Synthesizing a RTL Design

Introduction

This lab shows you the synthesis process and effect of changing of the synthesis settings. You will analyze the design and the generated reports.

Objectives

After completing this lab, you will be able to:

- Use the provided Xilinx Design Constraint (XDC) file to constrain the timing of the circuit
- Elaborate the design and understand the output
- Synthesize the design with the provided basic timing constraints
- Analyze the output of the synthesized design
- Change the synthesis settings and see their effects on the generated output
- Write a checkpoint after the synthesis so the results can be analyzed after re-loading it

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 lab2.

Design Description

The design consists of a UART receiver receiving the input typed on a keyboard and displaying the binary equivalent of the typed character on the LEDs. Depending on the target board, when a push button is pressed, the lower and upper nibbles are displayed on the lower-half and/or upper-half of the LED array. The block diagram is as shown in **Figure 1**.

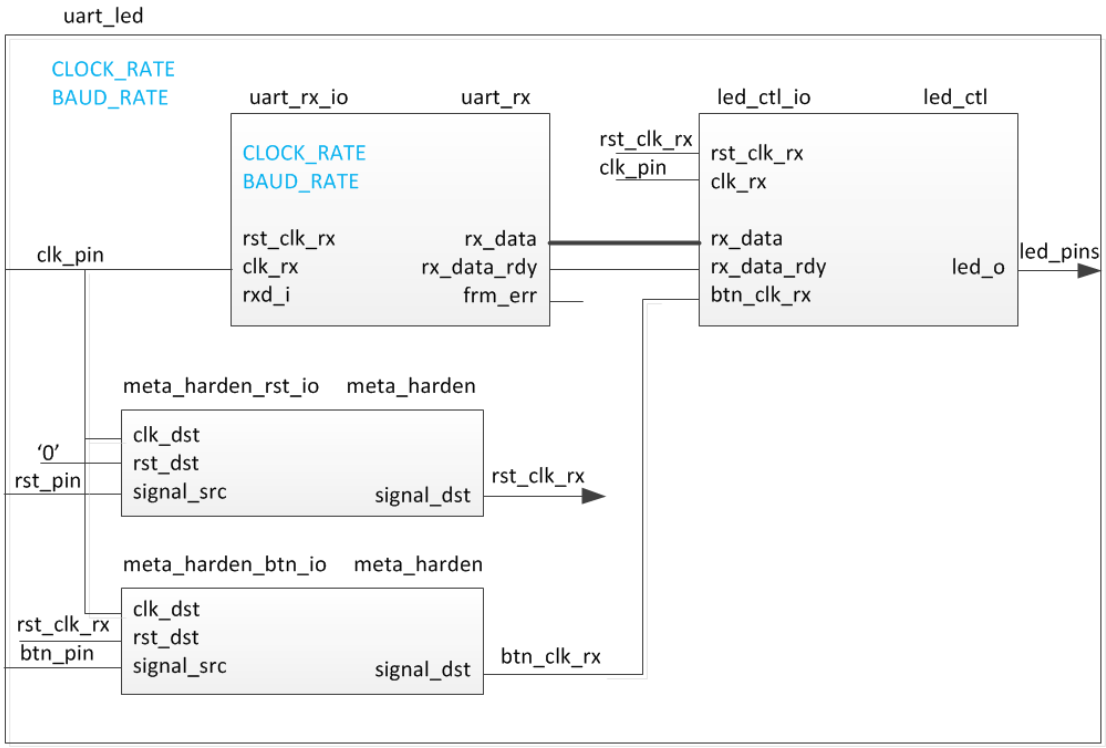
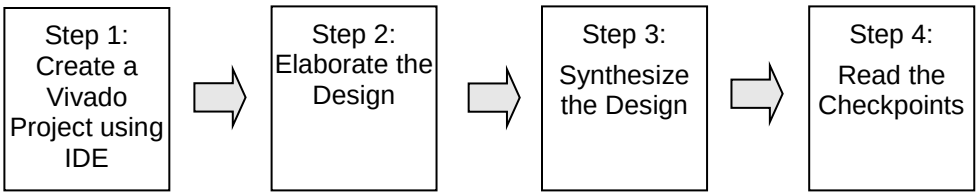


Figure 1. The Completed Design

General Flow



Create a Vivado Project using IDE — Step 1

1-1. Launch Vivado and create a project targeting the XC7Z020CLG484-1 or XC7Z010CLG400-1 device and using the Verilog HDL. Use the provided Verilog source files and uart_led_timing.xdc file from the *<2014_2_zynq_sources>\<board>\lab2* directory.

> References to **<2014_2_zynq_labs>** is a placeholder for the **c:\xup\fpga_flow\2014_2_zynq_labs** directory and **<2014_2_zynq_sources>** is a placeholder for the **c:\xup\fpga_flow\2014_2_zynq_sources** directory.
>
> Reference to **<board>** means either the **ZedBoard** or the **Zybo**.

1-1-1. Open Vivado by selecting **Start > All Programs > Xilinx Design Tools > Vivado 2014.2 > Vivado 2014.2**

1-1-2. Click **Create New Project** to start the wizard. You will see *Create A New Vivado Project* dialog box. Click **Next**.

1-1-3. Click the Browse button of the *Project location* field of the **New Project** form, browse to **<2014_2_zynq_labs>**, and click **Select**.

1-1-4. Enter **lab2** in the *Project name* field. Make sure that the *Create Project Subdirectory* box is checked. Click **Next**.

1-1-5. Select **RTL Project** option in the *Project Type* form, and click **Next**.

1-1-6. Select **Verilog** as the *Target Language* in the *Add Sources* form.

1-1-7. Click on the **Add Files…** button, browse to the **<2014_2_zynq_sources>\<board>\lab2** directory, select all the Verilog files *(led_ctl.v, meta_harden, uart_baud_gen, uart_led, uart_rx, and uart_rx_ctl),* click **OK**. Make sure the source files are copied into the project via the check box and click **Next**.

1-1-8. Click **Next** again to get to the *Add Cons*traints form.

1-1-9. You will see uart_led_timing.xdc and uart_led_pins.xdc have automatically been included. Remove the uart_led_pins.xdc, for now, by selecting the entry and clicking on the X on the right

hand side of the window. (If you don't see uart_led_timing.xdc entry then browse to the **<2014_2_zynq_sources>\<board>\lab2** directory, select *uart_led_timing.xdc* and click **Open**.

1-1-10. Click **Next.**

This Xilinx Design Constraints file assigns the basic timing constraints (period, input delay, and output delay) to the design.

1-1-11. In the *Default Part* form, using the **Parts** option and various drop-down fields of the **Filter** section , select the **XC7Z020-1CLG484C** device for the ZedBoard or the **XC7Z010-1CLG400C** device for the Zybo. Click **Next**.

1-1-12. Click **Finish** to create the Vivado project. If you have Lab1 previously open, click the option to close it when prompted.

1-2. Analyze the design source files hierarchy.

1-2-1. In the *Sources* pane, expand the **uart_led** entry and notice hierarchy of the lower-level modules.

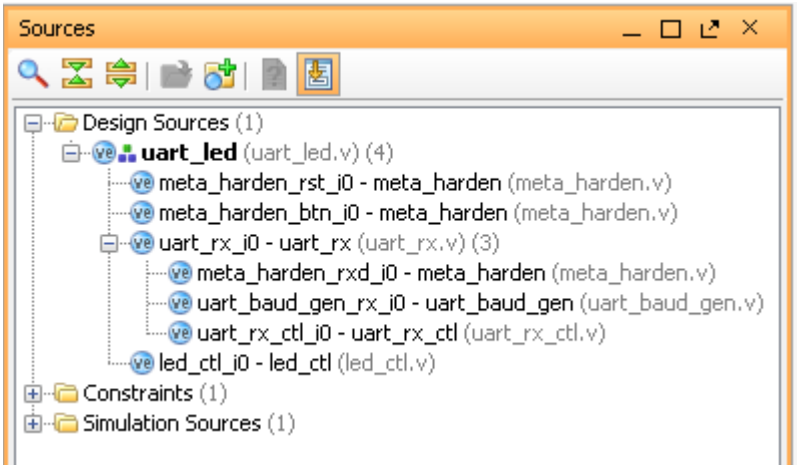


Figure 2. Opening the source file

1-2-2. Double-click on the **uart_led** entry to view its content.

