Using TimeQuest Timing Analyzer

For Quartus II 13.0

1 Introduction

This tutorial provides a basic introduction to TimeQuest Timing Analyzer. It demonstrates how to set up timing constraints and obtain timing information for a logic circuit.

The reader is expected to have the basic knowledge of Verilog hardware description language, as well as the basic use of the Altera Quartus II CAD software.

Contents:

- Introduction to timing analysis
- Setting up Quartus II to use TimeQuest
- Using TimeQuest
- Setting Up Timing Constraints

2 Background

Timing analysis is a process of analyzing delays in a logic circuit to determine the conditions under which the circuit operates reliably. These conditions include, but are not limited to, the maximum clock frequency (f_{max}) for which the circuit will produce a correct output. A simple example of the maximum clock frequency computation is shown in Figure 1.

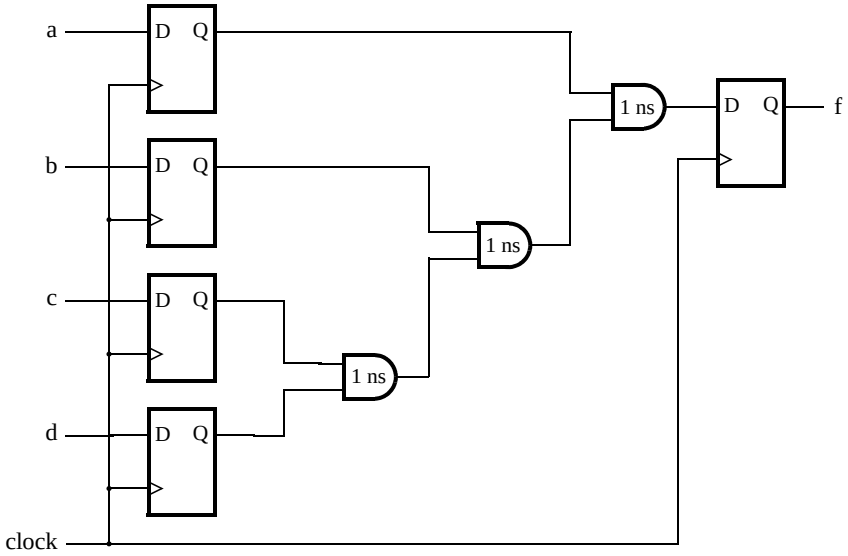

Figure 1. A example for timing analysis.

In this example, flip-flops on the left-hand side drive a combinational circuit that generates an output that is later stored in the flip-flop on the right-hand side. To operate correctly, the clock frequency is limited by the delay on the longest path in the circuit. If we assume that the clock-to-Q and setup times for each flip-flop are 1 ns, and the delay in each gate is 1 ns, then the maximum clock frequency for this circuit is:

$$f_{max} = \frac{1}{t_{cq} + 3 \times t_{and} + t_{su}} = \frac{1}{5 \text{ ns}} = 200 \text{ MHz}$$

Computing f_{max} is a basic function of a timing analyzer. The timing analyzer can be used to guide Computer-Aided Design tools in the implementation of logic circuits. For example, the circuit in Figure 1 shows an implementation of a 4-input function using 2-input AND gates. Without any timing requirements, the presented solution is acceptable. However, if a user requires the circuit to operate at a clock frequency of 250 MHz, the above solution is inadequate. By placing timing constraints on the maximum clock frequency, it is possible to direct the CAD tools to seek an implementation that meets those constraints. As a result, the CAD tools may arrive at a solution shown in Figure 2. The new circuit has $f_{max} = 250$ MHz and thus meets the required timing constraints.

In this tutorial, we demonstrate how to obtain timing information and how to set timing constraints using TimeQuest timing analyzer.

