy Stratix placement.

The following are two examples of LogicLock assignments.

LogicLock Region Definition in the HARDCOPY_FPGA_PROTOTYPE .qsf File

```
set_global_assignment -name LL_HEIGHT 15 -entity risc8 -section_id test
set_global_assignment -name LL_WIDTH 15 -entity risc8 -section_id test
set_global_assignment -name LL_STATE LOCKED -entity risc8 -section_id test
set_global_assignment -name LL_AUTO_SIZE OFF -entity risc8 -section_id test
```

LogicLock Region Definition in the Migrated HardCopy Stratix .qsf File

```
set_global_assignment -name LL_HEIGHT 15 -entity risc8 -section_id test
set_global_assignment -name LL_WIDTH 15 -entity risc8 -section_id test
set_global_assignment -name LL_STATE FLOATING -entity risc8 -section_id test
set_global_assignment -name LL_AUTO_SIZE ON -entity risc8 -section_id test
```

Checking Designs for HardCopy Design Guidelines

When you develop a design with HardCopy migration in mind, you must follow Altera-recommended design practices that ensure a straightforward migration process or the design will not be able to be implemented in a HardCopy device. Prior to starting migration of the design to a HardCopy device, you must review the design and identify and address all the design issues. Any design issues that have not been addressed can jeopardize silicon success.

Altera Recommended HDL Coding Guidelines

Designing for Altera PLD, FPGA, and HardCopy structured ASIC devices requires certain specific design guidelines and hardware description language (HDL) coding style recommendations be followed.

For more information on design recommendations and HDL coding styles, refer to the *Design Guidelines* Section in volume 1 of the *Quartus II Handbook*.

Design Assistant

The Quartus II software includes the Design Assistant feature to check your design against the HardCopy design guidelines. Some of the design rule checks performed by the Design Assistant include the following rules:

- Design should not contain combinational loops
- Design should not contain delay chains
- Design should not contain latches

To use the Design Assistant, you must have run Analysis and Synthesis on the design in the Quartus II software. Altera recommends that you run the **Design Assistant** to check for compliance with the HardCopy design guidelines early in the design process and after every compilation.

Design Assistant Settings

You must select the design rules in the **Design Assistant** page of the **Settings** dialog box (Assignments menu) prior to running the design. In this dialog box, you can choose whether to run the Design Assistant during compilation. Altera recommends enabling this feature to run the Design Assistant automatically during compilation of your design.

Running Design Assistant

To run Design Assistant independently of other Quartus II features, choose **Start > Start Design Assistant** (Processing menu).

The Design Assistant automatically runs in the background of the Quartus II software when the HardCopy Timing Optimization Wizard is launched, and does not display the Design Assistant results immediately to the display. The design is checked before the Quartus II software migrates the design and creates a new project directory for performing timing analysis.

Also, the Design Assistant runs automatically whenever you generate the HardCopy design database with the HardCopy Files Wizard. The Design Assistant report generated is used by the Altera HardCopy Design Center to review your design.

Reports & Summary

The results of running the Design Assistant on your design are available in the Design Assistant Results section of the Compilation Report. The Design Assistant also generates the summary report in the <*project name*>**hardcopy** subdirectory of the project directory. This report file is titled <*project name*>**_violations.datasheet**. Reports include the settings, run summary, results summary, and details of the results and messages. The Design Assistant report indicates the rule name, severity of the violation and the circuit path where any violation occurred.

To learn about the design rules and standard design practices to comply with HardCopy design rules, refer to the Quartus II Help and the *HardCopy Series Design Guidelines* chapter in Volume 1 of the *HardCopy Series Handbook*.

Generating the HardCopy Design Database

You can use the HardCopy Files Wizard to generate the complete set of deliverables required for migrating the design to a HardCopy device in a single click. The HardCopy Files Wizard asks questions related to the design and archives your design, settings, results, and database files for delivery to Altera. Your responses to the design details are stored in *<project name>*_**hardcopy_optimization***<project name>*.**hps.txt**.

You can generate the archive of the HardCopy design database only after compiling the design to a HardCopy Stratix device. The Quartus II Archive File (**.qar**) is generated at the same directory level as the targeted project, either before or after optimization. Table 13–4 shows the archive directory structure and files collected by the HardCopy Files Wizard.

Table 13–4. HardCopy Stratix Design Files Collected by the HardCopy Files Wizard

<*project name*>**_hardcopy_optimization**
- <*project name*>**.flow.rpt**
- <*project name*>**.qpf**
- <*project name*>**.asm.rpt**
- <*project name*>**.blf**
- <*project name*>**.fit.rpt**
- <*project name*>**.gclk**
- <*project name*>**.hps.txt**
- <*project name*>**.macr**
- <*project name*>**.pin**
- <*project name*>**.qsf**
- <*project name*>**.sof**
- <*project name*>**.tan.rpt**

hardcopy
- <*project name*>**.apc**
- <*project name*>**_cksum.datasheet**
- <*project name*>**_cpld.datasheet**
- <*project name*>**_hcpy.vo**
- <*project name*>**_hcpy_v.sdo**
- <*project name*>**_pt_hcpy_v.tcl**
- <*project name*>**_rba_pt_hcpy_v.tcl**
- <*project name*>**_target.datasheet**
- <*project name*>**_violations.datasheet**

hardcopy_fpga_prototype
- fpga_<*project name*>**.asm.rpt**
- fpga_<*project name*>**.cmp.rcf**
- fpga_<*project name*>**.cmp.xml**
- fpga_<*project name*>**.db_info**
- fpga_<*project name*>**.fit.rpt**
- fpga_<*project name*>**.map.atm**
- fpga_<*project name*>**.map.rpt**
- fpga_<*project name*>**.pin**
- fpga_<*project name*>**.qsf**
- fpga_<*project name*>**.tan.rpt**
- fpga_<*project name*>**_cksum.datasheet**
- fpga_<*project name*>**_cpld.datasheet**
- fpga_<*project name*>**_hcpy.vo**
- fpga_<*project name*>**_hcpy_v.sdo**
- fpga_<*project name*>**_pt_hcpy_v.tcl**
- fpga_<*project name*>**_rba_pt_hcpy_v.tcl**
- fpga_<*project name*>**_target.datasheet**
- fpga_<*project name*>**_violations.datasheet**

db_export
- <*project name*>**.db_info**
- <*project name*>**.map.atm**
- <*project name*>**.map.hdbx**

The Design Assistant automatically runs when the HardCopy Files Wizard is started.

After creating the migration database with the HardCopy Timing Optimization Wizard, you must compile the design before generating the project archive. You will receive an error if you create the archive before compiling the design.

Static Timing Analysis (STA)

In addition to performing timing analysis, the Quartus II software also provides all of the requisite netlists and Tcl scripts to perform static timing analysis (STA) using the Synopsys STA tool, PrimeTime. The following files, necessary for timing analysis with the PrimeTime tool, are generated by the HardCopy Files Wizard:

- <*project name*>**_hcpy.vo**—Verilog HDL output format
- <*project name*>**_hpcy_v.sdo**—Standard Delay Format Output File (SDF)
- <*project name*>**_pt_hcpy_v.tcl**—Tcl script

These files are available in the <*project name*>**\hardcopy** directory. PrimeTime libraries for the HardCopy Stratix and Stratix devices are included with the Quartus II software.

Use the HardCopy Stratix libraries for PrimeTime to perform STA during timing analysis of designs targeted to HARDCOPY_FPGA_PROTOTYPE device.

For more information on static timing analysis, refer to the *Quartus II Timing Analysis* and the *Synopsys PrimeTime Support* chapters in volume 3 of the *Quartus II Handbook*.

Early Power Estimation

You can use PowerPlay early power estimator to estimate the amount of power your HardCopy Stratix or HardCopy APEX device consumes. This tool is available on the Altera web site. Using the Early Power Estimator requires some knowledge of your design resources and specifications, including:

- Target device and package
- Clock networks used in the design
- Resource usage for logic elements (LEs), digital signal processing (DSP) blocks, phase-locked loops (PLL), and RAM blocks
- High speed differential interfaces (HSDI), general I/O power consumption requirements, and pin counts
- Environmental and thermal conditions

HardCopy Stratix Early Power Estimation

The PowerPlay Early Power Estimator provides an initial estimate of I_{CC} for any HardCopy Stratix device based on typical conditions. This calculation saves significant time and effort in gaining a quick understanding of the power requirements for the device. No stimulus vectors are necessary for power estimation, which is established by the clock frequency and toggle rate in each clock domain.

This calculation should only be used as an estimation of power, not as a specification. The actual I_{CC} should be verified during operation because this estimate is sensitive to the actual logic in the device and the environmental operating conditions.

For more information on simulation-based power estimation, refer to the *Power Estimation & Analysis* Section in volume 3 of the *Quartus II Handbook*.

On average, HardCopy Stratix devices are expected to consume 40% less power than the equivalent FPGA.

HardCopy APEX Early Power Estimation

The PowerPlay Early Power Estimator can be run from the Altera web site in the device support section (**http://www.altera.com/support/devices/dvs-index.html**). You cannot open this feature in the Quartus II software.

With the HardCopy APEX PowerPlay Early Power Estimator, you can estimate the power consumed by HardCopy APEX devices and design systems with the appropriate power budget. Refer to the web page for instructions on using the HardCopy APEX PowerPlay early power estimator.

HardCopy APEX devices are generally expected to consume about 40% less power than the equivalent APEX 20KE or APEX 20KC FPGA devices.

Tcl Support for HardCopy Stratix

The Quartus II software also supports the HardCopy Stratix design flow at the command prompt using Tcl scripts.

For details on Quartus II support for Tcl scripting, refer to the *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*.

Targeting Designs to HardCopy APEX Devices

Beginning with version 4.2, the Quartus II software supports targeting designs to HardCopy APEX device families. After compiling your design for one of the APEX 20KC or APEX 20KE FPGA devices supported by a HardCopy APEX device, run the HardCopy Files Wizard to generate the necessary set of files for HardCopy migration.

The HardCopy APEX device requires a different set of design files for migration than HardCopy Stratix. Table 13–5 shows the files collected for HardCopy APEX by the HardCopy Files Wizard.

Table 13–5. HardCopy APEX Files Collected by the HardCopy Files Wizard

<*project name*>**.tan.rpt**
<*project name*>**.asm.rpt**
<*project name*>**.fit.rpt**
<*project name*>**.hps.txt**
<*project name*>**.map.rpt**
<*project name*>**.pin**
<*project name*>**.sof**
<*project name*>**.qsf**
<*project name*>**_cksum.datasheet**
<*project name*>**_cpld.datasheet**
<*project name*>**_hcpy.vo**
<*project name*>**_hcpy_v.sdo**
<*project name*>**_pt_hcpy_v.tcl**
<*project name*>**_rba_pt_hcpy_v.tcl**
<*project name*>**_target.datasheet**
<*project name*>**_violations.datasheet**

Refer to "Generating the HardCopy Design Database" for information on generating the complete set of deliverables required for migrating the design to a HardCopy APEX device. After you have successfully run the HardCopy Files Wizard, you can submit your design archive to Altera to implement your design in a HardCopy device. You should contact Altera for more information on this process.

Conclusion

The methodology for designing HardCopy Stratix devices using the Quartus II software is the same as that for designing the Stratix FPGA equivalent. You can use the familiar Quartus II software tools and design flow, target designs to HardCopy Stratix devices, optimize designs for higher performance and lower power consumption than the Stratix FPGAs, and deliver the design database for migration to a HardCopy Stratix device. Compatible APEX FPGA designs can migrate to HardCopy APEX after compilation using the HardCopy Files Wizard to archive the design files. Submit the files to the HardCopy Design Center to complete the back-end migration.

Related Documents

For more information, refer to the following documentation:

- The *HardCopy Series Design Guidelines* chapter in volume 1 of the *HardCopy Series Handbook.*
- The *HardCopy Series Back-End Timing Closure* chapter in volume 1 of the *HardCopy Series Handbook.*

14. Design Guidelines for HardCopy Stratix Performance Improvement

H51027-1.1

Introduction

Advanced design techniques using Altera® HardCopy® Stratix® devices can yield tremendous performance improvements over the design implemented in a Stratix FPGA device. After you verify your Stratix FPGA design in system operation and are ready to migrate to a HardCopy Stratix device, additional device performance is possible through the migration. This chapter focuses on Quartus® II software advanced design techniques that apply to both Stratix FPGA devices and HardCopy Stratix devices. Use these techniques to increase your maximum clock frequency, improve input and output pin timing, and improve timing closure in HardCopy Stratix designs.

Every design is different. The techniques described in this chapter may not apply to every design, and may not yield the same level of improvement.

This document discusses the following topics:

- Planning Stratix FPGA design for HardCopy Stratix design conversion
- Using LogicLock™ regions in HardCopy Stratix designs
- Using Design Space Explorer (DSE) on HardCopy Stratix designs
- Design performance improvement example

Background Information

To understand the Quartus II software and device architecture, and to use the advanced design techniques described in this chapter, Altera recommends reading the *HardCopy Series Handbook* and the following chapters in the *Quartus II Software Handbook v5.0*:

- *Design Recommendations for Altera Devices*
- *Design Optimization for Altera Devices*
- *Design Space Explorer*
- *LogicLock Design Methodology*
- *Netlist Optimizations & Physical Synthesis*

Planning Stratix FPGA Design for HardCopy Stratix Design Conversion

In order to achieve greater performance improvement in your HardCopy Stratix device, additional Quartus II software constraints and placement techniques in the HARDCOPY_FPGA_PROTOTYPE design project may be necessary. This does not mean changing the source hardware description language (HDL) code or functionality, but providing additional constraints in the Quartus II software that specifically impact HardCopy Stratix timing optimization.

Planning ahead for migration to the HardCopy design, while still modifying the HARDCOPY_FPGA_PROTOTYPE design, can improve design performance results. You must anticipate how portions of your FPGA design are placed and connected in the HardCopy device floorplan. The HardCopy device floorplan is smaller than the FPGA device floorplan, allowing use of the customized metal routing in HardCopy Stratix devices.

Partitioning Your Design

Partitioning your design into functional blocks is essential in multi-million gate designs. With a HardCopy Stratix device, you can implement approximately one million ASIC gates of logic. Therefore, Altera recommends hierarchical-design partitioning based on system functions.

When using a hierarchical- or incremental-design methodology, you must consider how your design is partitioned to achieve good results. Altera recommends the following practices for partitioning designs as documented in the *Design Recommendations for Altera Devices* chapter in Volume 1 of the *Quartus II Development Software Handbook v5.0*:

- Partition your design at functional boundaries.
- Minimize the I/O connections between different partitions.
- Register all inputs and outputs of each block. This makes logic synchronous and avoids glitches and any delay penalty on signals that cross between partitions. Registering I/O pins typically eliminates the need to specify timing requirements for signals that connect between different blocks.
- Do not use glue logic or connection logic between hierarchical blocks. When you preserve hierarchy boundaries, glue logic is not merged with hierarchical blocks. Your synthesis software may optimize glue logic separately, which can degrade synthesis results and is not efficient when used with the LogicLock design methodology.
- Logic is not synthesized or optimized across partition boundaries. Any constant values (for example, signals set to GND), are not propagated across partitions.

- Do not use tri-state signals or bidirectional ports on hierarchical boundaries. If you use tri-state boundaries in a lower-level block, synthesis pushes the tri-state signals through the hierarchy to the top-level. This takes advantage of the tri-state drivers on the output pins of the Altera device. Since this requires optimizing through hierarchies, lower-level boundary tri-state signals are not supported with a block-level design methodology.
- Limit clocks to one per block. Partitioning your design into clock domains makes synthesis and timing analysis easier.
- Place state machines in separate blocks to speed optimization and provide greater encoding control.
- Separate timing-critical functions from non-timing-critical functions.
- Limit the critical timing path to one hierarchical block. Group the logic from several design blocks to ensure the critical path resides in one block.

These guidelines apply to all Altera device architectures including HardCopy Stratix devices. Partitioning functional boundaries to have all outputs immediately registered is crucial to using LogicLock regions effectively in HardCopy devices. With registered outputs, you allow the signals to leave a function block at the start of the clock period. This gives the signals more set-up time to reach their endpoints in the clock period. In large designs that are partitioned into multiple function blocks, the block-to-block interconnects are often the limiting factor for f_{MAX} performance. Registered outputs give the Quartus II Fitter the optimal place-and-route flexibility for interconnects between major function blocks.

Physical Synthesis Optimization

All physical synthesis settings in the Quartus II software can be used in the HARDCOPY_FPGA_PROTOTYPE design. These settings are found in the **Physical Synthesis Optimizations** section of the **Fitter Settings** dialog box (Assignments menu) and include the following settings:

- Physical synthesis for combinational logic
- Register duplication
- Register retiming

These settings can improve FPGA performance while developing the HARDCOPY_FPGA_PROTOTYPE. All modifications are passed along into the HardCopy Stratix project when you run the HardCopy Timing Optimization wizard. After running the HardCopy Timing Optimization wizard and subsequently opening the HardCopy project in the Quartus II software, these physical synthesis optimizations are disabled. No further modifications to the netlist are made.

Altera recommends physical synthesis optimizations for the HARDCOPY_FPGA_PROTOTYPE. The work done in the prototype enhances performance in the HardCopy Stratix device after migration. Duplicating combinational logic and registers can increase area utilization, which limits placement flexibility when designs exceed 95% Logic Element (LE) utilization. However, duplicating combinational logic and registers can help with performance by allowing critical paths to be duplicated when their endpoints must reach different areas of the device floorplan.

For more information on netlist and design optimization, refer to *Section III. Area Optimization & Timing Closure* in Volume 2 of the *Quartus II Development Software Handbook v5.0*.

Using LogicLock Regions in HardCopy Stratix Designs

Create LogicLock regions in the HARDCOPY_FPGA_PROTOTYPE project and migrate the regions into the HardCopy Stratix optimization project using the Quartus II software. LogicLock regions can provide significant benefits in design performance by carefully isolating critical blocks of logic, including:

- MegaCore® IP functions
- I/O interfaces
- Reset or other critical logic feeding global clock lines
- Partitioned function blocks

You must compile your design initially without LogicLock regions present and review the timing analysis reports to determine if additional constraints or LogicLock regions are necessary. This process allows you to determine which function blocks or data paths require LogicLock regions.

Create LogicLock regions in the HARDCOPY_FPGA_PROTOTYPE design project in the Quartus II software. This transfers the LogicLock regions to the HardCopy design project after the **HardCopy Timing Optimization Wizard** is run. Although the Quartus II software transfers the contents of the LogicLock region, the area, location, and soft boundary settings revert to their default settings in the HardCopy project immediately after the **HardCopy Timing Optimization Wizard** is run.

If you are using LogicLock regions, Altera recommends you use the **Migration Only** setting in the **HardCopy Timing Optimization Wizard** to create the HardCopy design project. You should not compile your design automatically using the **Full Compilation** or **Migrate and Compile** options in the wizard. Open the HardCopy design project and verify that the LogicLock region properties meet your desired settings before compiling the HardCopy optimization project. LogicLock soft

regions are turned on by default in the HardCopy Stratix design. While this does allow the Fitter to place all logic in your design with fewer restrictions, it is not optimal for performance improvement in the HardCopy Stratix design.

Recommended LogicLock Settings for HardCopy Stratix Designs

Altera recommends the following LogicLock region settings for the HARDCOPY_FPGA_PROTOTYPE:

- Turn on **Reserve Unused Logic**
- Turn off **Soft Region**
- Select either **Auto** or **Fixed** as the **Size** (design-dependent)
- Select either **Floating** or **Locked** as the **Location** (design-dependent)

When using the **Reserve Unused Logic** setting in a design with high resource utilization (> 95% LE utilization), and a large number of LogicLock regions, the design may not fit in the device. Turning off **Reserve Unused Logic** in less critical LogicLock regions can help Fitter placement. The LEs allowed to float in placement and be packed into unused LEs of LogicLock regions may not be placed optimally after migration to the HardCopy Stratix device since they are merged with other LogicLock regions.

After running the **HardCopy Timing Optimization Wizard**, the LogicLock region properties are reset to their default conditions. This allows a successful and immediate placement of your design in the Quartus II software. You can further refine the LogicLock region properties for additional benefits.

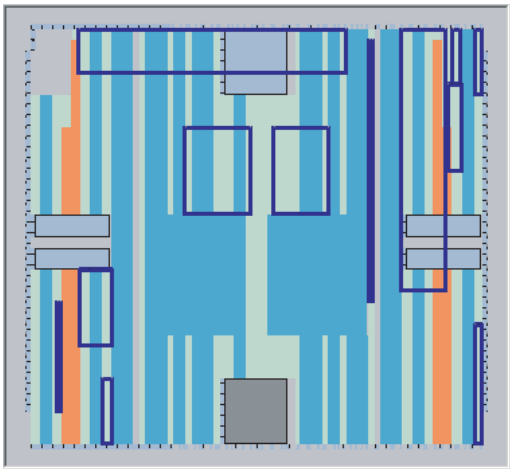
Altera recommends using the following properties for LogicLock regions in the HardCopy design project:

- Turn off **Soft Region**
- Select either **Auto** or **Fixed** as the **Size** after you are satisfied with the placement and timing result of a LogicLock region in a successful HardCopy Stratix compilation
- Select either **Floating** or **Locked** as the **Location** after you are satisfied with the placement and timing results
- **Reserve Unused Logic** is not applicable in the HardCopy Stratix device placement because logic array block (LAB) contents can not be changed after the **HardCopy Timing Optimization Wizard** is run

An example of a well partitioned design using LogicLock regions effectively for some portions of the design is shown in Figure 14–1. Only the most critical logic functions required are placed in LogicLock regions in order to achieve the desired performance in the HardCopy Stratix

device. The dark blue rectangles shown in Figure 14–1 are the user-assigned LogicLock regions that have fixed locations. In this example, the design needed to be constrained by LogicLock regions first inside the HARDCOPY_FPGA_PROTOTYPE with **Reserve Unused Logic** turned off in **Properties** in LogicLock regions. This selection allows the Quartus II software to isolate and compact the logic of these blocks in the HARDCOPY_FPGA_PROTOTYPE such that the placement is tightly controlled in the HardCopy Stratix device.

Figure 14–1. A Well Partitioned Design



In the example shown in Figure 14–1, once suitable locations were identified for LogicLock regions, the LogicLock region properties were changed from floating to locked. The Quartus II software can then reproduce their placement in subsequent compilations, while focusing attention on fixing other portions of the design.

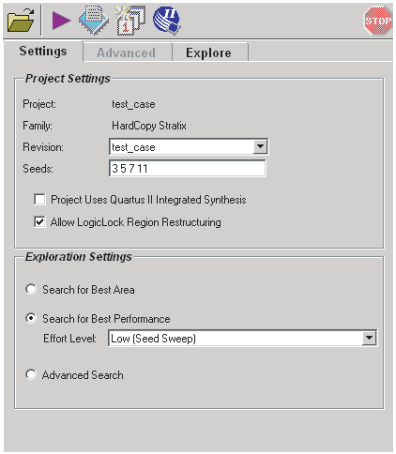
Using Design Space Explorer for HardCopy Stratix Designs

The DSE feature in the Quartus II software allows you to evaluate various compilation settings to achieve the best results for your FPGA designs. DSE can also be used in the HardCopy Stratix project after running the HardCopy Timing Optimization wizard.

Only some of the DSE settings affect HardCopy Stratix designs because HDL synthesis and physical optimization have been completed on the FPGA. No logic restructuring can occur after using the HardCopy Timing Optimization wizard. When you compile your design, the placement of LABs is optimized in the HardCopy Stratix device. To access the DSE GUI

in your open project in the Quartus II software, select **Launch Design Space Explorer** (Tools menu). An example of the DSE GUI and DSE Settings window for the HardCopy Stratix device (Figure 14–2).

Figure 14–2. DSE Settings Window in the DSE GUI



Recommended DSE Settings for HardCopy Stratix Designs

The HardCopy Stratix design does not require all advanced settings or effort-level settings in DSE. Altera recommends using the following settings in DSE for HardCopy Stratix designs:

- In the **Settings** tab (Figure 14–2), make the following selections:
  - Under **Project Settings**, enter several seed numbers in the **Seeds** box. Each seed number requires one full compile of the HardCopy Stratix project.
  - Under **Project Settings**, select **Allow LogicLock Region Restructuring.**
  - Under **Exploration Settings**, select **Search for Best Performance,** and select **Low (Seed Sweep)** from the **Effort Level** menu.
- Turn on **Archive all Compilations** (Options menu).

After running DSE with the seed sweep setting, view the results and identify which seed settings produced the best compilation results. Use the archive of the identified seed, or merge the compilation settings and seed number from the DSE archived project into your primary HardCopy Stratix project.

Performance Improvement Example

With the design used for the performance improvement example in this section, the designer was seeking performance improvement on an HC1S30F780 design for an intellectual property (IP) core consisting of approximately 5200 LEs, 75,000 bits of memory, and two digital signal processing (DSP) multiplier accumulators (MACs). The final application needed to fit in a reserved portion of the HC1S30 device floorplan, so the entire block of IP was initially bounded in a single LogicLock region. The IP block was evaluated as a stand-alone block.

Initial Design Example Settings

The default settings in the Quartus II software version 4.2 were used, with the following initial constraints added:

- The device was set to the target Stratix FPGA device which is the prototype for the HC1S30F780 device:
```
set_global_assignment -name DEVICE
EP1S30F780C6_HARDCOPY_FPGA_PROTOTYPE
```

- A LogicLock region was created for the block to bound it in the reserved region.

- The LogicLock region properties were set to **Auto Size** and **Floating Location**, and **Reserve Unused Logic** was turned on:
```
set_global_assignment -name LL_STATE FLOATING
set_global_assignment -name LL_AUTO_SIZE ON
set_global_assignment -name LL_RESERVED OFF
set_global_assignment -name LL_SOFT OFF
```

- Virtual I/O pins were used for the ports of the core since this core does not interface to pins in the parent design, and the I/O pins were placed outside the LogicLock region and are represented as registers in LEs.

The initial compilation results yielded 65.30-MHz f_{MAX} in the FPGA. The block was constrained through virtual I/O pins and a LogicLock region to keep the logic from spreading throughout the floorplan.

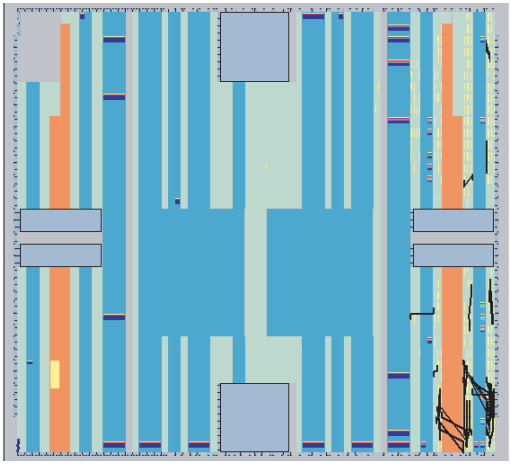
The initial compile-relevant statistics for this example are provided in Table 14–1.

Table 14–1. Initial Compilation Statistics

Result Type	Results
f_{MAX}	65.30 MHz
Total logic elements (LEs)	5,187/32,470 (15%)
Total LABs	564/3,247 (17%)
M512 blocks	20/295 (6%)
M4K blocks	16/171 (9%)
M-RAM blocks	0/2 (0%)
Total memory bits	74,752/2,137,536 (3%)
Total RAM block bits	85,248/2,137,536 (3%)
DSP block 9-bit elements	2/96 (2%)

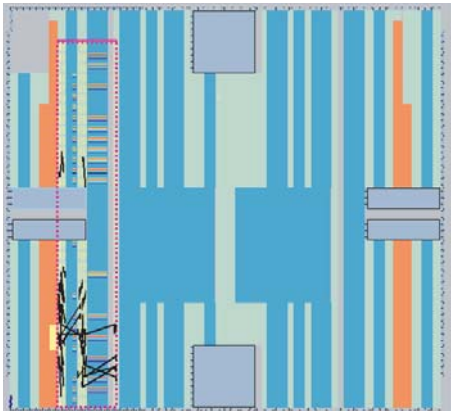
The design project was migrated to the HardCopy device using the **HardCopy Timing Optimization** wizard and was compiled. The default settings of the LogicLock region in a HardCopy Stratix project in the Quartus II software have the **Soft Region** option turned on. With this setting, the HardCopy Stratix compilation yields an f_{MAX} of 66.48 MHz, mainly due to the Fitter placement being scattered in an open design (Figure 14–3). Because the **Soft Region** is set to on, the LogicLock region is not bounded. This is not an optimal placement in the HardCopy Stratix design and is not the best possible performance.

Figure 14–3. HardCopy Stratix Device Floorplan with Soft Region On



To keep the LogicLock region contents bounded in the final placement in the HardCopy Stratix device floorplan, turn off the **Soft Region** option. After turning off the **Soft Region** option and compiling the HardCopy Stratix design, the result is an f_{MAX} of 88.14 MHz—a gain of 33% over the Stratix FPGA device performance. The bounded placement in the LogicLock region helps to achieve performance improvement in well-partitioned design blocks by taking advantage of the smaller die size and custom metal routing interconnect of the HardCopy Stratix device. The floorplan of the bounded LogicLock region is visible in Figure 14–4. In this figure, you can see the difference in disabling the Soft Region setting in the HardCopy Stratix design.

Figure 14–4. HardCopy Stratix Device Floorplan with Soft Region Off



Using Analysis & Synthesis Settings for Performance Improvement

After establishing the baseline for improvement for this design of 65.30 MHz FPGA/88.14 MHz HardCopy, you can gain additional performance improvement in the Stratix FPGA and HardCopy Stratix devices using the available features in the Quartus II software.

Changing the **Analysis & Synthesis Effort** from **Balanced** to **Speed** yields additional benefit in performance, but at the cost of additional LE resources. The Tcl command for this assignment is as follows:

```
set_global_assignment -name
STRATIX_OPTIMIZATION_TECHNIQUE SPEED
```

The relevant compilation results of the FPGA are provided in Table 14–2.

Table 14–2. Relevant Compile Results (Part 1 of 2)

Result Type	Results
f_{MAX}	68.88 MHz
Total logic elements	5,508/32,470 (16%)
Total LABs	598/3,247 (18%)
M512 blocks	20/295 (6%)
M4K blocks	16/171 (9%)

Table 14–2. Relevant Compile Results (Part 2 of 2)

Result Type	Results
M-RAM blocks	0/2 (0%)
Total memory bits	74,752/2,137,536 (3%)
Total RAM block bits	85,248/2,137,536 (3%)
DSP block 9-bit elements	2/96 (2%)

Increasing the LE resources by 6% only yielded an additional 3 MHz in performance in the FPGA, without using additional settings. However, after migrating this design to the HardCopy Stratix design and compiling it, the performance did not improve over the previous HardCopy Stratix design compile, and was slightly worse in performance at 87.34 MHz. This shows that the Quartus II software synthesis was very effective with the **Synthesis Effort Level** set to **Balanced**, and there was only marginal improvement in the FPGA when this option was set to **Speed**.

The next settings activated in this example were the **Synthesis Netlist Optimizations** shown below in Tcl format for WYSIWYG synthesis remapping and gate-level retiming after synthesis mapping:

```
set_global_assignment -name
ADV_NETLIST_OPT_SYNTH_WYSIWYG_REMAP ON

set_global_assignment -name
ADV_NETLIST_OPT_SYNTH_GATE_RETIME ON
```

Making these settings in the FPGA while leaving **Analysis & Synthesis Effort** set to **Speed** yielded some additional improvement in the FPGA as shown in Table 14–3.

Table 14–3. Results of Analysis & Synthesis Effort Set to Speed

Result Type	Results
f_{MAX}	70.28 MHz
Total logic elements	5,515/32,470 (16%)
Total LABs	597/3,247 (18%)

The WYSIWYG resynthesis added a minimal increase in LEs over the speed setting, and the design performance improved by 2 MHz in the FPGA. Using the **HardCopy Timing Optimization** wizard to migrate the

design to HardCopy and subsequently compiling the HardCopy Stratix design, we find that performance is not improved beyond previous compiles, with an f_{MAX} of 86.58 MHz.

The Quartus II software automatically optimizes state machines and restructures multiplexers when these settings are set to **Auto** in the **Analysis and Synthesis** settings. Changing these options from **Auto** usually does not yield performance improvement.

For example, changing the multiplexer restructuring and state machine processing settings from both set to **Auto**, to **On** and **One-Hot**, respectively, actually hurt performance, not allowing the Quartus II software to determine the optimization on a case-by-case basis. With these settings, the FPGA compiled to an f_{MAX} of 65.99 MHz, and the HardCopy Stratix design only performed at 83.77 MHz. For this design example, it is better to leave these settings to **Auto** as seen in the Tcl assignments in the "Using Fitter Assignments and Physical Synthesis Optimizations for Performance Improvement" section, and allow the Quartus II software to determine when to use these features.

Using Fitter Assignments and Physical Synthesis Optimizations for Performance Improvement

After exploring the Analysis & Synthesis optimization settings in the Quartus II software, you can use the Fitter Settings and Physical Synthesis Optimization features to gain further performance improvement in your Stratix FPGA and HardCopy Stratix devices. In this design example, multiplexer and state machine restructuring settings have been set to **Auto**, and the **Synthesis Optimization Technique** is set for **Speed**. The **Fitter effort** is set to **Standard Fit (highest effort)**. The next features enabled are the **Physical Synthesis Optimizations** as seen in the Tcl assignments below and in Figure 14–5:

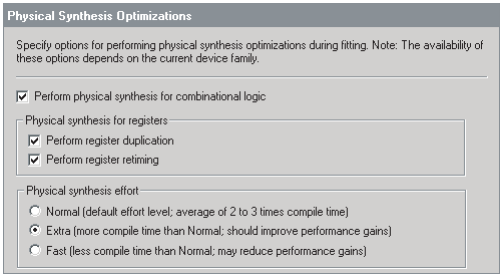
```
set_global_assignment -name
PHYSICAL_SYNTHESIS_COMBO_LOGIC ON

set_global_assignment -name
PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION ON

set_global_assignment -name
PHYSICAL_SYNTHESIS_REGISTER_RETIMING ON

set_global_assignment -name PHYSICAL_SYNTHESIS_EFFORT
EXTRA
```

Figure 14–5. Physical Synthesis Optimization Settings



The compiled design shows a performance increase in the FPGA, running at an f_{MAX} of 74.34 MHz, requiring additional LE resources as a result of the physical synthesis and logic duplication. In this example, you can see how performance can be increased in the Stratix FPGA device at the expense of additional LE resources, as this design's LE resources grew almost 12% over the beginning compilation. The compiled FPGA design's statistics are provided in Table 14–4.

