Using Timing Analysis in the Quartus Software

December 1999, ver. 1.0 Application Note 123

Introduction

As designs become more complex, the need for advanced timing analysis capability grows. Timing analysis measures the delay of every design path and reports the maximum system clock speed for the design. Because static timing analysis does not check design functionality, designers need to perform timing analysis together with simulation to verify overall design operation.

The Quartus™ software provides the features necessary to perform advanced timing analysis for today's System-on-a-Programmable-Chip™ designs. For example, during design compilation, the Quartus software can automatically activate the Static Timing Analyzer, removing the need to launch a separate timing analysis tool after each successful compilation. The Quartus Static Timing Analyzer also reports results in several separate tables and provides immediate and direct access to all timing analysis results.

This application note explains timing analysis basics and advanced features supported by the Quartus Static Timing Analyzer.

Timing Analysis Basics

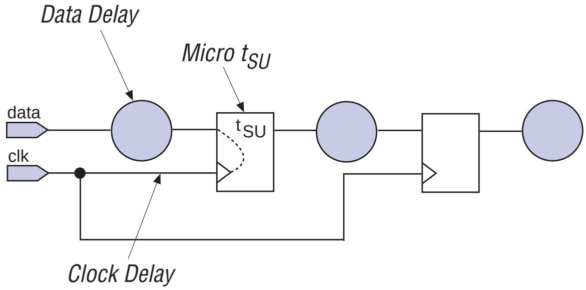
To perform comprehensive timing analysis, designers need to observe setup times, hold times, clock-to-output delays, clock skews, maximum clock frequencies, and slack times for their designs. Obtaining and analyzing this timing information lets designers validate circuit performance and identify possible timing violations. Undetected violations could present timing hazards and race conditions, both of which could lead to circuit failure. This section describes basic timing analysis measurements.

Clock Setup Time (t_{SU})

Data that feeds a register via its data or enable input(s) must arrive at the input pin before the register's clock signal is asserted at the clock pin. Clock setup time is the minimum length of time that this data must arrive before the active clock edge. Figure 1 shows a diagram of clock setup time.

A-AN-123-01

Figure 1. Clock Setup Time (t_{SU}) Diagram



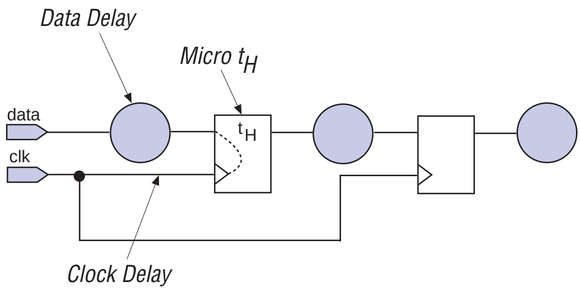
Micro $\mathbf{t_{SU}}$ is the intrinsic setup time of the register (i.e., it is an inherent characteristic of the register and is unaffected by the signals feeding the register). The following equation calculates the $\mathbf{t_{SU}}$ of the circuit shown in Figure 1.

$\mathbf{t_{SU}}$ = *Data Delay – Clock Delay + Micro* $\boldsymbol{t_{SU}}$

Clock Hold Time (t_H)

Data that feeds a register via its data or enable input(s) must be held at an input pin after the register's clock signal is asserted at the clock pin. Clock hold time is the minimum length of time that this data must be stable after the active clock edge. Figure 2 shows a diagram of clock hold time.

