Section I. HardCopy Stratix Device Family Data Sheet

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

This section contains the following:

- Chapter 1, Introduction to HardCopy Stratix Devices
- Chapter 2, Description, Architecture, and Features
- Chapter 3, Boundary-Scan Support
- Chapter 4, Operating Conditions
- Chapter 5, Quartus II Support for HardCopy Stratix Devices
- Chapter 6, Design Guidelines for HardCopy Stratix Performance Improvement

Revision History

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

1. Introduction to HardCopy Stratix Devices

H51001-2.4

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 up to 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 1–1 provides the details of the HardCopy Stratix devices.

Table 1–1. HardCopy Stratix Devices and 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 1–1:
(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℠) megafunctions
- Available in space-saving flip-chip FineLine BGA® and wire-bond packages (Tables 1–2 and 1–3)
- 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 1–2. HardCopy Stratix Device Package Options and 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			782
HC1S80			782

Notes to Table 1–2:

(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 1–3. 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

Document Revision History

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

Table 1–4. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
September 2008 v2.4	Revised chapter number and metadata.	—
June 2007 v2.3	Updated Introduction section. Updated Table 1–2.	—
December 2006 v2.2	Updated revision history.	—
March 2006	Formerly chapter 5; no content change.	—
October 2005 v2.1	Minor edits	—
January 2005 v2.0	Minor edits	—
June 2003 v1.0	Initial release of Chapter 5, *Introduction to HardCopy Stratix Devices*, in the *HardCopy Device Handbook*.	

2. Description, Architecture, and Features

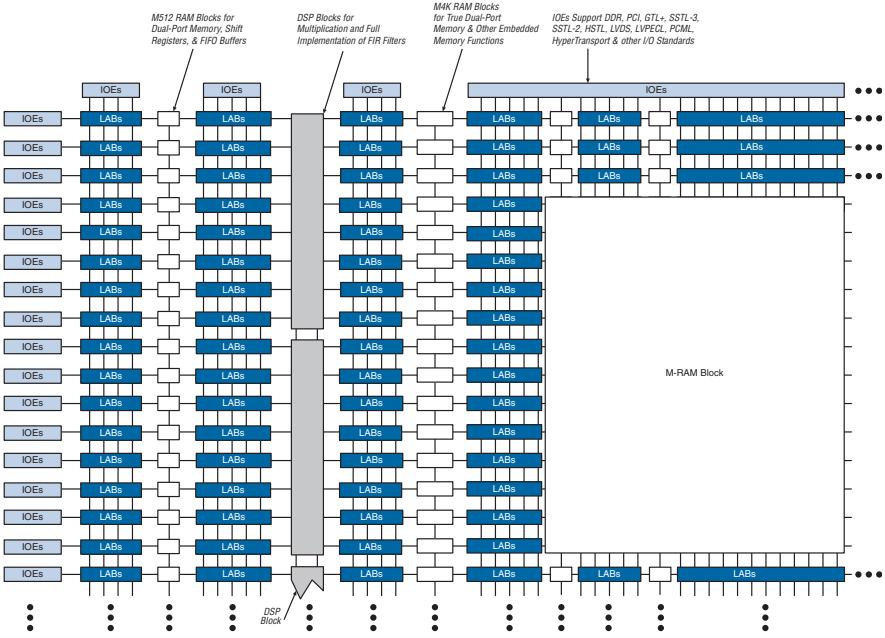
H51002-3.4

Introduction

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 2–1, 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 2–1. HardCopy Stratix Device Architecture



The HardCopy Stratix family consists of base arrays that are common to all designs for a particular device density. Design-specific customization is done within the top two metal layers. The base arrays use an area-efficient sea-of-logic-elements (SOLE) core and extend the flexibility of high-density Stratix FPGAs to a cost-effective, high-volume production solution. With a seamless migration process employed in numerous successful designs, functionality-verified Stratix FPGA designs can be migrated to fixed-function HardCopy Stratix devices with minimal risk and guaranteed first-time success.

The SRAM configuration cells of the original Stratix devices are replaced in HardCopy Stratix devices by metal connects, which define the function of each logic element (LE), digital signal processing (DSP) block, phase-locked loop (PLL), embedded memory, and I/O cell in the device. These resources are interconnected using metallization layers. Once a HardCopy Stratix device has been manufactured, the functionality of the device is fixed and no re-programming is possible. However, as is the case with Stratix FPGAs, the PLLs can be dynamically configured in HardCopy Stratix devices.

HardCopy Stratix and Stratix FPGA Differences

To ensure HardCopy Stratix device functionality and performance, designers should thoroughly test the original Stratix FPGA-based design for satisfactory results before committing the design for migration to a HardCopy Stratix device. Unlike Stratix FPGAs, HardCopy Stratix devices are customized at the time of manufacturing and therefore do not have programmability support.

Since HardCopy Stratix devices are customized within the top two metal layers, no configuration circuitry is required. Refer to "Power-Up Modes in HardCopy Stratix Devices" on page 2–7 for more information.

Depending on the design, HardCopy Stratix devices can provide, on average, a 50% performance improvement over equivalent Stratix FPGAs. The performance improvement is achieved by die size reduction, metal interconnect optimization, and customized signal buffering. HardCopy Stratix devices consume, on average, 40% less power than their equivalent Stratix FPGAs.

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

Table 2–1 illustrates the differences between HardCopy Stratix and Stratix devices.

Table 2–1. HardCopy Stratix and Stratix Device Comparison (Part 1 of 2)

HardCopy Stratix	Stratix
Customized device. All reprogrammability support is removed and no configuration is required.	Re-programmable with configuration is required upon power-up.
Average of 50% performance improvement over corresponding FPGA *(1)*.	High-performance FPGA.
Average of 40% less power consumption compared to corresponding FPGA *(1)*.	Standard FPGA power consumption.
Contact Altera for information regarding specific IP support.	IP support for all devices is available.
Double data rate (DDR) SDRAM maximum operating frequency is pending characterization.	DDR SDRAM can operate at 200 MHz for -5 speed grade devices.
All routing connections are direct and all unused routing is removed.	MultiTrack™ routing stitches together routing resources to provide a path.
HC1S30 and HC1S40 devices have two M-RAM blocks. HC1S80 devices have six M-RAM blocks.	EP1S30 and EP1S40 devices have four M-RAM blocks. EP1S80 devices have nine M-RAM blocks.
It is not possible to initialize M512 and M4K RAM contents during power-up.	The contents of M512 and M4K RAM blocks can be preloaded during configuration with data specified in a memory initialization file (**.mif**).
The contents of memory output registers are unknown after power-on reset (POR).	The contents of memory output registers are initialized to '0' after POR.
HC1S30 and HC1S40 devices have six PLLs.	HC1S30 devices have 10 PLLs. HC1S40 devices have 12 PLLs.
PLL dynamic reconfiguration uses ROM for information. This reconfiguration is performed in the back-end and does not affect the migration flow.	PLL dynamic reconfiguration uses a MIF to initialize a RAM resource with information.
The I/O elements (IOEs) are equivalent but not identical to FPGA IOEs due to slight design optimizations for HardCopy devices.	The IOEs are optimized for the FPGA architecture.

Table 2–1. HardCopy Stratix and Stratix Device Comparison (Part 2 of 2)

HardCopy Stratix	Stratix
The I/O drive strength for single-ended I/O pins are slightly different and is modeled in the HardCopy Stratix IBIS models.	The I/O drive strength for single-ended I/O pins are found in Stratix IBIS models.
In the HC1S40 780-pin FineLine BGA® device, the I/O pins U12 and U18 must be connected to ground.	In the HC1S40 780-pin FineLine BGA device, the I/O pins U12 and U18 are available as general-purpose I/O pins.
The BSDL file describes re-ordered Joint Test Action Group (JTAG) boundary-scan chains.	The JTAG boundary-scan chain is defined in the BSDL file.

Note to Table 2–1:

(1) Performance and power consumption are design dependant.

Logic Elements

Logic is implemented in HardCopy Stratix devices using the same architectural units as the Stratix device family. The basic unit is the logic element (LE) with logic array blocks (LAB) consisting of 10 LEs. The implementation of LEs and LABs is identical to the Stratix device family.

In the HardCopy Stratix device family, all extraneous routing resources not essential to the specific design are removed for performance and die size efficiency. Therefore, the MultiTrack interconnect for routing implementation between LABs and other device resources in the Stratix device family is no longer necessary in the HardCopy Stratix device family.

Table 2–2 illustrates the differences between HardCopy Stratix and Stratix logic.

Table 2–2. HardCopy Stratix and Stratix Logic Comparison

HardCopy Stratix	Stratix
All routing connections are direct and all unused routing is removed.	MultiTrack routing stitches routing resources together to provide a path.

Embedded Memory

TriMatrix™ memory blocks from Stratix devices, including M512, M4K, and M-RAM memory blocks, are available in HardCopy Stratix devices. Embedded memory is seamlessly implemented in the equivalent resource.

Although memory resource implementation is equivalent, the number of specific M-RAM blocks are not necessarily the same between corresponding Stratix and HardCopy Stratix devices. Table 2–3 shows the number of M-RAM blocks available in each device.

Table 2–3. HardCopy Stratix and Stratix M-RAM Block Comparison

HardCopy Stratix		Stratix	
Device	**M-RAM Blocks**	**Device**	**M-RAM Blocks**
HC1S25	2	EP1S25	2
HC1S30	2	EP1S30	4
HC1S40	2	EP1S40	4
HC1S60	6	EP1S60	6
HC1S830	6	EP1S830	9

In HardCopy Stratix devices, it is not possible to preload RAM contents using a MIF after powering up; the output registers of memory blocks will have unknown values. This occurs because there is no configuration process that is executed.

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

Table 2–4 illustrates the differences between HardCopy Stratix and Stratix memory.

Table 2–4. HardCopy Stratix and Stratix Memory Comparison

HardCopy Stratix	Stratix
HC1S30 and HC1S40 devices have two M-RAM blocks. HC1S80 devices have six M-RAM blocks.	EP1S30 and EP1S40 devices have four M-RAM blocks. EP1S80 devices have nine M-RAM blocks.
It is not possible to initialize M512 and M4k RAM contents during power-up.	The contents of M512 and M4K RAM blocks can be preloaded during configuration with data specified in a MIF.
The contents of memory output registers are unknown after POR.	The contents of memory output registers are initialized to ‘0’ after POR.

DSP Blocks

DSP blocks in HardCopy Stratix devices are architecturally identical to those in Stratix devices. The number of DSP blocks available in HardCopy Stratix devices matches the number of DSP blocks available in the corresponding Stratix device.

PLLs and Clock Networks

The PLLs in HardCopy Stratix devices are identical to those in Stratix devices. The clock networks are also implemented exactly as they are in Stratix devices. The number of PLLs can vary between corresponding Stratix and HardCopy Stratix devices. Table 2–5 shows the number of PLLs available in each device.

Table 2–5. HardCopy Stratix and Stratix PLL Comparison

HardCopy Stratix		Stratix	
Device	**PLLs**	**Device**	**PLLs**
HC1S25	6	EP1S25	6
HC1S30	6	EP1S30	10
HC1S40	6	EP1S40	12
HC1S60	12	EP1S60	12
EP1S830	12	EP1S830	12

Table 2–6 illustrates the differences between HardCopy Stratix and Stratix PLLs.

Table 2–6. HardCopy Stratix and Stratix PLL Differences

HardCopy Stratix	Stratix
HC1S30 and HC1S40 devices have six PLLs.	HC1S30 devices have 10 PLLs. HC1S40 devices have12 PLLs.
PLL dynamic reconfiguration uses ROM for information. This reconfiguration is performed in the back-end and does not affect the migration flow.	PLL dynamic reconfiguration uses a MIF to initialize a RAM resource with information.

I/O Structure and Features

The HardCopy Stratix IOEs are equivalent, but not identical to, the Stratix FPGA IOEs. This is due to the reduced die size, layout difference, and metal customization of the HardCopy Stratix device. The differences are minor but may be relevant to customers designing with tight DC and switching characteristics. However, no signal integrity concerns are introduced with HardCopy Stratix IOEs.

When designing with very tight timing constraints (for example, DDR or quad data rate [QDR]), or if using the programmable drive strength option, Altera recommends verifying final drive strength using updated IBIS models located on the Altera website at **www.altera.com**. Differential I/O standards are unaffected.

I/O pin placement and `VREF` pin placement rules are identical between HardCopy Stratix and Stratix devices. Unused pin settings will carry over from Stratix device settings and are implemented as tri-stated outputs driving ground or outputs driving V_{CC}.

In Stratix EP1S40 780-pin FineLine BGA FPGAs, the I/O pins `U12` and `U18` are available as general-purpose I/O pins. In the FPGA prototype, EP1S40F780_HARDCOPY_FPGA_PROTOTYPE, and in the Hardcopy Stratix HC1S40 780-pin FineLine BGA device, the I/O pins `U12` and `U18` must be connected to ground. HC1S40 780-pin FineLine BGA and EP1S40F780_HARDCOPY_FPGA_PROTOTYPE pin-outs are identical.

Table 2–7 illustrates the differences between HardCopy Stratix and Stratix I/O pins.

Table 2–7. HardCopy Stratix and Stratix I/O Pin Comparison

HardCopy Stratix	Stratix
The IOEs are equivalent, but not identical to, the FPGA IOEs due to slight design optimizations for HardCopy devices.	IOEs are optimized for the FPGA architecture.
The I/O drive strength for single-ended I/O pins are slightly different and are found in the HardCopy Stratix IBIS models.	The I/O drive strength for single-ended I/O pins are found in Stratix IBIS models.
In the HC1S40 780-pin FineLine BGA device, the I/O pins `U12` and `U18` must be connected to ground.	In the EP1S40 780-pin FineLine BGA device, the I/O pins `U12` and `U18` are available as general-purpose I/O pins.

Power-Up Modes in HardCopy Stratix Devices

Designers do not need to configure HardCopy Stratix devices, unlike their FPGA counterparts. However, to facilitate seamless migration, configuration can be emulated in HardCopy Stratix devices.

The modes in which a HardCopy Stratix device can be made ready for operation after power-up are: instant on, instant on after 50 ms, and configuration emulation. These modes are briefly described below.

- In instant on mode, the HardCopy Stratix device is available for use shortly after the device receives power. The on-chip POR circuit resets all registers. The CONF_DONE output is tri-stated once the POR has elapsed. No configuration device or configuration data is necessary.
- In instant on after 50 ms mode, the HardCopy Stratix device performs in a fashion similar to the instant on mode, except that there is an additional delay of 50 ms, during which time the device is held in reset stage. The CONF_DONE output is pulled low during this time, and then tri-stated after the 50 ms have elapsed. No configuration device or configuration data is necessary for this option.
- In configuration emulation mode, the HardCopy series device emulates the behavior of an APEX or Stratix FPGA during its configuration phase. When this mode is used, the HardCopy device uses a configuration emulation circuit to receive configuration bit streams. When all the configuration data is received, the HardCopy series device transitions into an initialization phase and releases the CONF_DONE pin to be pulled high. Pulling the CONF_DONE pin high signals that the HardCopy series device is ready for normal operation. If the optional open-drain INIT_DONE output is used, the normal operation is delayed until this signal is released by the HardCopy series device.

HardCopy II and some HardCopy Stratix devices do not support configuration emulation mode.

Instant on and instant on after 50 ms modes are the recommended power-up modes because these modes are similar to an ASIC's functionality upon power-up. No changes to the existing board design or the configuration software are required.

All three modes provide significant benefits to system designers. They enable seamless migration of the design from the FPGA device to the HardCopy device with no changes to the existing board design or the configuration software. The pull-up resistors on *n*CONFIG, *n*STATUS, and CONF_DONE should be left on the printed circuit board.

For more information, refer to the *HardCopy Series Configuration Emulation* chapter in the *HardCopy Series Handbook*.

Hot Socketing

HardCopy Stratix devices support hot socketing without any external components. In a hot socketing situation, a device's output buffers are turned off during system power up or power down. To simplify board design, HardCopy Stratix devices support any power-up or power-down sequence (V_{CCIO} and V_{CCINT}). For mixed-voltage environments, you can

drive signals into the device before or during power up or power down without damaging the device. HardCopy Stratix devices do not drive out until they have attained proper operating conditions.