IMPORTANT: The example design provided with this tutorial contains exactly one clock signal. When multiple clock signals are present, the initial behavior of the analyzer differs slightly. We recommended that readers com-

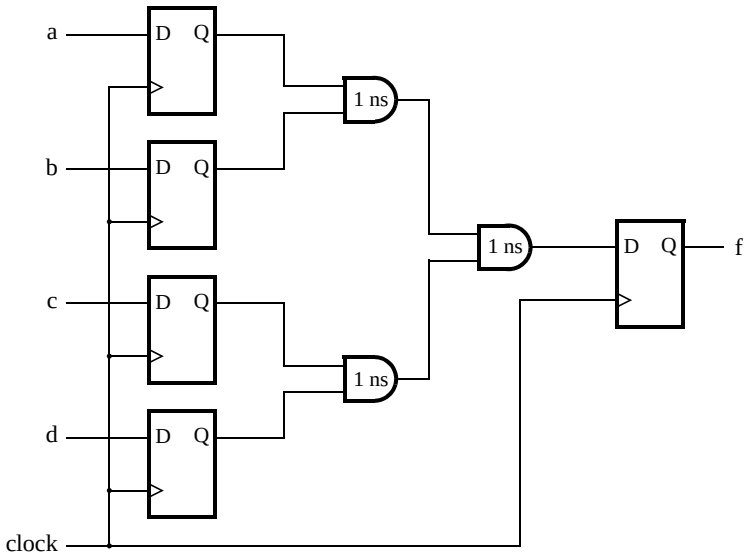

Figure 2. Functionally equivalent circuit with a different logic structure.

plete the tutorial using the provided example design before proceeding to use TimeQuest on their own design. We elaborate on this issue in Section 6.

3 Design Example

As an example we will use an adder that adds three 8-bit numbers and produces a sum output. The inputs are *A*, *B*, and *C*, which are stored in registers *reg_A*, *reg_B* and *reg_C* at the positive edge of the *clock*. The three registers provide inputs to the adder, whose result is stored in the *reg_sum* register. The output of the *reg_sum* register drives the output port *sum*. The diagram of the circuit is shown in Figure 3.

The Verilog source code for the design is given in Figure 4. Note that the ”synthesis keep” comment is included in this code. This comment is interpreted as a directive that instructs Quartus II software to retain the specified nodes in the final implementation of the circuit and keep their names as stated. This directive will allow us to refer to these nodes in the tutorial.

To begin the tutorial open the example project. It is available in the *add_three_numbers* directory provided with this tutorial.

Compile the example circuit to see the results of timing analysis. These results will be available in the compilation report, once the design is compiled. In this tutorial we will use the TimeQuest Graphical User Interface to inspect the timing analysis results.

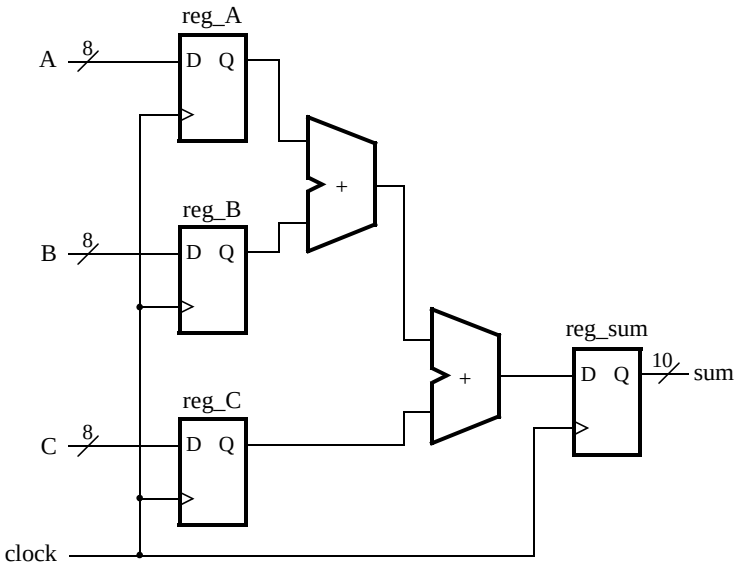

Figure 3. Diagram of the example circuit.

```
1.  module add_three_numbers(clock, A, B, C, sum);
2.      input clock;
3.      input [7:0] A,B,C;
4.      output [9:0] sum;
5.
6.      // Registers
7.      reg [7:0] reg_A, reg_B, reg_C /* synthesis keep */;
8.      reg [9:0] reg_sum /* synthesis keep */;
9.
10.     always @(posedge clock)
11.     begin
12.         reg_A <= A;
13.         reg_B <= B;
14.         reg_C <= C;
15.         reg_sum <= reg_A + reg_B + reg_C;
16.     end
17.     assign sum = reg_sum;
18. endmodule
```

Figure 4. Verilog code for the example circuit.

