7. Timing Constraints for HardCopy II Devices

Introduction

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

In contrast, a HardCopy® II design results in a mask-programmed, structured ASIC device. Timing problems may result in long design-change turn-around times and high NRE costs. To ensure a smooth transition through the Quartus® II software and back-end design in the Altera® HardCopy Design Center (HCDC), Altera strongly recommends that you use the TimeQuest timing analyzer provided with the Quartus II software and that you follow the timing considerations and timing constraint recommendations given in this chapter. Use of the TimeQuest timing analyzer for Design Review 2 (DR2) in the HardCopy II design flow will soon be mandatory.

The TimeQuest timing analyzer is a complete static timing analysis tool that you can use as a sign-off tool for Altera FPGAs and structured ASICs. As FPGA devices become denser and faster, they are the targets of complex designs and applications that previously were implemented in ASICs. These complex designs push the limits of the traditional Classic Timing Analyzer, affecting designer productivity. The Quartus II TimeQuest timing analyzer, in contrast, works well on complex designs. Its intuitive user interface, support of industry-standard Synopsys Design Constraints (SDC) format, and scripting capabilities all result in increased productivity and efficiency.

For more information on the features and capabilities of the TimeQuest timing analyzer, refer to the *TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

This chapter includes the following information:

- A description of timing-related differences between HardCopy II structured ASICs and Stratix II FPGAs
- Descriptions and a comparison of the TimeQuest timing analyzer and the Classic Timing Analyzer

■ An explanation of the use of timing constraints in the Quartus II software, including some of the important timing-related checks reported by the HardCopy II Advisor and Design Assistant

■ Timing constraint recommendations for your HardCopy II project and recommendations for handling legacy designs that use timing constraints not supported in the HardCopy II design flow

HardCopy II versus Stratix II Timing

The back-end design of your HardCopy II structured ASIC includes timing closure in accordance with the timing specification achieved in the Quartus II software for the Stratix II FPGA prototype and HardCopy II device. However, you should be aware that this does not mean that actual path timing in the Stratix II FPGA is duplicated in the HardCopy II device. In fact, because of the architectural differences between Stratix II and HardCopy II devices, you should expect that while internal and I/O path timing are within whatever timing constraints you applied, actual path delays are different.

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

■ The HardCopy II die is significantly smaller than its Stratix II counterpart

■ Coarse-grain adaptive logic modules (ALMs) in Stratix II devices are mapped to fine-grain HCell macros in HardCopy II devices

■ Design connections are implemented using custom metal routing in HardCopy II devices

■ HardCopy II devices contain no SRAM-configurable programmable connection points

■ Leaf sub-trees in HardCopy II global clock networks are custom routed

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

Internal Register-to-Register Timing

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

User-logic implementation in HardCopy II devices is more area efficient and often has improved timing when compared with the Stratix II FPGA. These advantages are the result of re-mapping the coarse-grain,

programmable ALMs in Stratix II devices to fine-grain HCell macros in HardCopy II devices. All ALM functions are re-mapped to HCells in HardCopy II devices. Using fine-grain HCells eliminates the need for the programmable routing multiplexers (MUXs) found inside the Stratix II ALM blocks. This reduces the number of levels of logic required to implement ALM functions from the Stratix II device. Consequently, the transport, or propagation, delays associated in the Stratix II FPGA with ALMs in register-to-register paths are smaller in the HardCopy II device.

The HardCopy II device does not require configuration SRAM, so die size is significantly smaller than for Stratix II counterpart devices. One effect of reduced die size is that overall routing length is shorter. In addition, HardCopy II devices use customization of metal layers 5 and 6 to implement user-logic connections. The fact that no configuration SRAM is required eliminates the need for SRAM-configurable routing switches and programmable connection points, all of which adversely affect timing. Therefore, overall, parasitic capacitance and resistance and crosstalk levels are often lower in the HardCopy II device, leading to faster connections than those found in the Stratix II FPGA.

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

The internal timing performance of digital signal processing (DSP) functions is similar in a Stratix II FPGA and its corresponding HardCopy II device. In Stratix II FPGAs, DSP functions are usually implemented in the embedded DSP blocks. These DSP blocks provide optimal area and performance for DSP functions. In HardCopy II devices, the same DSP functions are implemented in HCell DSP macros, which are designed to match the functionality and timing of the DSP blocks in Stratix II devices. However, the timing performance of paths between the DSP functions and other core logic is generally faster in the HardCopy II device than in the Stratix II FPGA.

RAM-block access time is similar in a Stratix II FPGA and its corresponding HardCopy II device. However, as for DSP functions, the timing performance of paths between the RAM blocks and other core logic is generally faster in the HardCopy II device than in the Stratix II FPGA.

