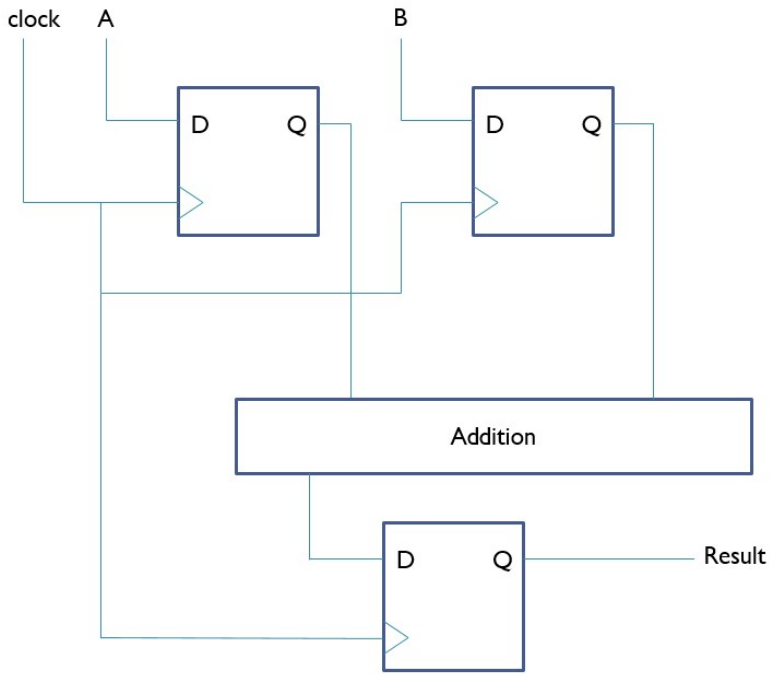
Education Objective
The educational objective of this demo is to become familiar with Altera's TimeQuest tool used to assist in properly constraining and analyzing paths in the FPGA on the DE1-SoC board.

Technical Objective
The technical objective of this laboratory is to design a simple adder on the DE1-SoC that will allow for a deeper look into timing failures, analysis, and eventually timing closure.