4 Using TimeQuest

To start TimeQuest, select Tools > TimeQuest Timing Analyzer from the main menu. The TimeQuest window, shown in Figure 5, will appear.

4.1 TimeQuest Graphical User Interface

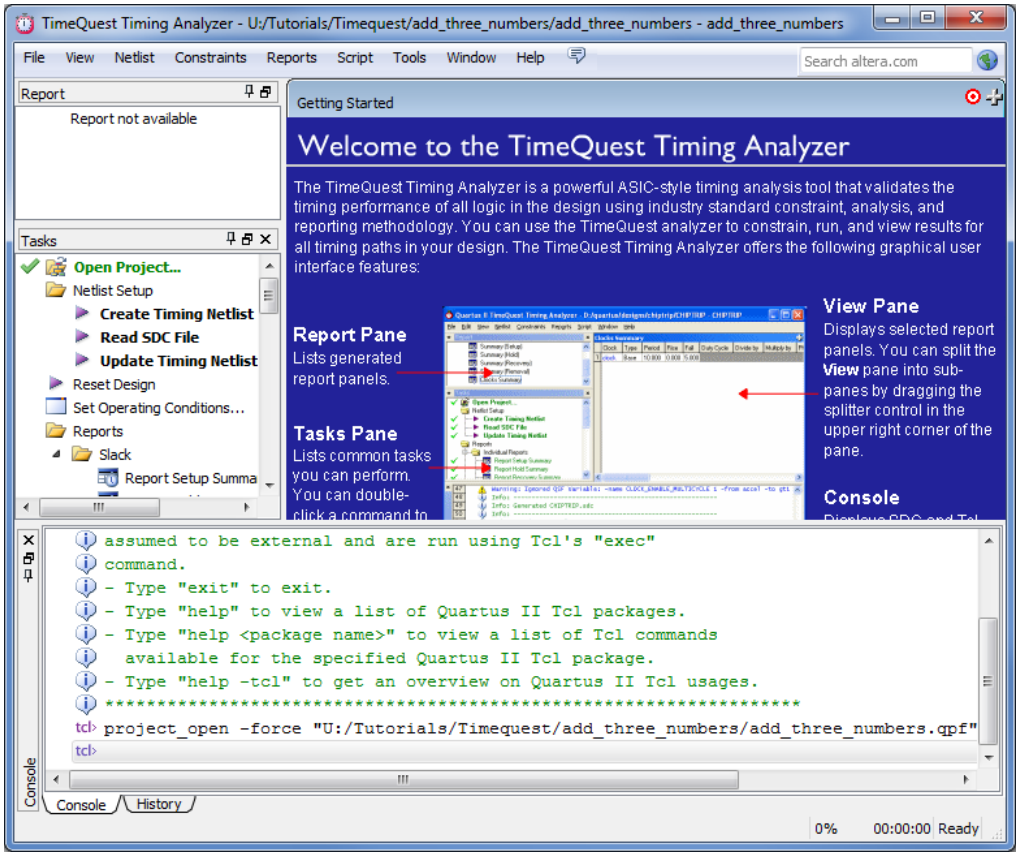

Figure 5. TimeQuest window.

The TimeQuest window consists of several sections. They include the main menu at the top, the Report pane in the top-left corner, the Tasks pane on the left, the View pane on the right, and the Console display at the bottom of the window. The main menu is used to interact with the TimeQuest tool and issue commands. The Report pane contains any reports generated when using the tool, and the Tasks pane contains a sequence of actions that can be performed to obtain timing reports. The View pane hosts any windows that are opened, and initially contains a brief description of each part of the TimeQuest GUI. The Console window at the bottom provides access to a command line for TimeQuest.

We will focus on two of the panes, the Tasks and the Report panes, shown in Figure 6. The Tasks pane provides a sequence of common actions that can be taken to obtain timing data for a design. These tasks include creating a timing netlist, reading a timing constraints file, performing timing analysis, generating reports and saving a timing

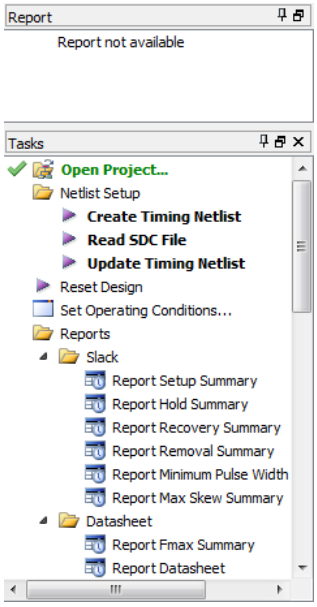

Figure 6. TimeQuest Tasks and Report Panes.

constraints file. The Report pane contains reports with detailed timing information about the design. These reports are generated using commands in the Tasks pane.