Notice in the Verilog code, the BAUD_RATE and CLOCK_RATE parameters are defined to be 115200 and 100 MHz respectively as shown in the design diagram (Figure 1). The clock rate of 100 MHz is defined for the ZedBoard, the Zybo will require the rate to be changed to 125 MHz.

Also notice that the lower level modules are instantiated. The meta_harden modules are used to synchronize the asynchronous reset and push-button inputs.

1-2-3. Expand **uart_rx_i0** instance to see its hierarchy.

This instance uses a baud rate generator and a finite state machine. The rxd_pin is sampled at x16 the baud rate.

1-3. Open the uart_led_timing.xdc source and analyze the content.

1-3-1. In the *Sources* pane, expand the *Constraints* folder and double-click the **uart_led_timing.xdc** entry to open the file in text mode.

```
1 # ZedBoard xdc
2 # define clock and period
3 create_clock -period 10.000 -name clk_pin -waveform {0.000 5.000} [get_ports clk_pin]
4
5 # Create a virual clock for IO constraints
6 create_clock -period 12.0 -name virtual_clock
7
8 # input delay
9 set_input_delay -clock clk_pin 0 [get_ports rxd_pin]
10 set_input_delay -clock clk_pin -min -0.5 [get_ports rxd_pin]
11
12 set_input_delay -clock virtual_clock -max 0.0 [get_ports btn_pin]
13 set_input_delay -clock virtual_clock -min -0.5 [get_ports btn_pin]
14
15 #output delay
16 set_output_delay -clock virtual_clock 0 [get_ports led_pins[*]]
17
18
```

Figure 3. Timing constraints for the ZedBoard

```
1 # ZYBO xdc
2 # define clock and period
3 create_clock -period 8.000 -name clk_pin -waveform {0.000 4.000} [get_ports clk_pin]
4
5 # Create a virual clock for IO constraints
6 create_clock -period 9.0 -name virtual_clock
7
8 # input delay
9 set_input_delay -clock clk_pin 0 [get_ports rxd_pin]
10 set_input_delay -clock clk_pin -min -0.5 [get_ports rxd_pin]
11
12 set_input_delay -clock virtual_clock -max 0.0 [get_ports btn_pin]
13 set_input_delay -clock virtual_clock -min -0.5 [get_ports btn_pin]
14
15 #output delay
16 set_output_delay -clock virtual_clock 0 [get_ports led_pins[*]]
17
```

Figure 3. Timing constraints for the Zybo

Line 3 creates the period constraint of 10 ns for the ZedBoard or 8 ns for the Zybo with a duty cycle of 50%. Line 6 creates a virtual clock of 12 ns for the ZedBoard or 9 ns for the Zybo. This clock can be viewed as the upstream device is generating its output with respect to its clock and

outputs data with respect to it. The rxd_pin is constrained with respect to the design clock (lines 9, and 10) whereas the btn_pin is constrained with respect to the upstream clock (lines 12, 13). The led_pins are constrained with respect to the upstream clock as the downstream device may be using it.

Elaborate the Design — Step 2

2-1. Elaborate and perform the RTL analysis on the source file.

2-1-1. Expand the *Open Elaborated Design* entry under the *RTL Analysis* tasks of the *Flow Navigator* pane and click on **Schematic**.

The model (design) will be elaborated and a logic view of the design is displayed.

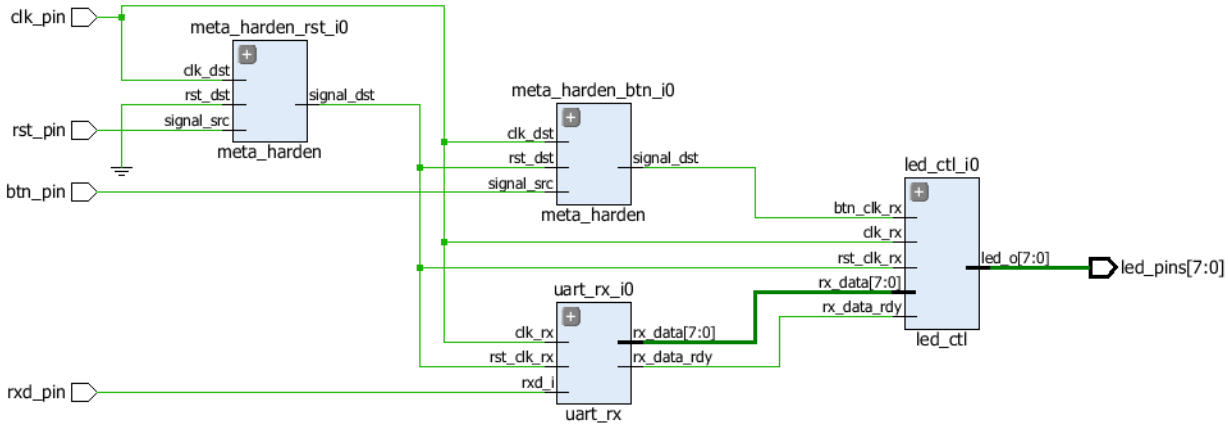


Figure 4. A logic view of the design for the ZedBoard

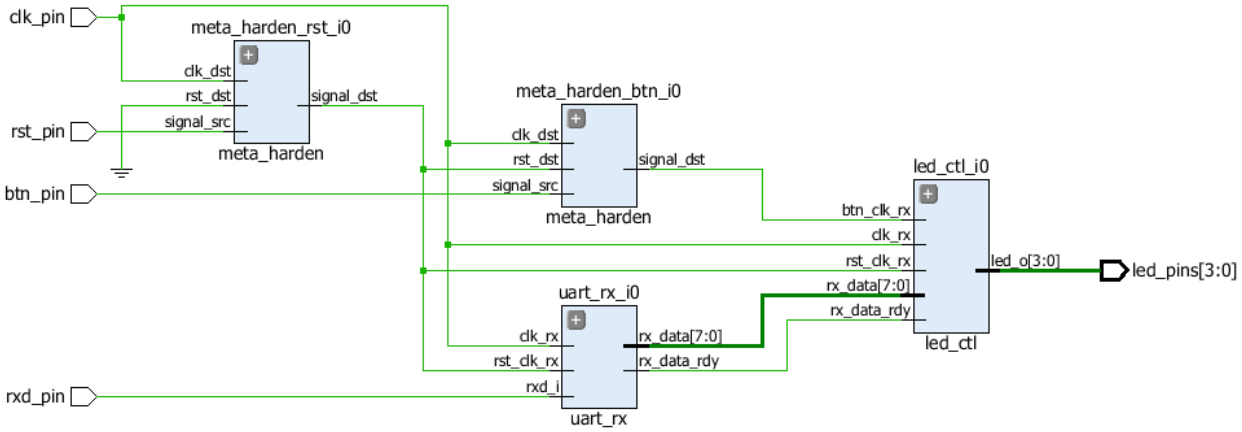


Figure 4. A logic view of the design for the Zybo

You will see five components at the top-level, 2 instances of meta_harden, one instance of uart_rx, one instance of led_ctl.

2-1-2. Return to the schematic. Click on the "+" symbol on the uart_rx_i0 instance in the schematic diagram to see the underlying components.

