HardCopy Series Handbook, Volume 1

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

H5V1-2.0

Contents

Chapter 15. Power-Up Modes & Configuration Emulation in HardCopy Series Devices

Chapter 16. Back-End Timing Closure for HardCopy Series Devices

Section V. HardCopy II Design Considerations

Chapter 17. Quartus II Support for HardCopy II Devices

Chapter 18. Prototyping Strategy for HardCopy II Devices

Section VI. HardCopy Stratix Design Considerations

Chapter 19. Quartus II Support for HardCopy Stratix Devices

Index

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: *January 2005*
Part number: *H51015-1.0*

Chapter 2. Description, Architecture & Features
Revised: *January 2005*
Part number: *H51016-1.0*

Chapter 3. Boundary-Scan Support
Revised: *January 2005*
Part number: *H51017-1.0*

Chapter 4. Operating Conditions
Revised: *January 2005*
Part number: *H51018-1.0*

Chapter 5. Introduction to HardCopy Stratix Devices
Revised: *January 2005*
Part number: *H51001-2.0*

Chapter 6. Description, Architecture & Features
Revised: *January 2005*
Part number: *H51002-2.0*

Chapter 7. Boundary-Scan Support
Revised: *January 2005*
Part number: *H51004-2.0*

Chapter 8. Operating Conditions
Revised: *January 2005*
Part number: *H51005-2.0*

Chapter 9. Introduction to HardCopy APEX Devices
Revised: *January 2005*
Part number: *H51006-1.0*

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.

It contains the following:

- Chapter 1, Introduction to HardCopy II Devices
- Chapter 2, Description, Architecture & Features
- Chapter 3, Boundary-Scan Support
- Chapter 4, Operating Conditions

Revision History

The table below shows the revision history for Chapters 1 through 4.

Chapter(s)	Date / Version	Changes Made
Chapter 1	January 2005 v1.0	Added document to the HardCopy Series Handbook.
Chapter 2	January 2005 v1.0	Added document to the HardCopy Series Handbook.
Chapter 3	January 2005 v1.0	Added document to the HardCopy Series Handbook.
Chapter 4	January 2005 v1.0	Added document to the HardCopy Series Handbook.

1. Introduction to HardCopy II Devices

H51015-1.0

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.

The 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, over 50% lower power consumption (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 18 to 22 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	609	768
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	6,345,216	8,847,360
Enhanced PLLs	2	2	2	4	4
Fast PLLs	2	2	2	4	8
Maximum user I/O pins *(4)*	*(5)*	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 power 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) The I/O pin counts include the dedicated `CLK` input pins, which can be used for clock signals or data inputs.

(5) Contact Altera® Applications for more information.

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, see 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 8,847,360 RAM bits, including parity bits, available

- 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, which allows clock networks to power down, 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 232 channels with DPA circuitry (preliminary) 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℠) megafunctions
- HardCopy II devices are available in wire-bond and flip-chip space-saving FineLine BGA packages (see 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	(5)					
HC210		334				
HC220			492			
				492		
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.
(5) Contact Altera Applications for more information.

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, see the *Stratix II FPGA Handbook*.

2. Description, Architecture & Features

H51016-1.0

Introduction

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.

Functional Description

The Hardcopy II device family provides greater flexibility, compared to previous HardCopy device families, 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 maximizing 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, see 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, see the *Prototyping Strategy for HardCopy II Using Stratix II FPGA* chapter in the *HardCopy Series Handbook*.

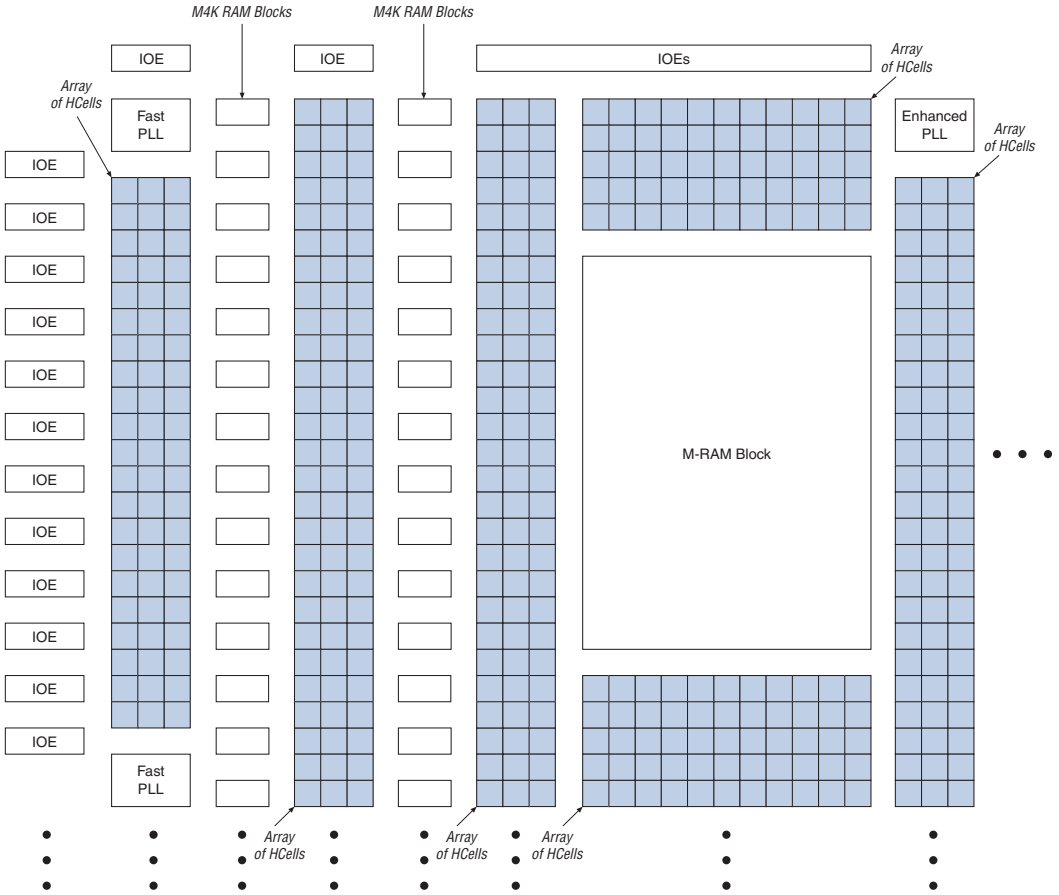
HardCopy II devices consist of base arrays that are common to all designs for a particular device density, with design-specific customization done using two metal layers. The reprogrammable FPGA logic, routing, memory, and FPGA configuration-related logic are stripped from HardCopy II devices. Removing 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. See Figure 2–1.

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



Note to Figure 2–1:

(1) This figure shows a graphical representation of the device floor plan. A detailed floor plan will be included in a future version of 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 2–1 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 2–1. 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 2–1:
(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. 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. See 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.

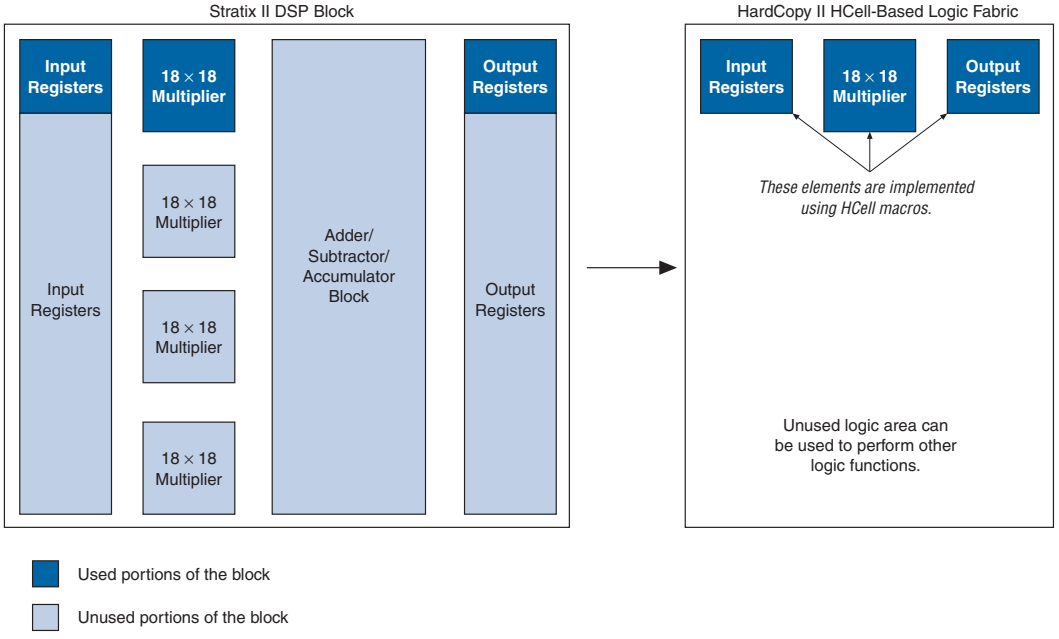
The Quartus II software uses the library of pre-characterized HCell macros to synthesize Stratix II ALM and DSP configurations into the HardCopy II HCell-based logic fabric. An HCell macro defines how a group of HCells are connected together within the array. HCell macros can construct all combinations 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 consists 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. HCell macros are stitched together to represent register and combinational logic, adder and multiplier functions for any Stratix II DSP block configuration. Only HCells that are required to implement the DSP functions of the user's design are enabled. HCells not needed for DSP functions can be used for ALM configurations, which results in efficient logic usage.

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 (see 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 DSP blocks are put together using entries from the library of HCell macros. In addition to area management, the placement of these HCell macros allows for optimized routing and performance.

HardCopy II devices use HCell macros to support the same four modes of operation of the Stratix II DSP blocks:

- Simple multiplier
- Multiply-accumulator
- Two-multipliers adder
- Four-multipliers adder

HardCopy II devices also support all Stratix II DSP configurations (9- × 9-, 18- × 18-, and 36- × 36-bit multipliers). HardCopy II devices support 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, which requires configuration logic. 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 memory blocks available differs based on density (see Table 2–2).

Table 2–2. Hardcopy II Embedded Memory Resources

Feature	HC210W	HC210	HC220	HC230	HC240
M4K RAM blocks (4k bits)	190	190	408	609	768
M-RAM blocks (512k bits)	0	0	2	6	9
Total RAM bits (bits)	875,520	875,520	3,059,712	6,345,216	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 2–3 shows the size and features of the different RAM blocks. For more information on the Stratix II memory block features, see the *Stratix II Device Handbook*.

Table 2–3. 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 2–3:
(1) Maximum performance information is preliminary until device characterization.

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 can be used for general-purpose clock management, supporting multiplication, division, phase shifting, and programmable duty cycle. In addition, 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 (see Table 2–4).

Table 2–4. 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 (see Table 2–6 on page 2–13), a fast PLL could be used in place of an enhanced PLL. The type of PLL used in the design should be chosen to accommodate the resources available in the HardCopy II device.

Table 2–5 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

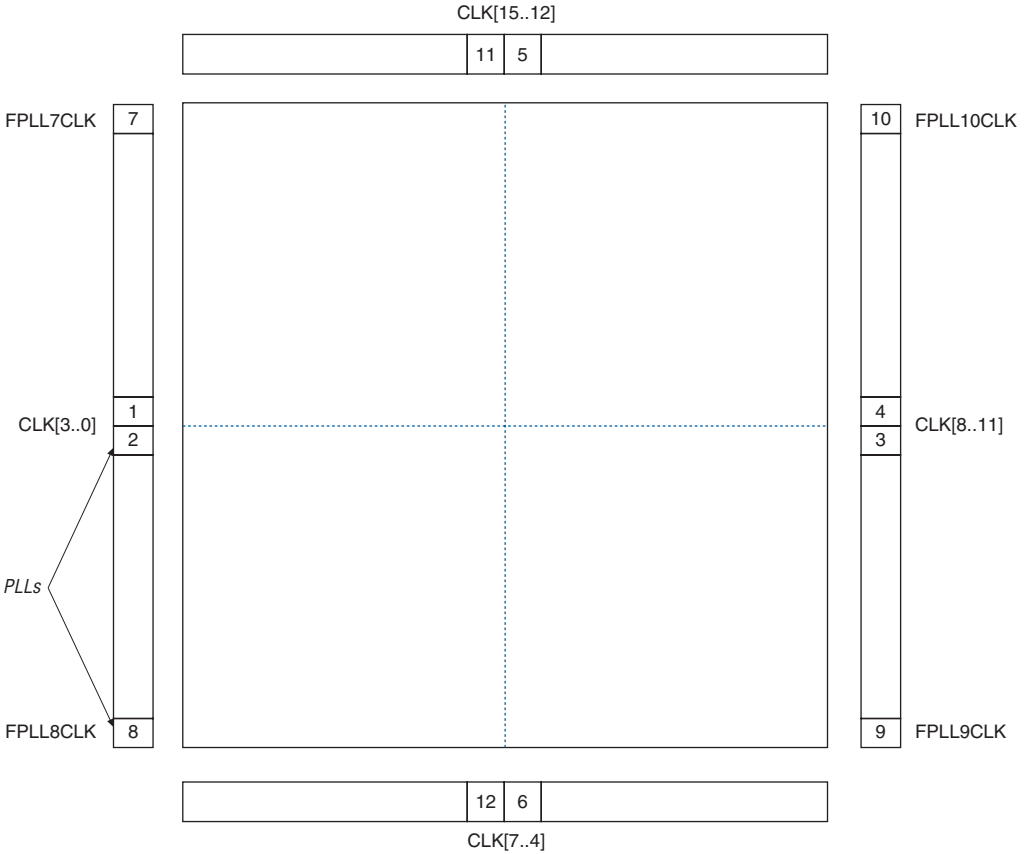
and fast PLLs that will be used in your HardCopy II device. Use Table 2–5 to ensure that the FPGA prototyping design uses the same PLL resources available in the HardCopy II device.

Table 2–5. 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	✓	✓							✓	✓		
HC220	✓	✓							✓	✓		
HC230	✓	✓			✓	✓			✓	✓	✓	✓
HC240	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Note to Table 2–5:
(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 2–4:
(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 2–6 shows the features of the different PLLs. For more information on the Stratix II PLL features, see the *Stratix II Device Handbook*.

Table 2–6. 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 2–6:

(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 dedicated 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.

The clock networks in HardCopy II devices are prefabricated. 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 2–7 lists the clock resources and features available in HardCopy II devices.

Table 2–7. 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 and LVPECL clock input support.

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 2–8 shows which I/O standards are supported by the different IOE types.

Table 2–8. Hardcopy II Supported I/O Standards (Part 1 of 2)

I/O Standard	Type	V_{CCIO} Level (V)		Memory 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	✓		
1.8-V HSTL class II	Voltage referenced	1.8	1.8	✓		

Table 2–8. Hardcopy II Supported I/O Standards (Part 2 of 2)

I/O Standard	Type	V_{CCIO} Level (V)		Memory IOEs	General Purpose IOEs	High-Speed IOEs
		Input	Output			
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	Differential	3.3/2.5/ 1.8/1.5		*(2)*	*(3)*	
Differential SSTL-2 class I and II output	Pseudo differential *(1)*		2.5	*(2)*	*(3)*	
Differential SSTL-18 class I and II input	Differential	3.3/2.5/ 1.8/1.5		*(2)*	*(3)*	
Differential SSTL-18 class I and II output	Pseudo differential *(1)*		1.8	*(2)*	*(3)*	
1.8-V differential HSTL class I and II input	Differential	3.3/2.5/ 1.8/1.5		*(2)*	*(3)*	
1.8-V differential HSTL class I and II output	Pseudo Differential *(1)*		1.8-V	*(2)*	*(3)*	
1.5-V differential HSTL class I and II input	Differential	3.3/2.5/ 1.8/1.5		*(2)*	*(3)*	
1.5-V differential HSTL class I and II output	Pseudo Differential *(1)*		1.5-V	*(2)*	*(3)*	
LVDS	Differential	2.5	2.5V	*(2)*	*(3)*, *(4)*	✓
HyperTransport™ technology	Differential	2.5	2.5V	*(2)*	*(3)*, *(4)*	✓
LVPECL	Differential	3.3/2.5/ 1.8/1.5	Not supported	*(5)*	*(6)*	

Notes to Table 2–8:

(1) Pseudo-differential HSTL and SSTL outputs use two single-ended outputs with the second output programmed as inverted.
(2) This I/O standard is only supported on the DQS, CLK and PLL_FB input pins or on the PLL_OUT output pins.
(3) This I/O standard is only supported on the bottom CLK and PLL_FB input pins or on the bottom PLL_OUT output pins.
(4) Also supported on CLK9 and CLK11 pins.
(5) This I/O standard is only supported on CLK and PLL_FB input pins.
(6) This I/O standard is only supported on the bottom CLK and PLL_FB input pins.

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)*

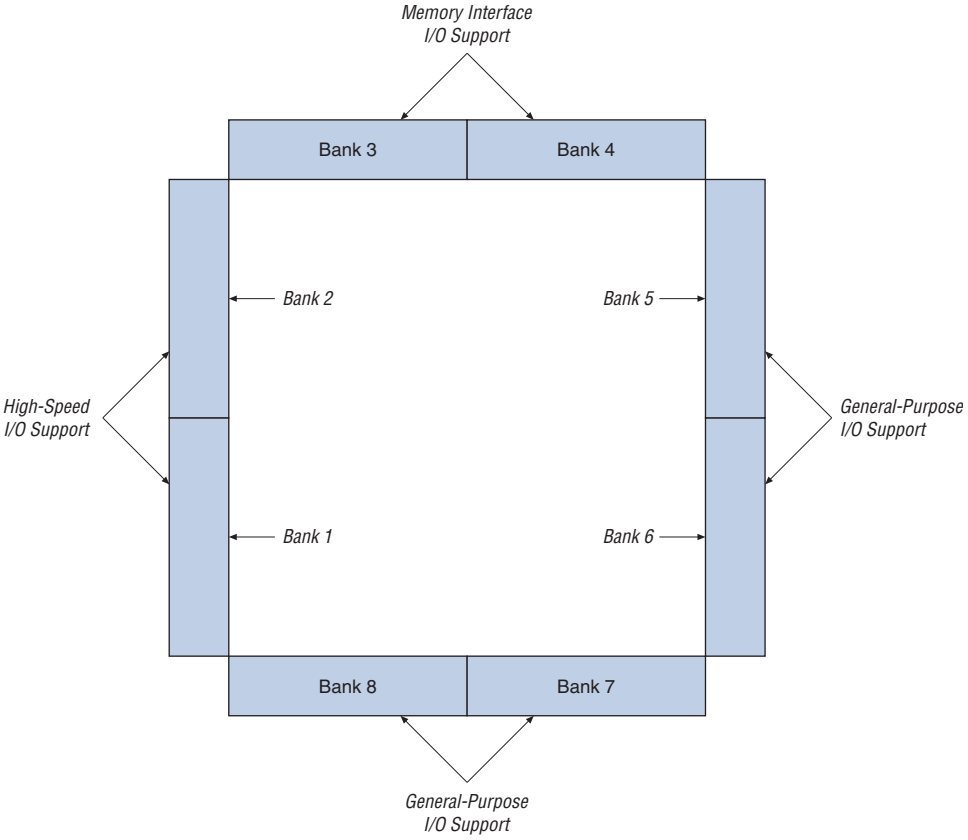


Figure 2–5. I/O Type Support in HC230 Devices *Notes (1), (2)*

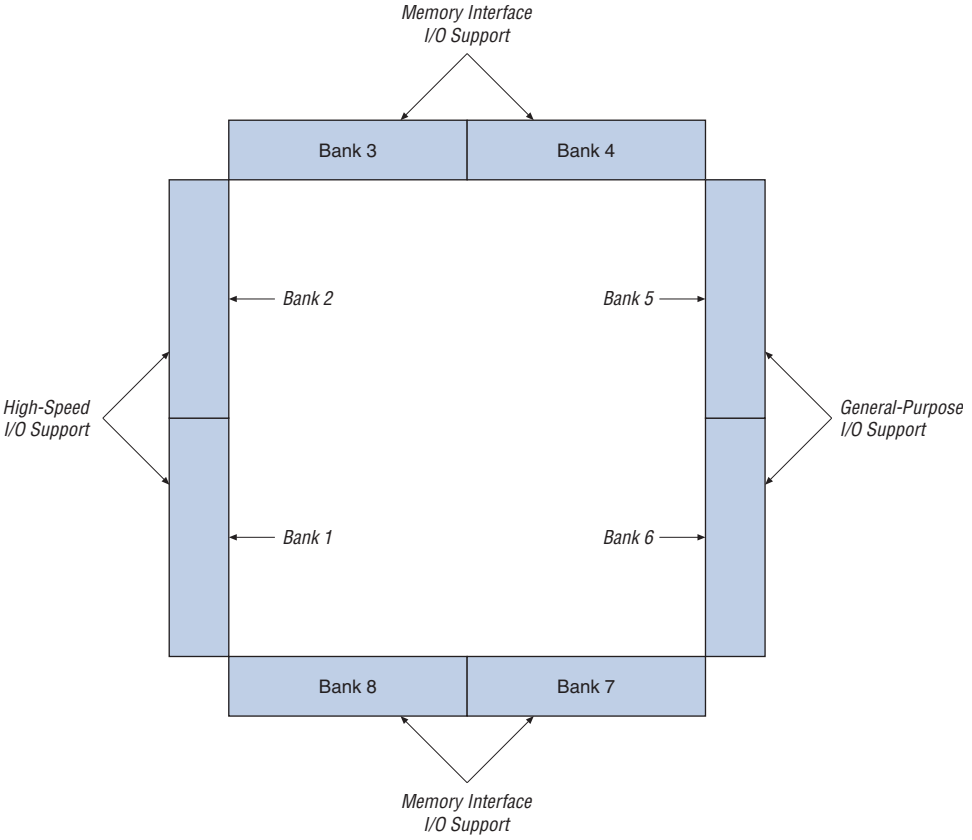
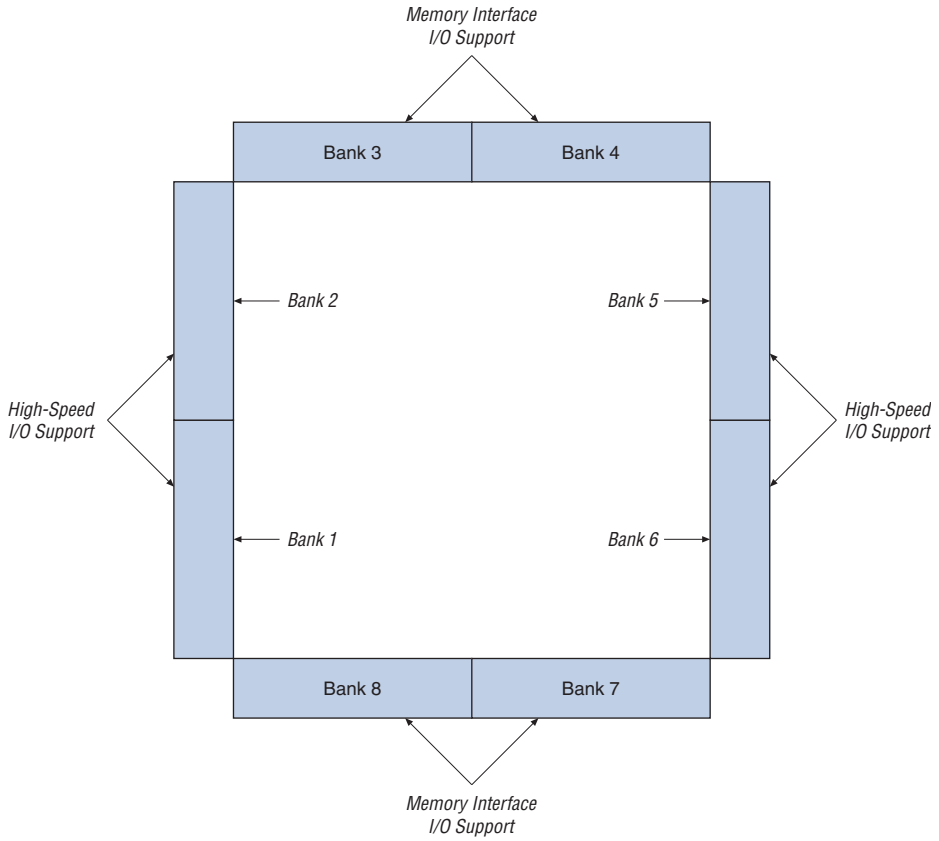


Figure 2–6. I/O Type Support in HC240 Devices *Notes (1), (2)*



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
- 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)

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:

- 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
- HyperTransport technology
- LVPECL (on input clocks only)

The programmable drive strengths available vary depending on the I/O standard being used and are listed in Table 2–9.

Table 2–9. 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-V I/O standards.

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
- HyperTransport technology
- LVPECL (on input clocks only)

Pseudo-differential HSTL and SSTL inputs are supported on dedicated CLK 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 2–10 shows the number of DQS/DQ groups supported in each HardCopy II device density and package.

Table 2–10. DQS & DQ Bus Mode Support (Part 1 of 2)

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

Table 2–10. DQS & DQ Bus Mode Support (Part 2 of 2)

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

The programmable drive strengths available vary depending on the I/O standard used. The options are listed in Table 2–11.

Table 2–11. 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	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.

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
- 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

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 circuitry and functionality is the same in Hardcopy II devices as in Stratix II FPGAs. Table 2–12 provides the number of differential channels per HardCopy II device.

Table 2–12. 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	*(3)*	19	29	29	44	88	116
Receiver channels	*(3)*	21	31	31	46	92	116

Notes to Table 2–12:

(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.

(3) Contact Altera Applications for more details.

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 2–13.

Table 2–13. 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-V I/O standards.

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 configuration circuitry is not needed in HardCopy II structured ASICs. 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 or configuration input signals.

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 `CONF_DONE` output will 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.

3. Boundary-Scan Support

H51017-1.0

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All HardCopy® II devices 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 pin and all JTAG input pins are powered by the 3.3-V V_{CCPD} pin. 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

Device	Boundary-Scan Register Length
HC210W	*(1)*
HC210	*(1)*
HC220	*(1)*
HC230	*(1)*
HC240	*(1)*

Note to Table 3–2:
(1) 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 1100 0001	000 0110 1110	1
HC210	0000	0010 0000 1100 0010	000 0110 1110	1
HC220	0000	0010 0000 1100 0011	000 0110 1110	1
HC230	0000	0010 0000 1100 0100	000 0110 1110	1
HC240	0000	0010 0000 1100 0101	000 0110 1110	1

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

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

Figure 3–1. HardCopy II JTAG Waveforms

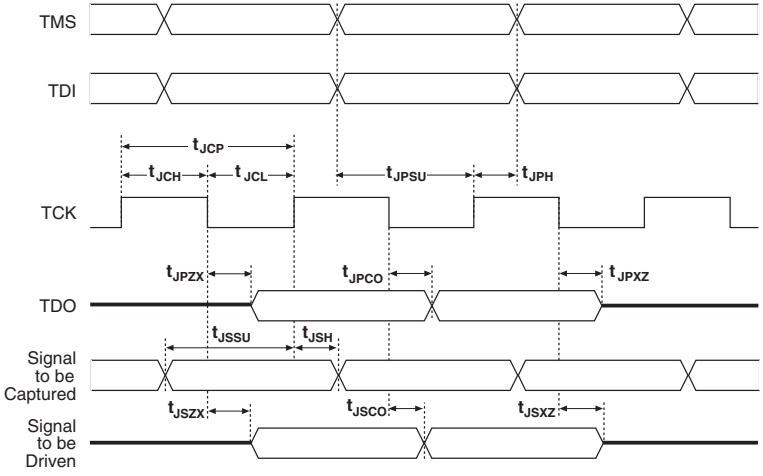


Table 3–4 shows the JTAG timing parameters and values for HardCopy II devices.