I/O Path Timing

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

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

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

Clock Distribution Effects

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

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

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

■ Clock tree latency and clock insertion delay
■ Clock skew
■ Clock jitter
■ PLL compensation delays

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

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

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

PLL Characteristics

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

HardCopy II Timing Closure Methodology

To achieve timing closure for your HardCopy II structured ASIC, it is imperative that you use a complete set of accurate timing constraints throughout the flow. For the Stratix II FPGA prototype, although you may verify timing and functionality in hardware, it is essential that the design be compiled and verified in the Quartus II software using a complete set of timing constraints. These constraints feed forward to the HardCopy II revision of the project, and ultimately to the HardCopy Design Center (HCDC).

The back-end design of your structured ASIC in the HCDC ensures that it conforms to whatever timing constraints are satisfied in the Quartus II software. It is important to remember that while the Quartus II timing constraints are respected, the actual Stratix II FPGA prototype timing you observe in hardware is not duplicated in the HardCopy II structured ASIC. The timing differences between the Stratix II device and the HardCopy II structured ASIC are inconsequential as long as both are checked against a complete set of timing constraints.

HardCopy II Timing Closure Flow

HardCopy II timing closure methodology is comprehensive and includes both the TimeQuest timing analyzer and Classic Timing Analyzer in the Quartus II software, an interface to a third-party static timing analyzer, and FPGA-prototype timing verification in the hardware.

Altera recommends you use the TimeQuest timing analyzer. You can specify that the TimeQuest timing analyzer be used by the Quartus II software rather than the default Classic Timing Analyzer.
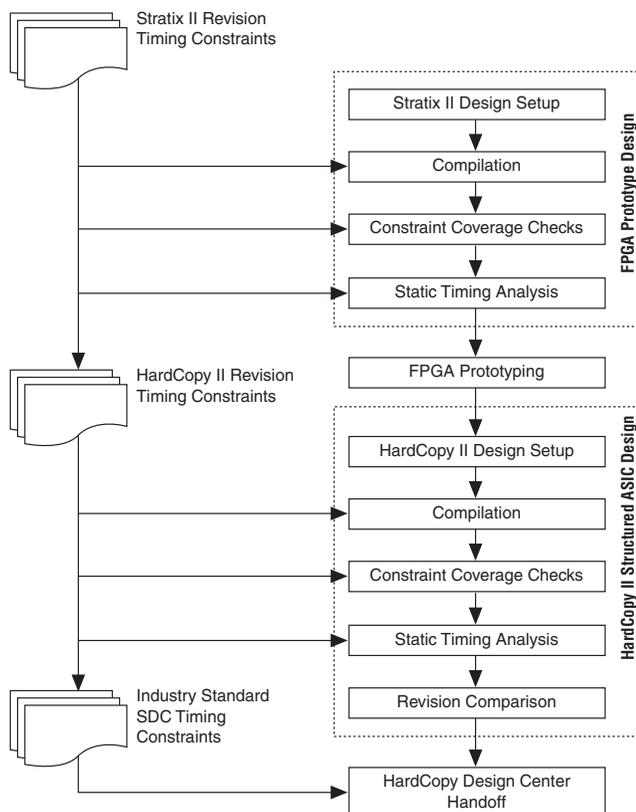
The TimeQuest timing analyzer validates the timing performance of all logic in your design using an industry-standard constraint, analysis, and reporting methodology. It provides powerful timing analysis features that enable thorough timing analysis of high-performance designs. The benefits of using TimeQuest for timing analysis include these features:

■ **Native SDC support**—You can leverage this powerful industry-standard timing constraint format to achieve a higher degree of productivity by using and reusing SDC- and Tcl-based scripts.

■ **Fast on-demand and interactive data reporting**—This feature saves time by allowing you to request more detailed timing analysis on critical paths only. A powerful GUI reports the timing analysis data in an intuitive graphical format that complements the fast, on-demand data reporting, further enhancing productivity.

Classic Timing Analyzer supports HardCopy II timing analysis. However, TimeQuest provides more powerful timing analysis features. Some Classic Timing Analyzer timing constraints may not be translated from the Quartus Setting file to SDC format constraints when the design is transferred to the HCDC, because translating these constraints is difficult and error-prone and often requires detailed analysis of the particular context in which the constraint is used.

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

Figure 7–1. Stratix II First Timing Closure Flow Note (1)

As you can see from Figure 7–1, timing constraints are used very early in the Quartus II design flow. During the Stratix II FPGA prototype compilation, these constraints are used as the timing target for timing-driven compilation. When the compilation is complete, the TimeQuest timing analyzer or Classic Timing Analyzer reports timing results for your design. Any failed timing reports mean that you must either modify your timing constraints, change your compile settings and recompile, or both. In addition, the timing constraint checkers in both TimeQuest and Classic Timing Analyzer report the unconstrained timing paths. See "Using the TimeQuest Timing Analyzer" on page 7–8 for details. For timing verification in third-party tools, the Quartus II

software can generate static timing analysis scripts for use in Synopsys PrimeTime tools. In addition, timing can be further verified in third-party, timing-driven simulation tools.