To demonstrate how to use the timing analyzer, we go through a set of basic steps to obtain timing data for the example design. Begin by double-clicking the Create Timing Netlist command in the Tasks pane to create a timing netlist, which will be used to perform the analysis. Then double-click Read SDC File to instruct the analyzer to read a Synopsys Design Contraints (SDC) file and apply the constraints during analysis. Specifying the constraints enables the analyzer to determine which parts of the design will operate correctly and which will not. Initially, no constraints are specified and the default constraint of 1 GHz on the clock signal is applied automatically. Third, double-click the Update Timing Netlist command to use the specified constraints to determine which parts of the circuit fail to meet them. Once the timing netlist is updated, reports can be generated.

4.2 Timing Analysis Reports

To generate a report, double-click on a report name in the Tasks pane. For example, double-click on the Setup Summary report. This command will bring up a window in the view pane as shown in Figure 7.

The setup summary report shows a summary for each clock domain. The columns in the report include slack, and total negative slack (TNS), which together indicate how well the design meets setup constraints for each clock domain. In this case, given a 1 GHz frequency requirement, the design fails to meet the constraints because the longest path in the design is 2.503 ns too long (from the slack column). We can examine the timing data in more detail by right-clicking on the row with the given clock and selecting Report Timing... as shown in Figure 8.

Selecting this option opens a window shown in Figure 9. There are several fields in this window that help specify the data to be reported. The first field is the Clocks field, which specifies the types of paths that will be reported.

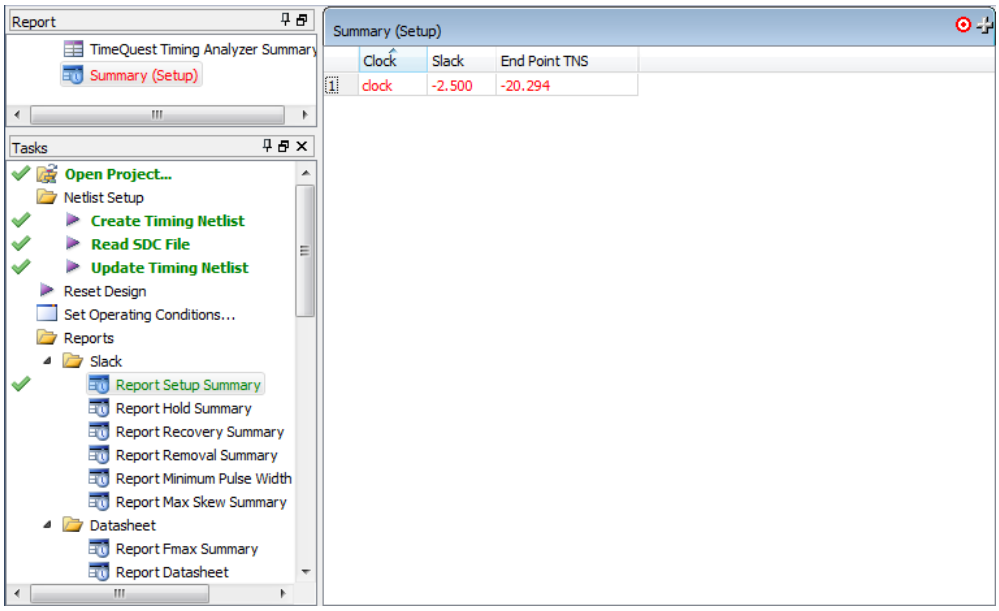

Figure 7. Setup summary.

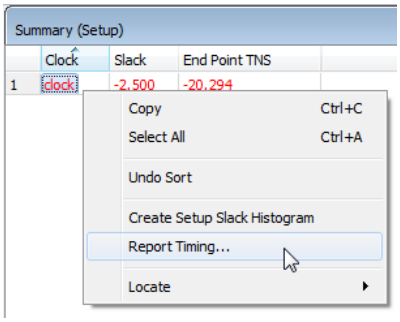

Figure 8. Pop-up menu to generate a detailed timing report.