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, see *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 is not needed once the design has been migrated to a HardCopy II device.

4. Operating Conditions

H51018-1.0

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) See 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	
V_{IN} (V)	Maximum Duty Cycles
4.0	100%
4.1	90%
4.2	50%
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
V_O	Output voltage		0	V_{CCIO}	V

Table 4–3. HardCopy II Device Recommended Operating Conditions (Part 2 of 2) Note (1)					
Symbol	**Parameter**	**Conditions**	**Minimum**	**Maximum**	**Unit**
T_J	Operating junction temperature	For commercial use	0	85	°C
		For industrial use	–40	100	°C
t_R	Input rise time			40	ns
t_F	Input fall time			40	ns

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 (Part 1 of 2) 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

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

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
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 to –24 mA *(2)*, *(3)*	2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 4 to 24 mA *(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 2–9, 2–11, and 2–13.

(3) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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	V_{CCIO} – 0.2		V
V_{OL}	Low-level output voltage	V_{CCIO} = 3.0, I_{OL} = 0.1 mA		0.2	V

Note 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.

Table 4–7. 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	4.0	V
V_{IL}	Low-level input voltage		–0.3	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)*, *(2)*	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)*, *(2)*		0.7	V

Notes to Table 4–7:
(1) Drive strength is programmable according to values in Tables 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter for more information.

Table 4–8. 1.8-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	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 *(1)*, *(2)*	V_{CCIO} – 0.45		V
V_{OL}	Low-level output voltage	I_{OL} = 2 to 8 mA *(1)*, *(2)*		0.45	V

Notes to Table 4–8:
(1) Drive strength is programmable according to values in Tables 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter for more information.

Table 4–9. 1.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	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 *(1)*, *(2)*	0.75 × V_{CCIO}		V
V_{OL}	Low-level output voltage	I_{OL} = 2 mA *(1)*, *(2)*		0.25 × V_{CCIO}	V

Notes to Table 4–9:
(1) Drive strength is programmable according to values in Tables 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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

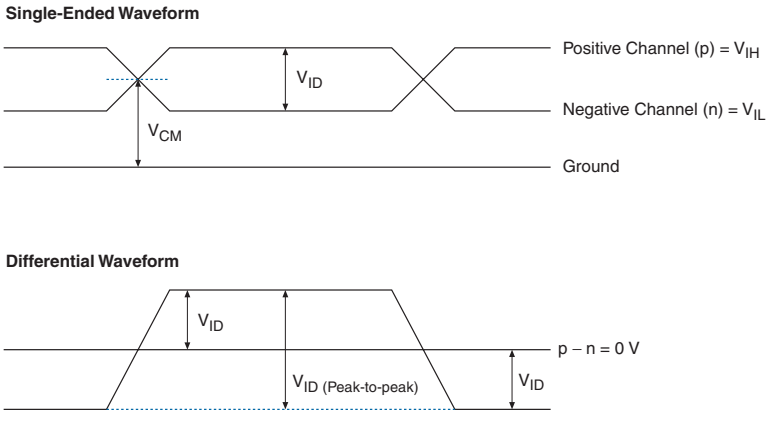


Figure 4–2. Transmitter Output Waveforms for Differential I/O Standards

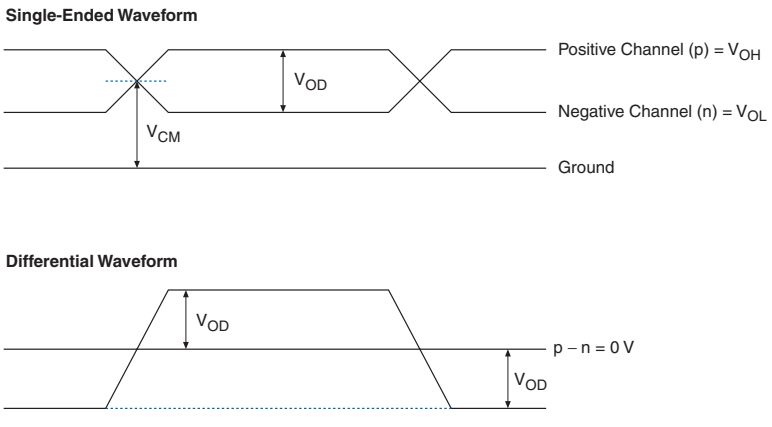


Table 4–10. 2.5-V LVDS I/O Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage for I/O banks that support high-speed IOEs		2.375	2.5	2.625	V
	I/O supply voltage for clock pins in top and bottom I/O banks (3, 4, 7, and 8)		3.135	3.3	3.465	V

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

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{ID} (peak-to-peak)	Input differential voltage swing (single-ended)		*(1)*	*(1)*	*(1)*	mV
V_{ICM}	Input common mode voltage		*(1)*	*(1)*	*(1)*	V
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	247	350	600	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	1.25	1.375	V
Δ V_{OCM}	Change in V_{OCM} between high and low	R_L = 100 Ω			50	mV
R_L	Receiver differential input discrete resistor (external to HardCopy II devices)		90	100	110	Ω

Note to Table 4–10:
(1) This specification is pending device characterization.

Table 4–11. LVPECL Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage		3.135	3.3	3.465	V
V_{ID} (peak-to-peak)	Input differential voltage swing (single-ended)		100	600	950	mV
V_{ICM}	Input common mode voltage		*(1)*	*(1)*	*(1)*	V
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	525	800	970	mV
V_{OCM}	Output common mode voltage	R_L = 100 Ω	*(1)*	*(1)*	*(1)*	V
R_L	Receiver differential input resistor		90	100	110	Ω

Note to Table 4–11:
(1) This specification is pending device characterization.

Table 4–12. HyperTransport Technology Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	I/O supply voltage for I/O banks that support high-speed IOEs		2.375	2.5	2.625	V
	I/O supply voltage for clock pins in top and bottom I/O banks (3, 4, 7, and 8)		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
V_{OD}	Output differential voltage (single-ended)	R_L = 100 Ω	400	600	820	mV
ΔV_{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
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 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 2–9, 2–11, and 2–13.

(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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 2–9, 2–11, and 2–13.

(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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 2–9, 2–11, and 2–13.

(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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 2–9, 2–11, and 2–13.
(2) Drive strength varies based on pin location. See the *Description, Architecture & Features* chapter 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.

Pin Capacitance

Table 4–28 shows the HardCopy II device family pin capacitance.

Table 4–28. HardCopy II Device Capacitance

Symbol	Parameter	Minimum	Typical	Maximum	Unit
C_{IOTB}	Input capacitance on I/O pins in I/O banks 3, 4, 7, and 8.		5.0	*(1)*	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.		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
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 4–28:
(1) This specification is pending device characterization.

ESD Protection Specifications

Table 4–29 shows the electro-static discharge (ESD) protection specifications for the HardCopy II device family.

Table 4–29. ESD Protection

Symbol	Parameter	Maximum	Unit
ESDHBM	Human body model	1,000	V
ESDCDM	Charged device model	500	V
	Charged device model for phase-locked loop (PLL) power pins and dedicated clocks 1, 3, 9, and 11	300	V

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 chapters:

Revision History

The table below shows the revision history for Chapters 5 through 8.

Chapter(s)	Date / Version	Changes Made
Chapter 5	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 6	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 7	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 8	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.*

5. Introduction to HardCopy Stratix Devices

H51001-2.0

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 5–1 provides the details of the HardCopy Stratix devices.

Table 5–1. 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 5–1:
(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 (see Tables 5–2 and 5–3)
- 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 5–2. 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 5–2:

(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 5–3. 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

6. Description, Architecture & Features

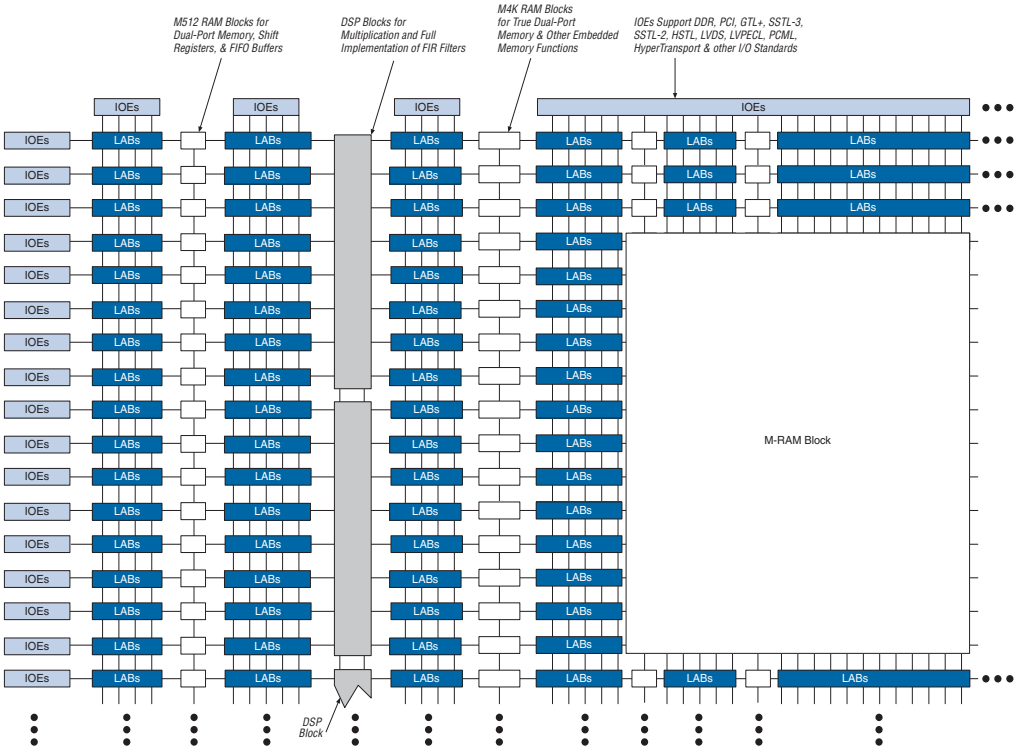
H51002-2.0

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 6–1, 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 6–1. 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 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

There are several differences between HardCopy Stratix and Stratix FPGAs.

- Unlike Stratix FPGAs, HardCopy Stratix devices are customized at the time of manufacturing with no programmability support. 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.
- 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 6–4 in this document 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.
- Quartus II software, beginning with version 4.1, contains the latest timing models. For designs with tight timing constraints, Altera strongly recommends compiling the design with 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.
- The HardCopy Stratix input/output elements (IOEs) are equivalent, but not identical, to the Stratix FPGA IOEs. This is due to the reduced die size, layout differences, 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 the HardCopy Stratix IOEs.
- When designing with very tight timing (such as double data rate or quad data rate), or if using the programmable drive strength option, Altera recommends verifying final drive strength using updated IBIS models located at the Altera web site (**www.altera.com**). Differential I/O standards are unaffected.
- The JTAG boundary scan order in the HardCopy Stratix device is different than that of the Stratix device. Use a HardCopy Stratix BSDL file that describes the re-ordered boundary scan chain.

- In the Stratix EP1S40F780 FPGA, 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 HC1S40F780 device, the I/O pins U12 and U18 must be connected to ground. The HC1S40F780 and EP1S40F780_HARDCOPY_FPGA_PROTOTYPE pin-outs are identical.

Designers can use the Quartus II software to design HardCopy Stratix devices, estimate performance and power consumption, and maximize system throughput.

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 power-on-reset circuit will reset all registers. The CONF_DONE output will be tri-stated once the power-on reset (POR) has elapsed. No configuration device or configuration data are 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 will be an additional delay of 50 ms, 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 device or configuration data are necessary for this option.
- In configuration emulation mode, after power-up, the HardCopy Stratix device undergoes an emulation of the full configuration sequence as if configured by an external processor, or an EPC device. In this mode, the device expects to see all configuration control and data input signals and the CONF_DONE signal is tri-stated once the emulation is completed.

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 `nCONFIG`, `nSTATUS`, and `CONF_DONE` should be left on the printed circuit board.

For more information, see 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.

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.

Some HARDCOPY_FPGA_PROTOTYPE devices, as indicated in Table 6–1, 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 6–1. 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	EP1830	6	HC1830	6	EP1830	9

For more information about how the various features in the Quartus II software can be used for designing HardCopy Stratix devices, see the *Quartus II Support for HardCopy Stratix Devices* chapter of the *HardCopy Series Handbook*.

Being FPGAs, HARDCOPY_FPGA_PROTOTYPE 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.

7. Boundary-Scan Support

H51004-2.0

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. HardCopy Stratix devices support the JTAG instructions shown in Table 7–1.

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**.

Table 7–1. HardCopy Stratix JTAG Instructions

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.
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 7–1:
(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

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 7–2 and 7–3 show the boundary-scan register length and device IDCODE information for HardCopy Stratix devices.

Table 7–2. HardCopy Stratix Boundary-Scan Register Length

Device	Boundary-Scan Register Length
HC1S25	2,157
HC1S30	2,253
HC1S40	2,529
HC1S60	3,129
HC1S80	3.777

Table 7–3. 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 7–3:
(1) The most significant bit (MSB) is on the left.
(2) The IDCODE's least significant bit (LSB) is always 1.

Figure 7–1 shows the timing requirements for the JTAG signals.

Figure 7–1. HardCopy Stratix JTAG Waveforms

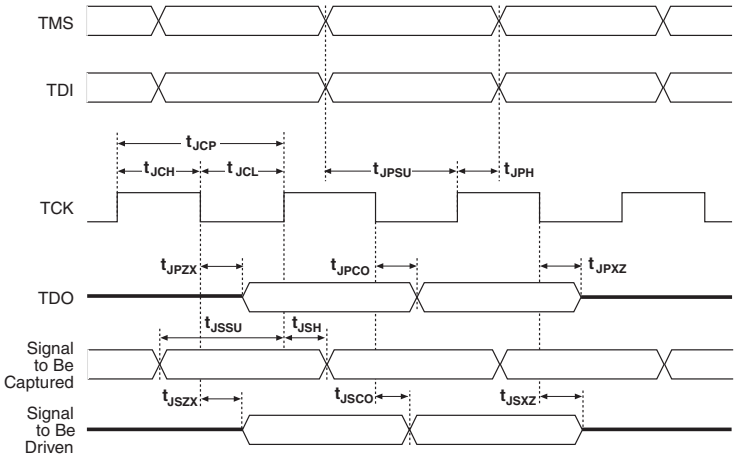


Table 7–4 shows the JTAG timing parameters and values for HardCopy Stratix devices.

Table 7–4. 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, see *AN 39: IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*.

8. Operating Conditions

H51005-2.0

Recommended Operating Conditions

Tables 8–1 through 8–3 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 1.5-V HardCopy® Stratix® devices.

Table 8–1. 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 8–2. 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 8–3. 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 8–1 through 8–3:

(1) See the *Operating Requirements for Altera Devices Data Sheet*.
(2) Conditions beyond those listed in Table 8–1 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 8–4 through 8–31 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 8–32 provides information on capacitance for 1.5-V HardCopy Stratix devices.

Table 8–4. 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 8–5. 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 8–6. 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 8–7. 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 8–8. 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 8–9. 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 8–9. 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 8–10. 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 8–11. 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 8–12. 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 8–13. 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 8–14. 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 8–15. 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 8–16. 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 8–17. 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 8–18. 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 8–19. 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}	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)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(1)*			V_{TT} – 0.57	V

Table 8–20. SSTL-2 Class II Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		2.3	2.5	2.7	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage		1.15	1.25	1.35	V
V_{IH}	High-level input voltage		V_{REF} + 0.18		V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.18	V
V_{OH}	High-level output voltage	I_{OH} = –16.4 mA *(1)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(1)*			V_{TT} – 0.76	V

Table 8–21. SSTL-3 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.0	3.3	3.6	V
V_{TT}	Termination voltage		V_{REF} – 0.05	V_{REF}	V_{REF} + 0.05	V
V_{REF}	Reference voltage		1.3	1.5	1.7	V
V_{IH}	High-level input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
V_{IL}	Low-level input voltage		–0.3		V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = –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 8–22. 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}	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 *(1)*			V_{TT} – 0.8	V

Table 8–23. 3.3-V AGP 2× Specifications

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
V_{IL}	Low-level input voltage *(4)*				0.3 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OUT} = –0.5 mA	0.9 × V_{CCIO}		3.6	V
V_{OL}	Low-level output voltage	I_{OUT} = 1.5 mA			0.1 × V_{CCIO}	V

Table 8–24. 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 × V_{CCIO}		V_{CCIO} + 0.5	V
V_{IL}	Low-level input voltage *(4)*				0.3 × V_{CCIO}	V
V_{OH}	High-level output voltage	I_{OUT} = –0.5 mA	0.9 × V_{CCIO}		3.6	V
V_{OL}	Low-level output voltage	I_{OUT} = 1.5 mA			0.1 × V_{CCIO}	V

Table 8–25. 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 8–26. 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 8–27. 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 8–28. 1.8-V HSTL Class II 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} = 16 mA *(1)*	$V_{CCIO} – 0.4$			V
V_{OL}	Low-level output voltage	I_{OH} = –16 mA *(1)*			0.4	V

Table 8–29. 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 8–30. CTT I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.0	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 ≤ V_{OUT} ≤ V_{CCIO}	–10		10	µA

Table 8–31. 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)			30		50		70		µA
High sustaining current	V_{IN} < V_{IH} (minimum)			–30		–50		–70		µA
Low overdrive current	0 V < V_{IN} < V_{CCIO}				200		300		500	µA
High overdrive current	0 V < V_{IN} < V_{CCIO}				–200		–300		–500	µA

Table 8–32. 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 8–4 through 8–32:

(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

Preliminary timing information for HardCopy Stratix devices is available in the Quartus II software. This timing information will provide an estimation of the device performance. The actual performance is available when the design migration is complete.

For more information, see the *HardCopy Series Back-End Timing Closure* chapter in the *HardCopy Series Handbook*.

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:

Revision History

The table below shows the revision history for Chapters 9 through 12.

Chapter(s)	Date / Version	Changes Made
Chapter 9	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 10	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 11	January 2005 v2.0	Update device names and other minor textual changes.
	June 2003 v1.0	Initial release of Chapter 11, Boundary-Scan Support, in the HardCopy Device Handbook.
Chapter 12	January 2005 v2.0	Update device names and other minor textual changes.
	June 2003 v1.0	Initial release of Chapter 12, Operating Conditions, in the HardCopy Device Handbook

9. 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 (see Table 9–1)
- Up to 51,840 logic elements (LEs)
- Up to 442,368 RAM bits that can be used without reducing available logic

Table 9–1. 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 9–1:
(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 (see Table 9–2)
- 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 9–2. 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 9–2:
(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 9–3 through 9–6 show the HardCopy APEX device ball-grid array (BGA) and FineLine BGA® package options, I/O counts, and sizes.

Table 9–3. HardCopy APEX Device BGA Package Options & I/O Count Note (1)

Device	652-Pin BGA
HC20K400	488
HC20K600	488
HC20K1000	488
HC20K1500	488

Table 9–4. 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 9–3 and 9–4:
(1) I/O counts include dedicated input and clock pins.

Table 9–5. 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 9–6. 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

10. 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 10–1 shows the device equivalence for HardCopy and APEX 20KE or APEX 20KC devices.

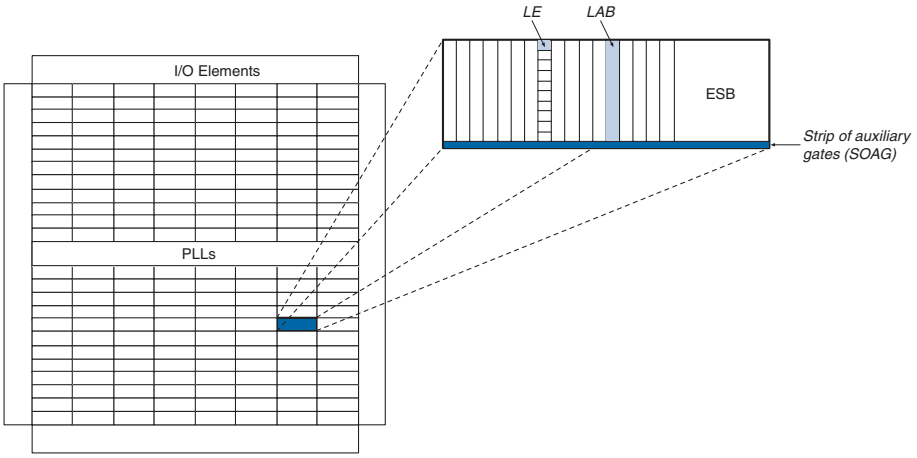
Table 10–1. HardCopy & APEX 20KE or APEX 20C Device Equivalence

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 10–1 shows a diagram of the HardCopy APEX device architecture.

Figure 10–1. 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 10–2 lists the features included in HardCopy APEX devices.

Table 10–2. 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 10–2. 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.

11. 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 11–1.

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 11–1. 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 11–2 and 11–3 show the boundary-scan register length and device IDCODE information for HardCopy devices.

Table 11–2. HardCopy APEX Boundary-Scan Register Length

Device	Boundary-Scan Register Length
HC20K400	1,506
HC20K600	1,806
HC20K1000	2,190
HC20K1500	2,502

Table 11–3. 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 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 JTAG Waveforms

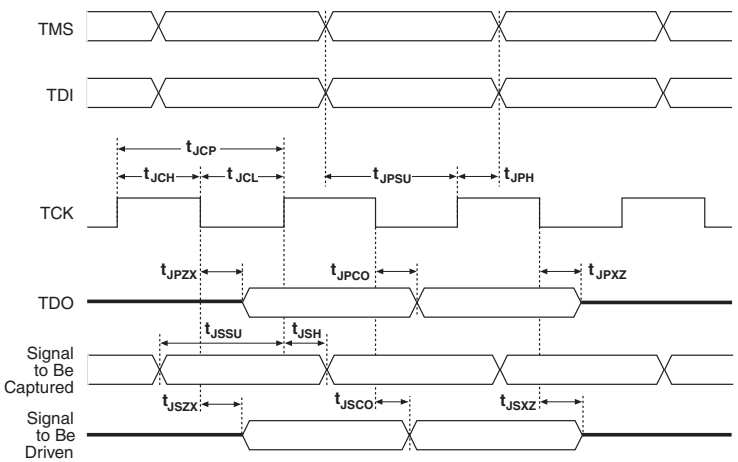


Table 11–4 shows the JTAG timing parameters and values for HardCopy devices.

Table 11–4. 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, see *Application Note 39 (IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices)*.

12. Operating Conditions

H51010-2.0

Recommended Operating Conditions

Tables 12–1 through 12–4 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 1.8-V HardCopy® APEX™ devices.

Table 12–1. 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 12–2. 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 12–3. 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
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

Table 12–3. HardCopy APEX Device DC Operating Conditions (Part 2 of 2) *Notes (6), (7), (8)*

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
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 12–4. 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 12–1 through 12–4:

(1) See 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 12–2 on page 12–1.
(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 12–5 through 12–20 list the DC operating specifications for the supported I/O standards. These tables list minimal specifications only; HardCopy devices may exceed these specifications.

Table 12–5. 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 12–6. 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 12–7. 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 12–8. 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 12–9. 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 12–9. 3.3-V PCI Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Units
V_{IL}	Low-level input voltage		–0.5		$0.3 \times V_{CCIO}$	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	$0.9 \times V_{CCIO}$			V
V_{OL}	Low-level output voltage	I_{OUT} = 1,500 µA			$0.1 \times V_{CCIO}$	V

Table 12–10. 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		$0.5 \times V_{CCIO}$		$V_{CCIO} + 0.5$	V
V_{IL}	Low-level input voltage		–0.5		$0.35 \times V_{CCIO}$	V
V_{IPU}	Input pull-up voltage		$0.7 \times V_{CCIO}$			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	$0.9 \times V_{CCIO}$			V
V_{OL}	Low-level output voltage	I_{OUT} = 1500 µA			$0.1 \times V_{CCIO}$	V
L_{PIN}	Pin Inductance				15.0	nH

Table 12–11. 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 12–11. 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 12–12. 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 12–13. 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 12–13. 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 12–14. 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 12–15. 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 12–16. 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 12–17. 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 12–18. 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 12–18. 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 12–19. 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 12–20. 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 12–20. 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 12–5 through 12–20:

(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 12–1 shows the output drive characteristics of HardCopy APEX devices.

Figure 12–1. Output Drive Characteristics of HardCopy APEX Devices

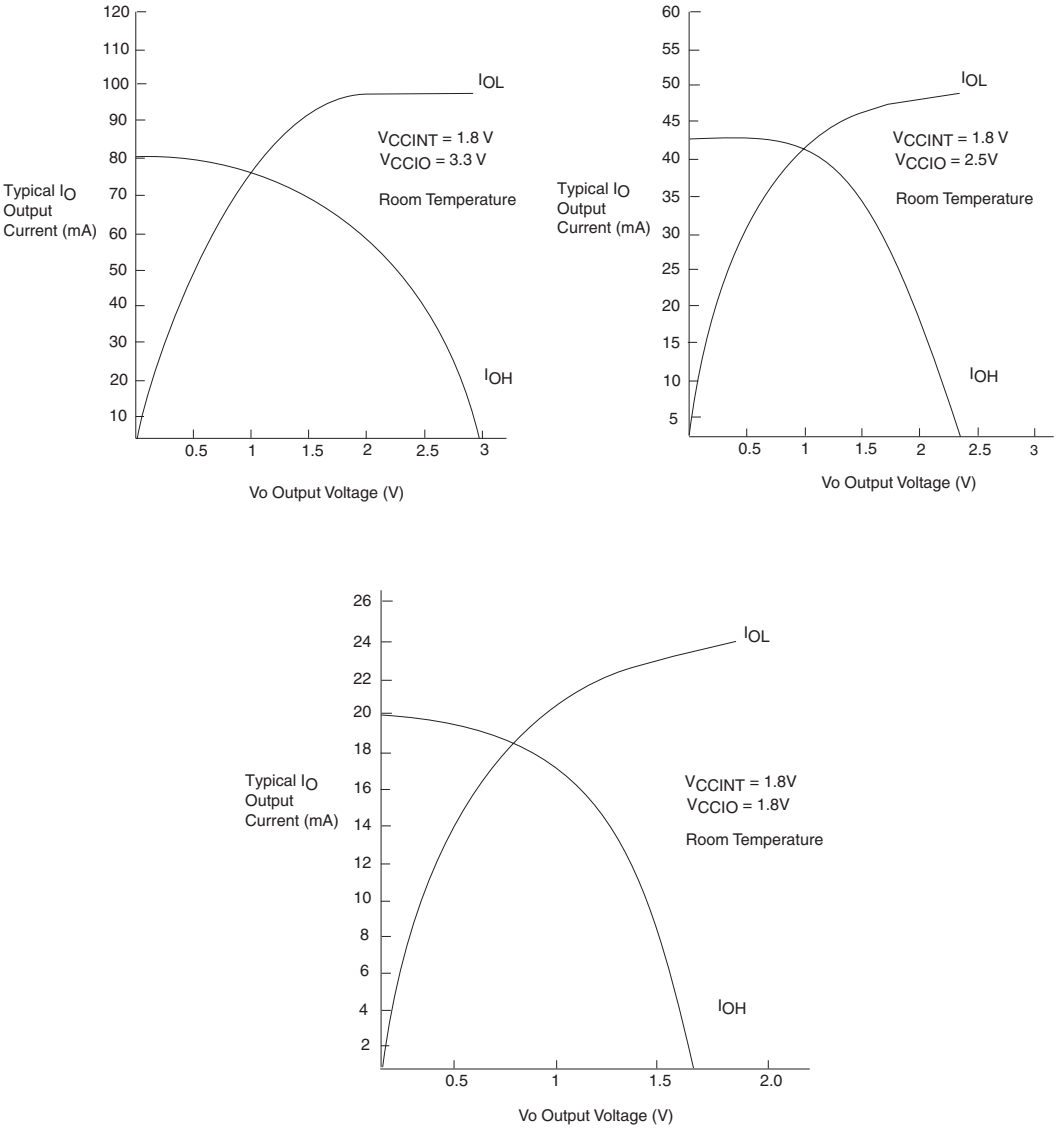
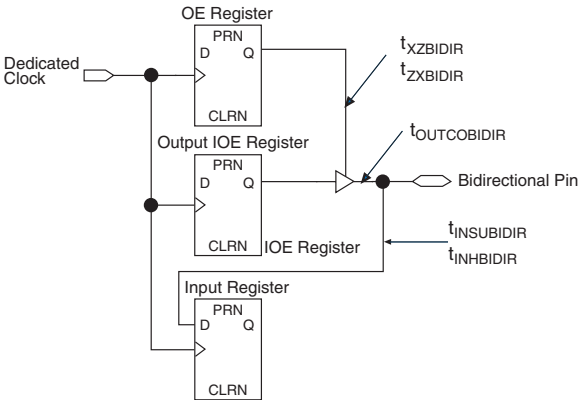


Figure 12–2 shows the timing model for bidirectional I/O pin timing.