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

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

One of the final steps in the HardCopy II design flow in the Quartus II software is the revision comparison check. Part of this check compares timing constraints and settings between the Stratix II and HardCopy II revisions of the project. Any differences between the two are reported. If you change the timing constraints after completing Stratix II FPGA prototyping, the Revision Compare tool will report the change and you will be asked to waive this difference in the design review.

When your Quartus II design is transferred to the HCDC, it includes an industry-standard (SDC) version of the HardCopy II timing constraints. This version is the set of legal timing constraints for the design that include commands only from the **sdc** package in the Quartus II software. For the HardCopy II design flow, you may not use any commands except those in the **sdc** package in the Quartus II software. In addition, you must correct all timing constraints that generate warning messages in the Quartus II software.

For more detailed information on the Quartus II **sdc** package, refer to the **sdc** package section in the *Tcl Packages and Commands* chapter of the *Quartus II Scripting Reference Manual*.

Using the TimeQuest Timing Analyzer

The TimeQuest timing analyzer plays an integral part in the Quartus II HardCopy II timing closure flow, from the specification of timing constraints to the verification of design requirements.

The TimeQuest timing analyzer provides a number of timing checks during the HardCopy II design flow. The HardCopy II Advisor guides you to launch the TimeQuest timing analyzer for these timing checks and ensures that the design is fully constrained, as shown in Figure 7–2.

Figure 7–2. TimeQuest Timing-Related Settings in the HardCopy II Advisor
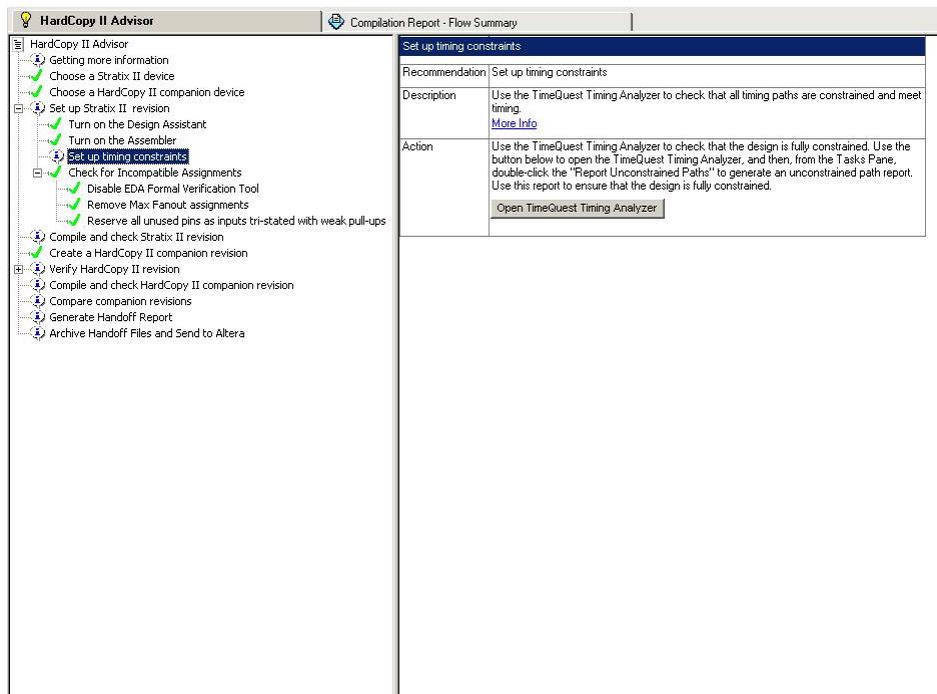
All timing paths must be fully constrained. The TimeQuest `report_ucp` command (or the TimeQuest GUI **Tasks** pane option **Report Unconstrained Paths**) generates a series of reports that detail all unconstrained paths in your design. These reports list unconstrained setup, hold, recovery, and removal timing paths in the design. You must correct any design errors the report shows you by applying additional constraints before running static timing analysis.

The TimeQuest timing analyzer supports most constraints in the SDC format for the HardCopy series of devices. The TimeQuest timing analyzer constraints are specified in commands from two Tcl packages in the Quartus II software. These packages are the **sdc** package and the **sdc_ext** package. The HardCopy II design flow requires that all timing constraints be specified in commands from the SDC Version 1.5

specification, as provided in the **sdc** package. Quartus II software returns warning messages in the early stage of the compilation for HardCopy II design flow if the SDC file contains any constraints that use commands from the TimeQuest extension to the SDC Version 1.5 specification, which are provided in the **sdc_ext** package. To enable a smooth transfer of the SDC file to the HCDC (HardCopy Design Center) for back-end design, you should avoid using commands and options from the **sdc_ext** package.

For more detailed information on the Quartus II **sdc** and **sdc_ext** packages, refer to the **sdc** package section in the *Tcl Packages and Commands* chapter of the *Quartus II Scripting Reference Manual* and to the *SDC and TimeQuest API Reference Manual.*

In addition to these timing-related checks, you should review the Quartus II timing report sections in the **Compilation Report** and resolve any timing violations that may be reported (Figure 7–3).