You can power up or power down the V_{CCIO} and V_{CCINT} pins in any sequence. The power supply ramp rates can range from 100 ns to 100 ms. During hot socketing, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

- The hot socketing DC specification is $| I_{IOPIN} | < 300$ µA.
- The hot socketing AC specification is $| I_{IOPIN} | < 8$ mA for 10 ns or less. This specification takes into account the pin capacitance only. Additional capacitance for trace, connector, and loading needs to be taken into consideration separately. I_{IOPIN} is the current at any user I/O pin on the device.

The DC specification applies when all V_{CC} supplies to the device are stable in the powered-up or powered-down conditions. For the AC specification, the peak current duration due to power-up transients is 10 ns or less.

HARDCOPY_ FPGA_ PROTOTYPE Devices

HARDCOPY_FPGA_PROTOTYPE devices are Stratix FPGAs available for designers to prototype their HardCopy Stratix designs and perform in-system verification before migration to a HardCopy Stratix device. The HARDCOPY_FPGA_PROTOTYPE devices have the same available resources as in the final HardCopy Stratix devices.

The Quartus II software version 4.1 and later contains the latest timing models. For designs with tight timing constraints, Altera strongly recommends compiling the design with the Quartus II software version 4.1 or later. To properly verify I/O features, it is important to design with the HARDCOPY_FPGA_PROTOTYPE device option prior to migrating to a HardCopy Stratix device.

Some HARDCOPY_FPGA_PROTOTYPE devices, as indicated in Table 2–8, have fewer M-RAM blocks compared to the equivalent Stratix FPGAs. The selective removal of these resources provides a significant price benefit to designers using HardCopy Stratix devices.

Table 2–8. M-RAM Block Comparison Between Various Devices

Number of LEs	HARDCOPY_FPGA_PROTOTYPE Devices		HardCopy Stratix Devices		Stratix Devices	
	Device	M-RAM Blocks	Device	M-RAM Blocks	Device	M-RAM Blocks
25,660	EP1S25	2	HC1S25	2	EP1S25	2
32,470	EP1S30	2	HC1S30	2	EP1S30	4
41,250	EP1S40	2	HC1S40	2	EP1S40	4
57,120	EP1S60	6	HC1S60	6	EP1S60	6
79,040	EP1S830	6	HC1S830	6	EP1S830	9

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

HARDCOPY_FPGA_PROTOTYPE FPGA devices have the identical speed grade as the equivalent Stratix FPGAs. However, HardCopy Stratix devices are customized and do not have any speed grading. HardCopy Stratix devices, on an average, can be 50% faster than their equivalent HARDCOPY_FPGA_PROTOTYPE devices. The actual improvement is design-dependent.

Document Revision History

Table 2–9 shows the revision history for this chapter.

Table 2–9. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
September 2008 v3.4	Revised chapter number and metadata.	—
June 2007 v3.3	• Updated Table 2–1. • Added note to the "Embedded Memory" section. • Updated the "Hot Socketing" section.	—

Table 2–9. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
December 2006 v3.2	Updated revision history.	—
March 2006	Formerly chapter 6; no content change.	—
October 2005 v3.1	• Minor edits • Updated graphics	Minor edits.
May 2005 v3.0	• Added Table 6-1 • Added the Logic Elements section • Added the Embedded Memory section • Added the DSP Blocks section • Added the PLLs and Clock Networks section • Added the I/O Structure and Features section	Minor update.
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	Minor update.
August 2003 v1.1	Edited section headings’ hierarchy.	Minor edits.
June 2003 v1.0	Initial release of Chapter 6, Description, Architecture and Features, in the *HardCopy Device Handbook*	—

3. Boundary-Scan Support

H51004-3.4

IEEE Std. 1149.1 (JTAG) Boundary-Scan Support

All HardCopy® Stratix® structured ASICs provide JTAG boundry-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`. HardCopy Stratix devices support the JTAG instructions as shown in Table 3–1.

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

JTAG Instruction	Instruction Code	Description
`SAMPLE/PRELOAD`	`00 0000 0101`	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation, and permits an initial data pattern to be output at the device pins.
`EXTEST` *(1)*	`00 0000 0000`	Allows the external circuitry and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
`BYPASS`	`11 1111 1111`	Places the 1-bit bypass register between the `TDI` and `TDO` pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.
`USERCODE`	`00 0000 0111`	Selects the 32-bit `USERCODE` register and places it between the `TDI` and `TDO` pins, allowing the `USERCODE` to be serially shifted out of `TDO`.
`IDCODE`	`00 0000 0110`	Selects the `IDCODE` register and places it between `TDI` and `TDO`, allowing the `IDCODE` to be serially shifted out of `TDO`.
`HIGHZ` *(1)*	`00 0000 1011`	Places the 1-bit bypass register between the `TDI` and `TDO` pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation, while tri-stating all of the I/O pins.

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

JTAG Instruction	Instruction Code	Description
CLAMP *(1)*	00 0000 1010	Places the 1-bit bypass register between the TDI and TDO pins, which allows the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding I/O pins to a state defined by the data in the boundary-scan register.

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

The boundary-scan description language (BSDL) files for HardCopy Stratix devices are different from the corresponding Stratix FPGAs. The BSDL files for HardCopy Stratix devices are available for download from the Altera website at **www.altera.com**.

The HardCopy Stratix device instruction register length is 10 bits; the USERCODE register length is 32 bits. The USERCODE registers are mask-programmed, so they are not re-programmable. The designer can choose an appropriate 32-bit sequence to program into the USERCODE registers.

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

Table 3–2. HardCopy Stratix Boundary-Scan Register Length

Device	Maximum Boundary-Scan Register Length
HC1S25 672-pin FineLine BGA	1,458
HC1S30 780-pin FineLine BGA	1,878
HC1S40 780-pin FineLine BGA	1,878
HC1S60 1,020-pin FineLine BGA	2,382
HC1S80 1,020-pin FineLine BGA	2,382

Table 3–3. 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 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 Stratix JTAG Waveforms

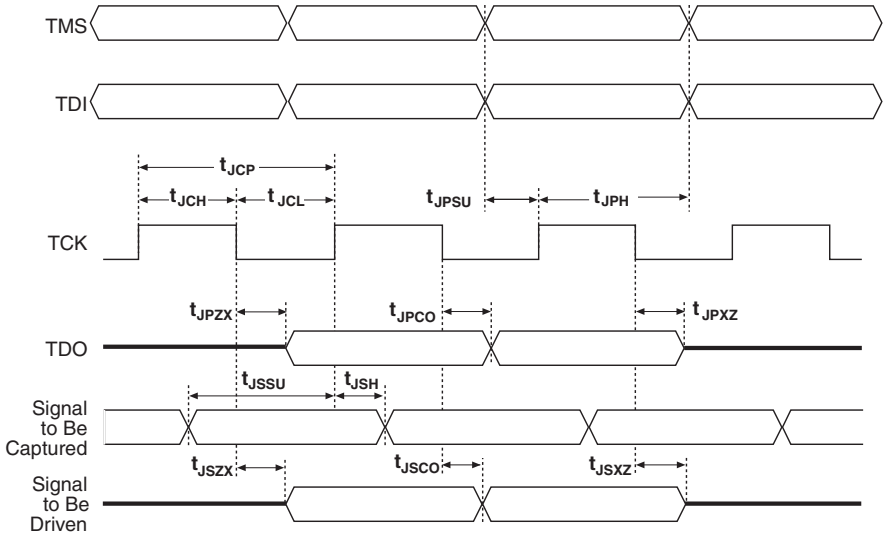


Table 3–4 shows the JTAG timing parameters and values for HardCopy Stratix devices.

Table 3–4. HardCopy Stratix JTAG Timing Parameters and Values

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

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

Document Revision History

Table 3–5 shows the revision history for this chapter.

Table 3–5. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
September 2008 v3.4	Updated chapter number and metadata.	—
June 2007 v3.3	Updated Figure 3–1.	—
December 2006 v3.2	Updated revision history.	—
March 2006	Formerly chapter 7; no content change.	—

Table 3–5. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
October 2005 v3.1	• Minor edits • Graphic updates	—
May 2005 v3.0	Updated “IEEE Std. 1149.1 (JTAG) Boundary-Scan Support” section	
January 2005 v2.0	Added information about `USERCODE` registers	
June 2003 v1.0	Initial release of Chapter 7, Boundary-Scan Support, in the *HardCopy Device Handbook*	

4. Operating Conditions

H51005-3.4

Recommended Operating Conditions

Tables 4–1 through 4–3 provide information on absolute maximum ratings, recommended operating conditions, DC operating conditions, and capacitance for 1.5-V HardCopy® Stratix® devices.

Table 4–1. 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 4–2. 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 4–3. 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	Vi=0; V_{CCIO} = 3.3 V *(9)*	15	25	50	kΩ
		Vi=0; V_{CCIO} = 2.5 V *(9)*	20	45	70	kΩ
		Vi=0; V_{CCIO} = 1.8 V *(9)*	30	65	100	kΩ
		Vi=0; V_{CCIO} = 1.5 V *(9)*	50	100	150	kΩ
	Recommended value of I/O pin external pull-down resistor before and during configuration			1	2	kΩ

Notes to Tables 4–1 through 4–3:

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

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

(3) Minimum DC input is –0.5 V. During transitions, the inputs may undershoot to –2 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 be lower if an external source drives the pin higher than V_{CCIO}.

Tables 4–4 through 4–31 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 4–32 provides information on capacitance for 1.5-V HardCopy Stratix devices.

Table 4–4. LVTTL Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage		3.0	3.6	V
V_{IH}	High-level input voltage		1.7	4.1	V
V_{IL}	Low-level input voltage		–0.5	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –4 to –24 mA *(1)*	2.4		V
V_{OL}	Low-level output voltage	I_{OL} = 4 to 24 mA *(1)*		0.45	V

Table 4–5. LVCMOS Specifications

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

Table 4–6. 2.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage		2.375	2.625	V
V_{IH}	High-level input voltage		1.7	4.1	V
V_{IL}	Low-level input voltage		–0.5	0.7	V
V_{OH}	High-level output voltage	I_{OH} = –0.1 mA	2.1		V
		I_{OH} = –1 mA	2.0		V
		I_{OH} = –2 to –16 mA *(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 4–7. 1.8-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage		1.65	1.95	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	2.25	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 4–8. 1.5-V I/O Specifications

Symbol	Parameter	Conditions	Minimum	Maximum	Unit
V_{CCIO}	Output supply voltage		1.4	1.6	V
V_{IH}	High-level input voltage		0.65 × V_{CCIO}	V_{CCIO} + 0.3	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 4–9. 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

Table 4–9. 3.3-V LVDS I/O Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
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
Δ V_{OCM}	Change in V_{OCM} between high and low	R_L = 100 Ω			50	mV
R_L	Receiver differential input resistor		90	100	110	Ω

Table 4–10. 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 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}	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 4–12. 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 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		–0.5		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 4–14. 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 4–15. 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 4–16. 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 4–17. 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 4–18. 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 4–19. SSTL-2 Class I Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		2.375	2.5	2.625	V
V_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V
V_{REF}	Reference voltage		1.15	1.25	1.35	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.18		3.0	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.35			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –8.1 mA *(1)*	V_{TT} + 0.57			V
V_{OL}	Low-level output voltage	I_{OL} = 8.1 mA *(1)*			V_{TT} – 0.57	V

Table 4–20. SSTL-2 Class II 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_{TT}	Termination voltage		V_{REF} – 0.04	V_{REF}	V_{REF} + 0.04	V

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

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{REF}	Reference voltage		1.15	1.25	1.35	V
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.18		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.18	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.35			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.35	V
V_{OH}	High-level output voltage	I_{OH} = –16.4 mA *(1)*	V_{TT} + 0.76			V
V_{OL}	Low-level output voltage	I_{OL} = 16.4 mA *(1)*			V_{TT} – 0.76	V

Table 4–21. 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
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.2	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.4			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.4	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA *(1)*	V_{TT} + 0.6			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA *(1)*			V_{TT} – 0.6	V

Table 4–22. SSTL-3 Class II Specifications (Part 1 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		3.0	3.3	3.6	V
V_{TT}	Termination voltage		V_{REF} – 0.05	V_{REF}	V_{REF} + 0.05	V
V_{REF}	Reference voltage		1.3	1.5	1.7	V

Table 4–22. SSTL-3 Class II Specifications (Part 2 of 2)

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
$V_{IH(DC)}$	High-level DC input voltage		V_{REF} + 0.2		V_{CCIO} + 0.3	V
$V_{IL(DC)}$	Low-level DC input voltage		–0.3		V_{REF} – 0.2	V
$V_{IH(AC)}$	High-level AC input voltage		V_{REF} + 0.4			V
$V_{IL(AC)}$	Low-level AC input voltage				V_{REF} – 0.4	V
V_{OH}	High-level output voltage	I_{OH} = –16 mA *(1)*	V_{TT} + 0.8			V
V_{OL}	Low-level output voltage	I_{OL} = 16 mA *(1)*			V_{TT} – 0.8	V

Table 4–23. 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 4–24. 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 4–25. 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 4–26. 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 4–27. 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 4–28. 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 4–29. 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 4–30. CTT I/O Specifications

Symbol	Parameter	Conditions	Minimum	Typical	Maximum	Unit
V_{CCIO}	Output supply voltage		2.05	3.3	3.6	V
V_{TT}/V_{REF}	Termination and input reference voltage		1.35	1.5	1.65	V
V_{IH}	High-level input voltage		V_{REF} + 0.2			V
V_{IL}	Low-level input voltage				V_{REF} – 0.2	V
V_{OH}	High-level output voltage	I_{OH} = –8 mA	V_{REF} + 0.4			V
V_{OL}	Low-level output voltage	I_{OL} = 8 mA			V_{REF} – 0.4	V
I_O	Output leakage current (when output is high *Z*)	GND ≤ V_{OUT} ≤ V_{CCIO}	–10		10	µA

Table 4–31. Bus Hold Parameters

Parameter	Conditions	V_{CCIO} Level								Unit
		1.5 V		1.8 V		2.5 V		3.3 V		
		Min	Max	Min	Max	Min	Max	Min	Max	
Low sustaining current	V_{IN} > V_{IL} (maximum)	25		30		50		70		µA
High sustaining current	V_{IN} < V_{IH} (minimum)	–25		–30		–50		–70		µA
Low overdrive current	0 V < V_{IN} < V_{CCIO}		160		200		300		500	µA
High overdrive current	0 V < V_{IN} < V_{CCIO}		–160		–200		–300		–500	µA
Bus hold trip point		0.5	1.0	0.68	1.07	0.7	1.7	0.8	2.0	V

Table 4–32. 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 4–4 through 4–32:

(1) Drive strength is programmable according to values in the *Stratix Architecture* chapter of the *Stratix Device Handbook*.

(2) When the `tx_outclock` port of the `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 website is typically used prior to designing the FPGA in order to get a magnitude estimate of the device power. The Quartus II software power estimation feature allows designers to apply test vectors against their design for more accurate power consumption modeling.

In both cases, these calculations should only be used as an estimation of power, not as a specification.

Timing Closure

The timing numbers in Tables 4–34 to 4–43 are only provided as an indication of allowable timing for HardCopy Stratix devices. The Quartus II software provides preliminary timing information for HardCopy Stratix designs, which can be used as an estimation of the device performance.

