Implementing the Design

Introduction

This lab continues with the previous lab. 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. Finally the design will be validated by programming the hardware in SDK using the software application running on A9 that is provided to you.

Objectives

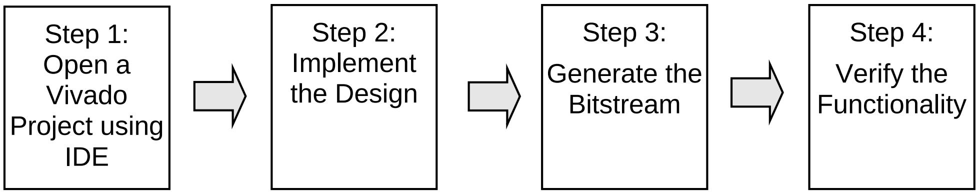
After completing this lab, you will be able to:

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

Procedure

This lab 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 lab.

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 lab2 project. Save the project as lab3 in the *<2016_2_ZYNQ_labs>* 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.

> References to **<2016_2_ZYNQ_labs>** is a placeholder for the **c:\xup\fpga_flow\2016_2_ZYNQ_labs** directory and **<2016_2_ZYNQ_sources>** is a place holder for the **c:\xup\fpga_flow\2016_2_ZYNQ_sources** directory.
>
> Reference to **<board>** means either the **ZedBoard or th**e **Zybo**.

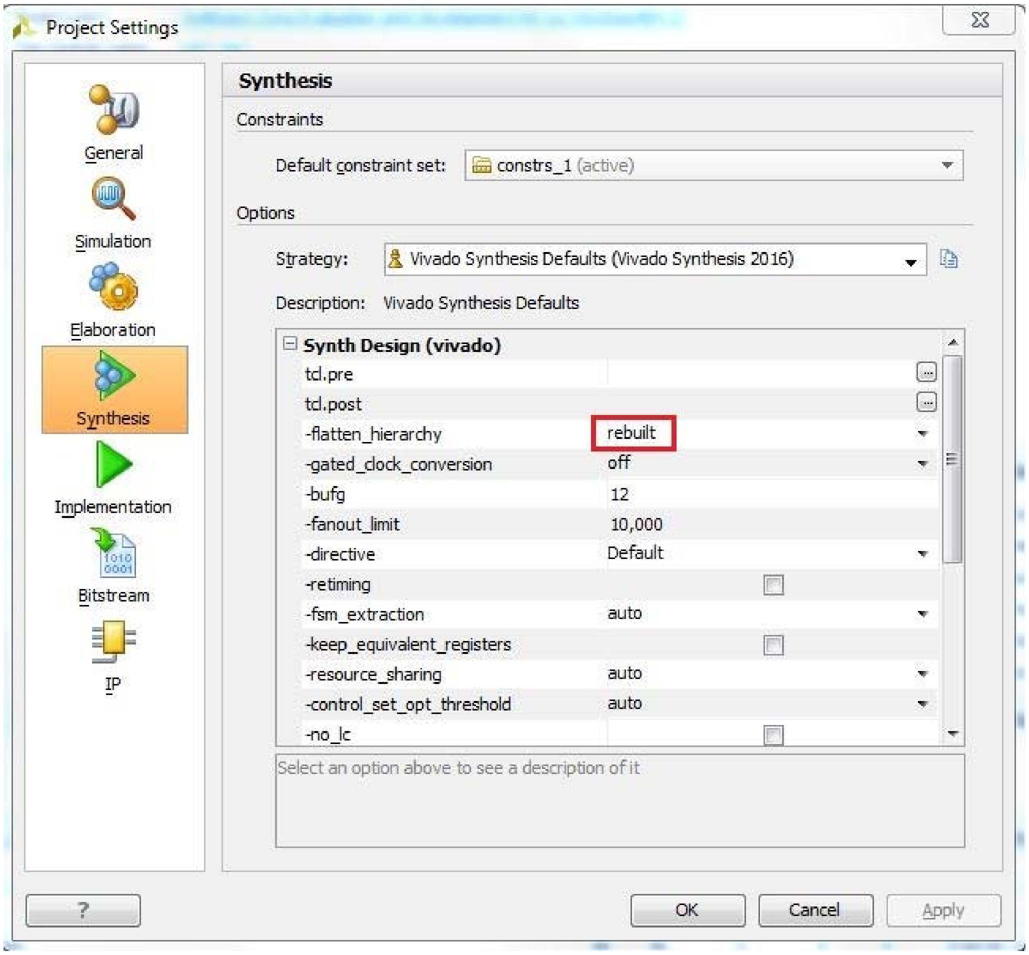
1-1-1. Start the Vivado if necessary and open either the lab2 project (lab2.xpr) you created in the previous lab or the lab2 project in the labsolution directory using the **Open Project** link in the Getting Started page.

© copyright 2016 Xilinx

1-1-2. Select **File > Save Project As …** to open the *Save Project As* dialog box. Enter **lab3** as the project name. Make sure that the *Create Project Subdirectory* option is checked, the project directory path is **<2016_2_ZYNQ_labs>** 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

Constraints

Default constraint set: constrs_1 (active)

Options

Strategy: Vivado Synthesis Defaults (Vivado Synthesis 2016)

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
-keep_equivalent_registers	□
-resource_sharing	auto
-control_set_opt_threshold	auto
-no_lc	□

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.

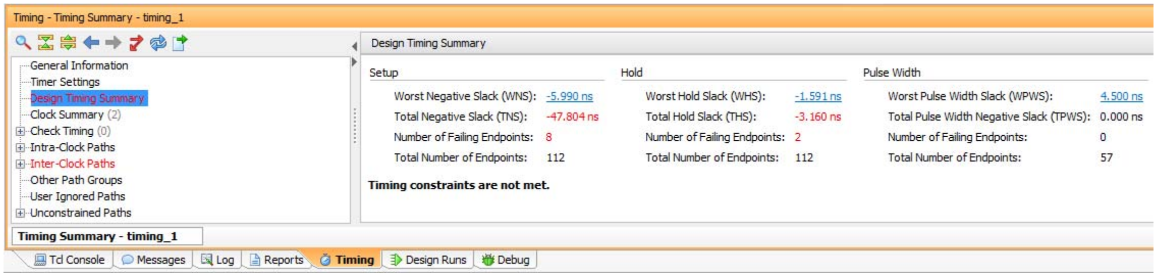
The synthesis process will be run on the uart_top.v and all its hierarchical files. When the process is completed a *Synthesis Completed* dialog box with three options will be displayed.