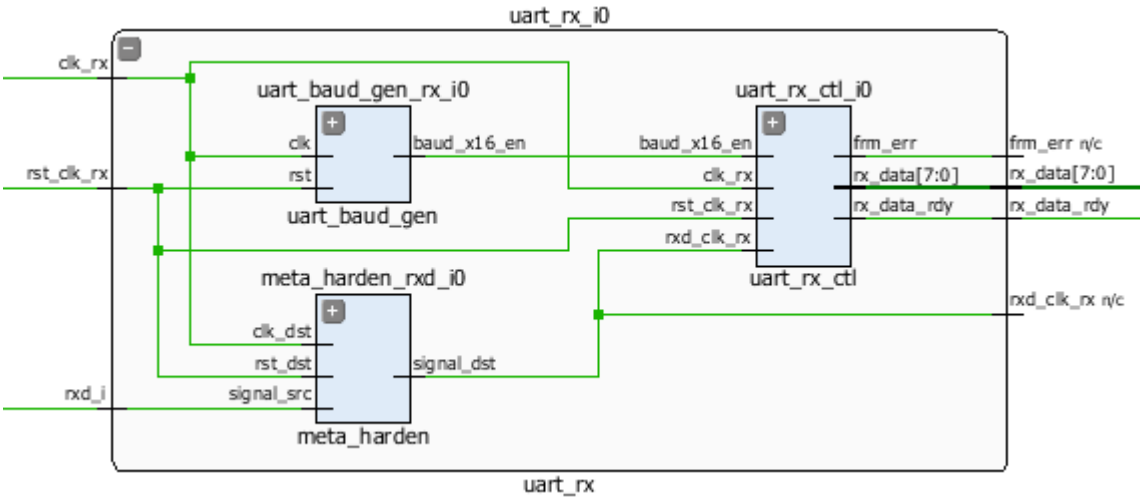


Figure 5. Lower level components of the uart_rx_i0 module

2-1-3. Click on **Report Noise** under the *Open Elaborated Design* entry of the *RTL Analysis* tasks of the *Flow Navigator* pane.

2-1-4. Click **OK** to generate the report named **ssn_1**. Do not export the output to a file.

2-1-5. View the ssn_1 report (in the bottom pane) and observe the unplaced ports, Summary, and I/O Bank Details are highlighted in red because the pin assignments were not done. Note that only output pins are reported as the noise analysis is done on the output pins.

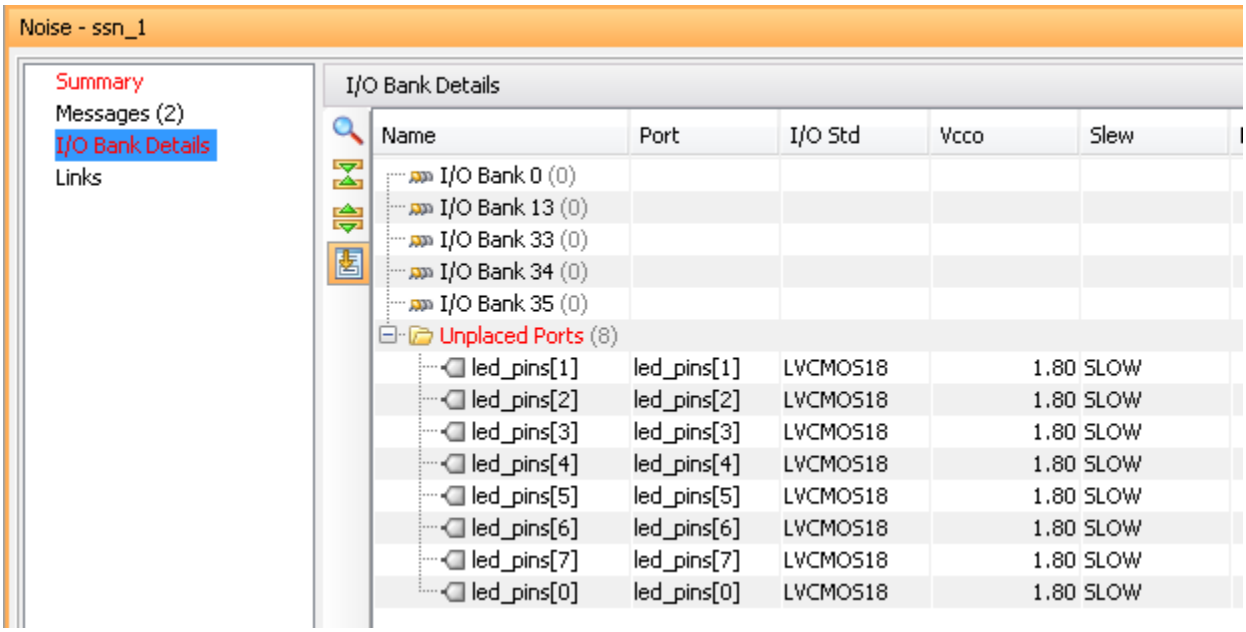


Figure 6. Noise report for the ZedBoard

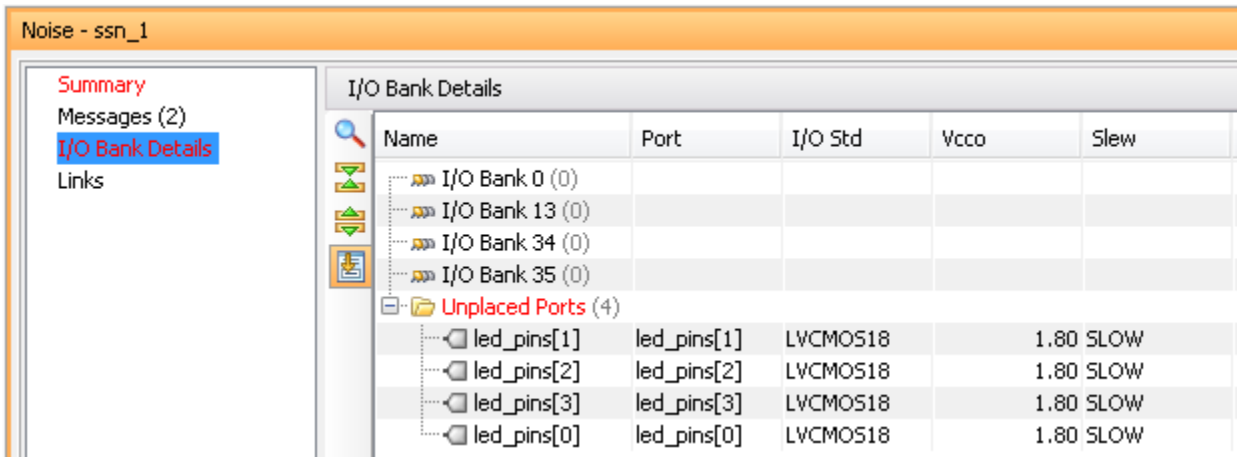


Figure 6. Noise report for the Zybo

2-1-6. Click on **Add Sources** under the *Project Navigator*, select *Add or Create Constraints* option and click **Next**.

2-1-7. Click on the **Add Files…** button, browse to the **<2014_2_zynq_sources>\<board>\lab2** directory, select the **uart_led_pins.xdc** file, click **OK.** Make sure the constraints file is copied into the project and then click **Finish** to add the pins location constraints.

Notice that the sources are modified and the tools detect it, showing a notification status bar to re-load the design.

2-1-8. Click on the **Reload** link. The constraints will be processed.

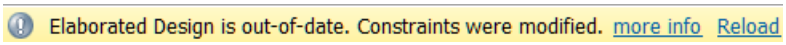


Figure 7. Reloading the Elaborated Design

2-1-9. Click on **Report Noise** and click **OK** to generate the report named **ssn_1**. Observe that this time it does not show any errors (no red).