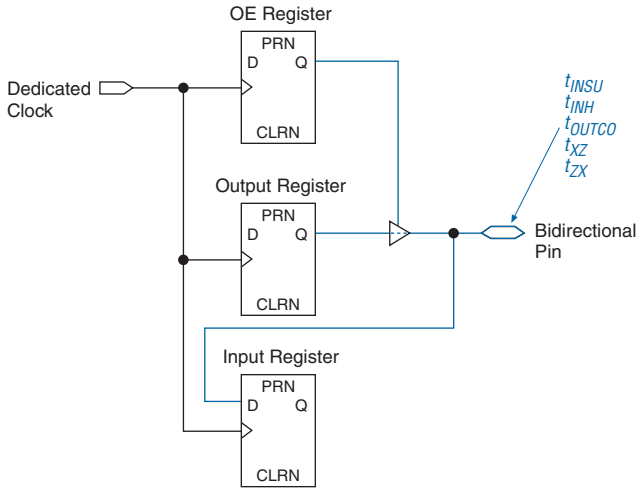
The final timing numbers and actual performance for each HardCopy Stratix design is available when the design migration is complete and are subject to verification and approval by Altera and the designer during the HardCopy Design review process.

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

External Timing Parameters

External timing parameters are specified by device density and speed grade. Figure 4–1 shows the pin-to-pin timing model for bidirectional IOE pin timing. All registers are within the IOE.

Figure 4–1. External Timing in HardCopy Stratix Devices



All external timing parameters reported in this section are defined with respect to the dedicated clock pin as the starting point. All external I/O timing parameters shown are for 3.3-V LVTTL I/O standard with the 4-mA current strength and fast slew rate. For external I/O timing using standards other than LVTTL or for different current strengths, use the I/O standard input and output delay adders in the *Stratix Device Handbook*.

Table 4–33 shows the external I/O timing parameters when using global clock networks.

Table 4–33. HardCopy Stratix Global Clock External I/O Timing Parameters *Notes (1), (2)*

Symbol	Parameter
t_{INSU}	Setup time for input or bidirectional pin using IOE input register with global clock fed by `CLK` pin
t_{INH}	Hold time for input or bidirectional pin using IOE input register with global clock fed by `CLK` pin
t_{OUTCO}	Clock-to-output delay output or bidirectional pin using IOE output register with global clock fed by `CLK` pin
$t_{INSUPLL}$	Setup time for input or bidirectional pin using IOE input register with global clock fed by Enhanced PLL with default phase setting
t_{INHPLL}	Hold time for input or bidirectional pin using IOE input register with global clock fed by Enhanced PLL with default phase setting
$t_{OUTCOPLL}$	Clock-to-output delay output or bidirectional pin using IOE output register with global clock Enhanced PLL with default phase setting
t_{XZPLL}	Synchronous IOE output enable register to output pin disable delay using global clock fed by Enhanced PLL with default phase setting
t_{ZXPLL}	Synchronous IOE output enable register to output pin enable delay using global clock fed by Enhanced PLL with default phase setting

Notes to Table 4–33:

(1) These timing parameters are sample-tested only.

(2) These timing parameters are for column and row IOE pins. Designers should use the Quartus II software to verify the external timing for any pin.

HardCopy Stratix External I/O Timing

These timing parameters are for both column IOE and row IOE pins. In HC1S30 devices and above, designers can decrease the t_{SU} time by using `FPLLCLK`, but may get positive hold time in HC1S60 and HC1S80 devices. Designers should use the Quartus II software to verify the external devices for any pin.

Tables 4–34 through 4–35 show the external timing parameters on column and row pins for HC1S25 devices.

Table 4–34. HC1S25 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.371		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.809	7.155	ns
t_{XZ}	2.749	7.040	ns
t_{ZX}	2.749	7.040	ns
$t_{INSUPLL}$	1.271		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.124	2.602	ns
t_{XZPLL}	1.064	2.487	ns
t_{ZXPLL}	1.064	2.487	ns

Table 4–35. HC1S25 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.665		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.834	7.194	ns
t_{XZ}	2.861	7.276	ns
t_{ZX}	2.861	7.276	ns
$t_{INSUPLL}$	1.538		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.164	2.653	ns
t_{XZPLL}	1.191	2.735	ns
t_{ZXPLL}	1.191	2.735	ns

Tables 4–36 through 4–37 show the external timing parameters on column and row pins for HC1S30 devices.

Table 4–36. HC1S30 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.935		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.814	7.274	ns
t_{XZ}	2.754	7.159	ns
t_{ZX}	2.754	7.159	ns
$t_{INSUPLL}$	1.265		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.068	2.423	ns
t_{XZPLL}	1.008	2.308	ns
t_{ZXPLL}	1.008	2.308	ns

Table 4–37. HC1S30 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	1.995		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.917	7.548	ns
t_{XZ}	2.944	7.630	ns
t_{ZX}	2.944	7.630	ns
$t_{INSUPLL}$	1.337		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.164	2.672	ns
t_{XZPLL}	1.191	2.754	ns
t_{ZXPLL}	1.191	2.754	ns

Tables 4–38 through 4–39 show the external timing parameters on column and row pins for HC1S40 devices.

Table 4–38. HC1S40 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.126		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.856	7.253	ns
t_{XZ}	2.796	7.138	ns
t_{ZX}	2.796	7.138	ns
$t_{INSUPLL}$	1.466		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.092	2.473	ns
t_{XZPLL}	1.032	2.358	ns
t_{ZXPLL}	1.032	2.358	ns

Table 4–39. HC1S40 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.020		ns
t_{INH}	0.000		ns
t_{OUTCO}	2.912	7.480	ns
t_{XZ}	2.939	7.562	ns
t_{ZX}	2.939	7.562	ns
$t_{INSUPLL}$	1.370		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.144	2.693	ns
t_{XZPLL}	1.171	2.775	ns
t_{ZXPLL}	1.171	2.775	ns

Tables 4–40 through 4–41 show the external timing parameters on column and row pins for HC1S60 devices.

Table 4–40. HC1S60 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.000		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.051	6.977	ns
t_{XZ}	2.991	6.853	ns
t_{ZX}	2.991	6.853	ns
$t_{INSUPLL}$	1.315		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.029	2.323	ns
t_{XZPLL}	0.969	2.199	ns
t_{ZXPLL}	0.969	2.199	ns

Table 4–41. HC1S60 External I/O Timing on Row Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	2.232		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.182	7.286	ns
t_{XZ}	3.209	7.354	ns
t_{ZX}	3.209	7.354	ns
$t_{INSUPLL}$	1.651		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.154	2.622	ns
t_{XZPLL}	1.181	2.690	ns
t_{ZXPLL}	1.181	2.690	ns

Tables 4–42 through 4–43 show the external timing parameters on column and row pins for HC1S80 devices.

Table 4–42. HC1S80 External I/O Timing on Column Pins Using Global Clock Networks

Parameter	Performance		Unit
	Min	Max	
t_{INSU}	0.884		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.267	7.415	ns
t_{XZ}	3.207	7.291	ns
t_{ZX}	3.207	7.291	ns
$t_{INSUPLL}$	0.506		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.635	2.828	ns
t_{XZPLL}	1.575	2.704	ns
t_{ZXPLL}	1.575	2.704	ns

Table 4–43. HC1S80 External I/O Timing on Rows Using Pin Global Clock Networks

Symbol	Performance		Unit
	Min	Max	
t_{INSU}	1.362		ns
t_{INH}	0.000		ns
t_{OUTCO}	3.457	7.859	ns
t_{XZ}	3.484	7.927	ns
t_{ZX}	3.484	7.927	ns
$t_{INSUPLL}$	0.994		ns
t_{INHPLL}	0.000		ns
$t_{OUTCOPLL}$	1.821	3.254	ns
t_{XZPLL}	1.848	3.322	ns
t_{ZXPLL}	1.848	3.322	ns

Maximum Input and Output Clock Rates

Tables 4–44 through 4–46 show the maximum input clock rate for column and row pins in HardCopy Stratix devices.

Table 4–44. HardCopy Stratix Maximum Input Clock Rate for CLK[7..4] and CLK[15..12] Pins

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
GTL	300	MHz
GTL+	300	MHz
SSTL-3 class I	400	MHz
SSTL-3 class II	400	MHz
SSTL-2 class I	400	MHz
SSTL-2 class II	400	MHz
SSTL-18 class I	400	MHz
SSTL-18 class II	400	MHz
1.5-V HSTL class I	400	MHz
1.5-V HSTL class II	400	MHz
1.8-V HSTL class I	400	MHz
1.8-V HSTL class II	400	MHz
3.3-V PCI	422	MHz
3.3-V PCI-X 1.0	422	MHz
Compact PCI	422	MHz
AGP 1×	422	MHz
AGP 2×	422	MHz
CTT	300	MHz
Differential HSTL	400	MHz
LVPECL *(1)*	645	MHz
PCML *(1)*	300	MHz
LVDS *(1)*	645	MHz
HyperTransport technology *(1)*	500	MHz

Table 4–45. HardCopy Stratix Maximum Input Clock Rate for CLK[0, 2, 9, 11] Pins and FPLL[10..7]CLK Pins

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
GTL	300	MHz
GTL+	300	MHz
SSTL-3 class I	400	MHz
SSTL-3 class II	400	MHz
SSTL-2 class I	400	MHz
SSTL-2 class II	400	MHz
SSTL-18 class I	400	MHz
SSTL-18 class II	400	MHz
1.5-V HSTL class I	400	MHz
1.5-V HSTL class II	400	MHz
1.8-V HSTL class I	400	MHz
1.8-V HSTL class II	400	MHz
3.3-V PCI	422	MHz
3.3-V PCI-X 1.0	422	MHz
Compact PCI	422	MHz
AGP 1×	422	MHz
AGP 2×	422	MHz
CTT	300	MHz
Differential HSTL	400	MHz
LVPECL *(1)*	717	MHz
PCML *(1)*	400	MHz
LVDS *(1)*	717	MHz
HyperTransport technology *(1)*	717	MHz

Table 4–46. HardCopy Stratix Maximum Input Clock Rate for CLK[1, 3, 8, 10] Pins

I/O Standard	Performance	Unit
LVTTL	422	MHz
2.5 V	422	MHz
1.8 V	422	MHz
1.5 V	422	MHz
LVCMOS	422	MHz
GTL	300	MHz
GTL+	300	MHz
SSTL-3 class I	400	MHz
SSTL-3 class II	400	MHz
SSTL-2 class I	400	MHz
SSTL-2 class II	400	MHz
SSTL-18 class I	400	MHz
SSTL-18 class II	400	MHz
1.5-V HSTL class I	400	MHz
1.5-V HSTL class II	400	MHz
1.8-V HSTL class I	400	MHz
1.8-V HSTL class II	400	MHz
3.3-V PCI	422	MHz
3.3-V PCI-X 1.0	422	MHz
Compact PCI	422	MHz
AGP 1×	422	MHz
AGP 2×	422	MHz
CTT	300	MHz
Differential HSTL	400	MHz
LVPECL *(1)*	645	MHz
PCML *(1)*	300	MHz
LVDS *(1)*	645	MHz
HyperTransport technology *(1)*	500	MHz

Note to Tables 4–44 through 4–46:

(1) These parameters are only available on row I/O pins.

Tables 4–47 through 4–48 show the maximum output clock rate for column and row pins in HardCopy Stratix devices.

Table 4–47. HardCopy Stratix Maximum Output Clock Rate for PLL[5, 6, 11, 12] Pins (Part 1 of 2)

I/O Standard	Performance	Unit
LVTTL	350	MHz
2.5 V	350	MHz
1.8 V	250	MHz
1.5 V	225	MHz
LVCMOS	350	MHz
GTL	200	MHz
GTL+	200	MHz
SSTL-3 class I	200	MHz
SSTL-3 class II	200	MHz
SSTL-2 class I *(3)*	200	MHz
SSTL-2 class I *(4)*	200	MHz
SSTL-2 class I *(5)*	150	MHz
SSTL-2 class II *(3)*	200	MHz
SSTL-2 class II *(4)*	200	MHz
SSTL-2 class II *(5)*	150	MHz
SSTL-18 class I	150	MHz
SSTL-18 class II	150	MHz
1.5-V HSTL class I	250	MHz
1.5-V HSTL class II	225	MHz
1.8-V HSTL class I	250	MHz
1.8-V HSTL class II	225	MHz
3.3-V PCI	350	MHz
3.3-V PCI-X 1.0	350	MHz
Compact PCI	350	MHz
AGP 1×	350	MHz
AGP 2×	350	MHz
CTT	200	MHz
Differential HSTL	225	MHz
Differential SSTL-2 *(6)*	200	MHz
LVPECL *(2)*	500	MHz
PCML *(2)*	350	MHz

Table 4–47. HardCopy Stratix Maximum Output Clock Rate for PLL[5, 6, 11, 12] Pins (Part 2 of 2)

I/O Standard	Performance	Unit
LVDS *(2)*	500	MHz
HyperTransport technology *(2)*	350	MHz

Table 4–48. HardCopy Stratix Maximum Output Clock Rate (Using I/O Pins) for PLL[1, 2, 3, 4] Pins (Part 1 of 2)

I/O Standard	Performance	Unit
LVTTL	400	MHz
2.5 V	400	MHz
1.8 V	400	MHz
1.5 V	350	MHz
LVCMOS	400	MHz
GTL	200	MHz
GTL+	200	MHz
SSTL-3 class I	167	MHz
SSTL-3 class II	167	MHz
SSTL-2 class I	150	MHz
SSTL-2 class II	150	MHz
SSTL-18 class I	150	MHz
SSTL-18 class II	150	MHz
1.5-V HSTL class I	250	MHz
1.5-V HSTL class II	225	MHz
1.8-V HSTL class I	250	MHz
1.8-V HSTL class II	225	MHz
3.3-V PCI	250	MHz
3.3-V PCI-X 1.0	225	MHz
Compact PCI	400	MHz
AGP 1×	400	MHz
AGP 2×	400	MHz
CTT	300	MHz
Differential HSTL	225	MHz
LVPECL *(2)*	717	MHz
PCML *(2)*	420	MHz

Table 4–48. HardCopy Stratix Maximum Output Clock Rate (Using I/O Pins) for PLL[1, 2, 3, 4] Pins (Part 2 of 2)

I/O Standard	Performance	Unit
LVDS *(2)*	717	MHz
HyperTransport technology *(2)*	420	MHz

Notes to Tables 4–47 through 4–48:
(1) Differential SSTL-2 outputs are only available on column clock pins.
(2) These parameters are only available on row I/O pins.
(3) SSTL-2 in maximum drive strength condition.
(4) SSTL-2 in minimum drive strength with ≤10pF output load condition.
(5) SSTL-2 in minimum drive strength with > 10pF output load condition.
(6) Differential SSTL-2 outputs are only supported on column clock pins.

High-Speed I/O Specification

Table 4–49 provides high-speed timing specifications definitions.

Table 4–49. High-Speed Timing Specifications and Terminology

High-Speed Timing Specification	Terminology
t_C	High-speed receiver/transmitter input and output clock period.
f_{HSCLK}	High-speed receiver/transmitter input and output clock frequency.
t_{RISE}	Low-to-high transmission time.
t_{FALL}	High-to-low transmission time.
Timing unit interval (TUI)	The timing budget allowed for skew, propagation delays, and data sampling window. (TUI = 1/(Receiver Input Clock Frequency × Multiplication Factor) = t_C/w).
f_{HSDR}	Maximum LVDS data transfer rate (f_{HSDR} = 1/TUI).
Channel-to-channel skew (TCCS)	The timing difference between the fastest and slowest output edges, including t_{CO} variation and clock skew. The clock is included in the TCCS measurement.
Sampling window (SW)	The period of time during which the data must be valid to be captured correctly. The setup and hold times determine the ideal strobe position within the sampling window. SW = t_{SW} (max) – t_{SW} (min).
Input jitter (peak-to-peak)	Peak-to-peak input jitter on high-speed PLLs.
Output jitter (peak-to-peak)	Peak-to-peak output jitter on high-speed PLLs.
t_{DUTY}	Duty cycle on high-speed transmitter output clock.
t_{LOCK}	Lock time for high-speed transmitter and receiver PLLs.

Table 4–50 shows the high-speed I/O timing for HardCopy Stratix devices.

