4. Quartus II Timing Analysis

qii53004-2.0

Introduction

As designs become more complex, the need for advanced timing analysis capability grows. Static timing analysis is a method of analyzing, debugging, and validating the timing performance of a design. Timing analysis measures the delay of every design path and reports the performance of the design in terms of the maximum clock speed. However, it does not check design functionality and should be used together with simulation to verify the overall design operation.

The Quartus® II software provides the features necessary to perform advanced timing analysis for today's system-on-a-programmable-chip (SOPC) designs. During compilation the Quartus II software automatically performs timing analysis so that you don't have to launch a separate timing analysis tool after each successful compilation. The Quartus II Timing Analyzer reports timing analysis results in the compilation reports, giving you immediate access to this data.

This chapter explains the basic principles of static timing analysis, and the advanced features supported by the Quartus II Timing Analyzer using TCL scripts and the Quartus II graphical user interface (GUI).

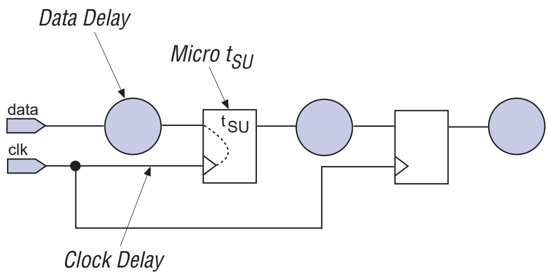
Timing Analysis Basics

A comprehensive timing analysis involves observing the setup times, hold times, clock-to-output delays, maximum clock frequencies, and slack times for the design. With this information you can validate circuit performance and detect possible timing violations. Undetected timing violations could result in incorrect circuit operation. This section describes the basic timing analysis measurements used by the Quartus II Timing Analyzer.

Clock Setup Time (t_{SU})

Data that feeds a register's data or enable inputs 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 data must be stable before the active clock edge. Figure 4–1 shows a diagram of clock setup time.

Figure 4–1. Clock Setup Time (t_{SU})



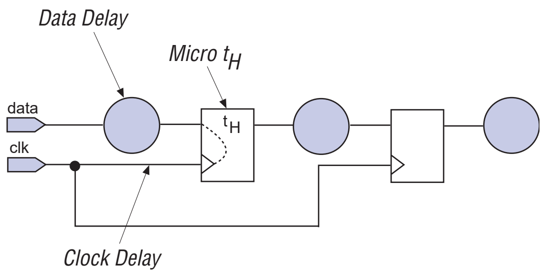
Micro t_{SU} is the internal setup time of the register (i.e., it is a characteristic of the register and is unaffected by the signals feeding the register). The following equation calculates the t_{SU} of the circuit shown in Figure 4–1.

t_{SU} = Data Delay – Clock Delay + Micro t_{SU}

Clock Hold Time (t_H)

Data that feeds a register via its data or enable inputs 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 4–2 shows a diagram of clock hold time.

Figure 4–2. Clock Hold Time (t_H)



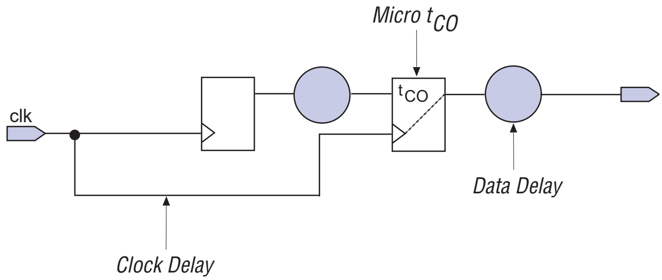
Micro t_H is the internal hold time of the register. The following equation calculates the t_H of the circuit shown in Figure 4–2.

t_H = Clock Delay – Data Delay + Micro t_H

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

Clock-to-output delay is the maximum time required to obtain a valid output at an output pin fed by a register, after a clock transition on the input pin that clocks the register. Micro t_{CO} is the internal clock-to-output delay of the register. Figure 4–3 shows a diagram of clock-to-output delay.

Figure 4–3. Clock-to-Output Delay (t_{CO})



The following equation calculates the t_{CO} of the circuit shown in Figure 4–3.

t_{CO} = Clock Delay + Micro t_{CO} + Data Delay

Pin-to-Pin Delay (t_{PD})

Pin-to-pin delay (t_{PD}) is the time required for a signal from an input pin to propagate through combinational logic and appear at an external output pin.

