Vivado Tutorial 3

Implementing the Design

Introduction

This tutorial continues with the previous tutorial. You will perform static timing analysis. You will implement the design with the default settings and generate a bitstream. Then you will open a hardware session and program the FPGA. You will use on-board UART of the Nexys4 DDR or the Basys3 board to validate your design.

Objectives

After completing this tutorial, you will be able to:

- Implement the design
- Generate various reports and analyze the results
- Run static timing analysis
- Generate bitstream and verify the functionality in hardware

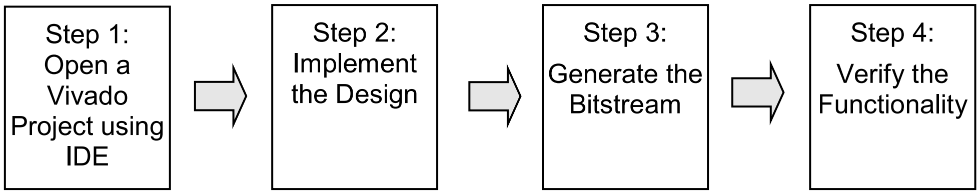
Procedure

This tutorial is broken into steps that consist of general overview statements providing information on the detailed instructions that follow. Follow these detailed instructions to progress through the tutorial.

*Note: The FPGA board equipped in Embedded Tutorial (Brickyard 215) is **Nexys4 DDR**.*

Updated by PSCLab at ASU

General Flow


Step 1: Open a Vivado Project using IDE

Step 2: Implement the Design

Step 3: Generate the Bitstream

Step 4: Verify the Functionality

Open a Vivado Project using IDE — Step 1

1-1. Launch Vivado and open the tutorial2 project. Save the project as tutorial3 in the <*CSE320_tutorial3*> directory making sure that the create subdirectory option is selected. Set the flatten_hierarchy setting to rebuilt. Create new synthesis run naming it as synth_2.

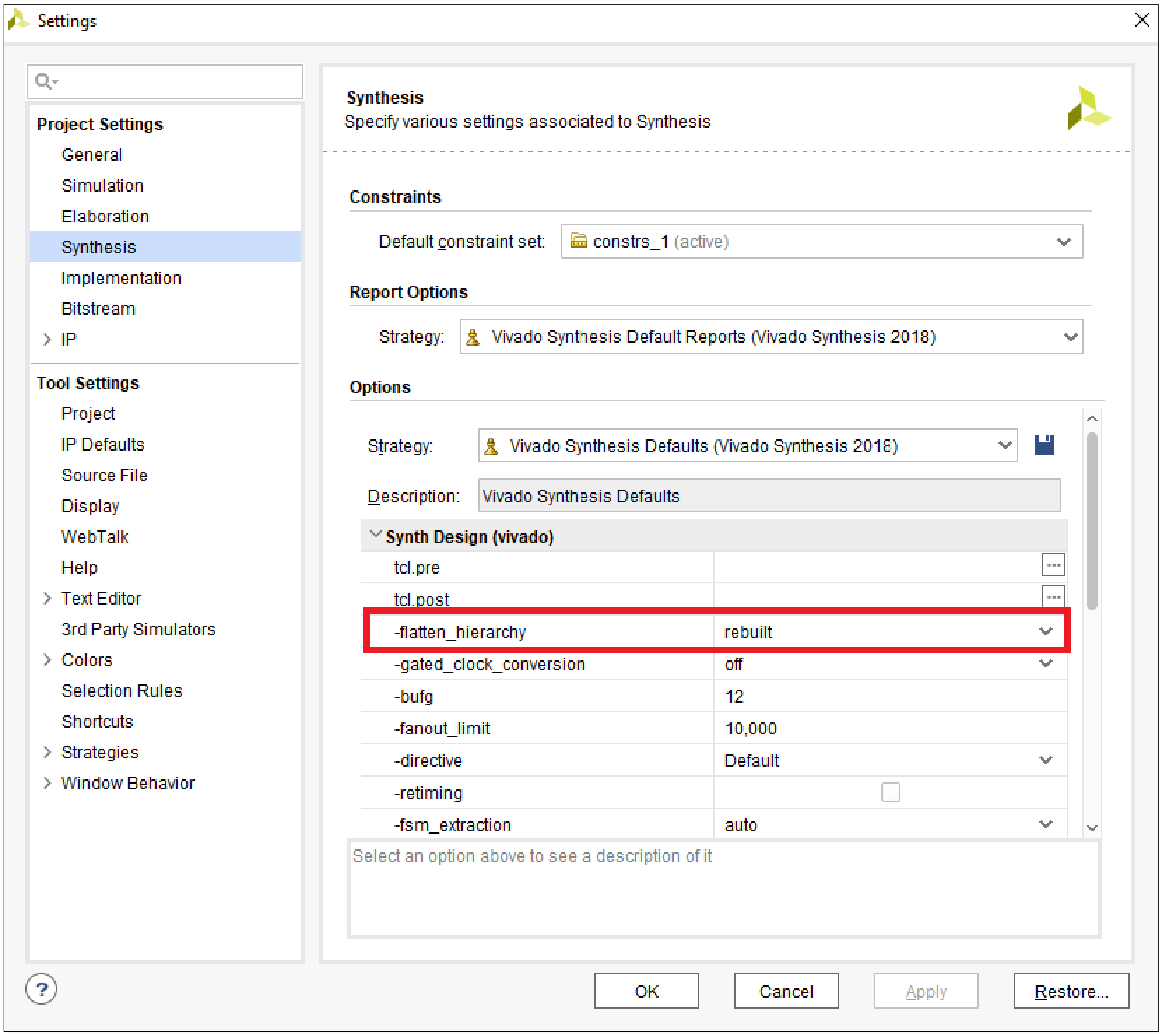
<CSE320_tutorial3> refers to the vivado project directory, e.g. c:\xup\digital\CSE320_tutorial2 and <CSE320_tutorial3_source> refers to the source file directory, e.g. c:\xup\digital\CSE320_tutorial2_source. Reference to **<board>** means either the **Nexys4 DDR** or the **Basys3**.

1-1-1. Start the Vivado if necessary and open the tutorial2 project (tutorial2.xpr) you created in the previous tutorial using the **Open Project** link in the Getting Started page.

1-1-2. Select **File > Save Project As …** to open the *Save Project As* dialog box. Enter **tutorial3** as the project name. Make sure that the *Create Project Subdirectory* option is checked, the project directory path is **<CSE320_tutorial3>** and click **OK.**

1-1-3. Click on the **Synthesis Settings** in the *Flow Navigator* pane.

1-1-4. Make sure that the *flatten_hierarchy* is set to **rebuilt**, which allows the design hierarchy to be preserved for synthesis, and then rebuilt which is more useful for design analysis because many logical references will be maintained.



Synthesis
Specify various settings associated to Synthesis

Constraints

Default constraint set: constrs_1 (active)

Report Options

Strategy: Vivado Synthesis Default Reports (Vivado Synthesis 2018)

Options

Strategy: Vivado Synthesis Defaults (Vivado Synthesis 2018)

Description: Vivado Synthesis Defaults

Synth Design (vivado)	
tcl.pre	
tcl.post	
-flatten_hierarchy	rebuilt
-gated_clock_conversion	off
-bufg	12
-fanout_limit	10,000
-directive	Default
-retiming	□
-fsm_extraction	auto

Select an option above to see a description of it

Figure 1. Setting hierarchy to rebuilt

1-1-5. Click **OK**.

A Create New Run dialog box will appear asking you if a new run should be created. Click **Yes** and then **OK** to create the new run with **synth_2** name.

1-2. Synthesize the design. Generate the timing summary and analyze the design.

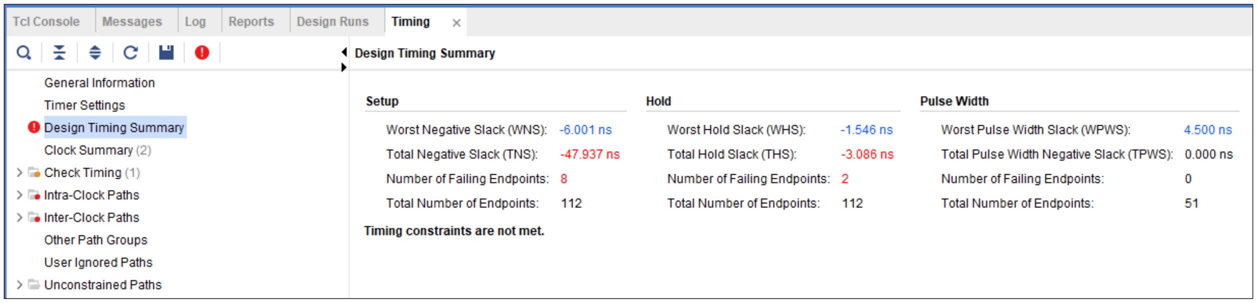
1-2-1. Click on **Run Synthesis** under the *Synthesis* tasks of the *Flow Navigator* pane.

The synthesis process will be run on the uart_led.v and all its hierarchical files. When the process is completed a *Synthesis Completed* dialog box with three options will be displayed.

1-2-2. Select the *Open Synthesized Design* option and click **OK** as we want to look at the synthesis output.

1-2-3. Click on **Report Timing Summary** under the *Synthesized Design* tasks of the *Flow Navigator* pane.

1-2-4. Leave all the settings unchanged, and click **OK** to generate a default timing report, *timing_1.*



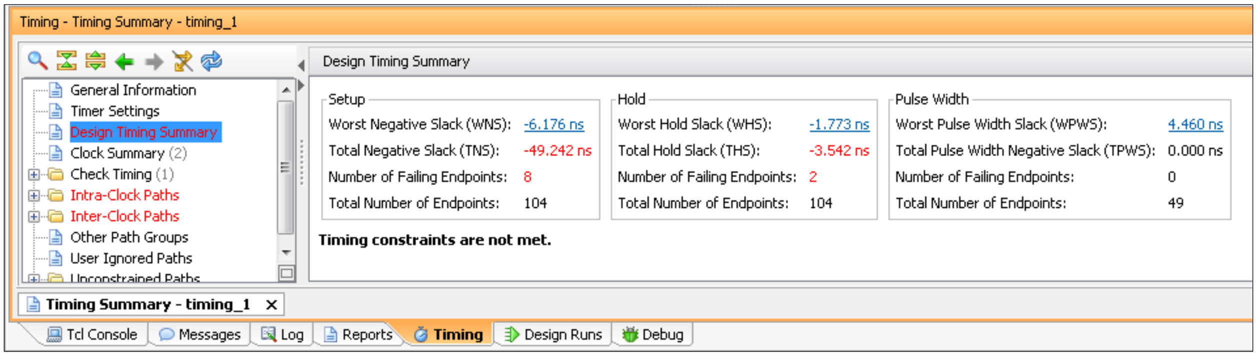
- General Information
- Timer Settings
- Design Timing Summary
- Clock Summary (2)
- Check Timing (1)
- Intra-Clock Paths
- Inter-Clock Paths
- Other Path Groups
- User Ignored Paths
- Unconstrained Paths

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-6.001 ns	Worst Hold Slack (WHS):	-1.546 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	-47.937 ns	Total Hold Slack (THS):	-3.086 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	8	Number of Failing Endpoints:	2	Number of Failing Endpoints:	0
Total Number of Endpoints:	112	Total Number of Endpoints:	112	Total Number of Endpoints:	51

Timing constraints are not met.

Figure 2. Timing report for the Nexys4 DDR



- General Information
- Timer Settings
- Design Timing Summary
- Clock Summary (2)
- Check Timing (1)
- Intra-Clock Paths
- Inter-Clock Paths
- Other Path Groups
- User Ignored Paths
- Unconstrained Paths

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-6.176 ns	Worst Hold Slack (WHS):	-1.773 ns	Worst Pulse Width Slack (WPWS):	4.460 ns
Total Negative Slack (TNS):	-49.242 ns	Total Hold Slack (THS):	-3.542 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	8	Number of Failing Endpoints:	2	Number of Failing Endpoints:	0
Total Number of Endpoints:	104	Total Number of Endpoints:	104	Total Number of Endpoints:	49

Timing constraints are not met.

Figure 2. Timing report for the Basys3

1-2-5. Click on the link beside the **Worst Negative Slack** (WNS) and see the 8 failing paths.