Table 4–50. High-Speed I/O Specifications (Part 1 of 2) Notes (1), (2)

Symbol	Conditions	Performance			Unit
		Min	Typ	Max	
f_{HSCLK} (Clock frequency) (LVDS, LVPECL, HyperTransport technology) $f_{HSCLK} = f_{HSDR} / W$	*W* = 4 to 30 (Serdes used)	10		210	MHz
	W = 2 (Serdes bypass)	50		231	MHz
	W = 2 (Serdes used)	150		420	MHz
	W = 1 (Serdes bypass)	100		462	MHz
	W = 1 (Serdes used)	300		717	MHz
f_{HSDR} Device operation (LVDS, LVPECL, HyperTransport technology)	*J* = 10	300		840	Mbps
	J = 8	300		840	Mbps
	J = 7	300		840	Mbps
	J = 4	300		840	Mbps
	J = 2	100		462	Mbps
	J = 1 (LVDS and LVPECL only)	100		462	Mbps
f_{HSCLK} (Clock frequency) (PCML) $f_{HSCLK} = f_{HSDR} / W$	*W* = 4 to 30 (Serdes used)	10		100	MHz
	W = 2 (Serdes bypass)	50		200	MHz
	W = 2 (Serdes used)	150		200	MHz
	W = 1 (Serdes bypass)	100		250	MHz
	W = 1 (Serdes used)	300		400	MHz
f_{HSDR} Device operation (PCML)	*J* = 10	300		400	Mbps
	J = 8	300		400	Mbps
	J = 7	300		400	Mbps
	J = 4	300		400	Mbps
	J = 2	100		400	Mbps
	J = 1	100		250	Mbps
TCCS	All			200	ps
SW	PCML (*J* = 4, 7, 8, 10)	750			ps
	PCML (*J* = 2)	900			ps
	PCML (*J* = 1)	1,500			ps
	LVDS and LVPECL (*J* = 1)	500			ps
	LVDS, LVPECL, HyperTransport technology (*J* = 2 through 10)	440			ps

Table 4–50. High-Speed I/O Specifications (Part 2 of 2) Notes (1), (2)

Symbol	Conditions	Performance			Unit
		Min	Typ	Max	
Input jitter tolerance (peak-to-peak)	All			250	ps
Output jitter (peak-to-peak)	All			160	ps
Output t_{RISE}	LVDS	80	110	120	ps
	HyperTransport technology	110	170	200	ps
	LVPECL	90	130	150	ps
	PCML	80	110	135	ps
Output t_{FALL}	LVDS	80	110	120	ps
	HyperTransport technology	110	170	200	ps
	LVPECL	90	130	160	ps
	PCML	105	140	175	ps
t_{DUTY}	LVDS (J = 2 through 10)	47.5	50	52.5	%
	LVDS (J =1) and LVPECL, PCML, HyperTransport technology	45	50	55	%
t_{LOCK}	All			100	µs

Notes to Table 4–50:

(1) When J = 4, 7, 8, and 10, the SERDES block is used.

(2) When J = 2 or J = 1, the SERDES is bypassed.

PLL Specifications

Table 4–51 describes the HardCopy Stratix device enhanced PLL specifications.

Table 4–51. Enhanced PLL Specifications (Part 1 of 3)

Symbol	Parameter	Min	Typ	Max	Unit
f_{IN}	Input clock frequency	3 *(1)*		684	MHz
f_{INDUTY}	Input clock duty cycle	40		60	%
$f_{EINDUTY}$	External feedback clock input duty cycle	40		60	%
$t_{INJITTER}$	Input clock period jitter			±200 *(2)*	ps
$t_{EINJITTER}$	External feedback clock period jitter			±200 *(2)*	ps
t_{FCOMP}	External feedback clock compensation time *(3)*			6	ns

Table 4–51. Enhanced PLL Specifications (Part 2 of 3)

Symbol	Parameter	Min	Typ	Max	Unit
f_{OUT}	Output frequency for internal global or regional clock	0.3		500	MHz
f_{OUT_EXT}	Output frequency for external clock *(2)*	0.3		526	MHz
$t_{OUTDUTY}$	Duty cycle for external clock output (when set to 50%)	45		55	%
t_{JITTER}	Period jitter for external clock output *(5)*			±100 ps for >200 MHz `outclk` ±20 mUI for <200 MHz `outclk`	ps or mUI
$t_{CONFIG5,6}$	Time required to reconfigure the scan chains for PLLs 5 and 6			$289/f_{SCANCLK}$	
$t_{CONFIG11,12}$	Time required to reconfigure the scan chains for PLLs 11 and 12			$193/f_{SCANCLK}$	
$t_{SCANCLK}$	`scanclk` frequency *(4)*			22	MHz
t_{DLOCK}	Time required to lock dynamically (after switchover or reconfiguring any non-post-scale counters/delays) *(6)*	*(8)*		100	µs
t_{LOCK}	Time required to lock from end of device configuration	10		400	µs
f_{VCO}	PLL internal VCO operating range	300		800 *(7)*	MHz
t_{LSKEW}	Clock skew between two external clock outputs driven by the same counter		±50		ps
t_{SKEW}	Clock skew between two external clock outputs driven by the different counters with the same settings		±75		ps
f_{SS}	Spread spectrum modulation frequency	30		150	kHz
% spread	Percentage spread for spread spectrum frequency *(9)*	0.4	0.5	0.6	%

Table 4–51. Enhanced PLL Specifications (Part 3 of 3)

Symbol	Parameter	Min	Typ	Max	Unit
t_{ARESET}	Minimum pulse width on ARESET signal	10 *(11)*			ns
		500 *(12)*			ns

Notes to Table 4–51:

(1) The minimum input clock frequency to the PFD (f_{IN}/N) must be at least 3 MHz for HardCopy Stratix device enhanced PLLs.
(2) Refer to "Maximum Input and Output Clock Rates".
(3) t_{FCOMP} can also equal 50% of the input clock period multiplied by the pre-scale divider *n* (whichever is less).
(4) This parameter is timing analyzed by the Quartus II software because the `scanclk` and `scandata` ports can be driven by the logic array.
(5) Actual jitter performance may vary based on the system configuration.
(6) Total required time to reconfigure and lock is equal to $t_{DLOCK} + t_{CONFIG}$. If only post-scale counters and delays are changed, then t_{DLOCK} is equal to 0.
(7) The VCO range is limited to 500 to 800 MHz when the spread spectrum feature is selected.
(8) Lock time is a function of PLL configuration and may be significantly faster depending on bandwidth settings or feedback counter change increment.
(9) Exact, user-controllable value depends on the PLL settings.
(10) The LOCK circuit on HardCopy Stratix PLLs does not work for industrial devices below –20°C unless the PFD frequency > 200 MHz. Refer to the *Stratix FPGA Errata Sheet* for more information on the PLL.
(11) Applicable when the PLL input clock has been running continuously for at least 10 µs.
(12) Applicable when the PLL input clock has stopped toggling or has been running continuously for less than 10 µs.

Table 4–52 describes the HardCopy Stratix device fast PLL specifications.

Table 4–52. Fast PLL Specifications

Symbol	Parameter	Min	Max	Unit
f_{IN}	CLKIN frequency (for *m* = 1) *(1)*, *(2)*	300	717	MHz
	CLKIN frequency (for *m* = 2 to 19)	300/*m*	1,000/*m*	MHz
	CLKIN frequency (for *m* = 20 to 32)	10	1,000/*m*	MHz
f_{OUT}	Output frequency for internal global or regional clock *(3)*	9.4	420	MHz
f_{OUT_EXT}	Output frequency for external clock *(2)*	9.375	717	MHz
f_{VCO}	VCO operating frequency	300	1,000	MHz
t_{INDUTY}	CLKIN duty cycle	40	60	%
$t_{INJITTER}$	Period jitter for CLKIN pin		±200	ps
t_{DUTY}	Duty cycle for DFFIO 1× CLKOUT pin *(4)*	45	55	%
t_{JITTER}	Period jitter for DFFIO clock out *(4)*		±80	ps
	Period jitter for internal global or regional clock		±100 ps for >200-MHz outclk ±20 mUI for <200-MHz outclk	ps or mUI
t_{LOCK}	Time required for PLL to acquire lock	10	100	µs
m	Multiplication factors for *m* counter *(4)*	1	32	Integer
*l*0, *l*1, *g*0	Multiplication factors for *l*0, *l*1, and *g*0 counter *(5)*, *(6)*	1	32	Integer
t_{ARESET}	Minimum pulse width on areset signal	10		ns

Notes to Table 4–52:

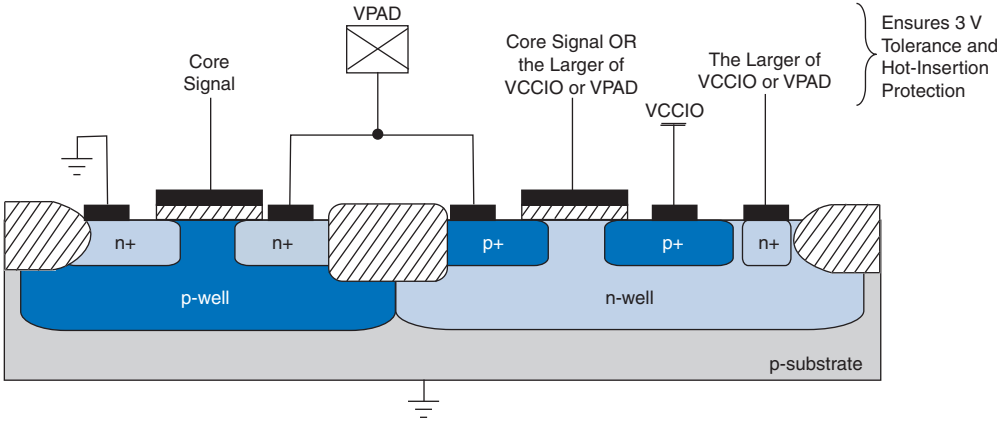
(1) Refer to "Maximum Input and Output Clock Rates" on page 4–23 for more information.
(2) PLLs 7, 8, 9, and 10 in the HC1S80 device support up to 717-MHz input and output.
(3) When using the SERDES, high-speed differential I/O mode supports a maximum output frequency of 210 MHz to the global or regional clocks (for example, the maximum data rate 840 Mbps divided by the smallest SERDES J factor of 4).
(4) This parameter is for high-speed differential I/O mode only.
(5) These counters have a maximum of 32 if programmed for 50/50 duty cycle. Otherwise, they have a maximum of 16.
(6) High-speed differential I/O mode supports *W* = 1 to 16 and *J* = 4, 7, 8, or 10.

Electrostatic Discharge

Electrostatic discharge (ESD) protection is a design practice that is integrated in Altera FPGAs and Structured ASIC devices. HardCopy Stratix devices are no exception, and they are designed with ESD protection on all I/O and power pins.

Figure 4–2 shows a transistor level cross section of the HardCopy Stratix CMOS I/O buffer structure which will be used to explain ESD protection.

Figure 4–2. Transistor-Level Cross Section of the HardCopy Stratix Device I/O Buffers



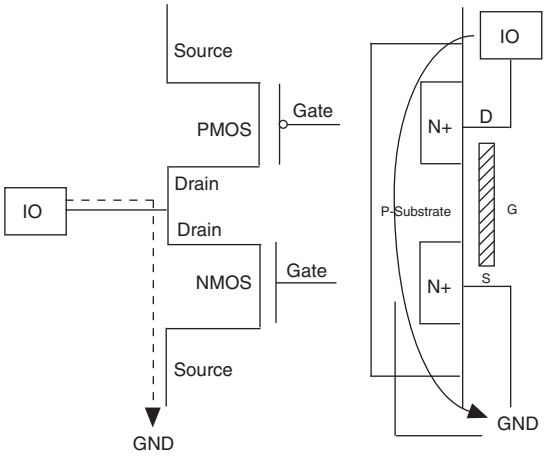
The CMOS output drivers in the I/O pins intrinsically provide electrostatic discharge protection. There are two cases to consider for ESD voltage strikes: positive voltage zap and negative voltage zap.

Positive Voltage Zap

A positive ESD voltage zap occurs when a positive voltage is present on an I/O pin due to an ESD charge event. This can cause the N+ (Drain)/P-Substrate) junction of the N-channel drain to break down and the N+ (Drain)/P-Substrate/N+ (Source) intrinsic bipolar transistor turns ON to discharge ESD current from I/O pin to GND.

The dashed line (Figure 4–3) shows the ESD current discharge path during a positive voltage zap.

Figure 4–3. ESD Protection During Positive Voltage Zap

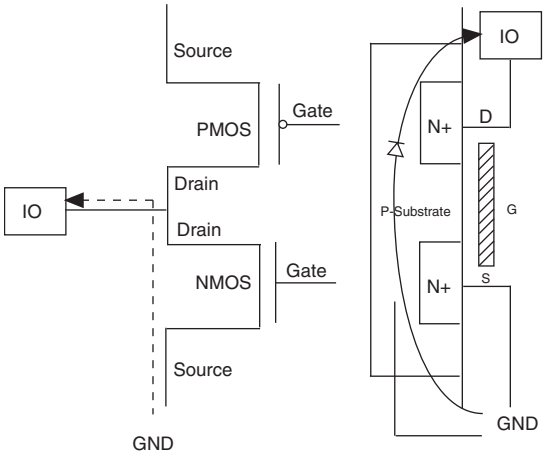


Negative Voltage Zap

When the I/O pin receives a negative ESD zap at the pin that is less than -0.7 V (0.7 V is the voltage drop across a diode), the intrinsic PSubstrate/N+ drain diode is forward biased. Hence, the discharge ESD current path is from GND to the I/O pin, as shown in Figure 4–4.

The dashed line (Figure 4–4) shows the ESD current discharge path during a negative voltage zap.

Figure 4–4. ESD Protection During Negative Voltage Zap



Details of ESD protection are also outlined in the *Hot-Socketing and Power-Sequencing Feature and Testing for Altera Devices* white paper located on the Altera website at www.altera.com.

For information on ESD results of Altera products, see the Reliability Report on the Altera website at www.altera.com.

Document Revision History

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

Table 4–53. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
September 2008 v3.4	Updated the revision history.	—
June 2007 v3.3	Updated R_{CONF} section of Table 4–3. Added the "Electrostatic Discharge" section.	—

Table 4–53. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
December 2006 v3.2	Updated chapter number and metadata.	—
March 2006	Formerly chapter 8; no content change.	—
October 2005 v3.1	• Minor edits • Graphic updates	
May 2005 v3.0	• Updated SSTL-2 and SSTL-3 specifications in Tables 8–19 through 8–22 • Updated CTT I/O specifications in Table 8–30 • Updated bus hold parameters in Table 8–31. • Added the External Timing Parameters, HardCopy Stratix External I/O Timing, and Maximum Input and Output Clock Rates sections • Added the High-Speed I/O Specification, and PLL Specifications sections	—
January 2005 v2.0	Removed recommended maximum rise and fall times (t_R and t_F) for input signals	—
June 2003 v1.0	Initial release of Chapter 8, Operating Conditions, in the *HardCopy Device Handbook*	

5. Quartus II Support for HardCopy Stratix Devices

H51014-3.4

Introduction

Altera® HardCopy devices provide a comprehensive alternative to ASICs. HardCopy structured ASICs offer a complete solution from prototype to high-volume production, and maintain the powerful features and high-performance architecture of their equivalent FPGAs with the programmability removed. You can use the Quartus II design software to design HardCopy devices in a manner similar to the traditional ASIC design flow and you can prototype with Altera's high density Stratix, APEX 20KC, and APEX 20KE FPGAs before seamlessly migrating to the corresponding HardCopy device for high-volume production.

