Synthesizing a RTL Design

Introduction

This lab shows you the synthesis process and effect of changing of 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 the lab.

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 8 LEDs. When a push button is pressed, the lower and upper nibbles are swapped. The block diagram is as shown in **Figure 1**.

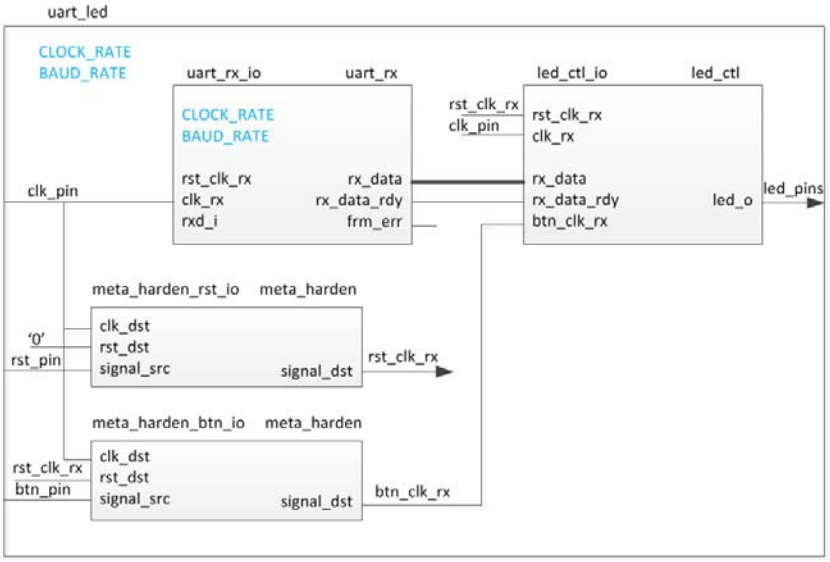
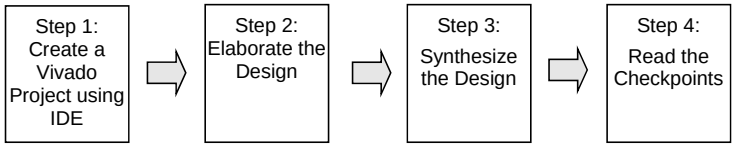


Figure 1. The Completed Design

General Flow



In the instructions below;
{***sources***} refers to: C:\xup\fpga_flow\2016_2_artix7_sources
{***labs***} refers to : C:\xup\fpga_flow\labs

Board support for the Basys3, Nexys4 DDR, Nexys Video is not included in Vivado 2016.2 by default. The relevant zip files need to be extracted and saved to: {Vivado installation}\data\boards\board_files\.

These files can be downloaded either from the Digilent, Inc. webpage (https://reference.digilentinc.com/vivado/boardfiles2015) or the XUP webpage (http://www.xilinx.com/support/university/vivado/vivado-workshops/Vivado-fpga-design-flow.html) where this material is also hosted.

Create a Vivado Project using IDE — Step 1

1-1. Launch Vivado and create a project targeting XC7A35TCPG236-1 (Basys3), or XC7A100TCSG324-1 (Nexys4 DDR), or XC7A200TSBG484-1 (Nexys Video), and using the Verilog HDL. Use the provided Verilog source files, uart_led_pins_<board>.xdc and uart_led_timing.xdc files from the {sources}*lab2* directory.

1-1-1. Open Vivado by selecting **Start > All Programs > Xilinx Design Tools > Vivado 2016.2 > Vivado 2016.2**

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 **{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. Using the drop-down buttons, select **Verilog** as the *Target Language* and *Simulator Language* in the *Add Sources* form.

1-1-7. Click on the **Green Plus** button, then the **Add Files…** button and browse to the **{sources}\lab2** directory, select all the Verilog files *(led_ctl.v, meta_harden.v, uart_baud_gen.v, uart_led.v, uart_rx.v, and uart_rx_ctl.v),* click **OK**, and then click **Next** to get to the *Add Existing IP* form.

1-1-8. Since we do not have any IP to add, click **Next** to get to the *Add Cons*traints form.