1-2-6. Double-click on the Path 23 to see a detailed view of the path. The path report shows four sections: (i) Summary, (ii) Source Clock Path, (iii) Data Path, and (iv) Destination Clock Path.

1-2-7. Select Path 23 in the timing summary panel, or the Path summary view, right-click, and select **Schematic.**

The schematic for the output data path will be displayed.

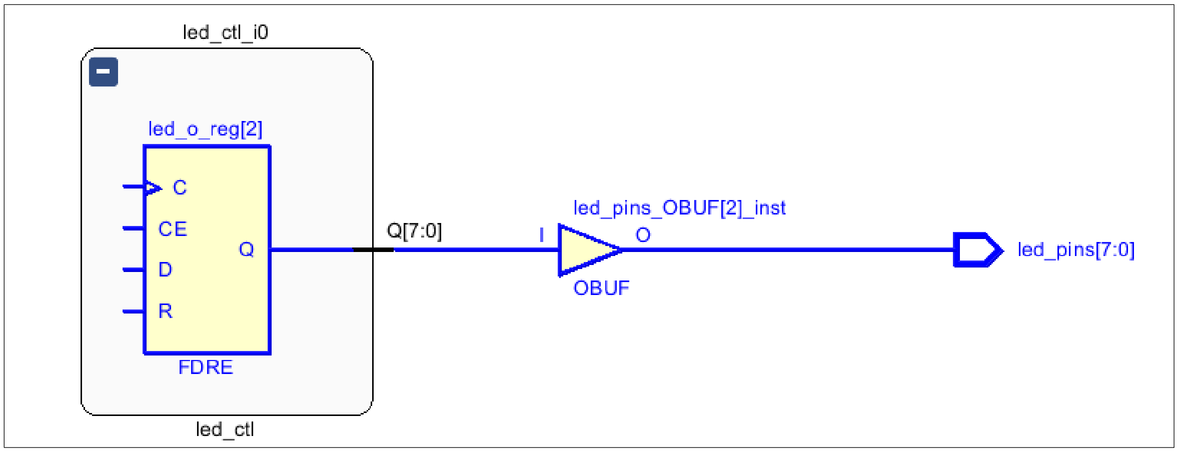


Figure 3. The output data path

1-2-8. In order to see how the Source Clock Path is made up in schematic form, double-click on left end of the C pin of the FDRE in the schematic.

This will show the net between the BUFG and C port of the FDRE.

1-2-9. Similarly, double-click on the left end of the BUFG to see the path between IBUF and BUFG.

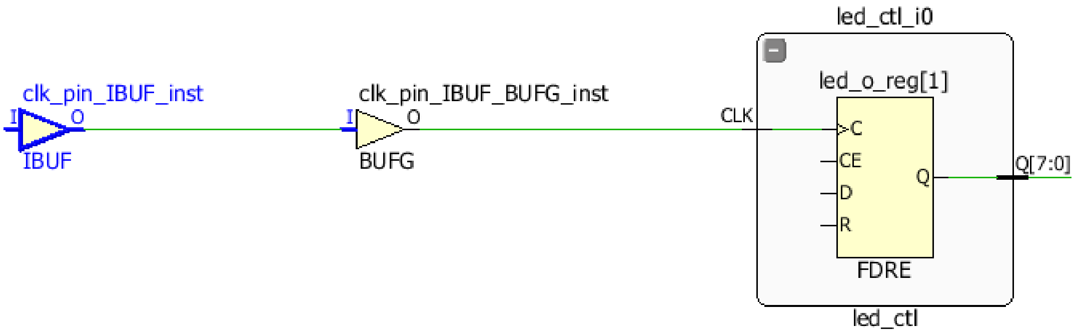


Figure 4. Source to clock port of the FDRE

1-2-10. Finally, double-click on the input pin of IBUF to see the path between the clock input pin and the IBUF.

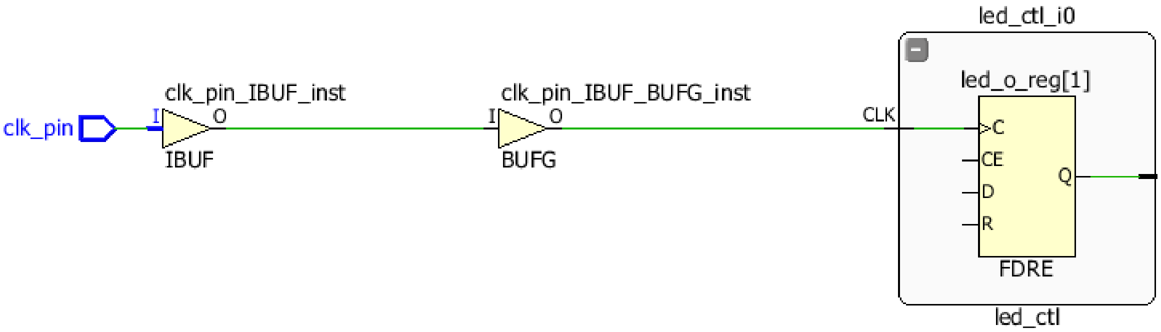
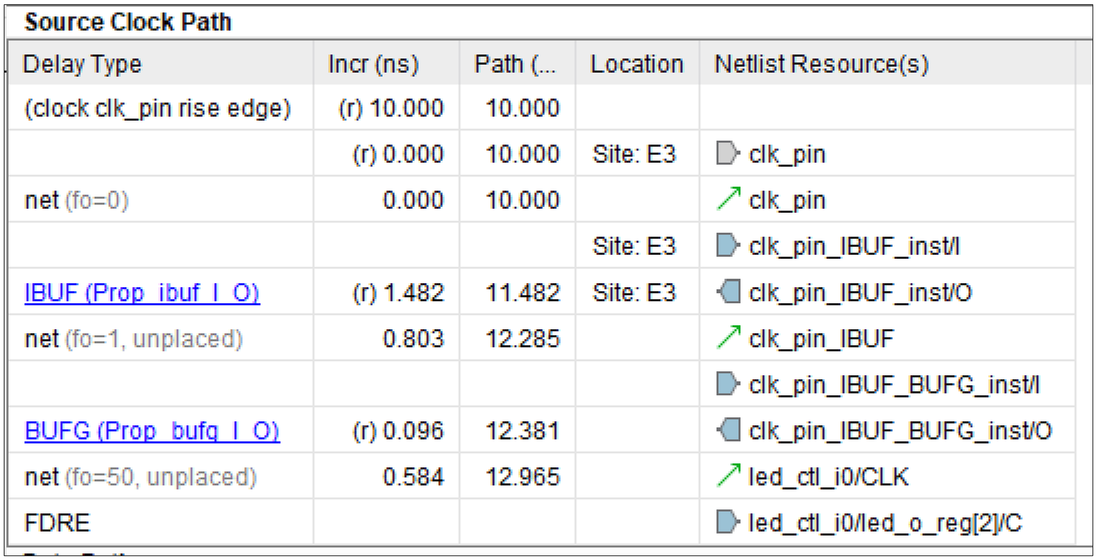


Figure 5. The schematic view of the source clock path

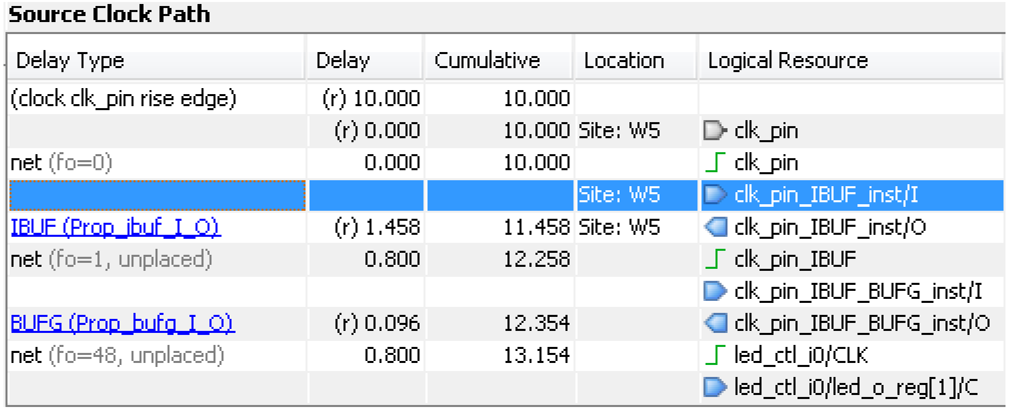
This corresponds to the Source Clock Path in the timing report.



Source Clock Path

Delay Type	Incr (ns)	Path (...	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 10.000	10.000		
	(r) 0.000	10.000	Site: E3	clk_pin
net (fo=0)	0.000	10.000		clk_pin
			Site: E3	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.482	11.482	Site: E3	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.803	12.285		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.096	12.381		clk_pin_IBUF_BUFG_inst/O
net (fo=50, unplaced)	0.584	12.965		led_ctl_i0/CLK
FDRE				led_ctl_i0/led_o_reg[2]/C

Figure 6. The source clock path for the Nexys4 DDR

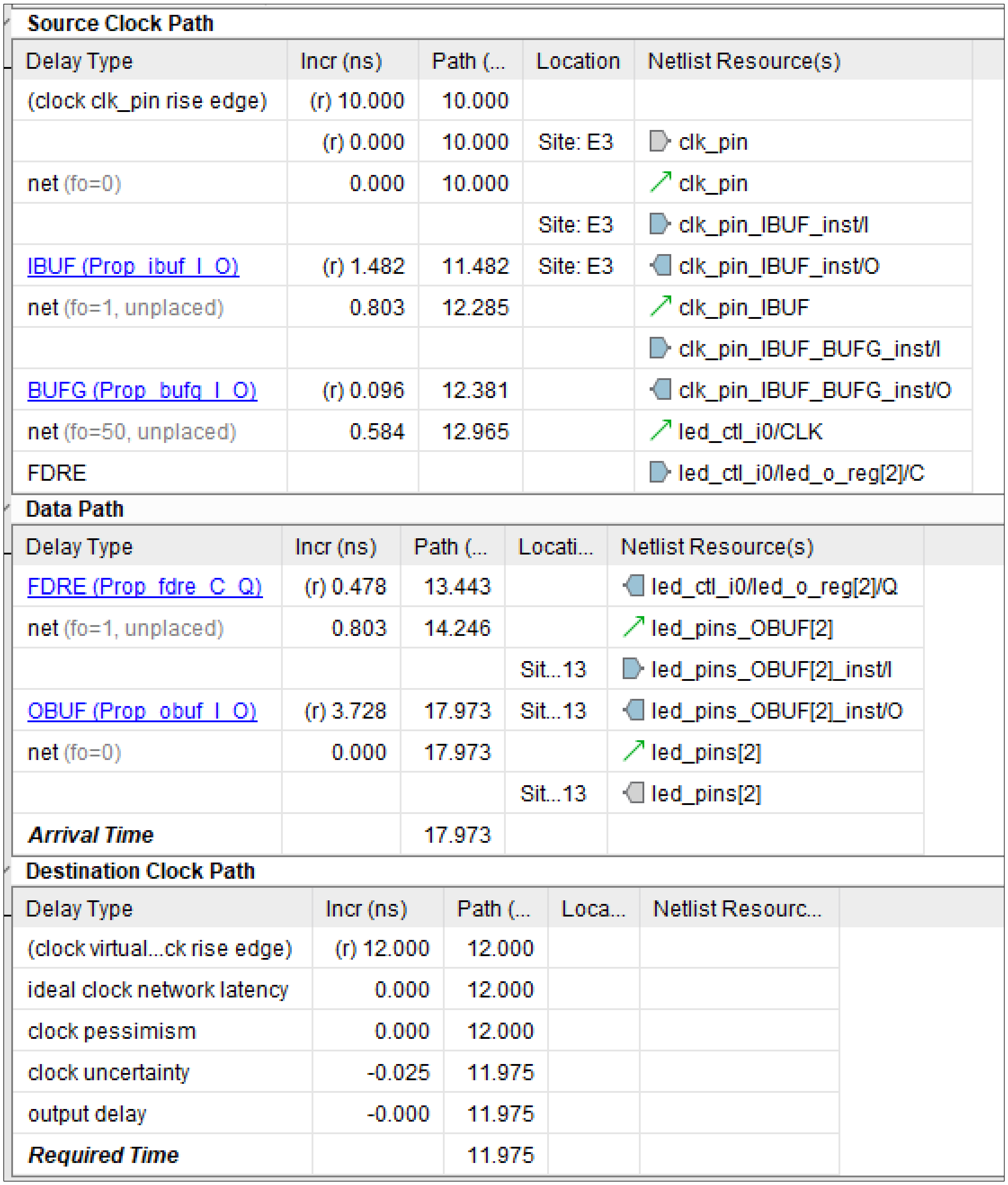


Source Clock Path

Delay Type	Delay	Cumulative	Location	Logical Resource
(clock clk_pin rise edge)	(r) 10.000	10.000		
	(r) 0.000	10.000	Site: W5	clk_pin
net (fo=0)	0.000	10.000		clk_pin
			Site: W5	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.458	11.458	Site: W5	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.800	12.258		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.096	12.354		clk_pin_IBUF_BUFG_inst/O
net (fo=48, unplaced)	0.800	13.154		led_ctl_i0/CLK
				led_ctl_i0/led_o_reg[1]/C

Figure 6. The source clock path for the Basys3

Since the virtual clock is synchronized to but slower (12 ns) than the clk_pin period (10 ns), the *worst* case data path delay includes the clock period of the clk_pin clock source.