Table 14–4. Compiled FPGA Design Statistics

Result Type	Results
f_{MAX}	74.34 MHz
Total logic elements	5,781/32,470 (17%)
Total LABs	610/3,247 (18%)

Running the **HardCopy Timing Optimization** wizard on this design and compiling the HardCopy Stratix project yields an f_{MAX} of 92.01 MHz, a 24% improvement over the FPGA timing.

Design Space Explorer

The available Fitter Settings produce an additional performance improvement. The DSE feature is used on the Stratix FPGA device to run through the various seeds in the design and select the best seed point to use for future compiles. This can often yield additional performance benefits as the Quartus II software further refines placement of the LEs and performs clustering of associated logic together.

For this design example, DSE was run with high effort (physical synthesis) and multiple placement seeds. Table 14–5 shows the DSE results. The base compile matches the fifth compile in the DSE variations, showing that the work already done on the design before DSE was optimal. The FPGA project was optimized before running DSE.

Table 14–5. DSE Results

Compile Point	Clock Period: CLK	Logic Cells
Base (Best)	13.451 ns (74.34 MHz)	5,781
1	13.954 ns	5,703
2	13.712 ns	6,447
3	14.615 ns	5,777
4	13.911 ns	5,742
5	13.451 ns	5,781
6	14.838 ns	5,407
7	14.177 ns	5,751
8	14.479 ns	5,827
9	14.863 ns	5,596
10	14.662 ns	5,605
11	14.250 ns	5,710
12	14.016 ns	5,708
13	13.840 ns	5,802
14	13.681 ns	5,788
15	14.829 ns	5,644

Additional correlation is seen inside the <*project*>**.dse.rpt** file, showing the summary of assignments used for each compile inside the Quartus II software. The base compile settings and the fifth compile settings show good correlation, as shown in Table 14–6. The MUX_RESTRUCTURE setting did not have any effect on the design performance. This may be due to an already efficient HDL coding for multiplexer structures, requiring no optimization.

Table 14–6. Base Compile and Fifth Compile Correlation

Setting	New Value	Base Value
PHYSICAL_SYNTHESIS_REGISTER_RETIMING	ON	ON
SEED	1	1
STATE_MACHINE_PROCESSING	AUTO	AUTO

Table 14–6. Base Compile and Fifth Compile Correlation

Setting	New Value	Base Value
MUX_RESTRUCTURE	OFF	AUTO
PHYSICAL_SYNTHESIS_COMBO_LOGIC	ON	ON
FITTER_EFFORT	STANDARD FIT	STANDARD FIT
AUTO_PACKED_REGISTERS_STRATIX	NORMAL	NORMAL
PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION	ON	ON
ADV_NETLIST_OPT_SYNTH_GATE_RETIME	ON	ON
STRATIX_OPTIMIZATION_TECHNIQUE	SPEED	SPEED
PHYSICAL_SYNTHESIS_EFFORT	EXTRA	EXTRA

The information presented in Table 14–6 confirms that the FPGA Prototype device has been optimized as much as possible without manual floorplan adjustments.

Design Space Explorer for HardCopy Stratix Devices

Migrating this compiled design to the HardCopy Stratix project and compiling the HardCopy Stratix design optimization, results in a design performance of 92.01 MHz. The next task is to run DSE on the HardCopy Stratix project using **Low Effort (Seed Sweep)** in the **Exploration Settings**, and entering a range of seed numbers with which to compile the project.

The results of the DSE run with the **Seed Sweep** option are summarized in Table 14–7.

Table 14–7. DSE Results Run with Seed Sweep

Compile Point	Clock Period: CLK
Base (Best)	10.868 ns
1	11.710 ns
2	11.040 ns
3	10.790 ns
4	10.945 ns
5	11.154 ns
6	11.707 ns

Table 14–7. DSE Results Run with Seed Sweep

Compile Point	Clock Period: CLK
7	11.648 ns
8	11.476 ns
9	11.423 ns
10	11.449 ns

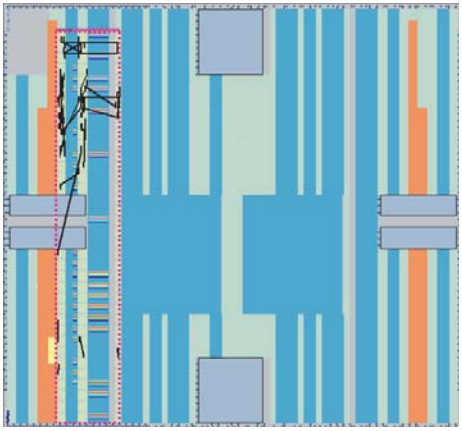
The results in Table 14–7 illustrate how the **Seed Sweep** option in DSE provides additional improvement in the HardCopy Stratix design, even after DSE has been run on the Stratix FPGA project. In this example, compile point 3 using seed value = 4 turns out to be slightly beneficial over other seeds in the Fitter Placement. The HardCopy Stratix device has an f_{MAX} of 92.71 MHz.

Back-Annotation & Location Assignment Adjustments

Another technique available for improving performance in the HardCopy Stratix design is manually adjusting placement and back-annotating location assignments from the placement results. These techniques should be one of the last steps taken for design optimization of HardCopy Stratix devices.

Observing the floorplan of the 92.71 MHz compile (Figure 14–6), the placement of the LogicLock region is stretched vertically, and additional improvement is possible if the aspect ratio of the LogicLock region is defined, and placement in it is refined.

Figure 14–6. Vertically Stretched LogicLock Region

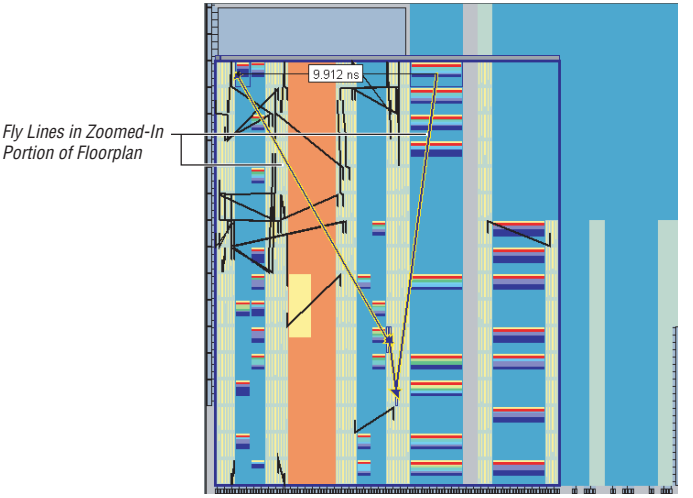


This floorplan would be better optimized if the LogicLock region had a more square shape, helping the paths that go from memory-to-memory, by containing the M4K and M512 memory blocks in a smaller space, and allowing LAB placement to be adjusted by the Fitter. In the HardCopy Stratix device, signals are routed between LABs, DSP blocks, and memory blocks using the customized metal layers. The reconfigurable routing tracks in the Stratix FPGA device limit the routing paths and delays between elements in the HardCopy Stratix device. This flexibility allows for aspect ratio changes in LogicLock regions, so the raw distance between points becomes the critical factor, and not the usage of available routing resources in the FPGA.

For the final placement optimization in this example, the LogicLock region was fixed in a square region that encompassed two columns of M4K blocks, four columns of M512 blocks, two columns of DSP blocks, and enough LABs to fit the remaining resources required. After compiling the design with these new LogicLock assignments, the performance increased to 93.46 MHz in the HardCopy Stratix device. The critical path and LogicLock region location can be seen in the zoomed-in area of the floorplan (Figure 14–7).

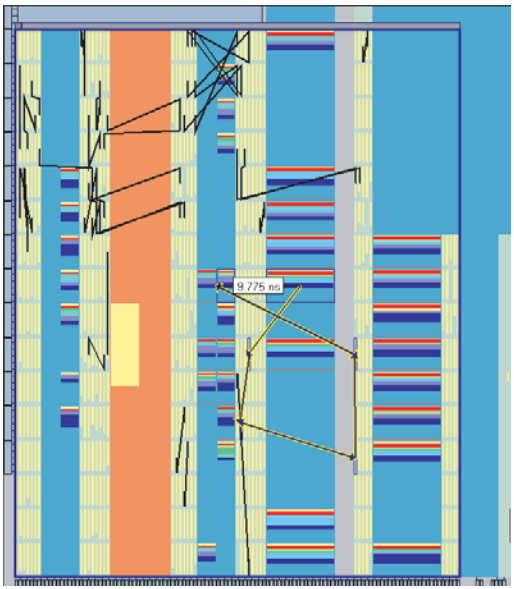
You can see in Figure 14–7 that the critical path shown is from an M4K block to an M512 block through several levels of logic. The placement of the memory blocks can be optimized manually, since the LogicLock region contains more memory blocks than necessary.

Figure 14–7. Critical Path & LogicLock Region



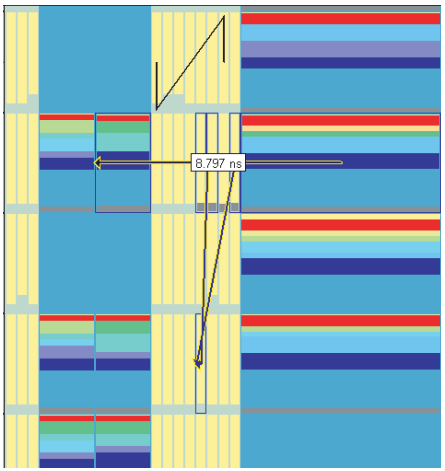
Using the critical path "fly lines" as a guide for placement optimization, manual location assignments were made for some of the M512 and M4K instances used in the design. The resulting compile improved the f_{MAX} to 94.67 MHz. The new critical path (Figure 14–8) shows how placement of all path elements are confined to a much smaller area. As a result, the routing distances and delays are smaller through the path.

Figure 14–8. New Critical Path



Examining this new critical path placement, you can see that there is room for further performance improvement through additional location assignments. The current slowest path is 9.775 ns of delay. Manually moving the LABs in this critical path and placing them between the M4K and M512 endpoints, and subsequently recompiling, shows improved results not only for this path, but for several other paths, as this path contained a major timing bottleneck. The critical path between this start and endpoint was reduced to 8.797 ns (Figure 14–9). However, the entire design only improved to 100.30 MHz because other paths are now the slowest paths in the design. This illustrates that fixing one major bottleneck path can raise the entire design performance since one high fanout node can affect multiple timing paths, as was the case in this example.

Figure 14–9. Improved Results



In summary, this design example started with 65.30 MHz in the Stratix FPGA device, and was improved to 74.34 MHz. It was then taken from the Stratix FPGA device compile and improved to 100.30 MHz in the HardCopy Stratix design, for a performance improvement of 35%.

Conclusion

Using performance-optimization techniques specifically for HardCopy Stratix devices can achieve significant performance improvement over the Stratix FPGA prototype device. Many of these changes must be incorporated up-front in the HARDCOPY_FPGA_PROTOTYPE so that your design is properly prepared for performance improvement after running the HardCopy Timing Optimization wizard.

The example discussed in this chapter demonstrates the process for performance improvement and various features in the Quartus II software available for use when optimizing your Stratix FPGA prototype and HardCopy Stratix device. It also demonstrates the importance of planning ahead for the HardCopy Stratix design implementation while continuing to work in the HARDCOPY_FPGA_PROTOTYPE design if you are going to seek performance improvement in the HardCopy Stratix device.

Section III. HardCopy
APEX Device Family Data Sheet

This section provides designers with the data sheet specifications for HardCopy® APEX™ devices. These 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 HardCopy APEX devices.

This section contains the following:

- Chapter 15, Introduction to HardCopy APEX Devices
- Chapter 16, Description, Architecture & Features
- Chapter 17, Boundary-Scan Support
- Chapter 18, Operating Conditions

Revision History

The table below shows the revision history for chapter 15 through 18.

Chapter(s)	Date / Version	Changes Made
Chapter 15	March 2006	Formerly chapter 9; no content change.
	January 2005 v2.0	Update device names and other minor textual changes
	June 2003 v1.0	Initial release of Chapter 9, *Introduction to HardCopy APEX Devices*, in the HardCopy Device Handbook
Chapter 16	March 2006	Formerly chapter 10; no content change.
	January 2005 v2.0	Update device names and other minor textual changes
	June 2003 v1.0	Initial release of Chapter 10, *Description, Architecture & Features*, in the HardCopy Device Handbook
Chapter 17	March 2006	Formerly chapter 11; no content change.
	January 2005 v2.0	Update device names and other minor textual changes.
	June 2003 v1.0	Initial release of *Boundary-Scan Support* in the HardCopy Device Handbook.
Chapter 18	March 2006	Formerly chapter 12; no content change.
	January 2005 v2.0	Update device names and other minor textual changes.
	June 2003 v1.0	Initial release of *Operating Conditions*, in the HardCopy Device Handbook

15. Introduction to HardCopy APEX Devices

H51006-2.0

Introduction

HardCopy® APEX™ devices enable high-density APEX 20KE device technology to be used in high-volume applications where significant cost reduction is desired. HardCopy APEX devices are physically and functionally compatible with APEX 20KC and APEX 20KE devices. They combine the time-to-market advantage, performance, and flexibility of APEX 20KE devices with the ability to move to high-volume, low-cost devices for production. The migration process from an APEX 20KE device to a HardCopy APEX device is fully automated, with designer involvement limited to providing a few Quartus® II software-generated output files.

Features...

HardCopy APEX devices are manufactured using an 0.18-µm CMOS six-layer-metal process technology:

- Preserves functionality of a configured APEX 20KC or APEX 20KE device
- Pin-compatible with APEX 20KC or APEX 20KE devices
- Meets or exceeds timing of configured APEX 20KE and APEX 20KC devices
- Optional emulation of original programmable logic device (PLD) programming sequence
- High-performance, low-power device
- MultiCore™ architecture integrating embedded memory and look-up table (LUT) logic used for register-intensive functions
- Embedded system blocks (ESBs) used to implement memory functions, including first-in first-out (FIFO) buffers, dual-port RAM, and content-addressable memory (CAM)
- Customization performed through metallization layers

High-density architecture:

- 400,000 to 1.5 million typical gates (Table 15–1)
- Up to 51,840 logic elements (LEs)
- Up to 442,368 RAM bits that can be used without reducing available logic

Table 15–1. HardCopy APEX Device Features *Note (1)*

Feature	HC20K400	HC20K600	HC20K1000	HC20K1500
Maximum system gates	1,052,000	1,537,000	1,772,000	2,392,000
Typical gates	400,000	600,000	1,000,000	1,500,000
LEs	16,640	24,320	38,400	51,840
ESBs	104	152	160	216
Maximum RAM bits	212,992	311,296	327,680	442,368
Phase-locked loops (PLLs)	4	4	4	4
Maximum macrocells	1,664	2,432	2,560	3,456
Maximum user I/O pins	488	588	708	808

Note to Table 15–1:
(1) The embedded IEEE Std. 1149.1 Joint Test Action Group (JTAG) boundary-scan circuitry contributes up to 57,000 additional gates.

…and more Features

Low-power operation:

- 1.8-V supply voltage (Table 15–2)
- MultiVolt™ I/O support for 1.8, 2.5, and 3.3 V interfaces
- ESBs offering power-saving mode

Flexible clock management circuitry with up to four phase-locked loops (PLLs):

- Built-in low-skew clock tree
- Up to eight global clock signals
- ClockLock™ feature reducing clock delay and skew
- ClockBoost™ feature providing clock multiplication and division
- ClockShift™ feature providing clock phase and delay shifting

Powerful I/O features:

- Compliant with peripheral component interconnect Special Interest Group (PCI SIG) *PCI Local Bus Specification, Revision 2.2* for 3.3-V operation at 33 or 66 MHz and 32 or 64 bits

- Support for high-speed external memories, including DDR, synchronous dynamic RAM (SDRAM), and ZBT static RAM (SRAM)
- 16 input and 16 output LVDS channels
- Fast t_{CO} and t_{SU} times for complex logic
- MultiVolt I/O support for 1.8-V, 2.5-V, and 3.3-V interfaces
- Individual tri-state output enable control for each pin
- Output slew-rate control to reduce switching noise
- Support for advanced I/O standards, including LVDS, LVPECL, PCI-X, AGP, CTT, SSTL-3 and SSTL-2, GTL+, and HSTL Class I
- Supports hot-socketing operation

Table 15–2. HardCopy APEX Device Supply Voltages

Feature	Voltage
Internal supply voltage (V_{CCINT})	1.8 V
MultiVolt I/O interface voltage levels (V_{CCIO})	1.8 V, 2.5 V, 3.3 V, 5.0 V *(1)*

Note to Table 15–2:
(1) HardCopy APEX devices can be 5.0-V tolerant by using an external resistor.

HardCopy APEX device implementation features:

- Customized interconnect for each design
- HardCopy APEX devices preserve APEX 20K device MegaLAB™ structure, LEs, ESBs, IOE, PLLs, and LVDS circuitry
- Up to four metal layers customizable for customer designs
- Completely automated proprietary design migration flow
  - Testability analysis and fix
  - Automatic test pattern generation (ATPG)
  - Automatic place and route
  - Static timing analysis
  - Static functional verification
  - Physical verification

Tables 15–3 through 15–6 show the HardCopy APEX device ball-grid array (BGA) and FineLine BGA® package options, I/O counts, and sizes.

Table 15–3. HardCopy APEX Device BGA Package Options & I/O Count Note (1)

Device	652-Pin BGA
HC20K400	488
HC20K600	488
HC20K1000	488
HC20K1500	488

Table 15–4. HardCopy APEX Device FineLine BGA Package Options & I/O Count Note (1)

Device	672-Pin	1,020-Pin
HC20K400	488	–
HC20K600	508	588
HC20K1000	508	708
HC20K1500	–	808

Note to Tables 15–3 and 15–4:
(1) I/O counts include dedicated input and clock pins.

Table 15–5. HardCopy APEX Device BGA Package Sizes

Feature	652-Pin BGA
Pitch (mm)	1.27
Area (mm^2)	2,025
Length × width (mm × mm)	45.0 × 45.0

Table 15–6. HardCopy APEX Device FineLine BGA Package Sizes

Feature	672-Pin	1,020-Pin
Pitch (mm)	1.00	1.00
Area (mm^2)	729	1,089
Length × width (mm × mm)	27 × 27	33 × 33

16. Description, Architecture & Features

H51007-2.0

General Description

HardCopy® APEX™ devices extend the flexibility of high-density FPGAs to a cost-effective, high-volume production solution. The migration process from an Altera® FPGA to a HardCopy APEX device offers seamless migration of a high-density system-on-a-programmable-chip (SOPC) design to a low-cost alternative device with minimal risk. Using HardCopy APEX devices, Altera's SOPC solutions can be leveraged from prototype to production, while reducing costs and speeding time-to-market.

A significant benefit of HardCopy devices is that customers do not need to be involved in the device migration process. Unlike application-specific integrated circuit (ASIC) development, the HardCopy design flow does not require generation of test benches, test vectors, or timing and functional simulation. The HardCopy migration process only requires the Quartus® II software-generated output files from a fully functional APEX 20KE or APEX 20KC device. Altera performs the migration and delivers functional prototypes in as few as seven weeks.

A risk-free alternative to ASICs, HardCopy APEX devices are customizable, full-featured devices created by Altera's proprietary design migration methodology. They are based on Altera's industry-leading high-density device architecture and use an area-efficient sea-of-logic-elements (SOLE) core.

HardCopy APEX devices retain all the same features as the APEX 20KE and APEX 20KC devices, which combine the strength of LUT-based and product-term-based devices in conjunction with the same embedded memory structures. All routing resources that were programmable in the APEX 20K device family are replaced by custom interconnect, resulting in a considerable die size reduction and subsequent cost saving.

The SRAM configuration cells of the original FPGA are replaced in HardCopy APEX devices by metal elements, which define the function of each logic element (LE), embedded memory, and I/O cell in the device. These resources are connected to each other using the same metallization layers. Once a HardCopy APEX device has been manufactured, the functionality of the device is fixed and no programming is possible. Altera performs the migration of the original FPGA design to an equivalent HardCopy APEX device using a proprietary design migration flow.

The migration of a FPGA to a HardCopy APEX device begins with a user design that has been implemented in an APEX 20KE or APEX 20KC device. Table 16–1 shows the device equivalence for HardCopy and APEX 20KE or APEX 20KC devices.

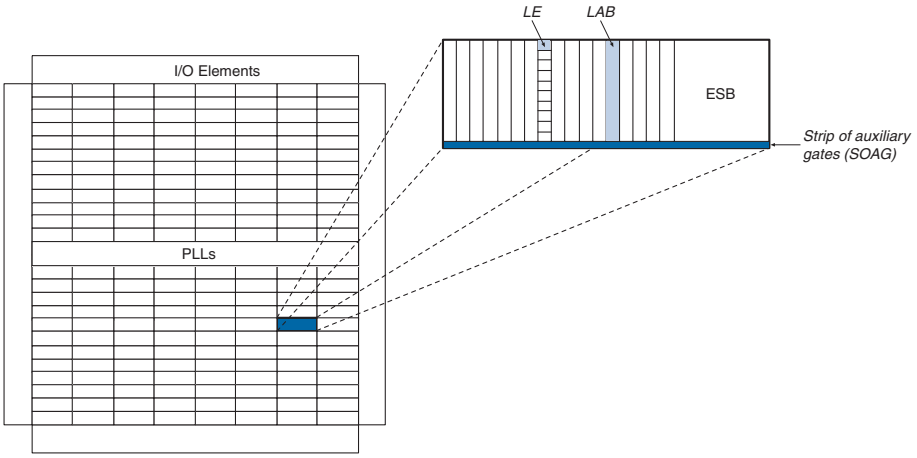
Table 16–1. HardCopy & APEX 20KE or APEX 20C Device Equivalence

HardCopy APEX Device	APEX 20KE Device	APEX 20KC Device
HC20K1500	EP20K1500E	EP20K1500C
HC20K1000	EP20K1000E	EP20K1000C
HC20K600	EP20K600E	EP20K600C
HC20K400	EP20K400E	EP20K400C

To ensure HardCopy device performance and functionality, the APEX 20K design must be completely debugged before committing the design to HardCopy device migration.

HardCopy APEX device implementation begins with extracting the Quartus II software-generated SRAM Object File (**.sof**) and converting its connectivity information into a structural Verilog HDL netlist. This netlist is then placed and routed in a similar fashion to a gate array. There are no dedicated routing channels. The router can exploit all available metal layers (up to four) and route over LE cells and other functional blocks. Altera's proprietary architecture and design methodology will guarantee virtually 100% routing of any APEX 20KE or APEX 20KC design compiled and fitted successfully using the Quartus II software. Place and route is timing-driven and will comply with the timing constraints of the original FPGA design as specified in the Quartus II software. Figure 16–1 shows a diagram of the HardCopy APEX device architecture.

Figure 16–1. HardCopy APEX Device Architecture



The strip of auxiliary gates (SOAG) is an Altera proprietary feature designed into the HardCopy APEX device and is used during the HardCopy device implementation process. The SOAG structures can be configured into several different types of functions through the use of metallization. For example, high fanout signals require adequate buffering, so buffers are built out of SOAG cells for this purpose.

HardCopy APEX devices include the same advanced features as the APEX 20KE and APEX 20KC devices, such as enhanced I/O standard support, content-addressable memory (CAM), additional global clocks, and enhanced ClockLock® circuitry. Table 16–2 lists the features included in HardCopy APEX devices.

Table 16–2. HardCopy APEX Device Features (Part 1 of 2)

Feature	HardCopy Devices
MultiCore™ system integration	Full support
Hot-socketing support	Full support
32-/64-bit, 33-MHz PCI	Full compliance
32-/64-bit, 66-MHz PCI	Full compliance
MultiVolt™ I/O operation	1.8-V, 2.5-V, or 3.3-V V_{CCIO} V_{CCIO} selected bank by bank 5.0-V tolerant with use of external resistor

Table 16–2. HardCopy APEX Device Features (Part 2 of 2)

Feature	HardCopy Devices
ClockLock support	Clock delay reduction *m*/(*n* × *v)* clock multiplication Drive ClockLock output off-chip External clock feedback ClockShift circuitry LVDS support Up to four PLLs ClockShift™, clock phase adjustment
Dedicated clock and input pins	Eight
I/O standard support	1.8-V, 2.5-V, 3.3-V, 5.0-V I/O 3.3-V PCI and PCI-X 3.3-V AGP CTT GTL+ LVCMOS LVTTL True-LVDS™ and LVPECL data pins LVDS and LVPECL clock pins HSTL class I PCI-X SSTL-2 class I and II SSTL-3 class I and II
Memory support	CAM Dual-port RAM FIFO RAM ROM

All HardCopy APEX devices are tested using ATPG vectors prior to shipment. For fully synchronous designs near 100%, fault coverage can be achieved through the built-in full-scan architecture. ATPG vectors allow the designer to focus on simulation and design verification.

Because the configuration of HardCopy APEX devices is built-in during manufacture, they cannot be configured in-system. However, if the APEX 20KE or APEC 20KC device configuration sequence must be emulated, the HardCopy APEX device has this capability.

All of the device features of APEX 20KE and APEX 20KC devices are available in HardCopy APEX devices. For a detailed description of these device features, refer to the *APEX 20K Programmable Logic Device Family Data Sheet* and the *APEX 20KC Programmable Logic Device Family Data Sheet*.

Differences Between HardCopy APEX & APEX 20K FPGAs

Several differences must be considered before a design is ready for implementation in HardCopy technology:

- HardCopy APEX devices are only customizable at the time they are manufactured. Make sure that the original APEX 20KE or APEX 20KC device has undergone thorough testing in the end-system before deciding to proceed with migration to a HardCopy APEX device, because no changes can be made to the HardCopy APEX device after it has been manufactured.
- ESBs that are configured as RAM or CAM will power-up un-initialized in the HardCopy APEX device. In the FPGA it is possible to configure, or "pre-load," the ESB memory as part of the configuration sequence, then overwrite it when the device is in normal functional mode. This pre-loaded memory feature of the FPGA is not available in HardCopy devices. If a design contains RAM or CAM with assumed data values at power-up, then the HardCopy APEX device will not operate as expected. If a design uses this feature, then it should be re-compiled without the memory pre-load. ESBs configured as ROM are fully supported.
- The JTAG boundary scan order in the HardCopy APEX device is different compared to the APEX 20K device. A HardCopy BSDL file that describes the re-ordered boundary scan chain should be used.

  The BSDL files for HardCopy APEX devices are different from the corresponding APEX 20KE or APEX 20KC devices. Download the correct HardCopy BSDL file from Altera's web site at **www.altera.com**.
- The advanced 0.18-µm aluminum metal process is used to support both APEX 20KE and APEX 20KC devices. The performance improvement achieved by the die size reduction and metal interconnect optimization more than offsets the need for copper in this case. Altera guarantees that a target HardCopy APEX device will provide the same or better performance as in the corresponding APEX 20KE or APEX 20KC device.

Power-up Mode & Configuration Emulation

Unlike their FPGA counterparts, HardCopy APEX devices do not need to be configured. However, to facilitate seamless migration, configuration can be emulated in these devices. There are three modes in which a

HardCopy APEX device can be prepared for operation after power up: instant on, instant on after 50 ms, and configuration emulation. Each mode is described below.

- In instant on mode, the HardCopy APEX device is available for use shortly after the device receives power. The on-chip power-on-reset circuit will set or reset all registers. The CONF_DONE output will be tri-stated once the power-on reset (POR) has elapsed. No configuration device or configuration input signals are necessary.
- In instant on after 50 ms mode, the HardCopy APEX device will perform in a similar fashion to the Instant On mode, except that there will be an additional delay of 50 ms (nominal), during which time the device will be held in reset stage. The CONF_DONE output is pulled low during this time and then tri-stated after the 50 ms have elapsed. No configuration devices or configuration input signals are necessary for this option.
- In configuration emulation mode, the HardCopy APEX device undergoes an emulation of a full configuration sequence as if configured by an external processor or an EPC device. In this mode, the CONF_DONE signal is tri-stated after the correct number of clock cycles. This mode may be useful where there is some dependency on the configuration sequence (e.g., multi-device configuration or processor initialization). In this mode, the device expects to see all configuration control and data input signals.

Speed Grades

Because HardCopy APEX devices are customized, no speed grading is performed. All HardCopy APEX devices will meet the timing requirements of the original FPGA of the fastest speed grade. Generally, HardCopy APEX devices will have a higher f_{MAX} than the corresponding FPGA, but the speed increase will vary on a design-by-design basis.

Quartus II-Generated Output Files

The HardCopy migration process requires several Quartus II software-generated files. These key output files are listed and explained below.

- The SRAM Object File (**.sof**) contains all of the necessary information needed to configure a FPGA
- The Compiler Report File (**.csf.rpt**) is parsed to extract useful information about the design
- The Verilog atom-based netlist file (**.vo**) is used to check the HardCopy netlist
- The pin out information file (**.pin**) contains user signal names and I/O configuration information
- The Delay Information File (**.sdo**) is used to check the original FPGA timing

- A completed HardCopy timing requirements file describes all necessary timing information on the design. A template of this text file is available for download from the Altera web site at **www.altera.com**.

The migration process consists of several steps. First, a netlist is constructed from the SOF. Then, the netlist is checked to ensure that the built-in scan test structures will operate correctly. The netlist is then fed into a place-and-route engine, and the design interconnect is generated. Static timing analysis ensures that all timing constraints are met, and static functional verification techniques are employed to ensure correct device migration. After successfully completing these stages, physical verification of the device takes place, and the metal mask layers are taped out to fabricate HardCopy APEX devices.

17. Boundary-Scan Support

H51009-2.0

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All HardCopy devices provide JTAG BST circuitry that complies with the IEEE Std. 1149.1-1990 specification. HardCopy® APEX™ devices support the JTAG instructions shown in Table 17–1.

The BSDL files for HardCopy devices are different from the corresponding APEX 20KE or APEX 20KC parts. Download the correct HardCopy BSDL file from Altera's web site at www.altera.com.

Table 17–1. HardCopy APEX JTAG Instructions

JTAG Instruction	Description
SAMPLE/PRELOAD	SAMPLE/PRELOAD 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. It is also used by the SignalTap embedded logic analyzer.
EXTEST	Allows the external circuitry and board-level interconnections to be tested by forcing a test pattern at the output pins and capturing test results at the input pins
BYPASS	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation
USERCODE	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	Selects the IDCODE register and places it between the TDI and TDO pins, allowing the IDCODE to be serially shifted out of TDO

HardCopy APEX devices instruction register length is 10 bits; the USERCODE register length is 32 bits. Tables 17–2 and 17–3 show the boundary-scan register length and device IDCODE information for HardCopy devices.

Table 17–2. HardCopy APEX Boundary-Scan Register Length

Device	Boundary-Scan Register Length
HC20K400	1,506
HC20K600	1,806
HC20K1000	2,190
HC20K1500	2,502

Table 17–3. 32-Bit HardCopy APEX Device IDCODE

Device	IDCODE (32 Bits) *Note (1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	1 (1 Bit) *(2)*
HC20K400	0000	1000 0100 0000 0000	000 0110 1110	1
HC20K600	0000	1000 0110 0000 0000	000 0110 1110	1
HC20K1000	0000	1001 0000 0000 0000	000 0110 1110	1
HC20K1500	0000	1001 0101 0000 0000	000 0110 1110	1

Notes to Table 17–3:
(1) The most significant bit (MSB) is on the left.
(2) The IDCODE's least significant bit (LSB) is always 1.

Figure 17–1 shows the timing requirements for the JTAG signals.

Figure 17–1. HardCopy JTAG Waveforms

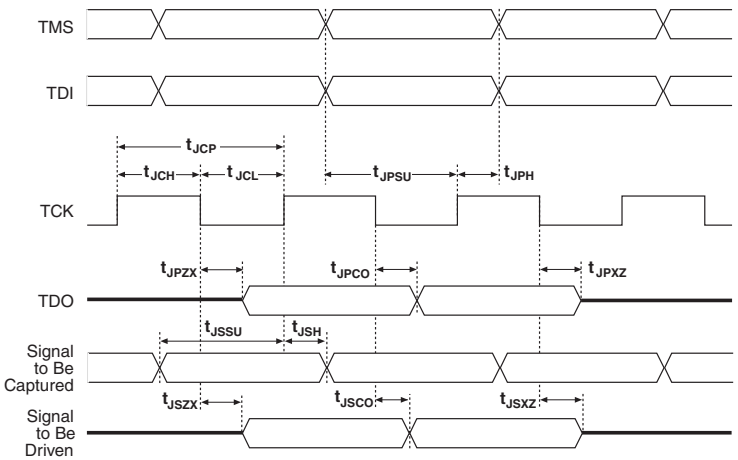


Table 17–4 shows the JTAG timing parameters and values for HardCopy devices.

Table 17–4. HardCopy APEX JTAG Timing Parameters & 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

For more information on using JTAG BST circuitry in Altera devices, refer to *Application Note 39 (IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices)*.

18. Operating Conditions

H51010-2.0

Recommended Operating Conditions

Tables 18–1 through 18–4 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 1.8 V HardCopy® APEX™ devices.