Demonstration Procedure

1. Design the following Adder in VHDL. A and B are two 16 bit vectors. Save the file as **TimeQuest_Demo.vhd.**

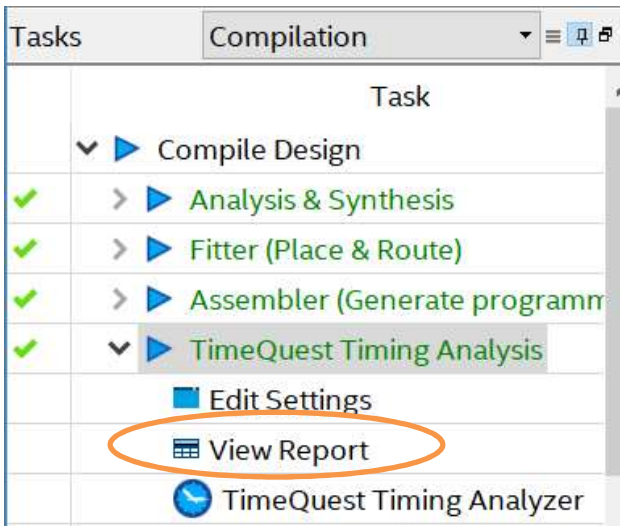


2. Create a folder named **TimeQuest_Demo** and within it create a folder named **TimeQuest_Demo_Quartus**.

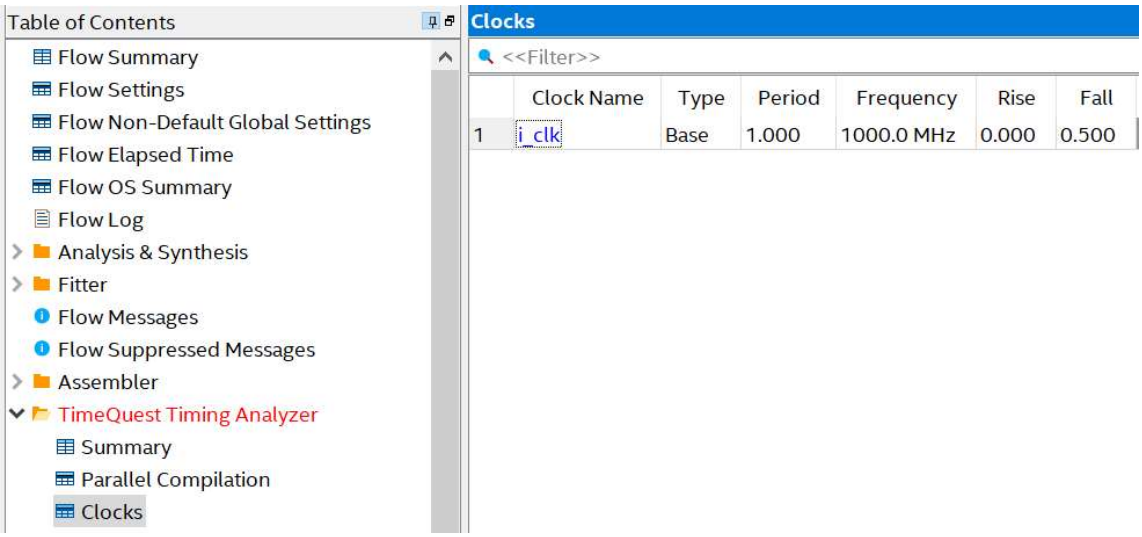
3. Create a new Quartus project and name it **TimeQuest_Demo**. Put the **TimeQuest_Demo.vhd** file in the **TimeQuest_Demo_Quartus** folder. Add to project.

4. Compile the design and view in the RTL viewer. **Tools > Netlist Viewers > RTL Viewer.** Does the RTL Viewer design closely match the figure above?

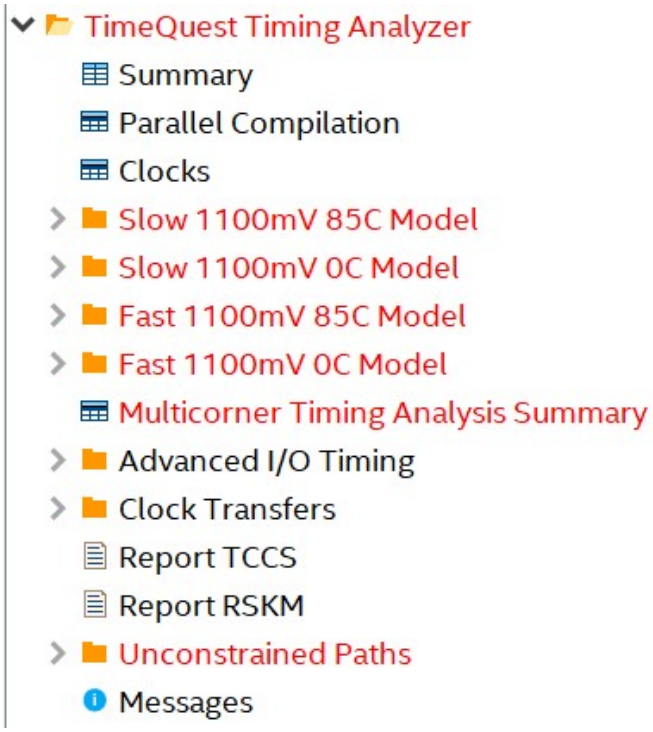
5. View the TimeQuest report by double clicking the **View Report option in the Task window under TimeQuest Timing Analysis.**



6. Click on the **Clocks** option under **TimeQuest Timing Analyzer.** Here you can see the initial assumptions TimeQuest is making about the adder design. Notice that the clock signal from the design has been given a clock period of 1ns (1000MHz). This constraint is a default that is used by the TimeQuest tool until the user defines a clock timing constraint.



7. Notice the TimeQuest Timing Analyzer is red. This indicates there were failures in the timing analysis of the design.



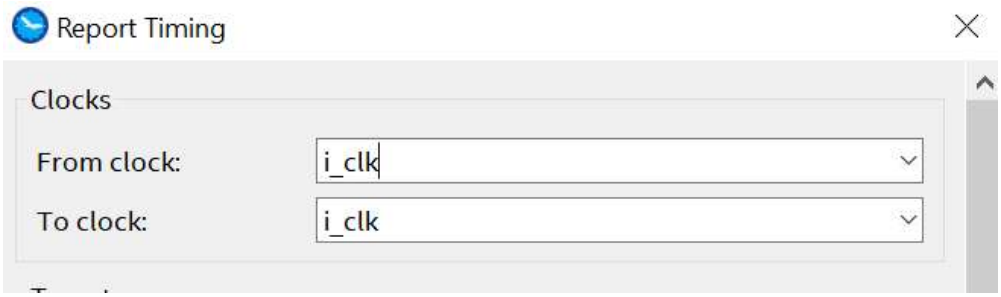
8. Open each model analysis and view the failures. Record the Fmax and Setup Time in a chart resembling the one below for submission with your demo. Fmax will be unavailable for the fast models.