© copyright 2016 Xilinx

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.*

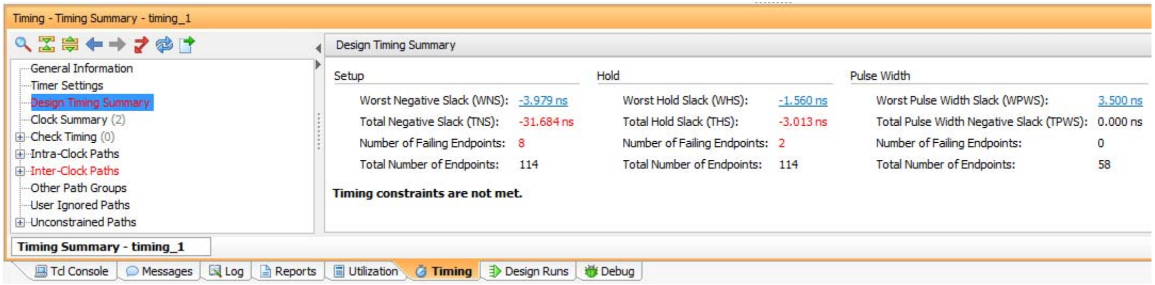


Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-5.990 ns	Worst Hold Slack (WHS):	-1.591 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	-47.804 ns	Total Hold Slack (THS):	-3.160 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:	57

Timing constraints are not met.

Figure 2. Timing report for the ZedBoard



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-3.979 ns	Worst Hold Slack (WHS):	-1.560 ns	Worst Pulse Width Slack (WPWS):	3.500 ns
Total Negative Slack (TNS):	-31.684 ns	Total Hold Slack (THS):	-3.013 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:	114	Total Number of Endpoints:	114	Total Number of Endpoints:	58

Timing constraints are not met.

Figure 2. Timing report for the Zybo

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 25 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 25 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.

© copyright 2016 Xilinx

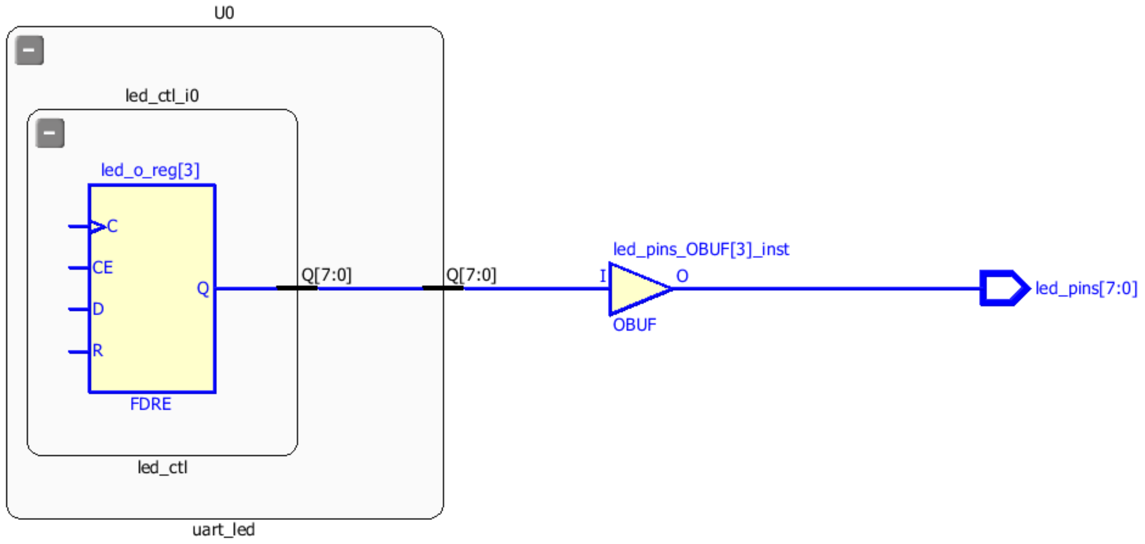


Figure 3. The output data path

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.

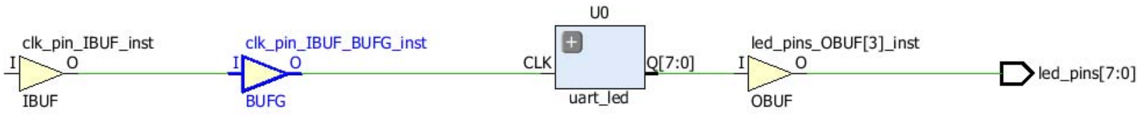


Figure 4. Source to clock port of the FDRE

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

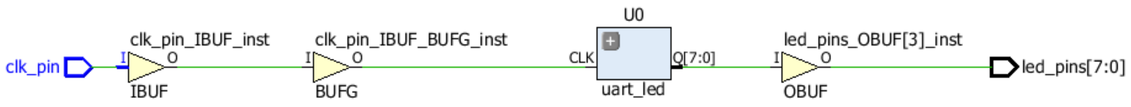
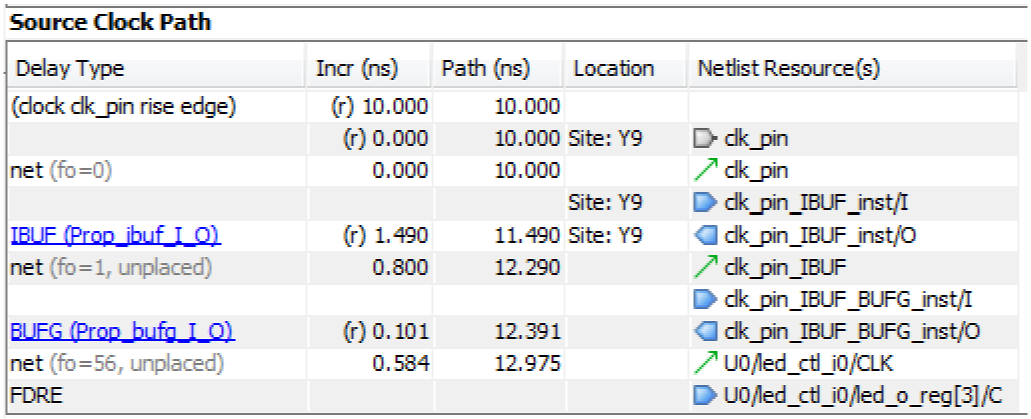


Figure 5. The schematic view of the source clock path

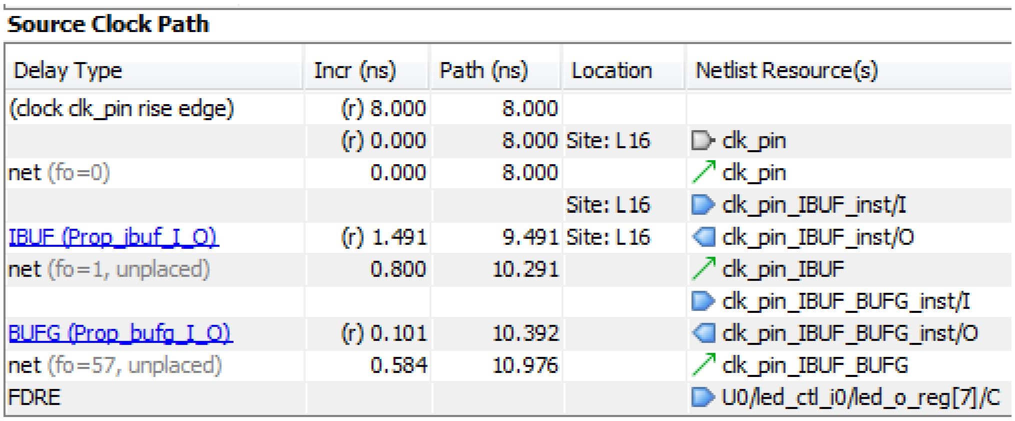
This corresponds to the Source Clock Path in the timing report.