Figure 2. Clock Hold Time (t_H) Diagram



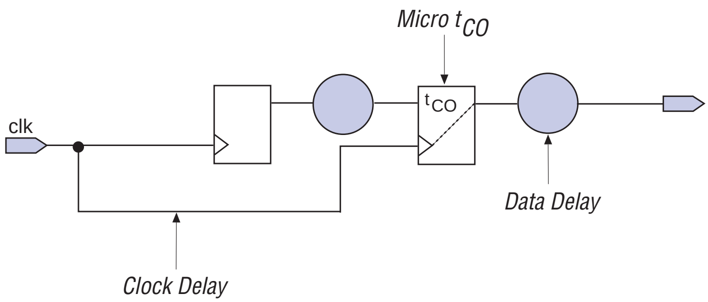
Micro $\mathbf{t_H}$ is the intrinsic hold time of the register. The following equation calculates the $\mathbf{t_H}$ of the circuit shown in Figure 2.

$\mathbf{t_H}$ = *Clock Delay – Data Delay + Micro* $\boldsymbol{t_H}$

Clock-to-Output Delay (t_{CO})

Clock-to-output delay is the time required for a clock signal to travel from an input pin through a register to an output pin. This time always represents an external pin-to-pin delay. Micro $\mathbf{t_{CO}}$ is the intrinsic clock-to-output delay of the register. Figure 3 shows a diagram of clock-to-output delay.

Figure 3. Clock-to-Output Delay (t_{CO}) Diagram



The following equation calculates the $\mathbf{t_{CO}}$ of the circuit shown in Figure 3.

$\mathbf{t_{CO}}$ = *Clock Delay* + *Micro* $\mathbf{t_{CO}}$ + *Data Delay*

Clock Skew

Clock skew is the difference in arrival time of a clock signal at two different registers. This timing difference occurs when two clock signal paths have different lengths. Clock skew is very common in designs that contain clock signals that are not routed globally. The Quartus software reports clock skews for all clocks, whether they are on pins or are internally derived clocks.

Maximum Clock Frequency (f_{MAX})

Maximum clock frequency is the fastest speed the design clock can run without violating internal setup and hold time requirements. The Quartus software can perform timing analysis on both single and multiple clock designs, reporting a design's internal and system $\mathbf{f_{MAX}}$. An internal f_{MAX} analysis calculates the register-to-register timing within the device. System $\mathbf{f_{MAX}}$ includes external delays to the device.

Calculating Internal f_{MAX}

To determine internal f_{MAX}, you must first calculate the circuit's clock period. The clock period depends on the data path delay, the clock skew between registers, the source register's clock-to-output time, and the destination register's setup time.

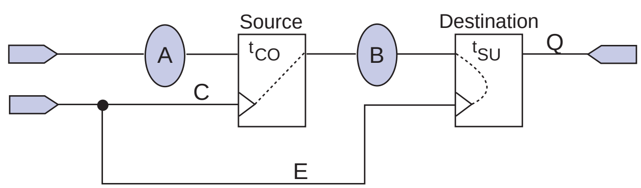
The Quartus software uses the following equations to calculate clock period and internal f_{MAX}. Register-to-register delay ($\mathbf{t_{RD}}$) in the clock period equation represents the data path delay between two registers.

Clock period = $\boldsymbol{t_{RD}}$ – *Clock Skew* + *Micro* $\boldsymbol{t_{CO}}$ + *Micro* $\boldsymbol{t_{SU}}$

Internal f_{MAX} = *1 / Clock Period*

Figure 4 shows a sample internal f_{MAX} diagram.

Figure 4. Quartus Internal f_{MAX} Diagram (Not Including External Delays)



The following equation calculates the internal f_{MAX} for the circuit shown in Figure 4.

Internal f_{MAX} = 1 / [*B* – (*E* – *C*) + *Source Micro* $\mathbf{t_{CO}}$ + *Destination Micro* $\mathbf{t_{SU}}$]

Calculating System f_{MAX}

The Quartus software calculates the system $\mathbf{f_{MAX}}$ by including external delays, assuming that all input pins are registered just before entering the device, and all output pins are registered just after leaving the device. The maximum $\mathbf{f_{MAX}}$ is restricted to the slowest clock period.