Figure 12–2. Synchronous Bidirectional Pin External Timing



Tables 12–21 and 12–22 describe HardCopy APEX device external timing parameters.

Table 12–21. 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 12–22. 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 12–22. 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 12–21 and 12–22:
(1) These timing parameters are sample-tested only.

Tables 12–23 and 12–24 show the external timing parameters for HC20K1500 devices.

Table 12–23. 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 12–24. 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 12–24. 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 12–23 and 12–24:

(1) Timing information is preliminary. Final timing information will be available in a future version of this data sheet.

Section IV. Hardware Design Considerations

This section provides information on hardware design considerations for HardCopy® series devices.

It contains the following:

- Chapter 13, Back-End Design Flow for HardCopy Series Devices
- Chapter 14, Design Guidelines for HardCopy Series Devices
- Chapter 15, Power-Up Modes & Configuration Emulation in HardCopy Series Devices
- Chapter 16, Back-End Timing Closure for HardCopy Series Devices

Revision History

The table below shows the revision history for Chapters 14 through 16.

Chapter(s)	Date / Version	Changes Made
Chapter 13	January 2005 v1.0	Initial release of Chapter 13, Back-End Design Flow for HardCopy Series Devices.
Chapter 14	January 2005 v2.0	• Chapter title changed to *Design Guidelines for HardCopy Series Devices*. • Updated Quartus® II software supported versions on page 14–1. • Updated HardCopy® Design Center support on page 14–2. • Updated heading "Using a Double Synchronizer for Single Bit Data Transfer" on page 14–4. • Added Stratix® II support for a global or regional clock on page 14–12. • Added support for Stratix II and HardCopy II to "Mixing Clock Edges" on page 14–14.
	August 2003 v1.1	Edited hierarchy of section headings.
	June 2003 v1.0	Initial release of Chapter 14, Design Guidelines for HardCopy Series Devices.

Chapter(s)	Date / Version	Changes Made
Chapter 15	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 15–9 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 15–10 and 15–11, highlighting power up recommendations for each HardCopy series family.
	June 2003 v1.0	Initial release of Chapter 15, Power-Up Modes & Configuration Emulation in HardCopy Series Devices.

Chapter(s)	Date / Version	Changes Made
Chapter 16	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 16–2. • HardCopy® Stratix® and HardCopy APEX™ equivalence to their respective FPGA is updated on page 16–2. • Stratix II migration added on page page 16–2. • Updated Table 16–2 on page 16–12. • Updated last paragraph in "Timing ECOs" on page 16–19.
	June 2003 v1.0	Initial release of Chapter 16, Back-End Timing Closure for HardCopy Series Devices.

13. Back-End Design Flow for HardCopy Series Devices

H51019-1.0

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:

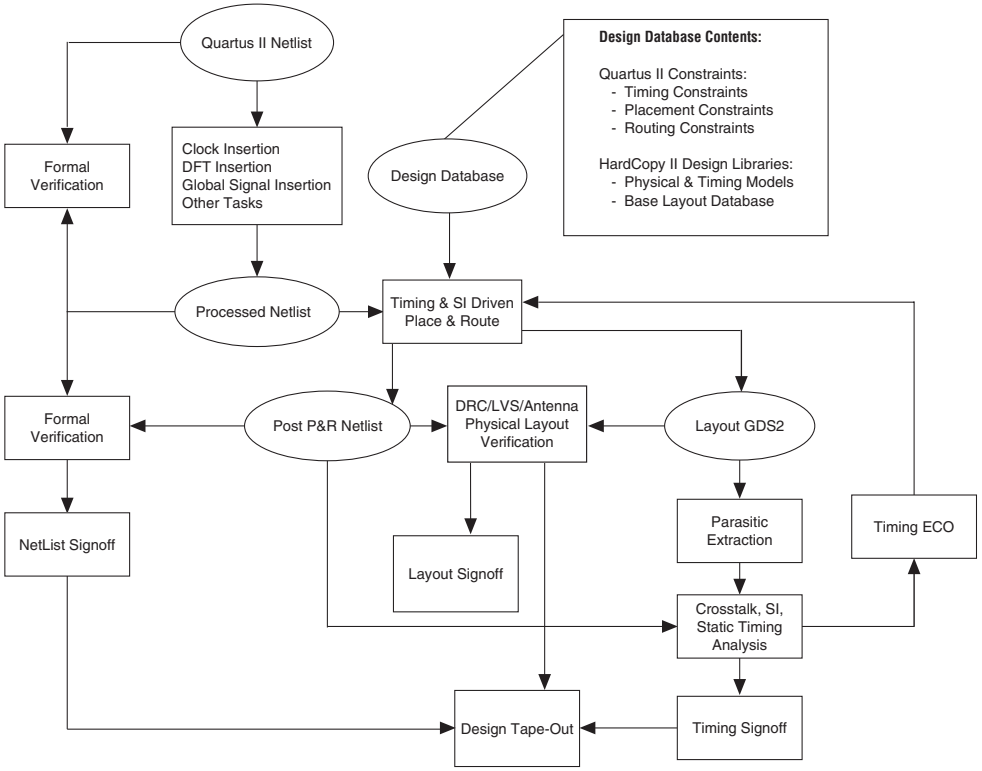
- HardCopy II Back-End Design Flow
- HardCopy Stratix® and HardCopy APEX™ Back-End Design Flow

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 13–1 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 13–1. 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 **.sof** file 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 have been added to the processed netlist, the HardCopy Design Center uses third-party EDA formal verification software to compared 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 be able to verify that your design's intended functionality has not been 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.

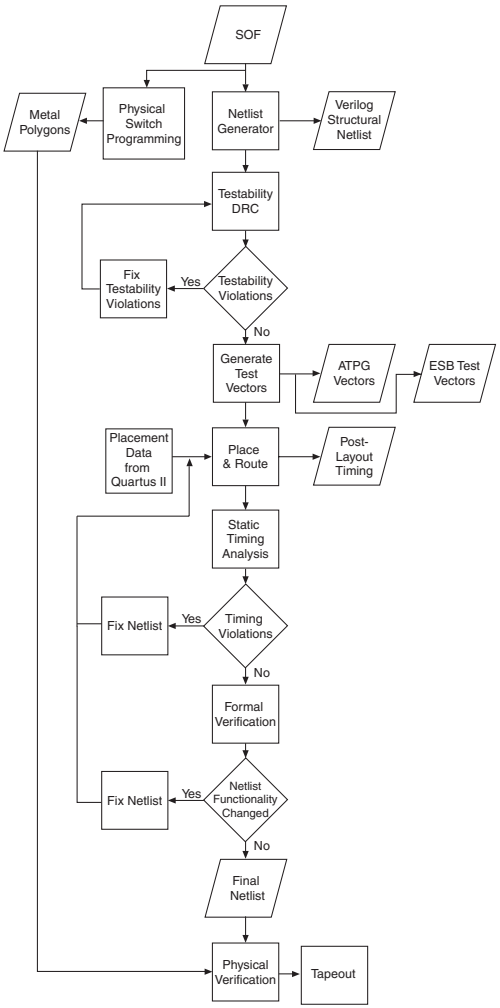
Design Signoff

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 13–2. The migration process uses both proprietary and third-party EDA tools.

Figure 13–2. 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 will contain 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 will make 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

The Quartus II software version 4.2 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, see the *Quartus II Support for HardCopy Stratix Devices* chapter. For more information on Quartus II features for HardCopy II devices, see 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 generation is 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 will operate 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, the designer 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.

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.

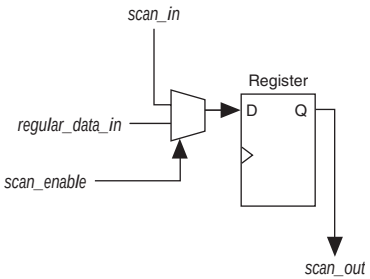
Testing

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.

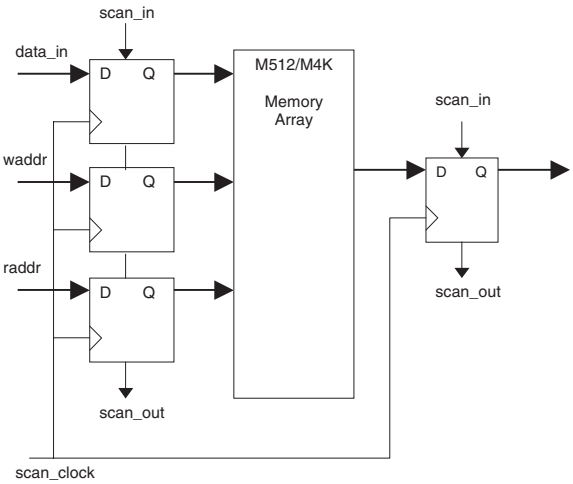
Every register in the HardCopy series device belongs to a scan chain. Scan chains are test features that exist in ASICs to ensure that there is access to all internal nodes of a design. With scan chains, defective parts can be screened out during the manufacturing process. Scan chain registers are constructed 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 13–3 shows a diagram of a scan register.

Figure 13–3. HardCopy Stratix Scan Chain Circuitry



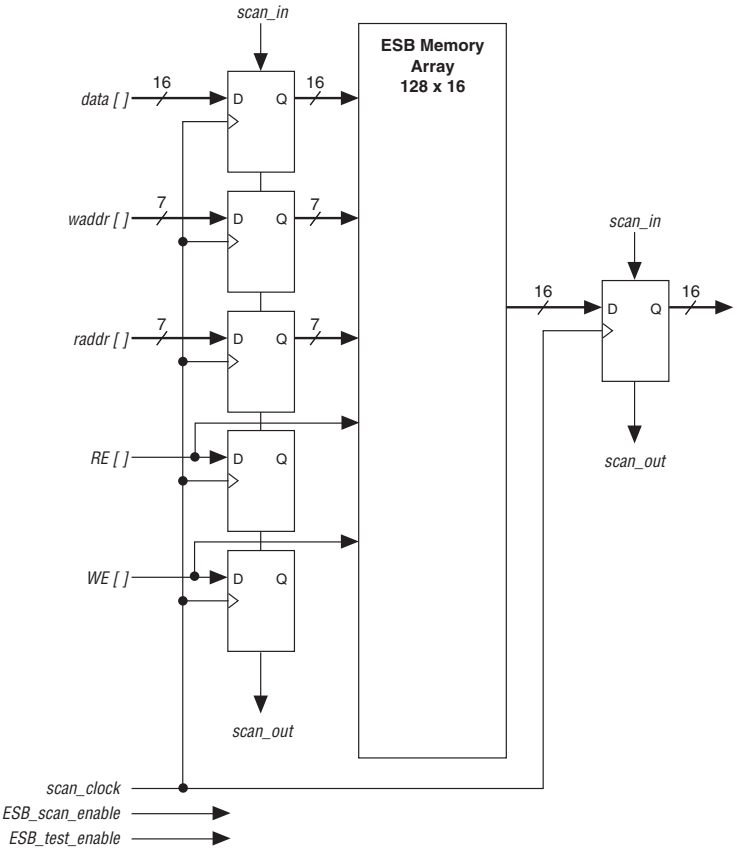
In addition to the scan circuitry (Figure 13–3), which is designed to test all LEs and IOEs, both M512 and M4K blocks (Figure 13–4) have the same scan chain structure so that all bits inside the memory array are tested for correct operation. The M512 and M4K RAM bits are tested by scanning data into 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 13–4 shows the M512 and M4K blocks' scan chain connectivity.

Figure 13–4. 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 will have 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 13–5 shows the ESB test mode configuration.

Figure 13–5. 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 will seamlessly migrate 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, see the *HardCopy Series Handbook*.

14. Design Guidelines for HardCopy Series Devices

H51011-2.0

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, designers must meet certain design rules while the FPGA implementation is still in progress. A design that meets standard, accepted coding styles for FPGAs, will be easier to adhere to recommended guidelines. This chapter describes some common situations that designers should avoid and provides alternatives on how to design in these situations.

Design Assistant Tool

The Design Assistant tool in the Quartus® II software allows designers 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. The set of rules that the tool checks can be customized to allow some rule violations. This is useful if it is known that the design violates a particular rule that is not critical. However, for HardCopy design, all Design Assistant rules must be enabled. 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 Design Assistant classifies messages using the four severity levels described in Table 14–1.

Table 14–1. 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 will not be able to 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 can be migrated to a HardCopy device.

Table 14–1. 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.

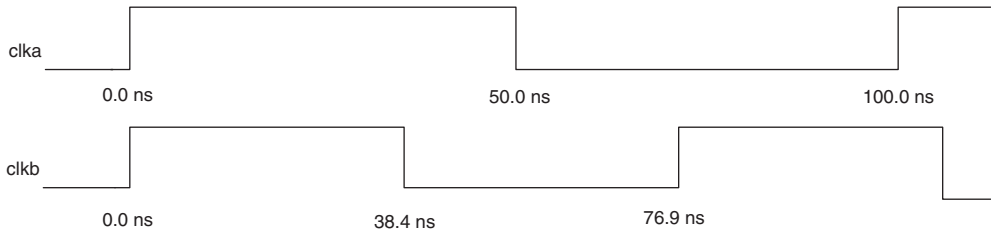
A design that adheres to Altera recommended design guidelines will not produce any critical, high, or medium level Design Assistant messages. If the Design Assistant generates these kinds of messages, Altera's HardCopy Design Center (which performs the migration) will carefully review each message before 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 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 14–1 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 14–1. Two Asynchronous Clock Signals *Notes (1), (2)*



Notes to Figure 14–1:
(1) clka = 10 MHz; clkb =13 MHz.
(2) Both clocks have 50% duty cycles.

In Figure 14–1, 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 will be 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 will be 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), will 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 the designer's original intention is 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, will have some frequency variation due to temperature fluctuations, and this may be different for each oscillator. This will result 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, some consideration needs to be given to how this operation can be performed reliably. Three examples of how to transfer data between two asynchronous clock domains are provided in this section, as follows.

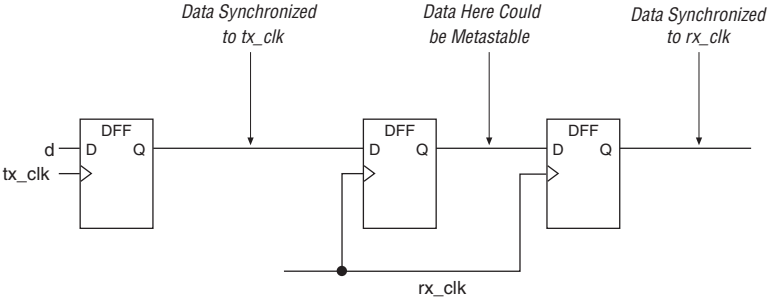
- Using a double synchronizer
- Using a first-in first-out (FIFO) buffer
- Using a handshake protocol

The choice of which one 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 14–2 shows a double synchronizer for simple 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.

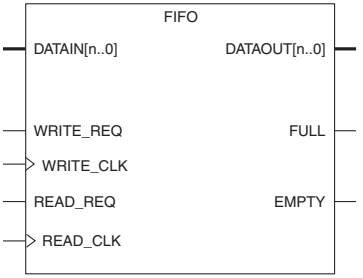
Figure 14–2. A Double Synchronizer Circuit



Using a FIFO Buffer

The advantage of using a FIFO buffer, shown in Figure 14–3, is that Altera's MegaWizard® Plug-In Manager makes it very easy to design a FIFO buffer. A FIFO buffer is useful when a data bus signal needs to be transferred across asynchronous clock domains, and some temporary storage of this data would also be beneficial. Additionally, a FIFO buffer circuit should not generate any Design Assistant warnings.

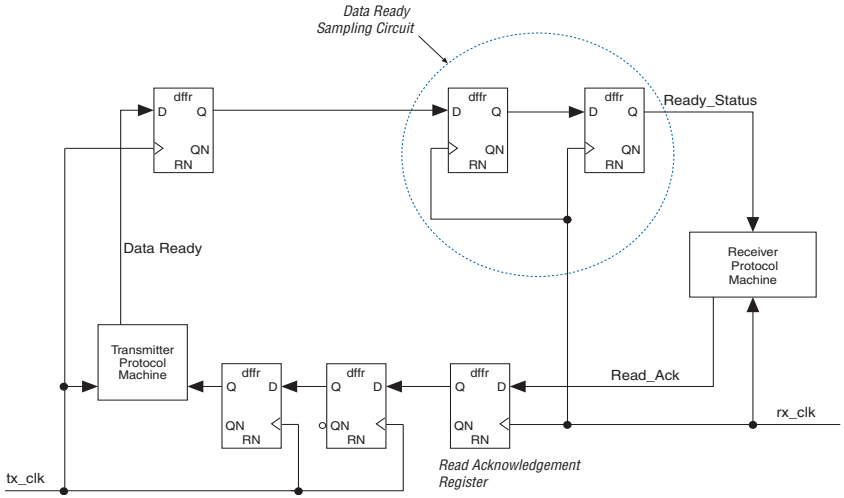
Figure 14–3. A FIFO Buffer



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 14–4, 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.

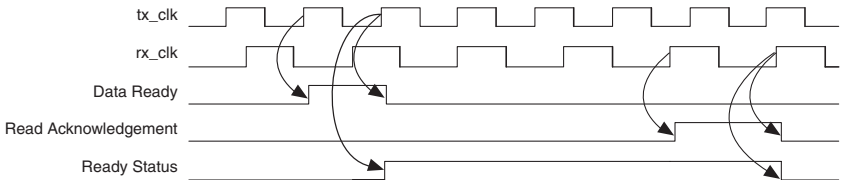
Figure 14–4. A Handshake Protocol Circuit



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 14–5.