Source Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 10.000	10.000		
	(r) 0.000	10.000	Site: Y9	clk_pin
net (fo=0)	0.000	10.000		clk_pin
			Site: Y9	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.490	11.490	Site: Y9	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.800	12.290		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.101	12.391		clk_pin_IBUF_BUFG_inst/O
net (fo=56, unplaced)	0.584	12.975		U0/led_ctl_i0/CLK
FDRE				U0/led_ctl_i0/led_o_reg[3]/C

Figure 6. The source clock path for the ZedBoard



Source Clock Path

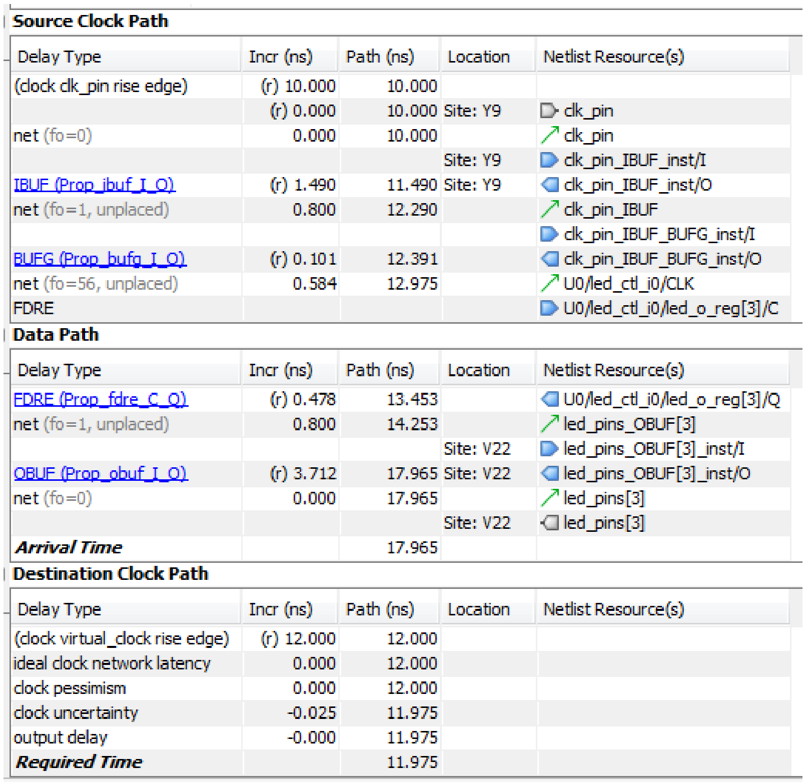
Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 8.000	8.000		
	(r) 0.000	8.000	Site: L16	clk_pin
net (fo=0)	0.000	8.000		clk_pin
			Site: L16	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.491	9.491	Site: L16	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.800	10.291		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.101	10.392		clk_pin_IBUF_BUFG_inst/O
net (fo=57, unplaced)	0.584	10.976		clk_pin_IBUF_BUFG
FDRE				U0/led_ctl_i0/led_o_reg[7]/C

Figure 6. The source clock path for the Zybo

Since the virtual clock is slower (12 ns) than the clk_pin period (10 ns), the data path delay includes the clock period of the clk_pin clock source.

Note: Zybo has an on-board clock of 125 Mhz and hence has the clk_pin period (8 ns), the data path delay includes the clock period of the clk_pin clock source.

© copyright 2016 Xilinx



Source Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 10.000	10.000		
	(r) 0.000	10.000	Site: Y9	clk_pin
net (fo=0)	0.000	10.000		clk_pin
			Site: Y9	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.490	11.490	Site: Y9	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.800	12.290		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.101	12.391		clk_pin_IBUF_BUFG_inst/O
net (fo=56, unplaced)	0.584	12.975		U0/led_ctl_i0/CLK
FDRE				U0/led_ctl_i0/led_o_reg[3]/C

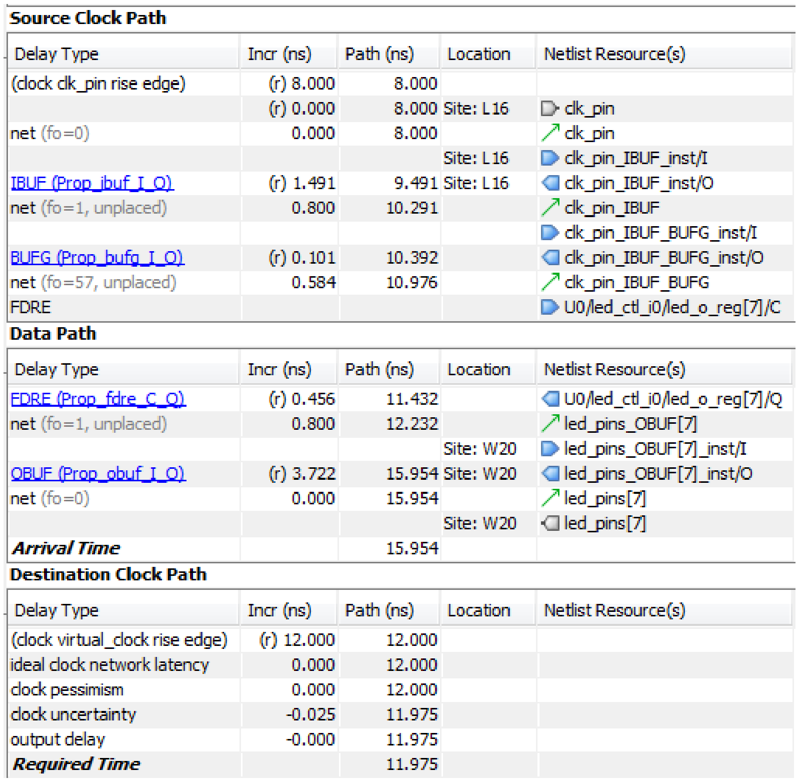
Data Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
FDRE (Prop_fdre_C_Q)	(r) 0.478	13.453		U0/led_ctl_i0/led_o_reg[3]/Q
net (fo=1, unplaced)	0.800	14.253		led_pins_OBUF[3]
			Site: V22	led_pins_OBUF[3]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.712	17.965	Site: V22	led_pins_OBUF[3]_inst/O
net (fo=0)	0.000	17.965		led_pins[3]
			Site: V22	led_pins[3]
Arrival Time		17.965		