In the Quartus II software, you can also make t_{PD} assignments between an input pin and a register, a register and a register, and a register and an output pin.

Maximum Clock Frequency (f_{MAX})

Maximum clock frequency is the fastest speed at which the design clock can run without violating internal setup and hold time requirements. The Quartus II software performs timing analysis on both single and multiple clock designs.

Slack

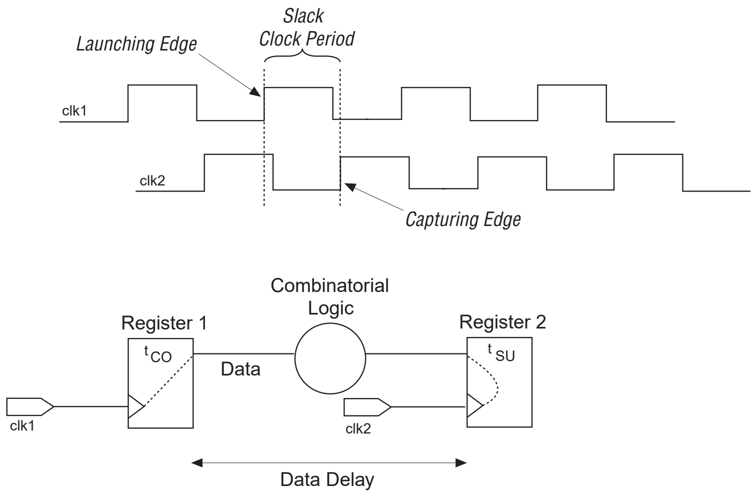
Slack is the margin by which a timing requirement (e.g., f_{MAX}) is met or not met. Positive slack indicates the margin by which a requirement is met. Negative slack indicates the margin by which a requirement was not met. The Quartus II software determines slack with the following equations.

Slack = Required clock period – Actual clock period

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

Figure 4–4 shows a slack calculation diagram.

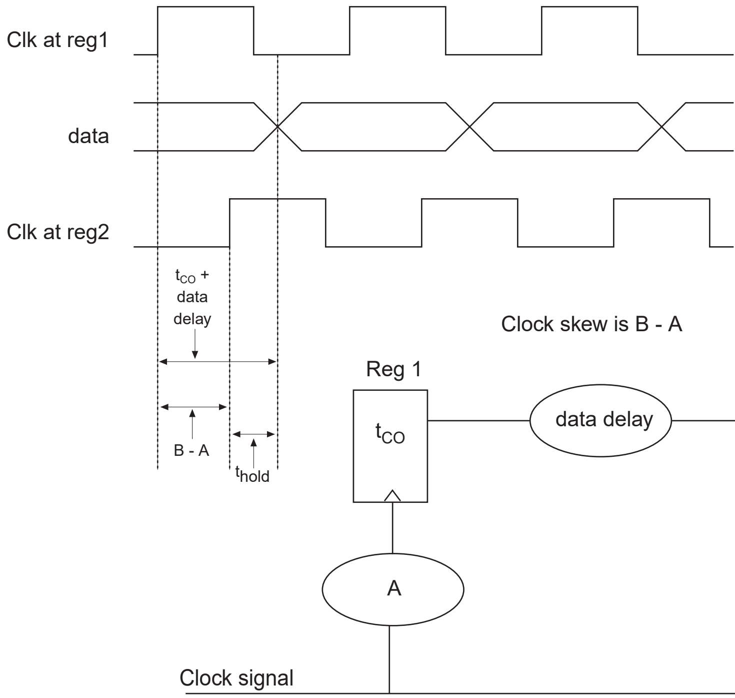
Figure 4–4. Slack Calculation Diagram



Hold Time Slack

Hold time slack is the margin by which the minimum hold time requirement is met or not met for a register-to-register path (Figure 4–5). Data is required to remain stable after the rising edge of a destination register's clock for at least the time equal to the micro hold time of the destination register. The primary cause of a hold time violation is excessive clock skew (B - A). As long as the data delay is greater than clock skew (B - A), no hold time violation occurs. Since the Quartus II software only reports hold time slack for paths that have hold time violations, only negative slacks are reported.