Figure 14–5. Data Transfer Between Two Asynchronous Clock Domains



Gated Clocks

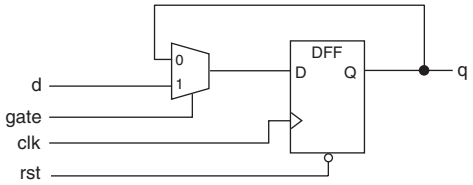
Clock gating is sometimes used to "turn off" parts of a circuit to reduce the total power consumption of a device. The gated clock signal prevents any of the logic driven by it from switching so the logic will not consume any power. This works best if the gating is done at the root of the clock tree. If the clock is gated at the leaf-cell level (for example, immediately before the input to the register), the device will not save much power because the whole clock network will still toggle. 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 14–6. 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 can be 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 14–6. 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
```

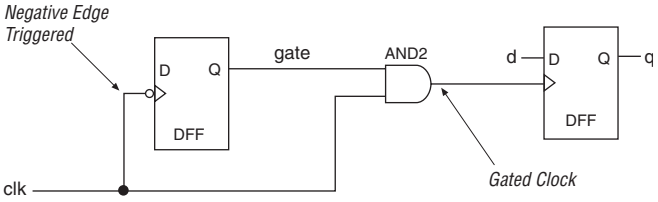
Alternative Clock Gating Circuits

If a clock gating circuit is absolutely necessary in the design, one of the following two circuits may also be used. The Design Assistant will not flag a violation for these circuits.

Clock Gating Circuit Using an AND Gate

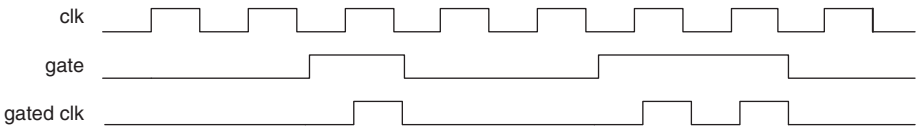
Designs can use a two input AND gate for a gated clock signal that feeds into positive-edge-triggered registers. One input to the AND gate is the original clock signal. The other input to the AND gate is the gating signal, which should be driven directly from a register clocked by the negative edge of the same original clock signal. Figure 14–7 shows this type of circuit.

Figure 14–7. 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 14–8 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 14–8. Timing Diagram for Clock Gating Circuit Using an AND Gate

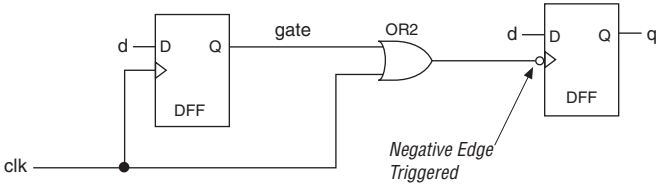


If the delay between the register that generates the `gate` signal and the `gate` input to the AND gate is greater than the low period of the clock, (one half of the clock period for a 50% duty cycle clock), the clock pulse width is narrowed.

Clock Gating Circuit Using an OR Gate

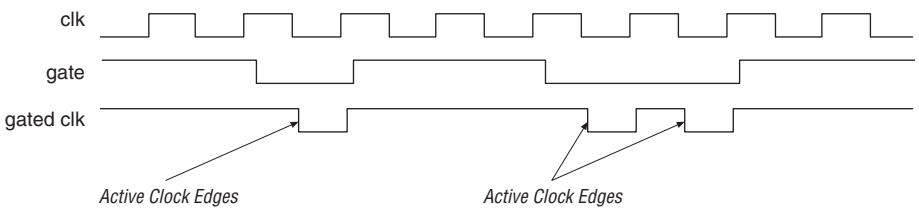
Use a two-input OR gate for a gated clock signal that feeds into a negative-edge-triggered register. One input to the OR gate is the original clock signal. The other input to the OR gate is the gating signal, which should be driven directly from a register clocked by the positive edge of the same original clock signal. Figure 14–9 shows this circuit.

Figure 14–9. 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 14–10 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 14–10. Timing Diagram for Clock Gating Circuit Using an OR Gate



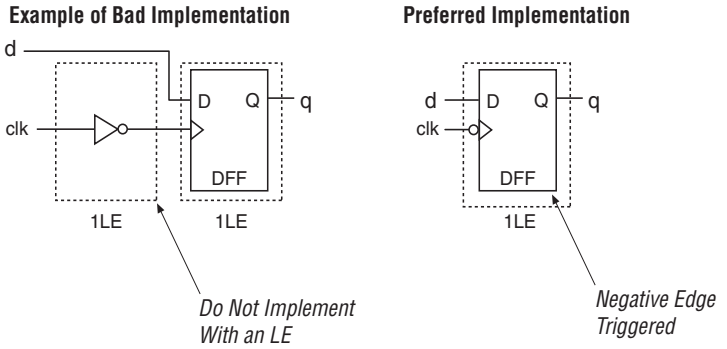
If the delay between the register that generates the `gate` signal and the `gate` input to the AND gate is greater than the low period of the clock, (one half of the clock period for a 50% duty cycle clock), the clock pulse width is narrowed.

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 14–11. 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 14–11. 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 will pose significant challenge to timing closure of the design. It will also consume more device resources than are necessary. See "Mixing Clock Edges" on page 14–14 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

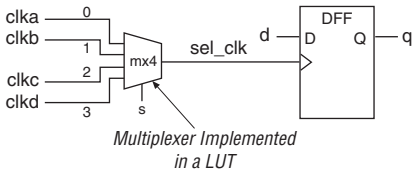
As a general guideline, clock sources should only be used to drive the register clock pins. There are exceptions to this rule, but every effort should be taken to minimize these exceptions or remove them altogether.

One category of exception is for various gated clocks, which are described in "Preferred Clock Gating Circuit" on page 14–7.

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 14–12, 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 will have 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. See the Quartus II Help for more information on preserving node names.

Figure 14–12. 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.

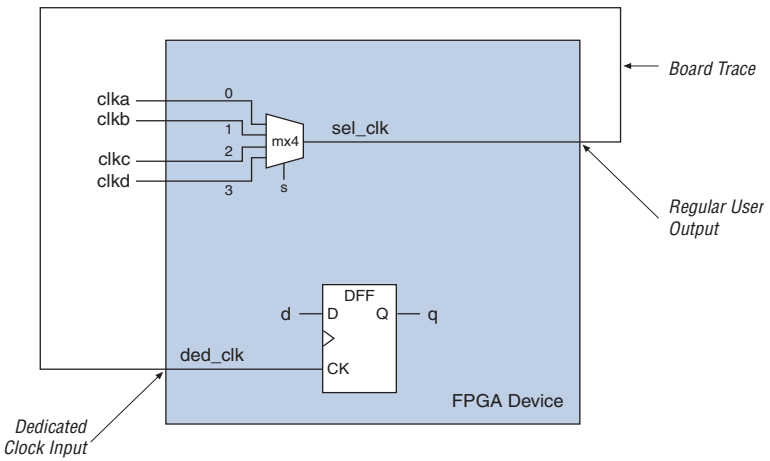
Enhanced PLL Clock Switchover

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 bring it back into the device, as shown in Figure 14–13.

Routing a multiplexed clock signal as shown in Figure 14–13 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 will still be a large delay through the multiplexing circuit and external board trace, the resulting clock skew will be 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 14–13. 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

Designers 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. Any time that a circuit needs to use both edges of a single clock, then the duty cycle of the clock has to be accurately be described to the Static Timing Analysis tool, otherwise inaccurate timing analysis could result. Figure 14–14 shows two clock waveforms. One has a 50% duty-cycle, the other has a 10% duty cycle.

Figure 14–14. Clock Waveforms with 50% and 10% Duty Cycles

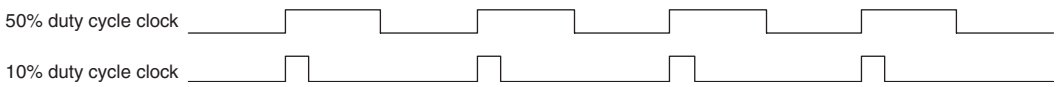


Figure 14–15 shows a circuit that uses only the positive edge of the clock. The distance between successive positive clock edges is always the same, for example, the clock period. For this circuit, the duty cycle of the clock has no effect on the performance of the circuit.

Figure 14–15. Circuit Using the Positive Edge of a Clock

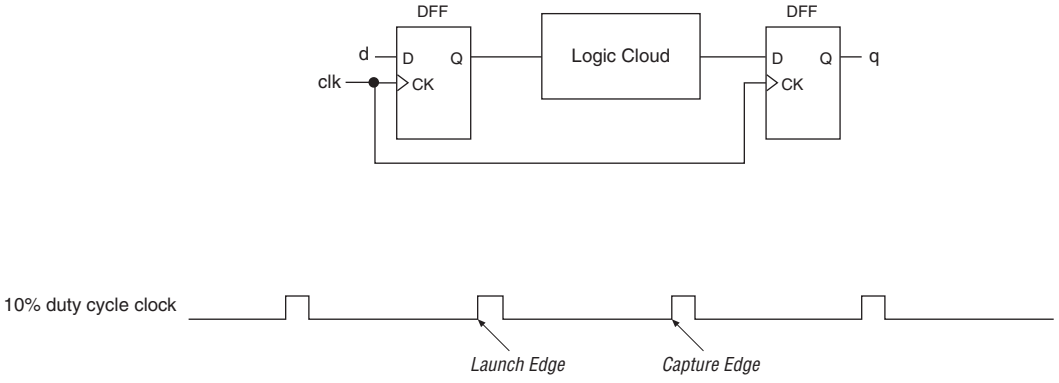
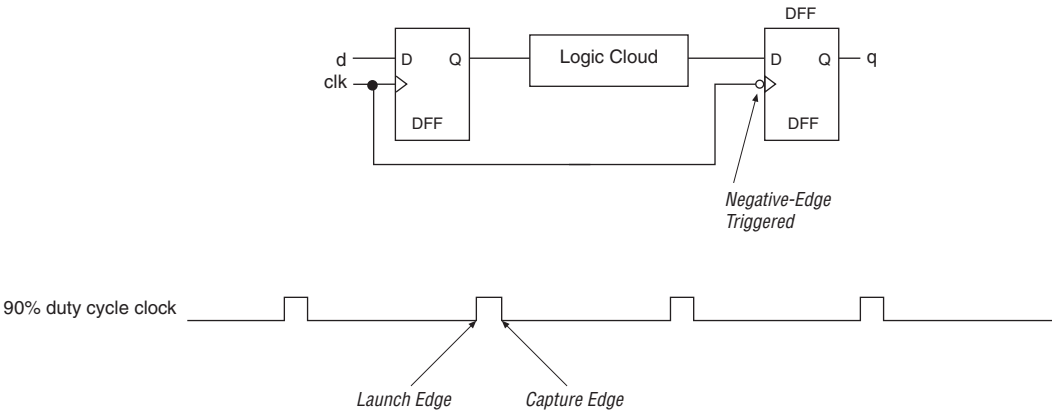


Figure 14–16 shows a circuit that used the positive clock edge to launch data and the negative clock edge to capture this data. Since this particular clock has a 10% duty cycle, the amount of time between the launch edge and capture edge is small. This small gap makes it difficult for the synthesis tool to optimize the cloud of logic so that no setup-time violations occur at the capture register.

Figure 14–16. Circuit Using the Positive and Negative Edges of a Clock



If you design a circuit that uses both clock edges, then you will potentially get the Design Assistant warning "Registers are Triggered by Different Edges of Same Clock." You will not get this warning under the following conditions:

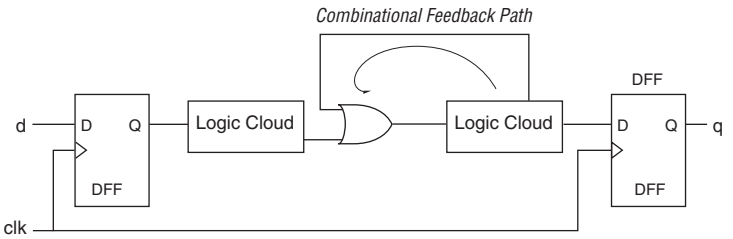
- If the opposite clock edge is used in a clock gating circuit
- A double data rate (DDR) memory interface circuit is used

Try to only use a single edge of a clock in a design.

Combinational Loops

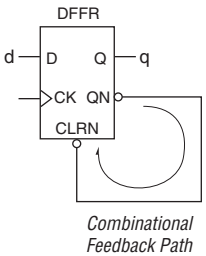
A combinational loop exists (see Figure 14–17) if the output of a logic gate (or gates) feeds back to the input of the same gate without first encountering a register. A design should not contain any combinational loops.

Figure 14–17. A Circuit Using a Combinational Loop



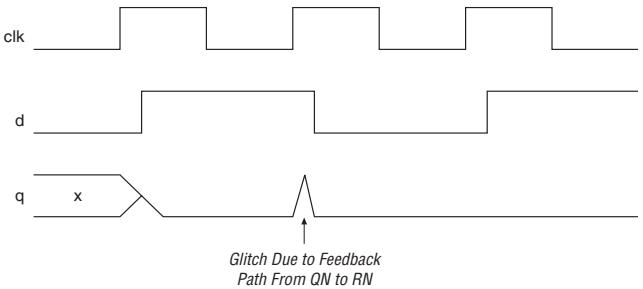
It is also possible to generate a combinational loop using a register (see Figure 14–18) if the register output pin drives the reset pin of the same register.

Figure 14–18. Generation of a Combinational Loop Using a Register



The timing diagram for this circuit is shown in Figure 14–19. 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 14–19. Timing Diagram for the Circuit Shown in Figure 14–18



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 will check 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 will behave 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.

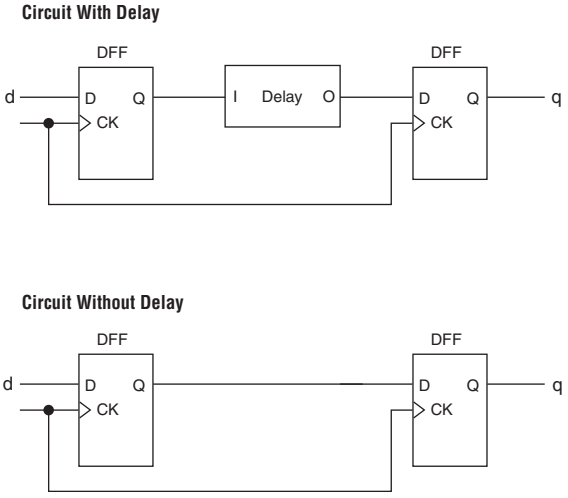
☞ Check your design for intentional and unintentional combinational loops, and remove them.

Intentional Delays

Altera does not recommend instantiating a cell that does not benefit a design. This cell will only delay the signal. For a synchronous circuit that uses a dedicated clock in the FPGA (see Figure 14–20), 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.

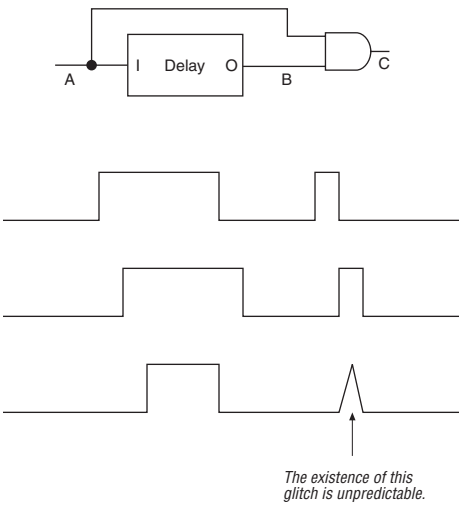
Figure 14–20 shows two versions of the same shift registers. Both circuits operate identically. The first version has a delay cell, possibly implemented using a LUT, in the data path from the Q output of the first register to the D input of the second register. The function of the delay cell is a non-inverting 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 14–20. 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 14–21. 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 14–21. 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.

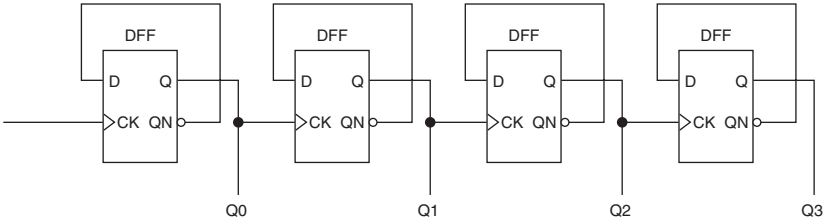
Be aware that not all cases of delay chains in a design are due to asynchronous circuitry. If the Design Assistant report states that you have delay chains that you are unaware of (or are not expecting), the delay chains may be a result of using pre-built intellectual property (IP) functions. Pre-built IP functions may contain delay chains which the Design Assistant will report. These functions are usually parametrizable, 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 14–22, 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 14–22. 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 14–23 shows the phase delay of the circuit in Figure 14–22.

Figure 14–23. Timing Diagram Showing Phase Delay of Circuit Shown in Figure 14–22

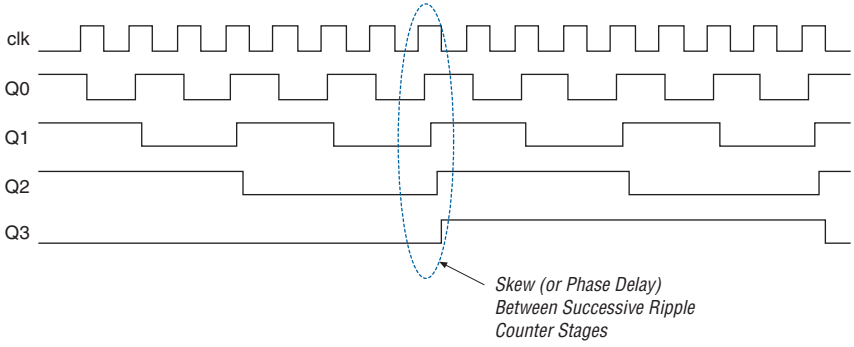
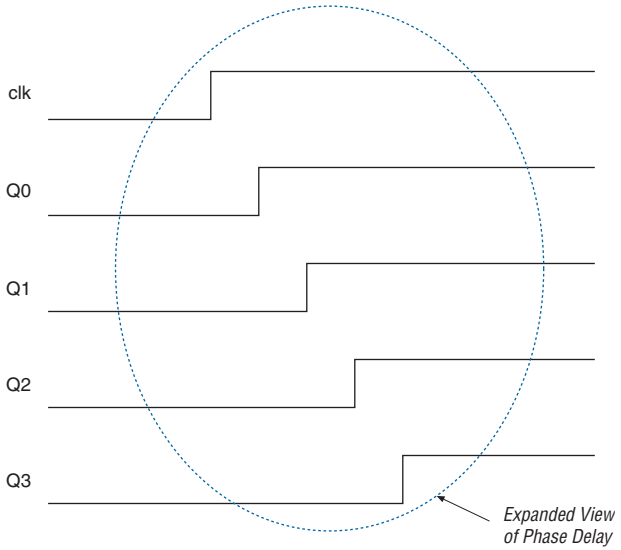


Figure 14–24 shows detailed view of the phase delay shown in Figure 14–23.

Figure 14–24. Detailed View of the Phase Delay Shown in Figure 14–23



This phase delay is problematic if the ripple counter outputs are used as clock signals for other circuits. Those other circuits will be 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 will become.

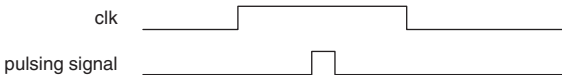
Altera recommends that you avoid using ripple counters under any circumstances.

Pulse Generators

A pulse generator is a circuit that generates a signal that has two or more transitions within a single clock period. Figure 14–25 on page 14–22 shows an example of a pulse generator waveform.

For more information on pulse generators, see "Intentional Delays" on page 14–18.

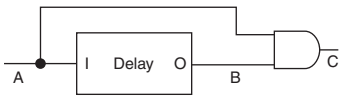
Figure 14–25. Example of a Pulse Generator Waveform



Creating Pulse Generators

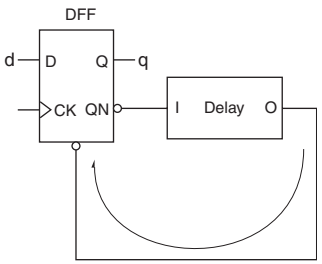
Pulse generators can be created in two ways. The first way to create a pulse generator is to increase the width of a glitch using a 2-input AND, NAND, OR, or NOR gate, where the source for the two gate inputs are the same, but the design delays the source for one of the gate inputs, as shown in Figure 14–26.

Figure 14–26. A Pulse Generator Circuit Using a 2-Input AND



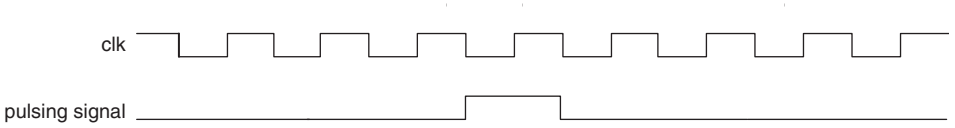
The second way to create a pulse generator is by using a register where the register output drives its own asynchronous reset signal through a delay chain, as shown in Figure 14–27.

Figure 14–27. Pulse Generator Circuit Using a Register Output to Drive a Reset Signal Through a Delay Chain



These pulse generators are asynchronous in nature and will be 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 14–28 on page 14–23.

Figure 14–28. An Example of a Synchronous Pulse Generator



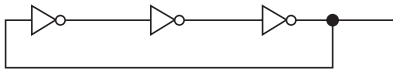
A synchronous pulse generator can be created with a simple section of Verilog HDL or VHDL code. The following is a Verilog HDL code fragment for a synchronous pulse generator circuit.

```
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

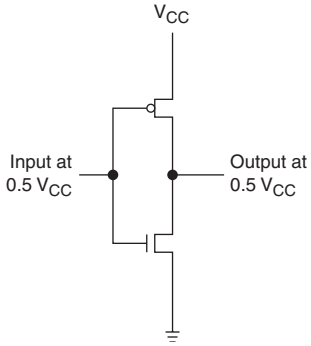
The circuit shown in Figure 14–29 on page 14–24 consists of a combinational logic gate whose inverted output feeds back to one of the inputs of the same gate. This feedback path causes the output to change state and, therefore, oscillate.

Figure 14–29. 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 will be 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 will go to a stable voltage at half of the supply voltage, as shown in Figure 14–30. This will cause 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 14–30. 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 14–31 shows a gated reset circuit where the signal driving into the register reset pin has glitches on it causing unintentional resetting.

Figure 14–31. A Gated Reset Circuit and its Associated Timing Diagram

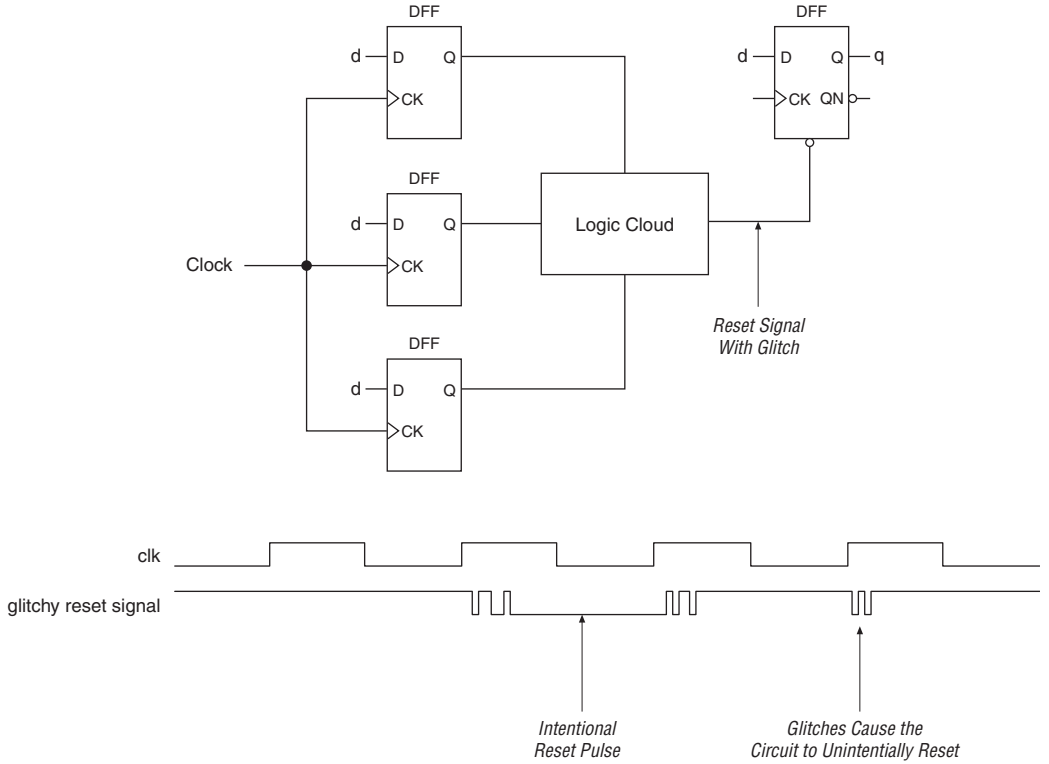
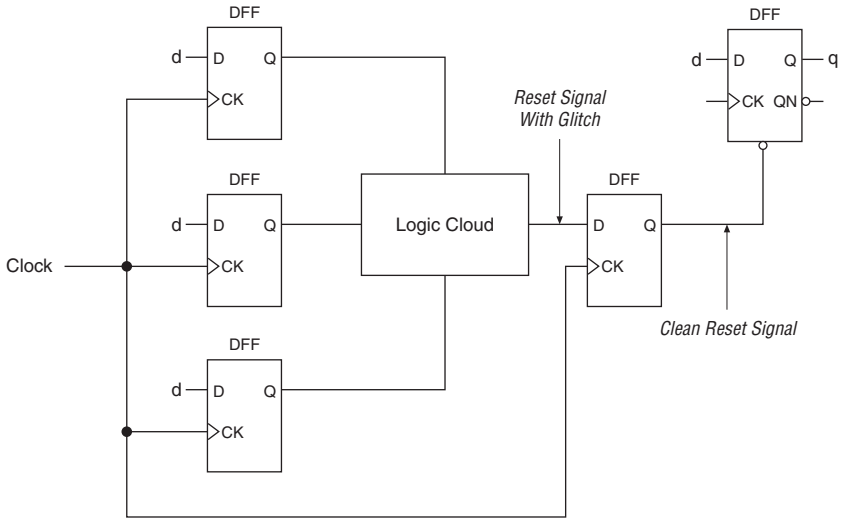


Figure 14–32 on page 14–26 shows a better approach to implement a gated reset circuit, by placing a register on the output of the reset-gating logic, thereby synchronizing it to a clock. The register output then becomes a glitch-free reset signal that drives the rest of the design. However, the resulting reset signal is delayed by an extra clock cycle.

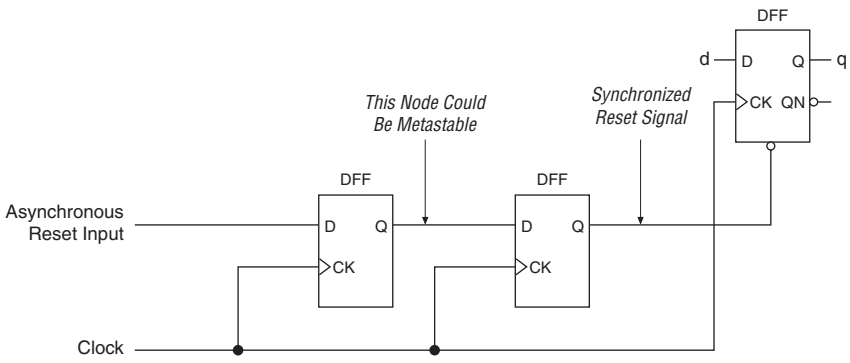
Figure 14–32. A Better Approach to the Gated Reset Circuit in Figure 14–31



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 14–33.

Figure 14–33. A Double-Buffer Circuit



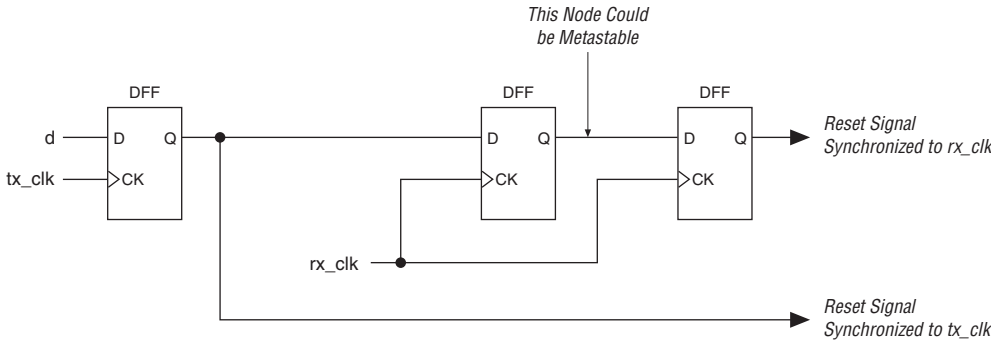
Synchronizing Reset Signals Across Clock Domains

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 14–34.

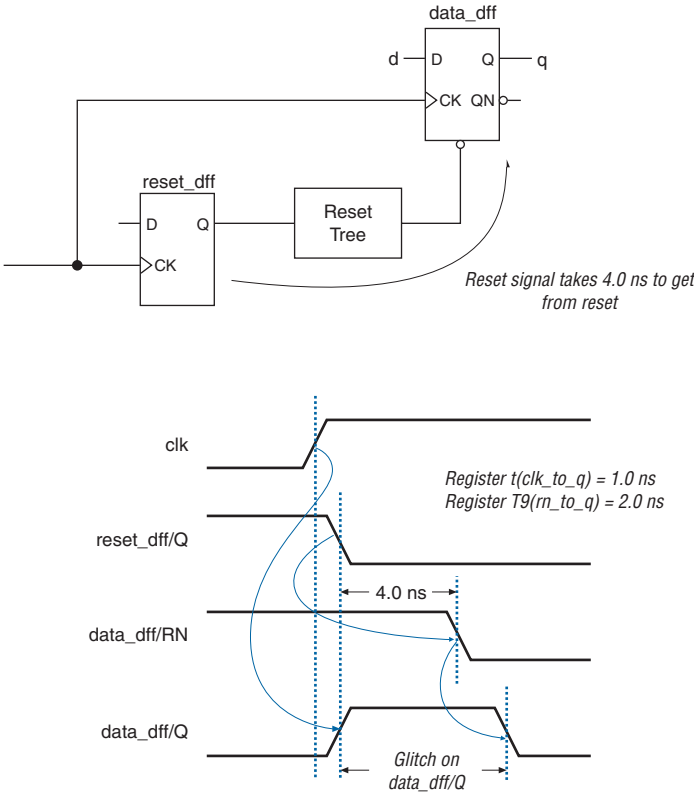
- The reset signal should be synchronized with two or more cascading registers in the receiving asynchronous clock domain.
- The cascading registers should be triggered on the same clock edge.
- There should be no logic between the output of the transmitting clock domain and the cascaded registers in the receiving asynchronous clock domain.

Figure 14–34. Circuit for a Synchronized Reset Signal Across Two Clock Domains



With either of the reset synchronization circuits described in Figures 14–33 and 14–34, 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 14–35, demonstrate this phenomenon.

Figure 14–35. Common Problem with Reset Synchronization Circuits



A purely synchronous reset circuit will not exhibit this behavior. The following Verilog HDL RTL code shows how to do this.

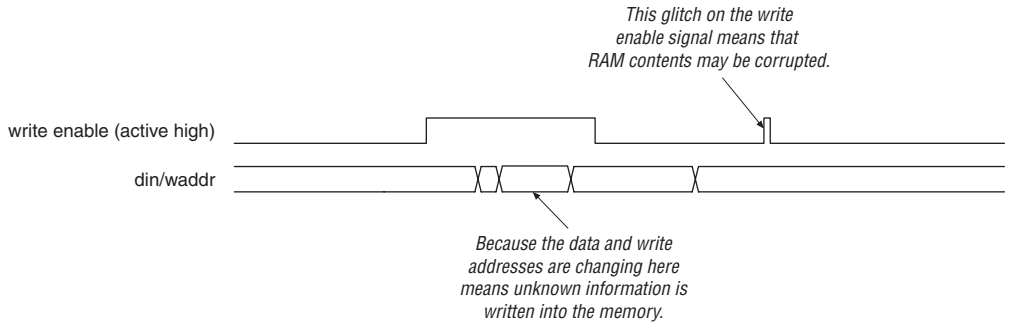
```
always @ (posedge clk)
      begin
      if (!rst)
         q <= 1'b0;
      else
         q <= d; end
```

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 14–36. 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 14–36. 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.

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

H51012-2.0

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

There are three power-up options available for HardCopy series devices:

- Instant on
- Instant on after 50 ms
- Configuration emulation of an FPGA configuration sequence

Designers 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. See the "Configuration Emulation of FPGA Configuration Sequence" section for more information.

HardCopy II and HardCopy Stratix devices retain the functionality of VCCSEL and PORSEL pins from the Stratix FPGAs. The signals can affect the HardCopy series power-up behavior using any power up option. See the *Stratix Device Handbook* for proper use of these additional signals.

Instant On

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, designers 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.

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.

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. 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.

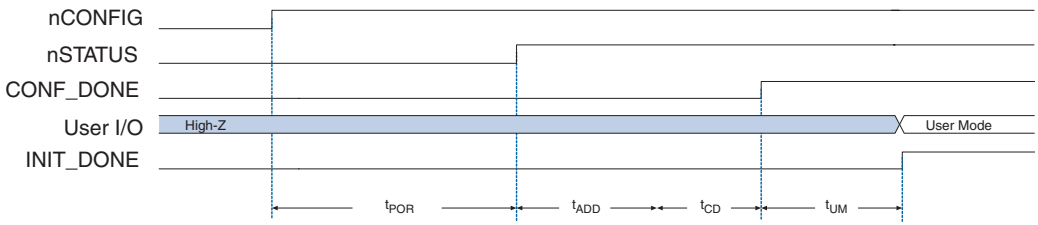
In the FPGA, the INIT_DONE pin can be optionally enabled in the Quartus II software. If INIT_DONE pin was used on the FPGA prototype, the HardCopy series device retains its function.

- 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 (e.g., 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 FPGA. For more information on PORSEL settings, see the *Configuration 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.
- 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.

Figure 15–1 shows a waveform of the configuration signals and user I/O pins in a HardCopy series device using the instant on power-up mode.