	Fmax	Setup Time
Slow 1100mV 85C		
Slow 110mV 0C		
Fast 1100mV 85C		
Fast 1100mV 0C		

9. Note the achievable Fmax. This frequency will be much higher than the 50MHz available on your DE1-SoC board, but is not meeting the default clock constraint requesting 1000MHz.
10. Review the lecture notes on setup time. Review the difference between negative slack vs. positive slack? Are these models passing with a clock constraint of 1ns?

11. To view a timing diagram of the clock and data delays select the Clocks option in the Table of Contents. Right clock on the clock. **Select Report Timing (in TimeQuest UI).**

12. Choose from clock to be the input clock to your design. The To clock will also be the input clock. Leave all defaults and select **Report Timing.**



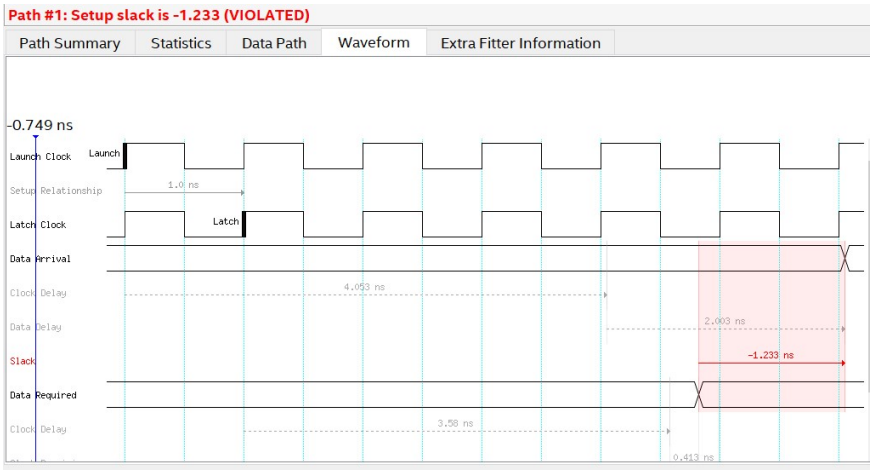
13. In the Summary of Paths there will be a list of failing paths. Find the largest path and record it. Select the largest path to be shown in the waveform window. In the case below the path is from store1(5) to temp(16).

Slow 1100mV 85C Model

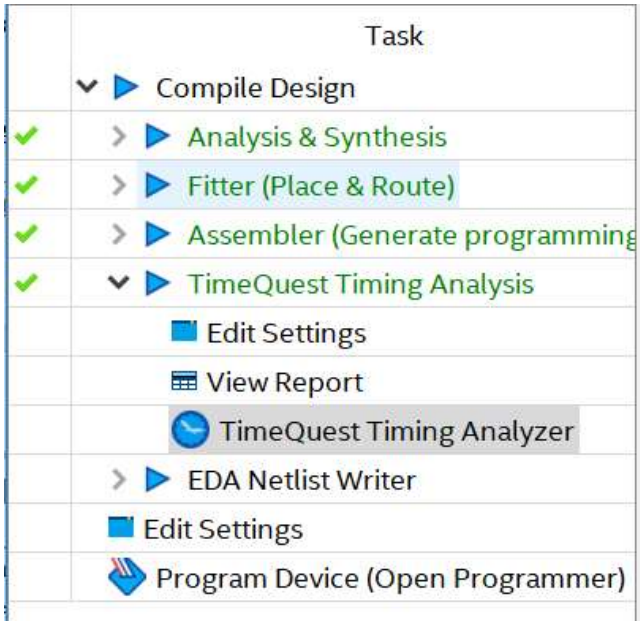
Command Info | Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	-1.233	store1[5]	temp[16]	i_clk	i_clk	1.000	-0.060	2.003
2	-1.199	store1[5]	temp[14]	i_clk	i_clk	1.000	-0.060	1.969