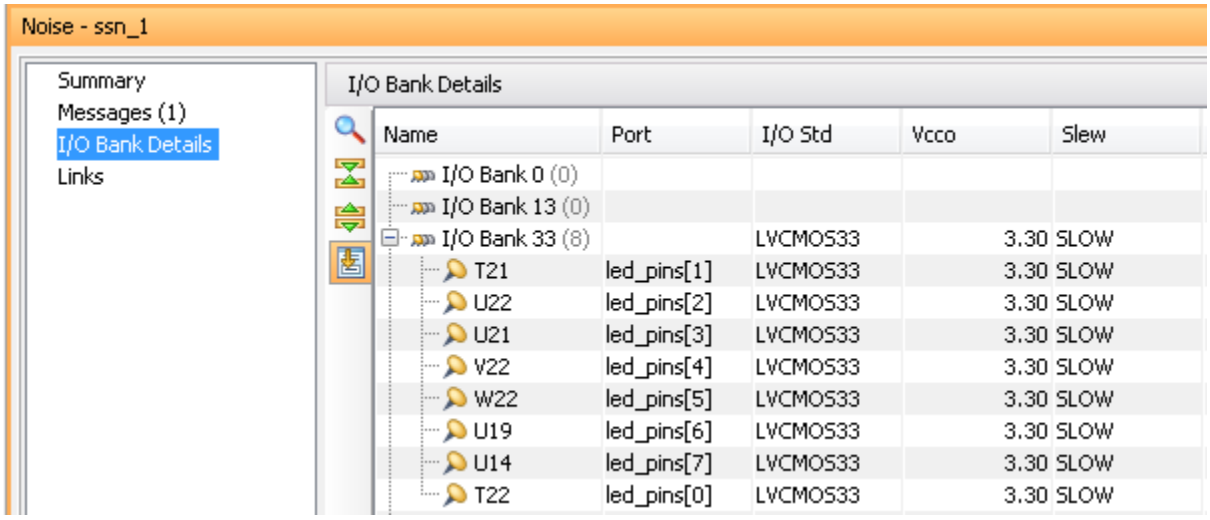


Figure 8. Noise analysis report after the locations constraints processed for the ZedBoard

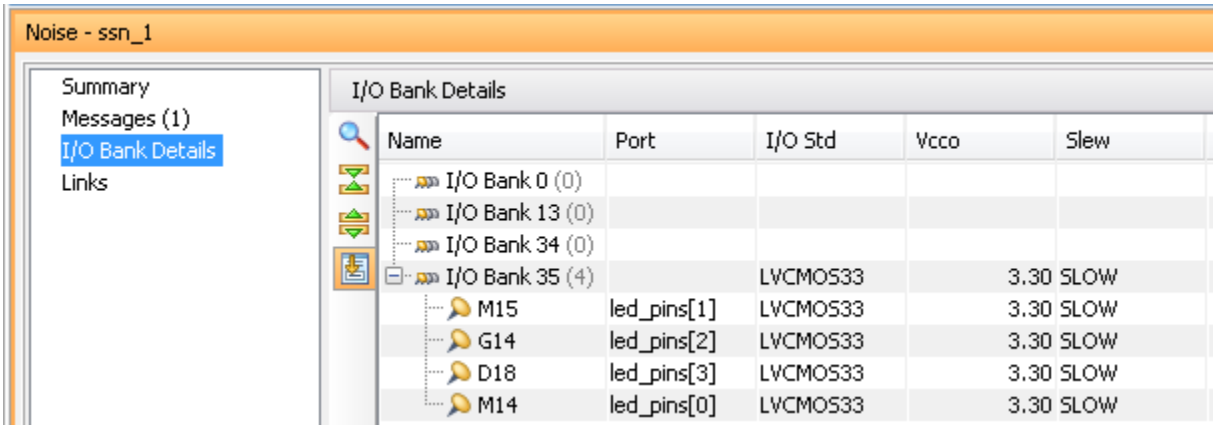


Figure 8. Noise analysis report after the locations constraints processed for the Zybo

Synthesize the Design — Step 3

3-1. Synthesize the design with the Vivado synthesis tool and analyze the Project Summary output.

3-1-1. 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.

3-1-2. Select the *Open Synthesized Design* option and click **OK** as we want to look at the synthesis output.

Click **Yes** to close the elaborated design if the dialog box is displayed.

3-1-3. Select the **Project Summary** tab

If you don't see the Project Summary tab then select **Layout > Default Layout, or** click the **Project Summary** icon  .

3-1-4. Click on the **Table** tab under the **Utilization** pane in the **Project Summary** tab and fill out the following information.

Question 1

Look through the table and find the number used of each of the following:

FF: _____
LUT: _____
I/O: _____
BUFG: _____

3-1-5. Click on **Schematic** under the *Synthesized Design* tasks of *Synthesis* tasks of the *Flow Navigator* pane to view the synthesized design in a schematic view.

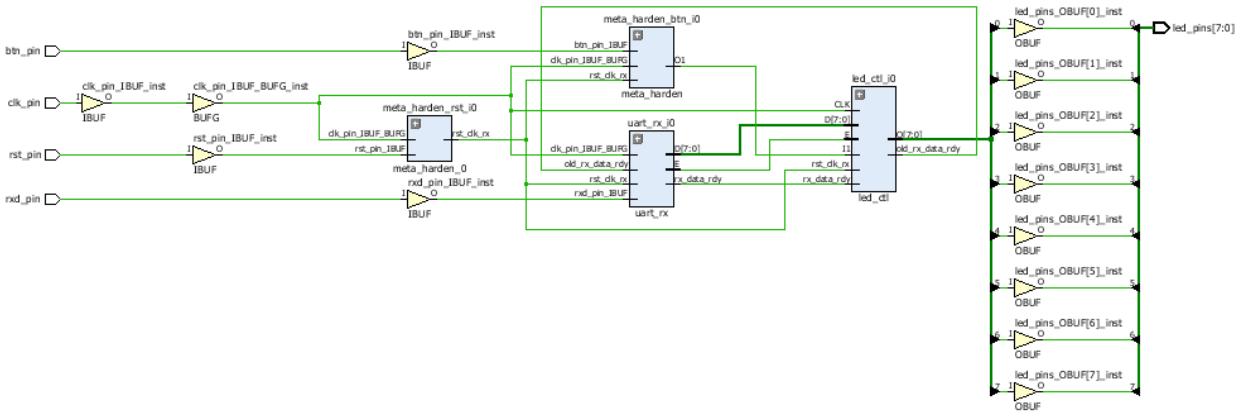


Figure 9. Synthesized design's schematic view for the ZedBoard

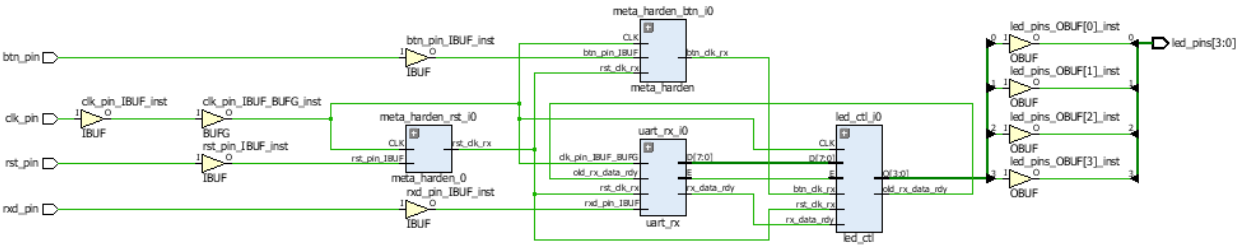


Figure 9. Synthesized design's schematic view for the Zybo

Notice that IBUF and OBUF are automatically instantiated (added) to the design as the input and output are buffered. There are still four lower level modules instantiated.

3-1-6. Expand by clicking the + symbol of the **uart_rx_i0** instance in the schematic view to see the underlying instances.

3-1-7. Select the **uart_baud_gen_rx_i0** instance, right-click, and select *Go To Source*.

Notice that line 86 is highlighted. Also notice that the CLOCK_RATE and BAUD_RATE parameters are passed to the module being called.

3-1-8. Go back to the schematic and double-click on the **meta_harden_rxd_io** instance to see how the synchronization circuit is being implemented using two FFs. This synchronization is necessary to reduce the likelihood of meta-stability.

3-1-9. Click on the ( ) in the schematic view to go back to its parent block.

3-2. Analyze the timing report.

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

3-2-2. Click **OK** to generate the Timing_1 report.