1-1-9. Click on the **Green Plus** button, then **Add Files…** and browse to the **{sources}\lab2** directory (if necessary), 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 **XC7A100TCSG324-1** (for the Nexys4 DDR), **XC7A35TCPG236-1** (for the Basys3) part, or **XC7A200tsbg484-1** (for the Nexys Video).

Using the **Boards** option, you may select the **Nexys4 DDR**, the **Basys3**, or the **Nexys Video** depending on your board.

1-1-12. Click **Next**.

1-1-13. Click **Finish** to create the Vivado project.

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.

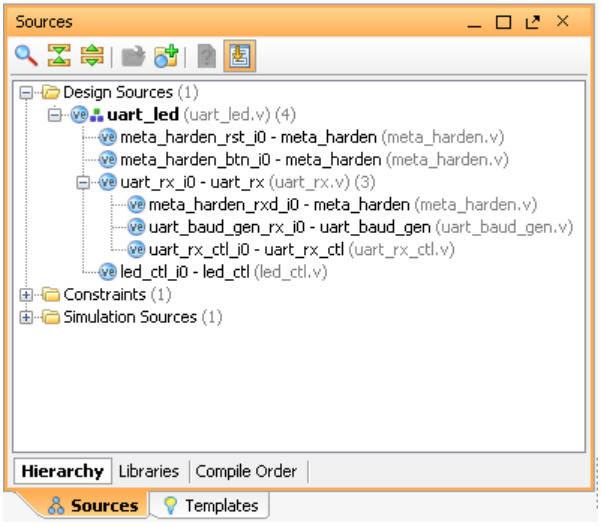


Figure 2. Opening the source file

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). 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 module uses the baud rate generator and a finite state machine. The rxd_pin is sampled at a 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.

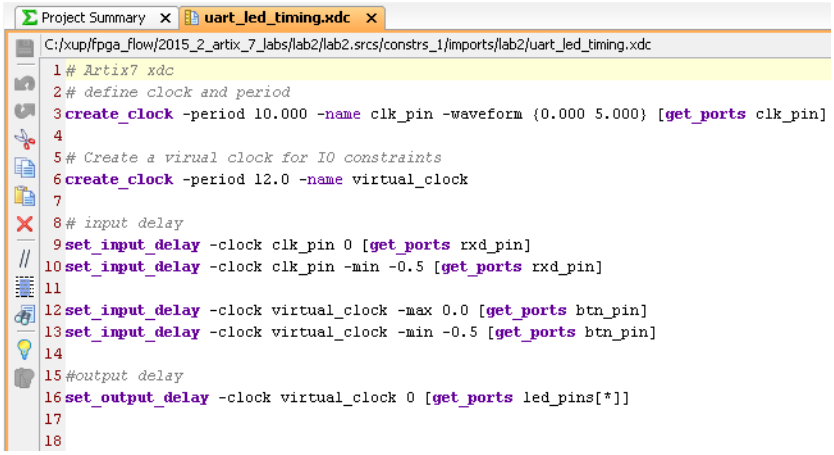


Figure 3. Timing constraints

Line 3 creates the period constraint of 10 ns with a duty cycle of 50%. Line 6 creates a virtual clock of 12 ns. 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 logical view of the design is displayed.

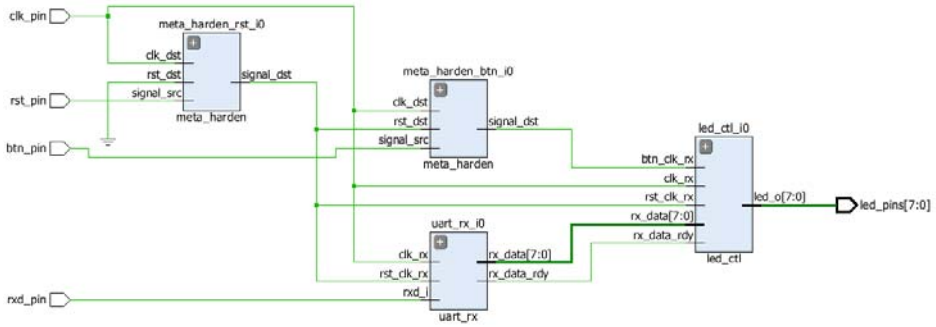


Figure 4. A logic view of the design

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

2-1-2. To see where the uart_rx_i0 gets generated, right-click on the uart_rx_i0 instance and select *Go To Source* and see that line 84 in the source code is generating it.