Table 18–1. HardCopy APEX Device Absolute Maximum Ratings *Note (1)*

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage	With respect to ground *(2)*	–0.5	2.5	V
V_{CCIO}			–0.5	4.6	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 18–2. HardCopy APEX Device Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Max	Unit
V_{CCINT}	Supply voltage for internal logic and input buffers	*(3)*, *(4)*	1.71 (1.71)	1.89 (1.89)	V
V_{CCIO}	Supply voltage for output buffers, 3.3-V operation	*(3)*, *(4)*	3.00 (3.00)	3.60 (3.60)	V
	Supply voltage for output buffers, 2.5-V operation	*(3)*, *(4)*	2.375 (2.375)	2.625 (2.625)	V
V_I	Input voltage	*(2)*, *(5)*	–0.5	4.1	V
V_O	Output voltage		0	V_{CCIO}	V
T_J	Junction temperature	For commercial use	0	85	° C
		For industrial use	–40	100	° C
t_R	Input rise time (10% to 90%)			40	ns
t_F	Input fall time (90% to 10%)			40	ns

Table 18–3. HardCopy APEX Device DC Operating Conditions (Part 1 of 2) *Notes (6), (7), (8)*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{IH}	High-level LVTTL, CMOS, or 3.3-V PCI input voltage		1.7, 0.5 × V_{CCIO} *(8)*		4.1	V
V_{IL}	Low-level LVTTL, CMOS, or 3.3-V PCI input voltage		–0.5		0.8, 0.3 × V_{CCIO} *(8)*	V
V_{OH}	3.3-V high-level LVTTL output voltage	I_{OH} = –12 mA DC, V_{CCIO} = 3.00 V *(9)*	2.4			V
	3.3-V high-level LVCMOS output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 3.00 V *(9)*	V_{CCIO} – 0.2			V
	3.3-V high-level PCI output voltage	I_{OH} = –0.5 mA DC, V_{CCIO} = 3.00 to 3.60 V *(9)*	0.9 × V_{CCIO}			V
	2.5-V high-level output voltage	I_{OH} = –0.1 mA DC, V_{CCIO} = 2.30 V *(9)*	2.1			V
		I_{OH} = –1 mA DC, V_{CCIO} = 2.30 V *(9)*	2.0			V
		I_{OH} = –2 mA DC, V_{CCIO} = 2.30 V *(9)*	1.7			V
V_{OL}	3.3-V low-level LVTTL output voltage	I_{OL} = 12 mA DC, V_{CCIO} = 3.00 V *(10)*			0.4	V
	3.3-V low-level LVCMOS output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 3.00 V *(10)*			0.2	V
	3.3-V low-level PCI output voltage	I_{OL} = 1.5 mA DC, V_{CCIO} = 3.00 to 3.60 V *(10)*			0.1 × V_{CCIO}	V
	2.5-V low-level output voltage	I_{OL} = 0.1 mA DC, V_{CCIO} = 2.30 V *(10)*			0.2	V
		I_{OL} = 1 mA DC, V_{CCIO} = 2.30 V *(10)*			0.4	V
		I_{OL} = 2 mA DC, V_{CCIO} = 2.30 V *(10)*			0.7	V
I_I	Input pin leakage current *(11)*	V_I = 4.1 to –0.5 V	–10		10	µA
I_{OZ}	Tri-stated I/O pin leakage current *(11)*	V_O = 4.1 to –0.5 V	–10		10	µA

Table 18–3. HardCopy APEX Device DC Operating Conditions (Part 2 of 2) Notes (6), (7), (8)

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I_{CC0}	V_{CC} supply current (standby) (All ESBs in power-down mode)	V_I = ground, no load, no toggling inputs, -7 speed grade		10		mA
		V_I = ground, no load, no toggling inputs, -8, -9 speed grades		5		mA
R_{CONF}	Value of I/O pin pull-up resistor before and during configuration emulation	V_{CCIO} = 3.0 V (12)	20		50	kΩ
		V_{CCIO} = 2.375 V (12)	30		80	kΩ
		V_{CCIO} = 1.71 V (12)	60		150	kΩ

Table 18–4. HardCopy APEX Device Capacitance Note (13)

Symbol	Parameter	Conditions	Min	Typ	Max
C_{IN}	Input capacitance	V_{IN} = 0 V, f = 1.0 MHz		8	pF
C_{INCLK}	Input capacitance on dedicated clock pin	V_{IN} = 0 V, f = 1.0 MHz		12	pF
C_{OUT}	Output capacitance	V_{OUT} = 0 V, f = 1.0 MHz		8	pF

Notes to Table 18–1 through 18–4:

(1) Refer to the *Operating Requirements for Altera Devices Data Sheet*.
(2) Minimum DC input is –0.5 V. During transitions, the inputs may undershoot to –0.5 V or overshoot to 4.6 V for input currents less than 100 mA and periods shorter than 20 ns.
(3) Numbers in parentheses are for industrial-temperature-range devices.
(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} = 1.8 V, and V_{CCIO} = 1.8 V, 2.5 V, or 3.3 V.
(7) These values are specified under the HardCopy device recommended operating conditions, shown in Table 18–2 on page 18–1.
(8) Refer to *AN 117: Using Selectable I/O Standards in Altera Devices* for the V_{IH}, V_{IL}, V_{OH}, V_{OL}, and I_I parameters when V_{CCIO} = 1.8 V.
(9) The APEX 20KE input buffers are compatible with 1.8-V, 2.5-V and 3.3-V (LVTTL and LVCMOS) signals. Additionally, the input buffers are 3.3-V PCI compliant. Input buffers also meet specifications for GTL+, CTT, AGP, SSTL-2, SSTL-3, and HSTL.
(10) The I_{OH} parameter refers to high-level TTL, PCI, or CMOS output current.
(11) This value is specified for normal device operation. The value may vary during power-up.
(12) Pin pull-up resistance values will be lower if an external source drives the pin higher than V_{CCIO}.
(13) Capacitance is sample-tested only.

Tables 18–5 through 18–20 list the DC operating specifications for the supported I/O standards. These tables list minimal specifications only; HardCopy devices may exceed these specifications.

Table 18–5. LVTTL I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Units
V_{CCIO}	Output supply voltage		3.0	3.6	V
V_{IH}	High-level input voltage		2.0	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3	0.8	V
I_I	Input pin leakage current	V_{IN} = 0 V or 3.3 V	–10	10	µA
V_{OH}	High-level output voltage	I_{OH} = –12 mA, V_{CCIO} = 3.0 V *(1)*	2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 12 mA, V_{CCIO} = 3.0 V *(2)*		0.4	V

Table 18–6. LVCMOS I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Units
V_{CCIO}	Power supply voltage range		3.0	3.6	V
V_{IH}	High-level input voltage		2.0	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3	0.8	V
I_I	Input pin leakage current	V_{IN} = 0 V or 3.3 V	–10	10	µA
V_{OH}	High-level output voltage	V_{CCIO} = 3.0 V I_{OH} = –0.1 mA *(1)*	V_{CCIO} – 0.2		V
V_{OL}	Low-level output voltage	V_{CCIO} = 3.0 V I_{OL} = 0.1 mA *(2)*		0.2	V

Table 18–7. 2.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Units
V_{CCIO}	Output supply voltage		2.375	2.625	V
V_{IH}	High-level input voltage		1.7	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3	0.7	V
I_I	Input pin leakage current	V_{IN} = 0 V or 3.3 V	–10	10	µA
V_{OH}	High-level output voltage	I_{OH} = –0.1 mA *(1)*	2.1		V
		I_{OH} = –1 mA *(1)*	2.0		V
		I_{OH} = –2 mA *(1)*	1.7		V
V_{OL}	Low-level output voltage	I_{OL} = 0.1 mA *(2)*		0.2	V
		I_{OL} = 1 mA *(2)*		0.4	V
		I_{OL} = 2 mA *(2)*		0.7	V

Table 18–8. 1.8-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Units
V_{CCIO}	Output supply voltage		1.7	1.9	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage			0.35 × V_{CCIO}	V
I_I	Input pin leakage current	V_{IN} = 0 V or 3.3 V	–10	10	µA
V_{OH}	High-level output voltage	I_{OH} = –2 mA *(1)*	V_{CCIO} – 0.45		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(2)*		0.45	V

Table 18–9. 3.3-V PCI Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		3.0	3.3	3.6	V
V_{IH}	High-level input voltage		0.5 × V_{CCIO}		V_{CCIO} + 0.5	V

Table 18–9. 3.3-V PCI Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{IL}	Low-level input voltage		–0.5		0.3 × V_{CCIO}	V
I_I	Input pin leakage current	0 < V_{IN} < V_{CCIO}	–10		10	µA
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 18–10. 3.3-V PCI-X Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
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		0.35 × V_{CCIO}	V
V_{IPU}	Input pull-up voltage		0.7 × V_{CCIO}			V
I_{IL}	Input pin leakage current	0 < V_{IN} < V_{CCIO}	–10.0		10.0	µA
V_{OH}	High-level output voltage	I_{OUT} = –500 µA	0.9 × V_{CCIO}			V
V_{OL}	Low-level output voltage	I_{OUT} = 1500 µA			0.1 × V_{CCIO}	V
L_{PIN}	Pin Inductance				15.0	nH

Table 18–11. 3.3-V LVDS I/O Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{OD}	Differential output voltage	R_L = 100 Ω	250		450	mV
V_{OD}	Change in VOD 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

Table 18–11. 3.3-V LVDS I/O Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{OS}	Change in VOS 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 (external to APEX 20K devices)		90	100	110	Ω

Table 18–12. GTL+ I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{TT}	Termination voltage		1.35	1.5	1.65	V
V_{REF}	Reference voltage		0.88	1.0	1.12	V
V_{IH}	High-level input voltage		V_{REF} + 0.1			V
V_{IL}	Low-level input voltage				V_{REF} – 0.1	V
V_{OL}	Low-level output voltage	I_{OL} = 36 mA *(2)*			0.65	V

Table 18–13. SSTL-2 Class I Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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		V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.18	V

Table 18–13. SSTL-2 Class I Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{OH}	High-level output voltage	I_{OH} = –7.6 mA *(1)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 7.6 mA *(2)*			V_{TT} – 0.57	V

Table 18–14. SSTL-2 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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		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} = –15.2 mA *(1)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 15.2 mA *(2)*			V_{TT} – 0.76	V

Table 18–15. SSTL-3 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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} = –8 mA *(1)*	V_{TT} + 0.6			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA *(2)*			V_{TT} – 0.6	V

Table 18–16. SSTL-3 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O 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 *(1)*	V_{TT} + 0.8			V
V_{OL}	Low-level output voltage	I_{OL} = 16 mA *(2)*			V_{TT} – 0.8	V

Table 18–17. HSTL Class I I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		1.71	1.8	1.89	V
V_{TT}	Termination voltage		V_{REF} – 0.05	V_{REF}	V_{REF} + 0.05	V
V_{REF}	Reference voltage		0.68	0.75	0.90	V
V_{IH}	High-level input voltage		V_{REF} + 0.1		V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.1	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA *(1)*	V_{CCIO} – 0.4			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA *(2)*			0.4	V

Table 18–18. LVPECL Specifications (Part 1 of 2)

Symbol	Parameter	Minimum	Typical	Maximum	Units
V_{CCIO}	Output Supply Voltage	3.135	3.3	3.465	V
V_{IH}	Low-level input voltage	1,300		1,700	mV
V_{IL}	High-level input voltage	2,100		2,600	mV
V_{OH}	Low-level output voltage	1,450		1,650	mV

Table 18–18. LVPECL Specifications (Part 2 of 2)

Symbol	Parameter	Minimum	Typical	Maximum	Units
V_{OL}	High-level output voltage	2,275		2,420	mV
V_{ID}	Input voltage differential	400	600	950	mV
V_{OD}	Output voltage differential	625	800	950	mV
t_r, t_f	Rise and fall time (20 to 80%)	85		325	ps
t_{DSKEW}	Differential skew			25	ps
t_O	Output load		150		Ω
R_L	Receiver differential input resistor		100		Ω

Table 18–19. 3.3-V AGP I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		3.15	3.3	3.45	V
V_{REF}	Reference voltage		$0.39 \times V_{CCIO}$		$0.41 \times V_{CCIO}$	V
V_{IH}	High-level input voltage		$0.5 \times V_{CCIO}$		$V_{CCIO} + 0.5$	V
V_{IL}	Low-level input voltage				$0.3 \times V_{CCIO}$	V
V_{OH}	High-level output voltage	$I_{OUT} = –500\ \mu A$	$0.9 \times V_{CCIO}$		3.6	V
V_{OL}	Low-level output voltage	$I_{OUT} = 1500\ \mu A$			$0.1 \times V_{CCIO}$	V
I_I	Input pin leakage current	$0 < V_{IN} < V_{CCIO}$	–10		10	µA

Table 18–20. CTT I/O Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{CCIO}	I/O supply voltage		3.0	3.3	3.6	V
V_{TT}/V_{REF} *(3)*	Termination and reference voltage		1.35	1.5	1.65	V
V_{IH}	High-level input voltage		$V_{REF} + 0.2$			V

Table 18–20. CTT I/O Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{IL}	Low-level input voltage				V_{REF} – 0.2	V
I_I	Input pin leakage current	$0 < V_{IN} < V_{CCIO}$	–10		10	µA
V_{OH}	High-level output voltage	I_{OH} = –8 mA *(1)*	V_{REF} + 0.4			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA *(2)*			V_{REF} – 0.4	V
I_O	Output leakage current (when output is high Z)	$GND \leq V_{OUT} \leq V_{CCIO}$	–10		10	µA

Notes to Tables 18–5 through 18–20:

(1) The I_{OH} parameter refers to high-level output current.

(2) The I_{OL} parameter refers to low-level output current. This parameter applies to open-drain pins as well as output pins.

(3) V_{REF} specifies center point of switching range.

Figure 18–1 shows the output drive characteristics of HardCopy APEX devices.

Figure 18–1. Output Drive Characteristics of HardCopy APEX Devices

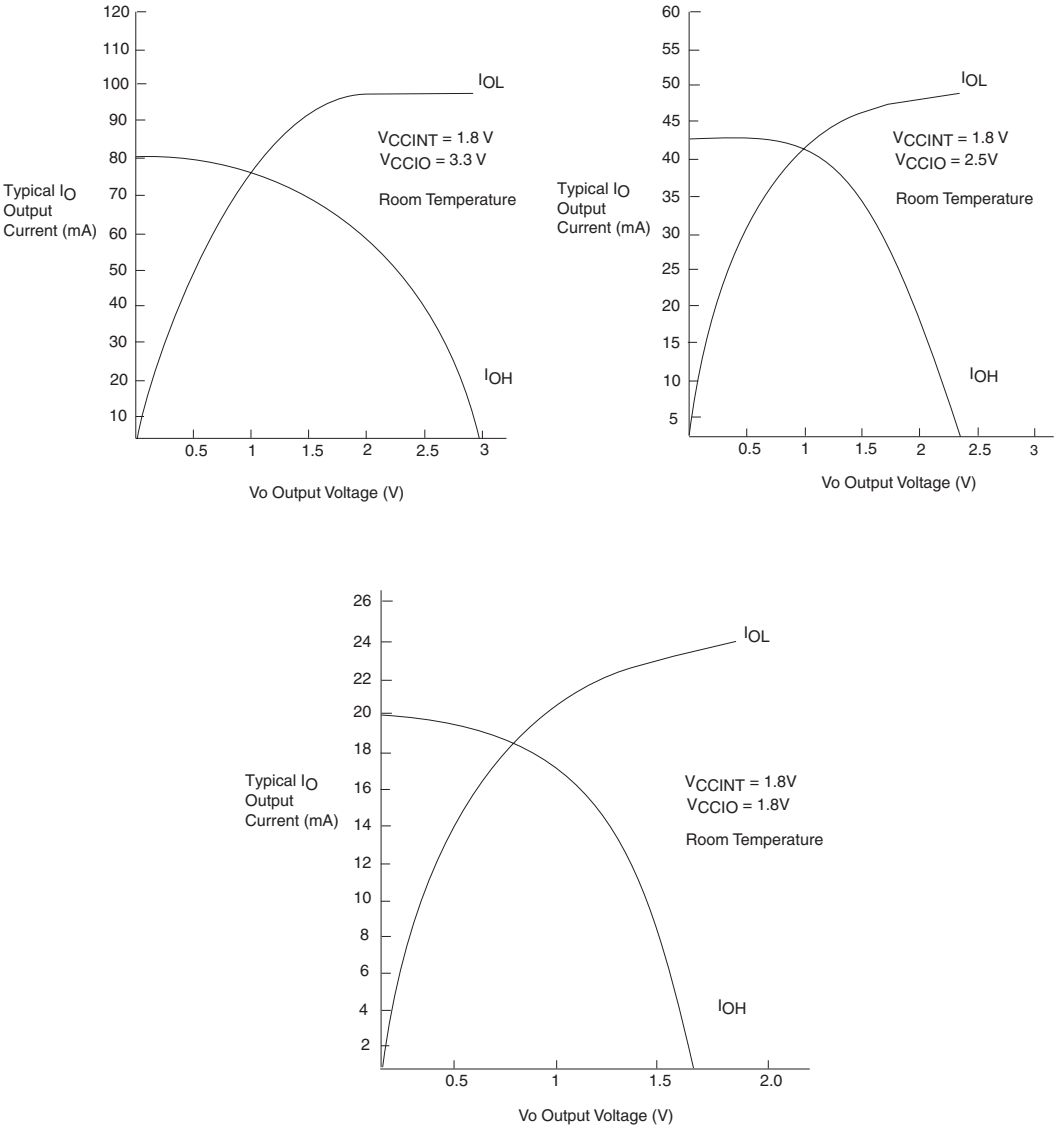
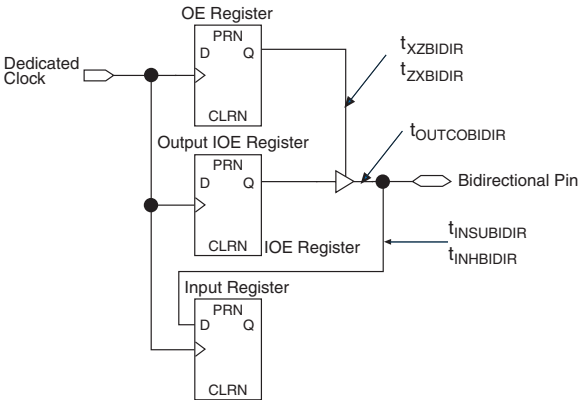


Figure 18–2 shows the timing model for bidirectional I/O pin timing.

Figure 18–2. Synchronous Bidirectional Pin External Timing



Tables 18–21 and 18–22 describe HardCopy APEX device external timing parameters.

Table 18–21. HardCopy APEX Device External Timing Parameters *Note (1)*

Symbol	Clock Parameter	Conditions
t_{INSU}	Setup time with global clock at IOE register	
t_{INH}	Hold time with global clock at IOE register	
t_{OUTCO}	Clock-to-output delay with global clock at IOE output register	C1 = 35 pF
$t_{INSUPLL}$	Setup time with PLL clock at IOE input register	
t_{INHPLL}	Hold time with PLL clock at IOE input register	
$t_{OUTCOPLL}$	Clock-to-output delay with PLL clock at IOE output register	C1 = 35 pF

Table 18–22. HardCopy APEX Device External Bidirectional Timing Parameters (Part 1 of 2) *Note (1)*

Symbol	Parameter	Condition
$t_{INSUBIDIR}$	Setup time for bidirectional pins with global clock at LAB-adjacent input register	
$t_{INHBIDIR}$	Hold time for bidirectional pins with global clock at LAB-adjacent input register	
$t_{OUTCOBIDIR}$	Clock-to-output delay for bidirectional pins with global clock at IOE register	C1 = 35 pF
$t_{XZBIDIR}$	Synchronous output enable register to output buffer disable delay	C1 = 35 pF

Table 18–22. HardCopy APEX Device External Bidirectional Timing Parameters (Part 2 of 2) Note (1)		
Symbol	**Parameter**	**Condition**
$t_{ZXBIDIR}$	Synchronous output enable register to output buffer enable delay	C1 = 35 pF
$t_{INSUBIDIRPLL}$	Setup time for bidirectional pins with PLL clock at LAB-adjacent input register	
$t_{INHBIDIRPLL}$	Hold time for bidirectional pins with PLL clock at LAB-adjacent input register	
$t_{OUTCOBIDIRPLL}$	Clock-to-output delay for bidirectional pins with PLL clock at IOE register	C1 = 35 pF
$t_{XZBIDIRPLL}$	Synchronous output enable register to output buffer disable delay with PLL	C1 = 35 pF
$t_{ZXBIDIRPLL}$	Synchronous output enable register to output buffer enable delay with PLL	C1 = 35 pF

Note to Tables 18–21 and 18–22:
(1) These timing parameters are sample-tested only.

Tables 18–23 and 18–24 show the external timing parameters for HC20K1500 devices.

Table 18–23. HC20K1500 External Timing Parameters Note (1)			
Symbol	**Min**	**Max**	**Unit**
t_{INSU}	2.0		ns
t_{INH}	0.0		ns
t_{OUTCO}	2.0	5.0	ns
$t_{INSUPLL}$	3.3		ns
t_{INHPLL}	0.0		ns
$t_{OUTCOPLL}$	0.5	2.1	ns

Table 18–24. HC20K1500 External Bidirectional Timing Parameters (Part 1 of 2) Note (1)			
Symbol	**Min**	**Max**	**Unit**
$t_{INSUBIDIR}$	1.9		ns
$t_{INHBIDIR}$	0.0		ns
$t_{OUTCOBIDIR}$	2.0	5.0	ns
$t_{XZBIDIR}$		7.1	ns
$t_{ZXBIDIR}$		7.1	ns
$t_{INSUBIDIRPLL}$	3.9		ns

Table 18–24. HC20K1500 External Bidirectional Timing Parameters (Part 2 of 2) Note (1)

Symbol	Min	Max	Unit
$t_{INHBIDIRPLL}$	0.0		ns
$t_{OUTCOBIDIRPLL}$	0.5	2.1	ns
$t_{XZBIDIRPLL}$		4.2	ns
$t_{ZXBIDIRPLL}$		4.2	ns

Note to Tables 18–23 and 18–24:

(1) Timing information is preliminary. Final timing information will be available in a future version of this data sheet.

Section IV. General HardCopy Series Design Considerations

This section provides information on hardware design considerations for HardCopy® series devices.

This section contains the following:

- Chapter 19, Design Guidelines for HardCopy Series Devices
- Chapter 20, Power-Up Modes & Configuration Emulation in HardCopy Series Devices

Revision History

The table below shows the revision history for Chapters 19 through 20.

Chapter(s)	Date / Version	Changes Made
Chapter 19	March 2006	Formerly chapter 14; no content change.
	October 2005, v3.1	• Graphic updates • Minor edits
	May 2005, v3.0	Updated the Using a FIFO Buffer section.
	January 2005, v2.0	• Chapter title changed to *Design Guidelines for HardCopy Series Devices*. • Updated *Quartus® II Software Supported Versions* • Updated *HardCopy® Design Center Support* • Updated heading U*sing a Double Synchronizer for Single-Bit Data Transfer* • Added *Stratix® II support for a global or regional clock* • Added *Support for Stratix II and HardCopy II to Mixing Clock Edges*
	August 2003, v1.1	Edited hierarchy of section headings.
	May 2003, v1.0	Initial release

Chapter(s)	Date / Version	Changes Made
Chapter 20	May 2006, v2.2	• Updated Table 20–1, Table 20–3, and Table 20–5.
	March 2006, v2.1	• Formerly chapter 16. • Re-organized *HardCopy Power-Up Options* section to eliminate redundancy. • Updated Figure 20–1, Figure 20–2, Figure 20–3. • Updated Table 20–1, Table 20–2, Table 20–3, Table 20–4, Table 20–5 Table 20–7. • Added *Power Up Options Summary When Designing With HardCopy Series Devices* section.
	October 2005, v2.0	Moved from Chapter 15 to Chapter 16 in Hardcopy Series Device Handbook 3.2
	January 2005, v2.0	• Chapter title changed to *Power-Up Modes & Configuration Emulation in HardCopy Series Devices*. • Added HardCopy II device information. • Updated external resistor requirements depending on chip configuration. • Added reference to some control and option pins that carry over functions from the FPGA design and affect the HardCopy power up. • Updated information on which HardCopy devices do not support emulation mode. • Added Table 16–9 which lists what power up options are supported by FPGAs and their HardCopy counterpart. • Added “Replacing One FPGA With One HardCopy Series Device”, “Replacing One or More FPGAs With One or More HardCopy Series Devices in a Multiple-Device Configuration Chain”, and “Replacing all FPGAs with HardCopy Series Devices in a Multiple-Device Configuration Chain” sections, including Tables 16–10 and 20–8, highlighting power up recommendations for each HardCopy series family.
	June 2003, v1.0	Initial release of Chapter 16, Power-Up Modes & Configuration Emulation in HardCopy Series Devices.

19. Design Guidelines for HardCopy Series Devices

H51011-3.1

Introduction

HardCopy® series devices provide dramatic cost savings, performance improvement, and reduced power consumption over their programmable counterparts. In order to ensure the smoothest possible transfer from the FPGA device to the equivalent HardCopy series device, you must meet certain design rules while the FPGA implementation is still in progress. A design that meets standard, accepted coding styles for FPGAs, adheres easier to recommended guidelines. This chapter describes some common situations that you should avoid. It also provides alternatives on how to design in these situations.

Design Assistant Tool

The Design Assistant tool in the Quartus® II software allows you to check for any potential design problems early in the design process. The Design Assistant is a design-rule checking tool that checks the compiled design for adherence to Altera® recommended design guidelines. It provides a summary of the violated rules that exist in a design together with explicit details of each violation instance. You can customize the set of rules that the tool checks to allow some rule violations in your design. This is useful if it is known that the design violates a particular rule that is not critical. However, for HardCopy design, you must enable all of the Design Assistant rules. All Design Assistant rules are enabled and run by default in the Quartus II software when using the HardCopy Timing Optimization Wizard in the **HardCopy Utilities** (Project menu). The HardCopy Advisor in the Quartus II software also checks to see if the Design Assistant is enabled.

The Design Assistant classifies messages using the four severity levels described in Table 19–1.

Table 19–1. Design Assistant Message Severity Levels (Part 1 of 2)

Severity Level	Description
Critical	The rule violation described in the message critically affects the reliability of the design. Altera cannot migrate the design successfully to a HardCopy device without closely reviewing these violations.
High	The rule violation described in the message affects the reliability of the design. Altera must review the violation before the design is migrated to a HardCopy device.

Table 19–1. Design Assistant Message Severity Levels (Part 2 of 2)

Severity Level	Description
Medium	The rule violation described in the message may result in implementation complexity. The violation may impact the schedule or effort required to migrate the design to a HardCopy series device.
Information only	The message contains information regarding a design rule.

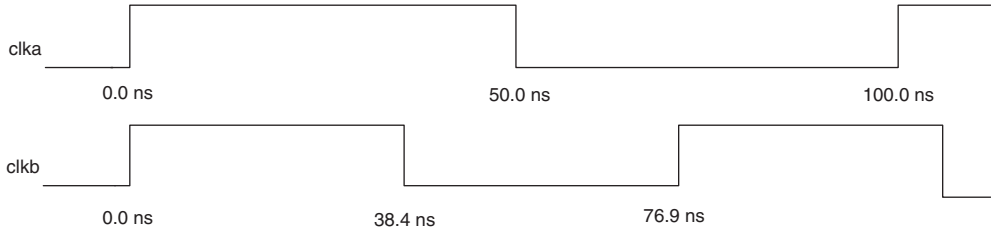
A design that adheres to Altera recommended design guidelines does not produce any critical, high, or medium level Design Assistant messages. If the Design Assistant generates these kinds of messages, Altera's HardCopy Design Center (which performs the migration) carefully reviews each message before considering implementing the FPGA design into a HardCopy design. After reviewing these messages with your design team, Altera may be able to implement the design in a HardCopy device. Informational messages are primarily for the benefit of the Altera HardCopy Design Center and are used to gather information about your design for the migration process from FPGA prototype to HardCopy production device.

Asynchronous Clock Domains

A design contains several clock sources, each driving a subsection of the design. A design subsection, driven by a single clock source is called a clock domain. The frequency and phase of each clock source can be different from the rest.

The timing diagram in Figure 19–1 shows two free-running clocks used to describe the nature of asynchronous clock domains. If the two clock signals do not have a synchronous, or fixed, relationship, they are asynchronous to each other. An example of asynchronous signals are two clock signals running at frequencies that have no obvious harmonic relationship.

Figure 19–1. Two Asynchronous Clock Signals *Notes (1), (2)*



Notes to Figure 19–1:
(1) clka = 10 MHz; clkb =13 MHz.
(2) Both clocks have 50% duty cycles.

In Figure 19–1, the clka signal is defined with a rising edge at 0.0 ns, a falling edge at 50 ns, and the next rising edge at 100 ns (1/10 MHz = 100 ns). Subsequent rising edges of clka are at 200 ns, 300 ns, 400 ns, and so on.

The clkb signal is defined with a rising edge at 0.0 ns, a falling edge at 38.45 ns, and the next rising edge at 76.9 ns. The subsequent rising edges of clkb are at 153.8 ns, 230.7 ns, 307.6 ns, 384.5 ns, and so on.

Not until the thousandth clock edge of clkb (1000 × 76.9 = 76,900 ns) or the 7,690th clock edge of clka (7,690 × 100 = 769,000 ns), does clka and clkb have coincident edges. It is very unlikely that these two clocks are intended to synchronize with each other every 76,900 ns, so these two clock domains are considered asynchronous to each other.

A more subtle case of asynchronous clock domains occurs when two clock domains have a very obvious frequency and phase relationship, especially when one is a multiple of the other. Consider a system with clocks running at 100 MHz and 50 MHz. The edges of one of these clocks are always a fixed distance away, in time, from the edges of the other clock. In this case, the clock domains may or may not be asynchronous, depending on what your original intention was regarding the interactions of these two clock domains.

Similarly, two clocks running at the same nominal frequency may be asynchronous to each other if there is no synchronization mechanism between them. For example two crystal oscillators, each running at 100 MHz on a PC board, have some frequency variations due to temperature fluctuations, and this may be different for each oscillator. This results in the two independent clock signals drifting in and out of phase with each other.

Transferring Data between Two Asynchronous Clock Domains

If two asynchronous clock domains need to communicate with each other, you need to consider how to reliably perform this operation. The following three examples shows how to transfer data between two asynchronous clock domains.:

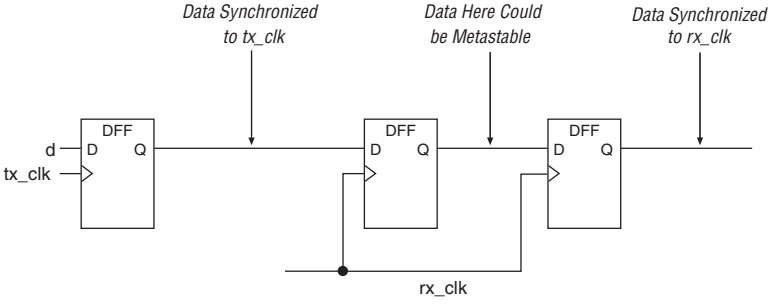
- Using a double synchronizer
- Using a first-in first-out (FIFO) buffer
- Using a handshake protocol

The choice of which to use depends on the particular application, the number of asynchronous signals crossing clock boundaries, and the resources available to perform the cross-domain transfers.

Using a Double Synchronizer for Single-Bit Data Transfer

Figure 19–2 shows a double synchronizer for single-bit data transfer consisting of a 2-bit shift register structure clocked by the receiving clock. The second stage of the shift register reduces the probability of metastability (unknown state) on the data output from the first register propagating through to the output of the second register. The data from the transmitting clock domain should come directly from a register. This technique is recommended only if single-data signals (for example, non-data buses) need to be transferred across clock domains. This is because it is possible that some bits of a data bus are captured in one clock cycle while other bits get captured in the next. More than two stages of the synchronizer circuit can be used at the expense of increased latency. The benefit of more stages is that the mean time between failures (MTBF) is increased with each additional stage.

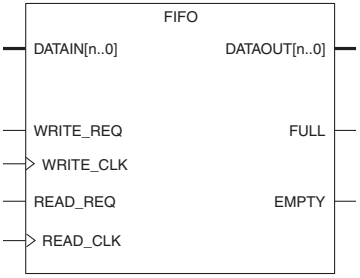
Figure 19–2. A Double Synchronizer Circuit



Using a FIFO Buffer

The advantage of using a FIFO buffer, shown in Figure 19–3, is that Altera's MegaWizard® Plug-In Manager makes it very easy to design a FIFO buffer. A FIFO buffer is useful when you need to transfer a data bus signal across an asynchronous clock domains, and it is beneficial to temporary storage of this data. A FIFO buffer circuit should not generate any Design Assistant warnings unless an asynchronous clear is used in the circuit. An asynchronous clear in the FIFO buffer circuit results in a warning stating that a reset signal generated in one clock domain is not being synchronized before being used in another clock domain. This occurs because a dual-clock FIFO megafunction only has one `aclr` pin to reset the entire FIFO buffer circuit. You cannot remove this warning in the case of a dual-clock FIFO buffer circuit. As a safeguard, Altera recommends using a reset signal that is synchronous to the clock domain of the write side of the FIFO buffer circuit.

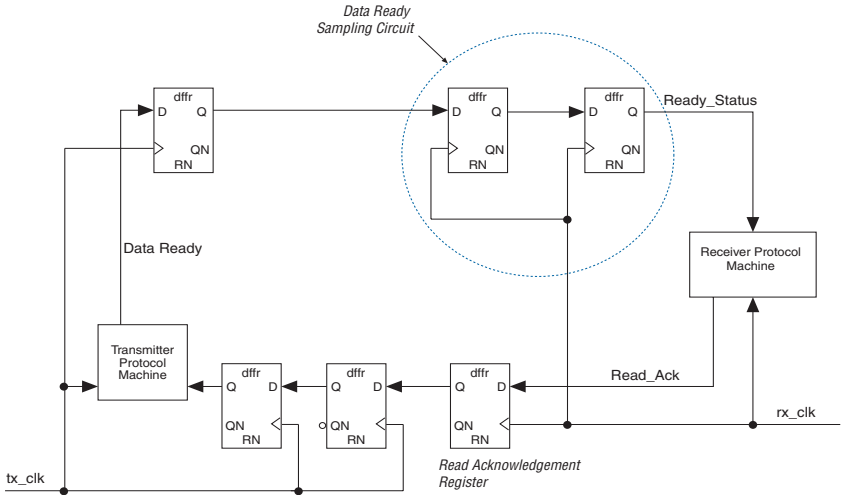
Figure 19–3. A FIFO Buffer



Using a Handshake Protocol

A handshake protocol circuit uses a small quantity of logic cells to implement and guarantees that all bits of a data bus crossing asynchronous clock domains are registered by the same clock edge in the receiving clock domain. This circuit, shown in Figure 19–4, is best used in cases where there is no memory available to be used as FIFO buffers, and the design has many data buses to transfer between clock domains.

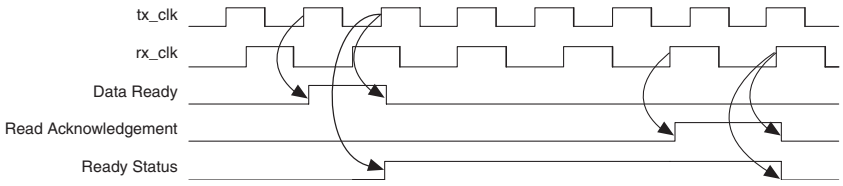
Figure 19–4. A Handshake Protocol Circuit



This circuit is initiated by a data ready signal going high in the transmitting clock domain `tx_clk`. This is clocked into the data ready sampling registers and causes the `Ready_Status` signal to go high. The `Data Ready` signal must be long enough in duration so that it is successfully sampled in the receiver domain. This is important if the `rx_clk` signal is slower than `tx_clk`.