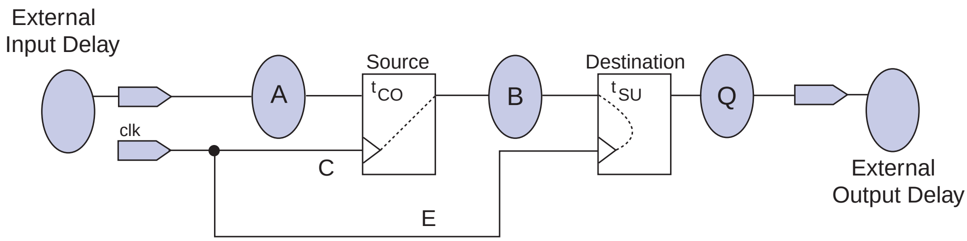
The Quartus software user can configure external board delays as the $\mathbf{t_{CO}}$ of an imaginary input register and the $\mathbf{t_{SU}}$ of an imaginary output register. In this scenario, the $\mathbf{t_{SU}}$ and $\mathbf{t_{CO}}$ values render the external conditions for an accurate system $\mathbf{f_{MAX}}$ analysis.

The Quartus software uses the following equation to calculate system $\mathbf{f_{MAX}}$.

System $\mathbf{f_{MAX}}$ = 1 / [*MAX (Input Clock Period, Clock Period, Output Clock Period)*]

Figure 5 shows a sample system $\mathbf{f_{MAX}}$ diagram.

Figure 5. Quartus System f_{MAX} (Including External Delays)



The following equations calculate the system $\mathbf{f_{MAX}}$ for the circuit shown in Figure 5.

$\mathbf{t_{SU}}$ (source) = Input Clock Period = *External Input Delay + A – C + Micro* $\boldsymbol{t_{SU}}$

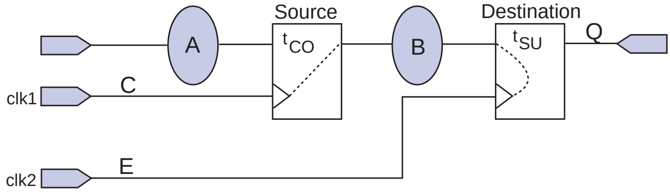
$\mathbf{t_{CO}}$ (destination) = Output Clock Period = *E + Micro* $\boldsymbol{t_{CO}}$ *+ Q + External Output Delay*

Multiclock f_{MAX} Calculation

Many complex designs contain paths between registers controlled by different clocks. The Quartus software can determine the $\mathbf{f_{MAX}}$ for these multiple clock designs.

Figure 6 shows a sample multiclock internal f_{MAX} diagram.

Figure 6. Quartus Multiclock Internal f_{MAX} Calculation Diagram

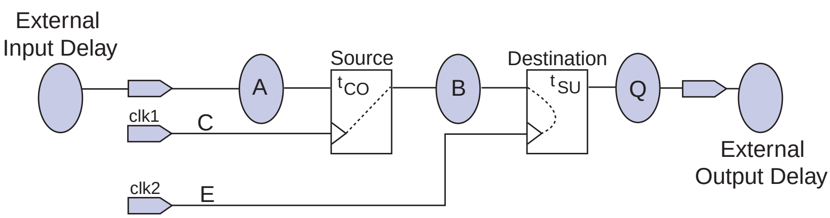


The following equation calculates the internal f_{MAX} for the circuit shown in Figure 6.

Internal f_{MAX} = 1 / [*B* – (*E* – *C*) + *Source Micro* t_{CO} + *Destination Micro* t_{SU}]

Figure 7 shows a sample multiclock system f_{MAX} diagram.