2-1-3. Double-click on the uart_rx_i0 instance in the schematic diagram to see the underlying components.

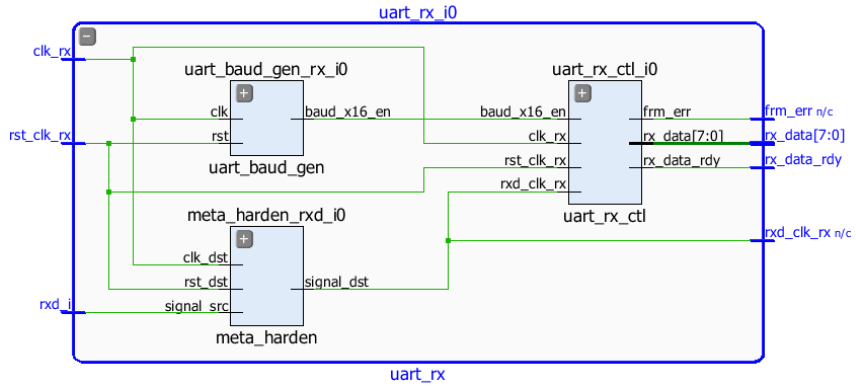


Figure 5. Lower level components of the uart_rx_i0 module

2-1-4. Click on **Report Noise** under the *Open Elaborated Design* entry of the *RTL Analysis* tasks of the *Flow Navigator* pane.

2-1-5. Click **OK** to generate the report named **ssn_1**.

2-1-6. View the ssn_1 report 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.

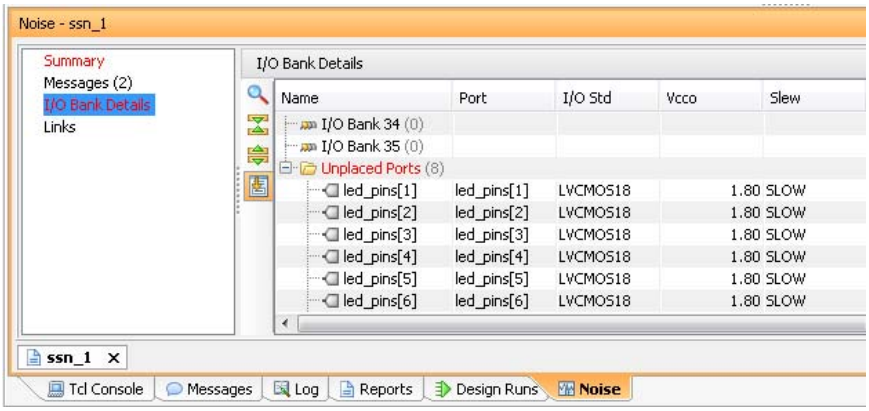
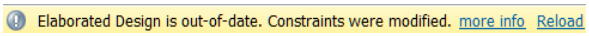


Figure 6. Noise report

2-1-7. Click on **Add Sources** under the *Project Navigator*, select *Add or Create Constraints* option and click **Next**.

2-1-8. Click on the **Green Plus** button, then the **Add Files…** button and browse to the **{sources}\lab2** directory, select the **uart_led_pins_<board>.xdc** file (depending on the target board), click **OK**, and then click **Finish** to add the pins location constraints.

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



2-1-9. Click on the **Reload** link. The constraints will be processed.

2-1-10. 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).

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 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 *Open Synthesized Design* tasks of *Synthesis* tasks of the *Flow Navigator* pane to view the synthesized design in a schematic view.