At this point, the receiving clock domain `rx_clk` can read the data from the transmitting clock domain `tx_clk`. After this read operation has finished, the receiving clock domain `rx_clk` generates a synchronous `Read_Ack` signal, which gets registered by the read acknowledge register. This registered signal is sampled by the `Read_Ack` sampling circuit in the transmitter domain. The `Read_Ack` signal must be long enough in duration so that it is successfully sampled in the transmitter domain. This is important if the transmitter clock is slower than the receiver clock. After this event, the data transfer between the two asynchronous domains is complete, as shown by the timing diagram in Figure 19–5.

Figure 19–5. Data Transfer Between Two Asynchronous Clock Domains



Gated Clocks

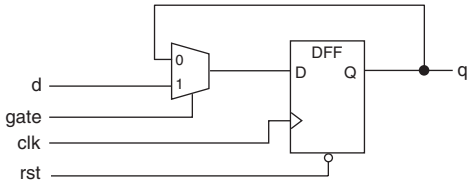
Clock gating is sometimes used to "turn off" parts of a circuit to reduce the total power consumption of a device. The gated clock signal prevents any of the logic driven by it from switching so the logic does not consume any power. This works best if the gating is done at the root of the clock tree. If the clock is gated at the leaf-cell level (for example, immediately before the input to the register), the device does not save much power because the whole clock network still toggles. The disadvantage in using this type of circuit is that it can lead to unexpected glitches on the resultant gated clock signal if certain rules are not adhered to. Rules are provided in the following subsections:

- Preferred Clock Gating Circuit
- Alternative Clock Gating Circuits
- Inverted Clocks
- Clocks Driving Non-Clock Pins
- Clock Signals Should Use Dedicated Clock Resources
- Mixing Clock Edges

Preferred Clock Gating Circuit

The preferred way to gate a clock signal is to use a purely synchronous circuit, as shown in Figure 19–6. In this implementation, the clock is not gated at all. Rather, the data signal into a register is gated. This circuit is sometimes represented as a register with a clock enable (CE) pin. This circuit is not sensitive to any glitches on the gate signal, so it gets generated directly from a register or any complex combinational function. The constraints on the gate or clock enable signal are exactly the same as those on the 'd' input of the gating multiplexer. Both of these signals must meet the setup and hold times of the register that they feed into.

Figure 19–6. Preferred Clock-Gating Circuit



This circuit only takes a few lines of VHDL or Verilog hardware description language (HDL) to describe.

The following is a VHDL code fragment for a synchronous clock gating circuit.

```
architecture rtl of vhdl_enable is
begin
    process (rst, clk)
    begin
        if (rst = '0') then
            q <= '0';
        elsif clk'event and clk = '1' then
            if (gate = '1') then
                q <= d;
            end if;
        end if;
    end process;
end rtl;
```

The following is a Verilog HDL code fragment for a synchronous clock gating circuit.

```
always @ (posedge clk or negedge rst)
    begin
        if (!rst)
            q <= 1'b0;
        else if (gate)
            q <= d;
        else
            q <= q;
    end
```

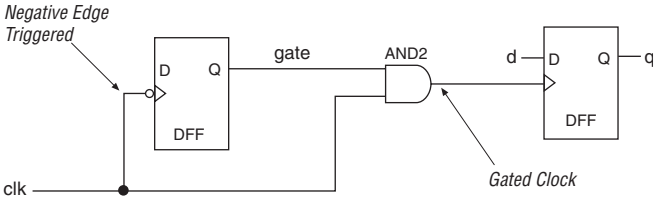
Alternative Clock Gating Circuits

If a clock gating circuit is absolutely necessary in the design, one of the following two circuits may also be used. The Design Assistant does not flag a violation for these circuits.

Clock Gating Circuit Using an AND Gate

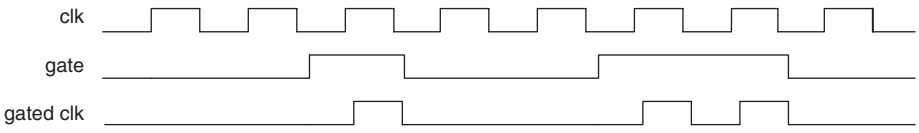
Designs can use a two input AND gate for a gated clock signal that feeds into positive-edge-triggered registers. One input to the AND gate is the original clock signal. The other input to the AND gate is the gating signal, which should be driven directly from a register clocked by the negative edge of the same original clock signal. Figure 19–7 shows this type of circuit.

Figure 19–7. Clock Gating Circuit Using an AND Gate



Because the register that generates the gate signal is triggered off of the negative edge of the same clock, the effect of using both edges of the same clock in the design should be considered. The timing diagram in Figure 19–8 shows the operation of this circuit. The `gate` signal occurs after the negative edge of the clock and comes directly from a register. The logical AND of this `gate` signal, with the original un-inverted clock, generates a clean clock signal.

Figure 19–8. Timing Diagram for Clock Gating Circuit Using an AND Gate

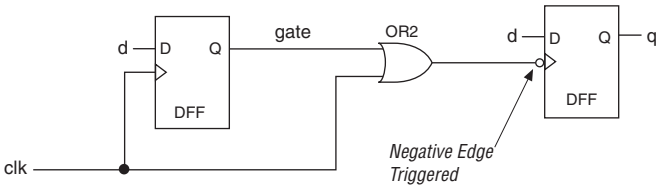


If the delay between the register that generates the `gate` signal and the `gate` input to the AND gate is greater than the low period of the clock, (one half of the clock period for a 50% duty cycle clock), the clock pulse width is narrowed.

Clock Gating Circuit Using an OR Gate

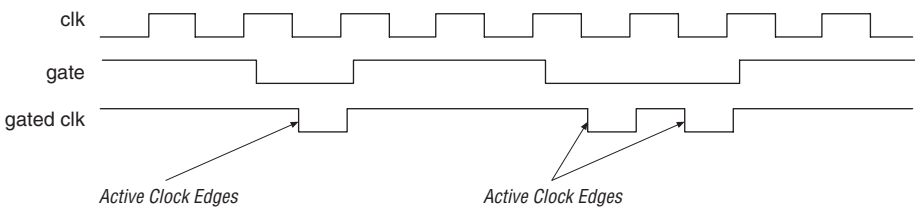
Use a two-input OR gate for a gated clock signal that feeds into a negative-edge-triggered register. One input to the OR gate is the original clock signal. The other input to the OR gate is the gating signal, which should be driven directly from a register clocked by the positive edge of the same original clock signal. Figure 19–9 shows this circuit.

Figure 19–9. Clock Gating Circuit Using an OR Gate



Because the register that generates the gate signal is triggered off the positive edge of the same clock, you need to consider the effect of using both edges of the same clock in your design. The timing diagram in Figure 19–10 shows the operation of this circuit. The `gate` signal occurs after the positive edge of the clock, and comes directly from a register. The logical OR of this `gate` signal with the original, un-inverted clock generates a clean clock signal. This clean, gated clock signal should only feed registers that use the negative edge of the same clock.

Figure 19–10. Timing Diagram for Clock Gating Circuit Using an OR Gate



If the delay between the register that generates the `gate` signal and the `gate` input to the AND gate is greater than the low period of the clock, (one half of the clock period for a 50% duty cycle clock), the clock pulse width is narrowed.

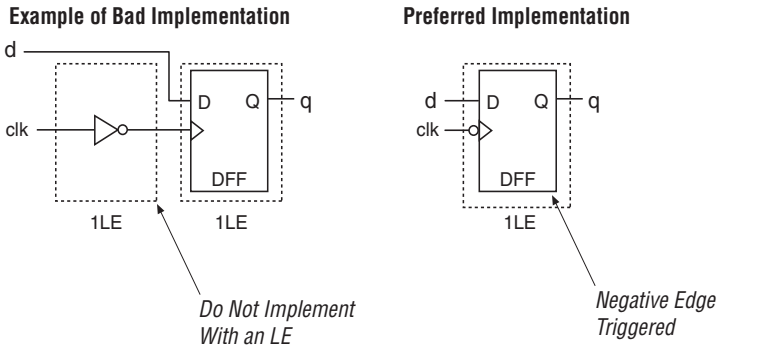
Altera recommends using a synchronous clock gating circuit because it is the only way to guarantee the duty cycle of the clock and to align the clock to the data.

Inverted Clocks

A design may require both the positive edge and negative edge of a clock, as shown in Figure 19–11. In Altera FPGAs, each logic element (LE) has a programmable clock inversion feature. Use this feature to generate an inverted clock.

Do not instantiate a LE look-up-table (LUT) configured as an inverter to generate the inverted clock signal.

Figure 19–11. An LE LUT Configured as an Inverter



Using a LUT to perform the clock inversion may lead to a clock insertion delay and skew, which poses a significant challenge to timing closure of the design. It also consumes more device resources than are necessary. refer to "Mixing Clock Edges" on page 19–14 for more information on this topic.

Do not generate schematics or register transfer level (RTL) code that instantiates LEs used to invert clocks. Instead, let the synthesis tool decide on the implementation of inverted clocks.

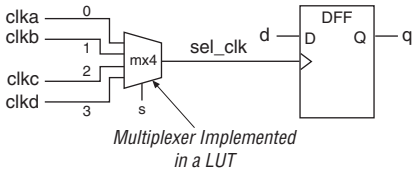
Clocks Driving Non-Clock Pins

As a general guideline, clock sources should only be used to drive the register clock pins. There are exceptions to this rule, but every effort should be taken to minimize these exceptions or remove them altogether.

One category of exception is for various gated clocks, which are described in "Preferred Clock Gating Circuit" on page 19–7.

You should avoid another exception, when possible, in which you use a clock multiplexer circuit to select one clock from a number of different clock sources, to drive non-clock pins. This type of circuit introduces complexity into the static timing analysis of HardCopy and FPGA implementations. For example, as shown in Figure 19–12, in order to investigate the timing of the `sel_clk` clock signal, it is necessary to make a clock assignment on the multiplexer output pin, which has a specific name. This name may change during the course of the design unless you preserve the node name in the Quartus II software settings. Refer to the Quartus II Help for more information on preserving node names.

Figure 19–12. A Circuit Showing a Multiplexer Implemented in a LUT



In the FPGA, a clock multiplexing circuit is built out of one or more LUTs, and the resulting multiplexer output clock may possibly no longer use one of the dedicated clock resources. Consequently, the skew and insertion delay of this multiplexed clock is potentially large, adversely impacting performance. The Quartus II Design Assistant traces clocks to their destination and, if it encounters a combinational gate, it issues a gated clock warning.

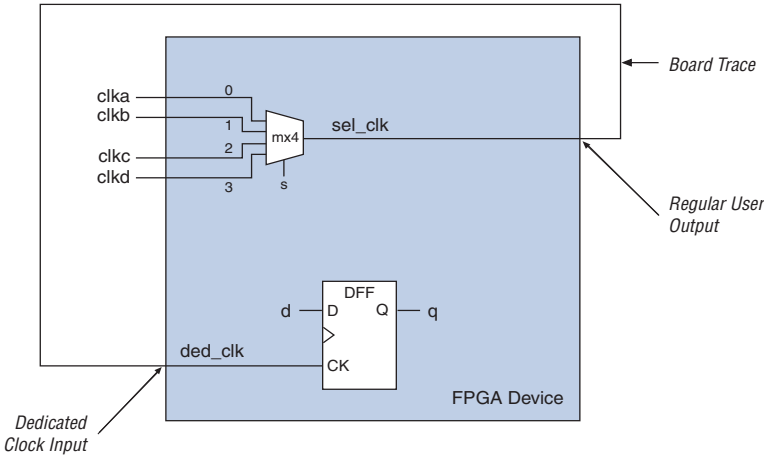
If the design requires this type of functionality, ensure that the multiplexer output drives one of the global routing resources in the FPGA. For example, this output should drive a fast line in an APEX™ 20KE device, or a global or regional clock in a Stratix® or Stratix II device.

Enhanced PLL Clock Switchover

Clock source multiplexing can be done using the enhanced PLL clock switchover feature in Stratix and Stratix II FPGAs, and in HardCopy Stratix and HardCopy II structured ASICs. The clock switchover feature allows multiple clock sources to be used as the reference clock of the enhanced PLL. The clock source switchover can be controlled by an input pin or internal logic. This generally eliminates the need for routing a multiplexed clock signal out to a board trace and bringing it back into the device, as shown in Figure 19–13.

Routing a multiplexed clock signal as shown in Figure 19–13 is only intended for APEX 20K FPGA and HardCopy APEX devices. This alternative to a clock multiplexing circuit ensures that a global clock resource is used to distribute the clock signal over the entire device by routing the multiplexed clock signal to a primary output pin. Outside of the device, this output pin then drives one of the dedicated clock inputs of the same device, possibly through a phase-locked loop (PLL) to reduce the clock insertion delay. Although there is a large delay through the multiplexing circuit and external board trace, the resulting clock skew is very small because the design uses the dedicated clock resource for the selected clock signal. The advantage that this circuit has over the other implementations is that the timing analysis becomes very simple, with only a single-clock domain to analyze, whose source is a primary input pin to the APEX 20K FPGA or HardCopy APEX device.

Figure 19–13. Routing a Multiplexed Clock Signal to a Primary Output Pin



Clock Signals Should Use Dedicated Clock Resources

All clock signals in a design should be assigned to the global clock networks that exist in the target FPGA. Clock signals that are mapped to use non-dedicated clock networks can negatively affect the performance of the design. This is because the clock must be distributed using regular FPGA routing resources, which can be slower and have a larger skew than the dedicated clock networks. If your design has more clocks than are available in the target FPGA, you should consider reducing the number of clocks, so that only dedicated clock resources are used in the FPGA for clock distribution. If you need to exceed the number of dedicated clock resources, implement the clock with the lowest fan-out

with regular (non-clock network) routing resources. Give priority to the fastest clock signals when deciding how to allocate dedicated clock resources.

In the Quartus II software, you can use the global signal logic option to specify that a clock signal is a global signal. You can also use the auto global clock logic option to allow the Fitter to automatically choose clock signals as global signals.

Altera recommends using the FPGA's built-in clock networks because they are pre-routed for low skew and for short insertion delay.

Mixing Clock Edges

You can use both edges of a single clock in a design. An example where both edges of a clock must be used in order to get the desired functionality is with a double data rate (DDR) memory interface. In Stratix II, Stratix, HardCopy II, and HardCopy Stratix devices, this interface logic is built into the I/O cell of the device, and rigorous simulation and characterization is performed on this interface to ensure its robustness. Consequently, this circuitry is an exception to the rule of using both edges of a clock. However, for general data transfers using generic logic resources, the design should only use a single edge of the clock. A circuit needs to use both edges of a single clock, then the duty cycle of the clock has to be accurately described to the Static Timing Analysis tool, otherwise inaccurate timing analysis could result. Figure 19–14 shows two clock waveforms. One has a 50% duty-cycle, the other has a 10% duty cycle.

Figure 19–14. Clock Waveforms with 50% & 10% Duty Cycles

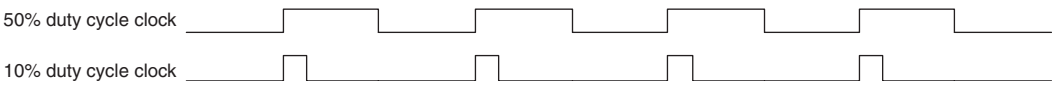


Figure 19–15 shows a circuit that uses only the positive edge of the clock. The distance between successive positive clock edges is always the same, for example, the clock period. For this circuit, the duty cycle of the clock has no effect on the performance of the circuit.

Figure 19–15. Circuit Using the Positive Edge of a Clock

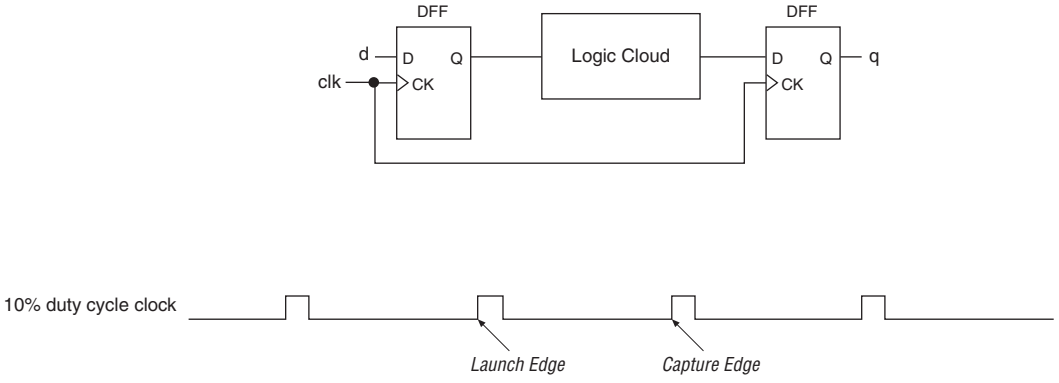
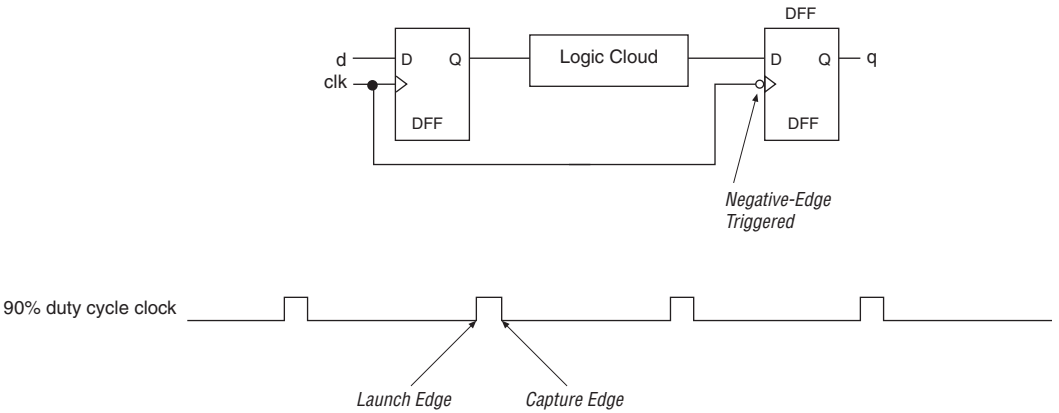


Figure 19–16 shows a circuit that used the positive clock edge to launch data and the negative clock edge to capture this data. Since this particular clock has a 10% duty cycle, the amount of time between the launch edge and capture edge is small. This small gap makes it difficult for the synthesis tool to optimize the cloud of logic so that no setup-time violations occur at the capture register.

Figure 19–16. Circuit Using the Positive & Negative Edges of a Clock



If you design a circuit that uses both clock edges, then you could get the Design Assistant warning "Registers are Triggered by Different Edges of Same Clock." You do not get this warning under the following conditions:

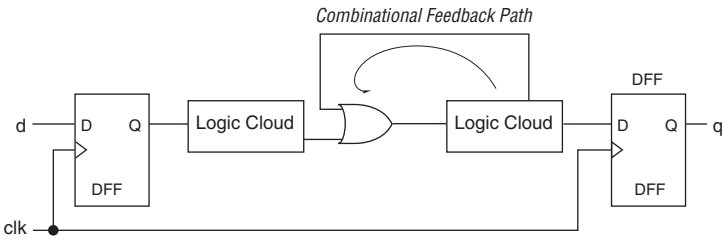
- If the opposite clock edge is used in a clock gating circuit
- A double data rate (DDR) memory interface circuit is used

Try to only use a single edge of a clock in a design.

Combinational Loops

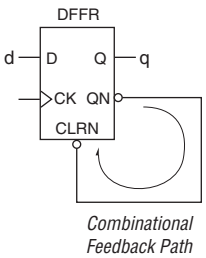
A combinational loop exists (Figure 19–17) if the output of a logic gate (or gates) feeds back to the input of the same gate without first encountering a register. A design should not contain any combinational loops.

Figure 19–17. A Circuit Using a Combinational Loop



It is also possible to generate a combinational loop using a register (Figure 19–18) if the register output pin drives the reset pin of the same register.

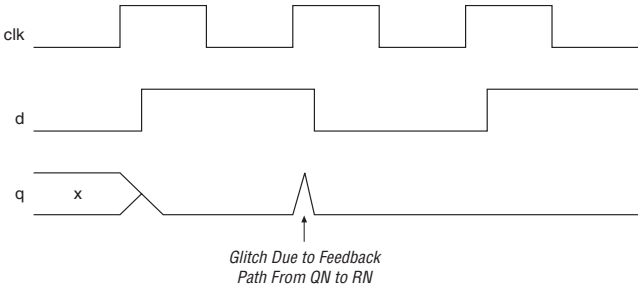
Figure 19–18. Generation of a Combinational Loop Using a Register



The timing diagram for this circuit is shown in Figure 19–19. When a logic 1 value on the register D input is clocked in, the logic 1 value appears on the Q output pin after the rising clock edge. The same clock

event causes the QN output pin to go low, which in turn, causes the register to be reset through RN. The Q register output consequently goes low. This circuit may not operate if there isn't sufficient delay in the QN-to-RN path, and is not recommended.

Figure 19–19. Timing Diagram for the Circuit Shown in Figure 19–18



Combinational feedback loops are either intentionally or unintentionally introduced into a design. Intentional feedback loops are typically introduced in the form of instantiated latches. An instantiated latch is an example of a combinational feedback loop in Altera FPGAs because its function has to be built out of a LUT, and there are no latch primitives in the FPGA logic fabric. Unintentional combinational feedback loops usually exist due to partially specified IF-THEN or CASE constructs in RTL. The Design Assistant checks your design for these circuit structures. If any are discovered, you should investigate and implement a fix to your RTL to remove unintended latches, or re-design the circuit so that no latch instantiation is required. In Altera FPGAs, many registers are available, so there should never be any need to use a latch.

Combinational loops can cause significant stability and reliability problems in a design because the behavior of a combinational loop often depends on the relative propagation delays of the loop's logic. This combinational loop circuit structure behaves differently under different operation conditions. A combinational loop is asynchronous in nature, and EDA tools operate best with synchronous circuits.

A storage element such as a level-sensitive latch or an edge-triggered register has particular timing checks associated with it. For example, there is a setup-and-hold requirement for the data input of an edge-triggered register. Similarly, there is also a setup-and-hold timing requirement for the data to be stable in a transparent latch when the gate signal turns the latch from transparent to opaque. When latches are built

out of combinational gates, these timing checks do not exist, so the static timing analysis tool is not able to perform the necessary checks on these latch circuits.

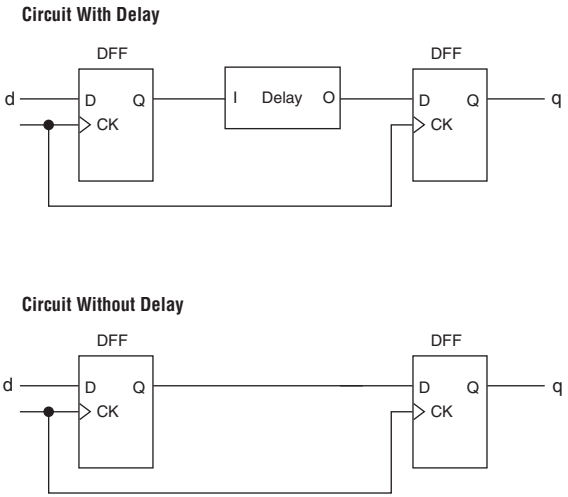
Check your design for intentional and unintentional combinational loops, and remove them.

Intentional Delays

Altera does not recommend instantiating a cell that does not benefit a design. This type of cell only delays the signal. For a synchronous circuit that uses a dedicated clock in the FPGA (Figure 19–20), this delay cell is not needed. In an ASIC, a delay cell is used to fix hold-time violations that occur due to the clock skew between two registers, being larger than the data path delay between those same two registers. The FPGA is designed with the clock skew and the clock-to-Q time of the FPGA registers in mind, to ensure that there is no need for a delay cell.

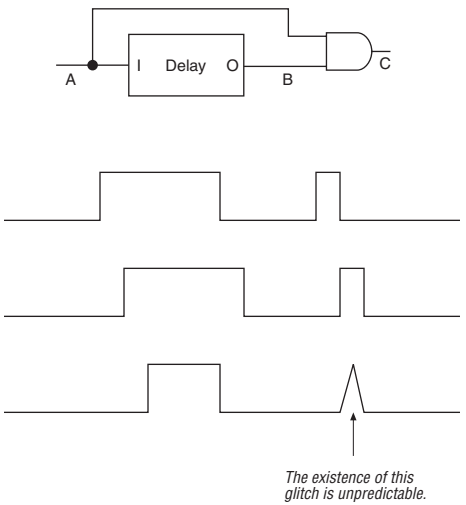
Figure 19–20 shows two versions of the same shift registers. Both circuits operate identically. The first version has a delay cell, possibly implemented using a LUT, in the data path from the Q output of the first register to the D input of the second register. The function of the delay cell is a non-inverting buffer. The second version of this circuit also shows a shift register function, but there is no delay cell in the data path. Both circuits operate identically.

Figure 19–20. Shift Register With & Without an Intentional Delay



If delay chains exist in a design, they are possibly symptomatic of an asynchronous circuit. One such case is shown in the circuit in Figure 19–21. This circuit relies on the delay between two inputs of an AND gate to generate a pulse on the AND gate output. The pulse may or may not be generated, depending on the shape of the waveform on the A input pin.

Figure 19–21. A Circuit & Corresponding Timing Diagram Showing a Delay Chain



Using delay chains can cause various design problems, including an increase in a design's sensitivity to operating conditions and a decrease in design reliability.

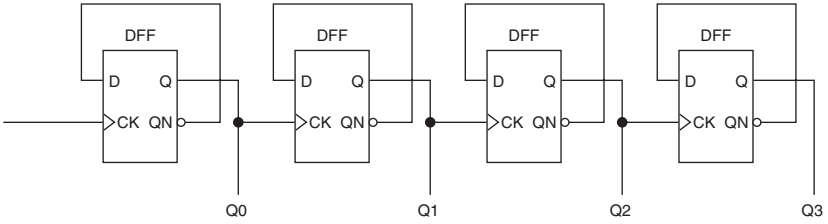
Be aware that not all cases of delay chains in a design are due to asynchronous circuitry. If the Design Assistant report states that you have delay chains that you are unaware of (or are not expecting), the delay chains may be a result of using pre-built intellectual property (IP) functions. Pre-built IP functions may contain delay chains which the Design Assistant reports. These functions are usually parameterizable, and have thousands of different combinations of parameter settings. The synthesis tool may not remove all unused LEs from these functions when particular parameter settings are used, but the resulting circuit is still synchronous. Check all Design Assistant delay chain warnings carefully.

☞ Avoid designing circuits that rely on the use of delay chains, and always carefully check any Design Assistant delay chain warnings.

Ripple Counters

Designs should not contain ripple counters. A ripple counter, shown in Figure 19–22, is a circuit structure where the Q output of the first counter stage drives into the clock input of the following counter stage. Each counter stage consists of a register with the inverted QN output pin feeding back into the D input of the same register.

Figure 19–22. A Typical Ripple Counter



This type of structure is used to make a counter out of the smallest amount of logic possible. However, the LE structure in Altera FPGA devices allows you to construct a counter using one LE per counter-bit, so there is no logic savings in using the ripple counter structure. Each stage of the counter in a ripple counter contributes some phase delay, which is cumulative in successive stages of the counter. Figure 19–23 shows the phase delay of the circuit in Figure 19–22.

Figure 19–23. Timing Diagram Showing Phase Delay of Circuit Shown in Figure 19–22

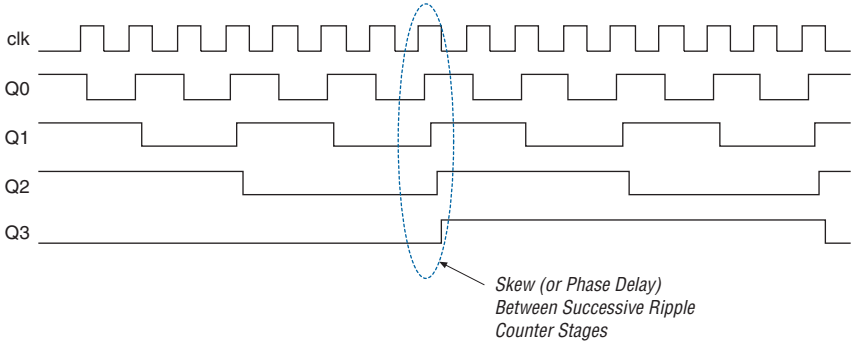
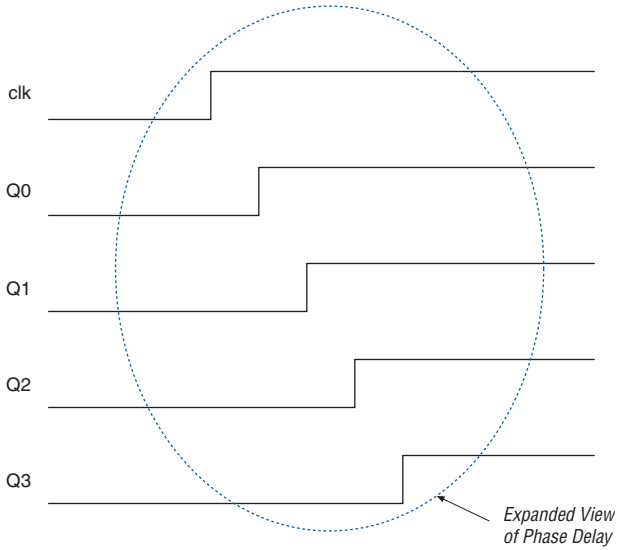


Figure 19–24 shows detailed view of the phase delay shown in Figure 19–23.

Figure 19–24. Detailed View of the Phase Delay Shown in Figure 19–23



This phase delay is problematic if the ripple counter outputs are used as clock signals for other circuits. Those other circuits are clocked by signals that have large skews.

Ripple counters are particularly challenging for static timing analysis tools to analyze as each stage in the ripple counter causes a new clock domain to be defined. The more clock domains that the static timing analysis tool has to deal with, the more complex and time-consuming the process becomes.

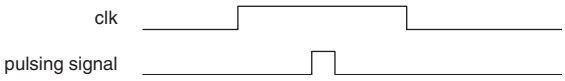
Altera recommends that you avoid using ripple counters under any circumstances.

Pulse Generators

A pulse generator is a circuit that generates a signal that has two or more transitions within a single clock period. Figure 19–25 shows an example of a pulse generator waveform.

For more information on pulse generators, refer to "Intentional Delays" on page 19–18.

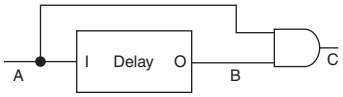
Figure 19–25. Example of a Pulse Generator Waveform



Creating Pulse Generators

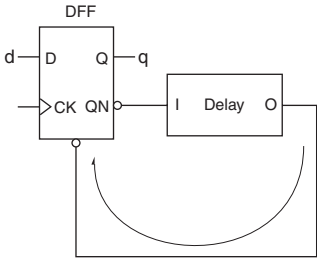
Pulse generators can be created in two ways. The first way to create a pulse generator is to increase the width of a glitch using a 2-input AND, NAND, OR, or NOR gate, where the source for the two gate inputs are the same, but the design delays the source for one of the gate inputs, as shown in Figure 19–26.

Figure 19–26. A Pulse Generator Circuit Using a 2-Input AND



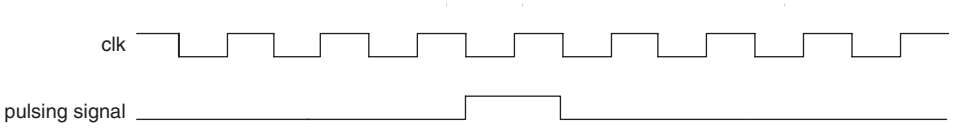
The second way to create a pulse generator is by using a register where the register output drives its own asynchronous reset signal through a delay chain, as shown in Figure 19–27.

Figure 19–27. Pulse Generator Circuit Using a Register Output to Drive a Reset Signal Through a Delay Chain



These pulse generators are asynchronous in nature and are detected by the Design Assistant as unacceptable circuit structures. If you need to generate a pulsed signal, you should do it in a purely synchronous manner. That is, where the duration of the pulse is equal to one or more clock periods, as shown in Figure 19–28.

Figure 19–28. An Example of a Synchronous Pulse Generator



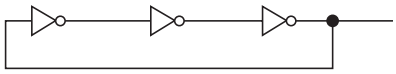
A synchronous pulse generator can be created with a simple section of Verilog HDL or VHDL code. The following is a Verilog HDL code fragment for a synchronous pulse generator circuit.

```
reg [2:0] count;
reg pulse;
always @ (posedge clk or negedge rst)
begin
   if (!rst)
      begin
         count[2:0] <= 3'b000;
         pulse <= 1'b0;
      end
   else
      begin
         count[2:0] <= count[2:0] + 1'b1;
         if (count == 3'b000)
            begin
               pulse <= 1'b1;
            end
         else
            begin
               pulse <= 1'b0;
            end
         end
      end
   end
end
```

Combinational Oscillator Circuits

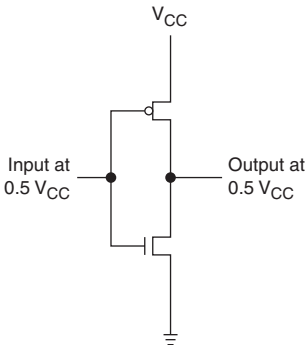
The circuit shown in Figure 19–29 on page 19–24 consists of a combinational logic gate whose inverted output feeds back to one of the inputs of the same gate. This feedback path causes the output to change state and, therefore, oscillate.

Figure 19–29. A Combinational Ring Oscillator Circuit



This circuit is sometimes built out of a series of cascaded inverters in a structure known as a ring oscillator. The frequency at which this circuit oscillates depends on the temperature, voltage, and process operating conditions of the device, and is completely asynchronous to any of the other clock domains in the device. Worse, the circuit may fail to oscillate at all, and the output of the inverter goes to a stable voltage at half of the supply voltage, as shown in Figure 19–30. This causes both the PMOS and NMOS transistors in the inverter chain to be switched on concurrently with a path from V_{CC} to GND, with no inverter function and consuming static current.