Source Clock Path

Delay Type	Incr (ns)	Path (...	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 10.000	10.000		
	(r) 0.000	10.000	Site: E3	clk_pin
net (fo=0)	0.000	10.000		clk_pin
			Site: E3	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.482	11.482	Site: E3	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.803	12.285		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.096	12.381		clk_pin_IBUF_BUFG_inst/O
net (fo=50, unplaced)	0.584	12.965		led_ctl_i0/CLK
FDRE				led_ctl_i0/led_o_reg[2]/C

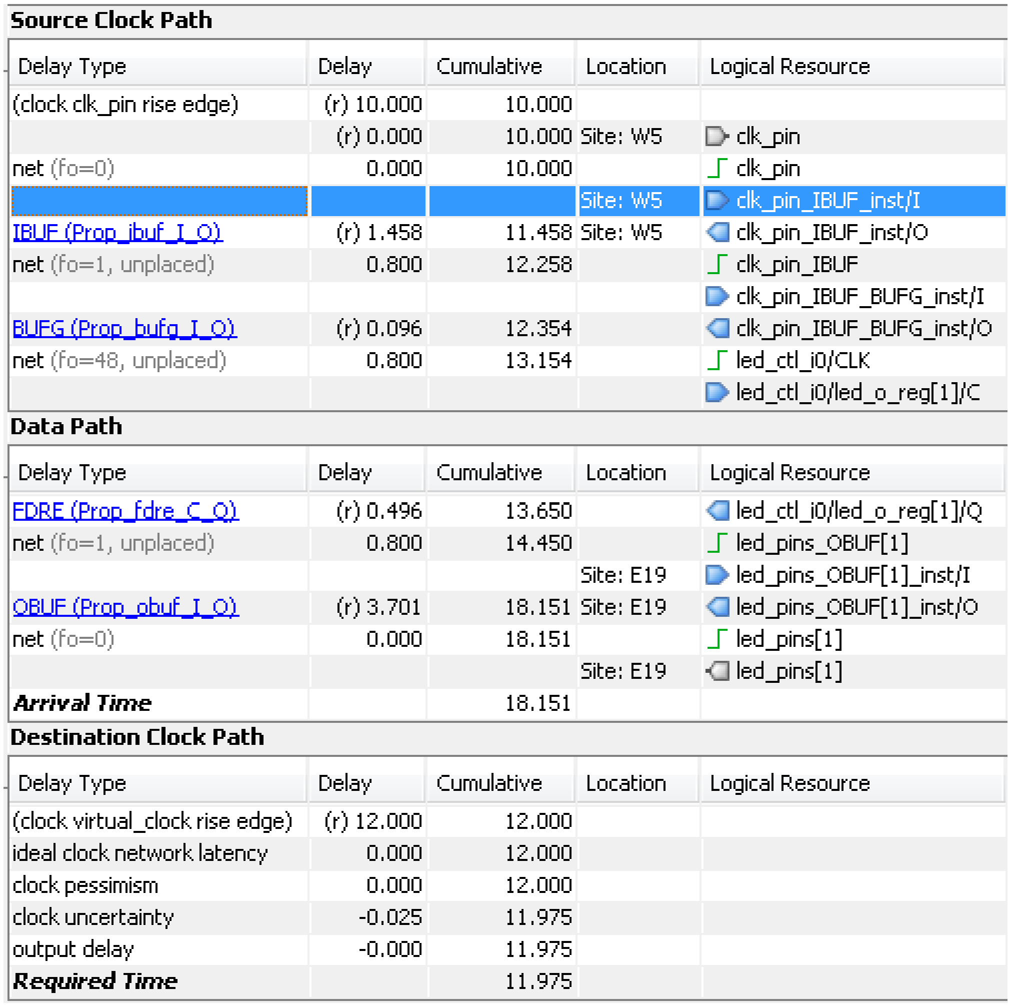
Data Path

Delay Type	Incr (ns)	Path (...	Locati...	Netlist Resource(s)
FDRE (Prop_fdre_C_Q)	(r) 0.478	13.443		led_ctl_i0/led_o_reg[2]/Q
net (fo=1, unplaced)	0.803	14.246		led_pins_OBUF[2]
			Sit...13	led_pins_OBUF[2]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.728	17.973	Sit...13	led_pins_OBUF[2]_inst/O
net (fo=0)	0.000	17.973		led_pins[2]
			Sit...13	led_pins[2]
Arrival Time		17.973		

Destination Clock Path

Delay Type	Incr (ns)	Path (...	Loca...	Netlist Resourc...
(clock virtual...ck rise edge)	(r) 12.000	12.000		
ideal clock network latency	0.000	12.000		
clock pessimism	0.000	12.000		
clock uncertainty	-0.025	11.975		
output delay	-0.000	11.975		
Required Time		11.975		

Figure 7. Worst failing path for the Nexys4 DDR



Source Clock Path

Delay Type	Delay	Cumulative	Location	Logical Resource
(clock clk_pin rise edge)	(r) 10.000	10.000		
	(r) 0.000	10.000	Site: W5	clk_pin
net (fo=0)	0.000	10.000		clk_pin
			Site: W5	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.458	11.458	Site: W5	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.800	12.258		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.096	12.354		clk_pin_IBUF_BUFG_inst/O
net (fo=48, unplaced)	0.800	13.154		led_ctl_i0/CLK
				led_ctl_i0/led_o_reg[1]/C

Data Path

Delay Type	Delay	Cumulative	Location	Logical Resource
FDRE (Prop_fdre_C_Q)	(r) 0.496	13.650		led_ctl_i0/led_o_reg[1]/Q
net (fo=1, unplaced)	0.800	14.450		led_pins_OBUF[1]
			Site: E19	led_pins_OBUF[1]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.701	18.151	Site: E19	led_pins_OBUF[1]_inst/O
net (fo=0)	0.000	18.151		led_pins[1]
			Site: E19	led_pins[1]
Arrival Time		18.151		

Destination Clock Path

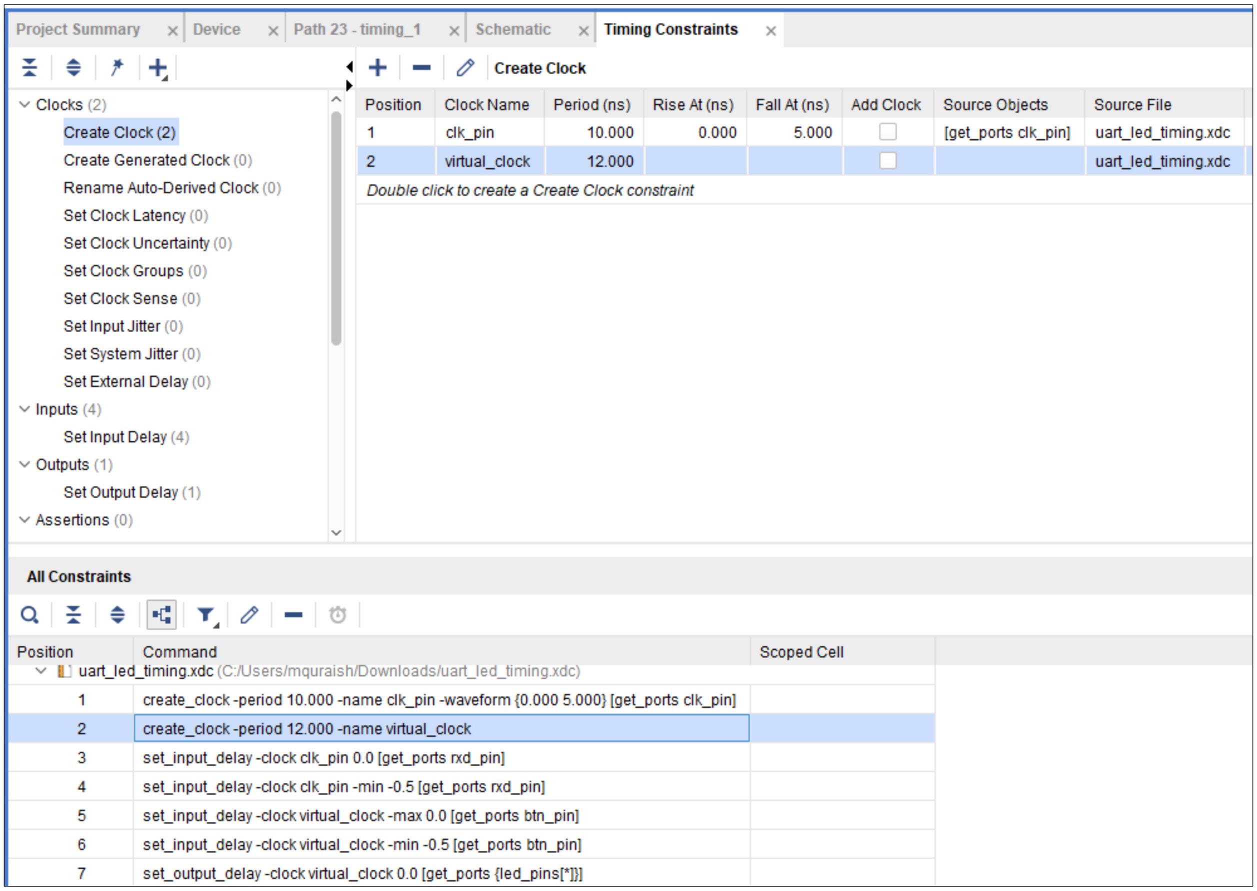
Delay Type	Delay	Cumulative	Location	Logical Resource
(clock virtual_clock rise edge)	(r) 12.000	12.000		
ideal clock network latency	0.000	12.000		
clock pessimism	0.000	12.000		
clock uncertainty	-0.025	11.975		
output delay	-0.000	11.975		
Required Time		11.975		

Figure 7. Worst failing path for the Basys3

1-3. Change the design constraint to constrain the virtual clock period to 10ns. Re-synthesize the design and analyze the results.

1-3-1. Click **Edit Timing Constraints** under the Synthesized Design.

The Timing Constraints GUI will appear, showing the design has two create clocks, four inputs, and one output constraints. It also shows the constraints in the text form in the All Constraints section.