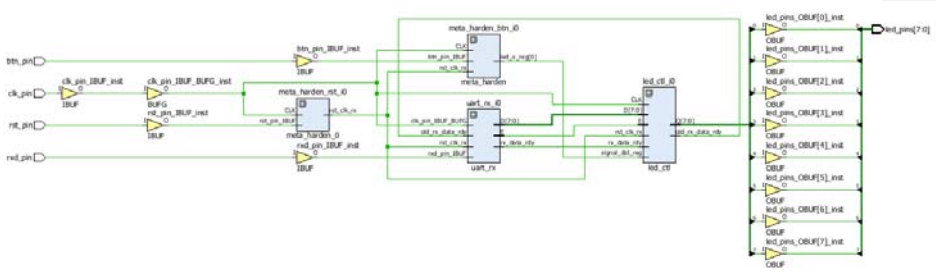


Figure 7. Synthesized design's schematic view

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. Double-click on 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 uart_baud_gen_rx_i0 is highlighted. Also notice that the CLOCK_RATE and BAUD_RATE parameters are passed to the module being called.

Commented [PP1]: Why this is highlighted?

3-1-8. 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.

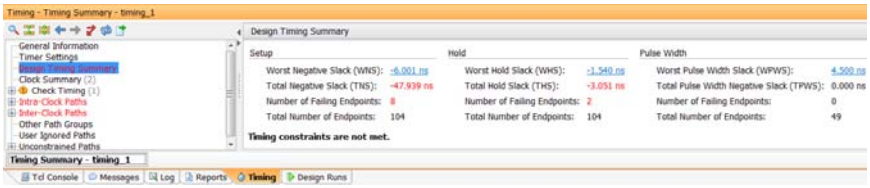


Figure 8. Timing report for the Nexys4 DDR

Figure 8. Timing report for the Basys3

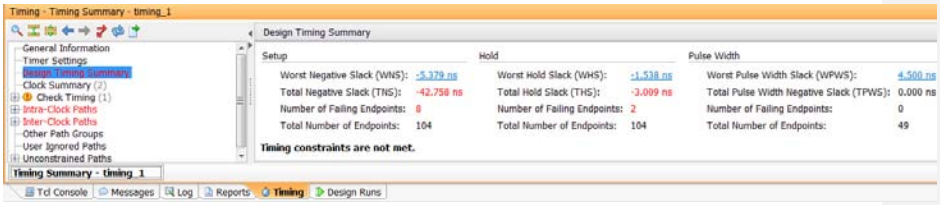


Figure 8. Timing report for the Nexys Video

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 indicate total number of failing paths.

3-2-3. Click on the WNS link and see the 8 failing paths.