Figure 19–30. An Inverter Biased at 0.5 V_{CC}



Avoid implementing any kind of combinational feedback oscillator circuit.

Reset Circuitry

Reset signals are control signals that synchronously or asynchronously affect the state of registers in a design. The special consideration given to clock signals also needs to be given to reset signals. Only the term "reset" is used in this document, but the information described here also applies to "set," "preset," and "clear" signals. Reset signals should only be used to put a circuit into a known initial condition. Also, both the `set` and `reset` pins of the same register should never be used together. If the signals driving them are both activated at the same time, the logic state of the register may be indeterminate.

Gated Reset

A gated reset is generated when combinational logic feeds into the asynchronous reset pin of a register. The gated reset signal may have glitches on it, causing unintentional resetting of the destination register. Figure 19–31 shows a gated reset circuit where the signal driving into the register reset pin has glitches on it causing unintentional resetting.

Figure 19–31. A Gated Reset Circuit & its Associated Timing Diagram

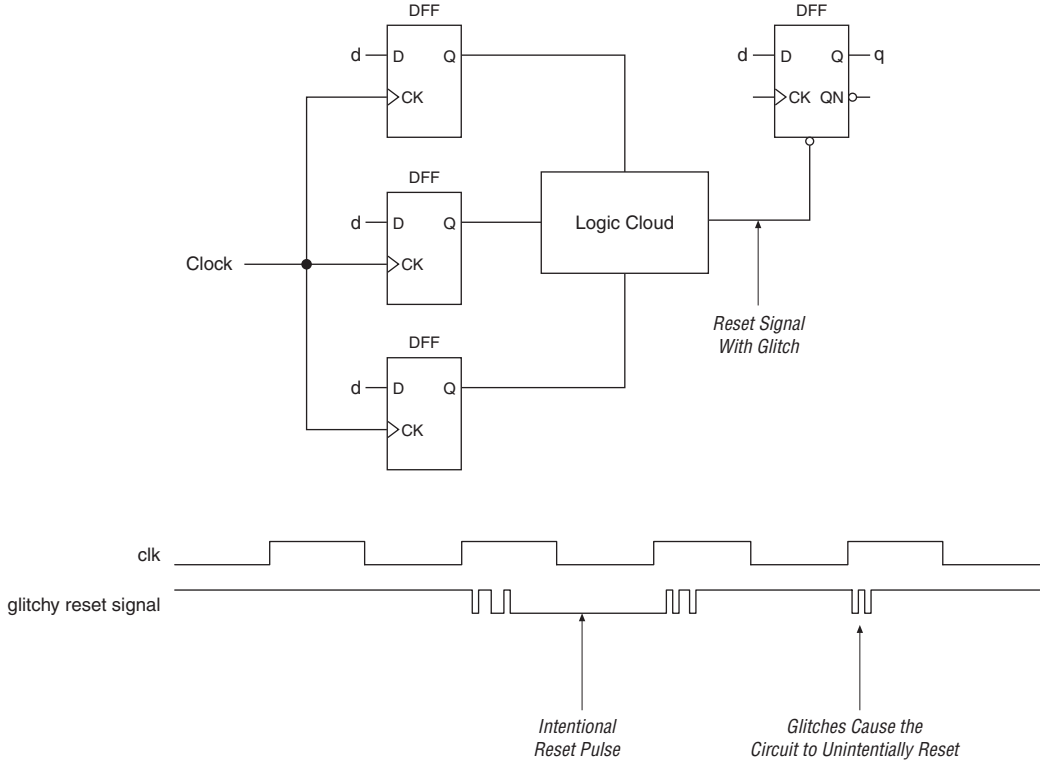
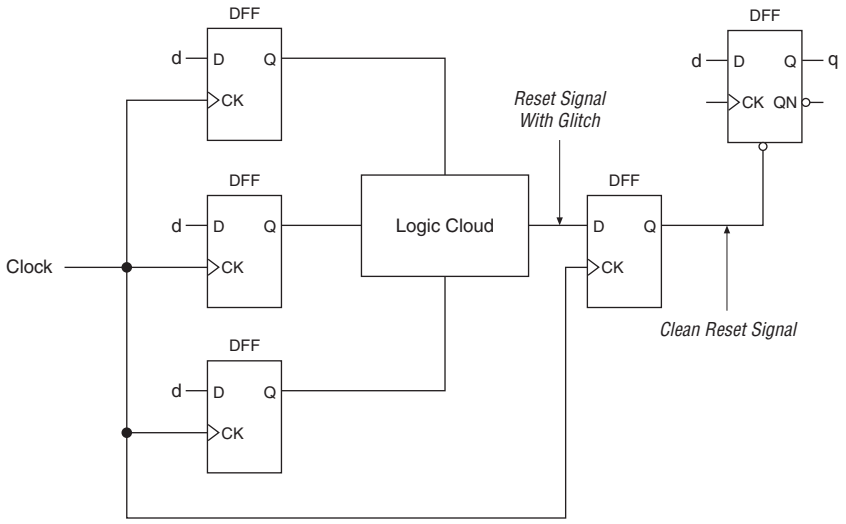


Figure 19–32 shows a better approach to implement a gated reset circuit, by placing a register on the output of the reset-gating logic, thereby synchronizing it to a clock. The register output then becomes a glitch-free reset signal that drives the rest of the design. However, the resulting reset signal is delayed by an extra clock cycle.

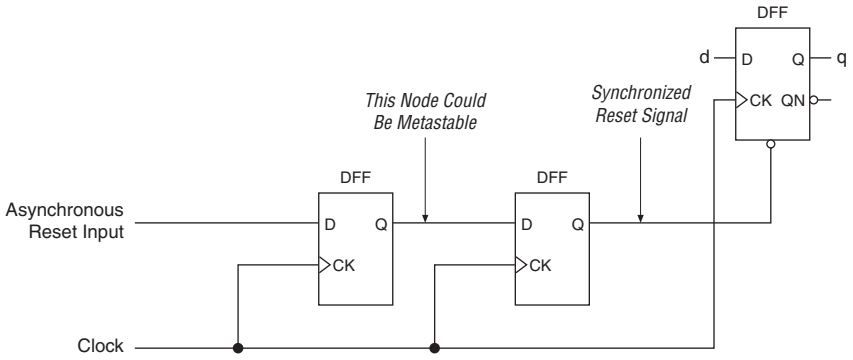
Figure 19–32. A Better Approach to the Gated Reset Circuit in Figure 19–31



Asynchronous Reset Synchronization

If the design needs to be put into a reset state in the absence of a clock signal, then the only way to achieve this is through the use of an asynchronous reset. However, it is possible to generate a synchronous reset signal from an asynchronous one by using a double-buffer circuit, as shown in Figure 19–33.

Figure 19–33. A Double-Buffer Circuit



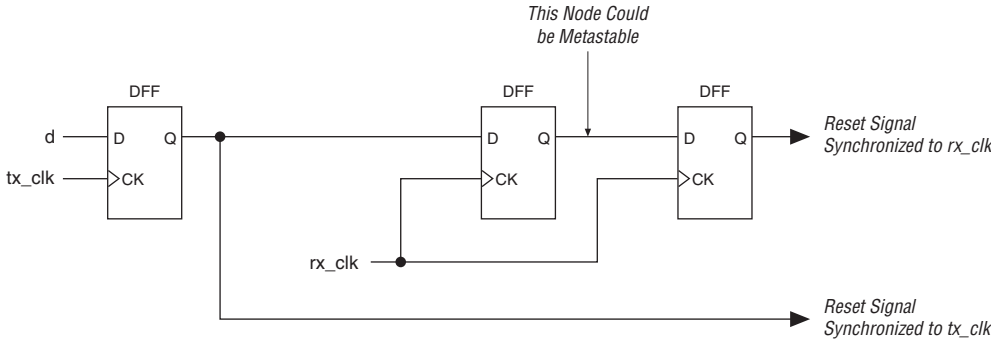
Synchronizing Reset Signals Across Clock Domains

In a design, an internally generated reset signal that is generated in one clock domain, and used in one or more other asynchronous clock domains, should be synchronized. A reset signal that is not synchronized can cause metastability problems.

The synchronization of the gated reset should follow these guidelines, as shown in Figure 19–34.

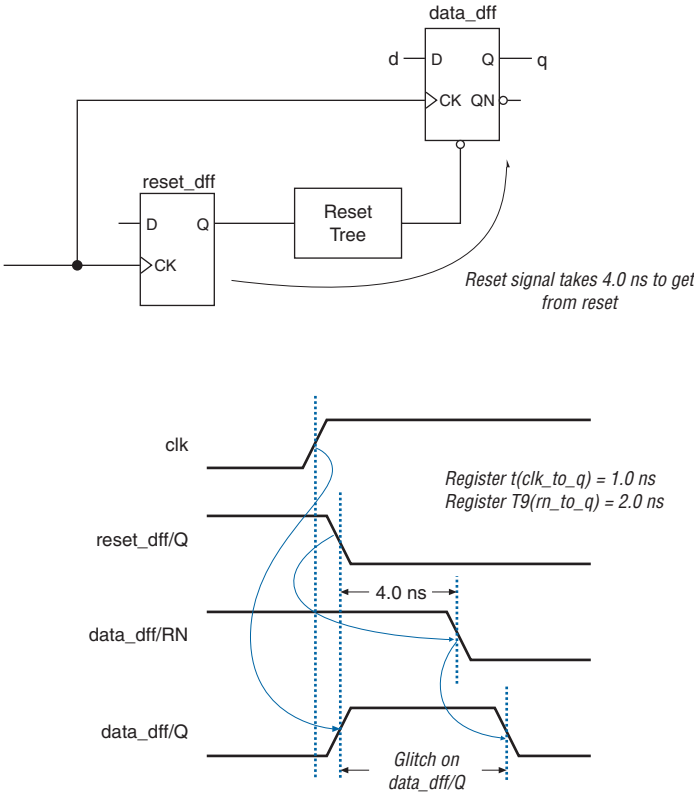
- The reset signal should be synchronized with two or more cascading registers in the receiving asynchronous clock domain.
- The cascading registers should be triggered on the same clock edge.
- There should be no logic between the output of the transmitting clock domain and the cascaded registers in the receiving asynchronous clock domain.

Figure 19–34. Circuit for a Synchronized Reset Signal Across Two Clock Domains



With either of the reset synchronization circuits described in Figures 19–33 and 19–34, when the reset is applied, the Q output of the registers in the design may send a wrong signal, momentarily causing some primary output pins to also send wrong signals. The circuit and its associated timing diagram, shown in Figure 19–35, demonstrate this phenomenon.

Figure 19–35. Common Problem with Reset Synchronization Circuits



A purely synchronous reset circuit does not exhibit this behavior. The following Verilog HDL RTL code shows how to do this.

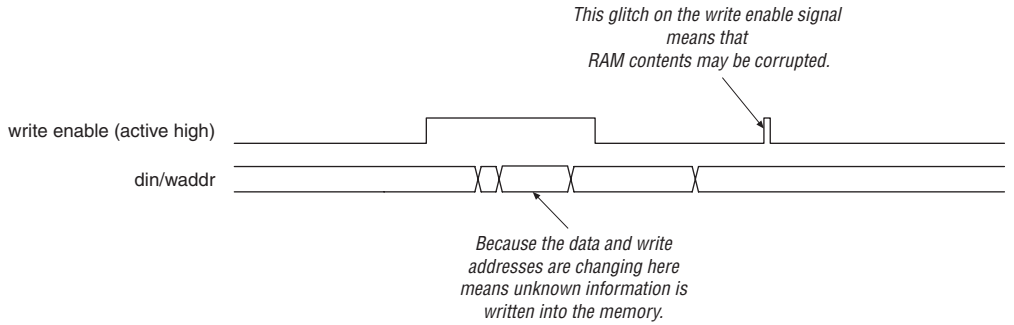
```
always @ (posedge clk)
      begin
      if (!rst)
         q <= 1'b0;
      else
         q <= d; end
```

Avoid using reset signals for anything other than circuit initialization, and be aware of the reset signal timing if reset-synchronizing circuitry is used.

Asynchronous RAM

Altera FPGA devices contain flexible embedded memory structures that can be configured into many different modes. One possible mode is asynchronous RAM. The definition of an asynchronous RAM circuit is one where the write-enable signal driving into the RAM causes data to be written into it, without a clock being required, as shown in Figure 19–36. This means that the RAM is sensitive to corruption if any glitches exist on the write-enable signal. Also, the data and write address ports of the RAM should be stable before the write pulse is asserted, and must remain stable until the write pulse is de-asserted. These limitations in using memory structures in this asynchronous mode imply that synchronous memories are always preferred. Synchronous memories also provide higher design performance.

Figure 19–36. Potential Problems of Using Asynchronous RAM Structures



Stratix, Stratix II, HardCopy Stratix, and HardCopy II device architectures do not support asynchronous RAM behavior. These devices always use synchronous RAM input registers. Altera recommends using RAM output registering; this is optional, however, not using output registering degrades performance.

APEX 20K FPGA and HardCopy APEX support both synchronous and asynchronous RAM using the embedded system block (ESB). Altera recommends using synchronous RAM structures. Immediately registering both input and output RAM interfaces improves performance and timing closure.

Conclusion

Most issues described in this document can be easily avoided while a design is still in its early stages. These issues not only apply to HardCopy devices, but to any digital logic integrated circuit design, whether it is a standard cell ASIC, gate array, or FPGA.

Sometimes, violating one or more of the above guidelines is unavoidable, but understanding the implications of doing so is very important. One must be prepared to justify to Altera the need to break those rules in this case, and to support it with as much documentation as possible.

Following the guidelines outlined in this document can ultimately lead to the design being more robust, quicker to implement, easier to debug, and fitted more easily into the target architecture, increasing the likelihood of success.

20. Power-Up Modes & Configuration Emulation in HardCopy Series Devices

H51012-2.2

Introduction

Configuring an FPGA is the process of loading the design data into the device. Altera's SRAM-based Stratix® II, Stratix, APEX™ 20KC, and APEX 20KE FPGAs require configuration each time the device is powered up. After the device is powered down, the configuration data within the Stratix II, Stratix, or APEX device is lost and must be loaded again on power up.

There are several ways to configure these FPGAs. The details on the various configuration schemes available for these FPGAs are explained in the *Configuration Handbook*.

HardCopy® series devices are mask-programmed and cannot be configured. However, in addition to the capability of being instantly on upon power up (like a traditional ASIC device), these devices can mimic the behavior of the FPGA during the configuration process if necessary.

This chapter addresses various power-up options for HardCopy series devices. This chapter also discusses how configuration is emulated in HardCopy series devices while retaining the benefits of seamless migration and provides examples of how to replace the FPGAs in the system with HardCopy series devices.

HardCopy Power-Up Options

HardCopy series devices feature three variations of instant on power up modes and a configuration emulation power up mode. They are as follows:

- Instant on
- Instant on after 50 ms
- Configuration emulation of an FPGA configuration sequence

You must choose the power-up option when submitting the design database to Altera for migrating to a HardCopy series device. Once the HardCopy series devices are manufactured, the power-up option cannot be changed.

HardCopy II and some HardCopy Stratix devices do not support configuration emulation. Refer to the "Configuration Emulation of FPGA Configuration Sequence" on page 20–9 section for more information.

HardCopy II and HardCopy Stratix devices retain the functionality of VCCSEL and PORSEL pins from the prototyping Stratix and Stratix II FPGAs. The signals can affect the HardCopy series power-up behavior using any power up option. Refer to the *Stratix Device Handbook* or the *Stratix II Device Handbook* for proper use of these additional signals.

Instant On Options

Instant on is the traditional power-up scheme of most ASIC and non-volatile devices. The instant on mode is the fastest power-up option of a HardCopy series device and is used when the HardCopy series device powers up independently while other components on the board still require initialization and configuration. Therefore, you must verify all signals that propagate to and from the HardCopy series device (for example, reference clocks and other input pins) are stable or do not affect the HardCopy series device operation.

There are two variations of instant on power-up modes available on all HardCopy devices.

- Instant on (no added delay)
- Instant on after 50 ms (additional delay)

Instant On (No Added Delay)

In the instant on power-up mode, once the power supplies ramp up above the HardCopy series device's power-on reset (POR) trip point, the device initiates an internal POR sequence. When this sequence is complete, the HardCopy series device transitions to an initialization phase, which releases the CONF_DONE signal to be pulled high. Pulling the CONF_DONE signal high indicates that the HardCopy series device is ready for normal operation. Figures 20–1 to 20–3 show the instant on timing waveform relationships of the configuration signals, $V_{CC,}$ and user I/O pins with respect to the HardCopy series device's normal operation mode.

During the power-up sequence, internal weak pull-up resistors can pull the user I/O pins high. Once POR and the initialization phase is complete, the I/O pins are released. Similar to the FPGA, if the nIO_pullup pin transitions high, the weak pull-up resistors are disabled. Refer to the table that provides recommended operating conditions in the handbook for the specific device.

The value of the internal weak pull-up resistors on the I/O pins is in the Operating Conditions table of the specific FPGA's device handbook.

Instant On After 50-ms Delay

The instant on after 50-ms delay power-up mode is similar to the instant on power-up mode. However, in this case, the device waits an additional 50 ms following the end of the internal POR sequence before releasing the CONF_DONE pin. This option is useful if other devices on the board (such as a microprocessor) must be initialized prior to the normal operation of the HardCopy series device.

An on-chip oscillator generates the 50-ms delay after the power-up sequence. During the POR sequence and delay period, all user I/O pins can be driven high by internal, weak pull-up resistors. Just like the instant on mode, these pull-up resistors are affected by the nIO_pullup pin.

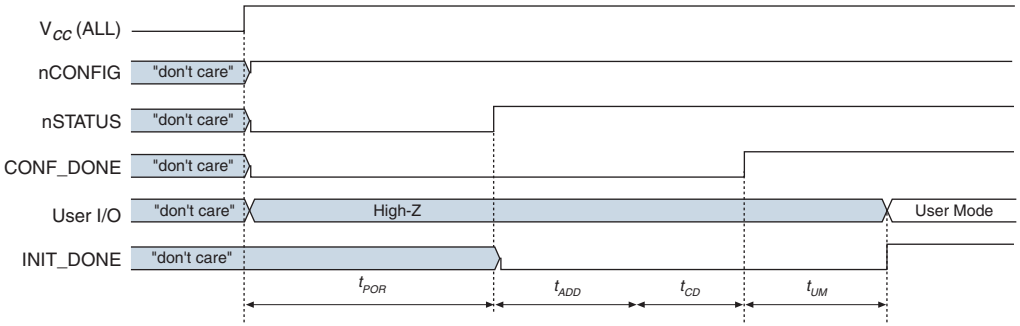
Similar to APEX 20K FPGAs, HardCopy APEX devices do not have an nIO_pullup function. Their internal, weak pull-up resistors are enabled during the power-up and initialization phase.

On the FPGA, an initialization phase occurs immediately after configuration where registers are reset, any PLLs used are initialized, and any I/O pins used are enabled as the device transitions into user mode. When the HardCopy series device uses instant on and instant on after 50-ms modes, a configuration sequence is not necessary, so the HardCopy series device transitions into the initialization phase after a power-up sequence immediately or after a 50-ms delay.

Figures 20–1 to 20–3 show instant on timing waveform relationships of the configuration signals, $V_{CC,}$ and user I/O pins with respect to the HardCopy series device's normal operation mode. Tables 20–1 to 20–3 define the timing parameters for each of the HardCopy series device waveforms, and also show the effect of the PORSEL pin on power up. The nCE pin must be driven low externally for these waveforms to apply.

Figure 20–1 shows an instant on power-up waveform, where the HardCopy device is powered up, and the nCONFIG, nSTATUS and CONF_DONE are not driven low externally.

Figure 20–1. Timing Waveform for Instant On Option Notes (1), (2), (3), (4), (5)



Notes to Figure 20–1:

(1) V_{CC} (ALL) represents either all of the power pins or the last power pin powered up to specified operating conditions. All HardCopy power pins must be powered within specifications as described under Hot Socketing sections.

(2) nCONFIG, nSTATUS, and CONF_DONE must not be driven low externally for this waveform to apply.

(3) User I/O pins may be tri-stated or driven before and during power up. See Hot Socketing sections for more details. The nIO_pullup pin can affect the state of the user I/O pins during the initialization phase.

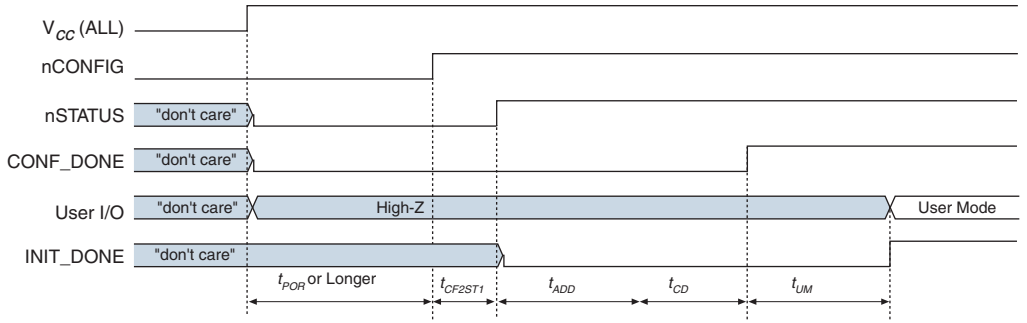
(4) INIT_DONE is an optional pin that can be enabled on the FPGA using the Quartus II software. HardCopy series devices carry over the INIT_DONE functionality from the prototyped FPGA design.

(5) The nCEO pin is asserted about the same time the CONF_DONE pin is released. However, the nCE pin must be driven low externally for this waveform to apply.

An alternative to the power-up waveform in Figure 20–1 is if the nCONFIG pin is externally held low longer than the PORSEL delay. This delays the initialization sequence by a small amount as indicated in Figure 20–2.

In addition, Figure 20–2 is an instant on power-up waveform where nCONFIG is momentarily held low and nSTATUS and CONF_DONE are not driven low externally.

Figure 20–2. Timing Waveform for Instant On Option Where nCONFIG is Held Low After Power Up *Notes (1), (2), (3), (4), (5), (6)*



Notes to Figure 20–2:

(1) This waveform applies if nCONFIG is held low longer than t_{POR} delay.

(2) V_{CC} (ALL) represents either all of the power pins or the last power pin powered up to specified operating conditions. All HardCopy power pins must be powered within specifications as described under Hot Socketing sections.

(3) nCONFIG, nSTATUS, and CONF_DONE must not be driven low externally for this waveform to apply.

(4) User I/O pins may be tri-stated or driven before and during power up. See Hot Socketing sections for more details. The nIO_pullup pin can affect the state of the user I/O pins during the initialization phase.

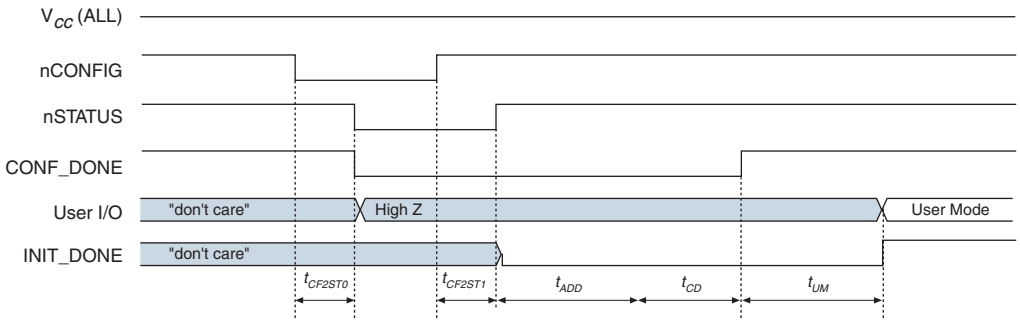
(5) INIT_DONE is an optional pin that can be enabled on the FPGA using the Quartus II software. HardCopy devices carry over the INIT_DONE functionality from the prototyped FPGA design.

(6) The nCEO pin is also asserted about the same time the CONF_DONE pin is released. However, the nCE pin must be driven low externally for this waveform to apply.

Pulsing the nCONFIG signal on an FPGA re-initializes the configuration sequence. The nCONFIG signal on a HardCopy series device also restarts the initialization sequence.

Figure 20–3 shows the instant on behavior of the configuration signals and user I/O pins if the nCONFIG pin is pulsed while the V_{CC} supplies are already powered up and stable.

Figure 20–3. Timing Waveform for Instant On Option When Pulsing NConfig Notes (1), (2), (3), (4), (5)



Notes to Figure 20–3:

(1) V_{CC} (ALL) represents either all of the power pins or the last power pin powered up to specified operating conditions. All HardCopy power pins must be powered within specifications as described under Hot Socketing sections.

(2) nSTATUS and CONF_DONE must not be driven low externally for this waveform to apply.

(3) The nIO_pullup pin can affect the state of the user I/O pins during the initialization phase.

(4) INIT_DONE is an optional pin that can be enabled on the FPGA using the Quartus II software. HardCopy devices carry over the INIT_DONE functionality from the prototyped FPGA design.

(5) The nCEO pin is also asserted about the same time the CONF_DONE pin is released. However, the nCE pin must be driven low externally for this waveform to apply.

In the FPGA, the INIT_DONE signal remains high for several clock cycles after the nCONFIG signal is asserted, after which time INIT_DONE goes low. In the HardCopy series device, the INIT_DONE signal starts low, as shown in Figure 20–3, regardless of the logic state of the nCONFIG signal. The INIT_DONE signal transitions high only after the CONF_DONE signal transitions high.

Tables 20–1 through 20–3 show the timing parameters for the instant on mode. These tables also show the time taken for completing the instant on power-up sequence in Figure 20–1 on page 20–4 for HardCopy series devices. This option is typical of an ASIC's functionality.

Table 20–1. Timing Parameters for Instant On Mode in HardCopy II Devices

Parameter	Description	Condition	Min	Typical	Max	Units
t_{POR}	PORSEL delay *(1)*	12		12		ms
		100		100		ms
t_{CF2ST0}	nCONFIG low to nSTATUS low *(1)*				800	ns
t_{CF2ST1}	nCONFIG high to nSTATUS high *(1)*				100	µs
t_{ADD}	Additional delay	Instant on	33		60	µs
		After 50 ms added delay	50		90	ms
t_{CD}	CONF_DONE delay		600		1100	ns
t_{UM}	User mode delay		25		55	µs

Note to Table 20–1:

(1) This parameter is similar to the Stratix II FPGA specifications. Refer to the *Configuration Handbook* for more information.

Table 20–2. Timing Parameters for Instant On Mode in HardCopy Stratix Devices

Parameter	Description	Condition	Min	Typical	Max	Units
t_{POR}	PORSEL delay	2	1	2		ms
		100	70	100		ms
t_{CF2ST0}	nCONFIG low to nSTATUS low *(1)*				800	ns
t_{CF2ST1}	nCONFIG high to nSTATUS high *(1)*				40	µs
t_{ADD}	Additional delay	Instant on	4		8	ms
		After 50 ms added delay	25	50	75	ms
t_{CD}	CONF_DONE delay		0.5		3	µs
t_{UM}	User mode delay		6.0		28	µs

Note to Table 20–2

(1) This parameter is similar to the Stratix FPGA specifications. Refer to the *Configuration Handbook* for more information.

Table 20–3. Timing Parameters for Instant On Mode in HardCopy APEX Devices

Parameter	Description	Condition	Min	Typical	Max	Units
t_{POR}	POR delay			5		µs
t_{CF2ST0}	nCONFIG low to nSTATUS low *(1)*				200	ns
t_{CF2ST1}	nCONFIG high to nSTATUS high *(1)*				1	µs
t_{ADD}	Additional delay	Instant on		0		µs
		After 50 ms added delay		50		ms
t_{CD}	CONF_DONE delay		0.5		3	µs
t_{UM}	User mode delay		2.5		8	µs

Note to Table 20–3:

(1) This parameter is similar to the APEX FPGA specifications. Refer to the *Configuration Handbook* for more information.

For correct operation of a HardCopy series device using the instant on option, pull the nSTATUS, nCONFIG, and CONF_DONE pins to V_{CC}. In the HardCopy series devices, these pins are designed with weak internal resistors pulled up to V_{CC}. Many FPGA configuration schemes require pull-up resistors on these I/O pins, so they may already be present on the board. In some HardCopy series device applications, you can remove these external pull-up resistors.

Altera recommends leaving external pull-up resistors on the board if one of the following conditions exists.

For more information, refer to the *Designing with 1.5-V Devices* chapter in the *Stratix Device Handbook*.

- There is more than one HardCopy series and/or FPGA on the board
- The HardCopy design uses configuration emulation
- The design uses MultiVolt™ I/O configurations

In the FPGA, you can enable INIT_DONE pin in the Quartus II software. If you used the INIT_DONE pin on the FPGA prototype, the HardCopy series device retains its function.

- In HardCopy series devices, the INIT_DONE settings option is masked-programmed into the device. You must submit these settings to Altera with the final design prior to migrating to a HardCopy series device. The use of the INIT_DONE option and other option pins (for example, DEV_CLRn and DEV_OE) are available in the Fitter Device Options sections of the Quartus II report file.
- For HardCopy II and HardCopy Stratix devices, the PORSEL pin setting delays the POR sequence similar to the prototyping FPGA. For more information on PORSEL settings for the FPGA, refer to the *Configuration Handbook*.

In some FPGA configuration schemes, inputs DCLK and DATA[7..0] float if the configuration device is removed from the board. In the HardCopy series devices, these I/O pins are designed with weak, internal pull-up resistors, so the pins can be left unconnected on the board.

Configuration Emulation of FPGA Configuration Sequence

In configuration emulation mode, the HardCopy series device emulates the behavior of an APEX or Stratix FPGA during its configuration phase. When this mode is used, the HardCopy device uses a configuration emulation circuit to receive configuration bit streams. When all the configuration data is received, the HardCopy series device transitions into an initialization phase and releases the CONF_DONE pin to be pulled high. Pulling the CONF_DONE pin high signals that the HardCopy series device is ready for normal operation. If the optional open-drain INIT_DONE output is used, the normal operation is delayed until this signal is released by the HardCopy series device.

HardCopy II and some HardCopy Stratix devices do not support configuration emulation mode.

During the emulation sequence, the user I/O pins can be pulled high by internal, weak pull-up resistors. Once the configuration emulation and initialization phase is completed, the I/O pins are released. Similar to the FPGA, if the nIO_pullup pin is driven high, the weak pull-up resistors are disabled. The value of the internal weak pull-up resistors on the I/O pins can be found in the Operating Conditions table of the specific FPGA's device handbook.

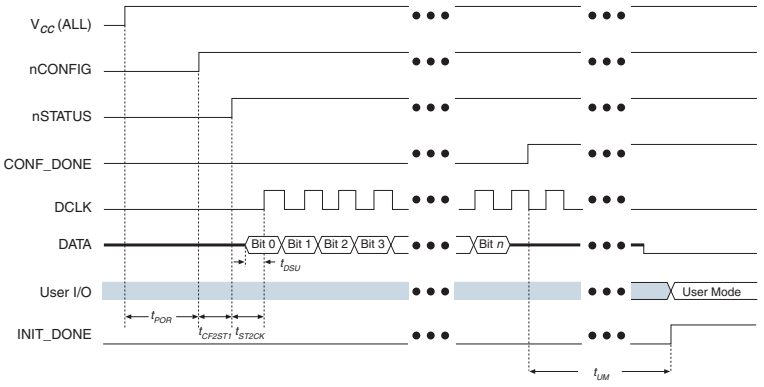
Similar to APEX 20K FPGAs, HardCopy APEX devices do not have an nIO_pullup function. Their internal weak pull-up resistors are enabled during the power up and initialization phase.

Similar to Stratix or APEX FPGAs, HardCopy Stratix or HardCopy APEX devices enter initialization phase immediately after a successful configuration sequence. At this time, registers are reset, any PLLs used are initialized, and any I/O pins used are enabled as the device transitions into user mode.

One application of the configuration emulation mode occurs when multiple programmable devices are cascaded in a configuration chain and only one device is replaced with a HardCopy series device. In this case, programming control signals and clock signals used to program the FPGA must also be used for the HardCopy series device. If this is not done, the HardCopy series device remains in the configuration emulation phase, the emulation sequence never ends, and the HardCopy `CONF_DONE` pin remains de-asserted. The proper configuration data stream and data clock is necessary so the HardCopy series device has the accurate emulation behavior.

Figure 20–4 shows a waveform of the configuration signals and the user I/O signals using configuration emulation mode.

Figure 20–4. Timing Waveform for Configuration Emulation Mode *Notes (1), (2), (3),(4),(5)*



Notes to Figures 20–4:

(1) V_{CC} (ALL) represents either all of the power pins or the last power pin powered up to specified operating conditions. All HardCopy power pins must be powered within specifications as described under Hot Socketing sections.

(2) `nCONFIG`, `nSTATUS`, and `CONF_DONE` must not be driven low externally for this waveform to apply.

(3) User I/O pins may be tri-stated or driven before and during power up. See the Hot Socketing sections for more details. The `nIO_pullup` pin can affect the state of the user I/O pins during the initialization phase.

(4) `INIT_DONE` is an optional pin that can be enabled on the FPGA using the Quartus Ii software. HardCopy devices will carry over the `INIT_DONE` functionality from the prototyped FPGA design.

(5) The `nCEO` pin is also asserted about the same time the `CONF_DONE` pin is released. However, the `nCE` pin must be driven low externally for this waveform to apply.

Configuration Emulation Timing Parameters

Tables 20–4 and 20–5 provide the timing parameters for the configuration emulation mode.

Table 20–4. Timing Parameters for Configuration Emulation Mode in HardCopy Stratix Devices *Note (1)*

Parameter	Description *(2)*	Condition	Min	Typ	Max	Units
t_{POR}	PORSEL delay	2	1	2		ms
		100	70	100		ms
t_{DSU}	Data setup time		7			ns
t_{CF2ST1}	nCONFIG high to nSTATUS				40	µs
t_{ST2CK}	nSTATUS to DCLK		1			µs
t_{UM}	User mode delay		6.0		28	µs

Notes to Table 20–4:
(1) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode.
(2) These parameters are similar to the Stratix FPGA specifications. Refer to the *Configuration Handbook* for more information.