Position	Clock Name	Period (ns)	Rise At (ns)	Fall At (ns)	Add Clock	Source Objects	Source File
1	clk_pin	10.000	0.000	5.000	□	[get_ports clk_pin]	uart_led_timing.xdc
2	virtual_clock	12.000			□		uart_led_timing.xdc

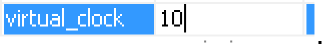
Double click to create a Create Clock constraint

uart_led_timing.xdc (C:/Users/mquraish/Downloads/uart_led_timing.xdc)

Position	Command	Scoped Cell
1	create_clock -period 10.000 -name clk_pin -waveform {0.000 5.000} [get_ports clk_pin]	
2	create_clock -period 12.000 -name virtual_clock	
3	set_input_delay -clock clk_pin 0.0 [get_ports rxd_pin]	
4	set_input_delay -clock clk_pin -min -0.5 [get_ports rxd_pin]	
5	set_input_delay -clock virtual_clock -max 0.0 [get_ports btn_pin]	
6	set_input_delay -clock virtual_clock -min -0.5 [get_ports btn_pin]	
7	set_output_delay -clock virtual_clock 0.0 [get_ports {led_pins[*]}]	

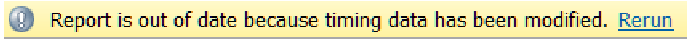
Figure 8. Timing Constraints showing 12 ns Virtual Clock period defined

1-3-2. Click in the **Period** cell of the *virtual_clock* and change the period from **12** to **10**


virtual_clock | 10

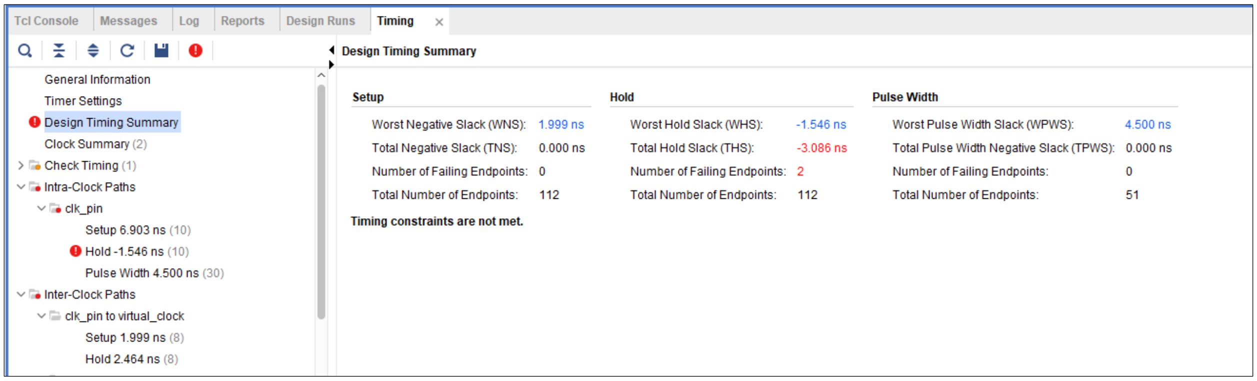
1-3-3. Click **Apply**.

Note that since the timing constraint has changed, a warning message in the console pane is displayed to rerun the report.


Report is out of date because timing data has been modified. Rerun

1-3-4. Click on **Rerun**.

Notice that setup timing violations are gone. However, there are still 2 failing paths for the Hold.

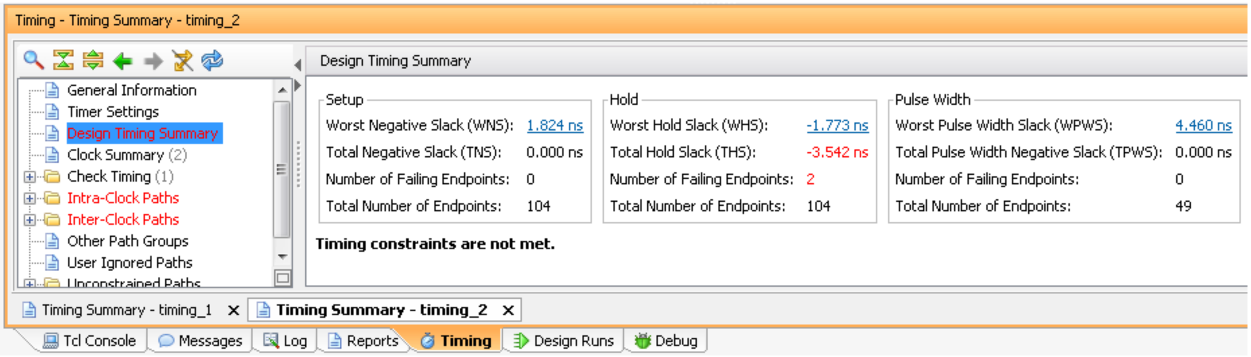


Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	1.999 ns	Worst Hold Slack (WHS):	-1.546 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	-3.086 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	2	Number of Failing Endpoints:	0
Total Number of Endpoints:	112	Total Number of Endpoints:	112	Total Number of Endpoints:	51

Timing constraints are not met.

Figure 9. Setup timing met for the Nexys4 DDR



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	1.824 ns	Worst Hold Slack (WHS):	-1.773 ns	Worst Pulse Width Slack (WPWS):	4.460 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	-3.542 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	2	Number of Failing Endpoints:	0
Total Number of Endpoints:	104	Total Number of Endpoints:	104	Total Number of Endpoints:	49

Timing constraints are not met.

Figure 9. Setup timing met for the Basys3

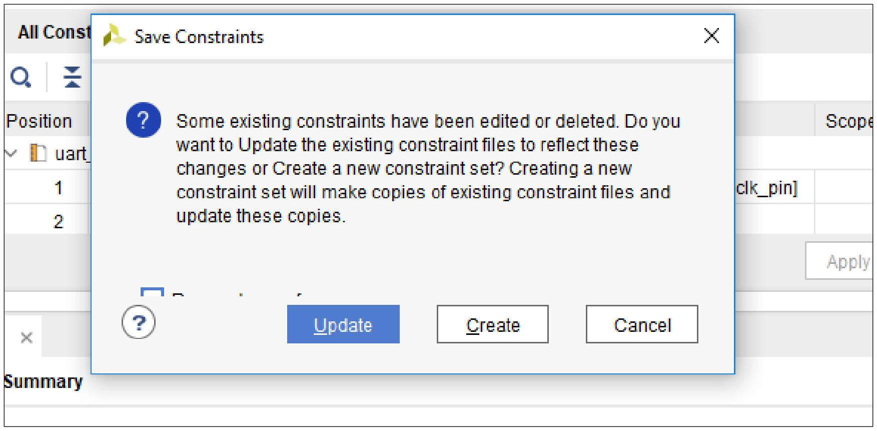
1-3-5. Click on the WHS link to see the paths.

1-3-6. Double-click on the first path to see the timing compositions. Notice that the clock path delay does not include the entire clock period.

1-3-7. Select **File > Save Constraints…**

Click **OK**.

Notice that the Synthesis Out-of-Date status is displayed on the top-right corner. When asked to update the existing constraints file, click on update option.



Some existing constraints have been edited or deleted. Do you want to Update the existing constraint files to reflect these changes or Create a new constraint set? Creating a new constraint set will make copies of existing constraint files and update these copies.

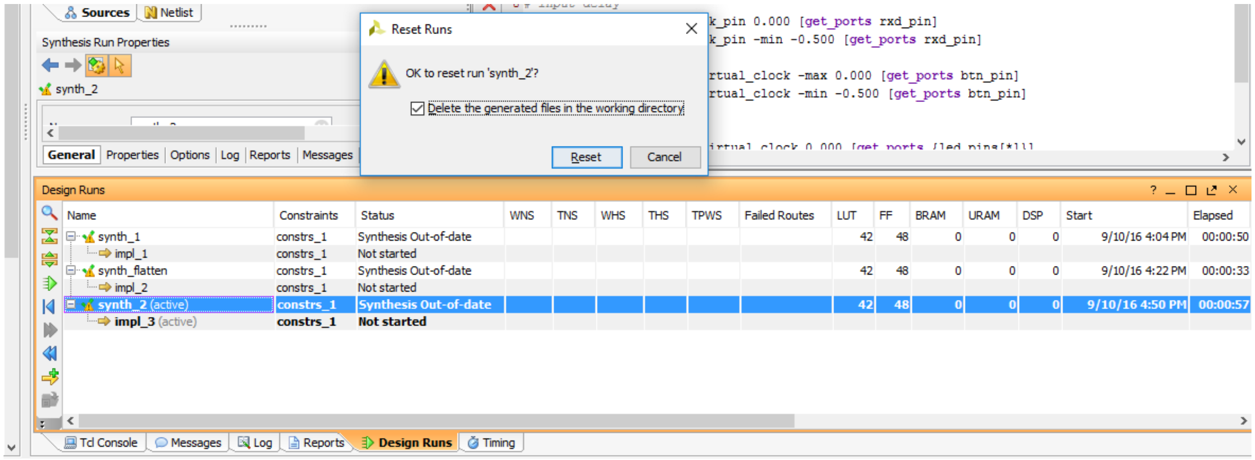
Next select uart_led_timing.xcd as the constraints file to update its contents and click OK.

Note: Accurate hold time check requires accurate estimation of the clock network delay. Since physical design information will by only available after "Implementation", the hold time check at the "Synthesis" stage is usually inaccurate. We could ignore the hold time violations here.

Implement the Design — Step 2

2-1. Run the implementation after saving the synthesis run. Perform the timing analysis.

2-1-1. In the Design Runs tab, right-click on the synth_2 and select **Reset Runs**. Make sure the generated files are deleted. Click **Reset**.



Reset Runs

OK to reset run 'synth_2'?

☑ Delete the generated files in the working directory

Reset | Cancel

```
k_pin 0.000 [get_ports rxd_pin]
k_pin -min -0.500 [get_ports rxd_pin]

rtual_clock -max 0.000 [get_ports btn_pin]
rtual_clock -min -0.500 [get_ports btn_pin]
```

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed
synth_1	constrs_1	Synthesis Out-of-date							42	48	0	0	0	9/10/16 4:04 PM	00:00:50
impl_1	constrs_1	Not started													
synth_flatten	constrs_1	Synthesis Out-of-date							42	48	0	0	0	9/10/16 4:22 PM	00:00:33
impl_2	constrs_1	Not started													
synth_2 (active)	constrs_1	Synthesis Out-of-date							42	48	0	0	0	9/10/16 4:50 PM	00:00:57
impl_3 (active)	constrs_1	Not started													

2-1-2. Click the **Closed Design** button in the status bar. If prompted, do not save anything.

2-1-3. Click on the **Run Implementation** in the *Flow Navigator* pane.

2-1-4. Click **OK** when prompted to run the synthesis first before running the implementation process.

When the implementation is completed, a dialog box will appear with three options.

2-1-5. Select the *Open Implemented Design* option and click **OK**.

2-2. View the amount of FPGA resources consumed by the design using Report Utilization.

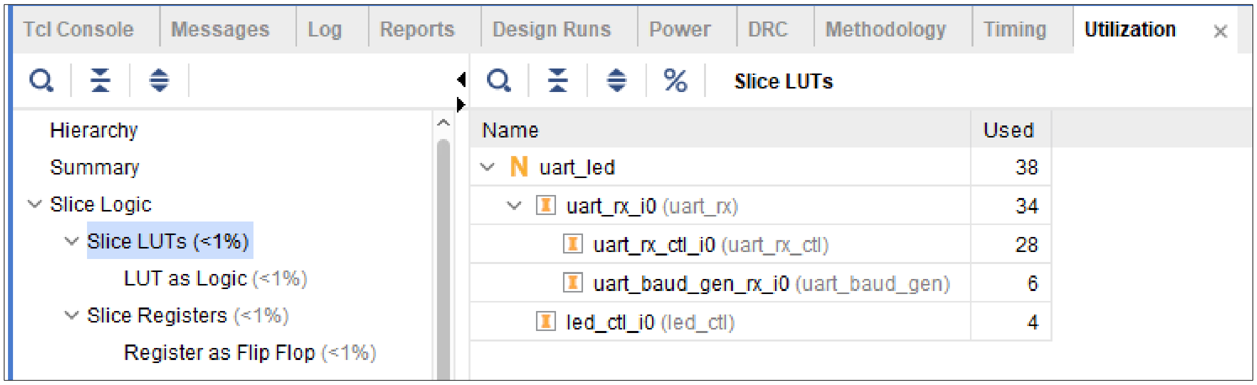
2-2-1. In the *Flow Navigator* pane, select **Open Implemented Design** > **Report Utilization**.

The Report Utilization dialog box opens.

2-2-2. Click **OK**.

The utilization report is displayed at the bottom of the Vivado IDE. You can select any of the resources on the left to view its corresponding utilization.