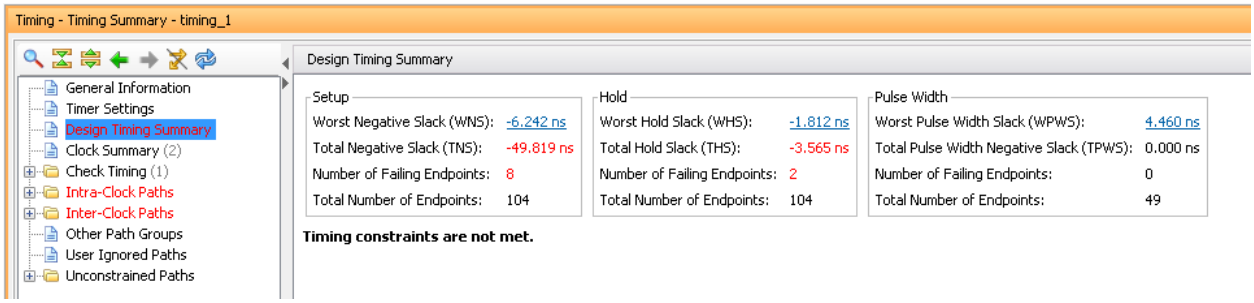


Figure 10. Timing report for the ZedBoard

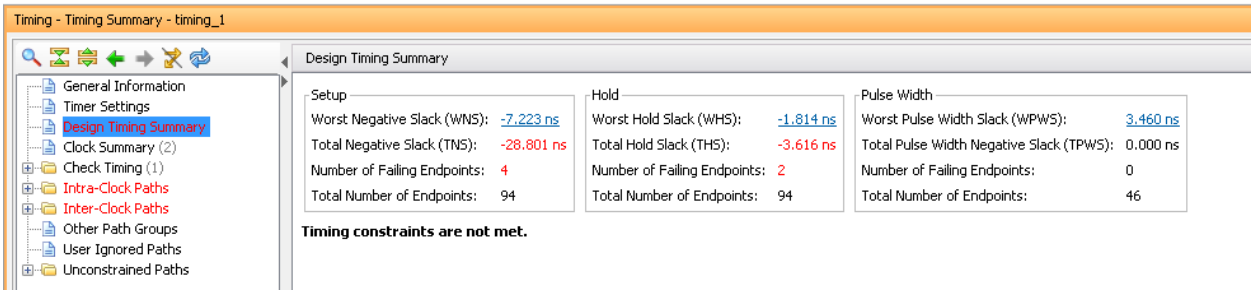


Figure 10. Timing report for the Zybo

Notice that the Design Timing Summary and Inter-Clock Paths entry in the left pane is highlighted in red indicating timing violations. In the right pane, the information is grouped in Setup, Hold, and Width columns.

Under the Setup column Worst Negative Slack (WNS) is linked indicating that clicking on it can give us insight on how the failing path has formed. The Total Negative Slack (TNS) is highlighted in red indicating the total amount of violations in the design and the Number of Failing Endpoints indicates total number of failing paths.

3-2-3. Click on the WNS link and see the 8 failing paths for the ZedBoard and 4 failing paths for the Zybo.

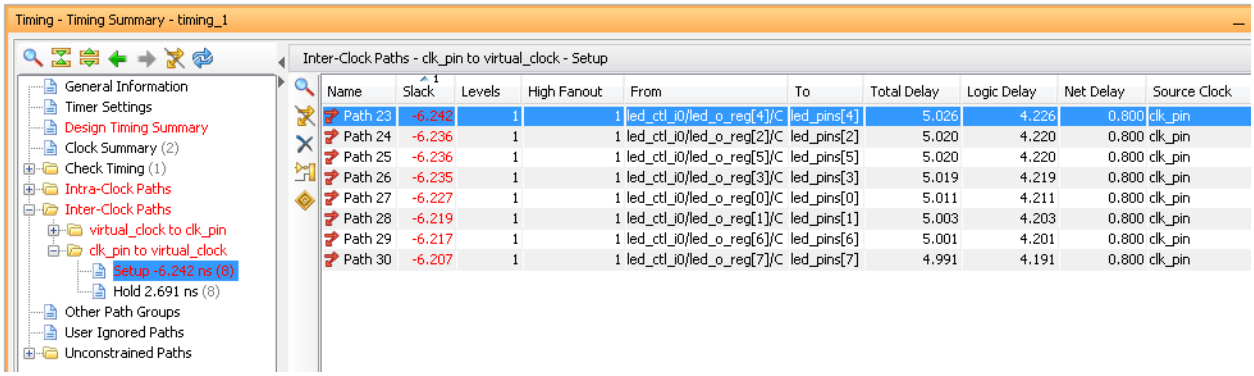


Figure 11. The failing paths for the ZedBoard

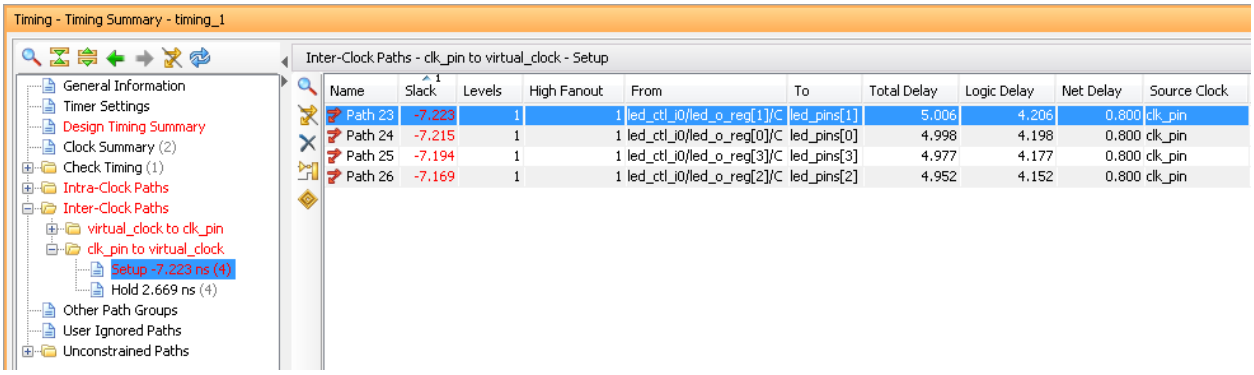


Figure 11. The failing paths for the Zybo

3-2-4. Double-click on the **Path 23** to see how the path is made.

Summary

Name	Path 23
Slack	-6.242ns
Source	led_ctl_i0/led_o_reg[4]/C (rising edge-triggered cell FDRE clocked by clk_pin {rise@0.000ns fall@5.000ns period=10.000ns})
Destination	led_pins[4] (output port clocked by virtual_clock {rise@0.000ns fall@6.000ns period=12.000ns})
Path Group	virtual_clock
Path Type	Max at Slow Process Corner
Requirement	2.000ns (virtual_clock rise@12.000ns - clk_pin rise@10.000ns)
Data Path Delay	5.026ns (logic 4.226ns (84.085%) route 0.800ns (15.915%))
Logic Levels	1 (OBUF=1)
Output Delay	0.000ns
Clock Path Skew	-3.191ns
Clock Uncertainty	0.025ns
Clock Domain 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	Delay	Cumulative	Location	Logical Resource
(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=48, unplaced)	0.800	13.191		led_ctl_i0/CLK
				led_ctl_i0/led_o_reg[4]/C

Data Path

Delay Type	Delay	Cumulative	Location	Logical Resource
FDRE (Prop_fdre_C_Q)	(r) 0.518	13.709		led_ctl_i0/led_o_reg[4]/Q
net (fo=1, unplaced)	0.800	14.509		led_pins_OBUF[4]
			Site: V22	led_pins_OBUF[4]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.708	18.217	Site: V22	led_pins_OBUF[4]_inst/O
net (fo=0)	0.000	18.217		led_pins[4]
			Site: V22	led_pins[4]
Arrival Time		18.217		

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 12. Worst failing path for the ZedBoard

Summary

Name	Path 23
Slack	-7.223ns
Source	led_ctl_i0/led_o_reg[1]/C (rising edge-triggered cell FDRE clocked by clk_pin {rise@0.000ns fall@4.000ns period=8.000ns})
Destination	led_pins[1] (output port clocked by virtual_clock {rise@0.000ns fall@4.500ns period=9.000ns})
Path Group	virtual_clock
Path Type	Max at Slow Process Corner
Requirement	1.000ns (virtual_clock rise@9.000ns - clk_pin rise@8.000ns)
Data Path Delay	5.006ns (logic 4.206ns (84.022%) route 0.800ns (15.978%))
Logic Levels	1 (OBUF=1)
Output Delay	0.000ns
Clock Path Skew	-3.192ns
Clock Uncertainty	0.025ns
Clock Domain 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	Delay	Cumulative	Location	Logical Resource
(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=45, unplaced)	0.800	11.192		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	11.688		led_ctl_i0/led_o_reg[1]/Q
net (fo=1, unplaced)	0.800	12.488		led_pins_OBUF[1]
			Site: M15	led_pins_OBUF[1]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.710	16.198	Site: M15	led_pins_OBUF[1]_inst/O
net (fo=0)	0.000	16.198		led_pins[1]
			Site: M15	led_pins[1]
Arrival Time		16.198		

Destination Clock Path

Delay Type	Delay	Cumulative	Location	Logical Resource
(clock virtual_clock rise edge)	(r) 9.000	9.000		
ideal clock network latency	0.000	9.000		
clock pessimism	0.000	9.000		
clock uncertainty	-0.025	8.975		
output delay	-0.000	8.975		
Required Time		8.975		

Figure 12. Worst failing path for the Zybo

Note that this is an estimate only. The nets are specified as unplaced and have all been allocated default values (0.8 ns). No actual routing delays are considered.