More precisely, it specifies the clock signal that launches the data and the clock signal that latches the data in. For this example, choose the signal named clock for the To and From clock fields. This will limit the reporting to the register-to-register paths only.

The next field is the Targets field. The Targets field further refines the report by focusing only on certain paths in the design. We can specify the starting and the ending point of the paths of interest by filling the From and To fields. In addition, we can look at only the paths that pass through certain nodes in the design. For this example, we leave these fields blank to indicate that every path should be taken into account for the report.

The next two fields are the Analysis type and Paths fields. The Analysis type field specifies if the report should contain setup, hold, recovery, or removal information. Each of these analyses looks for distinct timing characteristics in your design. For example, the setup analysis determines if the data arrives at a flip-flop sufficiently early for the

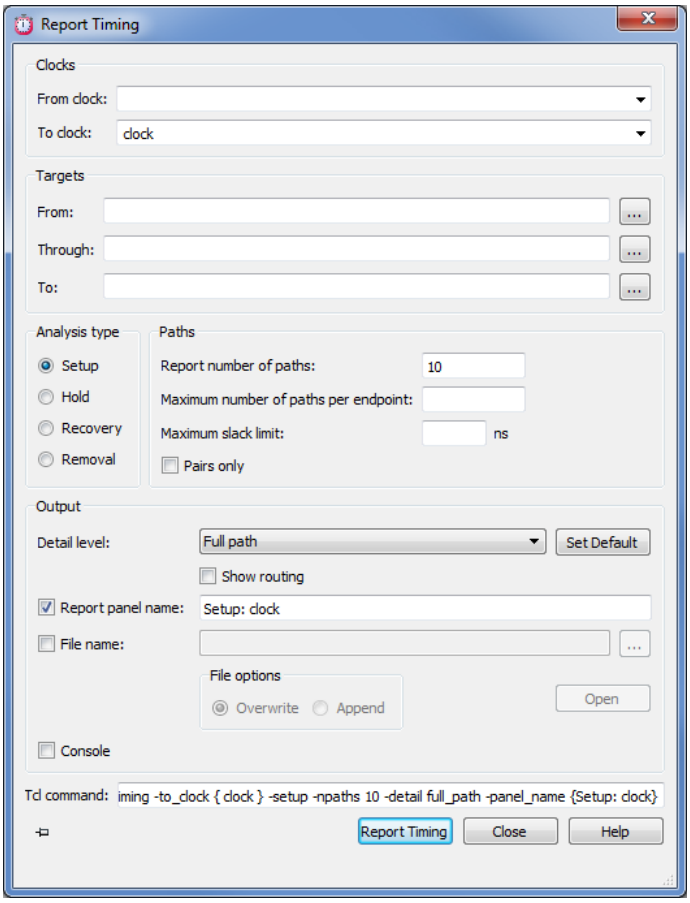

Figure 9. Timing report generation window.

flip-flop to store it reliably, given a clock period. On the other hand, the hold analysis determines if the data input at any given flip-flop remains stable after the positive edge of the clock long enough for the data to be stored in a flip-flop reliably. The Paths field specifies the maximum number of paths to be reported and the maximum slack required for a path to be included in the report. For this example, choose the type of analysis to be Setup and select 10 paths to be reported. This will generate a setup analysis report and show 10 paths with largest negative slack (the paths that violate the constraint the most).

The next set of fields specify the Output format and the level of detail in the report. The output could be to a window or a file. Set the Detail level to Path Only, then set the output to a window by checking the Report panel name check box (and not the File name check box). The window should be named Setup: clock by default, and that name will identify the report in the report pane.

Finally, the last field is the Tcl command field. This field shows a command that will be executed to generate the requested report. You do not need to edit this field. Press the Report Timing button to generate and display the report shown in Figure 10.