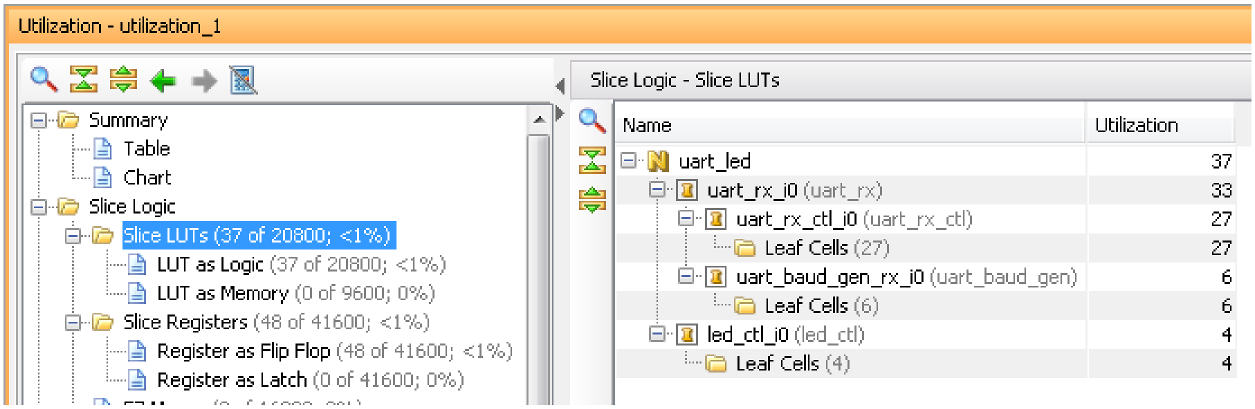
2-2-3. Select Slice LUTs to view how much and which module consumes the resource.



Hierarchy
Summary
Slice Logic
Slice LUTs (<1%)
LUT as Logic (<1%)
Slice Registers (<1%)
Register as Flip Flop (<1%)

Name	Used
uart_led	38
uart_rx_i0 (uart_rx)	34
uart_rx_ctl_i0 (uart_rx_ctl)	28
uart_baud_gen_rx_i0 (uart_baud_gen)	6
led_ctl_i0 (led_ctl)	4

Figure 10. Resource utilization for the Nexys4 DDR



Summary
Table
Chart
Slice Logic
Slice LUTs (37 of 20800; <1%)
LUT as Logic (37 of 20800; <1%)
LUT as Memory (0 of 9600; 0%)
Slice Registers (48 of 41600; <1%)
Register as Flip Flop (48 of 41600; <1%)
Register as Latch (0 of 41600; 0%)

Name	Utilization
uart_led	37
uart_rx_i0 (uart_rx)	33
uart_rx_ctl_i0 (uart_rx_ctl)	27
Leaf Cells (27)	27
uart_baud_gen_rx_i0 (uart_baud_gen)	6
Leaf Cells (6)	6
led_ctl_i0 (led_ctl)	4
Leaf Cells (4)	4

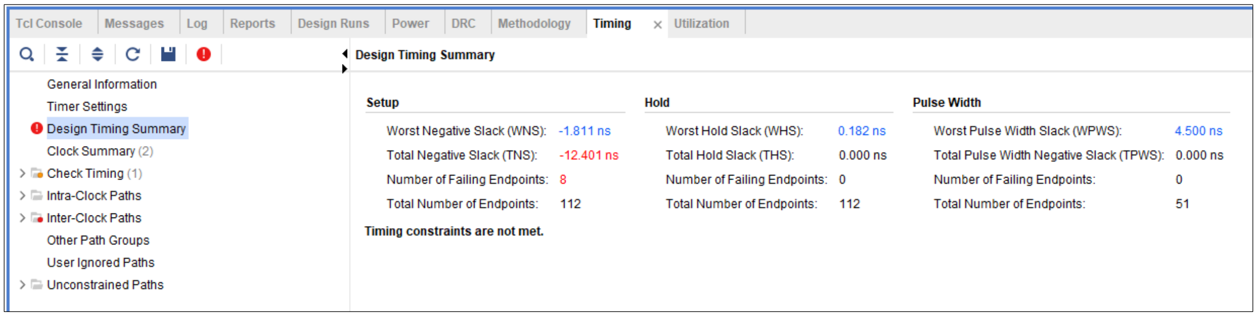
Figure 10. Resource utilization for the Basys3

2-3. Generate a timing summary report.

2-3-1. In the Flow Navigator, under Implementation > Implemented Design, click **Report Timing Summary**

The Report Timing Summary dialog box opens.

2-3-2. Leave all the settings unchanged and click **OK** to generate the report.



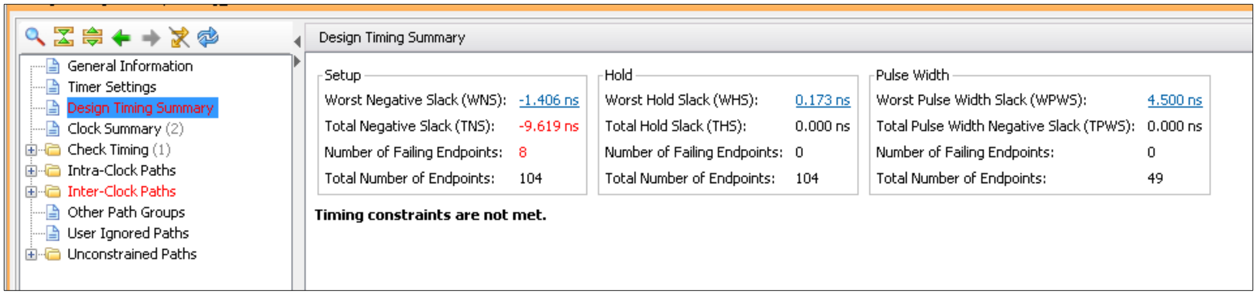
General Information
Timer Settings
Design Timing Summary
Clock Summary (2)
Check Timing (1)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
User Ignored Paths
Unconstrained Paths

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-1.811 ns	Worst Hold Slack (WHS):	0.182 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	-12.401 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	8	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	112	Total Number of Endpoints:	112	Total Number of Endpoints:	51

Timing constraints are not met.

Figure 11. The timing summary report showing timing violations for the Nexys4 DDR



Design Timing Summary

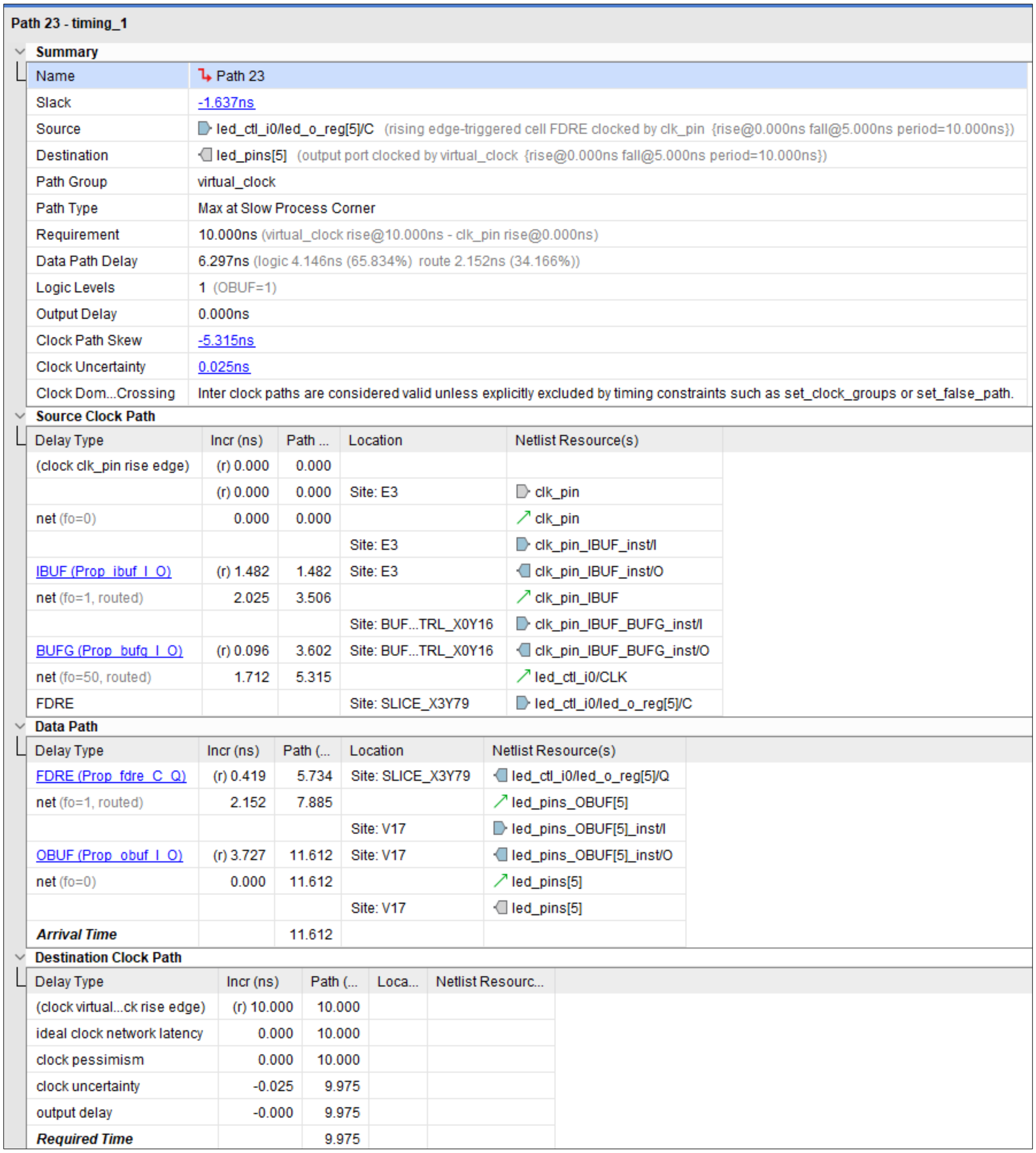
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-1.406 ns	Worst Hold Slack (WHS):	0.173 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	-9.619 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	8	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	104	Total Number of Endpoints:	104	Total Number of Endpoints:	49

Timing constraints are not met.

Figure 11. The timing summary report showing timing violations for the Basys3

2-3-3. Click on the WNS link to see a detailed report to determine the failing path entries.

2-3-4. Double-click on the first failing path to see why it is failing.



Path 23 - timing_1

Summary

Name	Path 23
Slack	-1.637ns
Source	led_ctl_i0/led_o_reg[5]/C (rising edge-triggered cell FDRE clocked by clk_pin {rise@0.000ns fall@5.000ns period=10.000ns})
Destination	led_pins[5] (output port clocked by virtual_clock {rise@0.000ns fall@5.000ns period=10.000ns})
Path Group	virtual_clock
Path Type	Max at Slow Process Corner
Requirement	10.000ns (virtual_clock rise@10.000ns - clk_pin rise@0.000ns)
Data Path Delay	6.297ns (logic 4.146ns (65.834%) route 2.152ns (34.166%))
Logic Levels	1 (OBUF=1)
Output Delay	0.000ns
Clock Path Skew	-5.315ns
Clock Uncertainty	0.025ns
Clock Dom...Crossing	Inter clock paths are considered valid unless explicitly excluded by timing constraints such as set_clock_groups or set_false_path.