3-3. Generate the utilization and power reports.

3-3-1. Click **Report Utilization** under the Synthesized Design, and click **OK** to generate the utilization report (in the bottom of the GUI).

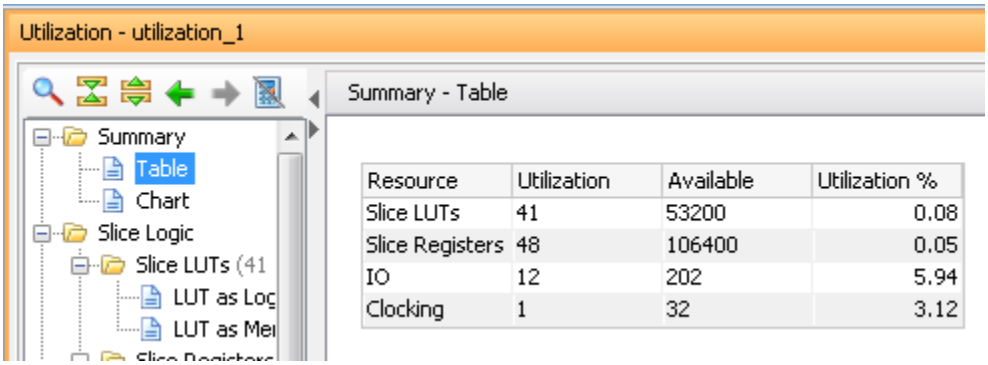


Figure 13. Utilization report for the ZedBoard

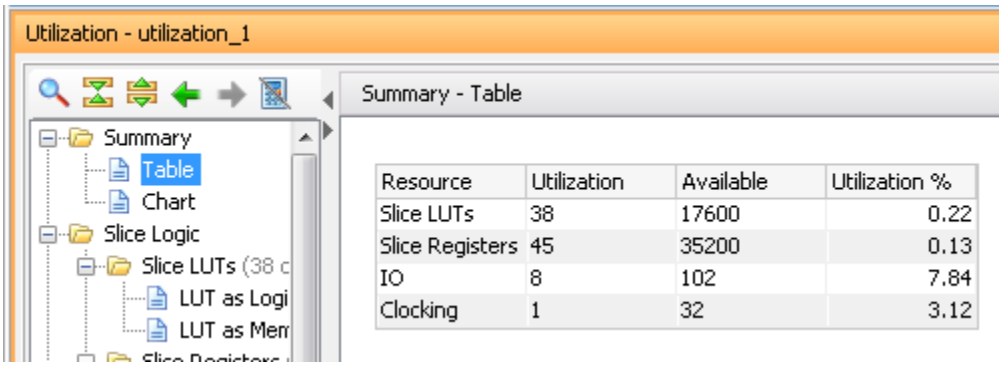


Figure 13. Utilization report for the Zybo

Question 2

Look through the report and find the number used of each of the following:

FF: _____
LUT: _____
I/O: _____
BUFG: _____

3-3-2. Select Slice LUTs entry in the left pane and see the utilization by lower-level instances. You can expand the instances in the right pane to see the complete hierarchy utilization.

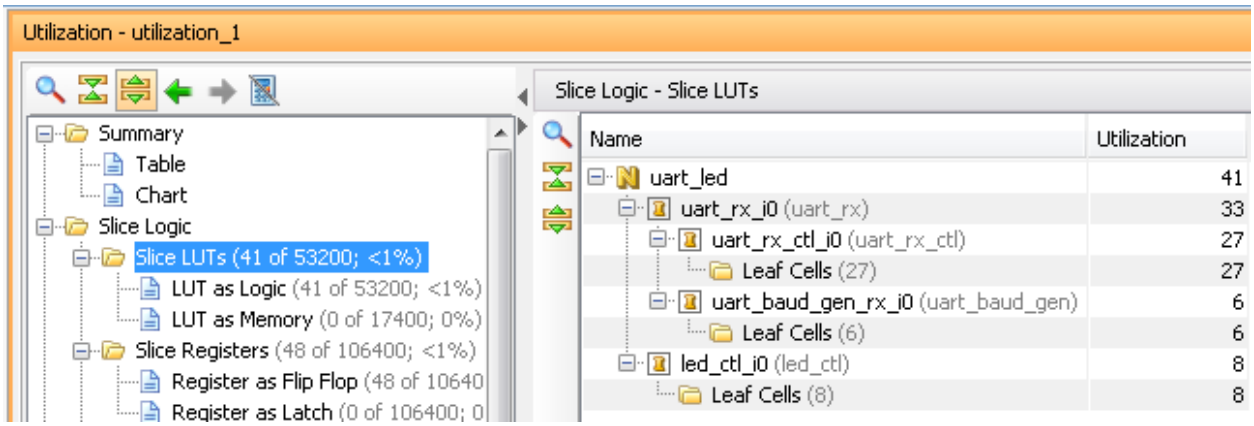


Figure 14. Utilization of lower-level modules for the ZedBoard

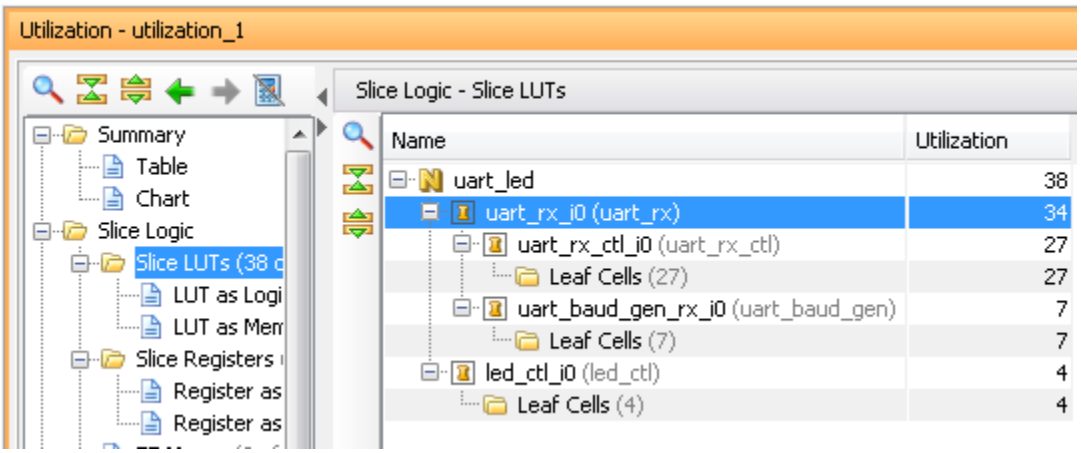


Figure 14. Utilization of lower-level modules for the Zybo

3-3-3. Click **Report Power** under the Synthesized Design, and click **OK** to generate the estimated power consumption report using default values.

Note that this is just an estimate as no simulation run data was provided and no accurate activity rates, or environment information was entered.