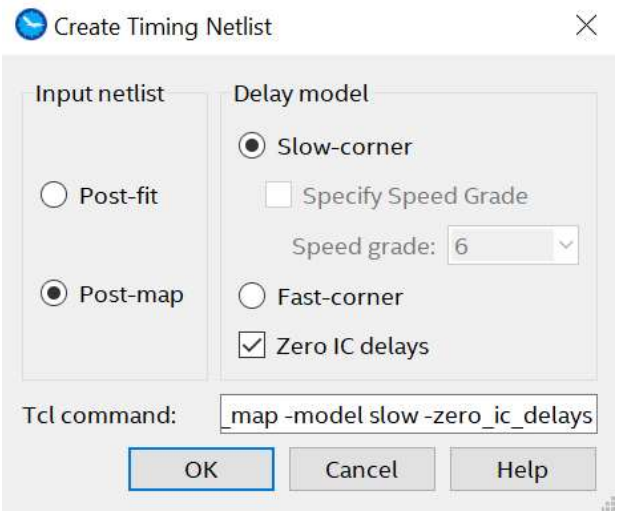
14. Record the clock delay, data delay, and slack. Notice that the slack region in the waveform is red. This should match the slack of the selected path (-1.233). Notice the slack is shown as the difference between the data arrival vs. the data required.



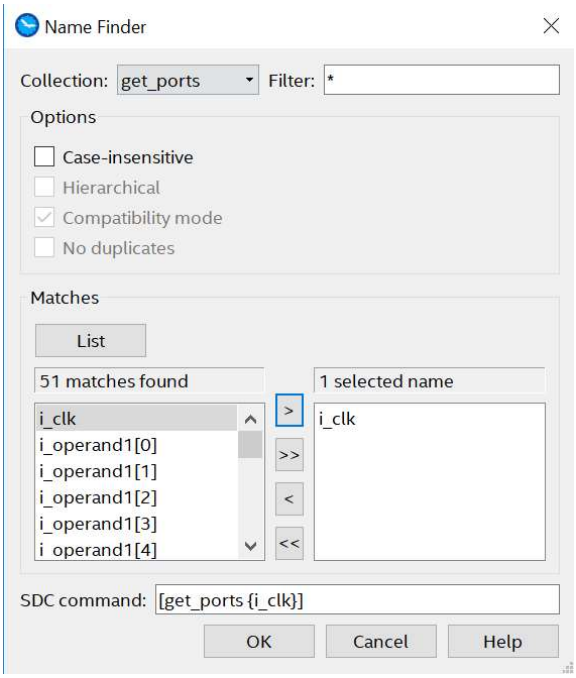
15. To update the clock constraint to the board clock frequency of 50MHz you will need to launch the TimeQuest tool. Double click on the TimeQuest Timing Analyzer in the task bar.



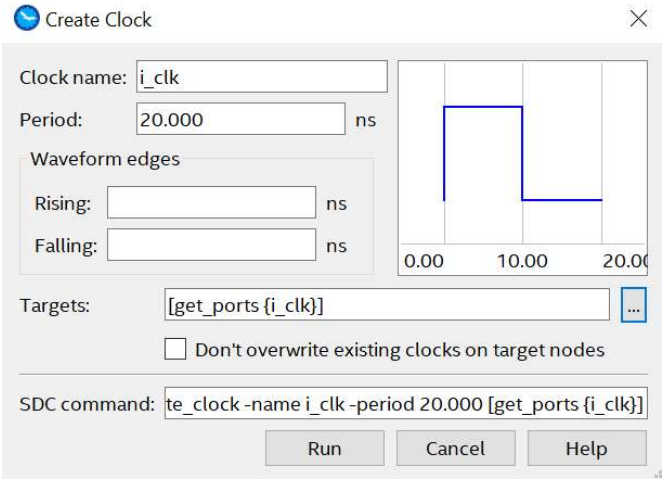
16. Once in the tool, begin by creating a timing netlist. **Netlist > Create Timing Netlist…** Choose **post-map** as the input netlist. Click OK.