Figure 7. Quartus Multiclock System f_{MAX} Calculation Diagram



The following equations calculate the system f_{MAX} for the circuit shown in Figure 7.

t_{SU} (source) = Input Clock Period = *External Input Delay* + *A* – *C* + *Micro* t_{SU}

t_{CO} (destination) = Output Clock Period = *E* + *Micro* t_{CO} + *Q* + *External Output Delay*

The Quartus software allows designers to specify the desired f_{MAX} of both `clk1` and `clk2` for the design. The Quartus Static Timing Analyzer reports whether these clock periods meet or violate design operability under the specified f_{MAX} values.

Slack

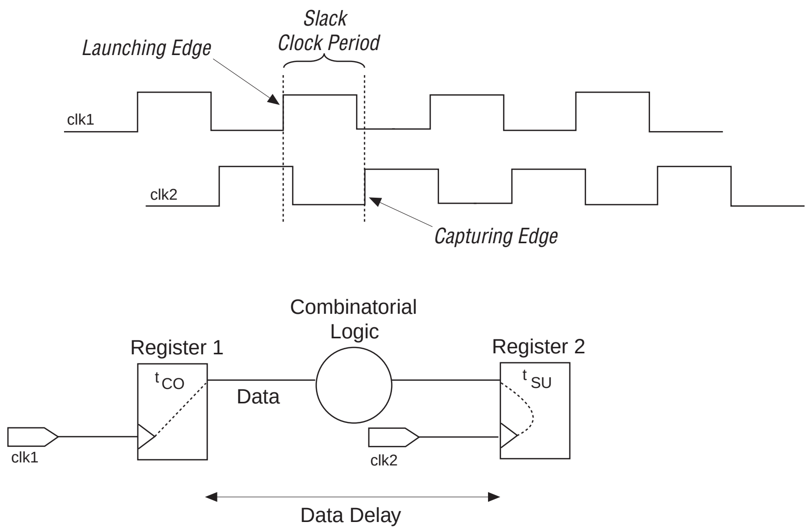
Slack is the margin by which a timing requirement (e.g., $\mathbf{f_{MAX}}$) was met or not met. A positive slack indicates that the circuit met the timing requirements; negative slack indicates that the design contains timing violations. The Quartus software determines slack with the following equations.

Slack = *Required clock period* – *Actual clock period*

Slack = *Slack clock period* – (*Micro* t_{CO} + *Data Delay* + *Micro* t_{SU})

Figure 8 shows a slack calculation diagram.

Figure 8. Slack Time Calculation Diagram



Advanced Features

The Quartus software can perform timing analysis of designs containing paths that cross multiple clock domains and designs that contain multicycle paths, offering designers greater control over design functionality. This section describes these advanced features.

For detailed instruction on how to use these or any of the Quartus Static Timing Analyzer features, see Quartus Help.

Multiple Clock Domains

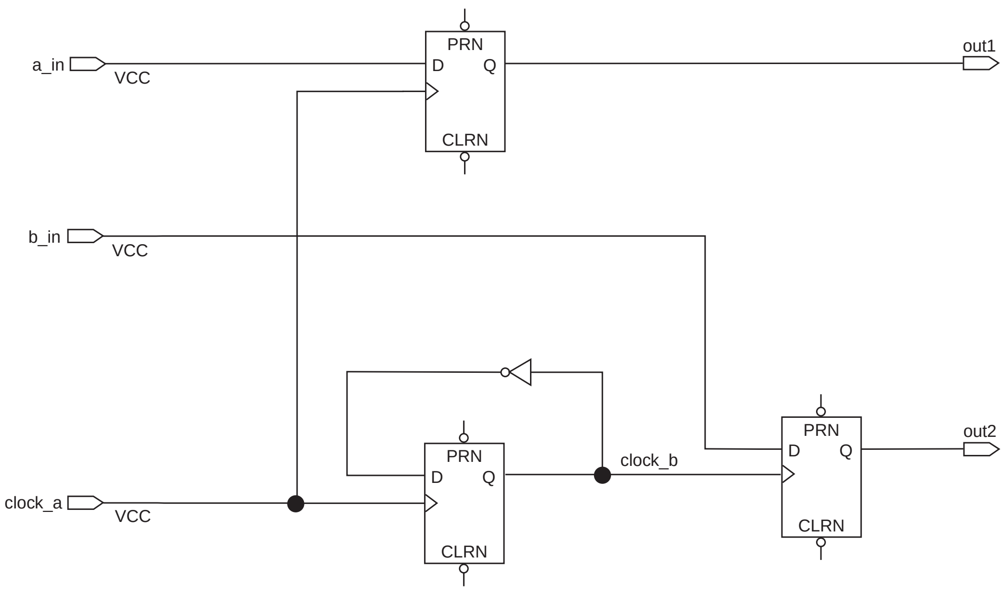
Multiple clock circuits are designs that have more than one clock driving a circuit. After the clocks are specified, the Quartus software analyzes timing for register-to-register paths controlled by different clocks, and the results are reported as slack. If the clocks are not specified, the Quartus software reports $\mathbf{f_{MAX}}$ for each and by default disregards any paths between unrelated clock domains.