Table 20–5. Timing Parameters for Configuration Emulation Mode in HardCopy APEX Devices

Parameter	Description *(1)*	Min	Typical	Max	Units
t_{POR}	POR delay		5		µs
t_{DSU}	Data setup time	10			ns
t_{CF2ST1}	nCONFIG high to nSTATUS			1	µs
t_{ST2CK}	nSTATUS to DCLK	1		3	µs
t_{UM}	User mode delay	2		8	µs

Notes to Table 20–5:
(1) These parameters are similar to the APEX FPGA specifications. Refer to the *Configuration Handbook* for more information.

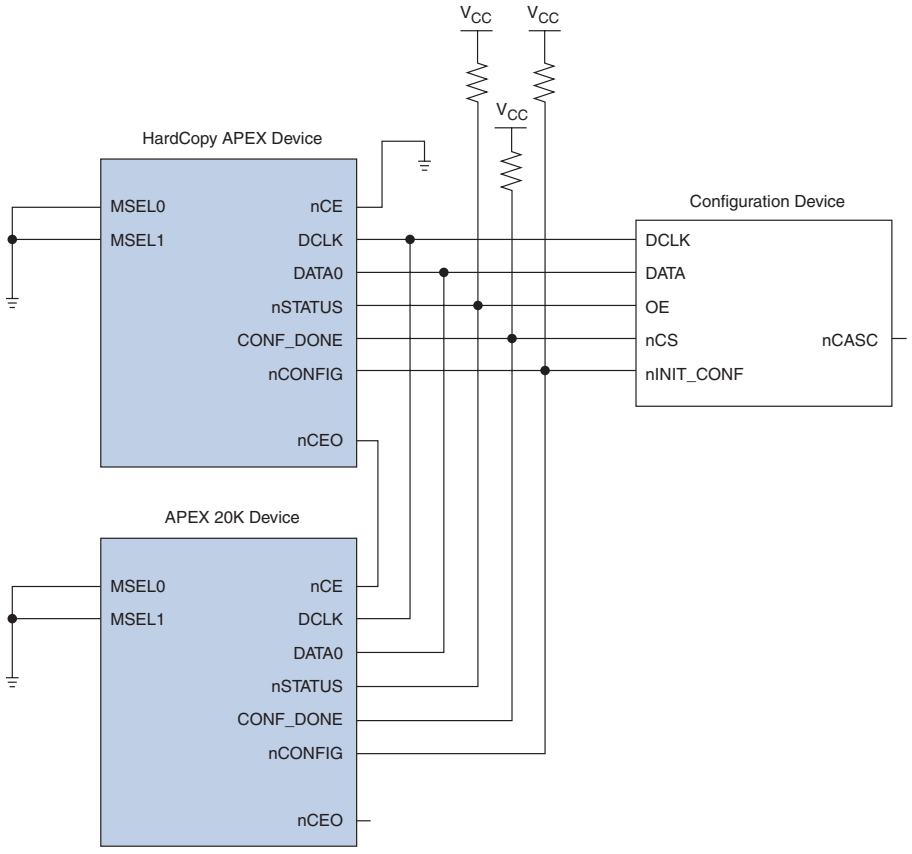
Benefits of Configuration Emulation

Configuration emulation in HardCopy series devices provides several advantages, including the following:

- Removes any necessity for changes to software, especially if the FPGA is configured using a microprocessor. Not having to change the software benefits the designer because microprocessor software changes demand significant system verification and qualification efforts, which also impact development time.
- Allows HardCopy series devices to co-exist with other FPGAs in a cascaded chain. None of the components need to be modified or added, and no design changes to the board are required. Additionally, no configuration software changes need to be made.
- Supports all configuration options available for the FPGA.

In this example, a single configuration device originally configured two APEX FPGAs. In Figure 20–5, a HardCopy APEX device replaces an APEX FPGA.

Figure 20–5. Emulation of Configuration Sequence



A HardCopy series device in configuration emulation mode requires the same configuration control signals as the FPGA that was replaced. In configuration emulation mode, the HardCopy series device responds in exactly the same way as the FPGA. The CONF_DONE signal of the HardCopy series device is asserted at exactly the same time as the FPGA.

Power Up Options Summary When Designing With HardCopy Series Devices

When designing a board for the prototyping FPGA with the intent of eventually replacing it with a HardCopy device, there are three power-up options that you should consider.

- Instant on
- Instant on after 50 ms
- Configuration emulation of an FPGA configuration sequence

You must choose the power-up option when submitting the design database to Altera for migrating to a HardCopy series device. Once the HardCopy series devices are manufactured, the power-up option cannot be changed.

HardCopy II and some HardCopy Stratix devices do not support configuration emulation mode.

HardCopy II and HardCopy Stratix devices retain the functionality of the VCCSEL and PORSEL pins from the prototyping Stratix II or Stratix FPGAs. For HardCopy II and HardCopy Stratix devices, the PORSEL pin setting delays the POR sequence similar to the prototyping FPGA.

For more information on PORSEL settings for the FPGA, refer to the *Configuration Handbook*.

The nCE and nCEO pins are functional in HardCopy series devices. The nCE pin must be held low for proper operation of the nCEO pin. If the nCE pin is driven low, the nCEO pin will be asserted after the initialization is completed and the CONF_DONE pin is released.

On the HardCopy II device, the nCE pin delays the initialization if it is not driven low. Like in the Stratix II device, nCEO and TDO of the HardCopy II device are powered by V_{CCIO}.

If you used the INIT_DONE pin on the FPGA prototype, the HardCopy series device retains its function. In HardCopy series devices, the INIT_DONE settings option is masked-programmed into the device. These settings must be submitted to Altera with the final design prior to migrating to a HardCopy series device. The use of the INIT_DONE option and other option pins (for example, DEV_CLRn and DEV_OE) are available in the Fitter Device Options sections of the Quartus II report file.

HardCopy II devices do not support the user-supplied start-up clock option available for Stratix II devices. The HardCopy II device uses its own internal clock for power-up circuitry. The startup clock selection is an option for configuring the FPGA, which you can set in the Quartus II software under Device and Pin Options.

HardCopy devices support device-wide reset (DEV_CLRn) and device-wide output enable (DEV_OE). The HardCopy settings follow the prototyping FPGA setting, which you set in the Quartus II software under Device and Pin Options.

For correct operation of a HardCopy series device using the instant on option, pull the nSTATUS, nCONFIG, and CONF_DONE pins to V_{CC}. In the HardCopy series devices, these pins are designed with weak, internal resistors pulled up to V_{CC}. Many FPGA configuration schemes require pull-up resistors on these I/O pins, so they may already be present on the board. In some HardCopy series device applications, you can remove these external pullup resistors.

Altera recommends leaving external pull-up resistors on the board if one of the following conditions exists:

- There is more than one HardCopy series and/or FPGA on the board
- The HardCopy design uses configuration emulation
- The design uses MultiVolt™ I/O configurations

For more information, refer to the *Designing with 1.5-V Devices* chapter in the *Stratix Device Handbook*.

In some FPGA configuration schemes, inputs DCLK and DATA[7..0] float if the configuration device is removed from the board. In the HardCopy series devices, these I/O pins are designed with weak internal pull-up resistors, so the pins can be left unconnected on the board.

When designing a board with a Stratix II prototype device and its companion HardCopy II device, most configuration pins required by the Stratix II device are not required by the HardCopy II device. To maximize I/O pin counts with HardCopy II device utilization, Altera recommends minimizing power-up and configuration pins that do not carry over from a Stratix II device into a HardCopy II device. More information can be found on the *Migrating Stratix II Device Resources* to *HardCopy II Devices* chapter.

HardCopy devices support the MSEL settings used on the FPGA. You are not required to change these settings on the board when replacing the prototyping FPGA with the HardCopy series device.

HardCopy II devices do not use MSEL pins and these pin locations are not connected in the package. It is acceptable to drive these pins to V_{CC} or GND as required by the prototyping Stratix II device.

Pulsing the nCONFIG signal on an FPGA re-initializes the configuration sequence. The nCONFIG signal on a HardCopy series device also restarts the initialization sequence.

The HardCopy device JTAG pin locations match their corresponding FPGA prototypes. Like the FPGAs, the JTAG pins have internal weak pull ups or pull downs on the four input pins TMS, TCK, TDI and TRST. There is no requirement to change the JTAG connections on the board when replacing the prototyping FPGA with the HardCopy series device. More information on JTAG pins is the corresponding *Boundary-Scan Support* chapter for each device.

Power-Up Option Selection & Examples

The HardCopy series device power-up option is mask-programmed. Therefore, it is important that the board design is verified to ensure that the HardCopy series device power-up option chosen will work properly. This section provides recommendations on selecting a power-up option and provides some examples.

Table 20–6 shows a comparison of applicable FPGA and HardCopy power up options.

Table 20–6. FPGA Configuration Modes & HardCopy Series Power-Up Schemes

Power Up Scheme	Device Family					
	Stratix II	Stratix	APEX 20K APEX 20KE APEX 20KC	HardCopy II *(1)*	HardCopy Stratix *(2)*	HardCopy APEX
Instant on				✓	✓	✓
Instant on after 50 ms				✓	✓	✓
Passive serial (PS)	✓	✓	✓		✓	✓
Active serial (AS)	✓					
Fast passive parallel (FPP)	✓	✓			✓	
Passive parallel synchronous (PPS)			✓			✓
Passive parallel asynchronous (PPA)	✓	✓	✓		✓	✓
Joint Test Action Group (JTAG)	✓	✓	✓		✓	✓
Remote local update FPP *(3)*	✓	✓				

Table 20–6. FPGA Configuration Modes & HardCopy Series Power-Up Schemes

Power Up Scheme	Device Family					
	Stratix II	Stratix	APEX 20K APEX 20KE APEX 20KC	HardCopy II *(1)*	HardCopy Stratix *(2)*	HardCopy APEX
Remote local update PPA *(3)*	✓	✓				
Remote local update PS *(3)*		✓				

Notes to Table 20–6:
(1) HardCopy II devices do not support emulation mode.
(2) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode.
(3) The remote/local update feature of Stratix devices is not supported in HardCopy Stratix devices.

Power-up option recommendations depend on the following board configurations:

- Single HardCopy series device replacing a single FPGA on the board
- One or more HardCopy series devices replacing one or more FPGA of a multiple-device configuration chain
- All HardCopy series devices replacing all FPGAs of a multiple-device configuration chain

In a multiple-device configuration chain, more than one FPGA on a board obtains configuration data from the same source.

Replacing One FPGA With One HardCopy Series Device

Altera recommends using the instant on or instant on after 50 ms mode when replacing an FPGA with a HardCopy series device regardless of the board configuration scheme. Table 20–7 gives a summary of HardCopy series device power-up options when a single HardCopy series device replaces a single FPGA on the board.

Table 20–7 does not include HardCopy II options because HardCopy II devices only support instant on and instant on after 50 ms modes.

Table 20–7. Summary of Power-Up Options for One HardCopy Series Device Replacing One FPGA

Configuration Scheme	HardCopy APEX Options	HardCopy Stratix Options	Comments
PS with configuration device(s) or download cable *(1)*	• Instant on • Instant on after 50 ms	• Instant on • Instant on after 50 ms	The configuration device(s) must be removed from the board.
FPP with enhanced configuration devices	• Not available	• Instant on • Instant on after 50 ms	The configuration device(s) must be removed from the board.
PS, PPA, PPS, FPP, with a microprocessor *(2)*	• Emulation	• Emulation *(3)*	If the microprocessor code can be changed, the design should use the instant on or instant on after 50 ms mode. However, the microprocessor still needs to drive a logic '1' value on the HardCopy `nCONFIG` pin
JTAG configuration	• Instant on after 50 ms • Emulation	• Instant on after 50 ms • Emulation *(3)*	Configuration emulation mode can be used but delays the initialization of the board or device.

Notes to Table 20–7:

(1) Download cable used may be either MasterBlaster™, USB Blaster, ByteBlaster™ II, or ByteBlasterMV™ hardware.

(2) For parallel programming modes, `DATA[7..1]` pins have weak pull up resistors on the HardCopy series device, which can be optionally enabled or disabled through metallization. `DCLK` and `DATA[0]` pins have internal weak pull-up resistors.

(3) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode.

Replacing One or More FPGAs With One or More HardCopy Series Devices in a Multiple-Device Configuration Chain

Altera recommends using the instant on or instant on after 50 ms mode when replacing an FPGA with a HardCopy series device, regardless of configuration scheme. Table 20–8 gives a summary of HardCopy series device power-up options when a single HardCopy series device replaces a single FPGA of a multiple-device configuration chain.

When using the instant on or instant on after 50 ms mode, the HardCopy series device could be in user-mode and ready before other configured devices on the board. It is important to verify that any signals that communicate to and from the HardCopy series device are stable or will not affect the HardCopy series device or other device operation while the devices are still in the power up or configuration stage. For example, if the HardCopy series design used a PLL reference clock that is not available until after other devices are fully powered up, the HardCopy series device PLL will not operate properly unless the PLLs are reset.

Table 20–8 does not include HardCopy II options because HardCopy II devices only support instant on and instant on after 50 ms modes.

Table 20–8. Power-Up Options for One or More HardCopy Series Devices Replacing FPGAs in a Multiple-Device Configuration Chain (Part 1 of 2)

Configuration Scheme	HardCopy APEX Options	HardCopy Stratix Options	Comments
PS with configuration device(s) or download cable *(1)* FPP with enhanced configuration device *(4)*	• Emulation • Instant on *(3)* • Instant on after 50 ms *(3)*	• Emulation *(2)* • Instant on *(3)* • Instant on after 50 ms *(3)*	Instant on or instant on after 50 ms modes can be used if the nCE pin of the following APEX or Stratix device can be tied to logic 0 on the board and the configuration data is modified to remove the HardCopy series device configuration data. The configuration sequence then skips the HardCopy series device.
PS, PPA, PPS, FPP, with a microprocessor *(4)*	• Emulation	• Emulation *(2)*	If the microprocessor code can be changed, the design should use the instant on or instant on after 50 ms mode. However, the microprocessor still needs to drive a logic '1' value on the HardCopy series device nCONFIG pin.

Table 20–8. Power-Up Options for One or More HardCopy Series Devices Replacing FPGAs in a Multiple-Device Configuration Chain (Part 2 of 2)

Configuration Scheme	HardCopy APEX Options	HardCopy Stratix Options	Comments
JTAG configuration	• Emulation	• Emulation *(2)*	If the HardCopy series device is put in BYPASS mode and the JTAG programming data is modified to remove the HardCopy configuration information, instant on or instant on after 50 ms modes can be used.

Notes to Table 20–8:

(1) Download cable used may be either MasterBlaster™, USB Blaster, ByteBlaster™ II, or ByteBlasterMV™ hardware.
(2) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode
(3) If the HardCopy series device is the last device in the configuration chain, Altera recommends using instant on modes.
(4) For parallel programming modes, `DATA[7..1]` pins have weak pull up resistors on the HardCopy series device, which can be optionally enabled or disabled through metallization. `DCLK` and `DATA[0]` pins also have weak pull-up resistors.

Replacing all FPGAs with HardCopy Series Devices in a Multiple-Device Configuration Chain

When all Stratix II, Stratix, and APEX FPGAs are replaced by HardCopy II, HardCopy Stratix, and HardCopy APEX devices, respectively, Altera recommends using the instant on or instant on after 50 ms mode, regardless of configuration scheme.

Once the HardCopy series devices replace the FPGAs, any configuration devices used to configure the FPGAs should be removed from the board. Microprocessor code, if applicable, should be changed to account for the HardCopy series device power-up scheme. You can use the JTAG chain to perform other JTAG operations except configuration.

FPGA to HardCopy Configuration Migration Examples

The following are examples of how HardCopy series devices replace FPGAs that use different FPGA configuration schemes.

HardCopy Series Device Replacing a Stand-Alone FPGA

In this example, the single HardCopy series device uses the instant on power-up option, as shown in Figure 20–7. The configuration device, now redundant, is removed, and no further board changes are necessary. The pull-up resistors on the `nCONFIG`, `nSTATUS`, and `CONF_DONE` pins can be removed, but should be left on the board if configuration emulation or multiple-voltage I/O standards are used. You could also use the instant on after 50 ms power-up mode in this example.

Figures 20–6 and 20–7 show how a HardCopy series device replaces an FPGA previously configured with an Altera configuration device.

Figure 20–6. Configuration of a Stand-Alone FPGA *Note (1)*

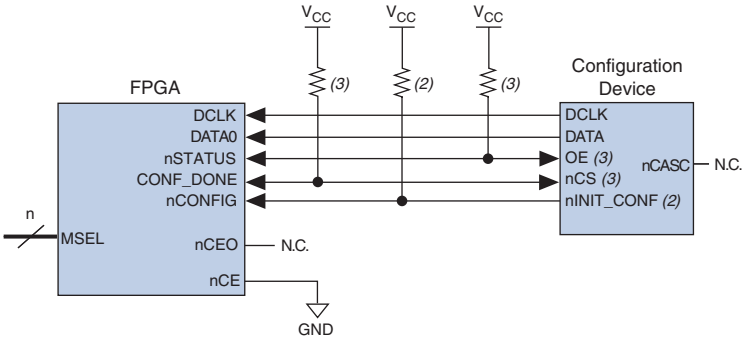
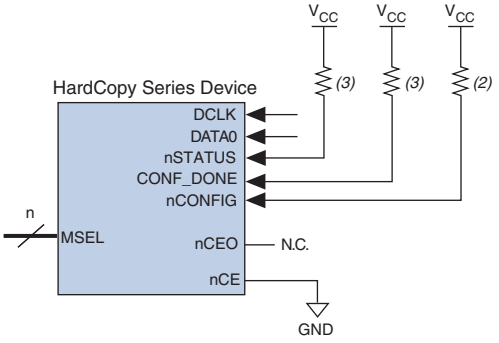


Figure 20–7. HardCopy Series Device Replacing Stand-Alone FPGA *Note (1)*



Notes to Figures 20–6 and 20–7:

(1) For details on configuration interface connections, refer to the *Configuration Handbook*. The handbook includes information on MSEL pins set to PS mode.

(2) The `nINIT_CONF` pin (available on enhanced configuration and EPC2 devices) has an internal pull-up resistor that is always active. Therefore, the `nINIT_CONF`/`nCONFIG` line does not require an external pull-up resistor. The `nINIT_CONF` pin does not need to be connected if its functionality is not used. If `nINIT_CONF` is not used or not available, use a resistor to pull the `nCONFIG` pin to V_{CC}.

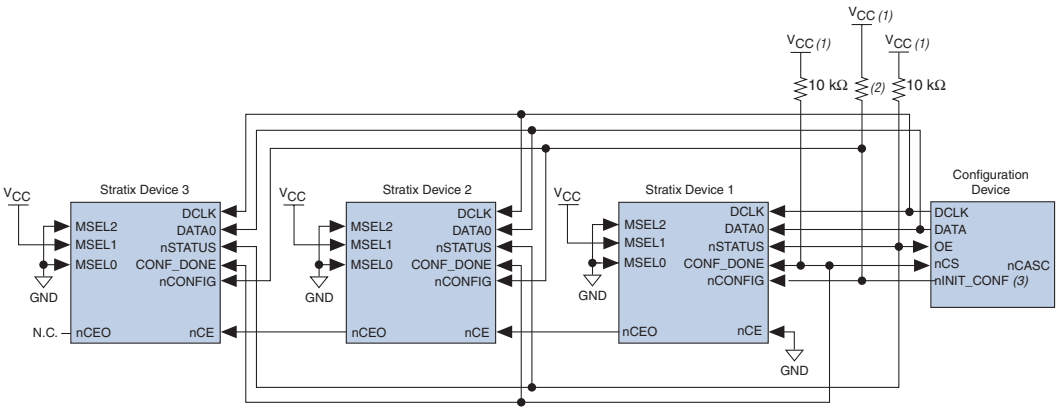
(3) Enhanced configuration and EPC2 devices have internal programmable pull-up resistors on `OE` and `nCS` pins. Refer to the *Configuration Handbook* for more details of this application in FPGAs. HardCopy series devices have internal weak pull-up resistors on `nSTATUS`, `nCONFIG`, and `CONF_DONE` pins.

HardCopy Series Device Replacing an FPGA in a Cascaded Configuration Chain

Figure 20–8 shows a design where the configuration data for the Stratix devices is stored in a single configuration device, and the FPGAs are connected in a multiple-device configuration chain. The second device in the chain is replaced with a HardCopy Stratix device, as shown in Figure 20–9.

For more information on Stratix FPGA configuration schemes, refer to the *Configuration Handbook*.

Figure 20–8. Configuration of Multiple FPGAs in a Cascade Chain



Notes to Figure 20–8:

(1) The pull-up resistors are connected to the same supply voltage as the configuration device.

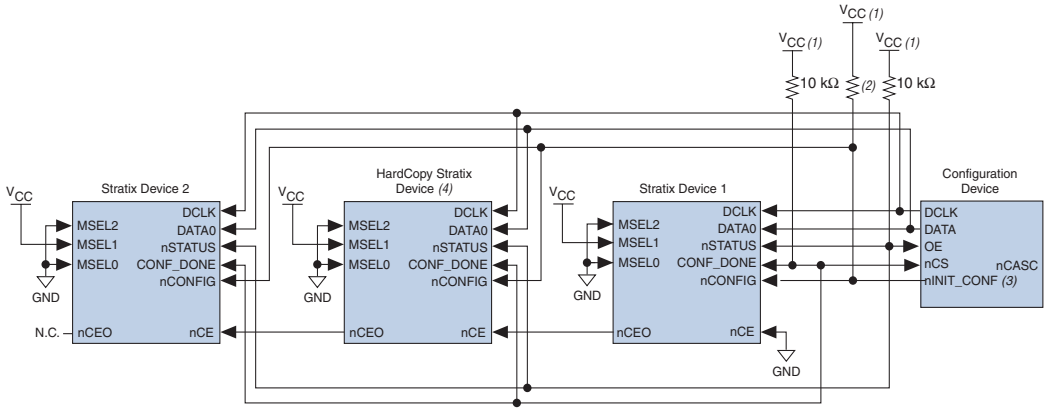
(2) The enhanced configuration devices and EPC2 devices have internal programmable pull-up resistors on the OE and nCS pins. Refer to the *Configuration Handbook* for more details.

(3) The nINIT_CONF pin is available on EPC16, EPC8, EPC4, and EPC2 devices. Refer to the *Configuration Handbook* for more details.

Configuration with the HardCopy Series Device in the Cascade Chain

Figure 20–9 shows the same cascade chain as Figure 20–8, but the second FPGA in the chain has been replaced with a HardCopy Stratix device.

Figure 20–9. Replacing an FPGA with a HardCopy Equivalent in the Cascade Chain



Notes to Figure 20–9:

(1) The pull-up resistors are connected to the same supply voltage as the configuration device.

(2) The enhanced configuration devices and EPC2 devices have internal programmable pull-up resistors on the OE and nCS pins. Refer to the *Configuration Handbook* for more details.

(3) The nINIT_CONF pin is available on EPC16, EPC8, EPC4, and EPC2 devices. Refer to the *Configuration Handbook* for more information.

(4) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode and cannot be used in this method.

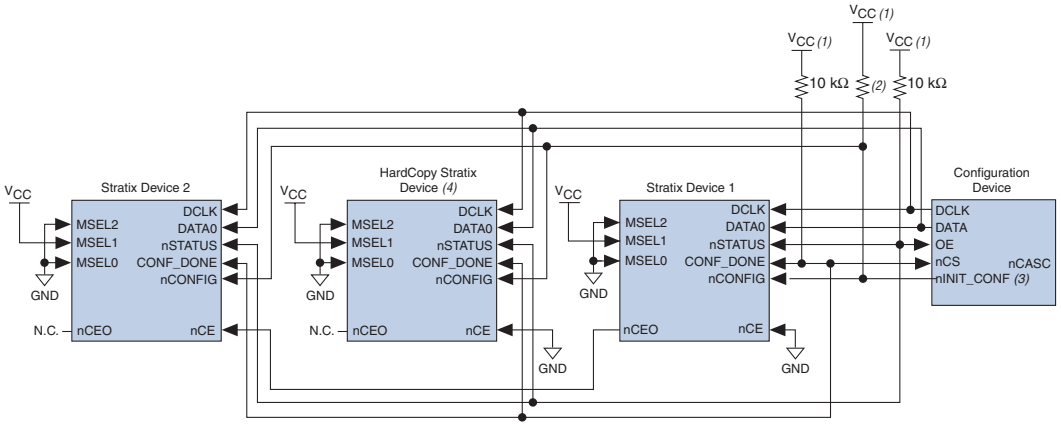
In this example, the HardCopy Stratix device can only be configured using the configuration emulation mode. The configuration device cannot be removed, as it is still required by other Stratix devices in the chain. While the HardCopy Stratix device does not need the data stored in the configuration device, the data in the configuration device is not modified to reflect this. The emulation mode ensures that the HardCopy series device nCEO pin is asserted correctly after the emulation of the configuration sequence. The nCEO pin enables the next device in the chain to receive the correct configuration data from the configuration device. Additionally, with the configuration emulation mode, you do not need to make any changes to the board.

Configuration With the HardCopy Series Device Removed From the Cascade Chain

An alternative method to configure FPGAs on a board with both HardCopy series devices and FPGAs is to remove the HardCopy series device from the cascade chain. Figure 20–10 shows how the devices are connected with the HardCopy series device removed from the chain.

The data in the configuration device should be modified to exclude the HardCopy series device configuration data. The HardCopy series device can use any of the three power-up options.

Figure 20–10. Configuration With the HardCopy Series Device Removed From the Cascade Chain



Notes to Figure 20–10:

(1) The pull-up resistors are connected to the same supply voltage as the configuration device.

(2) The enhanced configuration devices and EPC2 devices have internal programmable pull-up resistors on the OE and nCS pins. Refer to the *Configuration Handbook* for more details.

(3) The nINIT_CONF pin is available on EPC16, EPC8, EPC4, and EPC2 devices. Refer to the *Configuration Handbook* for more information.

(4) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode and cannot be used in this method.

Eliminating the HardCopy series device from the configuration chain requires the following changes on the board:

- The nCE pin of the HardCopy series device must be tied to GND.
- The nCE pin of the FPGA that was driven by the HardCopy series nCEO pin must now be driven by the nCEO pin of the FPGA that precedes the HardCopy series device in the chain.

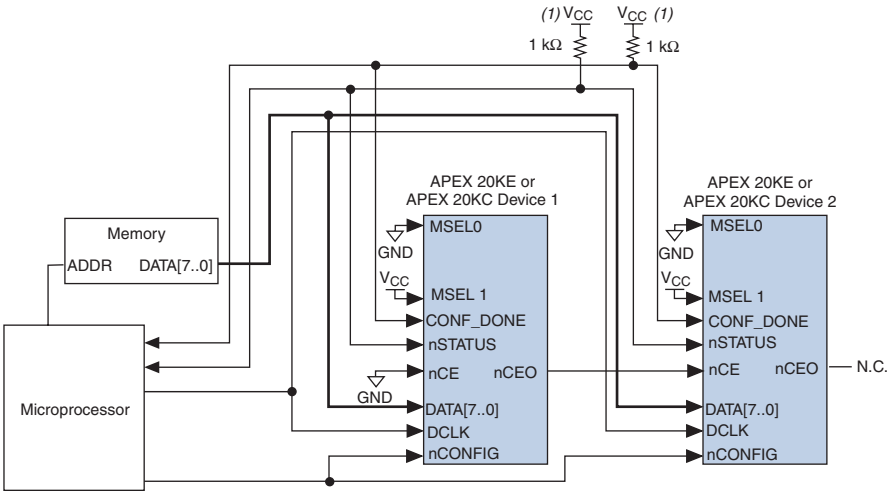
HardCopy Series Device Replacing an FPGA Configured Using a Microprocessor

The HardCopy series device can replace FPGAs that are configured using a microprocessor, as shown in Figures 20–12 and 20–13. While the instant on mode is the most efficient, designers can also use the instant on after 50 ms and configuration emulation mode.

Figure 20–11 shows an application where APEX FPGAs are configured using a microprocessor in the PPS configuration scheme.

For more information on the PPS configuration scheme, refer to the *Configuration Handbook*.

Figure 20–11. Configuring FPGAs Using a Microprocessor



Note to Figure 20–11:

(1) Connect the pull-up resistors to a supply that provides an acceptable input signal for all devices in the chain.

When the HardCopy series device replaces the last FPGA of the configuration sequence (as shown in Figure 20–12), use the instant on or instant on after 50 ms mode. However, you must modify the microprocessor code to eliminate the configuration data for the last FPGA of the configuration chain.

Figures 20–12 and 20–13 show the HardCopy APEX device replacing APEX FPGAs either first or last in the configuration chain.

Figure 20–12. Replacement of Last FPGA in the Chain With a HardCopy Series Device

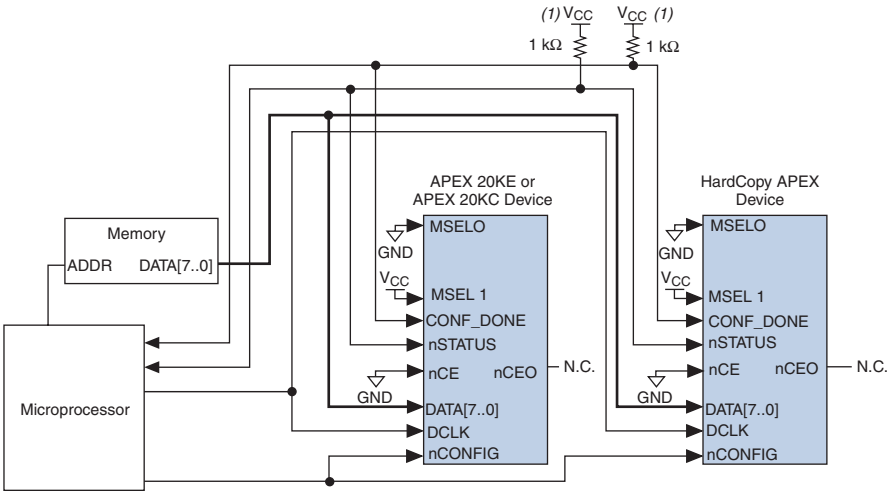
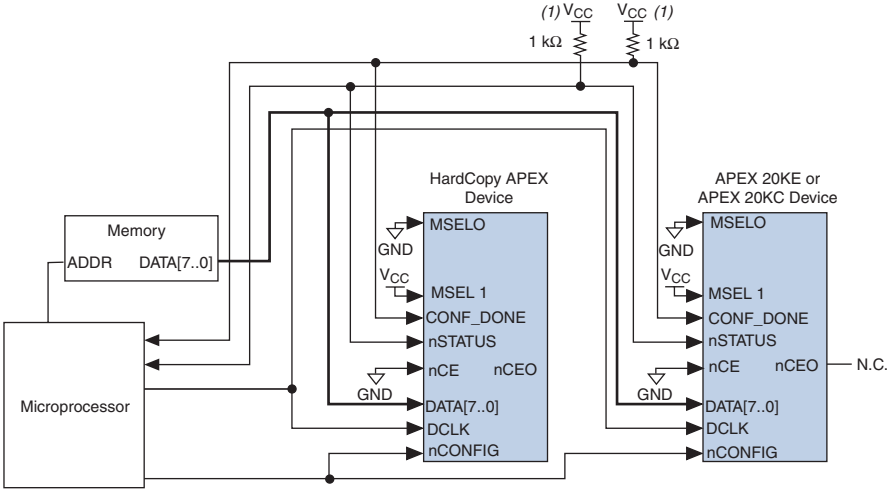


Figure 20–13. Replacement of First FPGA in the Chain With a HardCopy Series Device



Note to Figures 20–12 and 20–13:

(1) Connect the pull-up resistors to a supply that provides an acceptable input signal for all devices in the chain.

If the HardCopy series device is the first device in the chain as opposed to the second (as shown in Figure 20–13), you must take the following into consideration depending on the HardCopy power-up option used.

- Instant on mode—The microprocessor program code must be modified to remove the configuration code relevant to the HardCopy series device. The microprocessor must delay sending the first configuration data word to the FPGA until the nCEO pin on the HardCopy series device is asserted. The microprocessor then loads the first configuration data word into the FPGA.

- Instant on after 50 ms mode—The boot-up time of the microprocessor must be greater than 50 ms. The HardCopy series device asserts the nCEO pin after the 50-ms delay which, in turn, enables the following FPGA. The microprocessor can send the first configuration data word to the FPGA after the FPGA is enabled.

- Emulation mode—This option should be used if the microprocessor code pertaining to the configuration of the above devices cannot be modified.

HardCopy Stratix Device Replacing FPGA Configured in a JTAG Chain

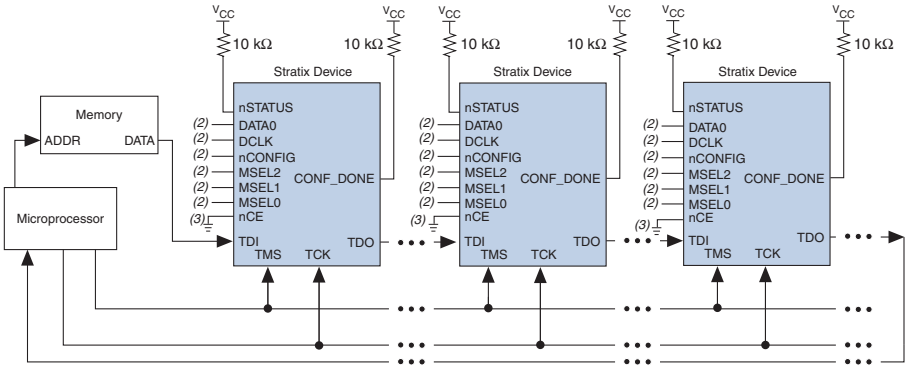
In this example, the circuit connectivity is maintained and there are no changes made to the board. The HardCopy series device can use either of the following power-up options when applicable.

- Instant on mode—Use the instant on power up mode if the microprocessor code can be modified so that it treats the HardCopy series device as a non-configurable device. The microprocessor can achieve this by issuing a BYPASS instruction to the HardCopy series device. With the HardCopy series device in BYPASS mode, the configuration data passes through it to the downstream FPGAs.

- Configuration emulation mode—Use the configuration emulation power up mode if the microprocessor code pertaining to the configuration of the above devices cannot be modified. HC1S80, HC1S60, and HC1S25 devices do not support this mode.

Figure 20–14 shows an example where there are multiple Stratix FPGAs. These devices are connected using the JTAG I/O pins for each device, and programmed using the JTAG port. An on-board microprocessor generates the configuration data.