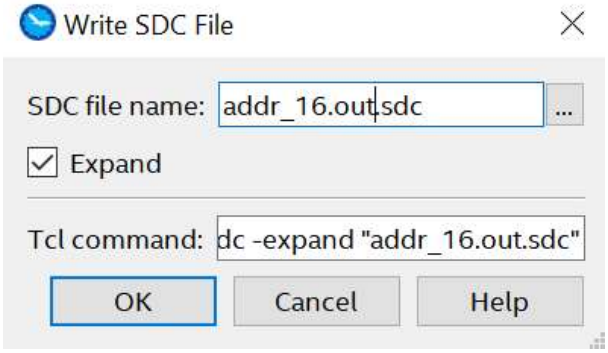
17. Choose **Constraints > Create Clock …** Assign a clock name. Set the **period to 20ns** (for 50MHz). Leave the rising and falling edges empty, the assumption will be 50% duty cycle. In the Targets field click on the … box to choose the target clock. Push the List button to see all ports. Select your clock signal and use the > to move it to the right window. Click OK.



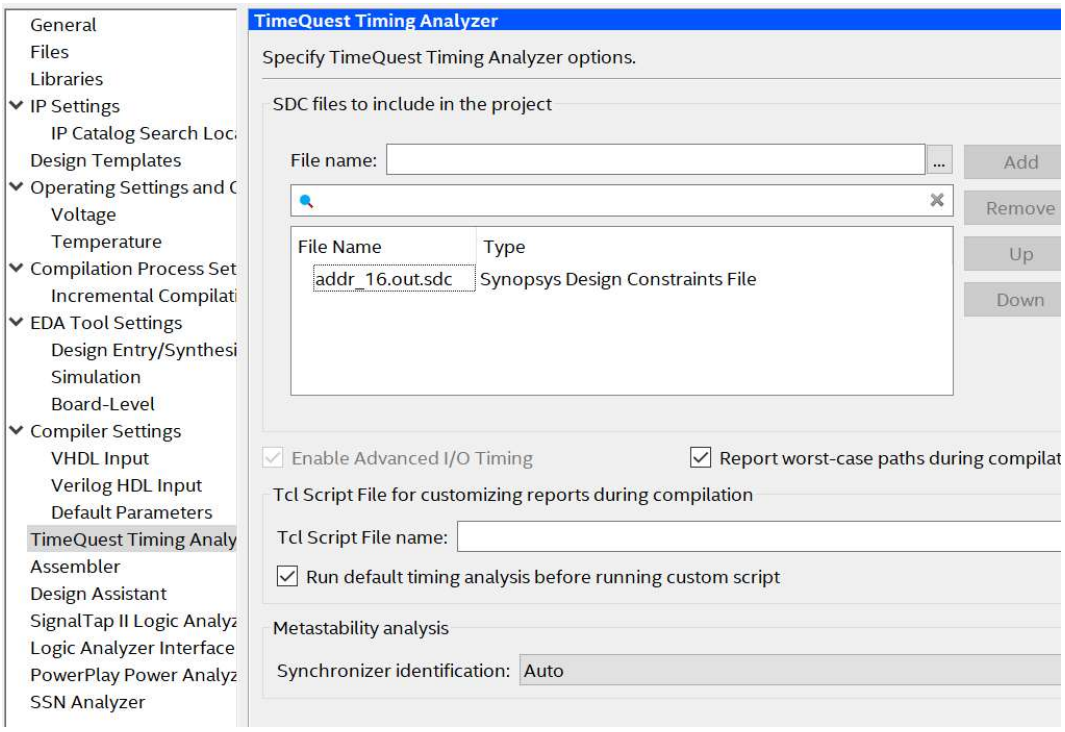
18. In the SDC command window you will a create_clock constraint. This may be copied and used in the constraints file, but for now we will use the tool to import constraints. Click **RUN** to create the clock constraint.



19. To write any created constraints to the SDC file go to Constraints > Write SDC File. You may rename the sdc file, but keep the correct extension. Click OK.



20. Close the TimeQuest tool. Go to Assignments > Settings. In the Category pane choose TimeQuest Timing Analyzer. Next to File Name choose the browser option. Navigate to the recently created SDC file and add it to the project. Click OK.



21. Open the SDC file and view the newly created clock constraint. This constraint is linking the i_clk port to a design clock named i_clk and constraining the period to 20 ns, rising at 0ns and falling at 10ns.