Figure 7–3. TimeQuest Unconstrained Timing Path Report
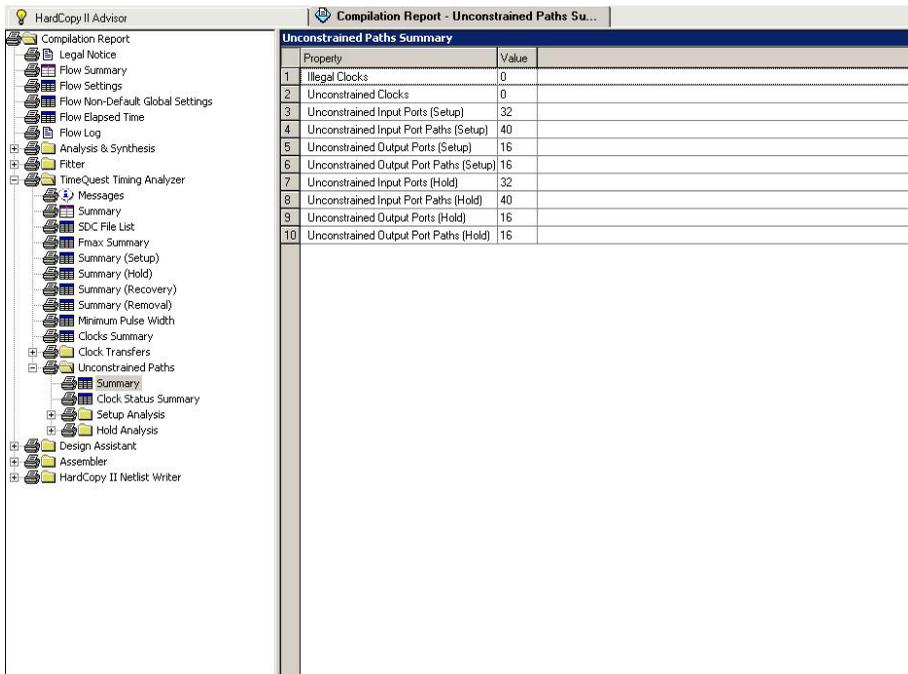
For more detailed information about the features and capabilities of the TimeQuest timing analyzer, refer to the *TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

Using Classic Timing Analyzer

Classic Timing Analyzer analyzes the delay of every design path and analyzes all timing requirements to ensure correct circuit operation. As part of the compilation flow, the Quartus II software automatically performs static timing analysis so that you do not need to launch a separate timing analysis tool. Classic Timing Analyzer checks every path in the design against your timing constraints for timing violations and reports results in the Timing Analysis reports, giving you immediate access to the data.

Quartus II Timing Related Checks and Settings

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

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

The HardCopy II Advisor advises on the correct Quartus II settings for timing analysis (Figure 7–4). These settings are necessary to ensure you generate accurate and complete timing reports. The list of settings includes the following:

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

In the Classic Timing Analysis flow, you must set the value of CUT_OFF_PATHS_BETWEEN_CLOCK_DOMAINS to **OFF**. Otherwise, the unconstrained path report (UCP report) will list all clock domain crossing paths as unconstrained. The report does not honor the ON setting, which cuts timing from clocks not originating from the same PLL.

Figure 7–4. Classic Timing-Related Settings in the HardCopy II Advisor
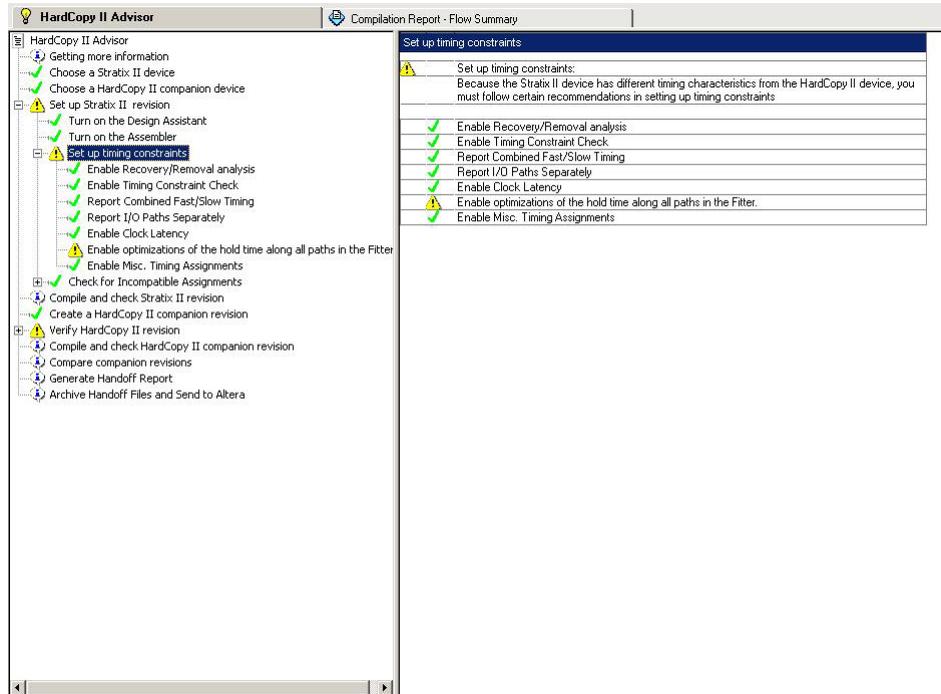
Classic Timing Analyzer, unlike the TimeQuest timing analyzer, supports some timing constraints that are incompatible with the HardCopy II design. In the HardCopy II Advisor, the **Remove Unsupported Global Timing Assignments** option and the **Remove Unsupported Instance Timing Assignments** option in the **Check for Incompatible Assignments** list (Figure 7–5) together list all the timing constraints that are incompatible with the HardCopy II design flow. These constraints are explained in "Unsupported HardCopy II Timing Constraints for Classic Timing Analyzer" on page 7–21.