To assign multiple clocks, you must define a base clock, specify a desired $\mathbf{f_{MAX}}$, and then define other clocks and their relationship, if any, to the base clock. These settings must then be assigned to the clock pins that supply the design's clock signals. Upon successful compilation, the Quartus Static Timing Analyzer automatically verifies circuit operability.

Derived Clocks

Derived clocks are signals feeding the clock ports of registers that were generated from internal device logic rather than from a device pin signal. Figure 9 shows a sample diagram of derived clocks. In this example, `clock_b` is a derived clock signal based on `clock_a`.

Figure 9. Diagram of Derived Clocks



The Quartus software supports static timing analysis of designs that contain derived clocks. The analysis can be set up by treating the derived clock as a relative clock, allowing the user to treat the design as a multiple-clock circuit.

Multicycle Paths

Multicycle paths are paths between registers that intentionally require more than one clock cycle to become stable. For example, a register may need to trigger a signal on every second or third rising clock edge. Figure 10 shows an example of a design with a multicycle path between the multiplier's input registers and output register.

Figure 10. Example Diagram of a Multicycle Path

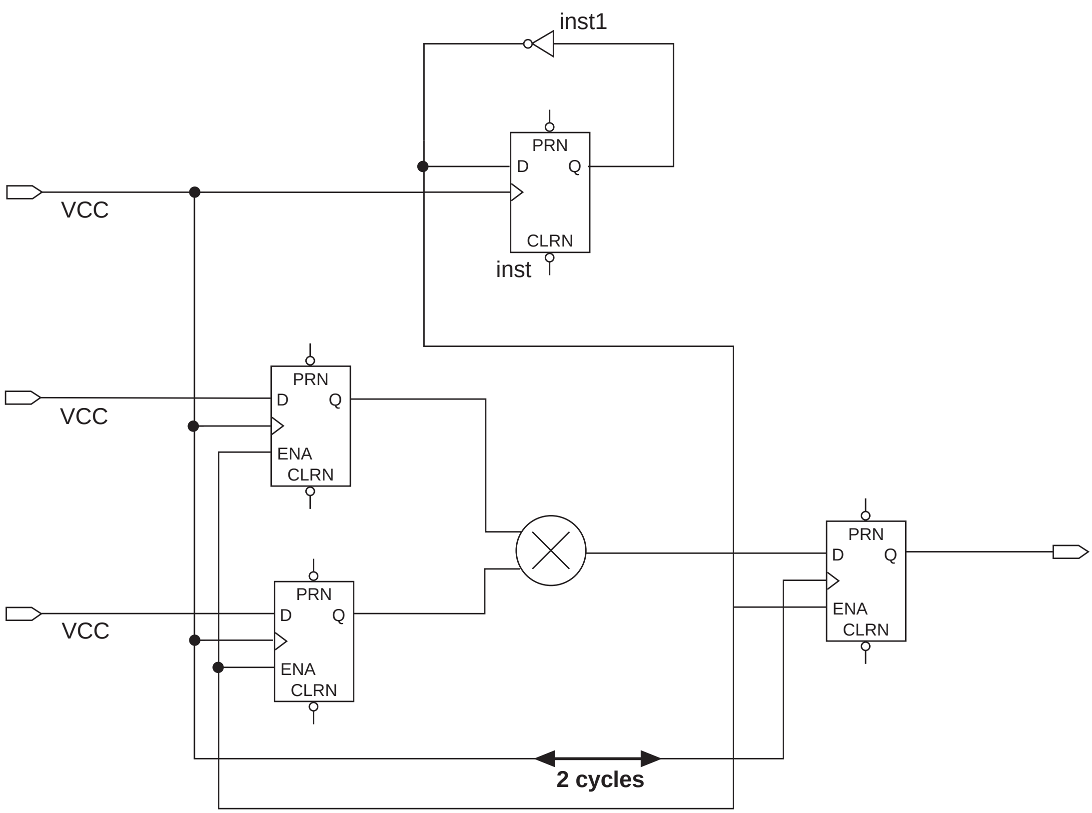


Figure 11 shows an example design that contains multicycle paths with multiple clocks in which the register is triggered every second or third cycle. Register 2 is controlled by a clock that is twice as fast as the signal feeding the enable port. Therefore, Register 2 operates at half the speed of `clkx2`.

Figure 11. Example of Multicycle Paths with Multiple Clocks

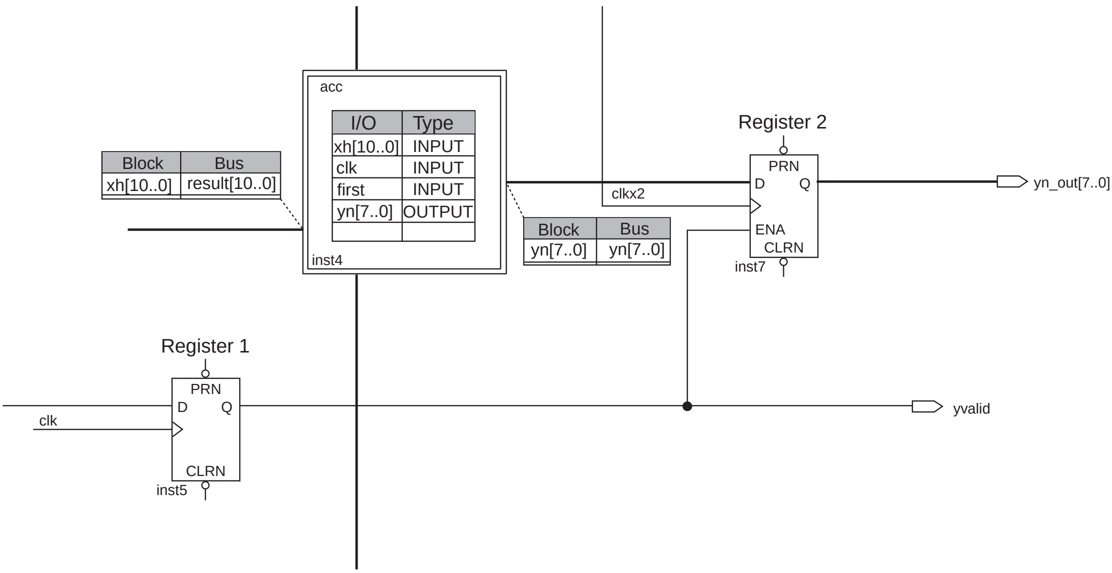
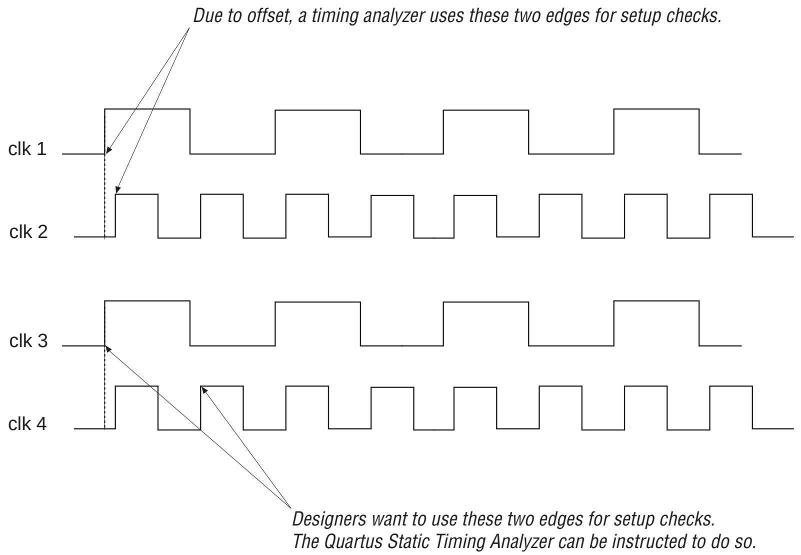