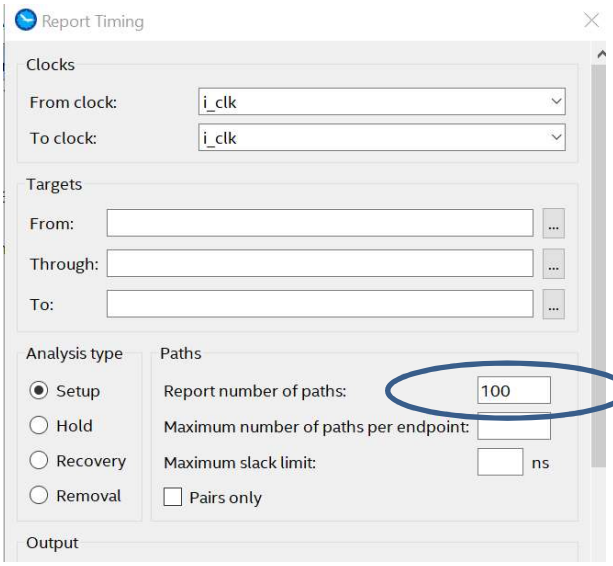
```
create_clock -name {i_clk} -period 20.000 -waveform { 0.000 10.000 } [get_ports {i_clk}]
```

22. Recompile the adder design and view the TimeQuest report. Has the clock been updated to match the created constraint?

23. Record the updated results in the table below. What do you notice about the updated Fmax? Why is the Fmax different? What about the setup time?

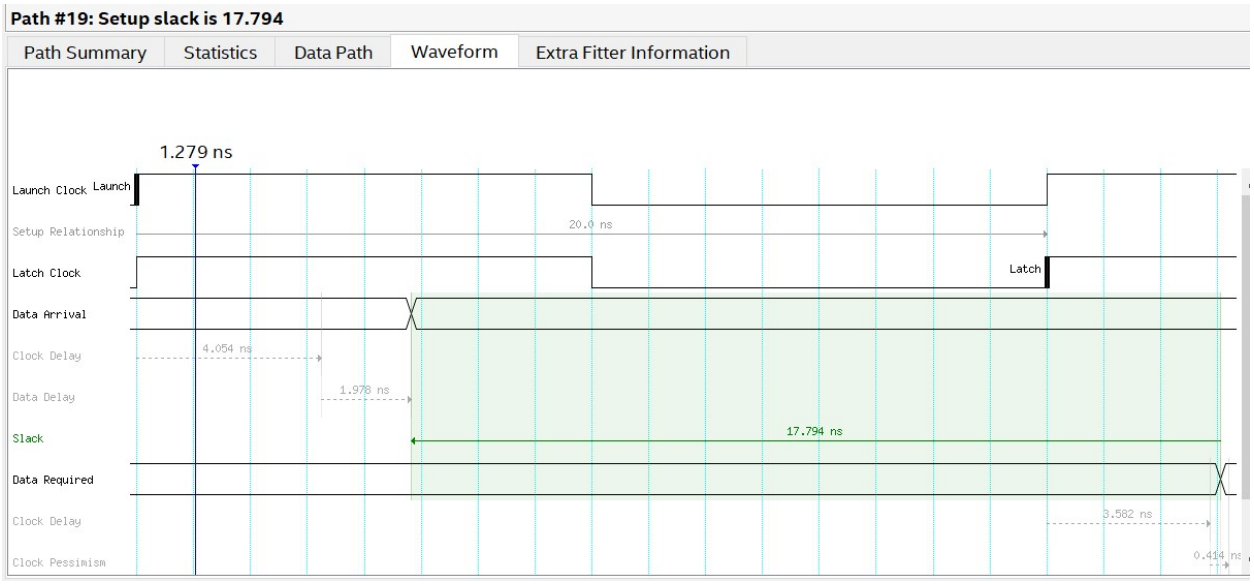
	Fmax	Setup Time
Slow 1100mV 85C		
Slow 110mV 0C		
Fast 1100mV 85C		
Fast 1100mV 0C		

24. Report the clock timing again with the new clock constraint (repeat steps 11-14). Use the failing path from the earlier example. When configuring the timing report, you may have to report more paths to find your original failing path. To do so, change the number of reported path when setting up your from/to clocks.



25. Record the new slack for the previously failing path. View the waveform, is it now passing? Now what is the clock delay, data delay, and slack? How do these numbers compare to the previous results?

19	17.794	store1[5]	temp[16]	i_clk	i_clk	20.000	-0.058	1.978



Demo Submission

Submit a small write up including tables and snapshots of all data collected in this demo, along with a short conclusion in relation to Fmax, setup time, and the TimeQuest tool.