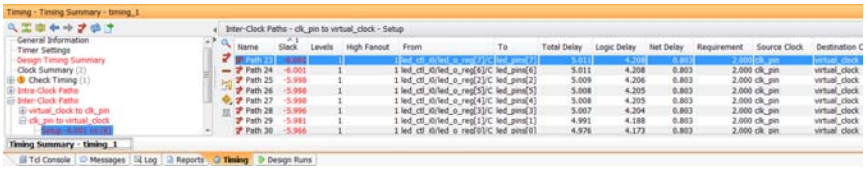


Figure 9. The 8 failing paths for the Nexys4 DDR

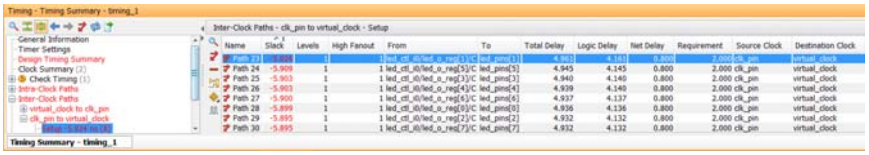


Figure 9. The 8 failing paths for the Basys3

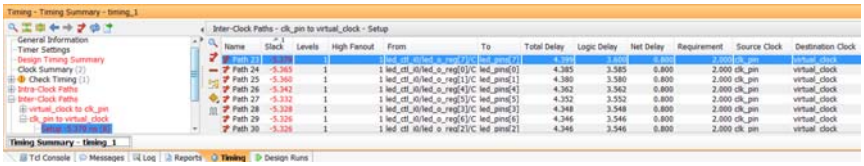
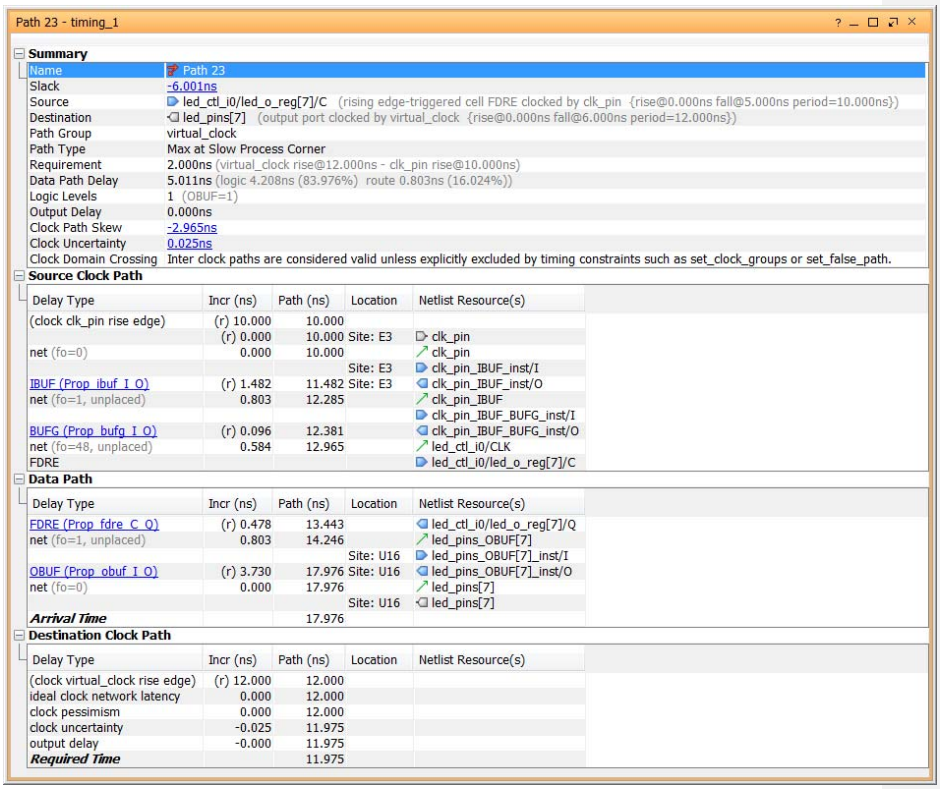


Figure 9. The 8 failing paths for the Nexys Video

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



Path 23 - timing_1

Summary

Name	Path 23
Slack	-6.001ns
Source	led_ctl_i0/led_o_reg[7]/C (rising edge-triggered cell FDRE clocked by clk_pin {rise@0.000ns fall@5.000ns period=10.000ns})
Destination	led_pins[7] (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	5.011ns (logic 4.208ns (83.976%) route 0.803ns (16.024%))
Logic Levels	1 (OBUF=1)
Output Delay	0.000ns
Clock Path Skew	-2.965ns
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	Incr (ns)	Path (ns)	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=48, unplaced)	0.584	12.965		led_ctl_i0/CLK
FDRE				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) 0.478	13.443		led_ctl_i0/led_o_reg[7]/Q
net (fo=1, unplaced)	0.803	14.246		led_pins_OBUF[7]
			Site: U16	led_pins_OBUF[7]_inst/I
OBUF (Prop_obuf_I_O)	(r) 3.730	17.976	Site: U16	led_pins_OBUF[7]_inst/O
net (fo=0)	0.000	17.976		led_pins[7]
			Site: U16	led_pins[7]
Arrival Time		17.976		

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 10. Worst failing path for the Nexys4 DDR