Source Clock Path

Delay Type	Incr (ns)	Path ...	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 0.000	0.000		
	(r) 0.000	0.000	Site: E3	clk_pin
net (fo=0)	0.000	0.000		clk_pin
			Site: E3	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.482	1.482	Site: E3	clk_pin_IBUF_inst/O
net (fo=1, routed)	2.025	3.506		clk_pin_IBUF
			Site: BUF...TRL_X0Y16	clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.096	3.602	Site: BUF...TRL_X0Y16	clk_pin_IBUF_BUFG_inst/O
net (fo=50, routed)	1.712	5.315		led_ctl_i0/CLK
FDRE			Site: SLICE_X3Y79	led_ctl_i0/led_o_reg[5]/C

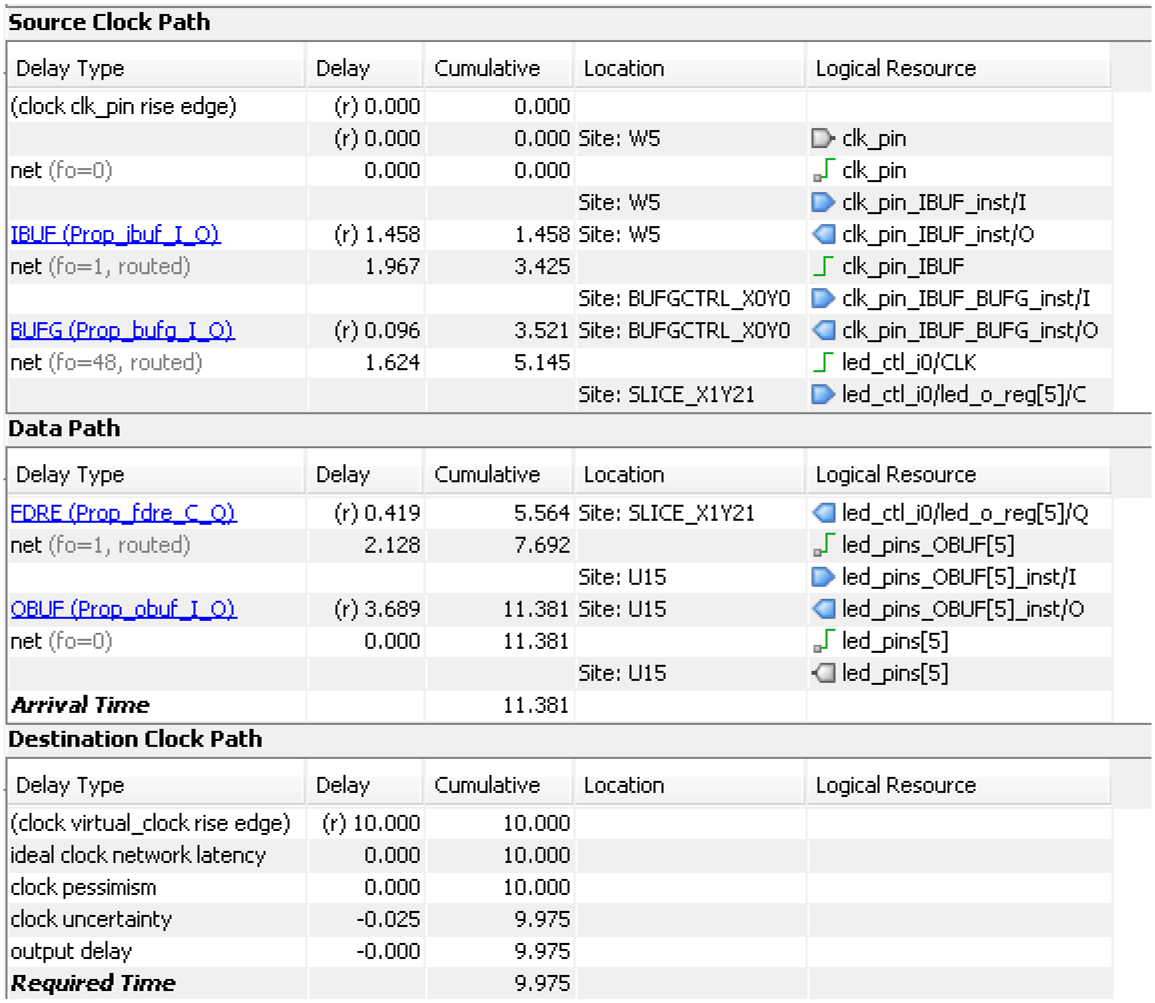
Data Path

Delay Type	Incr (ns)	Path (...	Location	Netlist Resource(s)
FDRE (Prop_fdre_C_Q)	(r) 0.419	5.734	Site: SLICE_X3Y79	led_ctl_i0/led_o_reg[5]/Q
net (fo=1, routed)	2.152	7.885		led_pins_OBUF[5]
			Site: V17	led_pins_OBUF[5]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.727	11.612	Site: V17	led_pins_OBUF[5]_inst/O
net (fo=0)	0.000	11.612		led_pins[5]
			Site: V17	led_pins[5]
Arrival Time		11.612		

Destination Clock Path

Delay Type	Incr (ns)	Path (...	Loca...	Netlist Resourc...
(clock virtual...ck rise edge)	(r) 10.000	10.000		
ideal clock network latency	0.000	10.000		
clock pessimism	0.000	10.000		
clock uncertainty	-0.025	9.975		
output delay	-0.000	9.975		
Required Time		9.975		

Figure 12. First failing path delays for the Nexys4 DDR



Source Clock Path

Delay Type	Delay	Cumulative	Location	Logical Resource
(clock clk_pin rise edge)	(r) 0.000	0.000		
	(r) 0.000	0.000	Site: W5	clk_pin
net (fo=0)	0.000	0.000		clk_pin
			Site: W5	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.458	1.458	Site: W5	clk_pin_IBUF_inst/O
net (fo=1, routed)	1.967	3.425		clk_pin_IBUF
			Site: BUFGCTRL_X0Y0	clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.096	3.521	Site: BUFGCTRL_X0Y0	clk_pin_IBUF_BUFG_inst/O
net (fo=48, routed)	1.624	5.145		led_ctl_i0/CLK
			Site: SLICE_X1Y21	led_ctl_i0/led_o_reg[5]/C

Data Path

Delay Type	Delay	Cumulative	Location	Logical Resource
FDRE (Prop_fdre_C_Q)	(r) 0.419	5.564	Site: SLICE_X1Y21	led_ctl_i0/led_o_reg[5]/Q
net (fo=1, routed)	2.128	7.692		led_pins_OBUF[5]
			Site: U15	led_pins_OBUF[5]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.689	11.381	Site: U15	led_pins_OBUF[5]_inst/O
net (fo=0)	0.000	11.381		led_pins[5]
			Site: U15	led_pins[5]
Arrival Time		11.381		

Destination Clock Path

Delay Type	Delay	Cumulative	Location	Logical Resource
(clock virtual_clock rise edge)	(r) 10.000	10.000		
ideal clock network latency	0.000	10.000		
clock pessimism	0.000	10.000		
clock uncertainty	-0.025	9.975		
output delay	-0.000	9.975		
Required Time		9.975		

Figure 12. First failing path delays for the Basys3

Note: Compared to delays from the Synthesis report, the net delays in the Implementation report are actual delays (rather than an estimated figure). This indicates that the overhead of interconnects after placement and routing may cause the timing to fail again if you have small slacks in Synthesis.

At this point we can ignore these setup time violations with respect to LED outputs. Because the LED display are actually not sequential logics. Therefore, neither setup nor hold time needs to be checked at the LED outputs. We set the output delay in the constraint file just for illustration purposes. We can also set the output delay to -2 ns to make the timings meet.

2-3-5. Select **Implemented Design > Edit Timing Constraints** the *Flow Navigator* pane.

2-3-6. Select the *Set Output Delay* entry in the left pane, and change the Delay Value on virtual clock to **-2.000** ns.

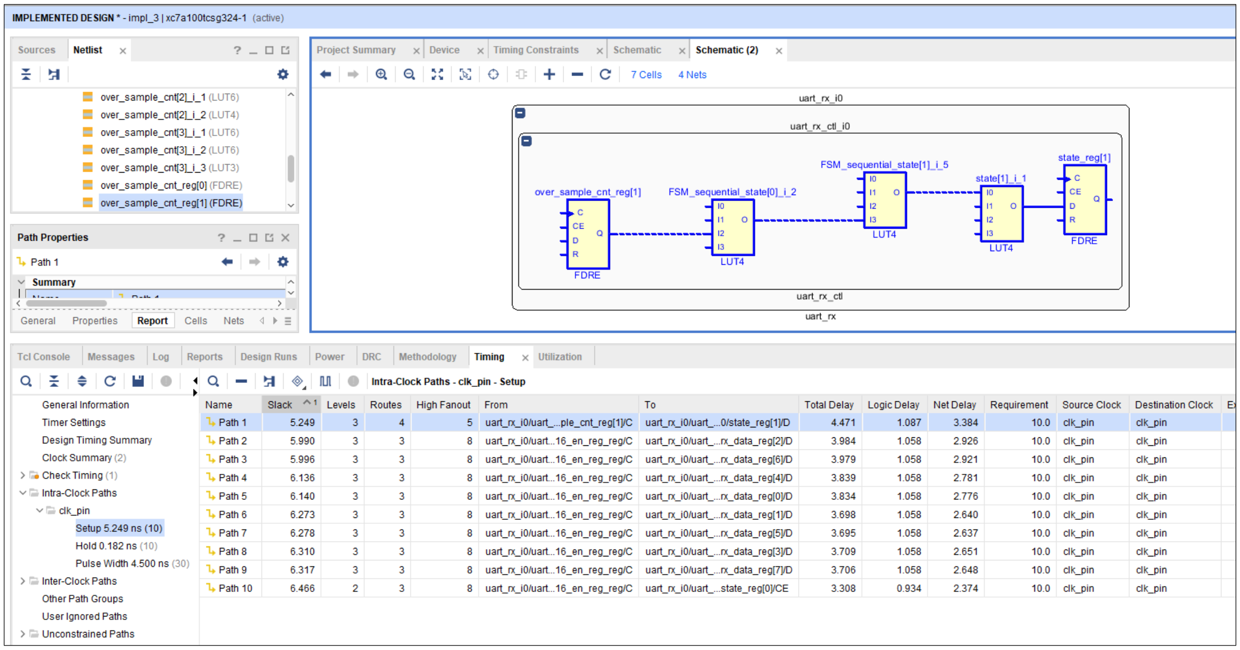
2-3-7. Click **Apply**.

2-3-8. Click **Rerun** link to re-run the timing report.

Observe that the timing violations of the Intra-clock paths are gone.

2-3-9. Expand the **Intra-Clock Paths** folder on the left, expand *clk_pin*, and select the Setup group to see the list of 10 worst case delays on the right side.

2-3-10. Double-click on the any path to see how that is made up of. Also right-click on it and select **Schematic.**



Intra-Clock Paths - clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock	Destination Clock
Path 1	5.249	3	4	5	uart_rx_i0/uart_...ple_cnt_reg[1]/C	uart_rx_i0/uart_...0/state_reg[1]/D	4.471	1.087	3.384	10.0	clk_pin	clk_pin
Path 2	5.990	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[2]/D	3.984	1.058	2.926	10.0	clk_pin	clk_pin
Path 3	5.996	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[6]/D	3.979	1.058	2.921	10.0	clk_pin	clk_pin
Path 4	6.136	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[4]/D	3.839	1.058	2.781	10.0	clk_pin	clk_pin
Path 5	6.140	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[0]/D	3.834	1.058	2.776	10.0	clk_pin	clk_pin
Path 6	6.273	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[1]/D	3.698	1.058	2.640	10.0	clk_pin	clk_pin
Path 7	6.278	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[5]/D	3.695	1.058	2.637	10.0	clk_pin	clk_pin
Path 8	6.310	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[3]/D	3.709	1.058	2.651	10.0	clk_pin	clk_pin
Path 9	6.317	3	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...rx_data_reg[7]/D	3.706	1.058	2.648	10.0	clk_pin	clk_pin
Path 10	6.466	2	3	8	uart_rx_i0/uart...16_en_reg_reg/C	uart_rx_i0/uart_...state_reg[0]/CE	3.308	0.934	2.374	10.0	clk_pin	clk_pin

Figure 13. Schematic view of the worst path delay

2-3-11. Click on the **Device** tab and see the highlighted path in the view.

You will have to zoom-in to see the path.