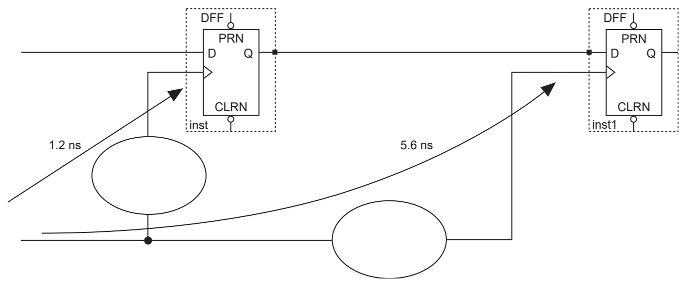
Figure 4–5. Hold Time Slack



Clock Skew

Clock skew is the difference in arrival time of a clock signal at two different registers (Figure 4–6). Clock skew occurs when two clock signal paths have different lengths. Clock skew is common in designs that contain clock signals that are not routed globally. The Quartus II Timing Analyzer reports clock skew for all clocks within the design.

Figure 4–6. Clock Skew



Executing Tcl Script-Based Timing Commands

You can make timing assignments, perform timing analysis, and analyze results in the Quartus II software GUI or with Tcl commands. You can use simple Tcl commands to perform customized timing reporting, and you can write scripts with advanced timing analysis commands to perform complex timing analysis and reporting.

You can use the command-based timing analyzer in an interactive shell mode where you can run timing analysis Tcl scripts.

To run the timing analyzer in interactive shell mode, type the following command:

```
quartus_tan -s
```

To run a Tcl script, type the following command:

`quartus_tan -t` <*tcl file*>

The following commands are frequently needed for executing timing-related scripts:

- Package require `::quartus::`<*advanced_timing*> (Different packages are required for a different set of commands.)
- `project_open` <*project_name*> (Open the project in the project directory.)
- `create_timing_netlist` (Timing information is created in the memory for analysis.)
- `project_close` (This command should be executed at the end of every script.)

The remainder of this chapter includes Tcl command examples for making timing assignments and performing timing analysis. Refer to the Quartus II Command-Line and Tcl API Help for complete information about the above commands, other Tcl commands related to timing analysis and reporting, and the complete Tcl command reference.

To run the Tcl API Help, type the following command:

```
quartus_sh --qhelp
```

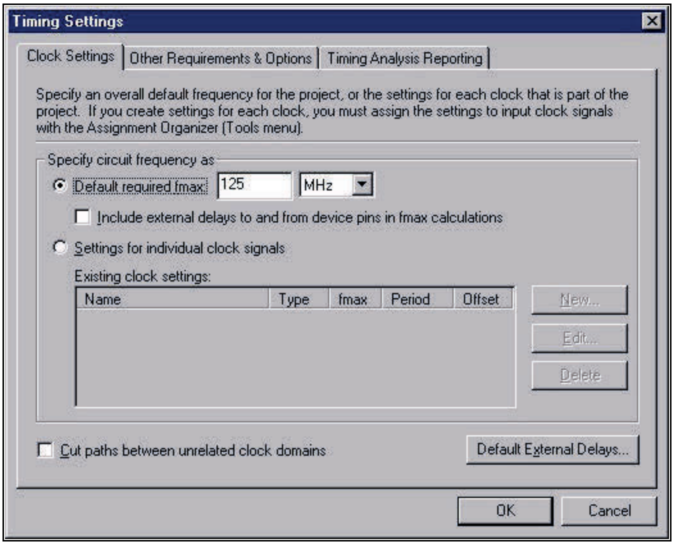
Setting up the Timing Analyzer

You can make certain timing assignments globally for a project, and you can make timing assignments to individual entities in a project. If a project has global and individual timing assignments, the individual timing assignments take precedence over the global timing assignments.

Setting Global Timing Assignments

You can make global timing assignments in the **Timing Requirements & Options** page of the **Timing Settings** dialog box (Assignments menu), shown in Figure 4–7.

Figure 4–7. Timing Settings Dialog Box



You can set global t_{SU}, t_{CO}, and t_{PD} requirements, as well as minimum t_H, t_{CO}, and t_{PD} requirements. You can set a global f_{MAX} requirement, or assign timing requirements and relationships for individual clocks.

For more information about path cutting options in the **Timing Requirements & Options** page, see "False Paths" on page 4–28.

Specifying Individual Clock Requirements

Apply clock requirements to each clock in your design. You can define clocks as absolute clocks (independent of other clocks) or derived clocks (dependent on other clocks). To define an absolute clock, you must specify the required f_{MAX} and the duty cycle. A derived clock is based on a previously defined clock. For a derived clock, you can specify the phase shift, offset, and multiplication and division factors relative to the absolute clock. You must define clock requirements and relationships with the Timing Wizard or by clicking **Clocks** in the **Timing**