Figure 12 shows a timing diagram of a multicycle path between registers that exists in a design with multiple clocks that have a small offset between them.

Figure 12. Multicycle Paths with Offset Between Clocks



Designers can set multicycle paths in their designs to instruct the Quartus Static Timing Analyzer to adjust its measurements, thus avoiding incorrect setup time violation reports. These assignments can be made in the Quartus **Assignment Organizer**.

False Paths

Designs may contain paths between registers that are not relevant to the circuit's operation. These paths are referred to as false paths. To obtain accurate results, designers can configure the Quartus Static Timing Analyzer to ignore (i.e., cut) these paths during static timing analysis. Examples of false path signals are signals that are not used under normal operation (e.g., reset or test-mode). Designers can cut paths in the Quartus software using the **Timing Settings** (Project menu) or **Assignment Organizer** (Tools menu) dialog boxes.

Timing Wizard

The Quartus **Timing Wizard** allows designers to create, edit, and delete timing assignments, as well as specify overall circuit performance.

The **Timing Wizard** (Project menu) helps designers assign or change one or more of the following timing performance requirements and timing analysis settings. All timing assignments in the **Timing Wizard** are consolidated under one menu.

- Clock signals or overall circuit frequency ($\mathbf{f_{MAX}}$)
- Default system $\mathbf{t_{SU}}$, $\mathbf{t_{H}}$, $\mathbf{t_{CO}}$, and input-to-nonregistered-output time
- Default external delays to and from device pins
- Settings to control timing analysis and timing-driven compilation

Designers can also change these and other timing settings with the **Timing Settings** command (Project menu), the **Compiler Settings** command (Processing menu), and the **Assignment Organizer** command (Tools menu).

Conclusion

Evolving design methodologies and aggressive process technologies call for larger and higher-performance designs to be implemented in programmable logic devices (PLDs). This increasing design complexity initiates a need for enhanced timing analysis tools that aid designers in verifying design timing requirements. Without advanced timing analysis tools, designers risk circuit failure of their intricate multiclock and multipath designs. The Quartus Static Timing Analyzer incorporates a set of powerful, new timing analysis features that are critical in enabling System-on-a-Programmable-Chip designs.

ALTERA®

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
(888) 3-ALTERA
lit_req@altera.com

Altera, Quartus, and System-on-a-Programmable-Chip are trademarks and/or service marks of Altera Corporation in the United States and other countries. Altera acknowledges the trademarks of other organizations for their respective products or services mentioned in this document. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Copyright © 1999 Altera Corporation. All rights reserved.

nsai
I.S. EN ISO 9001

Printed on Recycled Paper.