Figure 15–1. Timing Waveform for Instant On Option



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 15–1, regardless of the logic state of the nCONFIG signal. The INIT_DONE signal transitions high only after the CONF_DONE signal transitions high.

Pulsing the nCONFIG signal on an FPGA will re-initialize the configuration sequence. The nCONFIG signal on a HardCopy series device also restarts the initialization sequence.

Tables 15–1 through 15–3 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 15–1 for HardCopy series devices. This option is typical of an ASIC's functionality.

Table 15–1. Timing Parameters for Instant On Mode in HardCopy II Devices

Parameter	Description	Condition	Min	Typ	Max	Units
t_{POR}	PORSEL delay	12		12		ms
		100		100		ms
t_{ADD}	Additional delay	Instant on	*(1)*		*(1)*	ms
t_{CD}	CONF_DONE delay			*(1)*		µs
t_{UM}	User mode delay			*(1)*		µs

Note to Table 15–1:
(1) Contact Altera Applications.

Table 15–2. Timing Parameters for Instant On Mode in HardCopy Stratix Devices

Parameter	Description	Condition	Min	Typ	Max	Units
t_{POR}	PORSEL delay	2	1	2		ms
		100	70	100		ms
t_{ADD}	Additional delay	Instant on	4		8	ms
t_{CD}	CONF_DONE delay		0.5		3.0	µs
t_{UM}	User mode delay		6.0		28	µs

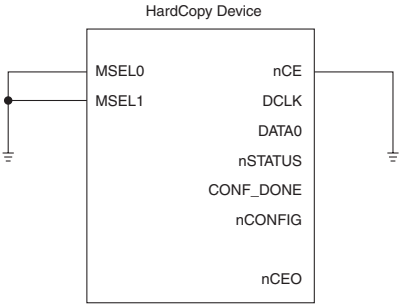
Table 15–3. Timing Parameters for Instant On Mode in HardCopy APEX Devices

Parameter	Description	Min	Max	Units
t_{POR}	POR delay	*(1)*	*(1)*	µs
t_{ADD}	Additional delay	0	0	µs
t_{CD}	CONF_DONE delay	0.5	3.0	µs
t_{UM}	User mode delay	2.5	8	µs

Note to Table 15–3:
(1) Contact Altera Applications.

Figure 15–2 shows a HardCopy APEX device in instant on mode where MSEL0 and MSEL1 are tied to ground. The FPGA uses the passive serial (PS) configuration scheme with a configuration device.

Figure 15–2. Instant On Power Up of a HardCopy APEX Device



The instant on option can also be used if the original APEX FPGA was configured using passive parallel asynchronous (PPA) or passive parallel synchronous (PPS) configuration schemes.

For HardCopy Stratix devices, pull the `MSEL0`, `MSEL1`, and `MSEL2` pins to appropriate logic levels (as defined for Stratix FPGAs) for the chosen power-up option.

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, these external pull-up resistors can be removed.

Altera recommends leaving external pull-up resistors on the board if one of the following conditions exists. For more information, see 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 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 may be left unconnected on the board.

Instant On After 50 ms Delay

The instant on after 50 ms delay power up mode is similar to the instant on power up option. 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) need to be initialized prior to the normal operation of the HardCopy series device.

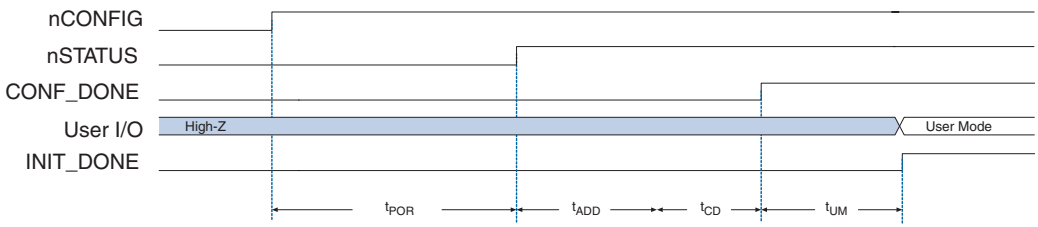
An on-chip oscillator generates the 50-ms delay after the power-up sequence. During the POR sequence and delay period, all user I/O pins can be driven high by internal weak pull-up resistors. Similar to the instant on mode, these internal weak pull-up resistors are affected by the nIO_pullup pin.

The FPGA enters initialization phase immediately after configuration. During this phase, 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.

Pulsing the nCONFIG signal on the FPGA re-initializes the configuration sequence. The nCONFIG signal on the HardCopy series device also restarts the initialization sequence.

Figure 15–2 shows a waveform of the configuration signals and user I/O pins when the instant on after 50 ms power up mode is used.

Figure 15–3. Timing Waveform for Instant On After 50 ms



The timing parameters for the instant on after 50 ms delay mode are given in Tables 15–4 through 15–6.

Table 15–4. Timing Parameters for Instant On After 50 ms Mode in HardCopy II Devices

Parameter	Description	Condition	Min	Typ	Max	Units
t_{POR}	PORSEL delay	12		12		ms
		100		100		ms
t_{ADD}	Additional delay	After 50 ms	25	50	75	ms
t_{CD}	CONF_DONE delay			*(1)*		µs
t_{UM}	User mode delay			*(1)*		µs

Note to Table 15–4:
(1) Contact Altera Applications.

Table 15–5. Timing Parameters for Instant On After 50 ms Mode in HardCopy Stratix Devices

Parameter	Description	Condition	Min	Typ	Max	Units
t_{POR}	PORSEL delay	2	1	2		ms
		100	70	100		ms
t_{ADD}	Additional delay	After 50 ms	25	50	75	ms
t_{CD}	CONF_DONE delay		0.5		3.0	µs
t_{UM}	User mode delay		6.0		28	µs

Table 15–6. Timing Parameters for Instant On After 50 ms Mode in HardCopy APEX Devices

Parameter	Description	Min	Max	Units
t_{POR}	POR delay	*(1)*	*(1)*	µs
t_{ADD}	Additional delay	25	75	ms
t_{CD}	CONF_DONE delay	0.5	3.0	µs
t_{UM}	User mode delay	2.5	8.0	µs

Note to Table 15–6:
(1) Contact Altera Applications.

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.

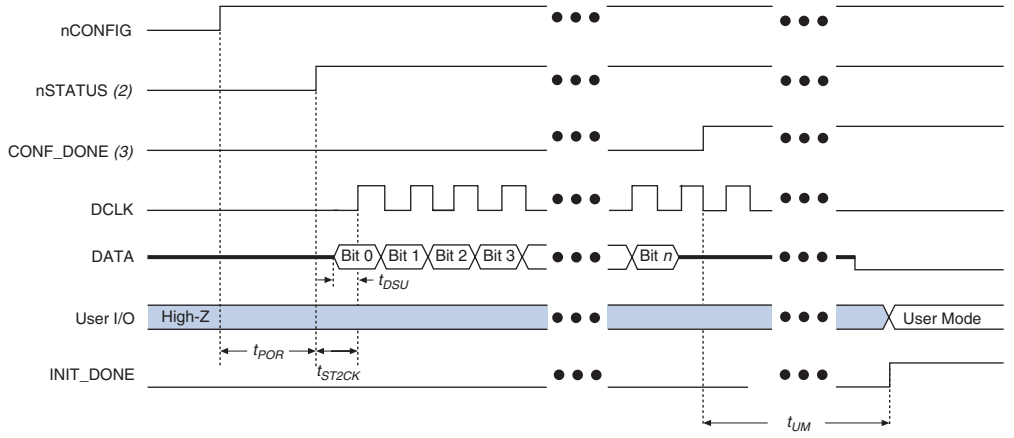
Similar to APEX 20K FPGAs, HardCopy APEX devices do not have an nIO_pullup function. Their internal weak pull-up resistors are enabled during the power up and initialization phase.

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 15–4 shows a waveform of the configuration signals and the user I/O signals using configuration emulation mode.

Figure 15–4. Timing Waveform for Configuration Emulation Mode



Configuration Emulation Timing Parameters

Tables 15–7 and 15–8 provide the timing parameters for the configuration emulation mode.

Table 15–7. Timing Parameters for Configuration Emulation Mode in HardCopy Stratix Devices *Note (1)*

Parameter	Description	Condition	Min	Typ	Max	Units
t_{POR}	PORSEL delay	2	1	2		ms
		100	70	100		ms
t_{DSU}	Data setup time *(2)*		7			ns
t_{ST2CK}	nSTATUS to DCLK *(2)*		1			µs
t_{UM}	User mode delay		6.0		28	µs

Notes to Table 15–7:
(1) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode.
(2) This parameter is similar to Stratix FPGA specifications. See the *Configuration Handbook* for more information.

Table 15–8. Timing Parameters for Configuration Emulation Mode in HardCopy APEX Devices

Parameter	Description	Min	Max	Units
t_{POR}	POR delay	*(1)*	*(1)*	ms
t_{DSU}	Data setup time *(2)*	25	75	ms
t_{ST2CK}	nSTATUS to DCLK *(2)*	1.0	3.0	µs
t_{UM}	User mode delay	2.5	8.0	µs

Notes to Table 15–8:
(1) Contact Altera Applications.
(2) This parameter is similar to APEX 20KE and APEX 20KC FPGA specifications. See the *Configuration Handbook* for more information.

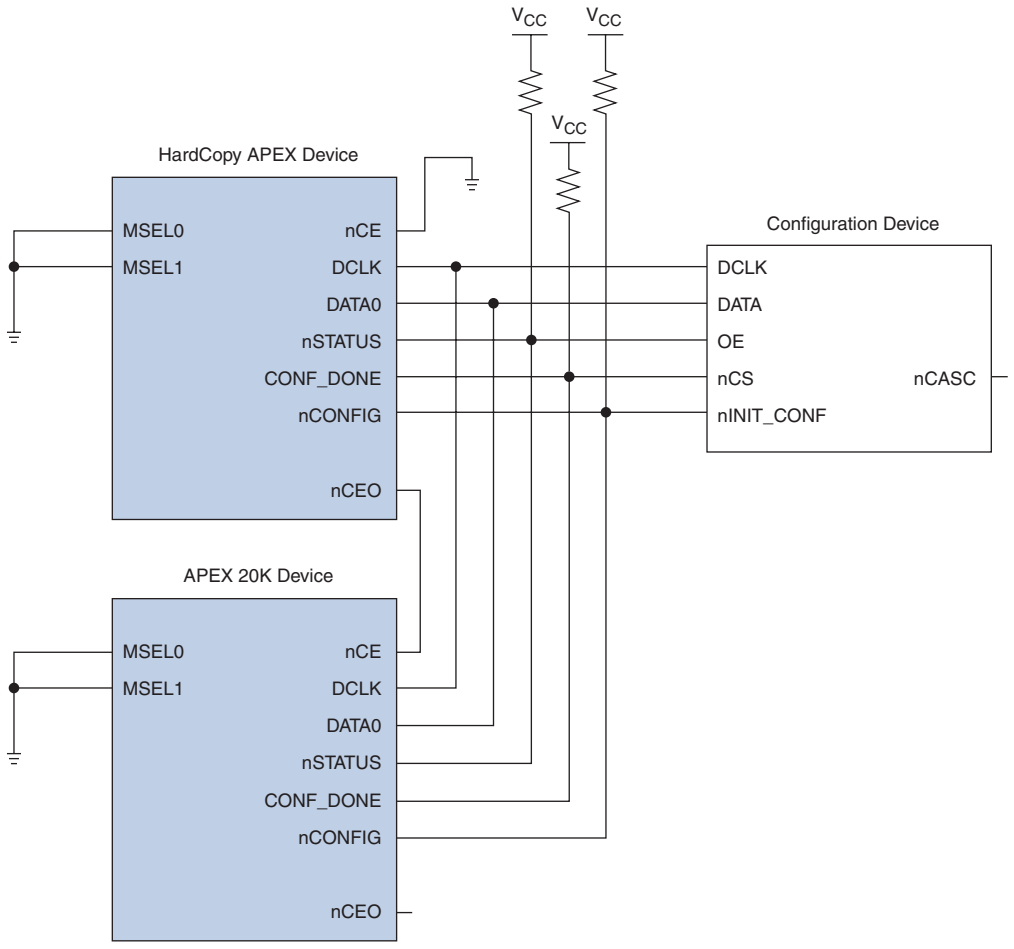
Benefits of Configuration Emulation

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 15–5, a HardCopy APEX device replaces an APEX FPGA.

Figure 15–5. 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 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 15–9 shows a comparison of applicable FPGA and HardCopy power up options.

Table 15–9. 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)*	✓	✓				
Remote local update PPA *(3)*	✓	✓				
Remote local update PS *(3)*		✓				

Notes to Table 15–9:

(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 15–10 gives a summary of HardCopy series device power up options when a single HardCopy series device replaces a single FPGA on the board.

Table 15–10 does not include HardCopy II options because HardCopy II devices only support instant on and instant on after 50 ms modes.

Table 15–10. 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 15–10:

(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 15–11 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 15–11 does not include HardCopy II options because HardCopy II devices only support instant on and instant on after 50 ms modes.

Table 15–11. 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 15–11. 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 15–11:

(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 15–7. 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 15–6 and 15–7 show how a HardCopy series device replaces an FPGA previously configured with an Altera configuration device.

Figure 15–6. Configuration of a Stand-Alone FPGA *Note (1)*

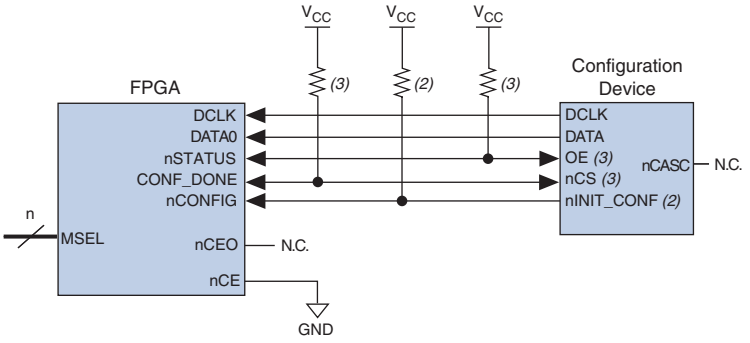
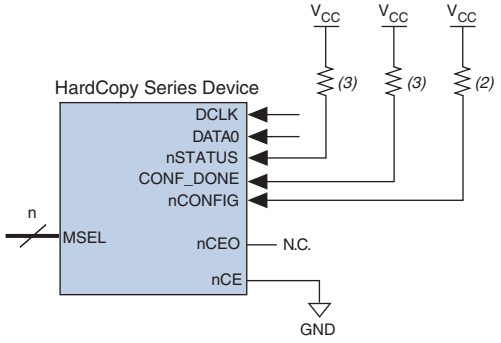


Figure 15–7. HardCopy Series Device Replacing Stand-Alone FPGA *Note (1)*



Notes to Figures 15–6 and 15–7:

(1) For details on configuration interface connections, see 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}.

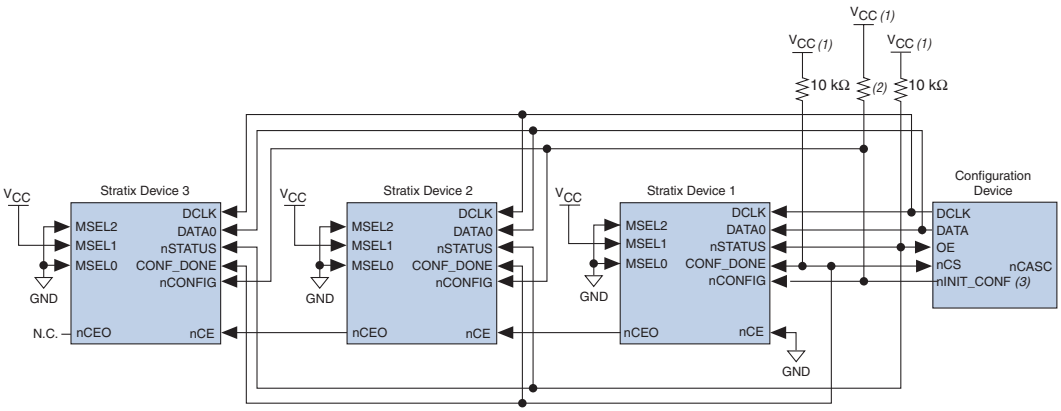
(3) Enhanced configuration and EPC2 devices have internal programmable pull-up resistors on `OE` and `nCS` pins. See the *Configuration Handbook* for more details of this application in FPGAs. HardCopy series devices have internal weak pull-up resistors on `nSTATUS`, `nCONFIG`, and `CONF_DONE` pins.

HardCopy Series Device Replacing an FPGA in a Cascaded Configuration Chain

Figure 15–8 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 15–9.

For more information on Stratix FPGA configuration schemes, see the *Configuration Handbook*.

Figure 15–8. Configuration of Multiple FPGAs in a Cascade Chain



Notes to Figure 15–8:

(1) The pull-up resistors are connected to the same supply voltage as the configuration device.

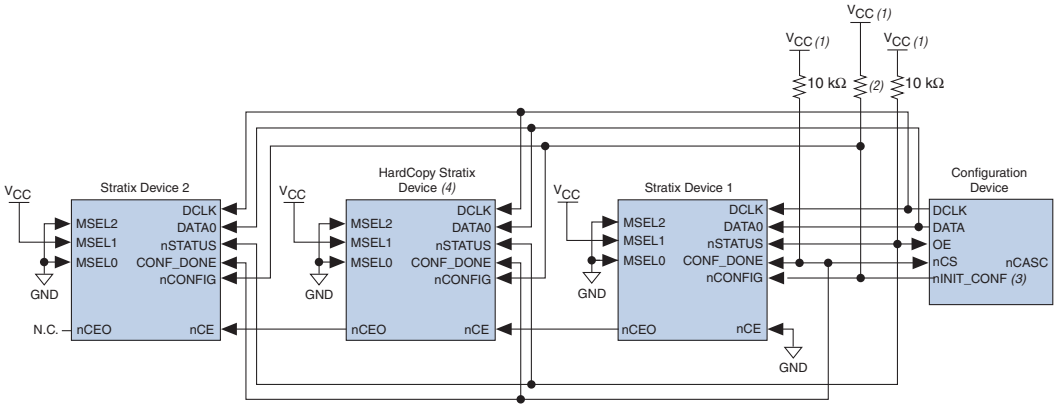
(2) The enhanced configuration devices and EPC2 devices have internal programmable pull-up resistors on the `OE` and `nCS` pins. See the *Configuration Handbook* for more details.

(3) The `nINIT_CONF` pin is available on EPC16, EPC8, EPC4, and EPC2 devices. See the *Configuration Handbook* for more details.

Configuration with the HardCopy Series Device in the Cascade Chain

Figure 15–9 shows the same cascade chain as Figure 15–8, but the second FPGA in the chain has been replaced with a HardCopy Stratix device.

Figure 15–9. Replacing an FPGA with a HardCopy Equivalent in the Cascade Chain



Notes to Figure 15–9:

(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. See the *Configuration Handbook* for more details.

(3) The nINIT_CONF pin is available on EPC16, EPC8, EPC4, and EPC2 devices. See the *Configuration Handbook* for more information.

(4) HC1S80, HC1S60, and HC1S25 devices do not support emulation mode and cannot be used in this method.

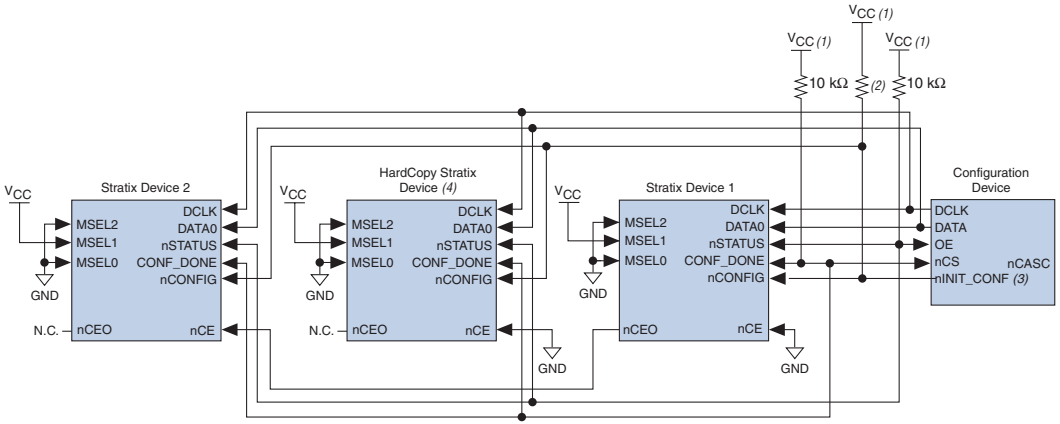
In this example, the HardCopy Stratix device can only be configured using the configuration emulation mode. The configuration device cannot be removed, as it is still required by other Stratix devices in the chain. While the HardCopy Stratix device does not need the data stored in the configuration device, 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 15–10 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 15–10. Configuration With the HardCopy Series Device Removed From the Cascade Chain



Notes to Figure 15–10:

(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. See the *Configuration Handbook* for more details.

(3) The nINIT_CONF pin is available on EPC16, EPC8, EPC4, and EPC2 devices. See 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.

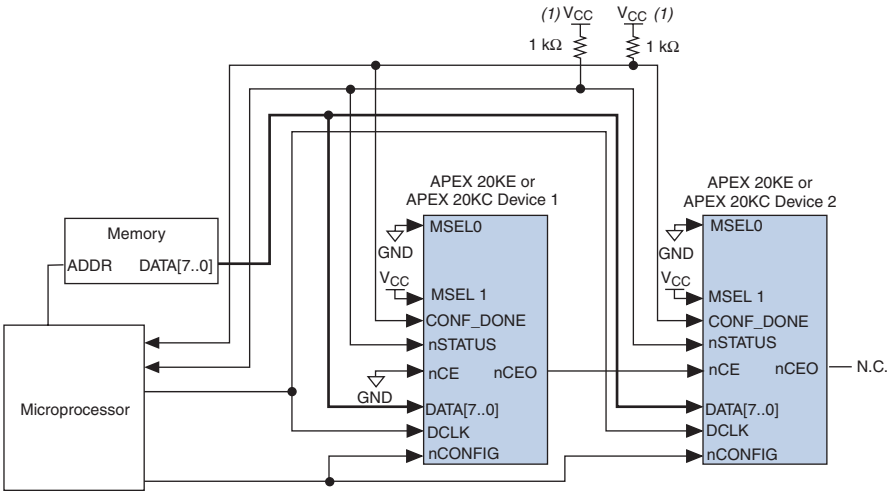
HardCopy Series Device Replacing an FPGA Configured Using a Microprocessor

The HardCopy series device can replace FPGAs that are configured using a microprocessor, as shown in Figures 15–12 and 15–13. While the instant on mode is the most efficient, designers can also use the instant on after 50 ms and configuration emulation mode.

Figure 15–11 shows an application where APEX FPGAs are configured using a microprocessor in the PPS configuration scheme.

For more information on the PPS configuration scheme, see the *Configuration Handbook*.

Figure 15–11. Configuring FPGAs Using a Microprocessor



Note to Figure 15–11:

(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 15–12), 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 15–12 and 15–13 show the HardCopy APEX device replacing APEX FPGAs either first or last in the configuration chain.

Figure 15–12. Replacement of Last FPGA in the Chain With a HardCopy Series Device

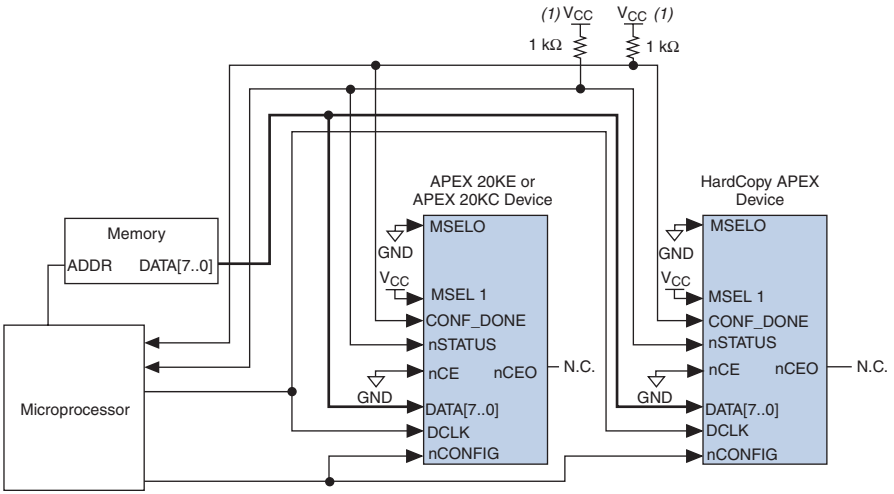
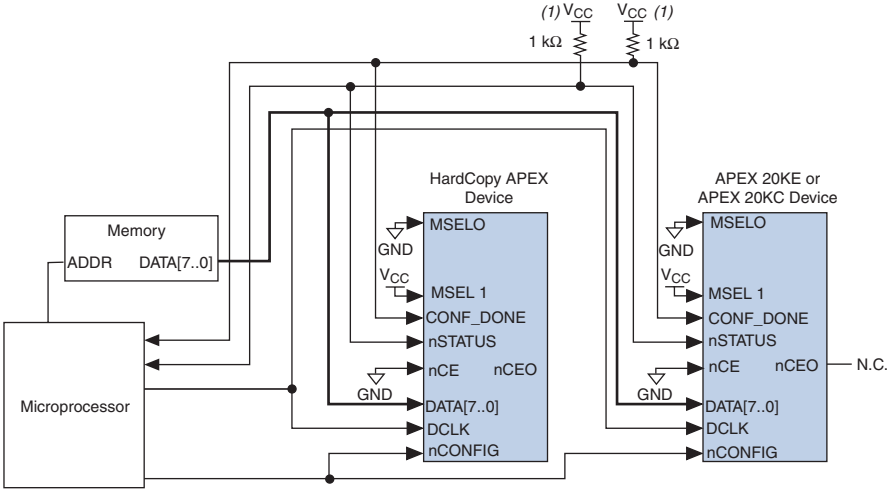


Figure 15–13. Replacement of First FPGA in the Chain With a HardCopy Series Device



Note to Figures 15–12 and 15–13:

(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 15–13), 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

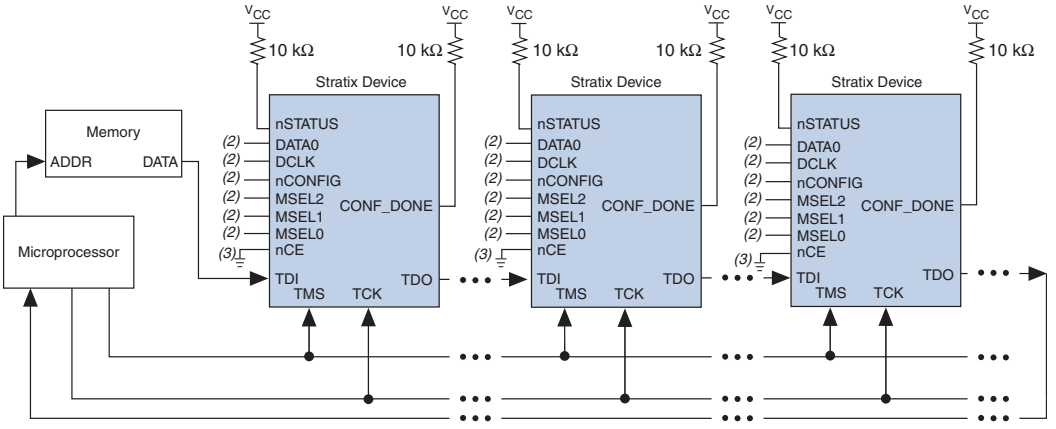
In this example, the circuit connectivity is maintained and there are no changes made to the board. The HardCopy series device can use either of the following power-up options when applicable.