The timing report in Figure 10 consists of three sections. The top part of the report contains a list of paths, including

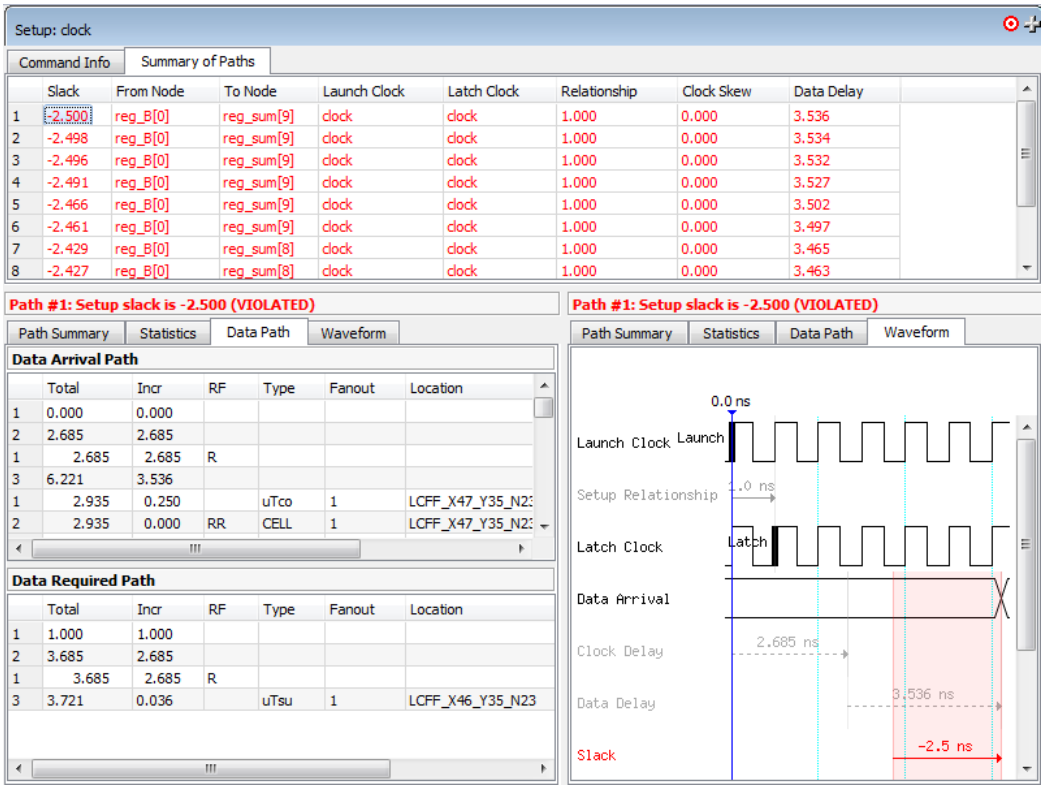

Figure 10. Detailed setup time information for a given register-to-register path.

timing information. In particular, the column called slack shows the difference between the time a signal is required to arrive at a destination FF, as per the desired clock period, and the actual arrival time. When the slack is negative, the path takes too long to compute, and the timing is considered to violate the clock constraint. In this example, the desired clock period was 1ns, and one of the paths in the circuit exceeded this constraint by 2.503ns. The timing analysis details for a given path can be displayed in the two panes at the bottom of the report by clicking on one of the paths in the report, as shown in Figure 10.

The bottom two sections of the report consist of identical sets of tabs. The tabs show path summary, statistics, data path, and waveforms. The left-hand side shows a set of components through which the path travels, including the delays along the way, while the right-hand side shows the waveform that explains how the timing violation occurred.

The waveform display is a useful tool in understanding the timing of a circuit. It includes both data delays on a given path as well as clock delays to source and destination registers. At a glance, the information provided in the waveform is comprehensive to experts, but may not be as intuitive to others. This is because the timing information is shown with respect to the time when inputs appear at the input pin to an FPGA device. For example, the clock signal shown in the first line is the signal at the pin of the device, and it arrives at the clock input of the source flip-flop later. In Figure 11 we show how to interpret the waveform information presented in Figure 10.

In the figure, the first two waveforms show the clock signals for the source and destination flip-flops, along with a timing requirement - in this case 1 ns. The launch and latch edges of each clock are indicated by thick lines. The