Destination Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(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 ZedBoard



Source Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 8.000	8.000		
	(r) 0.000	8.000	Site: L16	clk_pin
net (fo=0)	0.000	8.000		clk_pin
			Site: L16	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.491	9.491	Site: L16	clk_pin_IBUF_inst/O
net (fo=1, unplaced)	0.800	10.291		clk_pin_IBUF
				clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.101	10.392		clk_pin_IBUF_BUFG_inst/O
net (fo=57, unplaced)	0.584	10.976		clk_pin_IBUF_BUFG
FDRE				U0/led_ctl_i0/led_o_reg[7]/C

Data Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
FDRE (Prop_fdre_C_Q)	(r) 0.456	11.432		U0/led_ctl_i0/led_o_reg[7]/Q
net (fo=1, unplaced)	0.800	12.232		led_pins_OBUF[7]
			Site: W20	led_pins_OBUF[7]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.722	15.954	Site: W20	led_pins_OBUF[7]_inst/O
net (fo=0)	0.000	15.954		led_pins[7]
			Site: W20	led_pins[7]
Arrival Time		15.954		

Destination Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(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		

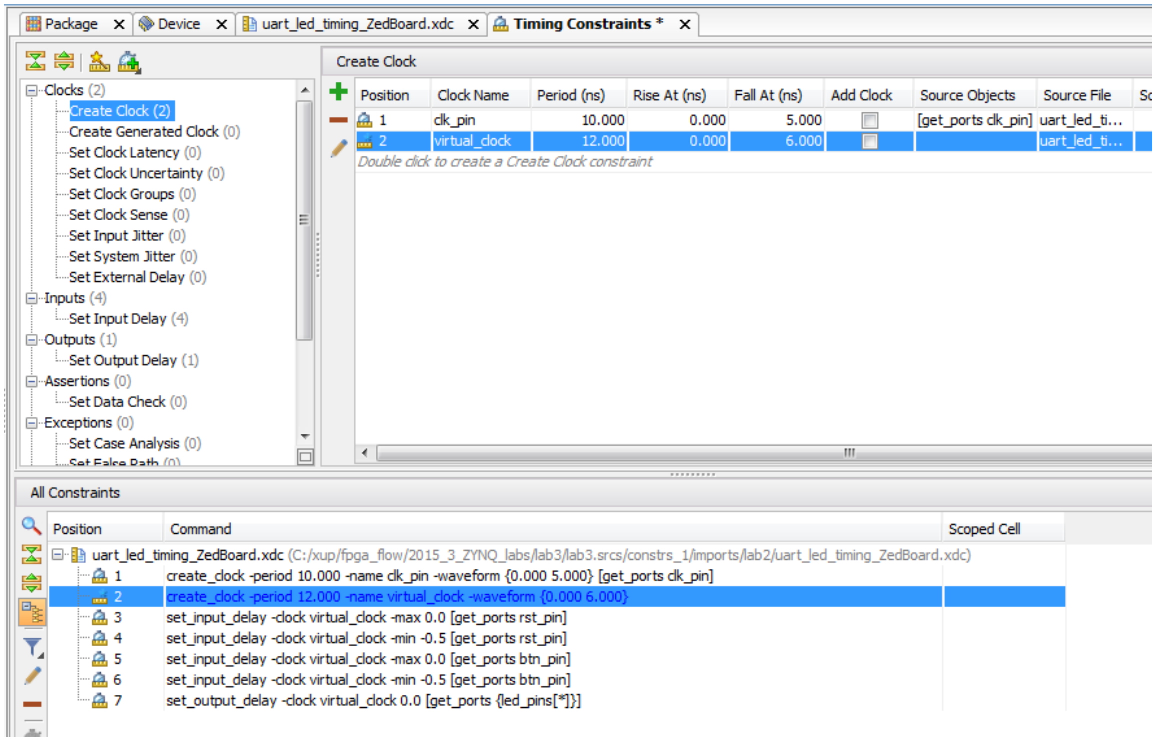
Figure 7. Worst failing path for the Zybo

© copyright 2016 Xilinx

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.



Create Clock

Position	Clock Name	Period (ns)	Rise At (ns)	Fall At (ns)	Add Clock	Source Objects	Source File	Sc
1	clk_pin	10.000	0.000	5.000	□	[get_ports clk_pin]	uart_led_ti...	
2	virtual_clock	12.000	0.000	6.000	□		uart_led_ti...	

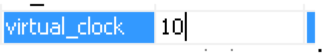
Double click to create a Create Clock constraint

All Constraints

Position	Command	Scoped Cell
uart_led_timing_ZedBoard.xdc (C:/xup/fpga_flow/2015_3_ZYNQ_labs/lab3/lab3.srcs/constrs_1/imports/lab2/uart_led_timing_ZedBoard.xdc)		
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 -waveform {0.000 6.000}	
3	set_input_delay -clock virtual_clock -max 0.0 [get_ports rst_pin]	
4	set_input_delay -clock virtual_clock -min -0.5 [get_ports rst_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

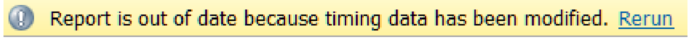


virtual_clock	10

Note: Zybo has an on-board clock of 125 Mhz and hence has the clk_pin period (8 ns), the Virtual_clock period is changed from 12 to 8.

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.

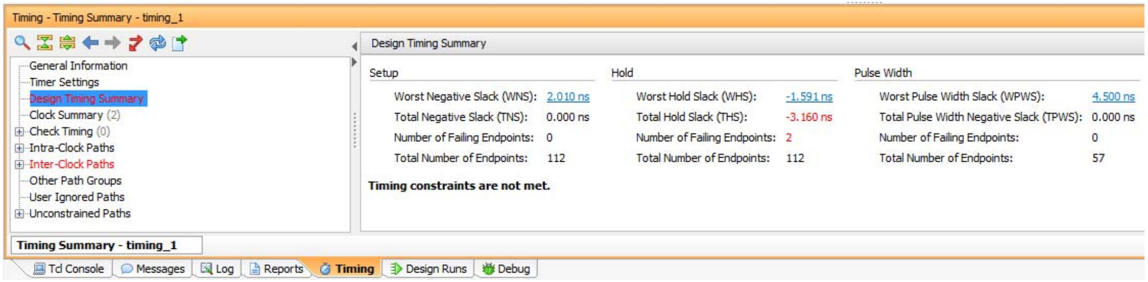


Report is out of date because timing data has been modified. Rerun

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

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

© copyright 2016 Xilinx

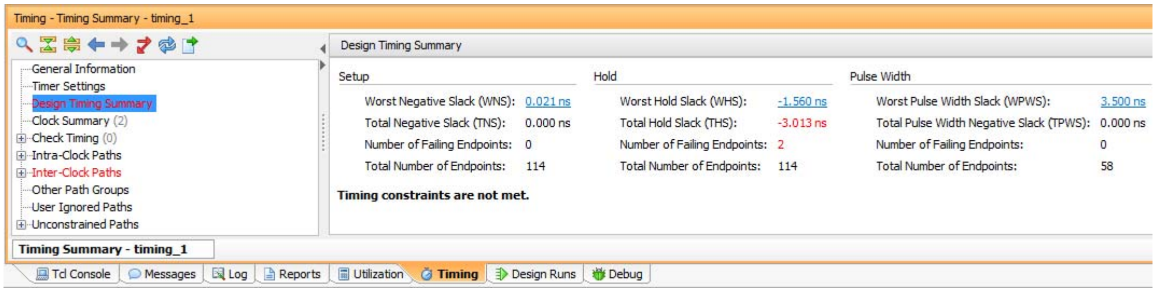


Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.010 ns	Worst Hold Slack (WHS):	-1.591 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	-3.160 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:	57

Timing constraints are not met.