- 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 15–14 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. The configuration data is generated by an on-board microprocessor.

Figure 15–14. Configuring FPGAs in a JTAG Chain Using a Microprocessor *Note (1)*



Notes to Figure 15–14:

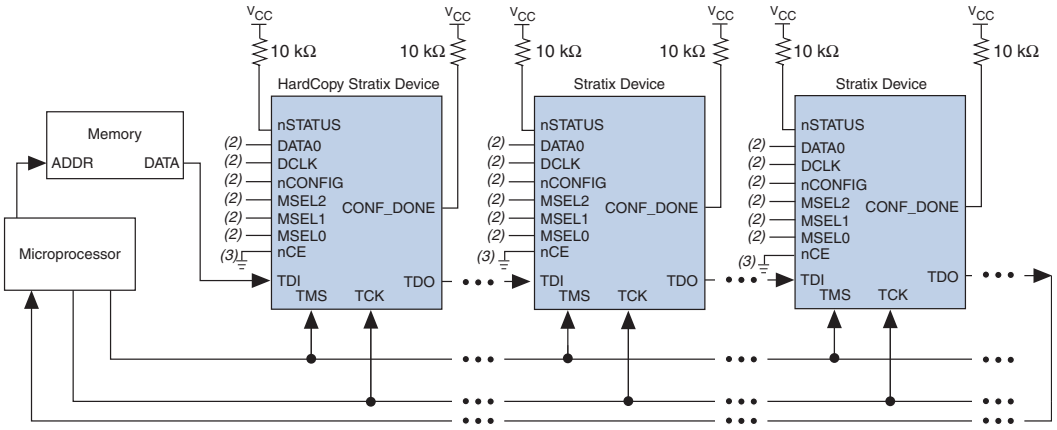
(1) Stratix II, Stratix, and APEX 20K devices can be placed within the same JTAG chain for device programming and configuration.

(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 15–15 shows an example where the first Stratix device in the JTAG chain is replaced by a HardCopy Stratix device.

Figure 15–15. Replacement of the First FPGA in the JTAG Chain With a HardCopy Series Device *Note (1)*



Notes to Figure 15–15:

(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.

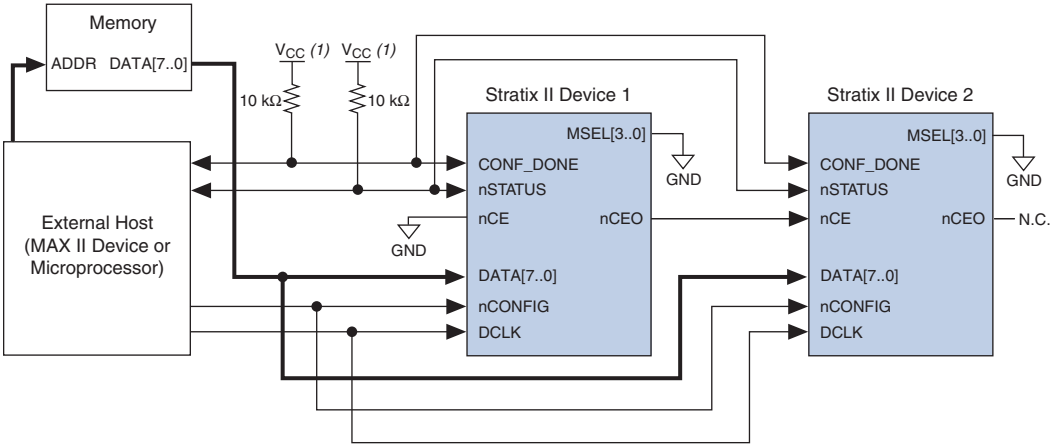
(3) `nCE` must be connected to GND or driven low for successful JTAG configuration.

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 15–16 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, see the *Configuration Handbook*.

Figure 15–16. Multiple-Device FPP Configuration Using a Microprocessor or MAX II Device

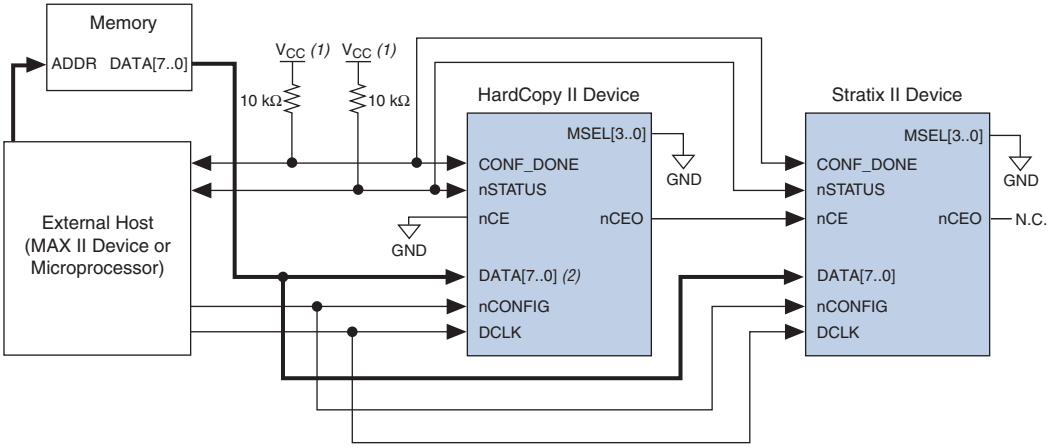


Note to Figure 15–16:

(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 15–17 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 15–17. Replacement of the First FPGA in the FPP Configuration Chain With a HardCopy Series Device



Notes to Figure 15–17:

(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 can be easily made early in 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.

16. Back-End Timing Closure for HardCopy Series Devices

H51013-2.0

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 will extract the final timing results after the HardCopy device back-end design process is completed. For more information on the HardCopy back-end design flow, see 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 see the *Timing Analysis* section of the *Quartus II Handbook*, Volume 2.

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 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 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 will 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 of these strips 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, see 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, you must also use for the HardCopy implementation. If you did not use timing constraints or 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 will be 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 16–1 and 16–2 show these clock attributes.

Figure 16–1. Clock Attributes

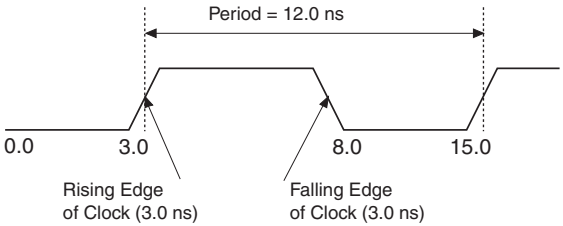
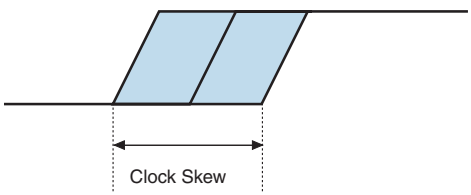


Figure 16–2. Clock Skew



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 16–3 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 16–4 shows a generic circuit with an on-chip hold-time constraint.

Figure 16–3. On-Chip Setup-Time Constraint

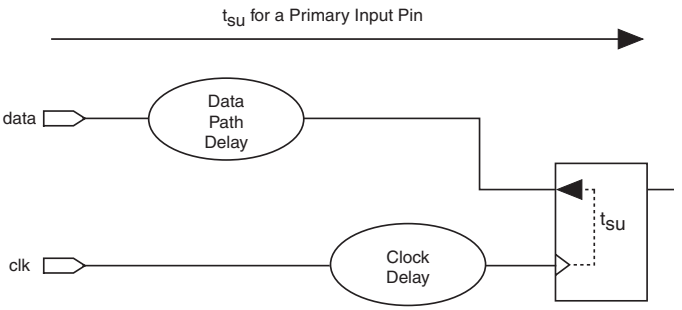
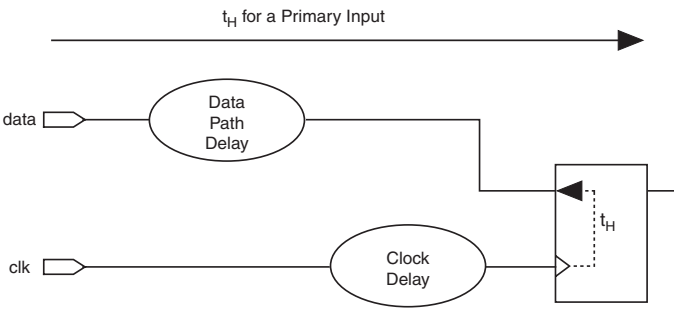


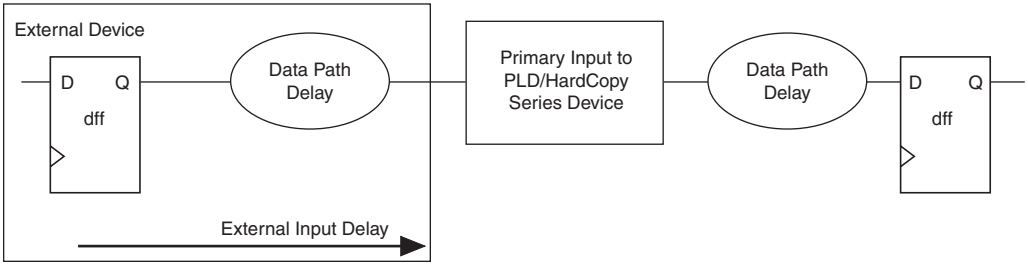
Figure 16–4. On-Chip Hold-Time Constraint



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 16–5 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 will occur.

Figure 16–5. External Timing Constraint Driving a Primary Input Pin



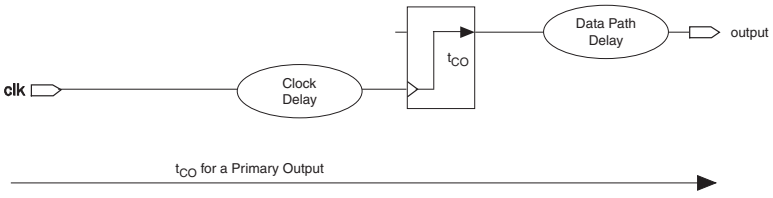
Primary Output Pin Timing

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 16–6 shows a generic circuit with an on-chip t_{CO} time constraint. In addition, there can be a minimum t_{CO} requirement.

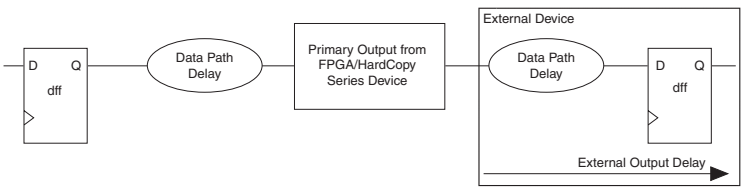
Figure 16–6. On-Chip Clock-to-Output (t_{CO}) Time Constraint



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 16–7 shows the external timing constraint driven by the primary output pin. The static timing analysis tool will use this information to check that the on-chip timing of the output signals is within the desired specification.

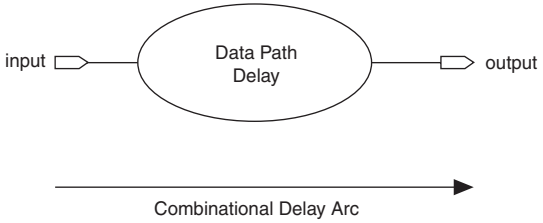
Figure 16–7. External Timing Constraint for a Primary Output Pin



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 16–8 shows for a generic circuit, the placement requirement for a combinational delay arc constraint.

Figure 16–8. 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 16–1 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 16–1. HardCopy APEX Static Timing Analysis Before Hold-Time Violation Fix (Part 1 of 3)

```
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 (mxcascout)	0.06 *	2.59 f	(3)
GR23_GC0_L20_LE8/um6/dcout (c1110)	0.00 *	2.59 f	(3)
data arrival time		2.59	

Table 16–1. HardCopy APEX Static Timing Analysis Before Hold-Time Violation Fix (Part 2 of 3)

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)

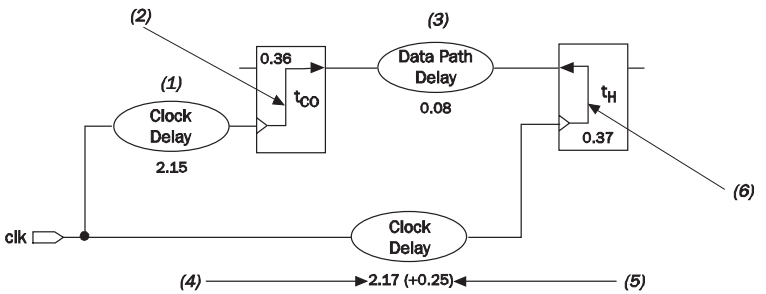
Table 16–1. HardCopy APEX Static Timing Analysis Before Hold-Time Violation Fix (Part 3 of 3)

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 16–1:
(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 16–9.

Figure 16–9 shows the circuit described by the Table 16–1 static timing analysis report.

Figure 16–9. 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 (see Table 16–2). The netlist changes are in ***bold italic*** type.

Table 16–2. HardCopy APEX Static Timing Analysis After Hold-Time Violation Fix (Part 1 of 2)

```
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)

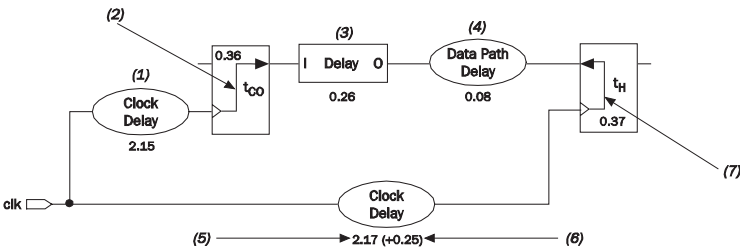
Table 16–2. HardCopy APEX Static Timing Analysis After Hold-Time Violation Fix (Part 2 of 2)

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 16–2:
(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 16–10.

Figure 16–10 shows the circuit described by the Table 16–2 static timing analysis report.

Figure 16–10. 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 slack = data delay – clock delay – μt_H

t_H slack = (2.15 + 0.36 + 0.26 + 0.08) – (2.17 + 0.25) – 0.37

t_H slack = +0.06 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 are maintained, 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 will contain 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 might be 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, which is before 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 16–3 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 16–4 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. For more information on this timing report, see Figure 16–11.

Table 16–3. HardCopy APEX Timing Report Before Place-&-Route Process (Part 1 of 2)

```
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)

Table 16–3. HardCopy APEX Timing Report Before Place-&-Route Process (Part 2 of 2)

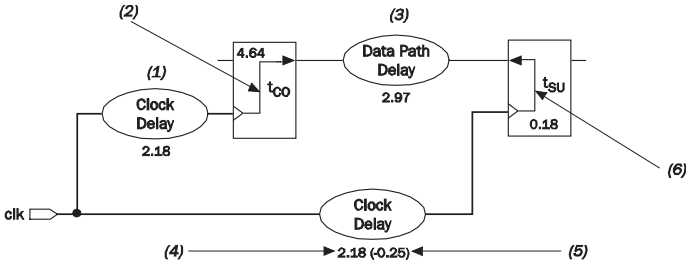
data arrival time		9.79	(3)
clock clkx' (fall edge)	7.41	7.41	
clock network delay (propagated)	2.18	9.59	(4)
clock uncertainty	-0.25	9.34	(5)
GR4_GC0_L5_LE2/um6/clk (c1110)		9.34 f	
Point	**Incr**	**Path**	**Reference Point (1)**
library setup time	-0.18	9.16	(6)
data required time		9.16	
data required time		9.16	
data arrival time		-9.79	
slack (VIOLATED)		-0.63	

Note to Table 16–3:

(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 16–11.

Figure 16–11 shows the circuit that Table 16–3 static timing analysis report describes.

Figure 16–11. 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 16–4 shows the timing report for the same path. The changes to the netlist are in ***bold italic*** type. For more information on this timing report, see Figure 16–12.

Table 16–4. HardCopy APEX Timing Report After the Place-&-Route Process (Part 1 of 2)

```
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)

Table 16–4. HardCopy APEX Timing Report After the Place-&-Route Process (Part 2 of 2)

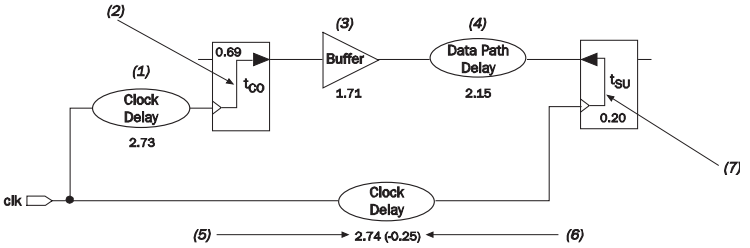
GR4_GC0_L5_LE0/um2/COMBOUT (icombout)	0.09 *	6.84 r	(4)
GR4_GC0_L5_LE0/COMBOUT (c1000_0029a)	0.00	6.84 r	(4)
GR4_GC0_L5_LE2/LUTC (c1000_0381a)	0.00	6.84 r	(4)
GR4_GC0_L5_LE2/um4/ltb (lt03b)	0.40 *	7.24 r	(4)
GR4_GC0_L5_LE2/um5/cascout (mxcascout)	0.05 *	7.28 r	(4)
GR4_GC0_L5_LE2/um6/dcout (c1110)	0.00 *	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 16–4:

(1) This column does not exist in the actual report. It is included in this document to provide corresponding reference points to Figure 16–12.

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 16–12 illustrates the circuit that the static timing analysis report shows. The buffer tree (buffer) is shown as a single cell.

Figure 16–12. 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.

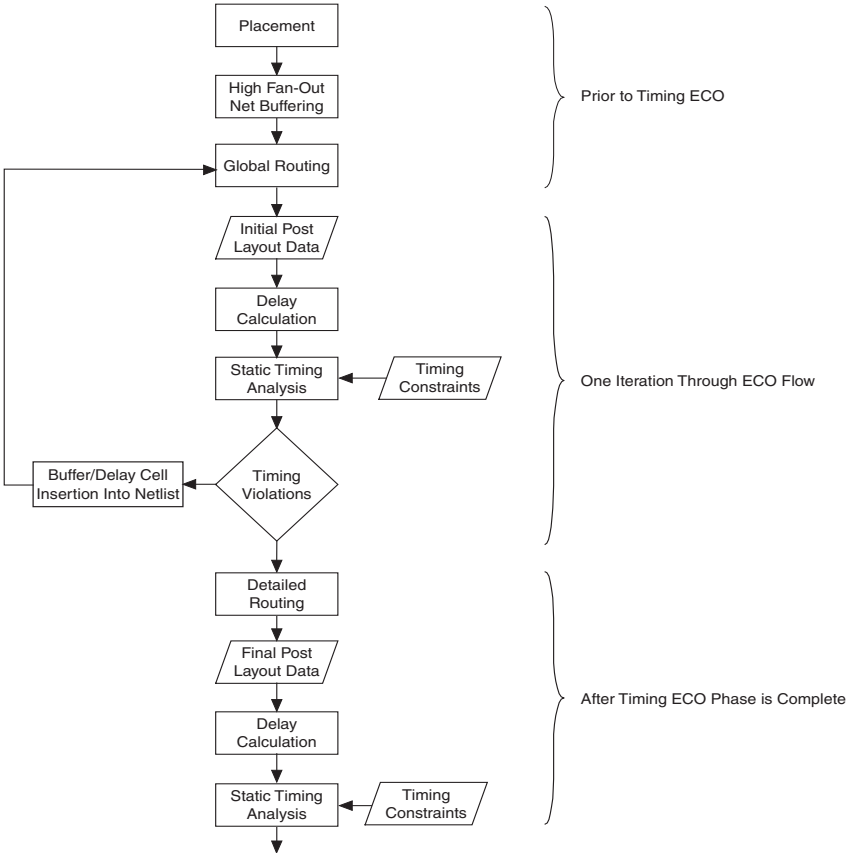
Timing ECOs

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 16–13 shows this flow in more detail.

Figure 16–13. ECO Flow Diagram



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 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.

Section V. HardCopy II Design Considerations

This section provides information on the software support for HardCopy® II devices.

It contains the following chapters:

- Chapter 17, Quartus II Support for HardCopy II Devices
- Chapter 18, Prototyping Strategy for HardCopy II Devices

Revision History

The table below shows the revision history for Chapters 17 and 18.

Chapter(s)	Date / Version	Changes Made
Chapter 17	January 2005 v1.0	Added document to the *HardCopy Series Handbook*.
Chapter 18	January 2005 v1.0	Added document to the *HardCopy Series Handbook*.

17. Quartus II Support for HardCopy II Devices

H51022-1.0

Introduction

The Altera® HardCopy® II device family in 1.2-V, 90-nm process technology provides a powerful 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 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.

The HardCopy II device family is Altera's third generation of HardCopy structured ASICs. Previous families of HardCopy devices include 0.13-µm HardCopy Stratix and 0.18-µm HardCopy APEX™ devices. These families offer low-cost solutions to the Stratix and APEX FPGA devices for volume production.

For more information on HardCopy II, HardCopy Stratix, and HardCopy APEX families, refer to the respective sections for these families 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 of your project.

Quartus II Features for HardCopy II Planning

Quartus II software provides the capability to design your HardCopy II device using a Stratix II device as a prototype. Beginning with version 4.2, Quartus II software contains two features for HardCopy II device planning:

- **HardCopy II Device Resource Guide**—Helps select the best HardCopy II device for migration by comparing the resources required for a design with the resources available in the various HardCopy II devices.
- **Migration Devices** dialog box—Identifies compatible Stratix II and HardCopy II devices for migration. This feature constrains the pins of your Stratix II FPGA prototype to be compatible with your HardCopy II device.

HardCopy II Prototyping Flow

The flow for developing a HardCopy II prototype is very similar to the flow for developing any other design for a Stratix II FPGA. The difference is that you must also specify which HardCopy II device the design eventually migrates to. The flowchart in Figure 17–1 provides an overview highlighting the process for specifying a HardCopy II device.

To prototype your HardCopy II design in the Quartus II software, you must first select the Stratix II device family in the **Device** category of the **Settings** dialog box (Assignments menu).

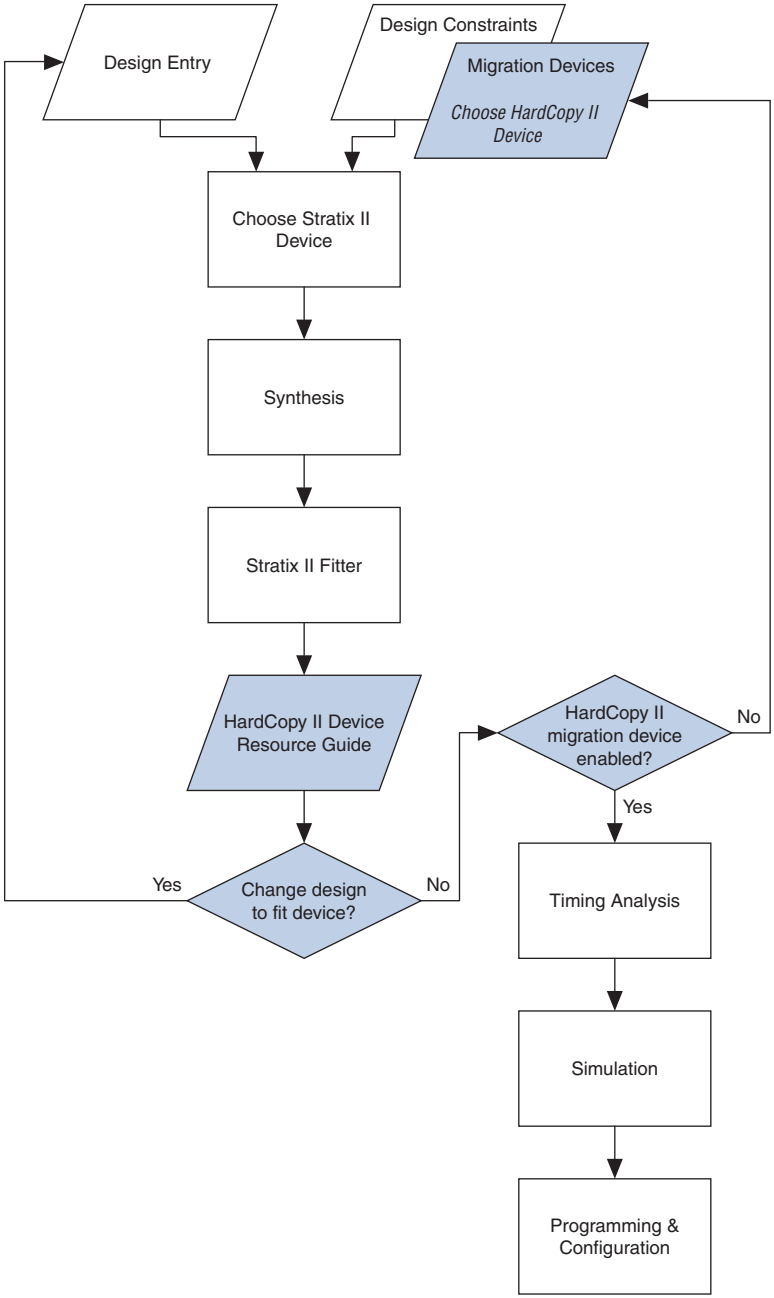
Once you compile your Stratix II design successfully, you can view the HardCopy II Device Resource Guide in the Fitter report to evaluate which HardCopy II devices meet your design's resource requirements. You may see that a small change in the design allows it to fit a smaller, less expensive device. If so, you can make the change and repeat the process. When you are satisfied with the compilation results and the choice of Stratix II and HardCopy II devices, select a HardCopy II device in the **Migration Devices** dialog box and recompile.

From here the flow continues normally with timing analysis and optimization of the design, simulation and verification of the design, and, finally, programming and configuring the Stratix II FPGA.

For more information on the prototyping process for HardCopy II devices, refer to the *Prototyping Strategy for HardCopy II Devices* chapter of the *HardCopy Series Handbook*.

For more information on the overall design flow using Quartus II software, see the *Introduction to Quartus II* manual in the **Products > Literature > Quartus II** section of the Altera web site (**www.altera.com**).

Figure 17–1. HardCopy II Device Prototype Flow



HardCopy II Device Resource Guide

The HardCopy II Device Resource Guide compares the resources required for a successfully compiled design to the resources available in the various HardCopy II devices. The Guide rates each HardCopy II device and each device resource for how well it fits the design. The HardCopy II Device Resource Guide is generated for all designs successfully compiled for Stratix II devices, and is found in the Fitter folder of the Compilation Report. Figure 17–2 shows an example of the HardCopy II Device Resource Guide. The color code is explained in Table 17–1.