Figure 20–14. Configuring FPGAs in a JTAG Chain Using a Microprocessor *Note (1)*



Notes to Figure 20–14:

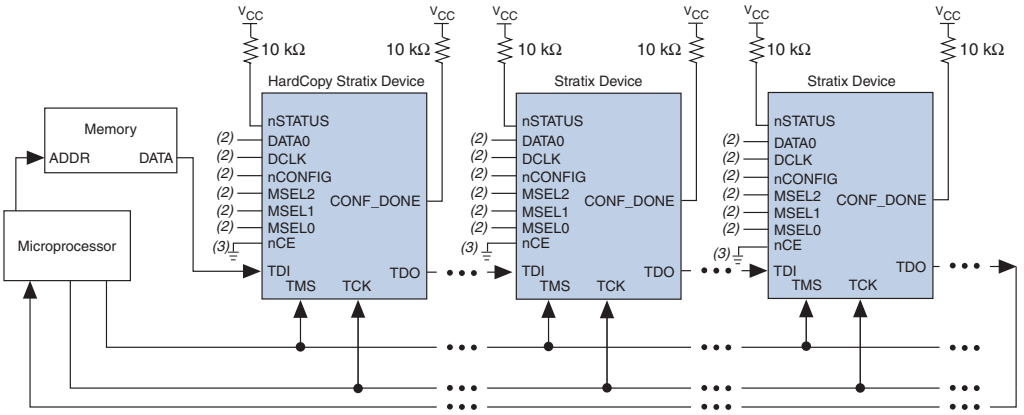
(1) Stratix II, Stratix, and APEX 20K devices can be placed within the same JTAG chain for device programming and configuration.

(2) Connect the nCONFIG, MSEL0, MSEL1, and MSEL2 pins to support a non-JTAG configuration scheme. If only JTAG configuration is used, connect nCONFIG to V_{CC}, and MSEL0, MSEL1, and MSEL2 to ground. Pull DATA0 and DCLK to either high or low.

(3) nCE must be connected to GND or driven low for successful JTAG configuration.

Figure 20–15 shows an example where the first Stratix device in the JTAG chain is replaced by a HardCopy Stratix device.

Figure 20–15. Replacement of the First FPGA in the JTAG Chain With a HardCopy Series Device *Note (1)*



Notes to Figure 20–15:

(1) Stratix II, Stratix, and APEX 20K devices can be placed within the same JTAG chain for device programming and configuration.

(2) Connect the nCONFIG, MSEL0, MSEL1, and MSEL2 pins to support a non-JTAG configuration scheme. If only JTAG configuration is used, connect nCONFIG to V_{CC}, and MSEL0, MSEL1, and MSEL2 to ground. Pull DATA0 and DCLK to either high or low.

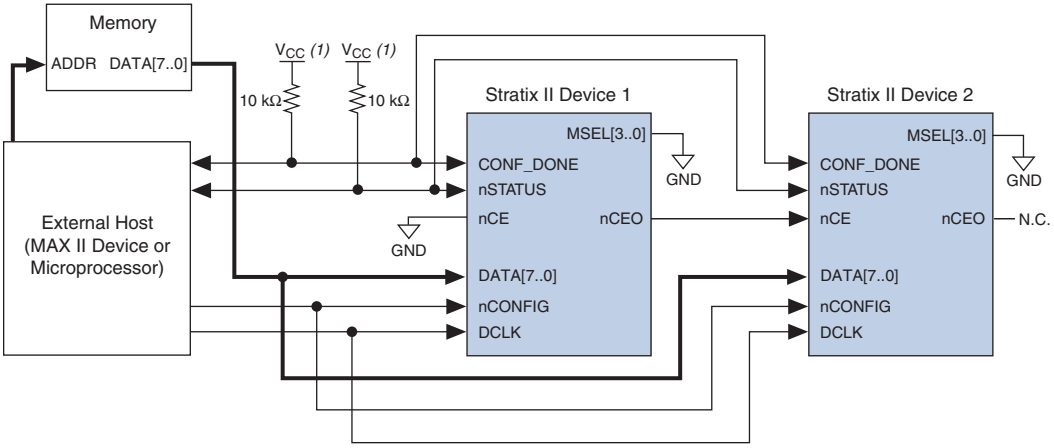
(3) nCE must be connected to GND or driven low for successful JTAG configuration.

HardCopy II Device Replacing Stratix II Device Configured With a Microprocessor

When replacing a Stratix II FPGA with a HardCopy II device, the HardCopy II device can only use the instant on and instant on after 50 ms modes. This example does not require any changes to the board. However, the microprocessor code must be modified to treat the HardCopy II device as a non-configurable device.

Figure 20–16 shows an example with two Stratix II devices configured using a microprocessor or MAX® II device and the FPP configuration scheme. For more information on Stratix II configuration, refer to the *Configuration Handbook*.

Figure 20–16. Multiple-Device FPP Configuration Using a Microprocessor or MAX II Device

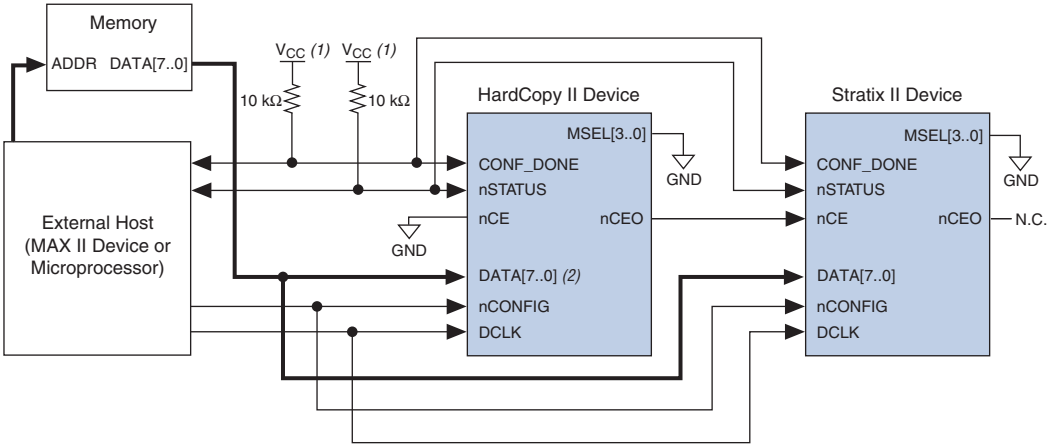


Note to Figure 20–16:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. The V_{CC} voltage meets the I/O standard's V_{IH} specification on the device and the external host.

Figure 20–17 shows how the first Stratix II device is replaced by a HardCopy II device. In this case, the microprocessor code must be modified to send configuration data only to the second device (the Stratix II device) of the configuration chain. The microprocessor can only send this data after its nCE pin is asserted by the first device (the HardCopy II device).

Figure 20–17. Replacement of the First FPGA in the FPP Configuration Chain With a HardCopy Series Device



Notes to Figure 20–17:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. The V_{CC} voltage meets the I/O standard's V_{IH} specification on the device and the external host.

(2) The `DATA[7..0]` pins are not used on the HardCopy II device, but they preserve the pin assignment and direction from the Stratix II device, allowing drop-in replacement.

Conclusion

HardCopy series devices can emulate a configuration sequence while maintaining the seamless migration benefits of the HardCopy methodology. Instant on mode, which is the simplest of the available options, provides ASIC-like operation at power on. This mode can be used in most cases without regard to the original FPGA configuration mode and without any hardware and/or software changes.

In some cases, however, a software revision and/or a board re-design may be necessary to guarantee that correct configuration data is sent to the remaining programmable devices. Such modifications are easily made in the early stages of the board design process if it is determined that one or more of the FPGAs will be replaced with an equivalent HardCopy series device. Board-design techniques like jumper connectors and 0-Ω resistors enable such modifications without the necessity to re-design the board.

The instant on after 50 ms mode is suitable in cases where a delay is necessary to accommodate the configuration device to become operational, or to allow one or more pre-determined events to be completed before the HardCopy series device asserts its `CONF_DONE` pin.

Finally, the emulation mode is the option to choose if software or hardware modifications are not possible. In such cases, the HardCopy series device co-exists with other FPGAs.

Section V. HardCopy Design Center Migration Process

This section provides information on the software support for HardCopy® Stratix® devices.

This section contains the following:

- Chapter 21, Back-End Design Flow for Hardcopy Series Devices
- Chapter 22, Back-End Timing Closure for HardCopy Series Devices

Revision History

The table below shows the revision history for Chapters 21 through 22.

Chapter(s)	Date / Version	Changes Made
Chapter 21	March 2006	Formerly chapter 13; no content change.
	October 2005 v1.1	• Graphic updates • Minor edits
	January 2005 v1.0	Initial release of Chapter 13, Back-End Design Flow for HardCopy Series Devices.
Chapter 22	March 2006	Formerly chapter 17; no content change.
	October 2005 v2.1	• Moved *Chapter 16 Back-End Timing Closure for Hardcopy Series Devices* to Chapter 17 in *HardCopy Series Device Handbook* release 3.2 • Updated graphics • Minor edits
	January 2005 v2.0	• Chapter title changed to *Back-End Timing Closure for HardCopy Series Devices*. • Sizes of silicon technology updated in "Timing Closure" on page 17–2. • HardCopy® Stratix® and HardCopy APEX™ equivalence to their respective FPGA is updated on page 17–2. • Stratix II migration added on page page 22–2. • Updated Table 17–2 on page 17–12. • Updated last paragraph in "Timing ECOs" on page 17–18.
	June 2003 v1.0	Initial release of Chapter 17, Back-End Timing Closure for HardCopy Series Devices.

21. Back-End Design Flow for HardCopy Series Devices

H51019-1.1

Introduction

This chapter discusses the back-end design flow executed by the HardCopy® Design Center when developing your HardCopy series device. The chapter is divided into two sections:

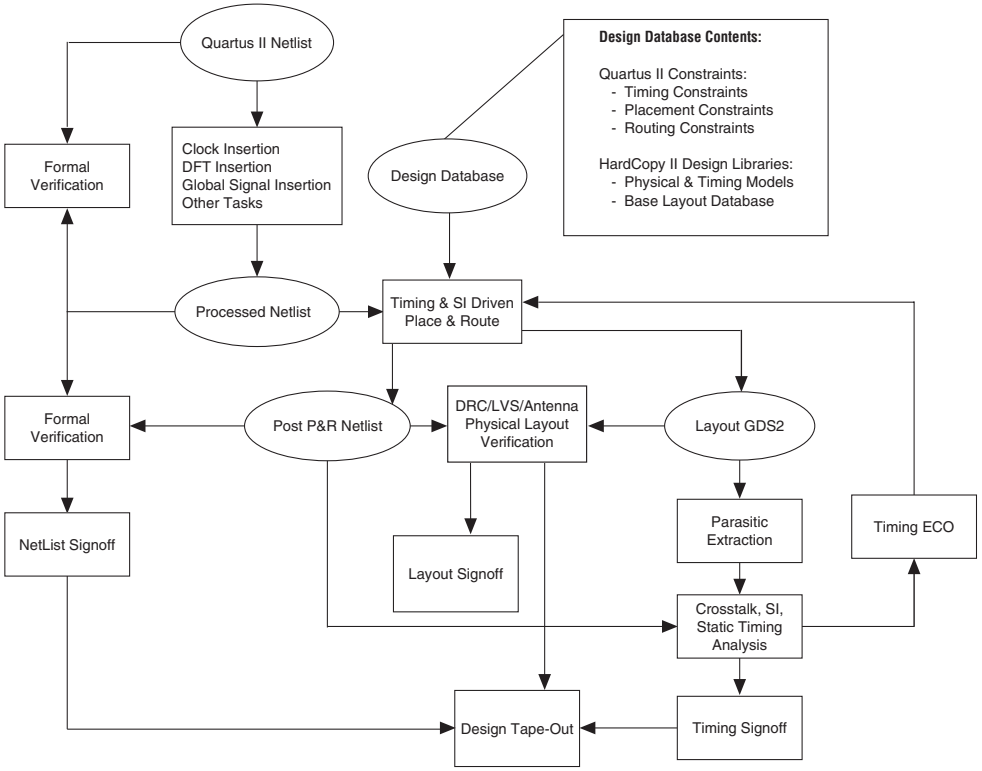
- HardCopy II Back-End Design Flow
- HardCopy Stratix® and HardCopy APEX™ Back-End Design Flow

For more information on the HardCopy II, HardCopy Stratix, and HardCopy APEX families, refer to the respective sections for these families in the *HardCopy Series Handbook*.

HardCopy II Back-End Design Flow

This section outlines the back-end design process for HardCopy II devices, which occurs in several steps. Figure 21–1 illustrates these steps. The design process uses both proprietary and third-party EDA tools. The HardCopy II device design flow is different from that of previous HardCopy families (HardCopy Stratix and HardCopy APEX devices). The following sections outline these differences.

Figure 21–1. HardCopy II Back-End Design Flow



Device Netlist Generation

For HardCopy II designs, the Quartus® II software generates a complete Verilog gate level netlist of your design. The HardCopy Design Center uses the netlist to start the migration process. HardCopy Stratix and HardCopy APEX designs use the SRAM Object file (**.sof**) to program the FPGA, as the primary starting point for generating the HardCopy device netlist.

HardCopy Stratix and HardCopy APEX designs use the **.sof** file to program the FPGA, as the primary starting point for generating the HardCopy device netlist. In addition to the Verilog gate level netlist and the **.sof** file, the Quartus II software generates additional information as part of the design database submitted to the HardCopy Design Center. This information includes timing constraints, placement constraints,

global routing information, and much more. Generation of this database provides the HardCopy Design Center with the necessary information to complete the design of your HardCopy II device.

Design for Testability Insertion

The HardCopy Design Center inserts the necessary test structures into the HardCopy II Verilog netlist. These test structures include full-scan capable registers and scan chains, JTAG, and memory testing. After adding the test structures, the modified netlist is verified using third-party EDA formal verification software against the original Verilog netlist to ensure that the test structures have not broken your netlist functionality. The "Formal Verification of the Processed Netlist"section explains the formal verification process.

Clock Tree & Global Signal Insertion

Along with adding testability, the HardCopy Design Center adds an additional local layer of clock tree buffering to connect the global clock resources to the locally placed registers in the design. Global signals with high fan-out may also use dedicated Global Clock Resources built into the base layers of all HardCopy II devices. The HardCopy Design Center does local buffering.

Formal Verification of the Processed Netlist

After all design-for-testability logic, clock tree buffering, and global signal buffering are added to the processed netlist, the HardCopy Design Center uses third-party EDA formal verification software to compare the processed netlist with your submitted Verilog netlist generated by the Quartus II software. Added test structures are constrained to bypass mode during formal verification to verify that your design's intended functionality was not broken.

Timing & Signal Integrity Driven Place & Route

Placement and global signal routing is principally done in the Quartus II software before submitting the HardCopy II design to the HardCopy Design Center. Using the Quartus II software, you control the placement and timing driven placement optimization of your design. The Quartus II software also does global routing of your signal nets, and passes this information in the design database to the HardCopy Design Center to do the final routing. After submitting the design to the HardCopy Design Center, Altera® engineers use the placement and global routing information provided in the design database to do final routing and timing closure and to perform signal integrity and crosstalk analysis. This may require buffer and delay cell insertion in the design through an

engineering change order (ECO). The resulting post-place and route netlist is verified again with the source netlist and the processed netlist to guarantee that functionality was not altered in the process.

Parasitic Extraction & Timing Analysis

After doing placement and routing on the design by the HardCopy Design Center, it generates the `gds2` design file and extracts the parasitic resistance and capacitance values for timing analysis. Parasitic extraction uses the physical layout of the design stored in a **.gds2** file to extract these resistance and capacitance values for all signal nets in the design. The HardCopy Design Center uses these parasitic values to calculate the path delays through the design for static timing analysis and crosstalk analysis.

Layout Verification

When the Timing Analysis reports that all timing requirements are met, the design layout goes into the final stage of verification for manufacturability. The HardCopy Design Center performs physical Design Rule Checking (DRC), antenna checking of long traces of signals in the layout, and a comparison of layout to the design netlist, commonly referred to as Layout Versus Schematic (LVS). These tasks guarantee that the layout contains the exact logic represented in the place-and-route netlist, and the physical layout conforms to 90-nm manufacturing rules.

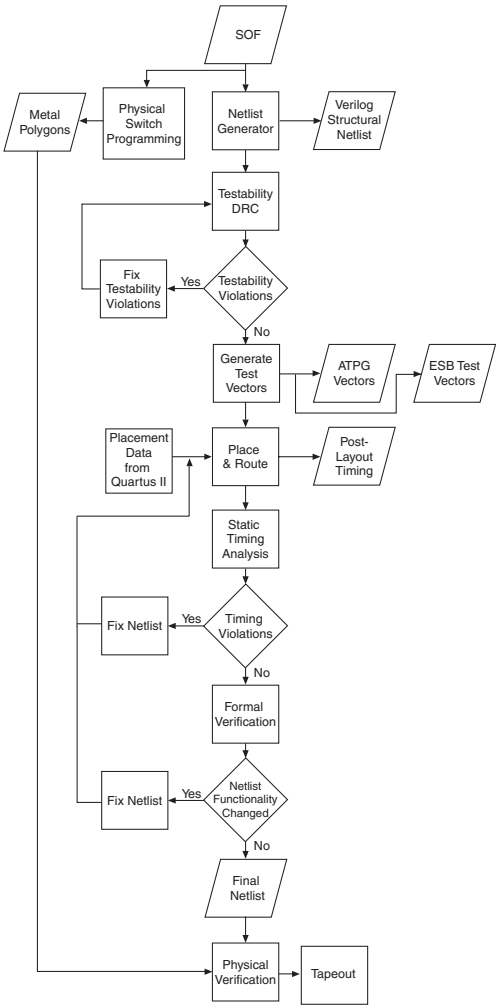
Design Signoff

The Altera HardCopy II back-end design methodology has a thorough verification and signoff process, guaranteeing your design's functionality. Signoff occurs after confirming the final place-and-route netlist functional verification, confirming layout verification for manufacturability, and the timing analysis reports meeting all requirements. After achieving all three signoff points, Altera begins the manufacturing of the HardCopy II devices.

HardCopy Stratix & HardCopy APEX Migration Flow

Design migration for HardCopy Stratix and HardCopy APEX devices occurs in several steps, outlined in this section and shown in Figure 21–2. The migration process uses both proprietary and third-party EDA tools.

Figure 21–2. HardCopy Stratix & HardCopy APEX Migration Flow Diagram



Netlist Generation

For HardCopy Stratix and HardCopy APEX designs, Altera migrates the Quartus II software-generated **.sof** file to a Verilog HDL structural netlist that describes how the following structural elements are configured in the design and how each structural element is connected to other structural elements:

- Logic element (LE)
- Phase-locked loop (PLL)
- Digital signal processing (DSP) block
- Memory block
- Input/output element (IOE)

The information that describes the structural element configuration is converted into a physical coordinate format so that metal elements can be implemented on top of the pre-defined HardCopy series device-base array. Using the **.sof** file for netlist extraction helps ensure that the HardCopy series device contains the same functional implementation that was used in the FPGA version of the design.

Testability Audit

The Design Center performs an audit for testability violations when the Verilog HDL netlist is available. This audit ensures that all built-in scan chain structures will work reliably while testing the HardCopy series devices. Certain circuit structures, such as gated clocks, gated resets, oscillators, pulse generators, or other types of asynchronous circuit structures makes the performance of scan chain structures unreliable. During the testability audit, all such circuit structures are detected and disabled when the device is put into test mode.

Placement

Beginning with version 4.2, the Quartus II software supports all HardCopy series devices. The HardCopy Timing Optimization Wizard in the Quartus II software is used for HardCopy Stratix devices and generates placement information of the design when it is mapped to the HardCopy Stratix base array. This placement information is read in and directly used by the place-and-route tool during migration to the equivalent HardCopy Stratix device.

For more information on how to use the HardCopy Timing Optimization Wizard, refer to the *Quartus II Support for HardCopy Stratix Devices* chapter. For more information on Quartus II features for HardCopy II devices, refer to the *Quartus II Support for HardCopy II Devices* chapter.

To generate placement data, the Quartus II software uses the **.sof** file to generate the netlist, as described in "Netlist Generation". The netlist is then read into a place-and-route tool. The placement optimization is based on the netlist connectivity and the design's timing constraints. The placement of all IOEs is fixed. After placement is complete, the Quartus II software generates the scan chain ordering information so the scan paths can be connected.

Test Vector Generation

Memory test vectors and memory built-in self-test (BIST) circuitry ensure that all memory bits function correctly. Automatic test pattern generation (ATPG) vectors test all LE, DSP, and IOE logic. These vectors ensure that a high stuck-at-fault coverage is achieved. The target fault coverage for all HardCopy Stratix devices is near 100%.

When the testability audit is successfully completed and the scan chains have been re-ordered, the Design Center can generate memory and ATPG test vectors. When test vector generations are complete, they are simulated to verify their correctness.

Routing

Routing involves generating the physical interconnect between every element in the design. At this stage, physical design rule violations are fixed. For example, nodes with large fan-outs need to be buffered. Otherwise, these signal transition times are too slow, and the device's power consumption increases. All other types of physical design rule violations are also fixed during this stage, such as antenna violations, crosstalk violations, and metal spacing violations.

Extracted Delay Calculation

Interconnect parasitic capacitance and resistance information is generated after the routing is complete. This information is then converted into a Standard Delay File (**.sdf)** with a delay calculation tool, and timing is generated for minimum and maximum delays.

Static Timing Analysis & Timing Closure

The design timing is checked and corrected after place and route using the post-layout generated **.sdf** file. Setup time violations are corrected in two ways. First, extra buffers can be inserted to speed up slow signals. Second, if buffer insertion does not completely fix the setup violation, the placement can be re-optimized.

Setup time violations are rare in HardCopy II and HardCopy Stratix devices because the die sizes are considerably smaller than the equivalent Stratix II and Stratix devices. Statistically, the interconnect loading and distance is much smaller in HardCopy Stratix devices, so the device operates at a higher clock frequency. Hold-time violations are fixed by inserting delay elements into fast data paths.

As part of the timing analysis process, crosstalk analysis is also performed to remove any crosstalk effects that could be encountered in the device after it has been manufactured. This ensures signal integrity in the device resulting in proper functionality and satisfactory performance.

After implementing all timing violation corrections in the netlist, the place and route is updated to reflect the changes. This process is repeated until all timing violations are removed. Typically, only a single iteration is required after the initial place and route. Finally, static functional verification is tested after this stage to double-check the netlist integrity.

Formal Verification

After any change to the netlist, you must verify its integrity through static functional verification (or formal verification) techniques. These techniques show whether two versions of a design are functionally identical when certain constraints are applied. For example, after test fixes, the netlist must be logically equivalent to the netlist state before test fixes, when the test mode is disabled. This technique does not rely on any customer-supplied functional simulation vectors.Altera uses third-party formal verification software to confirm that the back-end implementation matches the netlist generated from the FPGA's **.sof** programming file.

Physical Verification

Before manufacturing the metal customization layers, the physical programming information must be verified. This stage involves cross-checking for physical design rule violations in the layout database, and also checking that the circuit was physically implemented correctly. These processes are commonly known as running design rule check and layout-versus-schematic verification.

Manufacturing

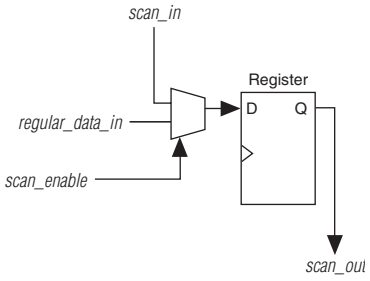
Metallization masks are created to manufacture HardCopy series devices. After manufacturing, the parts are tested using the test vectors that were developed as part of the implementation process.

Testing

HardCopy series devices are fully tested as part of the manufacturing process. Testing does not require user-specific simulation vectors, because every HardCopy series device utilizes full scan path technology. This means that every node inside the device is both controllable and observable through one or more of the package pins of the device. The scan paths, or "scan chains," are exercised through ATPG. This ensures a high-confidence level in the detection of all manufacturing defects.

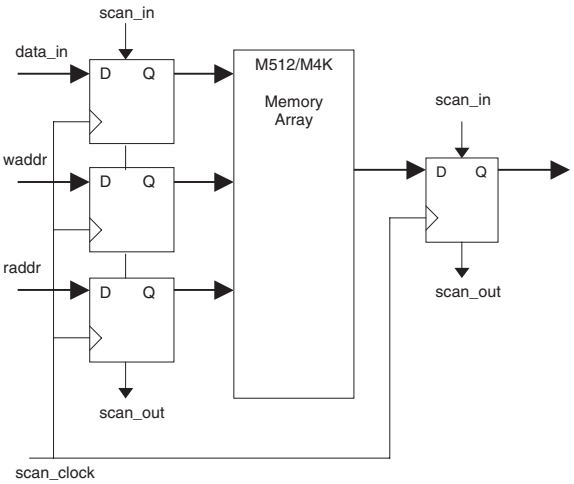
Every register in the HardCopy series device belongs to a scan chain. Scan chains are test features that exist in ASICs to ensure that there is access to all internal nodes of a design. With scan chains, defective parts can be screened out during the manufacturing process. Scan chain registers are constructed by combining the original FPGA register with a 2-to-1 multiplexer. In normal user mode, the multiplexer is transparent to the user. In scan mode, the registers in the device are connected into a long-shift register so that automatic test pattern generation vectors can be scanned into and out of the device. Several independent scan chains exist in the HardCopy series device to keep scan chain lengths short, and are run in parallel to keep tester time per device short. Figure 21–3 shows a diagram of a scan register.

Figure 21–3. HardCopy Stratix Scan Chain Circuitry



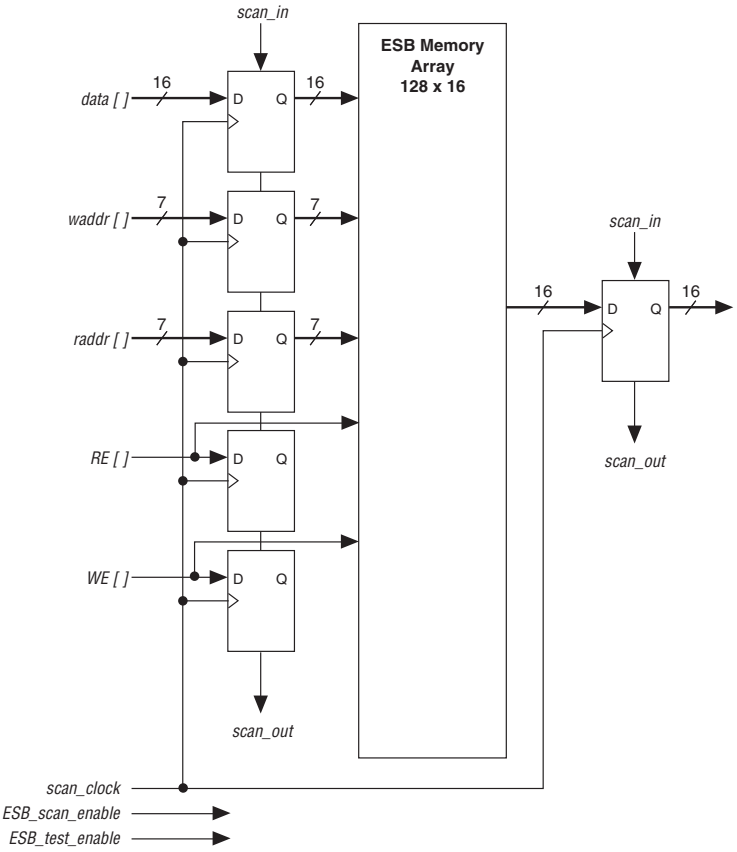
In addition to the scan circuitry (Figure 21–3), which is designed to test all LEs and IOEs, both M512 and M4K blocks (Figure 21–4) have the same scan chain structure so that all bits inside the memory array are tested for correct operation. The M512 and M4K RAM bits are tested by scanning data into the M512 and M4K blocks' `data_in`, write address (`waddr`), and read address (`raddr`) registers. After each vector has been scanned into the HardCopy Stratix device, a write enable (`WE`) pulse is generated to write the data into the M512 and M4K blocks. A read enable (`RE`) pulse is also generated to read data out of the M512 and M4K blocks. The data read back from the M512 and M4K blocks are scanned out of the device via the `data_out` registers. Figure 21–4 shows the M512 and M4K blocks' scan chain connectivity.

Figure 21–4. HardCopy Stratix M512 & M4K Block Scan Chain Connectivity



For HardCopy APEX devices, every embedded system block (ESB) contains dedicated test circuitry so that all bits inside the memory array are tested for correct operation. Access to the ESB memory is also facilitated through scan chains. The ESB also offers an ESB test mode in which the ESB is reconfigured into a 128 × 16 RAM block. In this mode, data is scanned into the ESB I/O registers and written into the ESB memory. For ESBs configured as product-term logic or ROM, the write-enable signal has no effect on the ESB memory array data. When the test mode is disabled (the default), the ESB reverts to the desired user functionality. Figure 21–5 shows the ESB test mode configuration.

Figure 21–5. HardCopy APEX ESB Test Mode Configuration



PLLs and M-RAM blocks are tested with BIST circuitry and test point additions. All test circuitry is disabled once the device is installed into the end user system so that the device then behaves in the expected normal functional mode.

Unused Resources

Unused resources in a customer design still exist in the HardCopy base. However, these resources are configured into a "parked" state. This is a state where all input pins of an unused resource are tied off to V_{CC} or GND so that the resource is in a low-power state. This is achieved using the same metal layers that are used to configure and connect all resources used in the design.

Conclusion

The HardCopy series back-end design methodology ensures that your design seamlessly migrates from your prototype FPGA to a HardCopy device. This methodology, matched with Altera's unique FPGA prototyping and migration process, provides an excellent way for you to develop your design for production.

For more information about how to start building your HardCopy series design, contact your Altera Field Applications Engineer.

For more information on HardCopy products and solutions, refer to the *HardCopy Series Handbook*.

22. Back-End Timing Closure for HardCopy Series Devices

H51013-2.1

Introduction

Back-end implementation of HardCopy® series devices meet design requirements through a proven timing closure flow. The timing closure process is similar to the methodology used for today's standard cell ASICs. Altera uses industry leading EDA software to complete the back-end layout and timing extraction of HardCopy series designs.

The Quartus® II software provides an estimation of your HardCopy design performance. The Altera® HardCopy Design Center extracts the final timing results after the HardCopy device back-end design process is completed.

For more information on the HardCopy back-end design flow, refer to the *HardCopy Series Back-End Design Flow* chapter in the *HardCopy Series Device Handbook*.

This chapter describes how Altera ensures that HardCopy series devices meet their required timing performance.

Timing Analysis of HardCopy Prototype Device

You should perform timing analysis on the FPGA prototype implementation of the design before migrating to HardCopy. Timing analysis determines whether the design's performance meets the required timing goals.

The timing analysis includes system clock frequency (f_{MAX}), setup and hold timing for the design's top-level input ports, as well as clock-to-output timing for all top-level output ports. Measuring these parameters against performance goals ensures that the FPGA design functions as planned in the end-target system.

For more information on timing analysis of Altera devices, refer to the *Timing Analysis* section in volume 2 of the *Quartus II Handbook*.

After the FPGA design is stabilized, fully tested in-system, and satisfies the HardCopy series design rules, the design can be migrated to a HardCopy series device. Altera performs rigorous timing analysis on the HardCopy series device during its implementation, ensuring that it meets the required timing goals. Because the critical timing paths of the HardCopy version of a design may be different from the corresponding

paths in the FPGA version, meeting the required timing goals constrained in the Quartus II software is particularly important. Additional performance gains are design dependent, and the percentage of performance improvement can be different for each clock domain of your design.

Timing differences between the FPGA design and the equivalent HardCopy series device can exist for several reasons. While maintaining the same set of features as the corresponding FPGA, HardCopy series devices have a highly optimized die size to make them as small as possible. Because of the customized interconnect structure that makes this optimization possible, the delay through each signal path is different from the original FPGA design.

Timing Closure

Many of today's developers must meet the timing goals of systems designed with an ASIC, which can consume many months of engineering effort. The slower development process exists because, in today's silicon technology (0.18 µm, 0.13 µm, and 90 nm), the delay associated with interconnect dominates the delay associated with the transistors used to make the logic gates. Consequently, ASIC performance is sensitive to the physical placement and routing of the logic blocks that make up the design.

HardCopy II devices use timing constraints to drive placement and routing of logic into the fabric of HCells. Each Stratix II Adaptive Look-up Table (ALUT) is implemented in HCell Macros in the HardCopy II device. HCell Macros are pre-defined and characterized libraries built out of HCells. The Quartus II software performs the placement and global routing of all HCell Macros and this information is transferred to the HardCopy Design Center for final back-end implementation and timing closure.

HardCopy Stratix® and HardCopy APEX™ are structurally identical to their respective FPGA counterparts. There is no re-synthesis or library re-mapping required. Since the interconnect lengths are much smaller in the HardCopy series device than they are in the FPGA, the place-and-route engine compiling the HardCopy series design has a considerably less difficult task than it does in an equivalent ASIC development. Coupled with detailed timing constraints, the place-and-route is timing driven.

Minimizing Clock Skew in HardCopy II

HardCopy II devices offer a fine-grained architecture of HCells which are used to build HCell Macros for standard logic functions. The pre-built metal layers of HardCopy II devices contain the same global clock tree resources as those available in Stratix II devices, though they are smaller in HardCopy II devices because of the difference in die size. The end

point of the dedicated global clock network in HardCopy II is not dedicated routing in the fixed metal layers. The last stage of clock tree buffering and routing is done using metal in the custom routing layers. Local buffering can also be done using HCell Macros to fix any clock skew issues. The need for doing this is because HCell Macros are used to create registers, and local custom routing is needed to connect the clock networks to these HCell Macro registers. These tasks are automated as part of the HardCopy Design Center process.

Minimizing Clock Skew in HardCopy Stratix

HardCopy Stratix devices have the same global clock tree resources as Stratix FPGA devices. The construction of non-customizable layers of silicon minimizes global clock tree skew. HardCopy Stratix devices with clock trees using global clock resources have smaller clock insertion delay than Stratix FPGA devices because the HardCopy Stratix devices have a smaller die area. The use of clock tree synthesis to build small-localized clock trees using the existing buffer resources that are available in HardCopy Stratix devices automatically implements clock trees using fast regional clock resources in Stratix FPGA devices.