Figure 9. Setup timing met for the ZedBoard



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.021 ns	Worst Hold Slack (WHS):	-1.560 ns	Worst Pulse Width Slack (WPWS):	3.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	-3.013 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:	114	Total Number of Endpoints:	114	Total Number of Endpoints:	58

Timing constraints are not met.

Figure 9. Setup timing met for the Zybo

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…**

1-3-8. Click **Update.**

1-3-9. Click **OK** and then **Yes** to save the synthesized design.

Notice that the Synthesis Out-of-Date status is displayed on the top-right corner.

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**.

2-1-2. Click the **Close Design** link 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.

© copyright 2016 Xilinx

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.

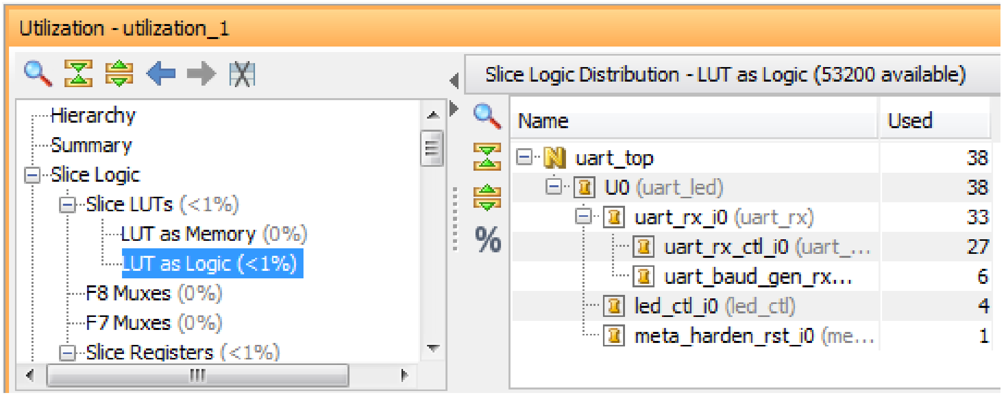
2-2-1. In the *Flow Navigator* pane, select **Open Implemented Design** > **Report Utilization**.

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.

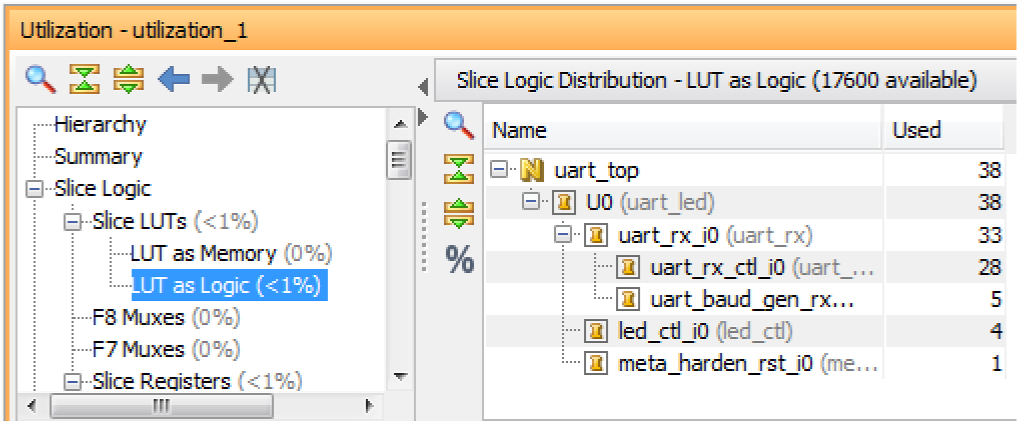
2-2-3. Select Slice LUTs to view how much and which module consumes the resource.



Slice Logic Distribution - LUT as Logic (53200 available)

Name	Used
uart_top	38
U0 (uart_led)	38
uart_rx_i0 (uart_rx)	33
uart_rx_ctl_i0 (uart_...	27
uart_baud_gen_rx...	6
led_ctl_i0 (led_ctl)	4
meta_harden_rst_i0 (me...	1

Figure 10. Resource utilization for the ZedBoard



Slice Logic Distribution - LUT as Logic (17600 available)

Name	Used
uart_top	38
U0 (uart_led)	38
uart_rx_i0 (uart_rx)	33
uart_rx_ctl_i0 (uart_...	28
uart_baud_gen_rx...	5
led_ctl_i0 (led_ctl)	4
meta_harden_rst_i0 (me...	1

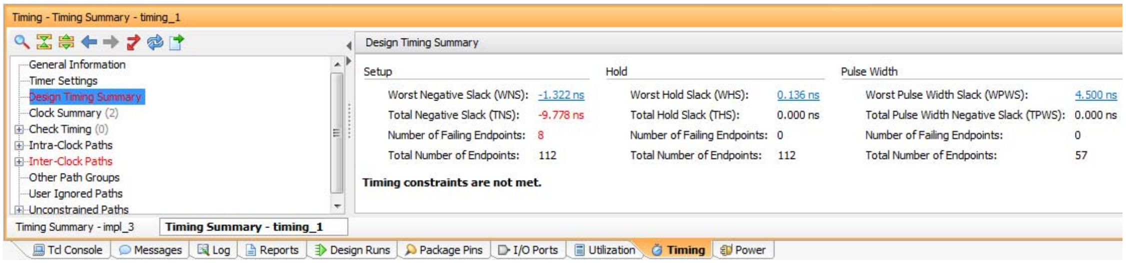
Figure 10. Resource utilization for the Zybo

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.