Although Quartus II successfully completes timing analysis if you do not remove these timing constraints, it is very important that you correct all unsupported timing assignments before you transfer the HardCopy II design to the HCDC. Failure to remove these incompatible constraints may result in delays during back-end timing closure.

Figure 7–5. Classic Timing Analyzer Unsupported Timing Assignments in HardCopy II Advisor

The **Compilation Report** for both the Stratix II and HardCopy II revisions of your project includes a **Timing Constraints Check** section (Figure 7–6). This section reports all unconstrained paths based on the coverage provided by the timing constraints used in the design. You should examine this report and verify that all internal and I/O paths and all clock domains are constrained for both setup and hold checks.

Figure 7–6. Classic Timing Analyzer Constraints Check in Compilation Report
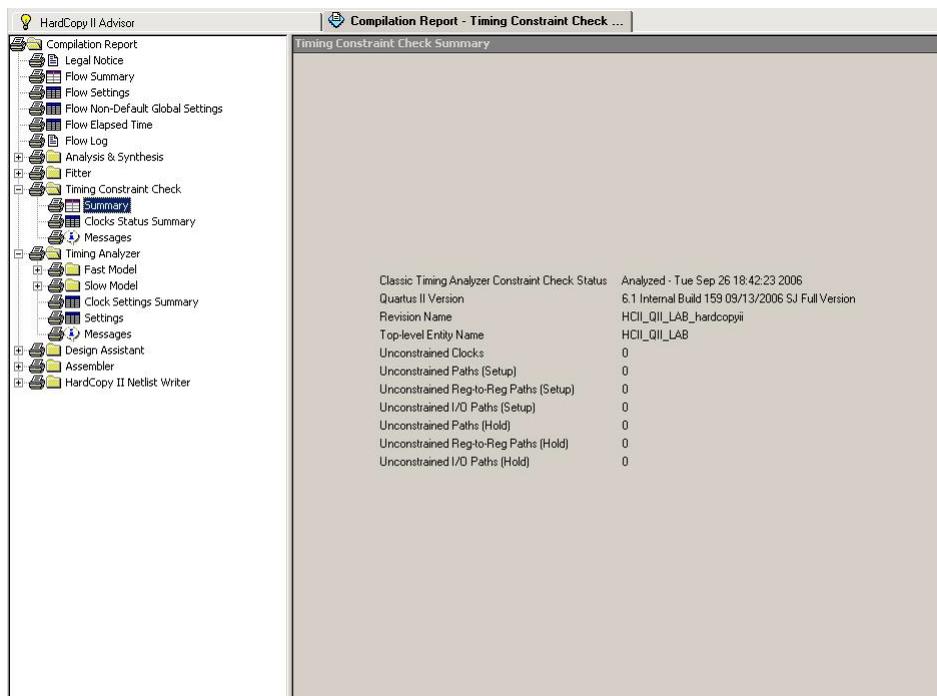
When using Classic Timing Analyzer, just as when using the TimeQuest timing analyzer, you should review the Quartus II timing report sections in the **Compilation Report** and resolve all reported timing violations.

Constraining Timing of HardCopy Series Devices

To ensure that the timing of the HardCopy device meets performance goals, the HardCopy Design Center runs static timing analysis on the design database. For this timing analysis to be meaningful, all timing constraints and timing exceptions that you applied to the design for the FPGA implementation, must also be used for the HardCopy implementation. If you did not use timing constraints or you used only partial timing constraints for the design, you must add constraints to

make the design fully constrained, and use the same constraints for both FPGA and HardCopy revisions in the flow. If you do not do this, you cannot determine whether the HardCopy series device meets the required timing of the end target system. The SDC format timing constraints can be generated using the Quartus II SDC File Editor which provides line numbering, syntax coloring, and call tips. You can enter timing constraints and exceptions directly or specify them from the Constraints menu. An example of the SDC commands is shown in the following section.