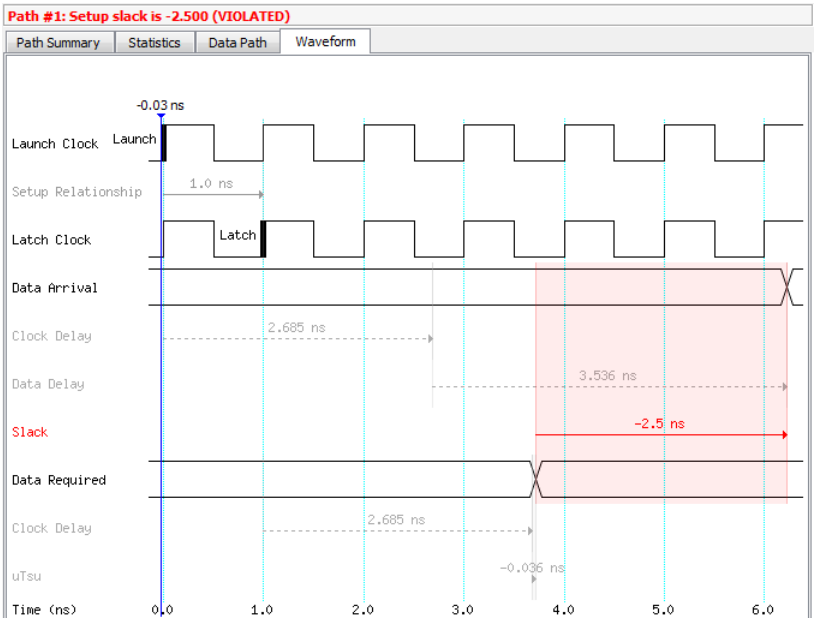

Figure 11. Interpreting setup time information for a given register-to-register path.

next two waveforms, Data Arrival and Data Delay, show the time it will take a signal to propagate from source to destination. Notice that Data Delay is measured from the positive edge of the clock named Launch Clock at Source FF. The second last waveform, called Data Required indicates the time when the data should have arrived at the destination flip-flop to be stored correctly, including the setup time (*uTsu*) shown in the last line. The Clock Delay line represents the time between the launch/latch edges at the pin of the device and at the clock inputs of the source/destination flip-flops.

From the diagram we see that the time the data arrives at its destination is after the time it is required to arrive if it is to meet the timing constraint. Thus, the timing constraint is violated. To indicate this, a negative slack value is shown to indicate by how much the timing constraint is violated. In contrast, a timing constraint that is satisfied has a positive slack value.

4.3 Setting Up Timing Constraints for a Design

TimeQuest provides a way to specify timing constraints to be included in the next compilation of your design through the Constraints menu. To assign a clock constraint, select Create Clock... from the Constraint menu. A window shown in Figure 12 will appear.

In the window, the constraint on the clock signal can be specified. To do this give the clock constraint a name (for example, the name of the clock for which constraints are specified) in the top field. Then, specify the clock period to be 4 ns in the field below. The next two fields define the time at which the clock changes from 0 to 1 and 1 to 0. Leave these fields empty to indicate that the rising edge of the clock should appear at time 0, and a falling edge at one half of the clock period. Finally, specify the Targets field to be *clock* as shown in the figure, to indicate that the given constraint is for the clock signal named *clock*. Then press the Run button to apply the constraint and save the

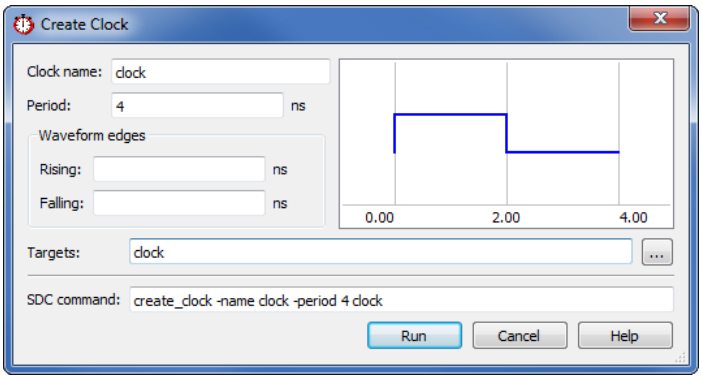

Figure 12. TimeQuest window to create a clock constraint.

constraints file into example.sdc file, by double-clicking on the Write SDC File... task as shown in Figure 13.

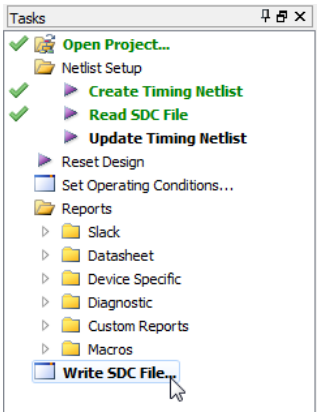

Figure 13. Saving a constraints file.