Timing - Timing Summary - timing_1

- General Information
- Timer Settings
- Design Timing Summary
- Clock Summary (2)
- Check Timing (0)
- Intra-Clock Paths
- Inter-Clock Paths
- Other Path Groups
- User Ignored Paths
- Unconstrained Paths

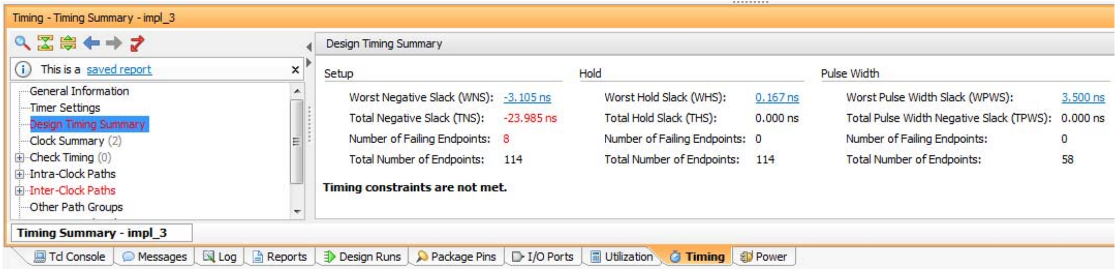
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-1.322 ns	Worst Hold Slack (WHS):	0.136 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	-9.778 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:	57

Timing constraints are not met.

Timing Summary - impl_3 | Timing Summary - timing_1

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



Timing - Timing Summary - impl_3

This is a saved report

- General Information
- Timer Settings
- Design Timing Summary
- Clock Summary (2)
- Check Timing (0)
- Intra-Clock Paths
- Inter-Clock Paths
- Other Path Groups

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-3.105 ns	Worst Hold Slack (WHS):	0.167 ns	Worst Pulse Width Slack (WPWS):	3.500 ns
Total Negative Slack (TNS):	-23.985 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:	114	Total Number of Endpoints:	114	Total Number of Endpoints:	58

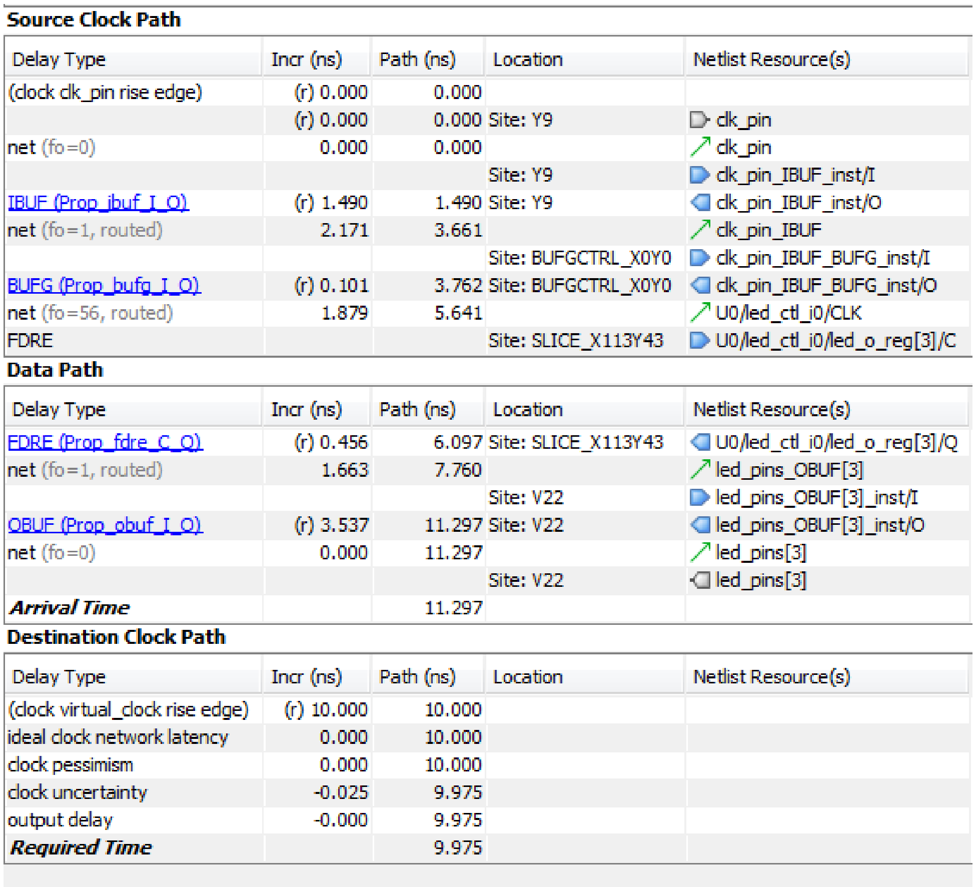
Timing constraints are not met.

Timing Summary - impl_3

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

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.



Source Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 0.000	0.000		
	(r) 0.000	0.000	Site: Y9	clk_pin
net (fo=0)	0.000	0.000		clk_pin
			Site: Y9	clk_pin_IBUF_inst/I
IBUF (Prop_ibuf_I_O)	(r) 1.490	1.490	Site: Y9	clk_pin_IBUF_inst/O
net (fo=1, routed)	2.171	3.661		clk_pin_IBUF
			Site: BUFGCTRL_X0Y0	clk_pin_IBUF_BUFG_inst/I
BUFG (Prop_bufg_I_O)	(r) 0.101	3.762	Site: BUFGCTRL_X0Y0	clk_pin_IBUF_BUFG_inst/O
net (fo=56, routed)	1.879	5.641		U0/led_ctl_i0/CLK
FDRE			Site: SLICE_X113Y43	U0/led_ctl_i0/led_o_reg[3]/C