The following constraints must be included:

- Clock definitions
- Primary input port timing
- Primary output port timing
- Combinational timing
- Timing exceptions

For information on the SDC editor, refer to the *TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

For more information on timing constraints for the TimeQuest timing analyzer, refer to the *TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

For more information on timing assignments for Classic Timing Analyzer, refer to the *Classic Timing Analyzer* chapter in volume 3 of the *Quartus II Handbook*.

Clock Definitions

You can use these definitions to describe the parameters of all different clock domains in a design. Clock parameters that must be defined are frequency, time at which the clock edge rises, time at which the clock edge falls, clock uncertainty (for example: jitter, noise, and designed in timing margin), and clock name. Figure 7–7 illustrates the attributes.

Figure 7–7. Clock Attributes
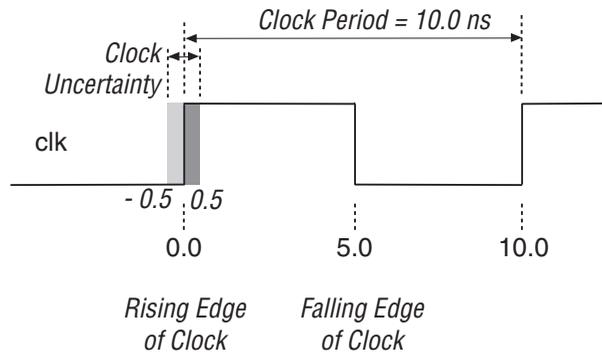
The clock settings for PLL clocks are derived automatically based on the PLL settings and reference clock characteristics. You can also override the default PLL clock settings for timing analysis by specifying clock settings for the input clock port on the PLL.

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

☞ Consult with your Altera Field Applications Engineer (FAE) or use MySupport regarding PLL clock uncertainty calculation for your design.

The SDC format provides a simple and easy method to constrain the simplest to the most complex designs. The following example illustrates the simplest SDC commands for a clock (port or pin) and for a generated clock at the PLL output pin for a design:

```
#Constrain the base clock
create_clock -period 10.000 [get_ports clkin]

#Constrain the PLL output clock
derive_pll_clocks
```

☞ Although derive_pll_clocks is in the **sdc_ext** package, it is the unique exception to the requirement that all timing constraints in the HardCopy II design flow must be in the **sdc** package. This command is automatically translated to the **sdc**-package command generated_pll_clock prior to transfer to the HCDC.

For a full list of available report APIs, refer to the *SDC and TimeQuest API Reference Manual*.

Primary Input Port Timing

You must specify the primary input port timing constraint for every primary input port in the design (and for the input path of every bidirectional port). The following two subsections describe how to constrain input port timing.

External Input Delay Specification

To constrain the input port timing, describe the external timing environment in terms of the maximum and minimum arrival times of the external signals that drive the primary input ports of the HardCopy series device or FPGA. Figure 7–8 shows the external timing constraint that drives the primary input port. The static timing analysis tool can use this external input delay time to check if there is enough time for the data to propagate to the internal nodes of the device. If there is not enough time, a timing violation occurs.

Figure 7–8. External Timing Constraint Driving a Primary Input Port
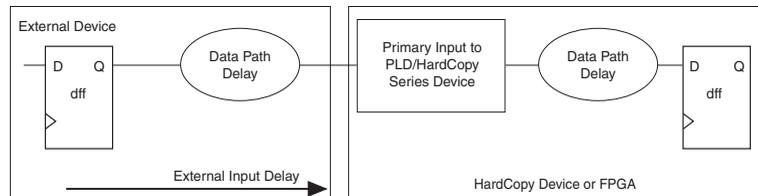
Internal Input Delay Specification

This approach describes the acceptable maximum on-chip delay for your design. For example, you can use this approach to describe the setup time of a primary input to any register in the design relative to a specific clock. Figure 7–9 shows a generic circuit with an on-chip setup-time constraint, which may be different for each clock domain. You may specify the minimum on-chip delay from any primary input port to describe input hold-time requirements.