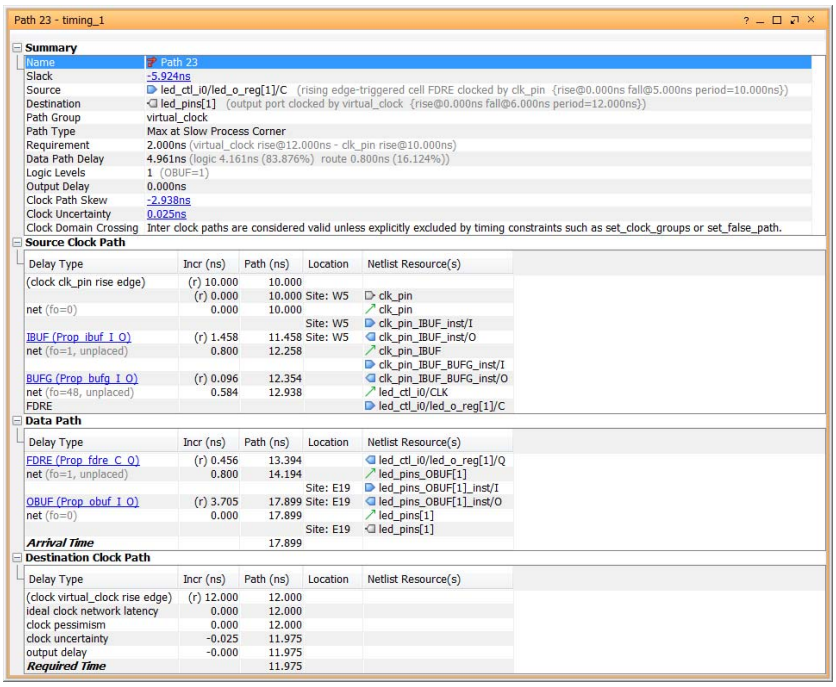


Figure 10. Worst failing path for the Basys3

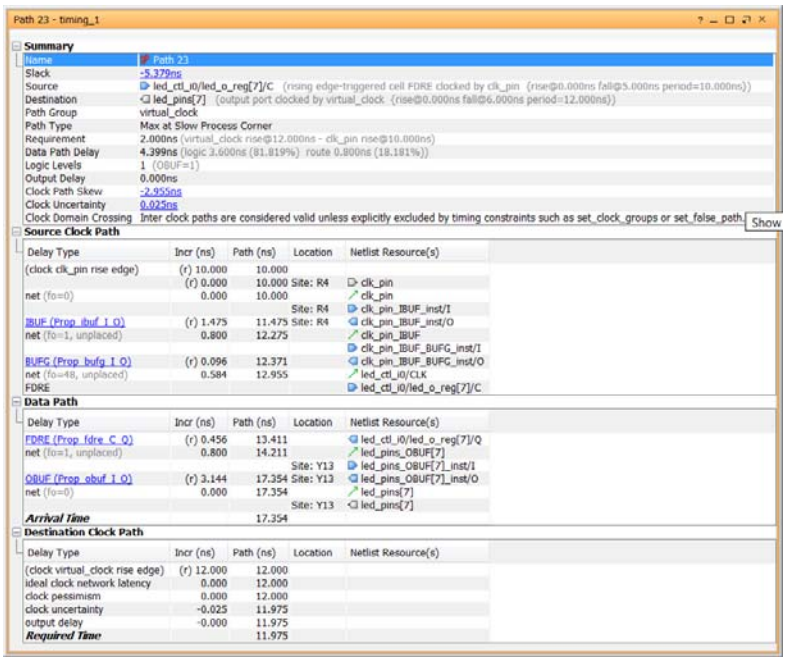


Figure 10. Worst failing path for the Nexys Video

Note that this is an estimate only. The nets are specified as unplaced and have all been allocated default values (0.800 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.

Resource	Utilization	Available	Utilization %
LUT	42	63400	0.07
FF	48	126800	0.04
IO	12	210	5.71
BUFG	1	32	3.13

Figure 11. Utilization report for the Nexys4 DDR

Resource	Utilization	Available	Utilization %
LUT	42	20800	0.20
FF	48	41600	0.12
IO	12	106	11.32
BUFG	1	32	3.13

Figure 11. Utilization report for the Basys3

Resource	Utilization	Available	Utilization %
LUT	42	134600	0.03
FF	48	269200	0.02
IO	12	285	4.21
BUFG	1	32	3.13

Figure 11. Utilization report for the Nexys Video

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.