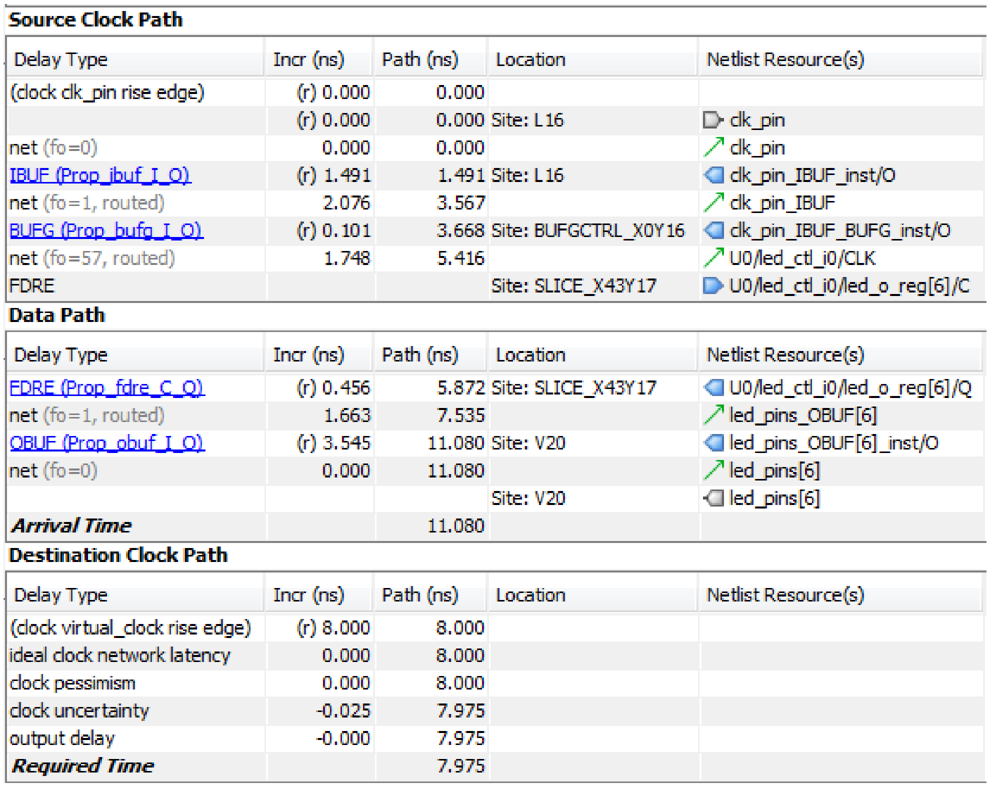
Data Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
FDRE (Prop_fdre_C_Q)	(r) 0.456	6.097	Site: SLICE_X113Y43	U0/led_ctl_i0/led_o_reg[3]/Q
net (fo=1, routed)	1.663	7.760		led_pins_OBUF[3]
			Site: V22	led_pins_OBUF[3]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.537	11.297	Site: V22	led_pins_OBUF[3]_inst/O
net (fo=0)	0.000	11.297		led_pins[3]
			Site: V22	led_pins[3]
Arrival Time		11.297		

Destination Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(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 ZedBoard



Source Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock clk_pin rise edge)	(r) 0.000	0.000		
	(r) 0.000	0.000	Site: L16	clk_pin
net (fo=0)	0.000	0.000		clk_pin
IBUF (Prop_ibuf_I_O)	(r) 1.491	1.491	Site: L16	clk_pin_IBUF_inst/O
net (fo=1, routed)	2.076	3.567		clk_pin_IBUF
BUFG (Prop_bufg_I_O)	(r) 0.101	3.668	Site: BUFGCTRL_X0Y16	clk_pin_IBUF_BUFG_inst/O
net (fo=57, routed)	1.748	5.416		U0/led_ctl_i0/CLK
FDRE			Site: SLICE_X43Y17	U0/led_ctl_i0/led_o_reg[6]/C

Data Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
FDRE (Prop_fdre_C_Q)	(r) 0.456	5.872	Site: SLICE_X43Y17	U0/led_ctl_i0/led_o_reg[6]/Q
net (fo=1, routed)	1.663	7.535		led_pins_OBUF[6]
OBUF (Prop_obuf_I_O)	(r) 3.545	11.080	Site: V20	led_pins_OBUF[6]_inst/O
net (fo=0)	0.000	11.080		led_pins[6]
			Site: V20	led_pins[6]
Arrival Time		11.080		

Destination Clock Path

Delay Type	Incr (ns)	Path (ns)	Location	Netlist Resource(s)
(clock virtual_clock rise edge)	(r) 8.000	8.000		
ideal clock network latency	0.000	8.000		
clock pessimism	0.000	8.000		
clock uncertainty	-0.025	7.975		
output delay	-0.000	7.975		
Required Time		7.975		

Figure 12. First failing path delays for the Zybo

Compared to delays from the synthesis report, the net delays are actual delays (rather than an estimated figure). The data path delay is longer than the destination clock path delay giving a negative slack (violation). The data path delay is 11.297 ns for the ZedBoard, the destination clock path is 9.975 ns and the negative slack is -1.322 ns.

The figures are 11.080 ns, 7.975 ns and -3.105 ns respectively for the Zybo.

At this point we can ignore this violation as the LED display change by a few nanoseconds won't be observable by human eyes. We can also change the output delay by -2 ns and make the timings meet.

Note: Change the output delay by -3.105 ns and add the following min_delay constraint in the uart_led_timing_Zybo.xdc file.
set_output_delay -clock virtual_clock -1.0 -min [get_ports {led_pins[*]}]

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 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.

© copyright 2016 Xilinx

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.**

Click on the **Device** tab and see the highlighted path in the view.

2-3-11. Select **Implemented Design > Report Clock Networks**.

2-3-12. Click **OK**.

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

2-3-13. Select *clk_pin* entry and observe the selected nets in the Device view.

The clock nets are spread across multiple clock regions.