Figure 17–2. 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

	Resource	Stratix II EP2S130	HC210W*	HC210	HC220	HC220	HC230	HC240	HC240
1	Migration Compatibility			None	None	None	Medium	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
4	Logic	--		40%	22%	22%	15%	10%	10%
5	-- Logic cells	35695 ALUTs		--	--	--	--	--	--
6	-- DSP elements	0		--	--	--	--	--	--
7	Pins								
8	-- Total	515		515 / 335	515 / 493	515 / 495	515 / 699	515 / 743	515 / 952
9	-- Differential Input	0		0 / 70	0 / 90	0 / 90	0 / 128	0 / 224	0 / 272
10	-- Differential Output	0		0 / 50	0 / 70	0 / 70	0 / 112	0 / 200	0 / 256
11	-- PCI / PCI-X	0		0 / 167	0 / 245	0 / 247	0 / 359	0 / 367	0 / 472
12	-- DQ	0		0 / 20	0 / 44	0 / 44	0 / 178	0 / 178	0 / 178
13	-- DQS	0		0 / 8	0 / 18	0 / 18	0 / 72	0 / 72	0 / 72
14	Memory								
15	-- M-RAM	6		6 / 0	6 / 2	6 / 2	6 / 6	6 / 9	6 / 9
16	-- M4K blocks & M512 blocks	44		44 / 190	44 / 408	44 / 408	44 / 609	44 / 768	44 / 768
17	PLLs								
18	-- Enhanced	2		2 / 2	2 / 2	2 / 2	2 / 4	2 / 4	2 / 4
19	-- Fast	0		0 / 2	0 / 2	0 / 2	0 / 4	0 / 8	0 / 8
20	DLLs	0		0 / 1	0 / 1	0 / 1	0 / 2	0 / 2	0 / 2
21	SERDES								
22	-- RX	0		0 / 19	0 / 31	0 / 31	0 / 44	0 / 92	0 / 116
23	-- TX	0		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
26	-- ASMI	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
28	-- JTAG	0		0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1

* Device is preliminary. Overall performance is expected to be degraded.

Table 17–1. 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 **Migration Devices** 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 **Migration Devices** 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 for Table 17–1:

(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.

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

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 and DSP blocks. The DSP blocks in HardCopy II devices match the functionality, performance, and timing of the Stratix II DSP blocks. The M4K and M-RAM memory blocks in HardCopy II devices are equivalent to the Stratix II memory blocks.

For more information on the HardCopy II device resources, see 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 17–2 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. Since the HC230F1020 device was not specified as a migration target during the compilation, its package and migration compatibility are 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 FineLine BGA HC240 device is rated red because it is only compatible with the Stratix II EP2S180F1020 device. The selection of the HC210 and HC220 devices is further restricted due to the total number of pins used and the number of M-RAM memory blocks required.

To use an HC220F780 device for this example design, reduce the total number of I/O pins used and the number of M-RAM blocks implemented, and change the Stratix II FPGA to a compatible device such as the EP2S130F780 device.

Figure 17–3 shows the report after the (unchanged) design was recompiled with the HardCopy II HC230F1020 device specified as a migration target. Now the HC230F1020 device's package and migration compatibility are rated green or High.

Figure 17–3. 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

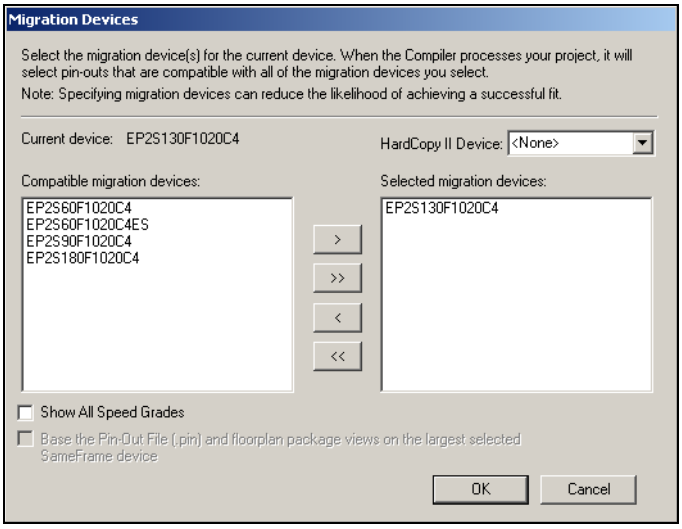
	Resource	Stratix II EP2S130	HC210W*	HC210	HC220	HC220	HC230	HC240	HC240
1	Migration Compatibility			None	None	None	High	None	None
2	Primary Migration Constraint			Package	Package	Package		Package	Package
3	Package	FBGA - 1020	FBGA - 484	FBGA - 484	FBGA - 672	FBGA - 780	FBGA - 1020	FBGA - 1020	FBGA -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

Migration Devices Dialog Box

The **Migration Devices** dialog box is part of the Quartus II design constraints. It constrains the pin assignments for your Stratix II device to also fit your migration targets. Use the **Migration Devices** dialog box to select the target HardCopy II device after you have compiled the Stratix II prototype design and reviewed the HardCopy II Device Resource Guide. The **Migration Devices** dialog box is found in the **Device** category of the **Settings** dialog box (Assignments menu).

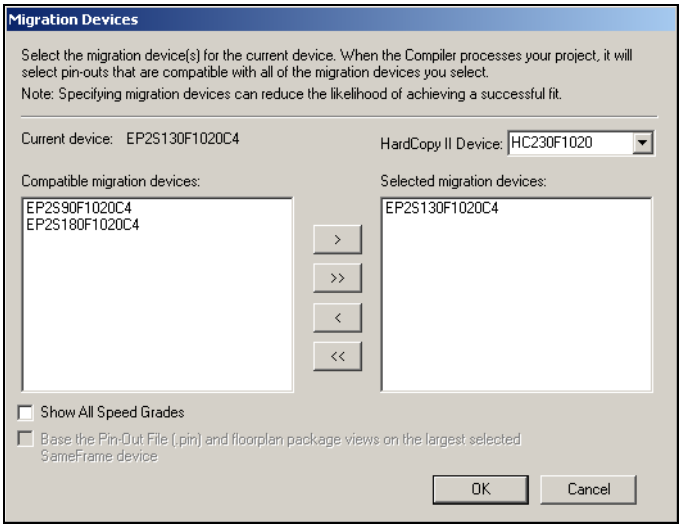
After the first time you compile a Stratix II design with the Quartus II software, the **Migration Devices** dialog box displays only those Stratix II devices that are migration-compatible with your current selected Stratix II device, as shown in Figure 17–4. Select a HardCopy II device from the **HardCopy II Device** list.

Figure 17–4. Migration Devices Dialog Box Showing Compatible Stratix II Devices



Once you have selected the HardCopy II device from the **HardCopy II Device** list, the Quartus II software filters the FPGAs in the **Compatible migration devices** list to show only those FPGA devices that can be used to prototype the selected HardCopy II device, as shown in Figure 17–5. The Quartus II software does not filter this list based on compiled information about your design, so your design may not fit in some of the Stratix II devices listed. You can select more than one FPGA, but only one HardCopy II device, for migration.

Figure 17–5. Migration Devices Dialog Box Showing Selected HardCopy II Device & Compatible Stratix II Devices



For example, a design compiled for a Stratix II EP2S130F1020 device with vertical package migration selected can migrate to an EP2S60F1020, EP2S90F1020, or EP2S180F1020 device. As a prototype, the design can migrate to the HardCopy II HC230F1020 device, which is selected from the **HardCopy II Device** list, as shown in Figure 17–5. After you have selected a HardCopy II device, the **Compatible migration devices** list is restricted to the Stratix II devices supported by the HardCopy II device. After the HC230F1020 device is selected, the only allowable migration devices are the EP2S90F1020, EP2S180F1020, and the (already selected) EP2S130F1020 devices.

Once you have targeted a HardCopy II device and selected any additional Stratix II devices for prototyping from the **Compatible Migration Devices** list in the **Migration Devices** dialog box, re-compile your design and proceed with your prototype optimization and verification.

For more information on the steps to migrate your design from your Stratix II FPGA prototype to a HardCopy II device, contact your Altera Product Field Applications Engineer.

Conclusion

Beginning with version 4.2, Altera's Quartus II software allows you to design for HardCopy II devices and to develop prototypes using Stratix II FPGAs. This is done using the standard development process with the addition of the HardCopy II Device Resource Guide and the **Migration Devices** dialog box. After successfully compiling a design, the HardCopy II Device Resource Guide helps you select a suitable HardCopy II device by comparing the resources required by the design to the resources available in the various HardCopy II devices. The **Migration Devices** dialog box is then used to specify the HardCopy II device and constrain the pin assignments to be compatible with both the HardCopy II device and the Stratix II prototype. You can then recompile and continue with the usual optimization and verification processes. The resulting Stratix II FPGA is a fully functional prototype for the HardCopy II device.

18. Prototyping Strategy for HardCopy II Devices

H51023-1.0

Introduction

Altera® HardCopy® II devices, manufactured using 1.2-V, 90-nm process technology, are the newest additions to the HardCopy series of structured ASICs and provide a powerful alternative to standard-cell ASICs. HardCopy II devices are the industry's lowest-risk structured ASICs and offer high logic density at a low unit cost, while requiring minimal up-front investment and development time. HardCopy II devices also serve as a low-cost migration device for users of Altera's Stratix® II FPGAs.

Beginning with Quartus® II software version 4.2, you can choose a HardCopy II structured ASIC to use for volume production and prototype a design with a Stratix II FPGA. Prototyping with Stratix II FPGA devices today lets you do early system evaluation while simultaneously planning for eventual migration to HardCopy II devices to reduce unit cost for volume production.

HardCopy II Device Prototyping Benefits

The HardCopy II family builds upon the success of the 0.13-µm HardCopy Stratix and 0.18-µm HardCopy APEX™ devices. The HardCopy II family of devices offers the lowest risk and fastest time-to-market development for structured ASICs by combining the FPGA prototyping and production development into a seamless migration process. Altera's proven seamless migration methodology guarantees first-silicon success by preserving the proven netlist while developing the HardCopy II structured ASIC, thereby minimizing design risk.

Prototyping your HardCopy II design using a Stratix II FPGA reduces the total design risk by addressing key decision areas:

- Verification—Functional verification, system development, and lab testing of the prototype guarantee first-silicon success with the HardCopy II device.
- Time to market—Prototyping in a Stratix II FPGA allows you to do early field trials and gain early market entry with your product. You can evaluate demand uncertainties in the market that might make a standard-cell ASIC too risky and expensive to develop.
- Total cost of ownership—Using HardCopy II devices reduces the total design cost when factoring in recurring engineering expenses for long ASIC design cycles, as well as the increasing infrastructure costs for EDA software, computing hardware, and ASIC vendor fees.

Prototyping and subsequent production with HardCopy II greatly reduces the risk of needing a re-spin of your design for functional bugs.

The Quartus II software integrates with popular EDA industry tools used today by ASIC designers, such as Synopsys DC FPGA, PrimeTime, and VCS; Synplicity Synplify; Mentor Graphics® ModelSim®; and Cadence NC-Verilog. So additional investment in software is minimal.

For more information on HardCopy Stratix and HardCopy APEX families, refer to their respective sections in the *HardCopy Series Handbook*.

HardCopy II Prototyping Options

The FPGA prototyping strategy allows you to verify your hardware description language (HDL) code, software for embedded processors, and system integration by having a working silicon prototype of your planned HardCopy II device. The variety of Stratix II devices that are available offers flexibility in choosing which FPGA to use as the prototype for your HardCopy II design.

There are multiple prototyping options possible using Stratix II and HardCopy II devices. The final logic resource, memory, and pin usage of the HardCopy II design are key determinants for selecting the appropriate Stratix II FPGA for prototyping. Table 18–1 shows the suitable migration paths between Stratix II and HardCopy II devices.

For example, referring to Table 18–1, you can see that a design estimated to contain 2 million ASIC gates, 6 Mbits of memory, and 500 user I/O pins would best fit into the HC230F1020 device. The design can be prototyped in the EP2S90F1020, EP2S130F1020, or EP2S180F1020 devices.

For more detailed information on the HardCopy II device family resources and architecture, see the *HardCopy II Device Family Data Sheet* in the *HardCopy Series Handbook*.

Table 18–1. HardCopy II Device Prototyping Options

HardCopy II Device	Package	Total Usable Logic Gates (Millions) *(1)*	Total Memory (Bits)	User I/O Pins *(2)*, *(3)*	Stratix II FPGA Prototype Options
HC210WF484	484-pin FineLine BGA® (wire-bond)	1.0	875,520	*(4)*	EP2S30F484 EP2S60F484 EP2S90H484 *(5)*
HC210F484	484-pin FineLine BGA	1.0	875,520	335	EP2S30F484 EP2S60F484 EP2S90H484 *(5)*
HC220F672	672-pin FineLine BGA	1.6	3,059,712	493	EP2S60F672
HC220F780	780-pin FineLine BGA	1.6	3,059,712	495	EP2S90F780 EP2S130F780
HC230F1020	1,020-pin FineLine BGA	2.2	6,345,216	699	EP2S90F1020 EP2S130F1020 EP2S180F1020
HC240F1020	1,020-pin FineLine BGA	2.2	8,847,360	743	EP2S180F1020
HC240F1508	1,508-pin FineLine BGA	2.2	8,847,360	952	EP2S180F1508

Notes to Table 18–1:

(1) 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.
(2) Pin counts are preliminary.
(3) Includes 8 dedicated clock pins for the HC210W484, HC210F484, HC220F672, and HC220F780 devices; 12 for the HC230F1020 device; and 16 for the HC240F1020 and HC240F1508 devices. These clock pins can also be used for data inputs.
(4) Contact Altera Applications.
(5) The Stratix II EP2S90H484 uses a 484-pin Hybrid FineLine BGA package with a 27-mm body to accommodate the larger cavity requirement for the die.

Unified Design Methodology

One benefit of using Stratix II devices to prototype HardCopy II devices is that they share the same development flow. The steps to develop a regular Stratix II FPGA are the same as to prototype a HardCopy II device. This unified design methodology saves you the complexity of managing different sets of EDA tools for FPGA and ASIC development.

Beginning with Quartus II software version 4.2, you can design your HardCopy II prototype using a Stratix II device and verify the design in-system to meet functionality specifications. In future releases of Quartus II, you can proceed with the migration of your prototype design to a HardCopy II device.

Prototype Flow Using Stratix II Devices

In order to use HardCopy II structured ASICs, first build your prototype design in a migration-compatible Stratix II device. The selection of which Stratix II and HardCopy II devices to use is design dependent. However not all Stratix II FPGA devices are suitable prototypes for HardCopy II structured ASICs. The Stratix II FPGA devices that are suitable for prototyping your HardCopy II design are shown in Table 18–1 on page 18–3. After a design is compiled for a Stratix II device, the Quartus II Fitter reports the resource utilization of the Stratix II device for comparison to the HardCopy II devices.

The following resources used in your design are some of the key determinants as to which HardCopy II device to target:

- Logic gates (Stratix II adaptive logic modules)
- Memory bit count and block count (M4K and M-RAM blocks)
- Digital signal processing blocks
- Phase-locked loops
- Delay-locked loops
- I/O pins
- I/O buffer types (differential, PCI/PCI-X, DDR, etc.)
- Packaging

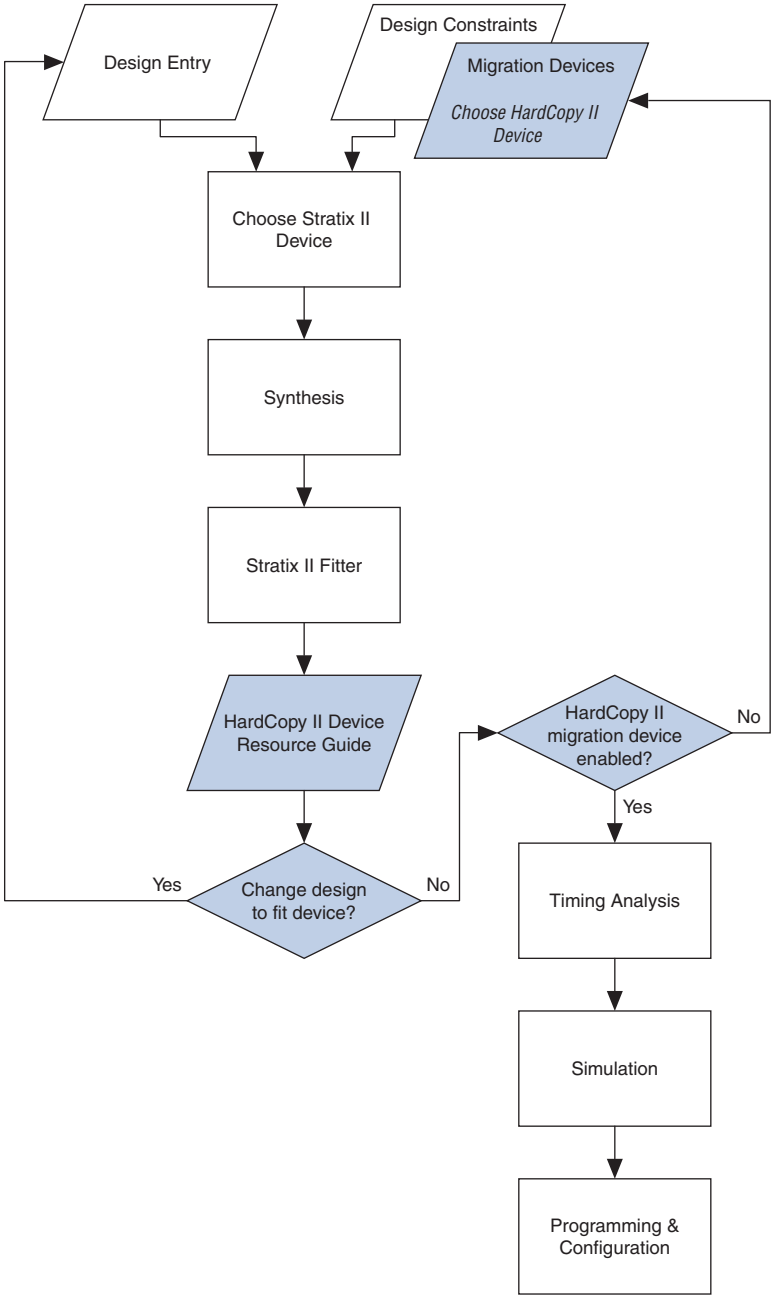
Once you have a general idea how many logic gates, memory bits, and I/O signals are required for your design, you can select a Stratix II FPGA and perform an initial compilation of the design using the Quartus II software. The total usable gates of HardCopy II devices is estimated in terms of ASIC gates, which you should use in your gate count estimates.

Once the Stratix II design is compiled, you can verify the design and evaluate the resources required for your design in order to determine which HardCopy II device to use for production.

The design process for the Stratix II prototype of the HardCopy II device follows the traditional FPGA development flow using the Quartus II software in most respects. Figure 18–1 illustrates the high-level design methodology for prototyping HardCopy II designs using Stratix II FPGAs and the available features in the Quartus II software. Each stage is explained further in the following sections.

For full details of the Quartus II software's features and design flow, see the *Introduction to Quartus II* manual in the **Products > Literature > Quartus II** section of the Altera web site (**www.altera.com**).

Figure 18–1. HardCopy II Device Prototype Flow Using a Stratix II Device



Design Entry for HardCopy II Prototypes

One of the key advantages in the HardCopy II design flow is the ability to re-use HDL code without modification between FPGA and ASIC versions of the design. For most standard-cell ASIC designs, HDL code written for the prototype FPGA needs to be customized for implementation in the ASIC vendor's design flow. The unified design methodology for HardCopy II devices and prototype FPGAs lets you maintain a single version of your HDL that can be used for both the prototype FPGAs and production structured ASICs.

Prototype HardCopy II designs can be developed in VHDL and Verilog HDL languages. In addition to Verilog and VHDL development, schematic entry can be done within the Quartus II software. The HDL coding guidelines used for Stratix II devices apply to HardCopy II devices.

For more information on HDL coding guidelines, see the *Recommended HDL Coding Styles* chapter in Volume 1 of the *Quartus II Handbook*.

Constructing your HDL code for application to the Stratix II architecture and adhering to Altera's recommended HDL coding styles will help ensure a successful design migration to HardCopy II devices.

Design Constraints for HardCopy II Prototypes

In order to achieve high performance and area optimization of your design in the Quartus II software, you must add design constraints. Quartus II software has many features and constraint settings to optimize the design compilation of your prototype. Design constraints such as Logic Lock regions, clock definitions, timing constraints, I/O buffer types, pin assignments, and Fitter optimization allow you to achieve the best performance and smallest area possible in your Stratix II prototype of your HardCopy II device.

For more information on the Quartus II software scripting, constraint entry, I/O planning, area optimization, and timing closure, see Volume 2 of the *Quartus II Handbook*.

Most constraint settings depend on your design and system requirements, but a few settings are required to create a HardCopy II prototype:

- Specify a Stratix II device or the Stratix II family.
- Disable M512 memories.
- Turn on Design Assistant.
- Do not assign pinouts.

Specify a Stratix II Device

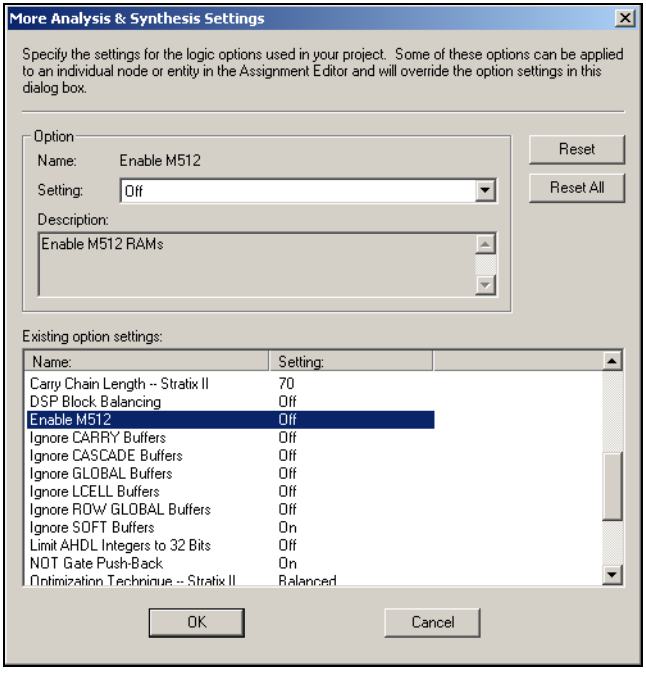
Specify a Stratix II device or, if you are not sure which device to use, specify the Stratix II family. If you specify the family, the Quartus II Fitter selects the smallest Stratix II device that fits your design. To specify a device, open the **Device** category of the **Settings** dialog box (Assignments menu).

Disable M512 Memory Blocks

The HardCopy II device family does not have M512 memory blocks, which are available in Stratix II FPGA devices. In order to insure a compatible prototype designed in a Stratix II device, edit the project settings in the Quartus II software to prevent memory instances in your design from being mapped into M512 memory blocks. Open the **More Analysis & Synthesis Settings** dialog box of the Analysis & Synthesis page of the **Settings** dialog box (Assignments Menu) and change the setting **Enable M512** to **OFF** as shown in Figure 18–2. Or enter the following command in the Tcl console:

```
set_global_assignment -name ENABLE_M512 OFF↵
```

Figure 18–2. Disabling M512 Memories



Design Rule Checking

The Quartus II Design Assistant examines the project for design rule violations that affect the implementation in Altera devices. The Design Assistant rule check can be performed for any FPGA design, but it is required for all HardCopy designs. The current Design Assistant rules apply to HardCopy II designs as well as to HardCopy Stratix and HardCopy APEX designs.

For more information on the design rules checked by the Design Assistant in the Quartus II software, see the *HardCopy Series Design Guidelines* chapter of the *HardCopy Series Handbook* or refer to topics on Design Assistant in the Quartus II Help.

Do Not Assign Pins

Do not assign pinouts, at least initially. The Fitter needs some freedom in assigning the pinouts to make sure the Stratix II and HardCopy II devices are fully drop-in compatible. After you specify a HardCopy II as a migration target and successfully recompile the design, you can experiment with adjusting the pin assignments.

Synthesis for HardCopy II Prototypes

Designing with HardCopy II devices saves you the effort of purchasing additional EDA software. The EDA tools you would use for an FPGA design can also be used for your HardCopy II device prototyping. You can synthesize your VHDL or Verilog HDL code to the Stratix II library using Quartus II Integrated Synthesis or with the approved synthesis software from an Altera Commitment to Cooperative Engineering Solutions (ACCESS) Program partner.

For a list of ACCESS Program® partners, see the **Products > Design Software > EDA Partners** section of the Altera web site (**www.altera.com**).

For more information on using these HDL synthesis options with Altera device families, see the *Synthesis* section in Volume 1 of the *Quartus II Handbook*.

Use the Stratix II library to synthesize your HardCopy II prototype, since your initial prototype will be in a Stratix II FPGA. Synthesis of your HardCopy II prototype device will follow the same synthesis guidelines as for a Stratix II design.

Quartus II Fitter for HardCopy II Prototypes

The HardCopy II prototyping flow does not require purchasing additional place-and-route software. The Quartus II Fitter places and routes your design into the device specified in your Quartus II project. If the Stratix II family is specified, but not a specific Stratix II device, the Quartus II software selects the smallest Stratix II device that your design fits into, and selects the fastest speed grade of the device that meets your performance requirements. If a HardCopy II device is also specified through the **Migration Devices** dialog box, the Fitter also constrains the prototype's pin assignments to be fully compatible with the HardCopy II device.

After the Fitter has run, the Quartus II software generates the HardCopy II Device Resource Guide in the Fitter section of the Compilation Report. This resource guide helps you select which HardCopy II device to use.

HardCopy II Device Resource Guide

The Quartus II software provides information on your compiled design that assists you in selecting which HardCopy II device is suitable for your compiled Stratix II design. The HardCopy II Device Resource Guide is automatically generated after a successful compilation of a Stratix II design. Using this report, you can identify which HardCopy II device meets your design needs. The Device Resource Guide highlights which categories of resources are limiting your choice of HardCopy II devices. Use this guide to make trade-off decisions about your design's functionality and what resources it uses to select the most cost-effective HardCopy II device for production. An example of the HardCopy II Device Resource Guide is shown in Figure 18–3.