Once the constraints file is saved, it can be used by Quartus II when compiling a project. This is done in Quartus II by going into Assignments > Settings... > TimeQuest Timing Analyzer, and adding the example.sdc file to the TimeQuest timing analyzer settings as shown in Figure 14. Once the constraint file is added, recompile the project and open up the setup summary report, as before, in TimeQuest. You will now notice that the timing constraint is met.

5 Regarding Designs with Multiple Clock Signals

TimeQuest is capable of analyzing circuits that contain multiple clocks. This includes cases where the designer used several clocks, or clock signals were automatically to support features such as the SignalTap II Logic Analyzer or a JTAG interface. Should the reader work with such designs, it is important to note that the initial experience with TimeQuest may differ from that described above. In designs with multiple clocks, it is important to apply constraints to each clock before performing timing analysis. Doing so will make the analyzer provide the same

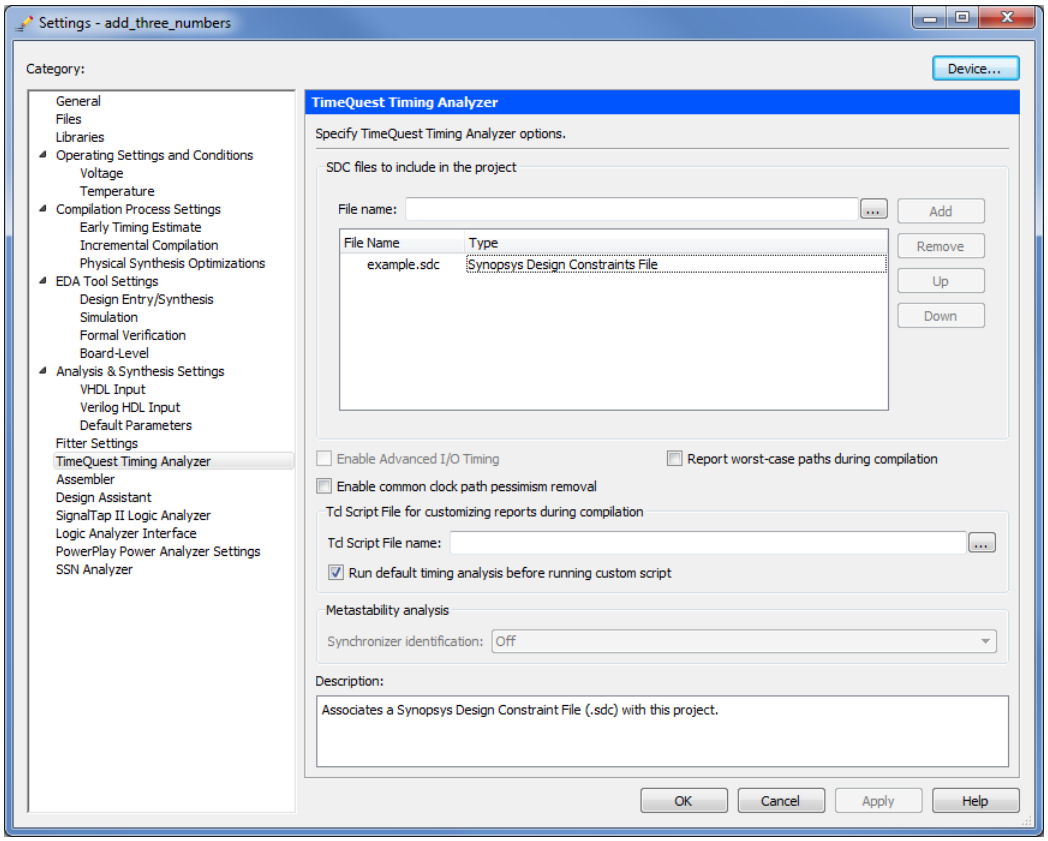

Figure 14. Including a constraints file in Quartus II project.

reports as described in previous sections.

6 Conclusion

This tutorial demonstrated the basic use of the TimeQuest timing analyzer. While the descriptions of timing analysis and setting up timing constraints were limited to clock constraints in a simple circuit, TimeQuest provides even more powerful tools to specify timing constraints for larger and more complex designs.