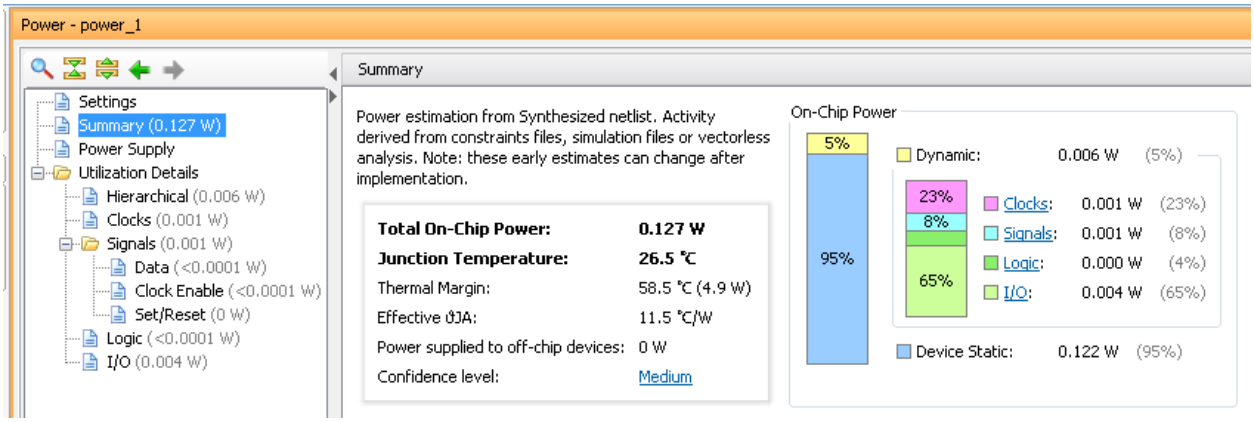


Figure 15. Power consumption estimation for the ZedBoard

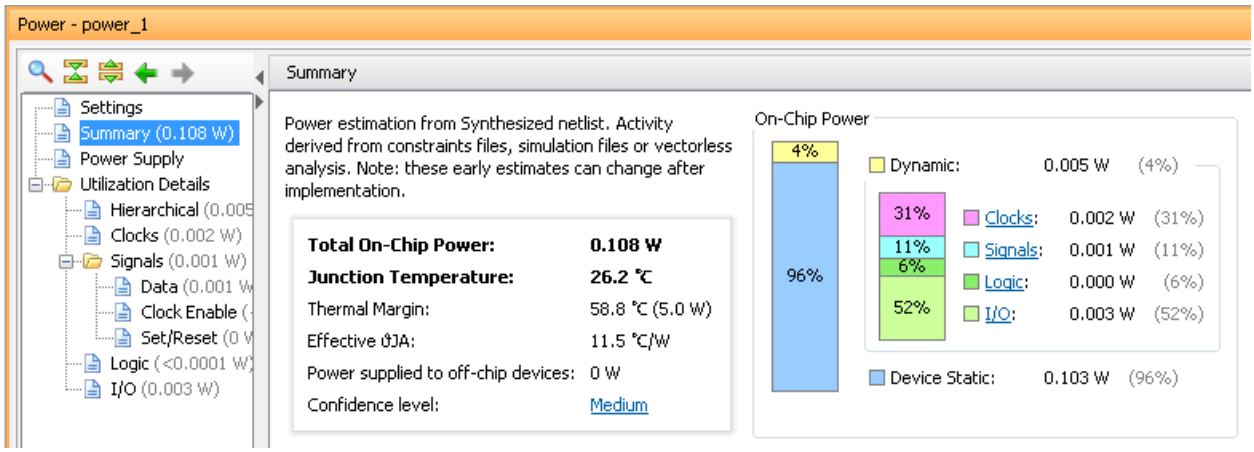


Figure 15. Power consumption estimation for the Zybo

Question 3

From the power report, find the % power consumption used by each of the following:

Clocks:	%
Signals:	%
Logic:	%
I/O:	%

You can move the mouse on the boxes which do not show the percentage to see the consumption.

3-4. Write the checkpoint in order to analyze the results without going through the actual synthesis process.

3-4-1. Select **File > Write Checkpoint…** to save the processed design so it can be opened later for further analysis.

3-4-2. A dialog box will appear showing the default name of the file in the current project directory.

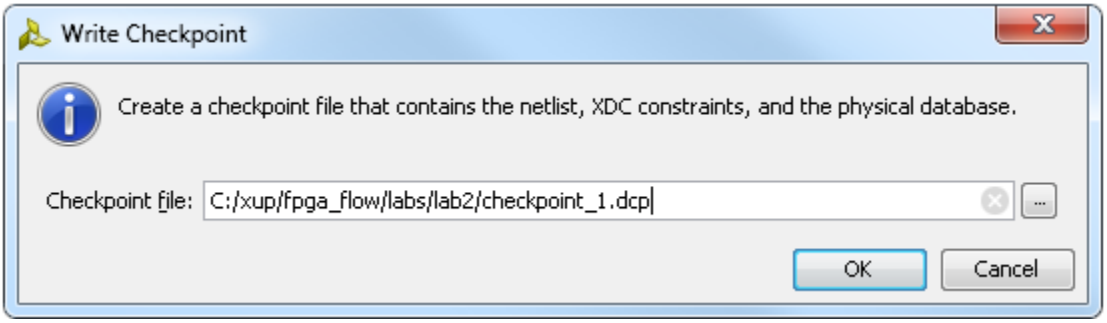


Figure 16. Writing checkpoint

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

3-5. Change the synthesis settings to flatten the design. Re-synthesize the design and analyze the results.

3-5-1. Click on the **Project Settings** under the *Project Manager*, and select **Synthesis**.

3-5-2. Click on the **flatten_hierarchy** drop-down button and select **full** to flatten the design.

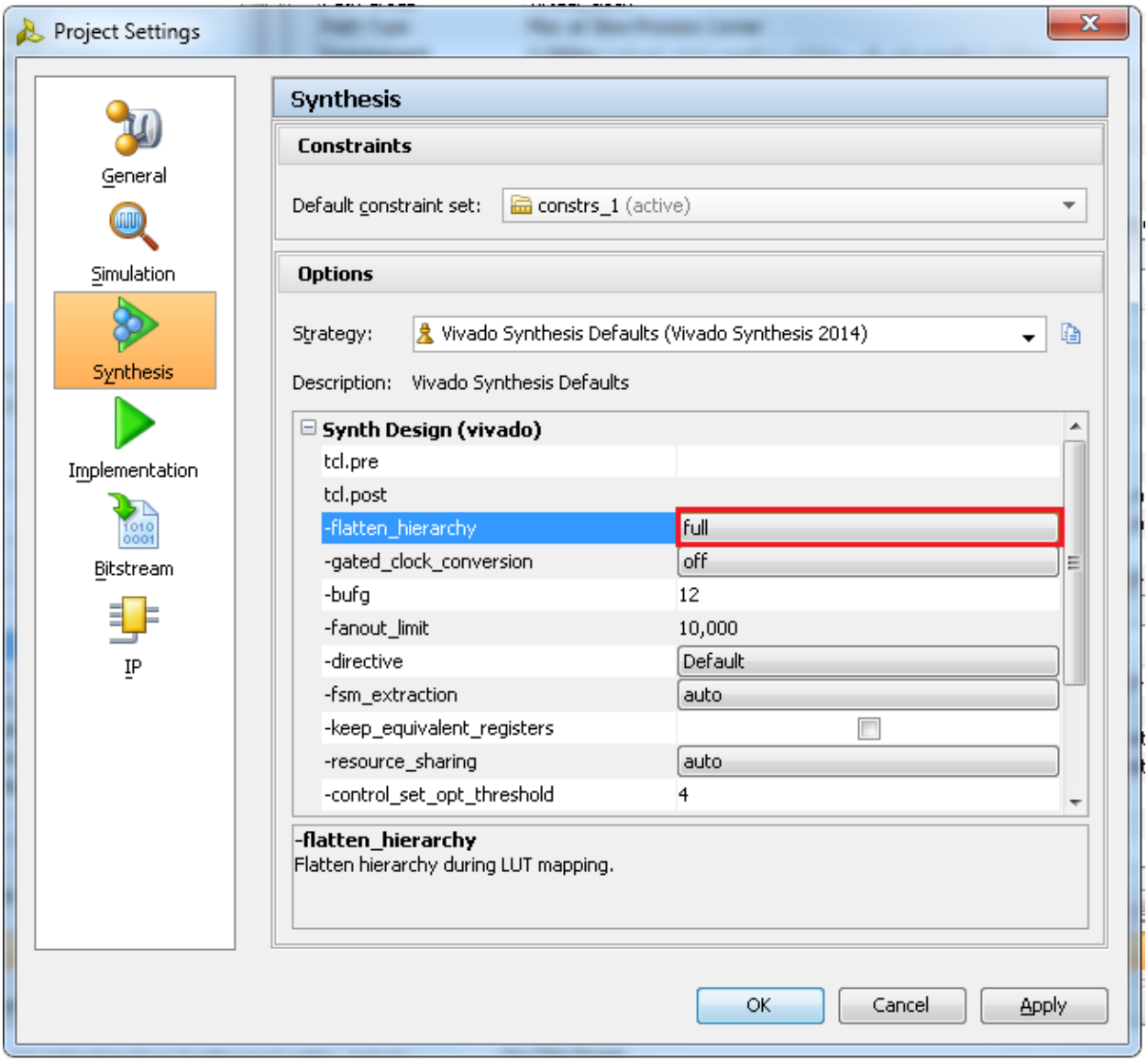


Figure 17. Selecting flatten hierarchy option

3-5-3. Click **OK**.