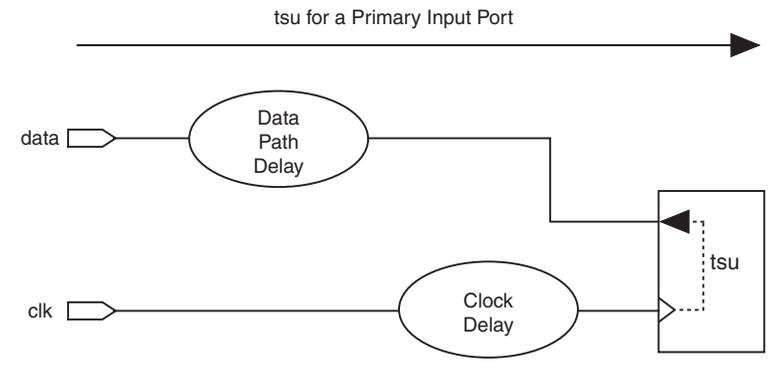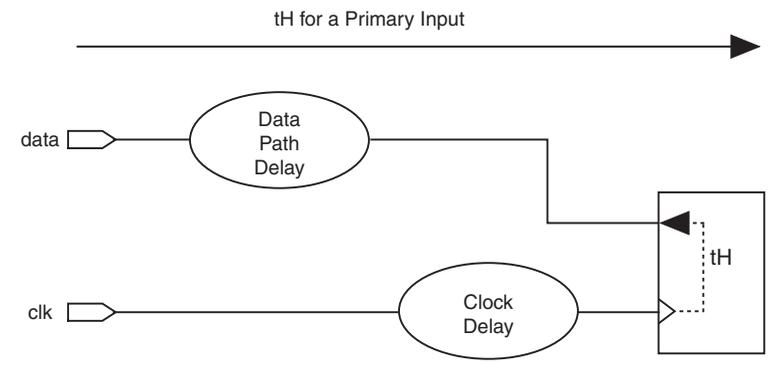*Figure 7–9. Internal Input Delay Specification (Setup)*

Figure 7–10 shows a generic circuit with an on-chip hold-time constraint.

Figure 7–10. Internal Input Delay Specification (Hold)


Primary Output Port Timing

You must specify the output port timing constraint for every primary output port in the design and for the output path of every bidirectional port. There are two ways to capture the output port timing, as described in the following two sections.

External Output Delay Specification

One way to capture output port timing is to describe the external timing environment, which is the maximum and minimum delay times of external signals that are driven by the primary output ports of the HardCopy series device. Figure 7–11 shows the external timing constraint driven by the primary output port. The static timing analysis tool uses this information to check that the on-chip timing of the output signals is within the desired specification.

Figure 7–11. External Timing Constraint for a Primary Output Port
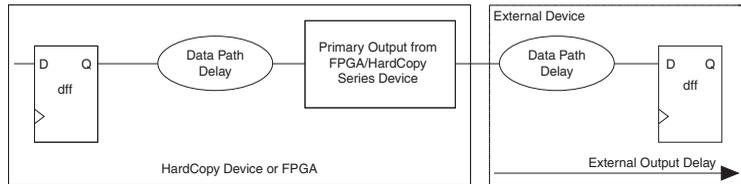
Internal Output Delay (Tco) Specification

This approach describes the acceptable maximum and minimum on-chip clock-to-output (T_{CO}) delay. For example, you can use this approach to describe the time it takes from the active edge of the clock to the data arriving at the primary output port. Figure 7–12 shows a generic circuit with an on-chip T_{CO} time constraint. In addition, there can be a minimum T_{CO} requirement.

Figure 7–12. On-Chip Clock-to-Output (T_{CO}) Time Constraint
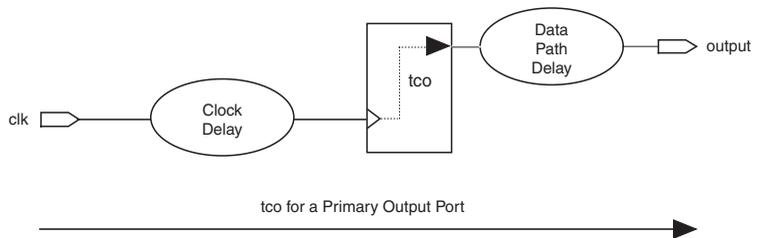
Combinational Timing

In combinational timing circuits, a path exists from a primary input port to a primary output port. This type of circuit does not contain any registers. Therefore, it does not require a clock for constraint specification. You only need the maximum and minimum delay from the primary input port to the primary output port to constrain the path for timing requirements. Figure 7–13 shows the placement requirement for a combinational delay arc constraint in a generic circuit.

Figure 7–13. Combinational Timing Constraint
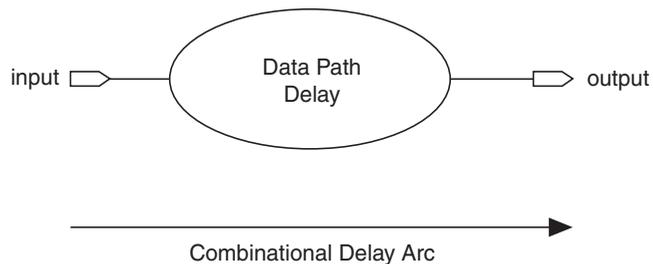
Timing Exceptions

Some circuit structures warrant special consideration. For example, you can ignore all timing paths between two clock domains when a design has more than one clock domain and the clock domains are not related. You can ignore all timing paths using the static timing analysis tool by specifying false paths for all signals that go from one clock domain to the other clock domain(s). Additionally, some circuits are not intended to operate in a single-clock cycle. These circuits require that you specify multi-cycle clock exceptions.