HardCopy structured ASICs provide the following key benefits:

- Improves performance, on the average, by 40% over the corresponding -6 speed grade FPGA device
- Lowers power consumption, on the average, by 40% over the corresponding FPGA
- Preserves the FPGA architecture and features and minimizes risk
- Guarantees first-silicon success through a proven, seamless migration process from the FPGA to the equivalent HardCopy device
- Offers a quick turnaround of the FPGA design to a structured ASIC device—samples are available in about eight weeks

Altera's Quartus II software has built-in support for HardCopy Stratix devices. The HardCopy design flow in Quartus II software offers the following advantages:

- Unified design flow from prototype to production
- Performance estimation of the HardCopy Stratix device allows you to design systems for maximum throughput
- Easy-to-use and inexpensive design tools from a single vendor
- An integrated design methodology that enables system-on-a-chip designs

This section discusses the following areas:

- How to design HardCopy Stratix and HardCopy APEX structured ASICs using the Quartus II software
- An explanation of what the HARDCOPY_FPGA_PROTOTYPE devices are and how to target designs to these devices
- Performance and power estimation of HardCopy Stratix devices
- How to generate the HardCopy design database for submitting HardCopy Stratix and HardCopy APEX designs to the HardCopy Design Center

Features

Beginning with version 4.2, the Quartus II software contains several powerful features that facilitate design of HardCopy Stratix and HardCopy APEX devices:

- **HARDCOPY_FPGA_PROTOTYPE Devices**
  These are virtual Stratix FPGA devices with features identical to HardCopy Stratix devices. You must use these FPGA devices to prototype your designs and verify the functionality in silicon.

- **HardCopy Timing Optimization Wizard**
  Using this feature, you can target your design to HardCopy Stratix devices, providing an estimate of the design's performance in a HardCopy Stratix device.

- **HardCopy Stratix Floorplans and Timing Models**
  The Quartus II software supports post-migration HardCopy Stratix device floorplans and timing models and facilitates design optimization for design performance.

- **Placement Constraints**
  Location and LogicLock constraints are supported at the HardCopy Stratix floorplan level to improve overall performance.

- **Improved Timing Estimation**
  Beginning with version 4.2, the Quartus II software determines routing and associated buffer insertion for HardCopy Stratix designs, and provides the Timing Analyzer with more accurate information about the delays than was possible in previous versions of the Quartus II software. The Quartus II Archive 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 allows you to deliver to Altera the design database and all the deliverables required for migration. This feature is used for HardCopy Stratix and HardCopy APEX devices.

The HardCopy Stratix and HardCopy APEX PowerPlay Early Power Estimator is available on the Altera website at **www.altera.com**.

HARDCOPY_FPGA_PROTOTYPE, HardCopy Stratix and Stratix Devices

You must use the HARDCOPY_FPGA_PROTOTYPE virtual devices available in the 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 5–1 compares HARDCOPY_FPGA_PROTOTYPE devices, Stratix devices, and HardCopy Stratix devices.

Table 5–1. Qualitative Comparison of HARDCOPY_FPGA_PROTOTYPE to Stratix and HardCopy Stratix Devices (Part 1 of 2)

Stratix Device	HARDCOPY_FPGA_PROTOTYPE Device	HardCopy Stratix Device
FPGA	Virtual FPGA	Structured ASIC
FPGA	Architecture identical to Stratix FPGA	Architecture identical to Stratix FPGA

Table 5–1. Qualitative Comparison of HARDCOPY_FPGA_PROTOTYPE to Stratix and HardCopy Stratix Devices (Part 2 of 2)

Stratix Device	HARDCOPY_FPGA_ PROTOTYPE Device	HardCopy Stratix Device
FPGA	Resources identical to HardCopy Stratix device	M-RAM resources different than Stratix FPGA in some devices
Ordered through Altera part number	Cannot be ordered, use the Altera Stratix FPGA part number	Ordered by Altera part number

Table 5–2 lists the resources available in each of the HardCopy Stratix devices.

Table 5–2. 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 5–2:

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

(2) The M-RAM resources for these HardCopy devices differ from the corresponding Stratix FPGA.

For a given device, the number of available M-RAM blocks in HardCopy Stratix devices is identical with the corresponding HARDCOPY_FPGA_PROTOTYPE devices, but may be different from the corresponding Stratix devices. Maintaining the identical resources between HARDCOPY_FPGA_PROTOTYPE and HardCopy Stratix devices facilitates seamless migration from the FPGA to the structured ASIC device.

For more information about HardCopy Stratix devices, refer to the *HardCopy Stratix Device Family Data Sheet* section in volume 1 of the *HardCopy Series Handbook.*

The three devices, Stratix FPGA, HARDCOPY_FPGA_PROTOTYPE, and HardCopy device, are distinct devices in the Quartus II software. The HARDCOPY_FPGA_PROTOTYPE programming files are used in the

Stratix FPGA for your design. The three devices are tied together with the same netlist, thus a single SRAM Object File (**.sof**) can be used to achieve the various goals at each stage. The same SRAM Object File is generated in the HARDCOPY_FPGA_PROTOTYPE design, and is used to program the Stratix FPGA device, the same way that it is used to generate the HardCopy Stratix device, guaranteeing a seamless migration.

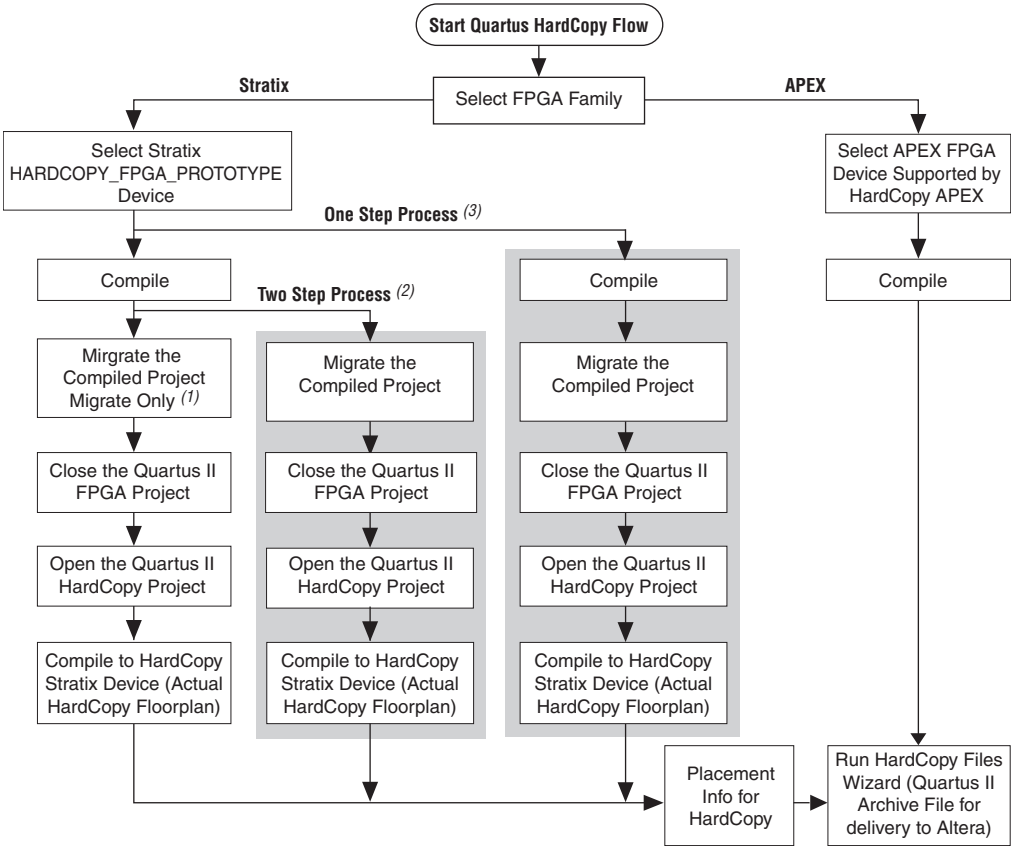
For more information about the SRAM Object File and programming Stratix FPGA devices, refer to the *Programming and Configuration* chapter of the *Introduction to Quartus II Manual*.

HardCopy Design Flow

Figure 5–1 shows a HardCopy design flow diagram. The design steps are explained in detail in the following sections of this chapter. The HardCopy Stratix design flow utilizes the HardCopy Timing Optimization Wizard to automate the migration process into a one-step process. The remainder of this section explains the tasks performed by this automated process.

For a detailed description of the HardCopy Timing Optimization Wizard and HardCopy Files Wizard, refer to “HardCopy Timing Optimization Wizard Summary” and “Generating the HardCopy Design Database”.

Figure 5–1. HardCopy Stratix and HardCopy APEX Design Flow Diagram



Notes to Figure 5–1:

(1) Migrate Only Process: The displayed flow is completed manually.

(2) Two Step Process: Migration and Compilation are done automatically (shaded area).

(3) One Step Process: Full HardCopy Compilation. The entire process is completed automatically (shaded area).

The Design Flow Steps of the One Step Process

The following sections describe each step of the full HardCopy compilation (the One Step Process), Figure 5–1.

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.

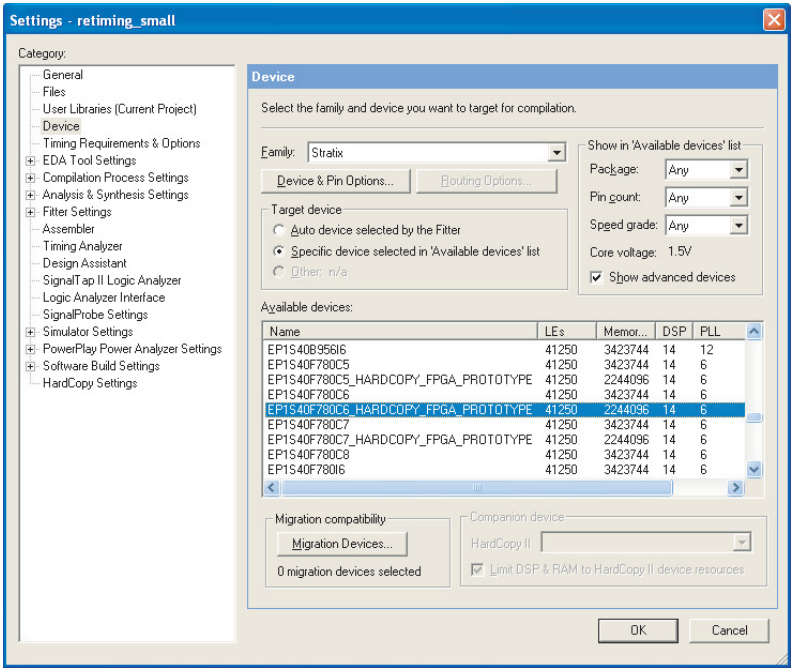
How to Design HardCopy Stratix Devices

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. On the Assignments menu, click **Settings**. In the **Category** list, select **Device**.
3. On the **Device** page, in the **Family** list, select **Stratix**. Select the desired HARDCOPY_FPGA_PROTOTYPE device in the **Available Devices** list (Figure 5–2).

Figure 5–2. Selecting a HARDCOPY_FPGA_PROTOTYPE Device



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.

4. On the Assignments menu, click **Settings**. 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.

5. Add constraints to your HARDCOPY_FPGA_PROTOTYPE device, and on the Processing menu, click **Start Compilation** to compile the design.

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, on the Project menu, point to HardCopy Utilities and click **HardCopy Timing Optimization Wizard**.

At this point, you are presented with the following three choices to target the designs to HardCopy Stratix devices (Figure 5–3).

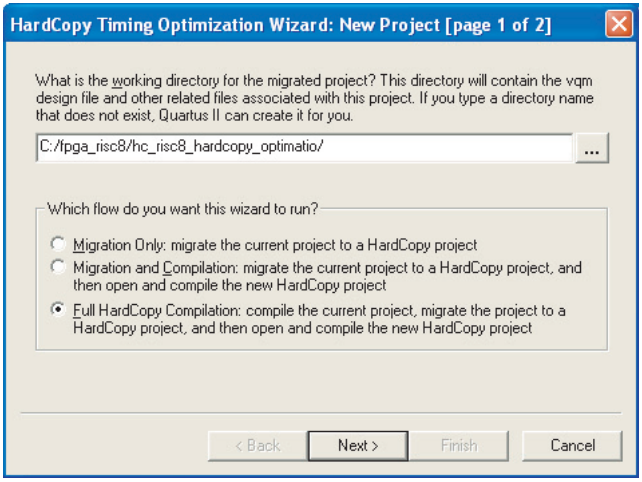
- **Migration Only**: You can select this option after compiling the HARDCOPY_FPGA_PROTOTYPE project to migrate the project to a HardCopy Stratix project.

  You can now perform the following tasks manually to target the design to a HardCopy Stratix device. Refer to"Performance Estimation" on page 5–12 for additional information about 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

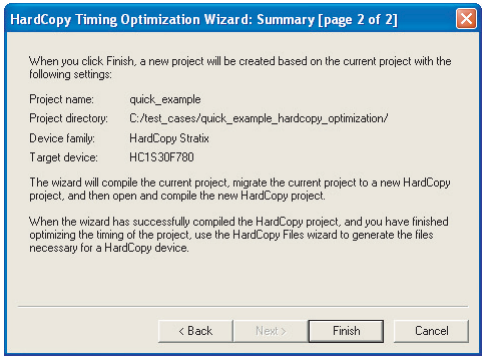
Figure 5–3. HardCopy Timing Optimization Wizard Options



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 5–4).

Figure 5–4. HardCopy Timing Optimization Wizard Summary Page



When either of the second two options in Figure 5–4 are selected (**Migration and Compilation** or **Full HardCopy Compilation**), designs are targeted to HardCopy Stratix devices and optimized using the HardCopy Stratix placement and timing analysis to estimate performance. For details on the performance optimization and estimation steps, refer to "Performance Estimation" on page 5–12. 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 and Performance Estimation

The HardCopy Timing Optimization Wizard creates the HardCopy Stratix project in the Quartus II software, where you can perform design optimization and performance estimation of your HardCopy Stratix device.

Design Optimization

Beginning with version 4.2, the Quartus II software supports HardCopy Stratix design optimization by providing floorplans for placement optimization and HardCopy Stratix timing models. These features allows 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 5–5 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:

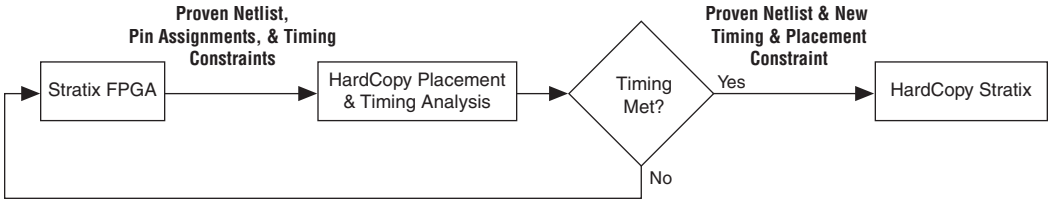
- Work to improve LAB placement in the HardCopy Stratix project.

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 5–5. Obtaining a HardCopy Performance Estimation



To perform Timing Analysis for a HardCopy Stratix device, follow these steps:

1. Open an existing project compiled for a HARDCOPY_FPGA_PROTOYPE device.

2. On the Project menu, point to HardCopy Utilities and click **HardCopy Timing Optimization Wizard**.

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.

CAUTION

If an existing project directory is selected when the HardCopy Timing Optimization Wizard is run, the existing information is overwritten with the new compile results.

The project directory is the directory that you chose for the migrated project. A snapshot of the files inside the *<project name>*_**hardcopy_optimization** directory is shown in Table 5–3.

Table 5–3. Directory Structure Generated by the HardCopy Timing Optimization Wizard

*<project name>*_**hardcopy_optimization**
- *<project name>*.**qsf**
- *<project name>*.**qpf**
- *<project name>*.**sof**
- *<project name>*.**macr**
- *<project name>*.**gclk**
- **db**
- **hardcopy_fpga_prototype**
  - fpga_*<project name>*_**violations.datasheet**
  - fpga_*<project name>*_**target.datasheet**
  - fpga_*<project name>*_**rba_pt_hcpy_v.tcl**
  - fpga_*<project name>*_**pt_hcpy_v.tcl**
  - fpga_*<project name>*_**hcpy_v.sdo**
  - fpga_*<project name>*_**hcpy.vo**
  - fpga_*<project name>*_**cpld.datasheet**
  - fpga_*<project name>*_**cksum.datasheet**
  - fpga_*<project name>*.**tan.rpt**
  - fpga_*<project name>*.**map.rpt**
  - fpga_*<project name>*.**map.atm**
  - fpga_*<project name>*.**fit.rpt**
  - fpga_*<project name>*.**db_info**
  - fpga_*<project name>*.**cmp.xml**
  - fpga_*<project name>*.**cmp.rcf**
  - fpga_*<project name>*.**cmp.atm**
  - fpga_*<project name>*.**asm.rpt**
  - fpga_*<project name>*.**qarlog**
  - fpga_*<project name>*.**qar**
  - fpga_*<project name>*.**qsf**
  - fpga_*<project name>*.**pin**
  - fpga_*<project name>*.**qpf**
- **db_export**
  - *<project name>*.**map.atm**
  - *<project name>*.**map.hdbx**
  - *<project name>*.**db_info**

4. Open the migrated Quartus II project created in Step 3.

5. Perform a full compilation.

   After successful compilation, the Timing Analysis section of the Compilation Report shows the performance of the design.