3-5-4. A Create New Run dialog box will appear asking you whether you want to create a new run since the settings have been changed.

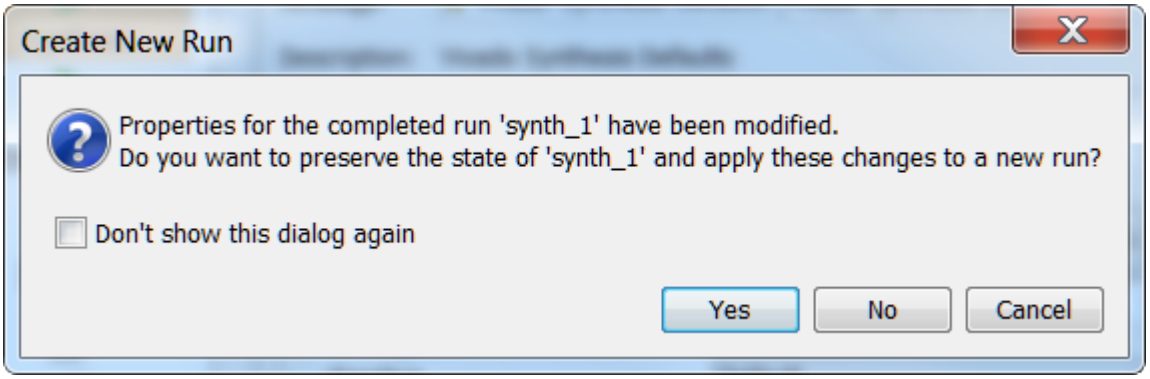


Figure 18. Create New Run dialog box

3-5-5. Click **Yes**.

3-5-6. Change the name from **synth_2** to **synth_flatten** and click **OK**. Close the synthesized design via **File > Close Synthesized Design**. Click **OK** to close the synthesized design.

3-5-7. Click **Run Synthesis** to synthesize the design.

3-5-8. Click **OK** to open the synthesized design when synthesis process is completed.

3-5-9. Click on **Schematic** under the *Synthesized Design* tasks of *Synthesis* tasks of the *Flow Navigator* pane to view the synthesized design in a schematic view.

Notice that the design is completely flattened.

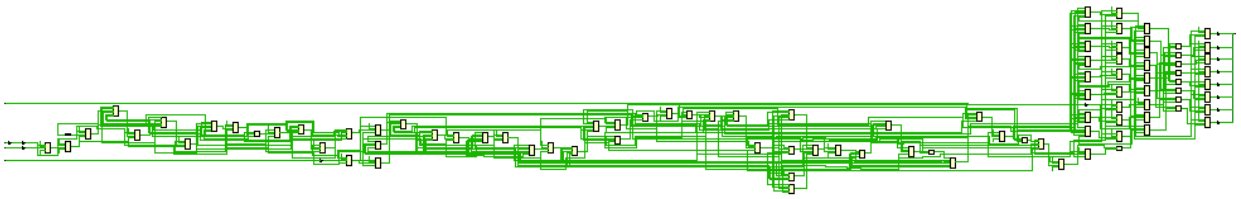


Figure 19. Flattened design for the ZedBoard

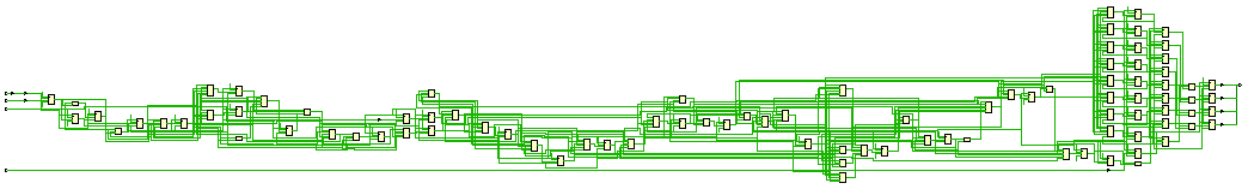


Figure 19. Flattened design for the Zybo

3-5-10. Click on **Report Utilization** and observe that the hierarchical utilization is no longer available.

3-6. Write the checkpoint in order to analyze the results without going through the actual synthesis process.

3-6-1. Select **File > Write Checkpoint…** to save the processed design so it can be opened later for further analysis.

3-6-2. A dialog box will appear showing the default name of the file (checkpoint_2.dcp) in the current project directory.

3-6-3. Click **OK**.

3-6-4. Close the project by selecting **File > Close Project**. Click **OK** to confirm closure.

Read the Checkpoints Step 4

4-1. Read the previously saved checkpoint (checkpoint_1) in order to analyze the results without going through the actual synthesis process.

4-1-1. Select **File > Open Checkpoint…** at the *Getting Started* screen.

4-1-2. Browse to **<2014_2_zynq_labs>\lab2** and select **checkpoint_1.**

4-1-3. Click **OK**.

4-1-4. The schematic will be shown. If it is not shown, in the netlist pane, select the top-level instance, **uart_led**, right-click and select **Schematic**.

You will see the hierarchical blocks. You can double-click on any of the first-level block and see the underlying blocks. You can also select any lower-level block in the netlist tab, right-click and select Schematic to see the corresponding level design.

4-1-5. In the netlist pane, select the top-level instance, **uart_led**, right-click and select **Show Hierarchy.**

You will see how the blocks are hierarchically connected.

4-1-6. Select **Tools > Timing > Report Timing Summary** and click **OK** to see the report you saw previously.

4-1-7. Select **Tools > Report > Report Utilization…** and click **OK** to see the utilization report you saw previously

4-1-8. Select **File > Open Checkpoint**, browse to **<2014_2_zynq_labs>\lab2** and select **checkpoint_2.**

4-1-9. Click **No** to keep the Checkpoint_1 open.

This will invoke a second Vivado GUI.

4-1-10. The schematic will be shown. If it is not shown, in the netlist tab, select the top-level instance, **uart_led**, right-click and select **Schematic**.

You will see the flattened design.

4-1-11. Right click on the top-level instance, **uart_led** and select **Show Hierarchy**. Notice the hierarchy has been flattened.

4-1-12. You can generate the desired reports on this checkpoint as you wish.

4-1-13. Close the **Vivado** program by selecting **File > Exit** and click **OK**.

Conclusion

In this lab you applied the timing constraints and synthesized the design. You viewed various post-synthesis reports. You wrote checkpoints and read it back to perform the analysis you were doing during the design flow. You saw the effect of changing synthesis settings.

Answers

1. Look through the table and find the number used of each of the following elements, applies to both the ZedBoard / Zybo:

   FF: 48 / 45

   LUT: 41 / 38

   I/O: 12 / 8

   BUFG: 1 / 1

2. Look through the report and find the number used of each of the following elements, applies to both the ZedBoard /Zybo:

   FF: 48 / 45

   LUT: 41 / 38

   I/O: 12 / 8

   BUFG: 1 / 1

3. From the power report, find the % power consumption used by each of the following elements, is specific to the ZedBoard / Zybo:

   Clocks: 23% / 31%

   Signals: 8% / 11%

   Logic: 4% / 6%

   I/O: 65% / 52%