Figure 18–3. 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

	Resource	Stratix II EP2S130	HC210W*	HC210	HC220	HC220	HC230	HC240	HC240
1	Migration Compatibility			None	None	None	Medium	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
4	Logic	--		40%	22%	22%	15%	10%	10%
5	-- Logic cells	35695 ALUTs		--	--	--	--	--	--
6	-- DSP elements	0		--	--	--	--	--	--
7	Pins								
8	-- Total	515		515 / 335	515 / 493	515 / 495	515 / 699	515 / 743	515 / 952
9	-- Differential Input	0		0 / 70	0 / 90	0 / 90	0 / 128	0 / 224	0 / 272
10	-- Differential Output	0		0 / 50	0 / 70	0 / 70	0 / 112	0 / 200	0 / 256
11	-- PCI / PCI-X	0		0 / 167	0 / 245	0 / 247	0 / 359	0 / 367	0 / 472
12	-- DQ	0		0 / 20	0 / 44	0 / 44	0 / 178	0 / 178	0 / 178
13	-- DQS	0		0 / 8	0 / 18	0 / 18	0 / 72	0 / 72	0 / 72
14	Memory								
15	-- M-RAM	6		6 / 0	6 / 2	6 / 2	6 / 6	6 / 9	6 / 9
16	-- M4K blocks & M512 blocks	44		44 / 190	44 / 408	44 / 408	44 / 609	44 / 768	44 / 768
17	PLLs								
18	-- Enhanced	2		2 / 2	2 / 2	2 / 2	2 / 4	2 / 4	2 / 4
19	-- Fast	0		0 / 2	0 / 2	0 / 2	0 / 4	0 / 8	0 / 8
20	DLLs	0		0 / 1	0 / 1	0 / 1	0 / 2	0 / 2	0 / 2
21	SERDES								
22	-- RX	0		0 / 19	0 / 31	0 / 31	0 / 44	0 / 92	0 / 116
23	-- TX	0		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
26	-- ASMI	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
28	-- JTAG	0		0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1

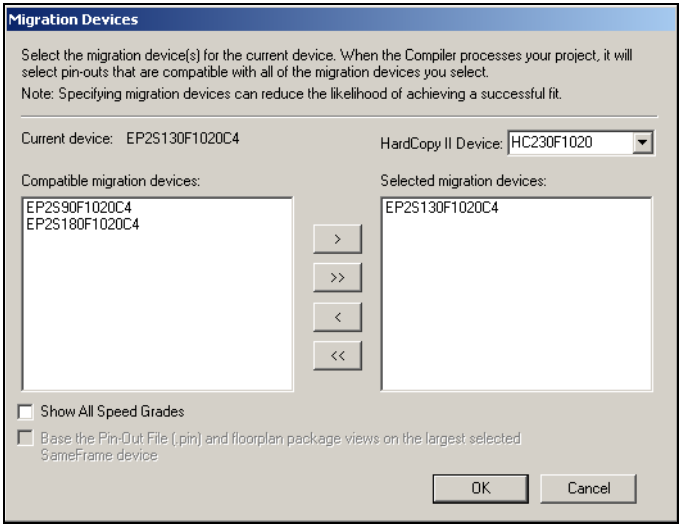
* Device is preliminary. Overall performance is expected to be degraded.

The report example in Figure 18–3 shows the resource estimates for a design compiled for a Stratix II EP2S130F1020 device. Based on the report information, the device selection is restricted due to package types being incompatible with the Stratix II device. The HC210 and HC220 devices are also restricted due to the total number of pins and the number of M-RAM memory blocks required. So the HC230F1020 device in the 1,020-pin FineLine BGA package would be an appropriate HardCopy II device for migration. If you want to target the HC220F780 device for this example design, reduce the total number of I/O pins used and the number of M-RAM blocks implemented, and change the Stratix II device to a compatible form such as the EP2S130F780 device.

Migration Devices

Once you have decided on a HardCopy II device, you need to add it to the design constraints. Use the **Migration Devices** dialog box found in the **Device** category of the **Settings** dialog box (Assignments menu) and shown in Figure 18–4. The constraint settings of the Migration Devices dialog box ensure that the Stratix II prototype is drop-in compatible with the selected HardCopy II device. So no modifications to your system or its circuit boards will be needed to migrate from the Stratix II prototype to the HardCopy II device.

Figure 18–4. Migration Devices Dialog Box



The **Migration Devices** dialog box shows the selected Stratix II device and devices with migration-compatible packages. Select a HardCopy II device and, optionally, any other Stratix II devices you want to try. When you recompile the design, the Fitter constrains the pin assignments of the prototype to be fully compatible with the selected migration devices.

For more information on the HardCopy II Device Resource Guide and the **Migration Devices** dialog box, see the *Quartus II Support for HardCopy II Devices* chapter of the *HardCopy Series Handbook*.

Timing Analysis for HardCopy II Prototypes

Timing analysis is performed by the Quartus II software automatically during a full compilation. No additional EDA tools are required for static timing analysis of your prototype in a Stratix II device, but you can also use Synopsys PrimeTime if you are more comfortable with that static timing analysis software. Timing analysis can be run independently for maximum and minimum operating conditions without requiring a re-compilation in the Quartus II software.

If timing and area results do not meet your requirements, modifying the constraints for synthesis or Fitter compilation may be necessary. Modifying the HDL code may also yield improved results. This is another advantage of the HardCopy II prototyping flow. The optimization you perform for the Stratix II FPGA is also implemented in your HardCopy II device because of the unified design methodology using the Quartus II software. All of your HDL and synthesis changes are maintained because you only need to maintain one version of your design to use in both FPGA prototype and ASIC production.

If timing meets your requirements, the Stratix II implementation of your prototype design can be used in a chip-level environment.

For more information on the Quartus II Timing Analyzer and Synopsys PrimeTime support, see the *Timing Analysis* section in Volume 3 of the *Quartus II Handbook*.

Simulation for HardCopy II Prototypes

Prototype verification often requires running functional simulation of RTL code in conjunction with in-system debugging and software development. The Quartus II EDA Netlist Writer writes functional simulation netlists for use in several popular third-party simulation platforms, including Mentor Graphics ModelSim, Synopsys VCS, and Cadence NC-Verilog.

For more information on using third-party simulation software for Altera devices, see the *Simulation* section in Volume 3 of the *Quartus II Handbook*.

In-System Verification for HardCopy II Prototypes

Simulation of multi-million gate ASIC designs can take a long time to run. The HardCopy II prototyping strategy is a time-saving solution for this. You can prototype your design, and verify it and your software in-system. This saves you time in the design cycle by allowing you to make

fast regression tests that can be verified in the development lab. The Quartus II Assembler generates programming files in several different formats for use in prototyping evaluation and in-system verification.

On-chip debugging of your prototype can be done using several Altera features including the SignalProbe™ feature, and the SignalTap® II Embedded Logic Analyzer. Debugging by engineering change order can be done through the use of Chip Editor. Incremental compilation and in-system updating of memory and constants help save time between design iterations. These features allow for interactive debugging of the Stratix II prototype of your HardCopy II device faster than simulation testing.

Changes made through on-chip debugging will not migrate to the HardCopy II device. When you finish debugging, make the changes in the RTL source code and recompile the design. Then repeat your in-system verification.

For more information about using on-chip debugging capabilities with Stratix II devices, see the *On-Chip Debugging* section in Volume 3 of the *Quartus II Handbook*.

Migrating to a HardCopy II Device

When the design is finalized and you are ready to begin the development of your HardCopy II design, consult your Altera representative for information on the next steps to migrate your design from your Stratix II prototype to a HardCopy II structured ASIC.

Conclusion

You can begin prototyping your HardCopy II design today in a Stratix II device. The ability to prototype your HardCopy II structured ASIC design in a Stratix II device gives you the fastest development cycle in the structured ASIC market. Altera's unified design methodology for HardCopy II devices preserves changes made to the prototype implemented in a Stratix II device, so no optimization work is lost when you begin the production of your HardCopy II design. You can verify the design in-system, as well as allowing software development to progress in parallel with the verification effort. This rapid verification allows you to gain early market entry with the Stratix II prototype while you migrate the design to a HardCopy II device. The HardCopy II devices provide a low-cost and low-risk ASIC solution, minimizing the total cost of design as compared to a standard-cell ASIC. HardCopy II devices also provide guaranteed functional equivalence between your prototype Stratix II FPGA and production HardCopy II structured ASIC device, reducing the risk of a costly design re-spin.

Section VI. HardCopy Stratix Design Considerations

This section provides information on the software support for HardCopy® Stratix® devices.

It contains the following:

- Chapter 19, Quartus II Support for HardCopy Stratix Devices

Revision History

The table below shows the revision history for Chapter 19.

Chapter(s)	Date / Version	Changes Made
Chapter 19	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 19, Quartus II Support for HardCopy Stratix Devices.

19. Quartus II Support for HardCopy Stratix Devices

H51014-2.0

Introduction

The 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 or 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 and power 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

The Quartus II software version 4.2 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**
  The Quartus II software version 4.2 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 (**.qar**) file 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

- **HardCopy Stratix Power Estimator**
  This power estimation tool is launched from the Quartus II software to estimate power consumed by the HardCopy Stratix devices.

The HardCopy APEX Power Estimator is available on the Altera web site (**www.altera.com**) in the **Products > Devices > HardCopy APEX > Design Utilities > HardCopy APEX Power Estimator**.

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 19–1 compares HARDCOPY_FPGA_PROTOTYPE devices, Stratix devices, and HardCopy Stratix devices.

Table 19–1. Qualitative Comparison of HARDCOPY_FPGA_PROTOTYPE to Stratix & HardCopy Stratix Devices

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
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 19–2 lists the resources available in each of the HardCopy Stratix devices.

Table 19–2. 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 19–2:

(1) Combinational and registered logic do not include digital signal processing (DSP) blocks, on-chip RAM, or phase-locked loop (PLL)s.

(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, see 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.

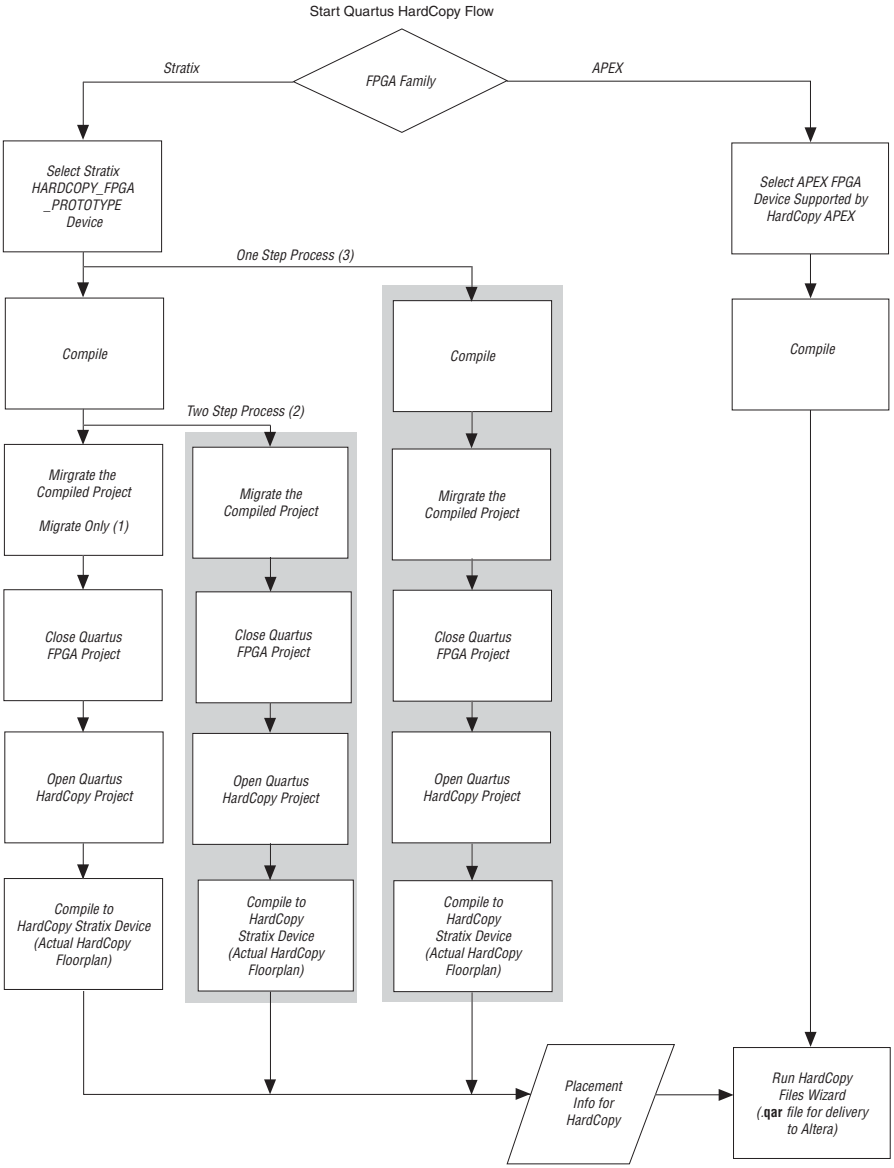
For more information on the **.sof** file and programming Stratix FPGA devices, see the *Programming and Configuration* chapter of the *Introduction to Quartus II Manual*.

HardCopy Design Flow

Figure 19–1 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, see "HardCopy Timing Optimization Wizard Summary Page" on page 19–12 and "Generating the HardCopy Design Database" on page 19–22.

Figure 19–1. HardCopy Stratix & HardCopy APEX Design Flow Diagram



Notes for Figure 19–1:
(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), as shown in Figure 19–1.

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 your design 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 for 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.

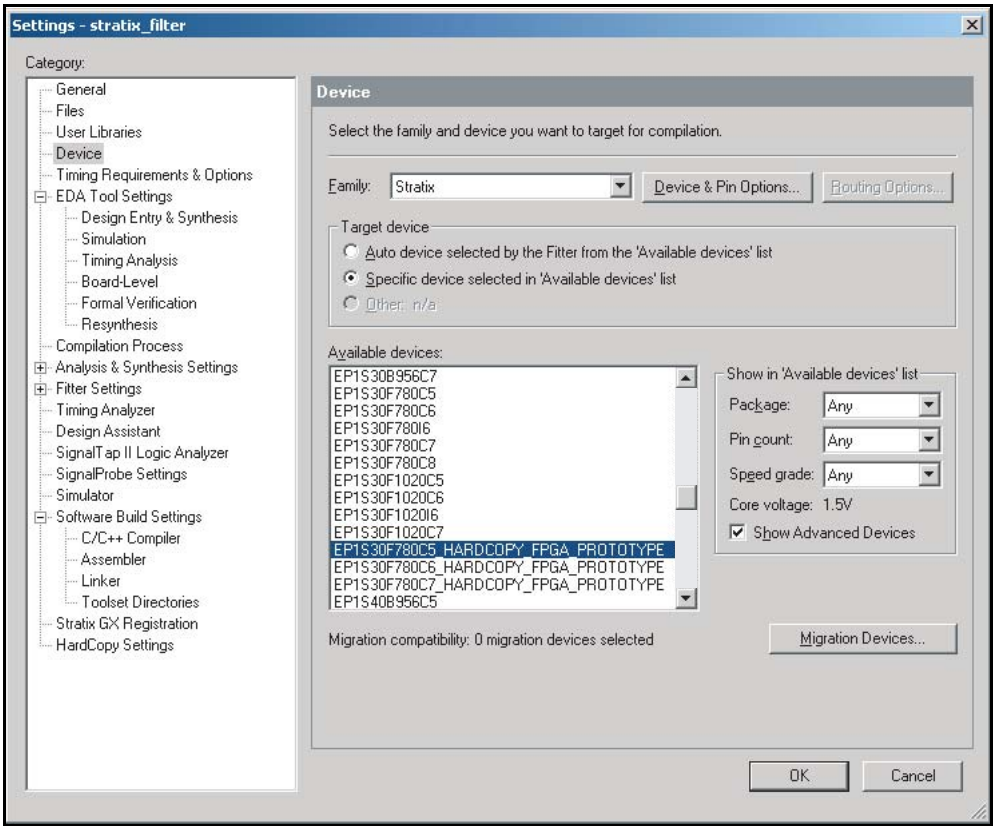
To target a design to a HardCopy Stratix device in the Quartus II software, follow these steps:

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 19–2 on page 19–9.

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 19–2. 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 19–3.

- **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. See "Performance Estimation" on page 19–14 if you need 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

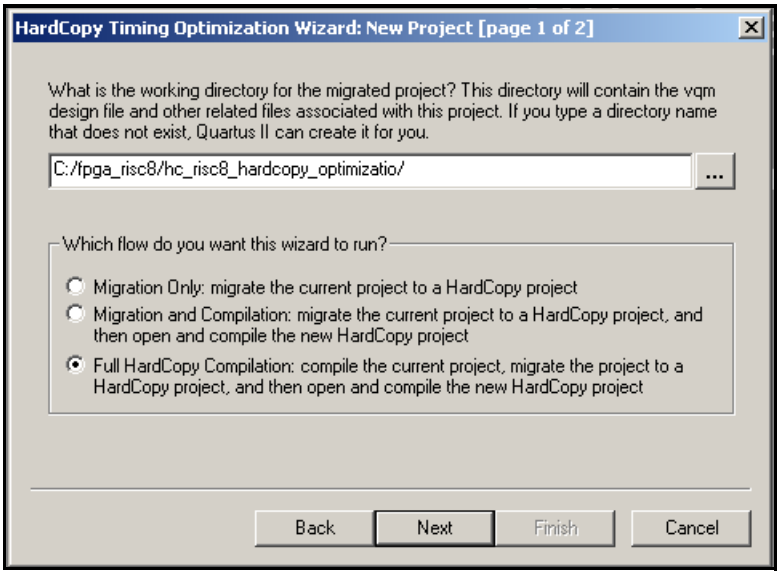
- **Migration and Compilation**: You can select this option after compiling the project. This option results in the following actions:

  - 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

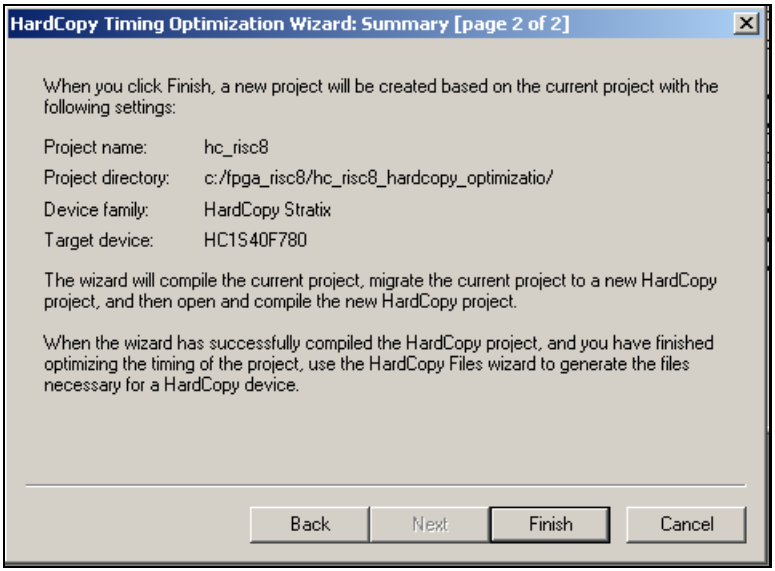
Figure 19–3. HardCopy Timing Optimization Wizard Options



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, as shown in Figure 19–4.

Figure 19–4. HardCopy Timing Optimization Wizard Summary Page



When either of the second two options in Figure 19–3 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, see "Performance Estimation" on page 19–14. 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

The Quartus II software version 4.2 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 see the available resources, and the location of the resources in the actual device.

Performance Estimation

Figure 19–5 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:

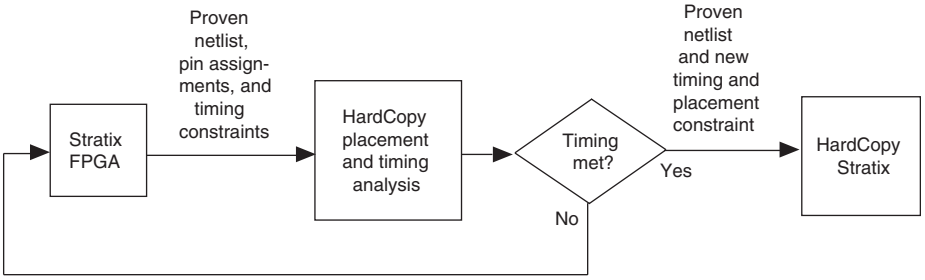
- Work to improve LAB placement in the HardCopy Stratix project.

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 19–5. 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_PROTOTYPE 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 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 timing analysis 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 19–3.

Table 19–3. 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.

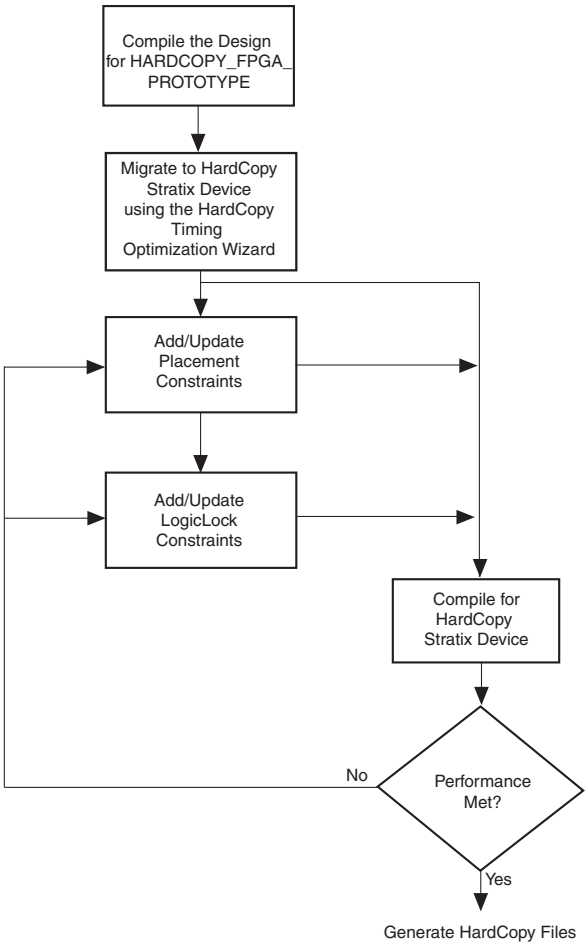
Buffer Insertion

The Quartus II software version 4.2 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 **.qar** file. This buffer insertion improves the estimation of the Quartus II Timing Analyzer for the HardCopy Stratix device.

Placement Constraints

The Quartus II software version 4.2 supports placement constraints and LogicLock regions for HardCopy Stratix devices. Figure 19–6 shows an iterative process to modify the placement constraints until the best placement for the HardCopy Stratix device is achieved.

Figure 19–6. 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, see 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, see 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 at least 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, see 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 **.qar** file is generated at the same directory level as the targeted project, either before or after optimization. Table 19–4 shows the archive directory structure and files collected by the HardCopy Files Wizard.

Table 19–4. 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, see the *Quartus II Timing Analysis* and the *Synopsys PrimeTime Support* chapters in Volume 3 of the *Quartus II Handbook*.

Power Estimation

The Quartus II software has built-in capability for estimating HardCopy Stratix device power consumption by evaluating the following design components:

- Target device and package
- Temperature grade
- Clock domain f_{MAX}
- Device resources used

HardCopy Stratix Power Estimator

The HardCopy Stratix 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, see the *Power Estimation & Analysis* Section in Volume 3 of the *Quartus II Handbook*.

Opening the HardCopy Stratix Power Estimator

The HardCopy Stratix Power Estimator page on the Altera web site is opened in the Quartus II software. The Quartus II software automatically fills in the necessary information when you open the page. You can modify the values and estimate the estimated power consumed under various conditions.

The estimator can also be opened independently of the Quartus II software on the Altera web site (**www.altera.com**) by clicking on **Products > Devices**. Select **HardCopy Stratix**, then click on the **Power Estimator** link.

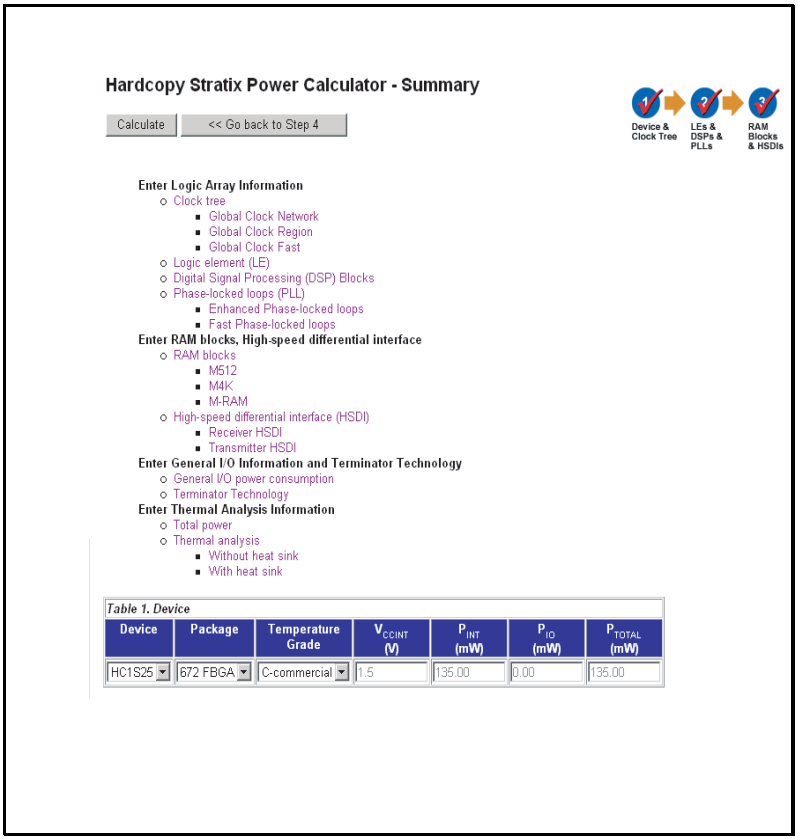
You must enter design-specific information manually if you run the estimator directly.

To estimate HardCopy Stratix power consumption, follow these steps:

1. After compiling the design for a HardCopy Stratix device, choose **HardCopy Utilities > HardCopy Power Estimation** (Project menu), and click **OK**.

   The Quartus II software exports all the necessary data and displays the Power Estimator, a section of which is shown in Figure 19–7.

Figure 19–7. HardCopy Stratix Power Calculator



2. Enter values for the following variables in the spreadsheet and click **Calculate** to get the total power (P_{TOTAL}).

   - Average number of logic elements
   - Average capacitive load
   - DC output power
   - Ambient temperature

For more information on power estimation, see the Quartus II Help.

The HardCopy Stratix Power Estimator is run from the Quartus II software when the target is still HARDCOPY_FPGA _PROTOTYPE device. However, power is estimated for the HardCopy Stratix device, not for the FPGA.

Use the Stratix FPGA Power Estimator to estimate power consumption for the HARDCOPY_FPGA_PROTOTYPE.

On average, HardCopy Stratix devices are expected to consume 40% less power than the equivalent FPGA.

HardCopy APEX Power Estimation

The HardCopy APEX Power Estimator is also a web-based estimator that can be run from the Altera web site (**www.altera.com**) in the **Products > Devices > HardCopy APEX > Design Utilities > HardCopy APEX Power Estimator**. You cannot open this feature in the Quartus II software.

With the HardCopy APEX Power Estimator, you can estimate the power consumed by HardCopy APEX devices and design systems with the appropriate power budget.

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, see the *Tcl Scripting* chapter in Volume 2 of the *Quartus II Handbook*.

Targeting Designs to HardCopy APEX Devices

The Quartus II software version 4.2 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 19–5 shows the files collected for HardCopy APEX by the HardCopy Files Wizard.

Table 19–5. 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**

See "Generating the HardCopy Design Database" on page 19–22 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 of 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.

Related Documents

For more information, see 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.*

Index

E

G

H

I

L

M

N

P

Q

R

S

T

U

V