Performance estimation is not supported for HardCopy APEX devices in the Quartus II software. Your design can be optimized by modifying the RTL code or the FPGA design and the constraints. You should contact Altera to discuss any desired performance improvements with HardCopy APEX devices.

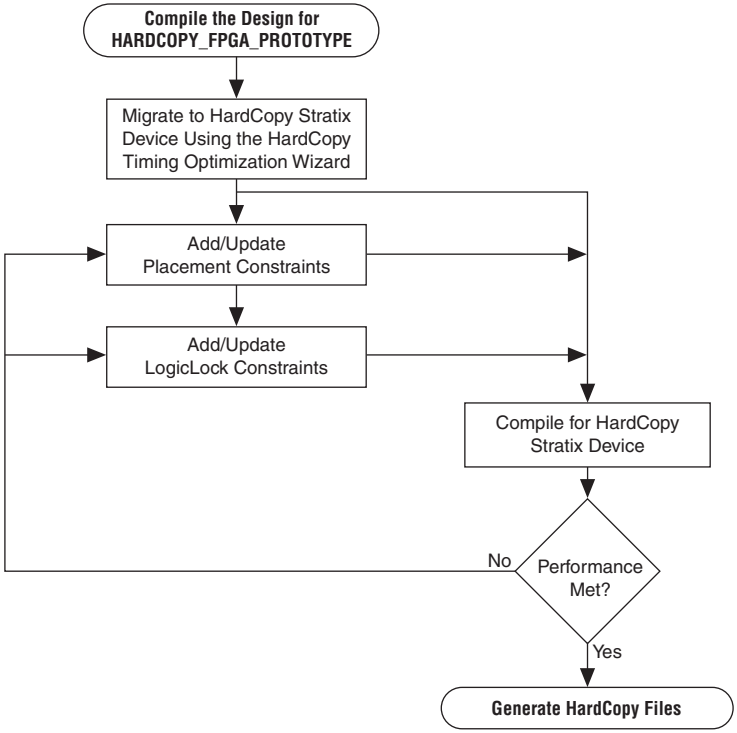
Buffer Insertion

Beginning with version 4.2, the Quartus II software provides improved HardCopy Stratix device timing closure and estimation, to more accurately reflect the results expected after back-end migration. The Quartus II software performs the necessary buffer insertion in your HardCopy Stratix device during the Fitter process, and stores the location of these buffers and necessary routing information in the Quartus II Archive File. This buffer insertion improves the estimation of the Quartus II Timing Analyzer for the HardCopy Stratix device.

Placement Constraints

Beginning with version 4.2, the Quartus II software supports placement constraints and LogicLock regions for HardCopy Stratix devices. Figure 5–6 shows an iterative process to modify the placement constraints until the best placement for the HardCopy Stratix device is achieved.

Figure 5–6. Placement Constraints Flow for HardCopy Stratix Devices



Location Constraints

This section provides information about HardCopy Stratix logic location constraints.

LAB Assignments

Logic placement in HardCopy Stratix is limited to LAB placement and optimization of the interconnecting signals between them. In a Stratix FPGA, individual logic elements (LE) are placed by the Quartus II Fitter into LABs. The HardCopy Stratix migration process requires that LAB contents cannot change after the Timing Optimization Wizard task is done. Therefore, you can only make LAB-level placement optimization and location assignments after migrating the HARDCOPY_FPGA_PROTOTYPE project to the HardCopy Stratix device.

The Quartus II software supports these LAB location constraints for HardCopy Stratix devices. The entire contents of a LAB is moved to an empty LAB when using LAB location assignments. If you want to move the logic contents of LAB A to LAB B, the entire contents of LAB A are moved to an empty LAB B. For example, the logic contents of LAB_X33_Y65 can be moved to an empty LAB at LAB_X43_Y56 but individual logic cell LC_X33_Y65_N1 can not be moved by itself in the HardCopy Stratix Timing Closure Floorplan.

LogicLock Assignments

The LogicLock feature of the Quartus II software provides a block-based design approach. Using this technique you can partition your design and create each block of logic independently, optimize placement and area, and integrate all blocks into the top level design.

To learn more about this methodology, refer to the *Quartus II Analyzing and Optimizing Design Floorplan* 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 Quartus II Settings 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 Quartus II Settings 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 about design recommendations and HDL coding styles, refer to the *Design Guidelines* section in volume 1 of the *Quartus II Handbook*.

Design Assistant

The Quartus II software includes the Design Assistant feature to check your design against the HardCopy design guidelines. Some of the design rule checks performed by the Design Assistant include the following rules:

- Design should not contain combinational loops
- Design should not contain delay chains
- Design should not contain latches

To use the Design Assistant, you must 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 prior to running the design. On the Assignments menu, click **Settings**. In the **Settings** dialog box, in the Category list, select **Design Assistant** and turn on **Run 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, on the Processing menu, point to Start and click **Start Design Assistant**.

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 and Summary

The results of running the Design Assistant on your design are available in the Design Assistant Results section of the Compilation Report. The Design Assistant also generates the summary report in the *<project name>***hardcopy** subdirectory of the project directory. This report file is titled *<project name>*_**violations.datasheet**. Reports include the settings, run summary, results summary, and details of the results and messages. The Design Assistant report indicates the rule name, severity of the violation, and the circuit path where any violation occurred.

To learn about the design rules and standard design practices to comply with HardCopy design rules, refer to the Quartus II Help and the *HardCopy Series Design Guidelines* chapter in volume 1 of the *HardCopy Series Handbook.*

Generating the HardCopy Design Database

You can use the HardCopy Files Wizard to generate the complete set of deliverables required for migrating the design to a HardCopy device in a single click. The HardCopy Files Wizard asks questions related to the design and archives your design, settings, results, and database files for delivery to Altera. Your responses to the design details are stored in *<project name>*_**hardcopy_optimization***<project name>*.**hps.txt**.

You can generate the archive of the HardCopy design database only after compiling the design to a HardCopy Stratix device. The Quartus II Archive File is generated at the same directory level as the targeted project, either before or after optimization.

The Design Assistant automatically runs when the HardCopy Files Wizard is started.

Figure 5–4 shows the archive directory structure and files collected by the HardCopy Files Wizard.