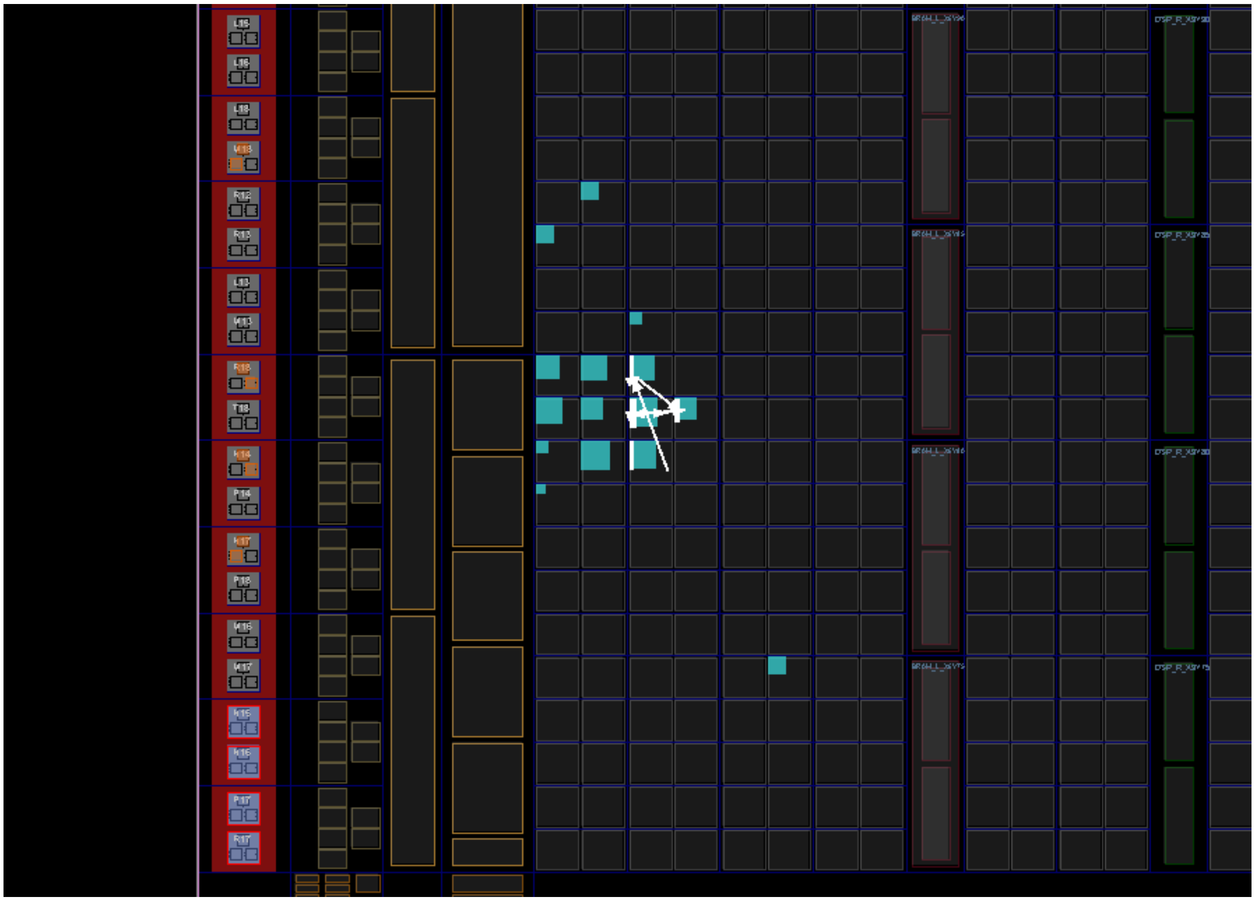


Figure 14. Path displayed in the Device view

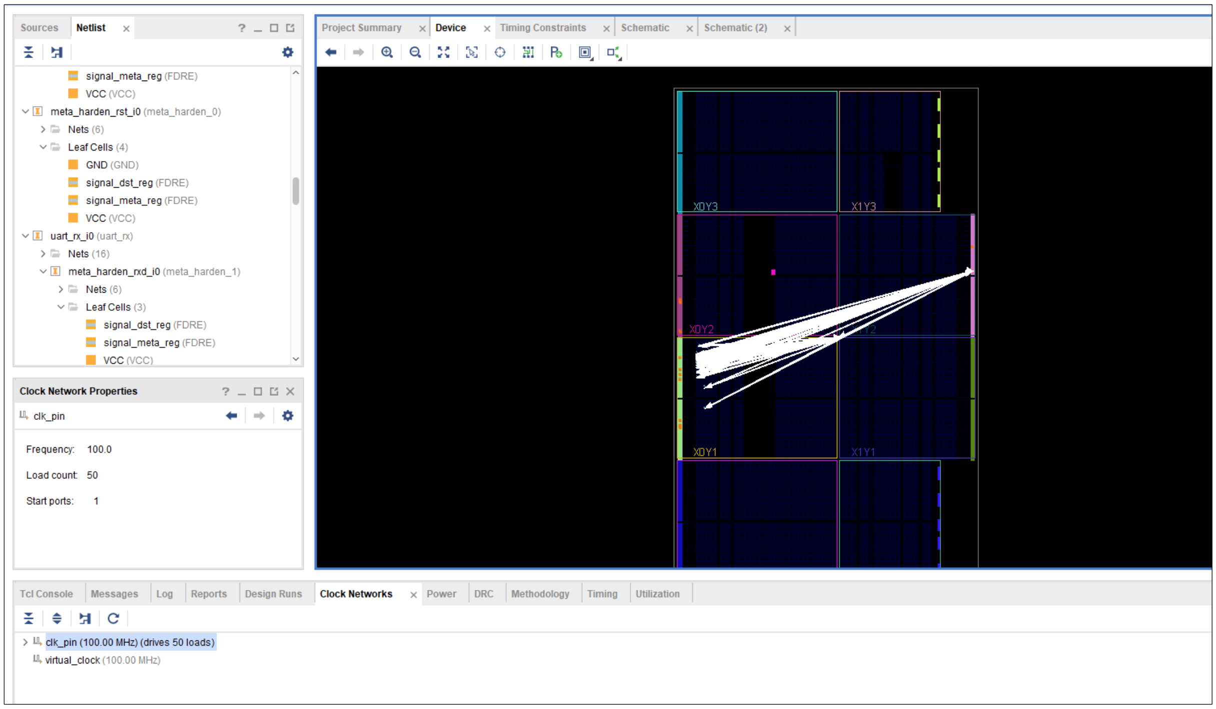
2-3-12. Select **Implemented Design > Report Clock Networks**.

2-3-13. Click **OK**.

The Clock Networks report will be displayed in the Console pane showing two clock net entries.

2-3-14. Select *clk_pin* entry and observe the selected nets in the Device view.

The clock nets are spread across multiple clock regions.



signal_meta_reg (FDRE)
VCC (VCC)
meta_harden_rst_i0 (meta_harden_0)
Nets (6)
Leaf Cells (4)
GND (GND)
signal_dst_reg (FDRE)
signal_meta_reg (FDRE)
VCC (VCC)
uart_rx_i0 (uart_rx)
Nets (16)
meta_harden_rxd_i0 (meta_harden_1)
Nets (6)
Leaf Cells (3)
signal_dst_reg (FDRE)
signal_meta_reg (FDRE)
VCC (VCC)

Clock Network Properties

clk_pin

Frequency: 100.0

Load count: 50

Start ports: 1

clk_pin (100.00 MHz) (drives 50 loads)
virtual_clock (100.00 MHz)

Figure 15. Clock nets for the Nexys4 DDR

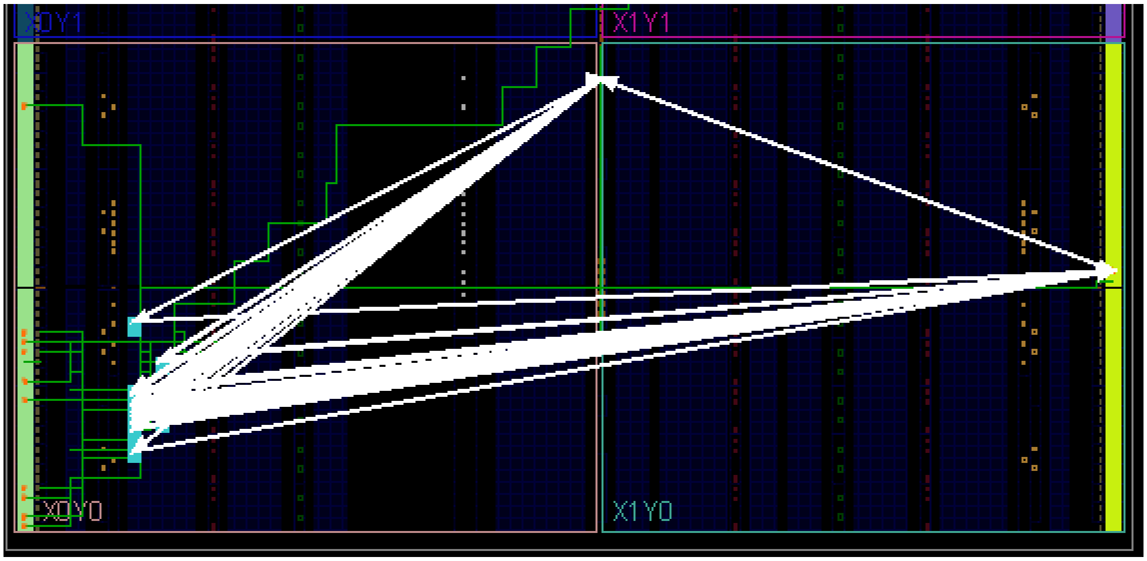


Figure 15. Clock nets for the Basys3

2-3-15. Now, let’s check come timing results of the internal logics. Select *clk_pin* entry under the *Intra-Clock Paths* in the Timing Summary tab. Select the worst case path that has the smallest setup time slack. Check out the detail **Summary** and observe the selected path in the **Device** view to understand the results.

Generate the Bitstream Step 3

3-1. Generate the bitstream.

3-1-1. In the Flow Navigator, under Program and Debug, click **Generate Bitstream.**

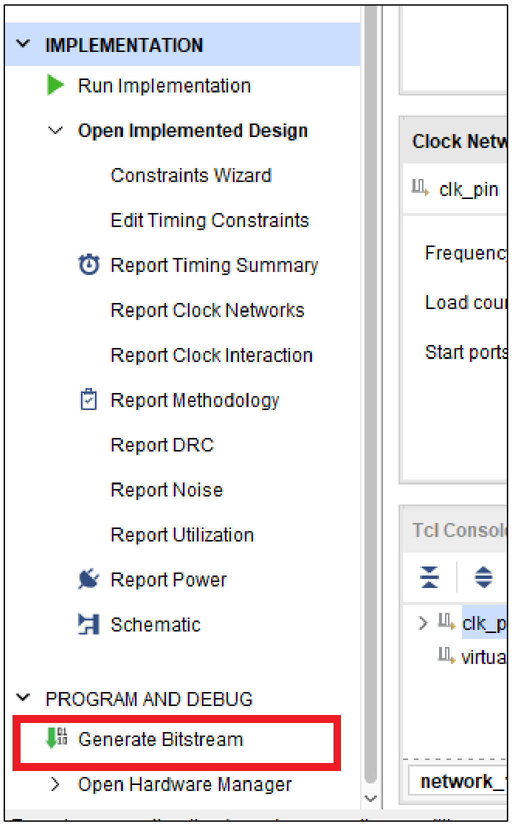


Figure 16, Generating the bitstream

3-1-2. Click **Save** to save the constraints since the timing constraints had been changed and then click **Yes** to reset the runs and re-run all the processes.

3-1-3. The `write_bitstream` command will be executed (you can verify it by looking in the Tcl console).

3-1-4. Click **Cancel** when the bitstream generation is completed.

Verify the Functionality Step 4

4-1. Connect the board and power it ON. Open a hardware session, and program the FPGA.

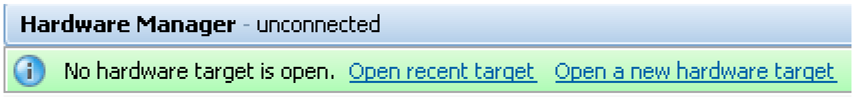
4-1-1. Make sure that the micro-USB cable is connected to the JTAG PROG connector (next to the power supply connector). Make sure that the jumper on the board is set to select USB power (JP3 for the Nexys4 DDR and JP2 for the Basys3).

4-1-2. Select the *Open Hardware Manager* option and click **OK**.

The Hardware Manager window will open indicating “unconnected” status.

4-1-3. Click on the **Open a new hardware target** link.

You can also click on the **Open recent target** link if the board was already targeted before.



Hardware Manager - unconnected

No hardware target is open. Open recent target Open a new hardware target

Figure 17. Opening new hardware target

4-1-4. Click **Next** to see the Hardware Server Settings form.

4-1-5. Click **Next** with the Target Hardware selected.

The JTAG cable which uses the digilent_plugin should be detected and identified as a hardware target. It will also show the hardware devices detected in the chain.

4-1-6. Click **Next** and **Finish**.

4-1-7. The Hardware Session status changes from Unconnected to the server name and the device is highlighted. Also notice that the Status indicates that it is not programmed.

4-1-8. Select the device in the *Hardware Device Properties,* and verify that the **uart_led.bit** is selected as the programming file in the General tab.

4-2. Program the FPGA and verify the functionality using a terminal program.

4-2-1. Start a terminal emulator program such as TeraTerm or HyperTerminal.