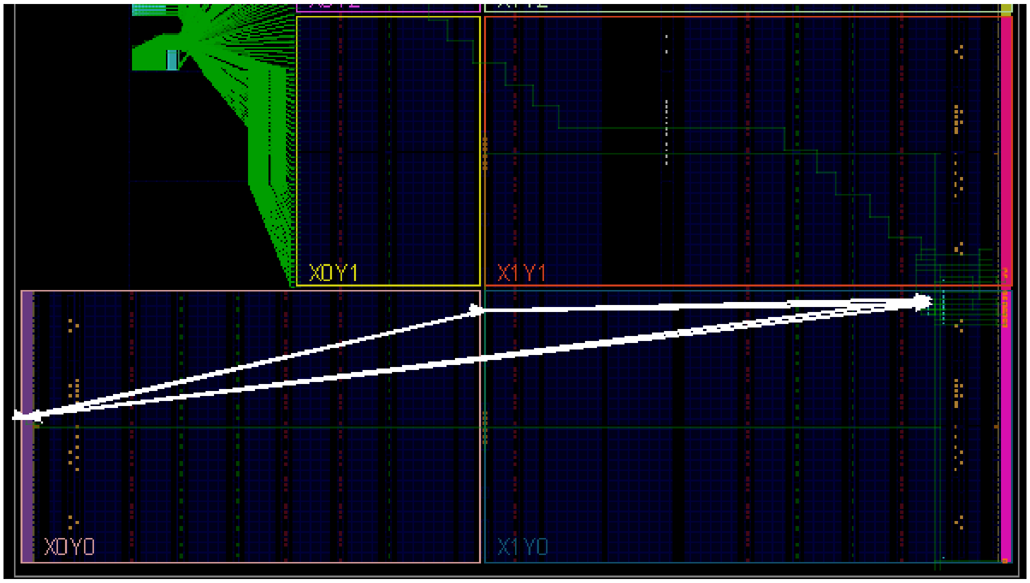


Figure 13. Clock nets for the ZedBoard

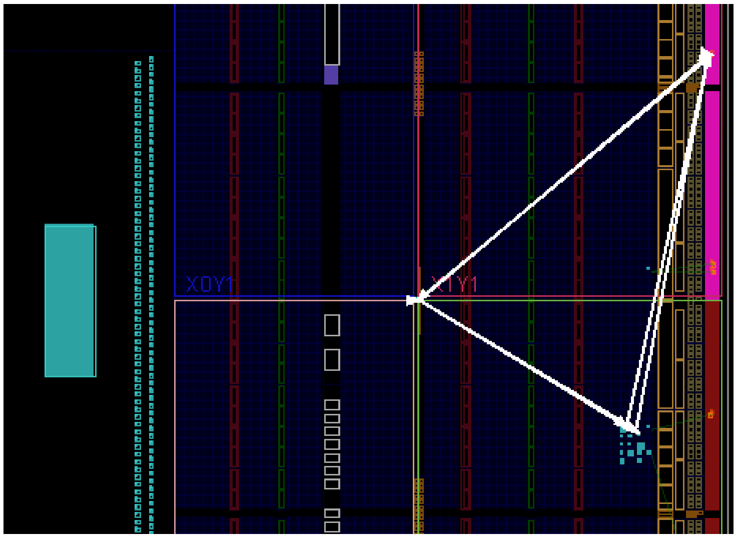


Figure 13. Clock nets for the Zybo

© copyright 2016 Xilinx

Generate the Bitstream Step 3

3-1. Generate the bitstream.

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

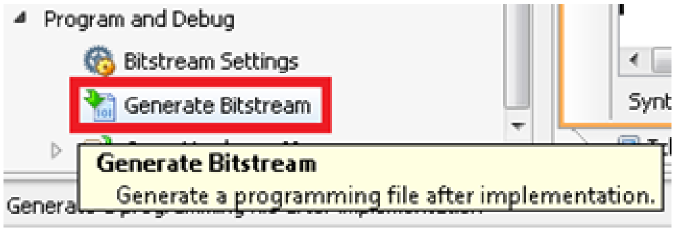


Figure 14, Generating the bitstream

3-1-2. Click **Save** to save the constraints since the timing constraints had been changed, click **OK**, and then **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

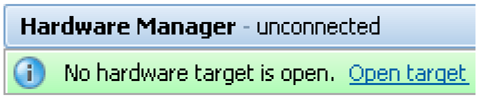
4-1. Connect the board and power it ON. Open a hardware session, and program the FPGA.

4-1-1. Make sure that the Micro-USB cable is connected to the JTAG PROG connector next to the power supply connector for the Zedboard. The Zybo JTAG PROG connector is located next to the power supply switch).

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 target** link, then **Auto Connect** from the dropdown menu.



Hardware Manager - unconnected

No hardware target is open. Open target

Figure 15. Opening new hardware target

4-1-4. 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.

4-1-5. Select the device in the *Hardware Device Properties,* and verify that the **uart_top.bit** is selected as the programming file in the General tab.

© copyright 2016 Xilinx

4-1-6. Right-click on the FPGA entry in the Hardware window and select **Program Device…**

4-1-7. 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. Start a terminal emulator program such as TeraTerm or HyperTerminal. Select an appropriate COM port (you can find the correct COM number using the Control Panel). Set the COM port for 115200 baud rate communication.

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

4-2-2. Select an appropriate COM port (you can find the correct COM number using the Control Panel).

4-2-3. Set the COM port for 115200 baud rate communication.

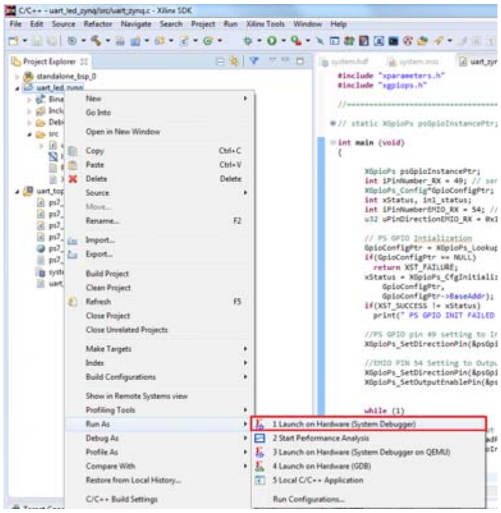
4-3. Program FPGA, Start a SDK session, point it to the c:/xup/fpga_flow/2016_2_ZYNQ_ *Sources* /lab3/<board>/lab3.sdk workspace.

4-3-1. Open **SDK** by selecting **Start > All Programs > Xilinx Design Tools > SDK 2016.2 > Xilinx SDK 2016.2**

4-3-2. In the **Select a workspace** window, click on the browse button, browse to c:/xup/fpga_flow/2016_2_ZYNQ_ *Sources* /lab3/ directory and select either c:/xup/fpga_flow/2016_2_ZYNQ_ *Sources* /lab3/Zybo/lab3.sdk or c:/xup/fpga_flow/2016_2_ZYNQ_ *Sources* /lab3/ ZedBoard/lab3.sdk and click **OK.**

4-3-3. Click **OK**.

4-3-4. In the *Project Explorer*, right-click on the *uart_led_zynq*, select *Run As*, and then **Launch on Hardware (System Debugger).**



```
#include "xparameters.h"
#include "xgpiops.h"

//*************************************

// static XGpioPs psGpioInstancePtr;

int main (void)
{

        XGpioPs psGpioInstancePtr;
        int iPinNumber_RX = 49; // ser
        XGpioPs_Config*GpioConfigPtr;
        int xStatus, in1_status;
        int iPinNumberEMIO_RX = 54; //
        u32 uPinDirectionEMIO_RX = 0x1

        // PS GPIO Intialization
        GpioConfigPtr = XGpioPs_Lookup
        if(GpioConfigPtr == NULL)
          return XST_FAILURE;
        xStatus = XGpioPs_CfgInitiali
            GpioConfigPtr,
            GpioConfigPtr->BaseAddr);
        if(XST_SUCCESS != xStatus)
          print(" PS GPIO INIT FAILED

        //PS GPIO pin 49 setting to Ir
        XGpioPs_SetDirectionPin(&psGpi

        //EMIO PIN 54 Setting to Outpu
        XGpioPs_SetDirectionPin(&psGpi
        XGpioPs_SetOutputEnablePin(&ps


        while (1)
```

Figure 16. Running the application

© copyright 2016 Xilinx

T■ program will be downloaded and the execution will begin indicated by the Terminate button ( ) in SDK.

4-3-5. In the serial terminal window, type in some characters in the terminal emulator window and see the corresponding ASCII equivalent bit pattern displayed on the LEDs.

Note: For Zybo you will see the lower four bits on the LEDs

4-3-6. Press and hold BTNU and see the the upper four bits are swapped with the lower four bits on the LEDs

Note: For Zybo press and hold BTN0 to see the upper four bits on the LEDs

4-3-7. When satisfied, close the terminal emulator program and power OFF the board.

4-3-8. Select **File > Close Hardware Manager**. Click **OK**.

4-3-9. Close the **Vivado** program by selecting **File > Exit** and click **OK**.

4-3-10. Close the **SDK** program by selecting **File > Exit** and click **OK**.

Conclusion

In this lab, 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.

© copyright 2016 Xilinx