Minimizing Clock Skew in HardCopy APEX

The HardCopy APEX device architecture is based on the APEX 20KE and APEX 20KC devices. The same dedicated clock trees (`CLK[3..0]`) that exist in APEX 20KE and APEX 20KC devices also exist in the corresponding HardCopy APEX device. These clock trees are carefully designed and optimized to minimize the clock skew over the entire device. The clock tree is balanced by maintaining the same loading at the end of each point of the clock tree, regardless of what resources (logic elements [LEs], embedded system blocks [ESBs], and input/output elements [IOEs]) are used in any design. The insertion delay of the HardCopy APEX dedicated clock trees is marginally faster than in the corresponding APEX 20KE or APEX 20KC FPGA device because of the smaller footprint of the HardCopy device silicon. This difference is less than 1 ns.

Because there is a large area overhead for the global signals that may not be used on every design, the FAST bidirectional pins (`FAST[3..0]`) do not have dedicated pre-built clock or buffer trees in HardCopy APEX devices. If any of the FAST signals are used as clocks, the place-and-route tool synthesizes a clock tree after the placement of the design has occurred. The skew and insertion delay of these synthesized clock trees is carefully controlled, ensuring that the timing requirements of the design are met. You can also use the FAST signals as high fan-out reset or enable signals. For these cases, skew is usually less important than insertion delay. To reiterate, a buffer tree is synthesized after the design placement.

The clock or buffer trees that are synthesized for the FAST pins are built out of special cells in the HardCopy APEX base design. These cells do not exist in the FPGA, and are used in the HardCopy APEX design exclusively to meet timing and testing goals. They are not available to make any logical changes to the design as implemented in the FPGA. These resources are called the strip of auxiliary gates (SOAG). There is one strip per MegaLAB™ structure in HardCopy APEX devices. Each SOAG consists of a number of primitive cells, and there are approximately 10 SOAG primitive cells per logic array block (LAB). Several SOAG primitives can be combined to form more complex logic, but the majority of SOAG resources are used for buffer tree, clock tree, and delay cell generation.

For detailed information on the HardCopy series device architecture, including SOAG resources, refer to the *HardCopy APEX Device Family Data Sheet* section in volume 1 of the *HardCopy Series Handbook*.

Checking the HardCopy Series Device Timing

To ensure that the timing of the HardCopy series device meets performance goals, you must run detailed static timing analysis on the HardCopy series design database. For this timing analysis to be meaningful, all timing constraints and timing exceptions that you applied to the design for the FPGA implementation, must also be used for the HardCopy implementation. If you did not use timing constraints or you used only partial timing constraints for the FPGA design, you must fully specify them for the HardCopy series design. If you do not do this, there is no way of knowing if the HardCopy series device meets the required timing of the end target system. The timing constraints of the FPGA can be captured through the HardCopy Timing Optimization Wizard in the Altera Quartus II software.

The following is the list of constraints that must be included:

- Clock definitions
- Primary input pin timing
- Primary output pin timing
- Combinational timing
- Timing exceptions

Clock Definitions

You can use these definitions to describe the parameters of all different clock domains in a design. Clock parameters that must be defined are frequency, time at which the clock edge rises, time at which the clock edge falls, clock uncertainty (or skew), and clock name. Figures 22–1 and 22–2 show these clock attributes.

Figure 22–1. Clock Attributes

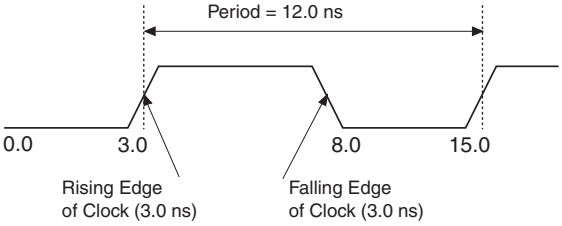
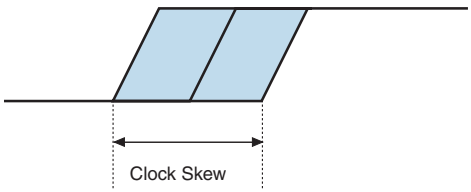


Figure 22–2. Clock Skew



Primary Input Pin Timing

You must specify the primary input pin timing constraint for every primary input pin in the design (and for the input path of every bidirectional pin). The input pin timing can be constrained as described in one of the following two subsections.

Describe the Acceptable Maximum On-Chip Input Delay

This approach describes the acceptable maximum on-chip delay. For example, the setup time of a primary input to any register in the design relative to a specific clock. Figure 22–3 shows a generic circuit with an on-chip setup-time constraint, which may be different for each clock domain. You must specify the minimum on-chip delay from any primary input pin to describe input hold-time requirements. Figure 22–4 shows a generic circuit with an on-chip hold-time constraint.

Figure 22–3. On-Chip Setup-Time Constraint

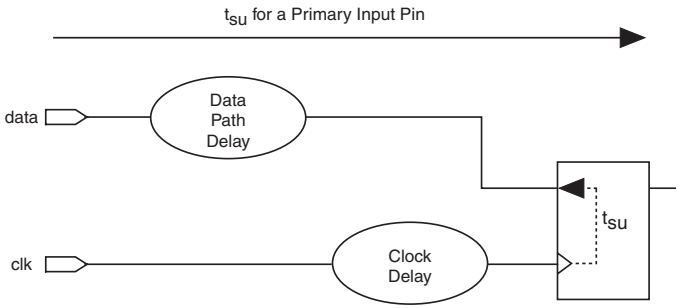
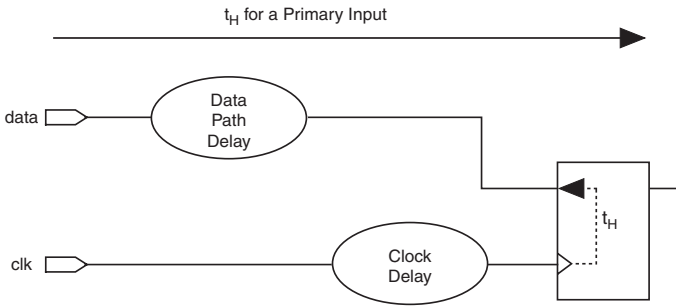


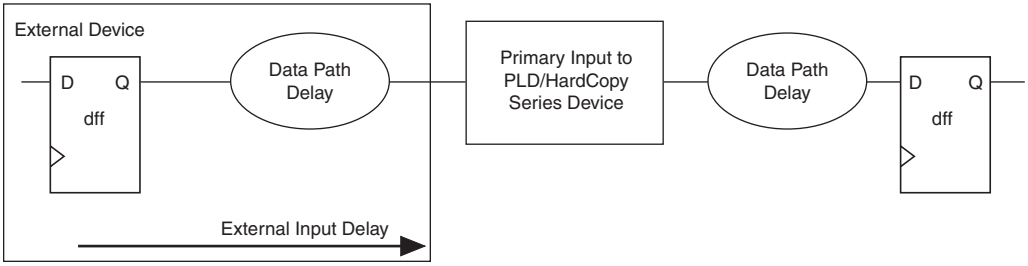
Figure 22–4. On-Chip Hold-Time Constraint



Describe the External Input Timing Environment

Another way to constrain the input pin timing is to describe the external timing environment, which is the maximum and minimum arrival times of the external signals that drive the primary input pins of the HardCopy series device or FPGA. Figure 22–5 shows the external timing constraint that drives the primary input pin. The static timing analysis tool can use this external input delay time to check if there is enough time for the data to propagate to the internal nodes of the device. If there is not enough time, a timing violation occurs.

Figure 22–5. External Timing Constraint Driving a Primary Input Pin



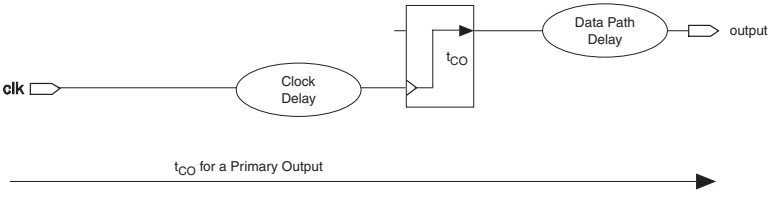
Primary Output Pin Timing

You must specify the output pin timing constraint for every primary output pin in the design and for the output path of every bidirectional pin. There are two ways to capture the output pin timing as described in one of the following two sections.

Describe the Acceptable Maximum and Minimum Clock-to-Output Delay

This approach describes the acceptable maximum and minimum on-chip clock-to-output (t_{CO}) delay. For example, the time it takes from the active edge of the clock to the data arriving at the primary output pin. Figure 22–6 shows a generic circuit with an on-chip t_{CO} time constraint. In addition, there can be a minimum t_{CO} requirement.

Figure 22–6. On-Chip Clock-to-Output (t_{CO}) Time Constraint

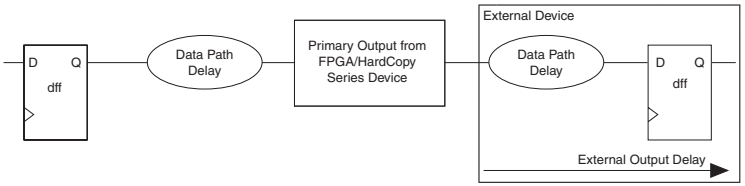


Describe the Acceptable External Output Delay

Another way to capture output pin timing is to describe the external timing environment, which is the maximum and minimum delay times of external signals that are driven by the primary output pins of the HardCopy series device. Figure 22–7 shows the external timing

constraint driven by the primary output pin. The static timing analysis tool uses this information to check that the on-chip timing of the output signals is within the desired specification.

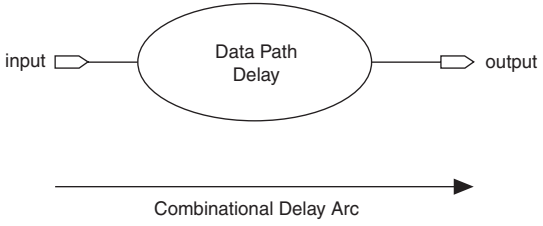
Figure 22–7. External Timing Constraint for a Primary Output Pin



Combinational Timing

In combinational timing circuits, a path exists from a primary input pin to a primary output pin. This type of circuit does not contain any registers. Therefore, it does not require a clock for constraint specification. You only need the maximum and minimum delay from the primary input pin to the primary output pin to constrain the path for timing requirements. Figure 22–8 shows for a generic circuit, the placement requirement for a combinational delay arc constraint.

Figure 22–8. Combinational Timing Constraint



Timing Exceptions

Some circuit structures warrant special consideration. For example, you can ignore all timing paths between the two-clock domain when a design has more than one clock domain and the clock domains are not related. You can ignore all timing paths using the static timing analysis tool by specifying false paths for all signals that go from one clock domain to the other clock domain(s). Additionally, some circuits are not intended to operate in a single-clock cycle. These circuits require that you specify multi-cycle clock exceptions.

After capturing the information, Altera directly checks all timing of the HardCopy series device before tape-out occurs. If any timing violations occur in the HardCopy series device due to over aggressive timing constraints, Altera must fix them, or you must waive them.

Correcting Timing Violations

After generating the customized metal interconnect for the HardCopy series device, Altera checks the design timing with a static timing analysis tool. The static timing analysis tool reports timing violations and then the HardCopy Design Center corrects the violations.

Hold-Time Violations

Because the interconnect in a HardCopy series device is customized for a particular application, it is possible that hold-time (t_H) violations exist in the HardCopy series device after place-and-route occurs. A hold violation exists if the sum of the delay in the clock path between two registers plus the micro hold time of the destination register is greater than the delay of the data path from the source register to the destination register. The following equation describes this relationship.

$$t_H \text{ slack} = \text{data delay} - \text{clock delay} - \mu t_H$$

If a negative slack value exists, this means that there is a hold-time violation. Any hold-time violation present in the HardCopy series design database after the interconnect data is generated is removed by inserting the appropriate delay in the data path. The inserted delay is large enough to guarantee no hold violation under fast, low-temperature, high-voltage conditions.

An Example HardCopy APEX Hold-Time Violation Fix

Table 22–1 shows an example report of a Synopsys PrimeTime static timing analysis of a HardCopy APEX design. The first report shows that the circuit has a hold-time violation and a negative slack value. The second result shows the timing report for the same path after fixing the hold violation. Part of the HardCopy implementation process is to generate the instance and cell names shown in these reports. The physical location of those elements in the device determines the generation of the names.

Table 22–1. HardCopy APEX Static Timing Analysis Before Hold-Time Violation Fix

```
Startpoint: GR23_GC0_L19_LE1/um6
(falling edge-triggered flip-flop clocked by CLK0')
Endpoint: GR23_GC0_L20_LE8/um6
(falling edge-triggered flip-flop clocked by CLK0')
Path Group: CLK0
Path Type: min
```

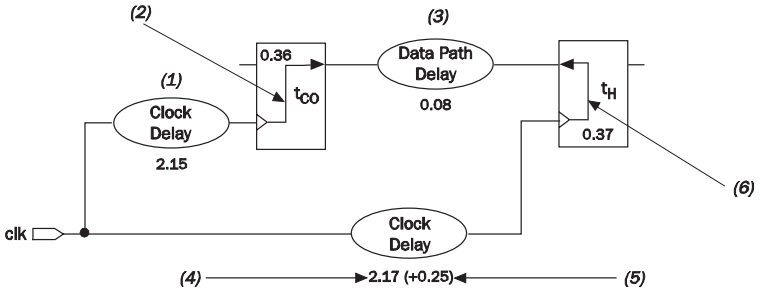
Point	Incr	Path	Reference Point *(1)*
clock CLK0' (fall edge)	0.00	0.00	
clock network delay (propagated)	2.15	2.15	(1)
GR23_GC0_L19_LE1/um6/clk (c1110)	0.00	2.15 f	(2)
GR23_GC0_L19_LE1/um6/regout (c1110)	0.36 *	2.52 r	(2)
GR23_GC0_L19_LE1/REGOUT (c1000_2d7a8)	0.00	2.52 r	(2)
GR23_GC0_L20_LE8/LUTD (c1000_56502)	0.00	2.52 r	(3)
GR23_GC0_L20_LE8/um1/datad (indsim)	0.01 *	2.52 r	(3)
GR23_GC0_L20_LE8/um1/ndsim (indsim)	0.01 *	2.53 f	(3)
GR23_GC0_L20_LE8/um5/ndsim (mxcascout)	0.00 *	2.53 f	(3)
GR23_GC0_L20_LE8/um5/cascout	0.06 *	2.59 f	(3)
GR23_GC0_L20_LE8/um6/dcout (c1110)	0.00 *	2.59 f	(3)
data arrival time		2.59	
clock CLK0' (fall edge)	0.00	0.00	
clock network delay (propagated)	2.17	2.17	(4)
clock uncertainty	0.25	2.42	(5)
GR23_GC0_L20_LE8/um6/clk (c1110)		2.42 f	(6)
library hold time	0.37 *	2.79	
data required time		2.79	
data arrival time		2.59	
data required time		-2.79	
slack (VIOLATED)		-0.20	

Note to Table 22–1:

(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 22–9.

Figure 22–9 shows the circuit described by the Table 22–1 static timing analysis report.

Figure 22–9. Circuit With a Hold-Time Violation



Placing the values from the static timing analysis report into the hold-time slack equation results in the following:

t_H slack = data delay –clock delay –µt_H

t_H slack = (2.15 + 0.36 + 0.08) –(2.17 + 0.25) –0.37

t_H slack = -0.20 ns

This result shows that there is negative slack in this path, meaning that there is a hold-time violation of 0.20 ns.

After fixing the hold violation, the timing report for the same path is re-generated (Table 22–2). The netlist changes are in ***bold italic*** type.

Table 22–2. HardCopy APEX Static Timing Analysis After Hold-Time Violation Fix

```
Startpoint: GR23_GC0_L19_LE1/um6
(falling edge-triggered flip-flop clocked by CLK0')
Endpoint: GR23_GC0_L20_LE8/um6
(falling edge-triggered flip-flop clocked by CLK0')
Path Group: CLK0
Path Type: min
Static Timing Analysis After Hold-Time Violation Fix
```

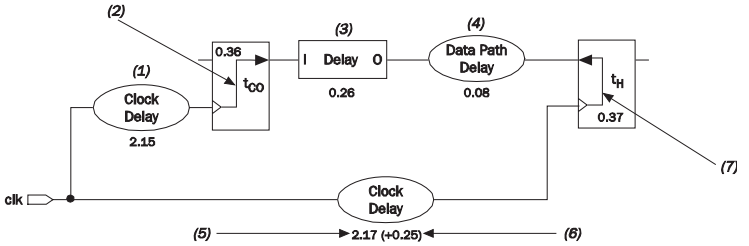
Point	Incr	Path	Reference Point *(1)*
clock CLK0' (fall edge)	0.00	0.00	(1)
clock network delay (propagated)	2.15	2.15	(1)
GR23_GC0_L19_LE1/um6/clk (c1110)	0.00	2.15 f	(2)
GR23_GC0_L19_LE1/um6/regout (c1110)	0.36 *	2.52 r	(2)
GR23_GC0_L19_LE1/REGOUT (c1000_2d7a8)	0.00	2.52 r	(2)
thc_916/A (de105)	***0.01 ****	***2.52 r***	(3)
thc_916/Z (de105)	***0.25 ****	***2.78 r***	(3)
GR23_GC0_L20_LE8/LUTD (c1000_56502)	0.00	2.78 r	(3)
GR23_GC0_L20_LE8/um1/datad (indsim)	0.01 *	2.78 r	(3)
GR23_GC0_L20_LE8/um1/ndsim (indsim)	0.01 *	2.79 f	(3)
GR23_GC0_L20_LE8/um5/ndsim (mxcascout)	0.00 *	2.79 f	(3)
GR23_GC0_L20_LE8/um5/cascout (mxcascout)	0.06 *	2.85 f	(3)
GR23_GC0_L20_LE8/um6/dcout (c1110)	0.00 *	2.85 f	(3)
data arrival time		2.85	
clock CLK0' (fall edge)	0.00	0.00	
clock network delay (propagated)	2.17	2.17	(4)
clock uncertainty	0.25	2.42	(5)
GR23_GC0_L20_LE8/um6/clk (c1110)		2.42 f	(6)
library hold time	0.37 *	2.79	
data required time		2.79	
data arrival time		2.85	
data required time		-2.79	
slack (MET)		0.06	

Note to Table 22–2:

(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 22–10.

Figure 22–10 shows the circuit described by the Table 22–2 static timing analysis report.

Figure 22–10. Circuit Including a Fixed Hold-Time Violation



Placing the values from the static timing analysis report into the hold-time slack equation, results in the following.

$$t_H \text{ slack} = \text{data delay} - \text{clock delay} - \mu t_H$$

$$t_H \text{ slack} = (2.15 + 0.36 + 0.26 + 0.08) - (2.17 + 0.25) - 0.37$$

$$t_H \text{ slack} = + \ 0.06 \text{ ns}$$

In this timing report, the slack of this path is reported as 0.06 ns. Therefore, this path does not have a hold-time violation. This path was fixed by the insertion of a delay cell (`del05`) into the data path, which starts at the REGOUT pin of cell `GR23_GC0_L19_LE1` and finishes at the `LUTD` input of cell `GR23_GC0_L20_LE8`. The instance name of the delay cell in this case is `thc_916`.

This timing report specifies a clock uncertainty of 0.25 ns, and adds extra margin during the hold-time calculation, making the design more robust. This feature is a part of the static timing analysis tool, not of the HardCopy series design.

The SOAG resources that exist in the HardCopy APEX base design create the delay cell. The HardCopy Stratix base design contains auxiliary buffer cells of varying drive strength used to fix setup and hold time violations.

Setup-Time Violations

A setup violation exists if the sum of the delay in the data path between two registers plus the micro setup time (t_{SU}) of the destination register is greater than the sum of the clock period and the clock delay at the destination register. The following equation describes this relationship:

$$t_{SU} \text{ slack} = \text{clock period} + \ \text{clock delay} - (\text{data delay} + \ \mu t_{SU})$$

If there is a negative slack value, it means that there is a setup-time violation. Several potential mechanisms can cause a setup time violation. The first is when the synthesis tool is unable to meet the required timing goals. However, a HardCopy series design does not rely on any re-synthesis to a new cell library; synthesis results are generated as part of the original FPGA design, meaning that the HardCopy implementation of a design uses exactly the same structural netlist as its FPGA counterpart. For example, if you used a particular synthesis option to ensure that a particular path only contained a certain number of logic levels, the HardCopy series design contains exactly the same number of logic levels for that path. Consequently, if the FPGA was free of setup-time violations, no setup-time violations will occur in the HardCopy series device due to the netlist structure.

The second mechanism that can cause setup-time violations is differing placement of the resources in the netlist for the HardCopy series device compared to the original FPGA. This scenario is extremely unlikely as the place-and-route tool used during the HardCopy implementation performs timing-driven placement. In extreme cases, some manual placement modification are necessary. The placement is performed at the LAB and ESB level, meaning that the placement of logic cells inside each LAB is fixed, and is identical to the placement of the FPGA. IOEs have fixed placement to maintain the pin and package compatibility of the original FPGA.

The third, and most likely, mechanism for setup-time violations occurring in the HardCopy series device is a signal with a high fan-out. In the FPGA, high fan-out signals are buffered by large drivers that are integral parts of the programmable interconnect structure. Consequently, a signal that was fast in the FPGA can be initially slower in the HardCopy version. This occurs even though you have not inserting any buffering into the HardCopy series design to increase the speed of the slow signal. The place-and-route tool detects these signals and automatically creates buffer trees using SOAG resources, ensuring that the heavily loaded, high fan-out signal is fast enough to meet performance requirements.

An Example HardCopy APEX Setup-Time Violation Fix

Table 22–3 shows the timing report for a path in a HardCopy APEX design that contains a high fan-out signal before the place-and-route process. Figure 22–4 shows the timing report for a path that contains a high fan-out signal after the place-and-route process. Before the place-and-route process, there is a large delay on the high fan-out net driven by the pin `GR12_GC0_L2_LE4/REGOUT`. This delay is due to the large capacitive load that the pin has to drive. Figure 22–11 shows the timing report information.

.

Table 22–3. HardCopy APEX Timing Report Before Place-&-Route Process

```
Startpoint: GR12_GC0_L2_LE4/um6
(falling edge-triggered flip-flop clocked by clkx')
Endpoint: GR4_GC0_L5_LE2/um6
(falling edge-triggered flip-flop clocked by clkx')
Path Group: clkx
Path Type: max
```

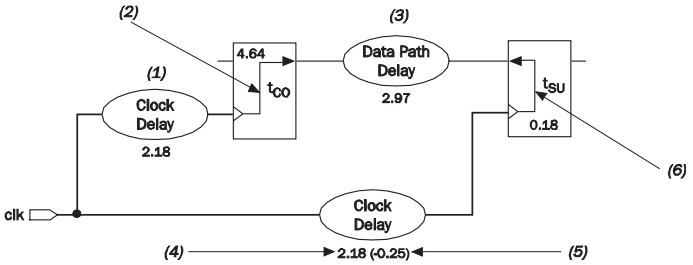
Point	Incr	Path	Reference Point *(1)*
clock clkx' (fall edge)	0.00	0.00	(1)
clock network delay (propagated)	2.18	2.18	(1)
GR12_GC0_L2_LE4/um6/clk (c1110)	0.00	2.18 f	(2)
GR12_GC0_L2_LE4/um6/regout (c1110)			(2)
GR12_GC0_L2_LE4/REGOUT (c1000_7f802) <-			(2)
GR4_GC0_L5_LE0/LUTC (c1000_0029a)			(3)
GR4_GC0_L5_LE0/um4/ltb (lt53b)	2.36	9.18 f	(3)
GR4_GC0_L5_LE0/um5/cascout (mxcascout)	0.07	9.24 f	(3)
GR4_GC0_L5_LE0/um2/COMBOUT (icombout)	0.09	9.34 r	(3)
GR4_GC0_L5_LE0/COMBOUT (c1000_0029a)	0.00	9.34 r	(3)
GR4_GC0_L5_LE2/LUTC (c1000_0381a)	0.00	9.34 r	(3)
GR4_GC0_L5_LE2/um4/ltb (lt03b)	0.40	9.73 r	(3)
GR4_GC0_L5_LE2/um5/cascout (mxcascout)	0.05	9.78 r	(3)
GR4_GC0_L5_LE2/um6/dcout (c1110)	0.00	9.78 r	(3)
data arrival time		9.79	(3)
clock clkx' (fall edge)	7.41	7.41	
clock network delay (propagated)	2.18	9.59	(4)
clock uncertainty	-0.25	9.34	(5)
GR4_GC0_L5_LE2/um6/clk (c1110)		9.34 f	
Point	Incr	Path	Reference Point *(1)*
library setup time	-0.18	9.16	(6)
data required time		9.16	
data required time		9.16	
data arrival time		-9.79	
slack (VIOLATED)		-0.63	

Note to Table 22–3:

(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 22–11.

Figure 22–11 shows the circuit that Table 22–3 static timing analysis report describes.

Figure 22–11. Circuit That Has a Setup-Time Violation



The timing numbers in this report are based on pre-layout estimated delays.

Placing the values from the static timing analysis report into the setup-time slack equation, results in the following.

t_{SU} slack = clock period + clock delay – (data delay +μt_{SU})

t_{SU} slack = 7.41 + (2.18 – 0.25) – (2.18 + 4.64 + 2.97 + 0.18)

t_{SU} slack = –0.63 ns

This result shows that there is negative slack for this path, meaning that there is a setup-time violation of 0.63 ns.

After place-and-route, a buffer tree is constructed on the high fan-out net and the setup-time violation is fixed. Table 22–4 shows the timing report for the same path. The changes to the netlist are in ***bold italic*** type. Figure 22–12 shows more information on this timing report.

.

Table 22–4. HardCopy APEX Timing Report After the Place-&-Route Process

```
Startpoint: GR12_GC0_L2_LE4/um6
(falling edge-triggered flip-flop clocked by clkx')
Endpoint: GR4_GC0_L5_LE2/um6
(falling edge-triggered flip-flop clocked by clkx')
Path Group: clkx
Path Type: max
```

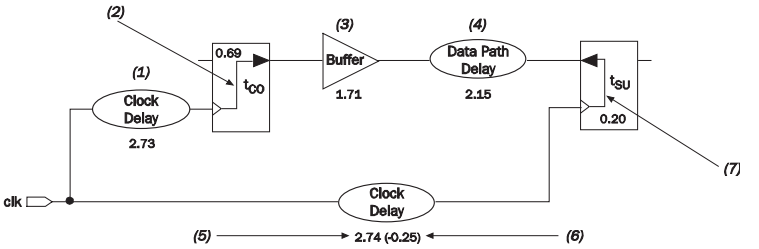
Point	Incr	Path	Reference Point *(1)*
clock clkx' (fall edge)	0.00	0.00	
clock network delay (propagated)	2.73	2.73	(1)
GR12_GC0_L2_LE4/um6/clk (c1110)	0.00	2.73 f	(2)
GR12_GC0_L2_LE4/um6/regout (c1110)	0.69 *	3.42 r	(2)
GR12_GC0_L2_LE4/REGOUT (c1000_7f802) <-	0.00	3.42 r	(2)
N1188_iv06_1_0/Z (iv06)	***0.06 ****	***3.49 f***	(3)
N1188_iv06_2_0/Z (iv06)	***0.19 ****	***3.68 r***	(3)
N1188_iv06_3_0/Z (iv06)	***0.12 ****	***3.80 f***	(3)
N1188_iv06_4_0/Z (iv06)	***0.10 ****	***3.90 r***	(3)
N1188_iv06_5_0/Z (iv06)	***0.08 ****	***3.97 f***	(3)
N1188_iv06_6_2/Z (iv06)	***1.16 ****	***5.13 r***	(3)
GR4_GC0_L5_LE0/LUTC (c1000_0029a)	0.00	5.13 r	(4)
GR4_GC0_L5_LE0/um4/ltb (lt53b)	1.55 *	6.68 f	(4)
GR4_GC0_L5_LE0/um5/cascout (mxcascout)	0.06 *	6.74 f	(4)
GR4_GC0_L5_LE0/um2/COMBOUT (icombout)	0.09 *	6.84 r	(4)
GR4_GC0_L5_LE0/COMBOUT (c1000_0029a)	0.00	6.84 r	(4)
GR4_GC0_L5_LE2/LUTC (c1000_0381a)	0.00	6.84 r	(4)
GR4_GC0_L5_LE2/um4/ltb (lt03b)	0.40 *	7.24 r	(4)
GR4_GC0_L5_LE2/um5/cascout (mxcascout)	0.05 *	7.28 r	(4)
GR4_GC0_L5_LE2/um6/dcout (c1110)	0.00 *	7.28 r	(4)
data arrival time		7.28	(4)
Point	Incr	Path	Reference Point *(1)*
clock clkx' (fall edge)	7.41	7.41	
clock network delay (propagated)	2.74	10.15	(5)
clock uncertainty	-0.25	9.90	(6)
GR4_GC0_L5_LE2/um6/clk (c1110)		9.90 f	
library setup time	-0.20 *	9.70	(7)
data required time		9.70	
data required time		9.70	
data arrival time		-7.28	
slack (MET)		2.42	

Note to Table 22–4:

(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 22–12.

The GR12_GC0_L2_LE4/REGOUT pin now has the loading on it reduced by the introduction of several levels of buffering (in this case, six levels of inverters). The inverters have instance names similar to N1188_iv06_1_0, and are of type iv06, as shown in the static timing analysis report. As a result, the original setup-time violation of –0.63 ns turned into a slack of +2.42 ns, meaning the setup-time violation is fixed. Figure 22–12 illustrates the circuit that the static timing analysis report shows. The buffer tree (buffer) is shown as a single cell.

Figure 22–12. Circuit Post Place-and-Route



Placing the values from the static timing analysis report into the setup-time slack equation, results in the following:

t_{SU} slack = clock period + clock delay – (data delay + μt_{SU})

t_{SU} slack = 7.41 + (2.74 - 0.25) – (2.73 + 0.69 + 1.71 + 2.15 + 0.20)

t_{SU} Slack = +2.42 ns

This result shows that there is positive slack for this path, meaning that there is now no setup-time violation.

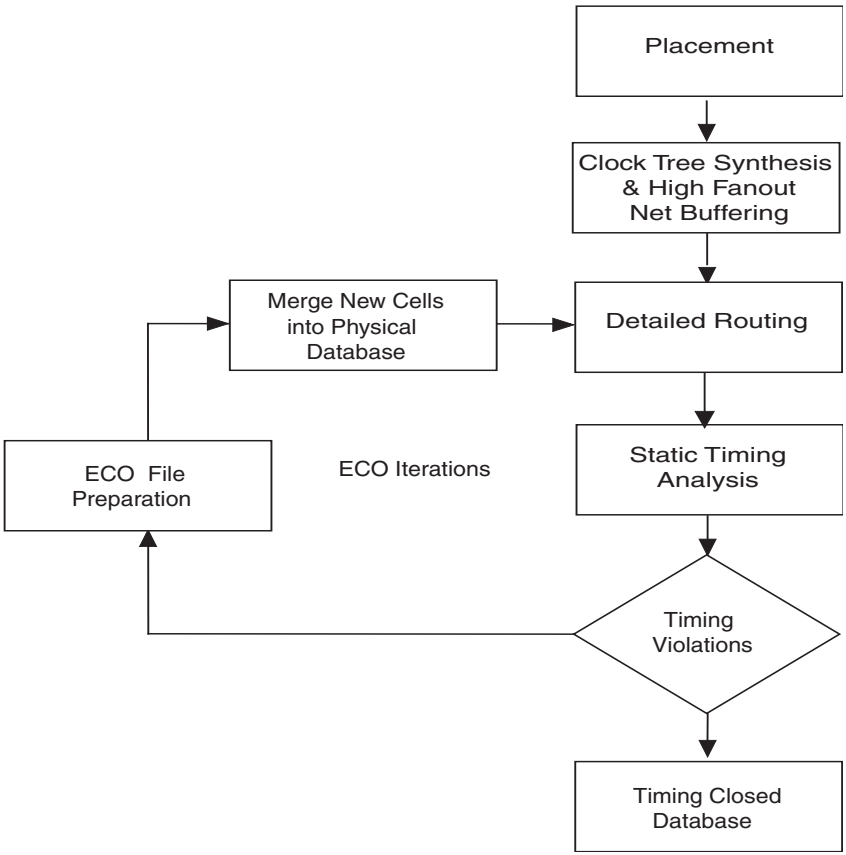
Timing ECOs

In an ASIC, small incremental changes to a design database are termed engineering change orders (ECOs). In the HardCopy series design flow, ECOs are performed after the initial post-layout timing data is available.

You run static timing analysis on the design, which generates a list of paths with timing violations. An automatically updated netlist reflects changes that correct these timing violations (for example, the addition of delay cells to fix hold-time violations). After the netlist update, the updated place-and-route database reflects the netlist changes. The impact to this database is minimal by maintaining all of the pre existing placement and routing, and only changes the routing of newly inserted cells.

The parasitic (undesirable, but unavoidable) resistances and capacitances of the customized interconnect are extracted, and are used in conjunction with the static timing analysis tool to re-check the timing of the design. Detected crosstalk violations on signals are fixed by adding additional buffering to increase the setup or hold margin on victim signals. In-line buffering and small buffer tree insertion is done for signals with high fan-out, high transition times, or high capacitive loading. Figure 22–13 shows this flow in more detail.

Figure 22–13. ECO Flow Diagram



The back-end flow in HardCopy produces the final sign-off timing for your HardCopy device. The Quartus II software produces timing report for HardCopy based on a global route and does not factor in exact physical parasitics of the routed nets, nor does it factor in the crosstalk effect that neighboring nets can have on interconnect capacitance.

Conclusion

When migrating a design from an FPGA implementation to a HardCopy implementation, it is critical to maintain performance even though timing changes will occur within the design. These timing differences are inevitable. They become inconsequential to the device's behavior in the end-system environment if the HardCopy series device meets the system timing constraints. As a standard and automated part of the HardCopy process, this is achieved through the exhaustive timing analysis that the design undergoes in conjunction with sophisticated timing driven place-and-route. Static timing analysis can reveal and automatically fix timing violations, as part of the HardCopy series design process.

Consult with Altera should you have questions on what areas to concentrate your efforts in achieving timing closure within the Quartus fitter for HardCopy design submission.