Table 5–4. 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
```

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

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
- *<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 about static timing analysis, refer to the *Classic Timing Analyzer* and the *Synopsys PrimeTime Support* chapters in volume 3 of the *Quartus II Handbook*.

Early Power Estimation

You can use PowerPlay Early Power Estimation to estimate the amount of power your HardCopy Stratix or HardCopy APEX device will consume. This tool is available on the Altera website. Using the Early Power Estimator requires some knowledge of your design resources and specifications, including:

- Target device and package
- Clock networks used in the design
- Resource usage for LEs, DSP blocks, PLL, and RAM blocks
- High speed differential interfaces (HSDI), general I/O power consumption requirements, and pin counts
- Environmental and thermal conditions

HardCopy Stratix Early Power Estimation

The PowerPlay Early Power Estimator provides an initial estimate of I_{CC} for any HardCopy Stratix device based on typical conditions. This calculation saves significant time and effort in gaining a quick understanding of the power requirements for the device. No stimulus vectors are necessary for power estimation, which is established by the clock frequency and toggle rate in each clock domain.

This calculation should only be used as an estimation of power, not as a specification. The actual I_{CC} should be verified during operation because this estimate is sensitive to the actual logic in the device and the environmental operating conditions.

For more information about simulation-based power estimation, refer to the *Power Estimation and Analysis* Section in volume 3 of the *Quartus II Handbook*.

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

HardCopy APEX Early Power Estimation

The PowerPlay Early Power Estimator can be run from the Altera website in the device support section (**http://www.altera.com/support/devices/dvs-index.html**). You cannot open this feature in the Quartus II software.

With the HardCopy APEX PowerPlay Early Power Estimator, you can estimate the power consumed by HardCopy APEX devices and design systems with the appropriate power budget. Refer to the web page for instructions on using the HardCopy APEX PowerPlay Early Power Estimator.

HardCopy APEX devices are generally expected to consume about 40% less power than the equivalent APEX 20KE or APEX 20KC FPGA devices.

Tcl Support for HardCopy Stratix

The Quartus II software also supports the HardCopy Stratix design flow at the command prompt using Tcl scripts.

For details on Quartus II support for Tcl scripting, refer to the *Tcl Scripting* chapter in volume 2 of the *Quartus II Handbook*.

Targeting Designs to HardCopy APEX Devices

Beginning with version 4.2, the Quartus II software supports targeting designs to HardCopy APEX device families. After compiling your design for one of the APEX 20KC or APEX 20KE FPGA devices supported by a HardCopy APEX device, run the HardCopy Files Wizard to generate the necessary set of files for HardCopy migration.

The HardCopy APEX device requires a different set of design files for migration than HardCopy Stratix. Table 5–5 shows the files collected for HardCopy APEX by the HardCopy Files Wizard.

Table 5–5. HardCopy APEX Files Collected by the HardCopy Files Wizard

<project name>.tan.rpt
<project name>.asm.rpt
<project name>.fit.rpt
<project name>.hps.txt
<project name>.map.rpt
<project name>.pin
<project name>.sof
<project name>.qsf
*<project name>*_cksum.datasheet
*<project name>*_cpld.datasheet
*<project name>*_hcpy.vo
*<project name>*_hcpy_v.sdo
*<project name>*_pt_hcpy_v.tcl
*<project name>*_rba_pt_hcpy_v.tcl
*<project name>*_target.datasheet
*<project name>*_violations.datasheet

Refer to "Generating the HardCopy Design Database" on page 5–21 for information about generating the complete set of deliverables required for migrating the design to a HardCopy APEX device. After you have successfully run the HardCopy Files Wizard, you can submit your design archive to Altera to implement your design in a HardCopy device. You should contact Altera for more information about this process.

Conclusion

The methodology for designing HardCopy Stratix devices using the Quartus II software is the same as that for designing the Stratix FPGA equivalent. You can use the familiar Quartus II software tools and design flow, target designs to HardCopy Stratix devices, optimize designs for higher performance and lower power consumption than the Stratix FPGAs, and deliver the design database for migration to a HardCopy Stratix device. Compatible APEX FPGA designs can migrate to HardCopy APEX after compilation using the HardCopy Files Wizard to archive the design files. Submit the files to the HardCopy Design Center to complete the back-end migration.

Related Documents

For more information, refer to the following documentation:

- The *HardCopy Series Design Guidelines* chapter in volume 1 of the *HardCopy Series Handbook.*
- The *HardCopy Series Back-End Timing Closure* chapter in volume 1 of the *HardCopy Series Handbook.*

Document Revision History

Table 5–6 shows the revision history for this chapter.

Table 5–6. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
September 2008 v3.4	Updated chapter number and metadata.	—
June 2007 v3.3	Updated with the current Quartus II software version 7.1 information.	—
December 2006 v3.2	Updated revision history.	—
March 2006	Formerly chapter 20; no content change.	—
October 2005 v3.1	• Updated for technical contents for Quartus II 5.1 release • Minor edits	Minor edits.
May 2005 v3.0	Added PowerPlay early Power estimator information.	—
January 2005 v2.0	This revision was previously the *Quartus® II Support for HardCopy Devices* chapter in the *Quartus II Development Software Handbook, v4.1.*	—
August 2003 v1.1	Overall edit; added Tcl script appendix.	—
June 2003 v1.0	Initial release of Chapter 20, Quartus II Support for HardCopy Stratix Devices.	—

6. Design Guidelines for HardCopy Stratix Performance Improvement

H51027-1.4

Introduction

Advanced design techniques using Altera® HardCopy® Stratix® devices can yield tremendous performance improvements over the design implemented in a Stratix FPGA device. After you verify your Stratix FPGA design in system operation and are ready to migrate to a HardCopy Stratix device, additional device performance is possible through the migration. This chapter focuses on Quartus® II software advanced design techniques that apply to both Stratix FPGA devices and HardCopy Stratix devices. Use these techniques to increase your maximum clock frequency, improve input and output pin timing, and improve timing closure in HardCopy Stratix designs.

Every design is different. The techniques described in this chapter may not apply to every design, and may not yield the same level of improvement.

This document discusses the following topics:

- Planning Stratix FPGA design for HardCopy Stratix design conversion
- Using LogicLock™ regions in HardCopy Stratix designs
- Using Design Space Explorer (DSE) on HardCopy Stratix designs
- Design performance improvement example

Background Information

To understand the Quartus II software and device architecture, and to use the advanced design techniques described in this chapter, Altera recommends reading the *HardCopy Series Handbook* and the following chapters in the *Quartus II Software Handbook*:

- *Design Recommendations for Altera Devices and the Quartus II Design Assistant*
- *Design Optimization for Altera Devices*
- *Design Space Explorer*
- *Analyzing and Optimizing the Design Floorplan*
- *Netlist Optimizations and Physical Synthesis*

Planning Stratix FPGA Design for HardCopy Stratix Design Conversion

In order to achieve greater performance improvement in your HardCopy Stratix device, additional Quartus II software constraints and placement techniques in the HARDCOPY_FPGA_PROTOTYPE design project may be necessary. This does not mean changing the source hardware description language (HDL) code or functionality, but providing additional constraints in the Quartus II software that specifically impact HardCopy Stratix timing optimization.

Planning ahead for migration to the HardCopy design, while still modifying the HARDCOPY_FPGA_PROTOTYPE design, can improve design performance results. You must anticipate how portions of your FPGA design are placed and connected in the HardCopy device floorplan. The HardCopy device floorplan is smaller than the FPGA device floorplan, allowing use of the customized metal routing in HardCopy Stratix devices.

Partitioning Your Design

Partitioning your design into functional blocks is essential in multi-million gate designs. With a HardCopy Stratix device, you can implement approximately one million ASIC gates of logic. Therefore, Altera recommends hierarchical-design partitioning based on system functions.

When using a hierarchical- or incremental-design methodology, you must consider how your design is partitioned to achieve good results. Altera recommends the following practices for partitioning designs as documented in the *Design Recommendations for Altera Devices* chapter in volume 1 of the *Quartus II Development Software Handbook*:

- Partition your design at functional boundaries.
- Minimize the I/O connections between different partitions.
- Register all inputs and outputs of each block. This makes logic synchronous and avoids glitches and any delay penalty on signals that cross between partitions. Registering I/O pins typically eliminates the need to specify timing requirements for signals that connect between different blocks.
- Do not use glue logic or connection logic between hierarchical blocks. When you preserve hierarchy boundaries, glue logic is not merged with hierarchical blocks. Your synthesis software may optimize glue logic separately, which can degrade synthesis results and is not efficient when used with the LogicLock design methodology.
- Logic is not synthesized or optimized across partition boundaries. Any constant values (for example, signals set to GND), are not propagated across partitions.

- Do not use tri-state signals or bidirectional ports on hierarchical boundaries. If you use tri-state boundaries in a lower-level block, synthesis pushes the tri-state signals through the hierarchy to the top-level. This takes advantage of the tri-state drivers on the output pins of the Altera device. Since this requires optimizing through hierarchies, lower-level boundary tri-state signals are not supported with a block-level design methodology.
- Limit clocks to one per block. Partitioning your design into clock domains makes synthesis and timing analysis easier.
- Place state machines in separate blocks to speed optimization and provide greater encoding control.
- Separate timing-critical functions from non-timing-critical functions.
- Limit the critical timing path to one hierarchical block. Group the logic from several design blocks to ensure the critical path resides in one block.

These guidelines apply to all Altera device architectures including HardCopy Stratix devices. Partitioning functional boundaries to have all outputs immediately registered is crucial to using LogicLock regions effectively in HardCopy devices. With registered outputs, you allow the signals to leave a function block at the start of the clock period. This gives the signals more set-up time to reach their endpoints in the clock period. In large designs that are partitioned into multiple function blocks, the block-to-block interconnects are often the limiting factor for f_{MAX} performance. Registered outputs give the Quartus II Fitter the optimal place-and-route flexibility for interconnects between major function blocks.

Physical Synthesis Optimization

All physical synthesis settings in the Quartus II software can be used in the HARDCOPY_FPGA_PROTOTYPE design. These settings are found in the **Physical Synthesis Optimizations** section of the **Fitter Settings** dialog box (Assignments menu) and include the following settings:

- Physical synthesis for combinational logic
- Register duplication
- Register retiming

These settings can improve FPGA performance while developing the HARDCOPY_FPGA_PROTOTYPE. All modifications are passed along into the HardCopy Stratix project when you run the HardCopy Timing Optimization wizard. After running the HardCopy Timing Optimization wizard and subsequently opening the HardCopy project in the Quartus II software, these physical synthesis optimizations are disabled. No further modifications to the netlist are made.

Altera recommends physical synthesis optimizations for the HARDCOPY_FPGA_PROTOTYPE. The work done in the prototype enhances performance in the HardCopy Stratix device after migration. Duplicating combinational logic and registers can increase area utilization, which limits placement flexibility when designs exceed 95% logic element (LE) utilization. However, duplicating combinational logic and registers can help with performance by allowing critical paths to be duplicated when their endpoints must reach different areas of the device floorplan.

For more information on netlist and design optimization, refer to *Area Optimization and Timing Closure* in volume 2 of the *Quartus II Development Software Handbook*.

Using LogicLock Regions in HardCopy Stratix Designs

Create LogicLock regions in the HARDCOPY_FPGA_PROTOTYPE project and migrate the regions into the HardCopy Stratix optimization project using the Quartus II software. LogicLock regions can provide significant benefits in design performance by carefully isolating critical blocks of logic, including:

- MegaCore® IP functions
- I/O interfaces
- Reset or other critical logic feeding global clock lines
- Partitioned function blocks

You must compile your design initially without LogicLock regions present and review the timing analysis reports to determine if additional constraints or LogicLock regions are necessary. This process allows you to determine which function blocks or data paths require LogicLock regions.

Create LogicLock regions in the HARDCOPY_FPGA_PROTOTYPE design project in the Quartus II software. This transfers the LogicLock regions to the HardCopy design project after the HardCopy Timing Optimization Wizard is run. Although the Quartus II software transfers the contents of the LogicLock region, the area, location, and soft boundary settings revert to their default settings in the HardCopy project immediately after the HardCopy Timing Optimization Wizard is run.

If you are using LogicLock regions, Altera recommends you use the **Migration Only** setting in the HardCopy Timing Optimization Wizard to create the HardCopy design project. You should not compile your design automatically using the **Full Compilation** or **Migrate and Compile** options in the wizard. Open the HardCopy design project and verify that the LogicLock region properties meet your desired settings before compiling the HardCopy optimization project. LogicLock soft regions are

turned on by default in the HardCopy Stratix design. While this does allow the Fitter to place all logic in your design with fewer restrictions, it is not optimal for performance improvement in the HardCopy Stratix design.

Recommended LogicLock Settings for HardCopy Stratix Designs

Altera recommends the following LogicLock region settings for the HARDCOPY_FPGA_PROTOTYPE:

- Turn on **Reserve Unused Logic**
- Turn off **Soft Region**
- Select either **Auto** or **Fixed** as the **Size** (design-dependent)
- Select either **Floating** or **Locked** as the **Location** (design-dependent)

When using the **Reserve Unused Logic** setting in a design with high resource utilization (> 95% LE utilization), and a large number of LogicLock regions, the design may not fit in the device. Turning off **Reserve Unused Logic** in less critical LogicLock regions can help Fitter placement. The LEs allowed to float in placement and be packed into unused LEs of LogicLock regions may not be placed optimally after migration to the HardCopy Stratix device since they are merged with other LogicLock regions.

After running the HardCopy Timing Optimization Wizard, the LogicLock region properties are reset to their default conditions. This allows a successful and immediate placement of your design in the Quartus II software. You can further refine the LogicLock region properties for additional benefits.

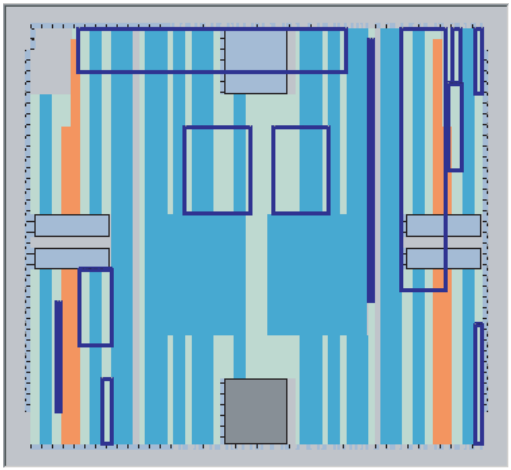
Altera recommends using the following properties for LogicLock regions in the HardCopy design project:

- Turn off **Soft Region**
- Select either **Auto** or **Fixed** as the **Size** after you are satisfied with the placement and timing result of a LogicLock region in a successful HardCopy Stratix compilation
- Select either **Floating** or **Locked** as the **Location** after you are satisfied with the placement and timing results
- **Reserve Unused Logic** is not applicable in the HardCopy Stratix device placement because logic array block (LAB) contents can not be changed after the HardCopy Timing Optimization Wizard is run

An example of a well partitioned design using LogicLock regions effectively for some portions of the design is shown in Figure 6–1. Only the most critical logic functions required are placed in LogicLock regions in order to achieve the desired performance in the HardCopy Stratix

device. The dark blue rectangles shown in Figure 6–1 are the user-assigned LogicLock regions that have fixed locations. In this example, the design needed to be constrained by LogicLock regions first inside the HARDCOPY_FPGA_PROTOTYPE with **Reserve Unused Logic** turned off in **Properties** in LogicLock regions. This selection allows the Quartus II software to isolate and compact the logic of these blocks in the HARDCOPY_FPGA_PROTOTYPE such that the placement is tightly controlled in the HardCopy Stratix device.

Figure 6–1. A Well Partitioned Design



In the example shown in Figure 6–1, once suitable locations were identified for LogicLock regions, the LogicLock region properties were changed from floating to locked. The Quartus II software can then reproduce their placement in subsequent compilations, while focusing attention on fixing other portions of the design.

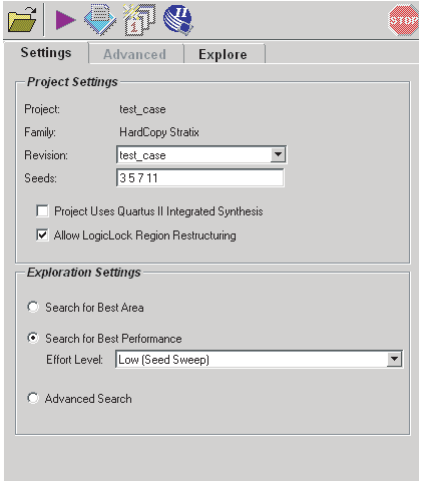
Using Design Space Explorer for HardCopy Stratix Designs

The DSE feature in the Quartus II software allows you to evaluate various compilation settings to achieve the best results for your FPGA designs. DSE can also be used in the HardCopy Stratix project after running the HardCopy Timing Optimization wizard.

Only some of the DSE settings affect HardCopy Stratix designs because HDL synthesis and physical optimization have been completed on the FPGA. No logic restructuring can occur after using the HardCopy Timing Optimization wizard. When you compile your design, the placement of LABs is optimized in the HardCopy Stratix device. To access the DSE GUI

in your open project in the Quartus II software, select **Launch Design Space Explorer** (Tools menu). An example of the DSE GUI and DSE Settings window for the HardCopy Stratix device is shown in Figure 6–2.

Figure 6–2. DSE Settings Window in the DSE GUI



Recommended DSE Settings for HardCopy Stratix Designs

The HardCopy Stratix design does not require all advanced settings or effort-level settings in DSE. Altera recommends using the following settings in DSE for HardCopy Stratix designs:

- In the **Settings** tab (Figure 6–2), make the following selections:
  - Under **Project Settings**, enter several seed numbers in the **Seeds** box. Each seed number requires one full compile of the HardCopy Stratix project.
  - Under **Project Settings**, select **Allow LogicLock Region Restructuring.**
  - Under **Exploration Settings**, select **Search for Best Performance,** and select **Low (Seed Sweep)** from the **Effort Level** menu.
- Turn on **Archive all Compilations** (Options menu).

After running DSE with the seed sweep setting, view the results and identify which seed settings produced the best compilation results. Use the archive of the identified seed, or merge the compilation settings and seed number from the DSE archived project into your primary HardCopy Stratix project.

Performance Improvement Example

With the design used for the performance improvement example in this section, the designer was seeking performance improvement on an HC1S30F780 design for an intellectual property (IP) core consisting of approximately 5200 LEs, 75,000 bits of memory, and two digital signal processing (DSP) multiplier accumulators (MACs). The final application needed to fit in a reserved portion of the HC1S30 device floorplan, so the entire block of IP was initially bounded in a single LogicLock region. The IP block was evaluated as a stand-alone block.

Initial Design Example Settings

The default settings in the Quartus II software version 4.2 were used, with the following initial constraints added:

- The device was set to the target Stratix FPGA device which is the prototype for the HC1S30F780 device:
  ```
  set_global_assignment -name DEVICE
  EP1S30F780C6_HARDCOPY_FPGA_PROTOTYPE
  ```
- A LogicLock region was created for the block to bound it in the reserved region.
- The LogicLock region properties were set to **Auto Size** and **Floating Location**, and **Reserve Unused Logic** was turned on:
  ```
  set_global_assignment -name LL_STATE FLOATING
  set_global_assignment -name LL_AUTO_SIZE ON
  set_global_assignment -name LL_RESERVED OFF
  set_global_assignment -name LL_SOFT OFF
  ```
- Virtual I/O pins were used for the ports of the core since this core does not interface to pins in the parent design, and the I/O pins were placed outside the LogicLock region and are represented as registers in LEs.

The initial compilation results yielded 65.30-MHz f_{MAX} in the FPGA. The block was constrained through virtual I/O pins and a LogicLock region to keep the logic from spreading throughout the floorplan.

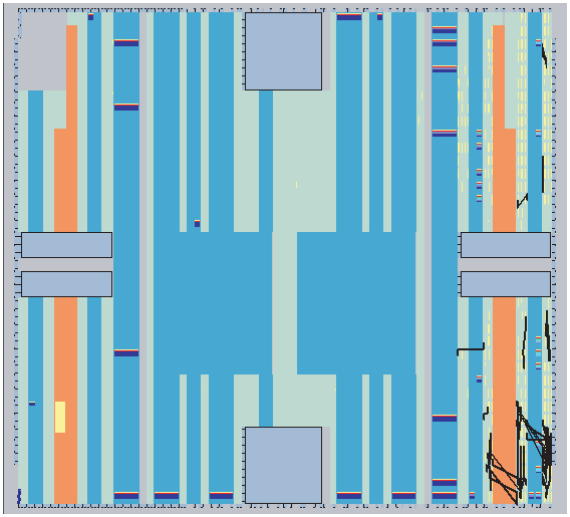
The initial compile-relevant statistics for this example are provided in Table 6–1.

Table 6–1. Initial Compilation Statistics

Result Type	Results
f_{MAX}	65.30 MHz
Total logic elements (LEs)	5,187/32,470 (15%)
Total LABs	564/3,247 (17%)
M512 blocks	20/295 (6%)
M4K blocks	16/171 (9%)
M-RAM blocks	0/2 (0%)
Total memory bits	74,752/2,137,536 (3%)
Total RAM block bits	85,248/2,137,536 (3%)
DSP block 9-bit elements	2/96 (2%)

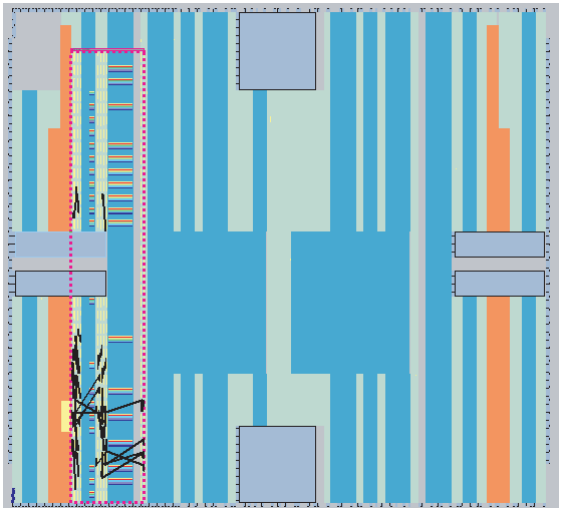
The design project was migrated to the HardCopy device using the HardCopy Timing Optimization wizard and was compiled. The default settings of the LogicLock region in a HardCopy Stratix project in the Quartus II software have the **Soft Region** option turned on. With this setting, the HardCopy Stratix compilation yields an f_{MAX} of 66.48 MHz, mainly due to the Fitter placement being scattered in an open design (Figure 6–3). Because the **Soft Region** is set to on, the LogicLock region is not bounded. This is not an optimal placement in the HardCopy Stratix design and is not the best possible performance.

Figure 6–3. HardCopy Stratix Device Floorplan with Soft Region On



To keep the LogicLock region contents bounded in the final placement in the HardCopy Stratix device floorplan, turn off the **Soft Region** option. After turning off the **Soft Region** option and compiling the HardCopy Stratix design, the result is an f_{MAX} of 88.14 MHz—a gain of 33% over the Stratix FPGA device performance. The bounded placement in the LogicLock region helps to achieve performance improvement in well-partitioned design blocks by taking advantage of the smaller die size and custom metal routing interconnect of the HardCopy Stratix device. The floorplan of the bounded LogicLock region is visible in Figure 6–4. In this figure, you can see the difference in disabling the Soft Region setting in the HardCopy Stratix design.

Figure 6–4. HardCopy Stratix Device Floorplan with Soft Region Off



Using Analysis and Synthesis Settings for Performance Improvement

After establishing the baseline for improvement for this design of 65.30 MHz FPGA/88.14 MHz HardCopy, you can gain additional performance improvement in the Stratix FPGA and HardCopy Stratix devices using the available features in the Quartus II software.

Changing the **Analysis & Synthesis Effort** from **Balanced** to **Speed** yields additional benefit in performance, but at the cost of additional LE resources. The Tcl command for this assignment is as follows:

```
set_global_assignment -name
STRATIX_OPTIMIZATION_TECHNIQUE SPEED
```

The relevant compilation results of the FPGA are provided in Table 6–2.

Table 6–2. Relevant Compile Results

Result Type	Results
f_{MAX}	68.88 MHz
Total logic elements	5,508/32,470 (16%)
Total LABs	598/3,247 (18%)
M512 blocks	20/295 (6%)
M4K blocks	16/171 (9%)
M-RAM blocks	0/2 (0%)
Total memory bits	74,752/2,137,536 (3%)
Total RAM block bits	85,248/2,137,536 (3%)
DSP block 9-bit elements	2/96 (2%)

Increasing the LE resources by 6% only yielded an additional 3 MHz in performance in the FPGA, without using additional settings. However, after migrating this design to the HardCopy Stratix design and compiling it, the performance did not improve over the previous HardCopy Stratix design compile, and was slightly worse in performance at 87.34 MHz. This shows that the Quartus II software synthesis was very effective with the **Synthesis Effort Level** set to **Balanced**, and there was only marginal improvement in the FPGA when this option was set to **Speed**.

The next settings activated in this example were the **Synthesis Netlist Optimizations** shown below in Tcl format for WYSIWYG synthesis remapping and gate-level retiming after synthesis mapping:

```
set_global_assignment -name
ADV_NETLIST_OPT_SYNTH_WYSIWYG_REMAP ON

set_global_assignment -name
ADV_NETLIST_OPT_SYNTH_GATE_RETIME ON
```

Making these settings in the FPGA while leaving **Analysis & Synthesis Effort** set to **Speed** yielded some additional improvement in the FPGA as shown in Table 6–3.

Table 6–3. Results of Analysis & Synthesis Effort Set to Speed

Result Type	Results
f_{MAX}	70.28 MHz
Total logic elements	5,515/32,470 (16%)
Total LABs	597/3,247 (18%)

The WYSIWYG resynthesis added a minimal increase in LEs over the speed setting, and the design performance improved by 2 MHz in the FPGA. Using the HardCopy Timing Optimization wizard to migrate the design to HardCopy and subsequently compiling the HardCopy Stratix design, we find that performance is not improved beyond previous compiles, with an f_{MAX} of 86.58 MHz.

The Quartus II software automatically optimizes state machines and restructures multiplexers when these settings are set to **Auto** in the **Analysis & Synthesis** settings. Changing these options from **Auto** usually does not yield performance improvement.

For example, changing the multiplexer restructuring and state machine processing settings from both set to **Auto**, to **On** and **One-Hot**, respectively, actually hurt performance, not allowing the Quartus II software to determine the optimization on a case-by-case basis. With these settings, the FPGA compiled to an f_{MAX} of 65.99 MHz, and the HardCopy Stratix design only performed at 83.77 MHz. For this design example, it is better to leave these settings to **Auto** as seen in the Tcl assignments in the "Using Fitter Assignments and Physical Synthesis Optimizations for Performance Improvement" section, and allow the Quartus II software to determine when to use these features.

Using Fitter Assignments and Physical Synthesis Optimizations for Performance Improvement

After exploring the Analysis & Synthesis optimization settings in the Quartus II software, you can use the Fitter Settings and Physical Synthesis Optimization features to gain further performance improvement in your Stratix FPGA and HardCopy Stratix devices. In this design example, multiplexer and state machine restructuring settings have been set to **Auto**, and the **Synthesis Optimization Technique** is set

for **Speed**. The **Fitter effort** is set to **Standard Fit (highest effort)**. The next features enabled are the **Physical Synthesis Optimizations** as seen in the Tcl assignments below and in Figure 6–5:

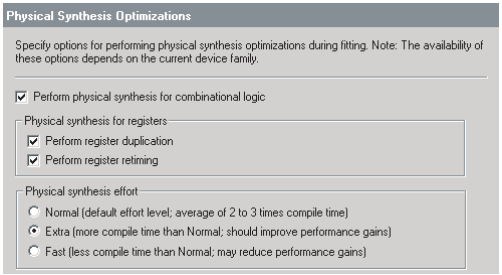
```
set_global_assignment -name
PHYSICAL_SYNTHESIS_COMBO_LOGIC ON

set_global_assignment -name
PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION ON

set_global_assignment -name
PHYSICAL_SYNTHESIS_REGISTER_RETIMING ON

set_global_assignment -name PHYSICAL_SYNTHESIS_EFFORT
EXTRA
```

Figure 6–5. Physical Synthesis Optimization Settings



The compiled design shows a performance increase in the FPGA, running at an f_{MAX} of 74.34 MHz, requiring additional LE resources as a result of the physical synthesis and logic duplication. In this example, you can see how performance can be increased in the Stratix FPGA device at the expense of additional LE resources, as this design's LE resources grew almost 12% over the beginning compilation. The compiled FPGA design's statistics are provided in Table 6–4.

Table 6–4. Compiled FPGA Design Statistics

Result Type	Results
f_{MAX}	74.34 MHz
Total logic elements	5,781/32,470 (17%)
Total LABs	610/3,247 (18%)

Running the HardCopy Timing Optimization wizard on this design and compiling the HardCopy Stratix project yields an f_{MAX} of 92.01 MHz, a 24% improvement over the FPGA timing.

Design Space Explorer

The available Fitter Settings produce an additional performance improvement. The DSE feature is used on the Stratix FPGA device to run through the various seeds in the design and select the best seed point to use for future compiles. This can often yield additional performance benefits as the Quartus II software further refines placement of the LEs and performs clustering of associated logic together.

For this design example, DSE was run with high effort (physical synthesis) and multiple placement seeds. Table 6–5 shows the DSE results. The base compile matches the fifth compile in the DSE variations, showing that the work already done on the design before DSE was optimal. The FPGA project was optimized before running DSE.

Table 6–5. DSE Results

Compile Point	Clock Period: CLK	Logic Cells
Base (Best)	13.451 ns (74.34 MHz)	5,781
1	13.954 ns	5,703
2	13.712 ns	6,447
3	14.615 ns	5,777
4	13.911 ns	5,742
5	13.451 ns	5,781
6	14.838 ns	5,407
7	14.177 ns	5,751
8	14.479 ns	5,827
9	14.863 ns	5,596
10	14.662 ns	5,605
11	14.250 ns	5,710
12	14.016 ns	5,708
13	13.840 ns	5,802
14	13.681 ns	5,788
15	14.829 ns	5,644

Additional correlation is seen inside the <*project*>**.dse.rpt** file, showing the summary of assignments used for each compile inside the Quartus II software. The base compile settings and the fifth compile settings show good correlation, as shown in Table 6–6. The `MUX_RESTRUCTURE` setting did not have any effect on the design performance. This may be due to an already efficient HDL coding for multiplexer structures, requiring no optimization.

Table 6–6. Base Compile and Fifth Compile Correlation

Setting	New Value	Base Value
PHYSICAL_SYNTHESIS_REGISTER_RETIMING	ON	ON
SEED	1	1
STATE_MACHINE_PROCESSING	AUTO	AUTO
MUX_RESTRUCTURE	OFF	AUTO
PHYSICAL_SYNTHESIS_COMBO_LOGIC	ON	ON
FITTER_EFFORT	STANDARD FIT	STANDARD FIT
AUTO_PACKED_REGISTERS_STRATIX	NORMAL	NORMAL
PHYSICAL_SYNTHESIS_REGISTER_DUPLICATION	ON	ON
ADV_NETLIST_OPT_SYNTH_GATE_RETIME	ON	ON
STRATIX_OPTIMIZATION_TECHNIQUE	SPEED	SPEED
PHYSICAL_SYNTHESIS_EFFORT	EXTRA	EXTRA

The information presented in Table 6–6 confirms that the FPGA Prototype device has been optimized as much as possible without manual floorplan adjustments.

Design Space Explorer for HardCopy Stratix Devices

Migrating this compiled design to the HardCopy Stratix project and compiling the HardCopy Stratix design optimization, results in a design performance of 92.01 MHz. The next task is to run DSE on the HardCopy Stratix project using **Low Effort (Seed Sweep)** in the **Exploration Settings**, and entering a range of seed numbers with which to compile the project.

The results of the DSE run with the **Seed Sweep** option are summarized in Table 6–7.

Table 6–7. DSE Results Run with Seed Sweep

Compile Point	Clock Period: CLK
Base (Best)	10.868 ns
1	11.710 ns
2	11.040 ns
3	10.790 ns
4	10.945 ns
5	11.154 ns
6	11.707 ns
7	11.648 ns
8	11.476 ns
9	11.423 ns
10	11.449 ns

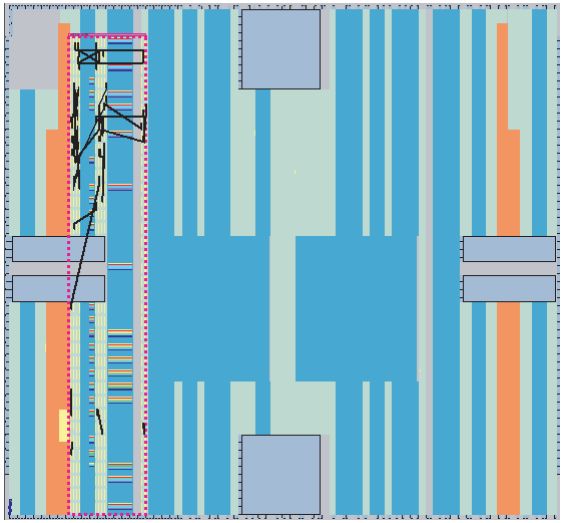
The results in Table 6–7 illustrate how the **Seed Sweep** option in DSE provides additional improvement in the HardCopy Stratix design, even after DSE has been run on the Stratix FPGA project. In this example, compile point 3 using seed value = 4 turns out to be slightly beneficial over other seeds in the Fitter Placement. The HardCopy Stratix device has an f_{MAX} of 92.71 MHz.

Back-Annotation and Location Assignment Adjustments

Another technique available for improving performance in the HardCopy Stratix design is manually adjusting placement and back-annotating location assignments from the placement results. These techniques should be one of the last steps taken for design optimization of HardCopy Stratix devices.

Observing the floorplan of the 92.71 MHz compile (Figure 6–6), the placement of the LogicLock region is stretched vertically, and additional improvement is possible if the aspect ratio of the LogicLock region is defined, and placement in it is refined.

Figure 6–6. Vertically Stretched LogicLock Region

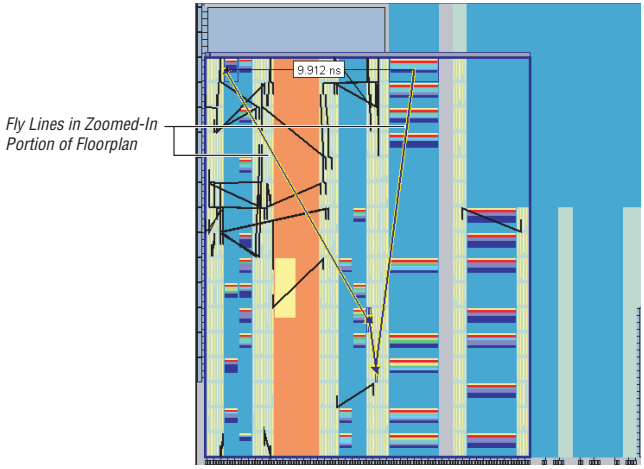


This floorplan would be better optimized if the LogicLock region had a more square shape, helping the paths that go from memory-to-memory, by containing the M4K and M512 memory blocks in a smaller space, and allowing LAB placement to be adjusted by the Fitter. In the HardCopy Stratix device, signals are routed between LABs, DSP blocks, and memory blocks using the customized metal layers. The reconfigurable routing tracks in the Stratix FPGA device limit the routing paths and delays between elements in the HardCopy Stratix device. This flexibility allows for aspect ratio changes in LogicLock regions, so the raw distance between points becomes the critical factor, and not the usage of available routing resources in the FPGA.

For the final placement optimization in this example, the LogicLock region was fixed in a square region that encompassed two columns of M4K blocks, four columns of M512 blocks, two columns of DSP blocks, and enough LABs to fit the remaining resources required. After compiling the design with these new LogicLock assignments, the performance increased to 93.46 MHz in the HardCopy Stratix device. The critical path and LogicLock region location can be seen in the zoomed-in area of the floorplan (Figure 6–7).

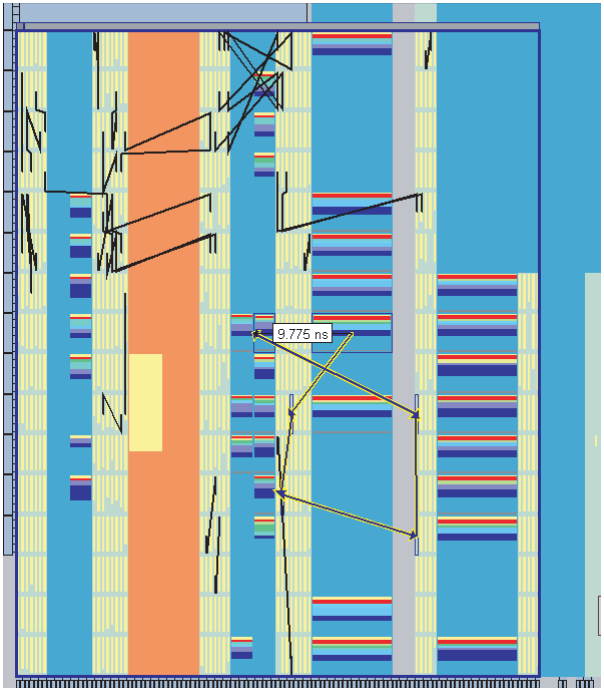
You can see in Figure 6–7 that the critical path shown is from an M4K block to an M512 block through several levels of logic. The placement of the memory blocks can be optimized manually, since the LogicLock region contains more memory blocks than necessary.

Figure 6–7. Critical Path and LogicLock Region



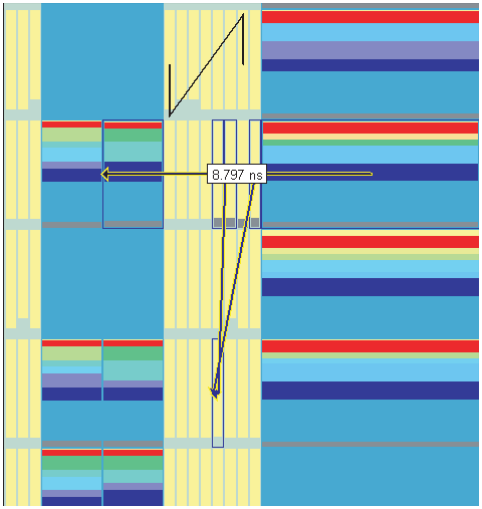
Using the critical path "fly lines" as a guide for placement optimization, manual location assignments were made for some of the M512 and M4K instances used in the design. The resulting compile improved the f_{MAX} to 94.67 MHz. The new critical path (Figure 6–8) shows how placement of all path elements are confined to a much smaller area. As a result, the routing distances and delays are smaller through the path.

Figure 6–8. New Critical Path



Examining this new critical path placement, you can see that there is room for further performance improvement through additional location assignments. The current slowest path is 9.775 ns of delay. Manually moving the LABs in this critical path and placing them between the M4K and M512 endpoints, and subsequently recompiling, shows improved results not only for this path, but for several other paths, as this path contained a major timing bottleneck. The critical path between this start and endpoint was reduced to 8.797 ns (Figure 6–9). However, the entire design only improved to 100.30 MHz because other paths are now the slowest paths in the design. This illustrates that fixing one major bottleneck path can raise the entire design performance since one high fanout node can affect multiple timing paths, as was the case in this example.

Figure 6–9. Improved Results



In summary, this design example started with 65.30 MHz in the Stratix FPGA device, and was improved to 74.34 MHz. It was then taken from the Stratix FPGA device compile and improved to 100.30 MHz in the HardCopy Stratix design, for a performance improvement of 35%.

Conclusion

Using performance-optimization techniques specifically for HardCopy Stratix devices can achieve significant performance improvement over the Stratix FPGA prototype device. Many of these changes must be incorporated up-front in the HARDCOPY_FPGA_PROTOTYPE so that your design is properly prepared for performance improvement after running the HardCopy Timing Optimization wizard.

The example discussed in this chapter demonstrates the process for performance improvement and various features in the Quartus II software available for use when optimizing your Stratix FPGA prototype and HardCopy Stratix device. It also demonstrates the importance of planning ahead for the HardCopy Stratix design implementation while continuing to work in the HARDCOPY_FPGA_PROTOTYPE design if you are going to seek performance improvement in the HardCopy Stratix device.

Document Revision History

Table 6–8 shows the revision history for this chapter.

Table 6–8. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
September 2008 v1.4	Updated chapter number and metadata.	—
June 2007 v1.3	• Updated the "Background Information" section. • Completed minor typographical updates.	—
December 2006 v1.2	Updated revision history.	—
March 2006	Formerly chapter 21; no content change.	—
October 2005 v1.1	• Updated graphics • Minor edits	—
July 2005 v1.0	Initial release of Chapter 21, Design Guidelines for HardCopy Stratix Performance Improvement.	—