4-2-1.1. google teraterm for windows and install the program (can use this link http://tera-term.en.lo4d.com/download). Install the software

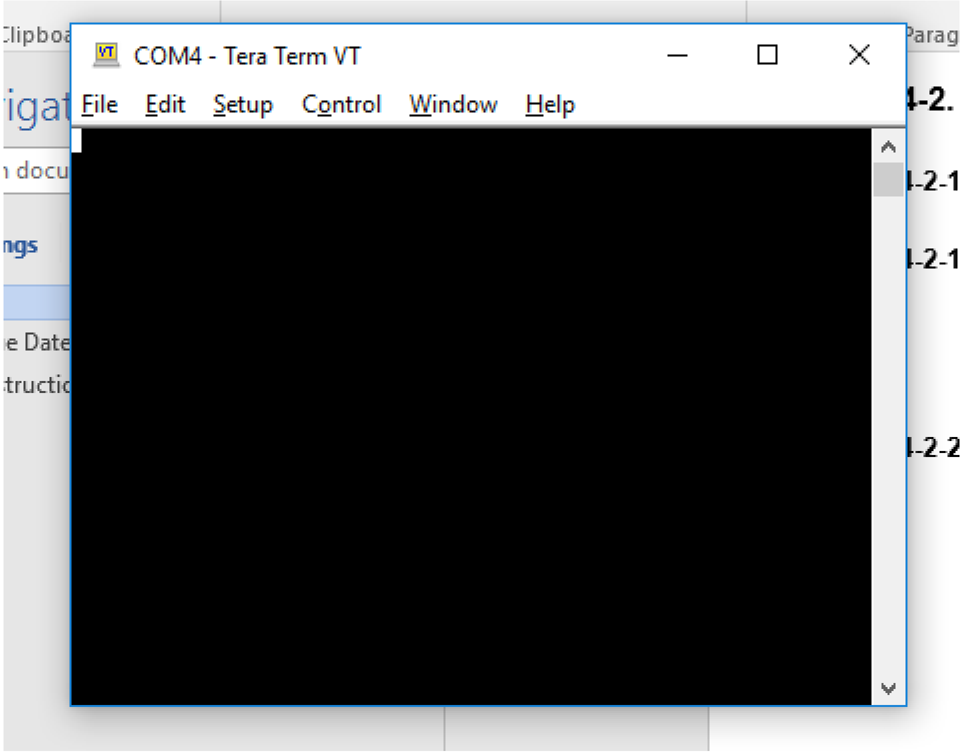


Figure 18. Teraterm window

4-2-2. Select an appropriate COM port (you can find the correct COM number using the Control Panel). If more than one COM ports are given, use both of them and one should work.

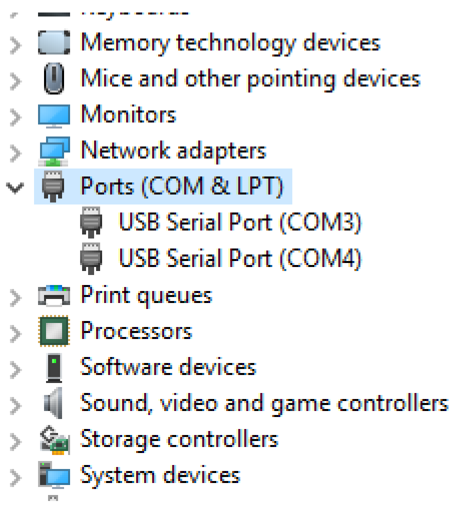
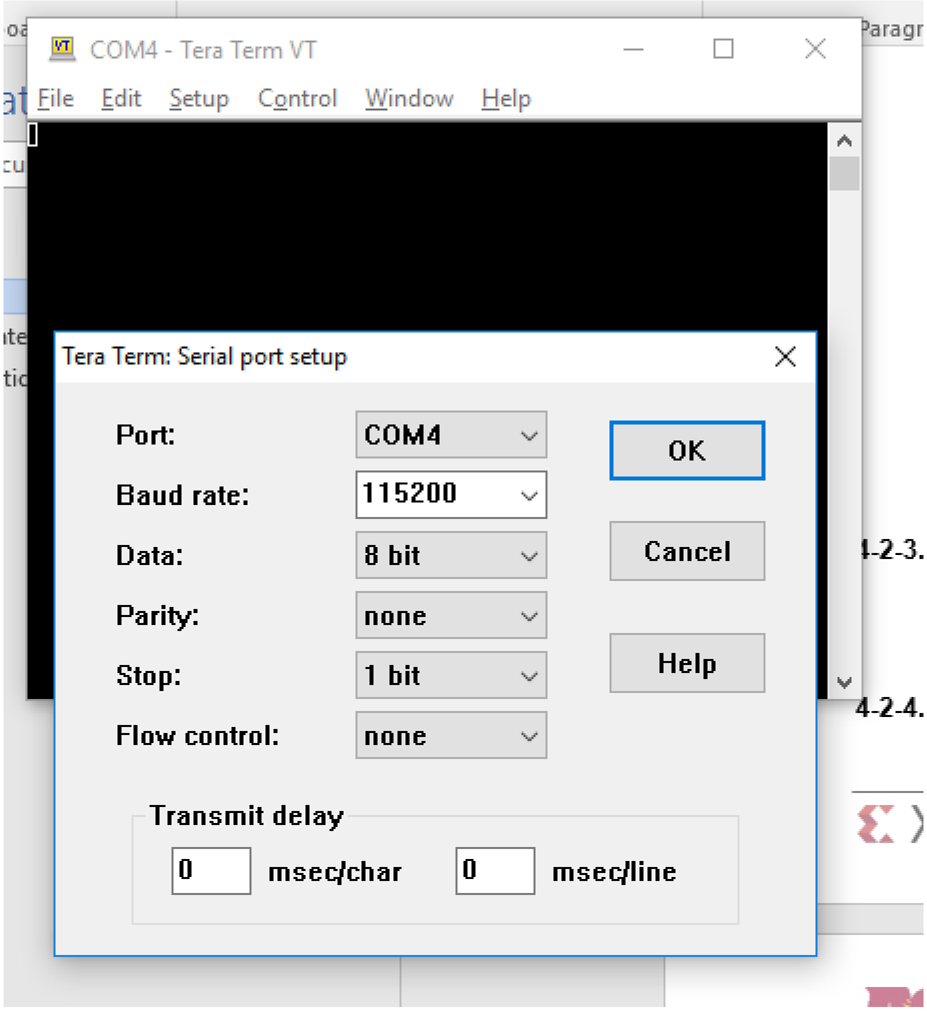


Figure 19, COM ports shown in control panel->device manager

4-2-3. In teraterm, select setup->Serial port.. Set the COM port for 115200 baud rate communication. Keep other options default.

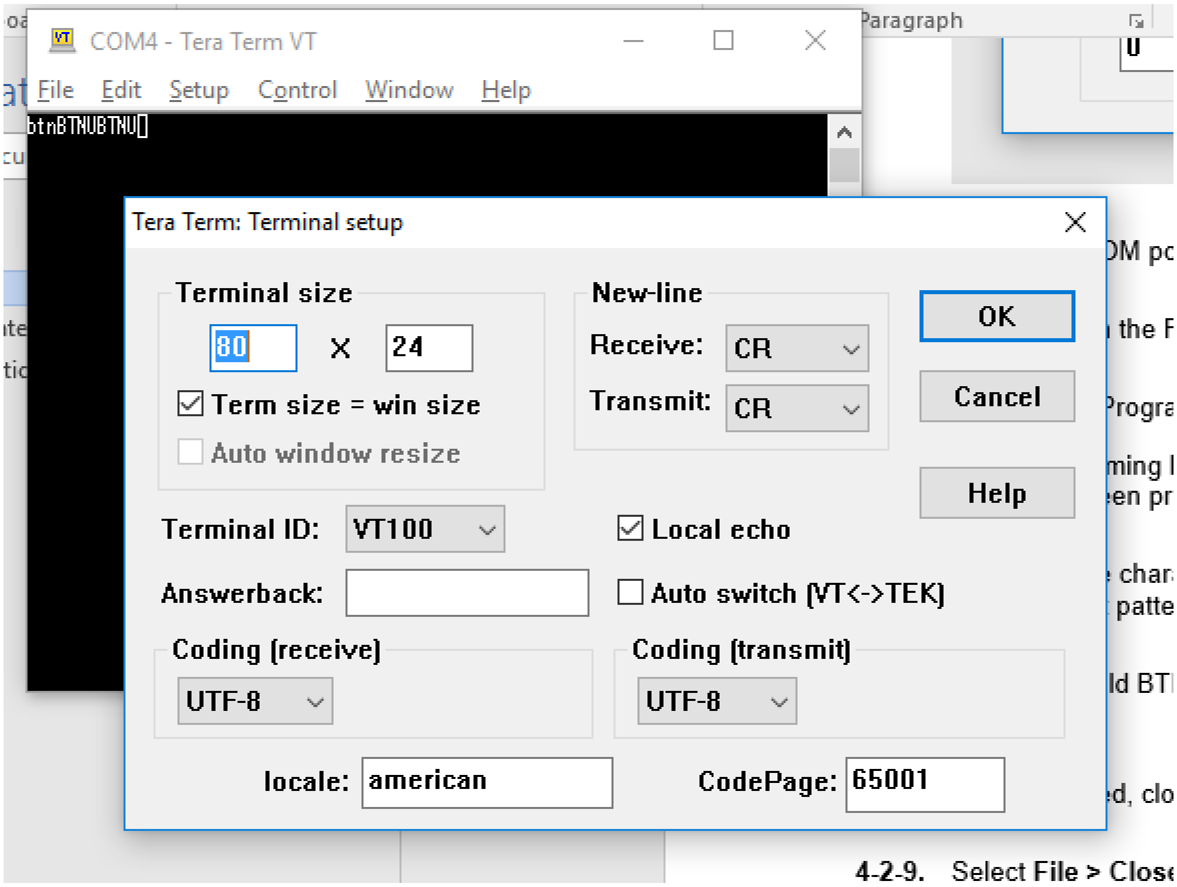


COM4 - Tera Term VT
File Edit Setup Control Window Help
Tera Term: Serial port setup
OK
Cancel
Help

Port: COM4
Baud rate: 115200
Data: 8 bit
Parity: none
Stop: 1 bit
Flow control: none
Transmit delay
0 msec/char 0 msec/line

Figure 20, COM ports settings for baudrate

4-2-4. In teraterm window select setup->terminal and check echo box. This prints whatever is typed in teraterm terminal window.


COM4 - Tera Term VT
File Edit Setup Control Window Help
Tera Term: Terminal setup
OK
Cancel
Help

```
btnBTNUBTNU
```

Terminal size: 80 × 24

☑ Term size = win size

□ Auto window resize

New-line
Receive: CR
Transmit: CR

Terminal ID: VT100

☑ Local echo

Answerback:

□ Auto switch (VT<->TEK)

Coding (receive): UTF-8

Coding (transmit): UTF-8

locale: american

CodePage: 65001

Figure 21, COM ports settings for terminal

4-2-5. Right-click on the FPGA entry in the Hardware window and select Program Device…

4-2-6. Click on the Program button.

The programming bit file will be downloaded and the DONE light will be turned ON when the FPGA has been programmed.

4-2-7. Type in some characters in the terminal emulator window and see the corresponding ASCII equivalent bit pattern displayed on the LEDs. Eg. When A is typed it displays $(01000001)_2$ its binary equivalent.

4-2-8. Press and hold BTNU push button on the board and see the the upper four bits are swapped with the lower four bits on the LEDs.

4-2-9. When satisfied, close the terminal emulator program and power OFF the board.

4-2-10. Select **File > Close Hardware Manager**. Click **OK**.

4-2-11. Close the **Vivado** program by selecting **File > Exit** and click **OK**.

Conclusion

In this tutorial, you learned about many of the reports available to designers in the Vivado IDE. You had the opportunity to learn basic design analysis tools including the Schematic viewer, delay path properties and reports viewer, Device viewer, and selecting primitive parents. You also learned about the basic timing report options that are at your disposal. You verified the functionality in hardware by typing characters on the host machine and seeing the LED pattern changes.

Due Date

Refer to Canvas.

Instructions on submission

1. A submission should include:
    - project folder
    - the bit file generated for FPGA programming: ***xx.bit*** (copy to the outside of project folder)
2. Compress the files into a zip archive file named **cse320-firstname_lastname-tut03.zip**. Note that other codes or any temporary build files should not be included in the submission.
3. Submit the zip archive to Canvas by the due date and time.
4. Failure to follow these instructions may cause TA or instructor to deduct points while grading your assignment.