After capturing the information, the Altera HCDC directly checks all timing of the HardCopy series device before tape-out occurs. If any timing violations occur in the HardCopy series device due to overly aggressive timing constraints, Altera must fix them, or you must waive them.

Unsupported HardCopy II Timing Constraints for Classic Timing Analyzer

The Quartus II software supports a wide variety of complex timing constraints. When using Classic Timing Analyzer for HardCopy II design, however, some of these constraints are not translated to SDC format constraints when the design is transferred to the HCDC. The unsupported timing constraints for HardCopy II are listed below:

■ Clock enable multicycle paths
■ Inverted clocks
■ TSU, Th, TCO, and Min T_{CO}
■ Internal T_{PD}
■ Virtual clocks
■ Maximum clock and data skew
■ Maximum and minimum delay

If these constraints are used, you can still perform timing analysis in the Quartus II software and produce the correct results. However, when a HardCopy II archive for handoff is created, they will be ignored. The translation of Quartus II timing constraints to SDC constraints simply drops unsupported constraints; they do not feed forward to the HCDC. Any unsupported constraints in a design are listed under the **Incompatible Assignments** section in the HardCopy II Advisor (see Figure 7–5).

While it is possible to translate unsupported constraints to constraints that are supported, the process is difficult and error-prone, often requiring detailed analysis of the particular context in which the constraint is used.

For this reason, Altera recommends that you use timing constraints in the industry-standard SDC format with the TimeQuest timing analyzer or use only supported timing constraints for Classic Timing Analyzer from the start of your HardCopy II project. This approach avoids any translation or constraint coverage issues that may occur later in a project and the inevitable delay and risk that results.

In some cases, a HardCopy II project in the Quartus II software may already be using the unsupported constraints, and you may choose either to translate the existing, unsupported constraints, or replace them with a new set of constraints that use only the recommended HardCopy II timing assignments. In many cases, you may find it easier to rebuild the constraints rather than translate existing constraints. This is because of the ambiguous nature of many unsupported timing constraints, which often require additional information outside of the Quartus II software before the translation can be properly resolved. Verifying that the translations produce the same timing constraint coverage and the same timing analysis results can also be a time-consuming and error-prone exercise.

If you do wish to translate existing, unsupported timing constraints to recommended constraints, use Table 7–1 as a rough guide. It shows how values used in TCO, Th, TSU, and Min T_{CO} assignments normally convert to values used in recommended HardCopy II assignments. In the table, unsupported constraints are listed in the left hand column. Recommended constraints are listed along the top row. To use the table, cross-reference the unsupported constraints you wish to translate against a recommended constraint. The cross reference cell contains the conversion of the original, unsupported constraint value that should be used with the new, recommended constraint. It is very important to note that these translations are not valid in every design scenario.

Table 7–1. TSU, TH, TCO, and Minimum T_{CO} Timing Constraint Conversion *Notes (1)*, *(2)*, *(3)*, *(4)*, *(5)*

	setup_relationship	set_input_delay	hold_relationship	set_output_delay
TSU Req	TSU	-max \<TCK-TSU>		
Th Req		-min Th	-Th	
TCO Req	TCO			-max \<TCK-TCO>
Min T_{CO} Req			Min T_{CO}	-min <- Min T_{CO}>

Note to Table 7–1:
(1) TSU = value used in the TSU requirement assignment.
(2) TCO = value used in the TCO requirement assignment.
(3) Th = value used in the Th requirement assignment.
(4) Min T_{CO} = value used in Min T_{CO} requirement assignment.
(5) TCK = period of the clock for registers associated with the TSU and TCO requirements.

Conclusion

This chapter described timing considerations and Quartus II timing constraint recommendations for HardCopy II projects. By understanding these considerations and following the recommendations in your design, you ensure a smooth transition through the Quartus II software and subsequent transfer to the Altera HardCopy Design Center for the back-end design of your structured ASIC. Following the recommendations in this chapter will help ensure success in your HardCopy II project.

Document Revision History

Table 7–2 shows the revision history for this chapter.

Table 7–2. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
September 2008, v2.2	Updated chapter number and metadata.	—
June 2007, v2.1	Minor text edits.	—
December 2006 v2.0	Major updates for the Quartus II software version 6.1.0 ● Added information on TimeQuest timing analyzer, newly available in Quartus II software version 6.1.0, and recommended for use in HardCopy II design timing analysis. ● Added "Using the TimeQuest Timing Analyzer" section. ● Brought in "Constraining Timing of HardCopy Series Devices" section, previously in Chapter 22. ● Updated "HardCopy II Timing Closure Methodology" section. ● Added revision history.	A major update to the chapter, due to changes in the Quartus II software version 6.1 release, especially the inclusion of the TimeQuest timing analyzer; most changes were in the "HardCopy II Timing Closure Methodology" section, and the addition of the "Using the TimeQuest Timing Analyzer" and "Constraining Timing of HardCopy Series Devices"sections.
March 2006, v1.0	Added document to the HardCopy Series Handbook.	—