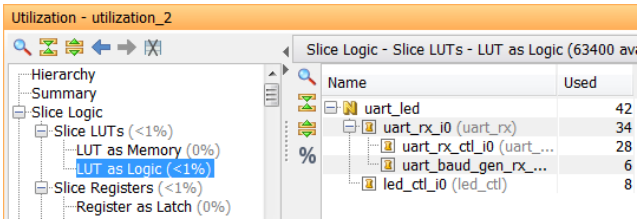


Figure 12. Utilization of lower-level modules for the Nexys4 DDR

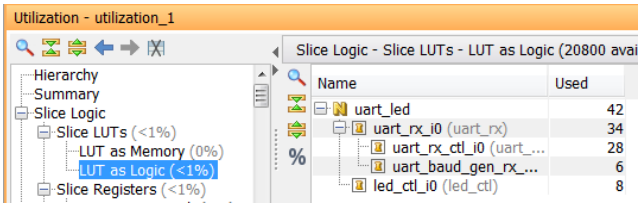


Figure 12. Utilization of lower-level modules for the Basys3

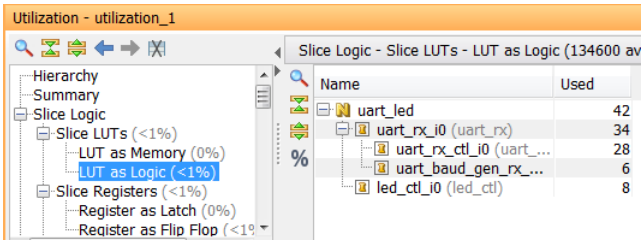


Figure 12. Utilization of lower-level modules for the Nexys Video

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 rate, or environment information was entered.

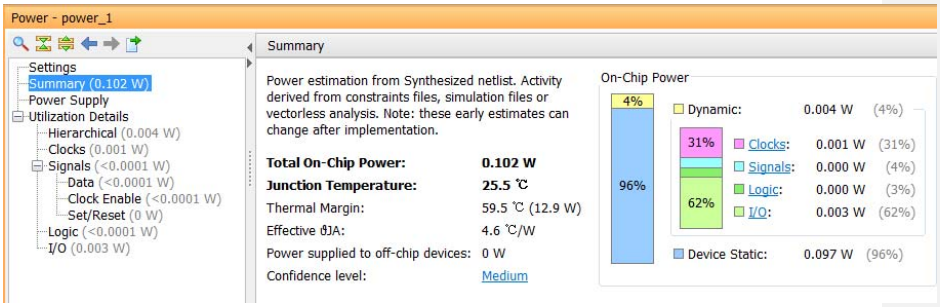


Figure 13. Power consumption estimation for the Nexys4 DDR

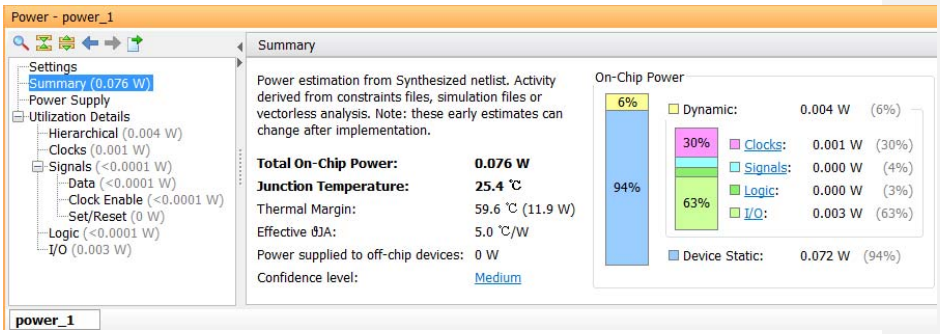


Figure 13. Power consumption estimation for the Basys3

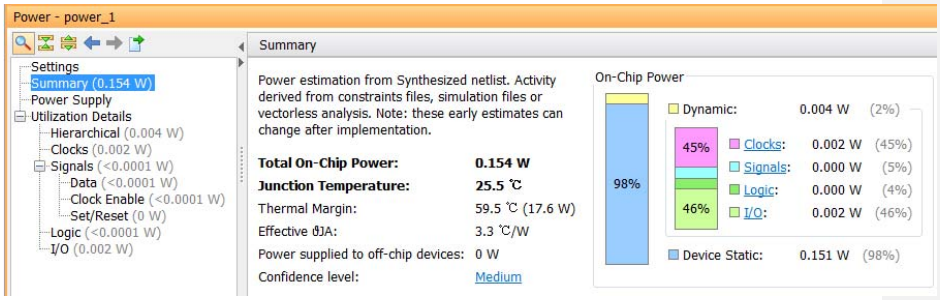


Figure 13. Power consumption estimate for the Nexys Video

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.

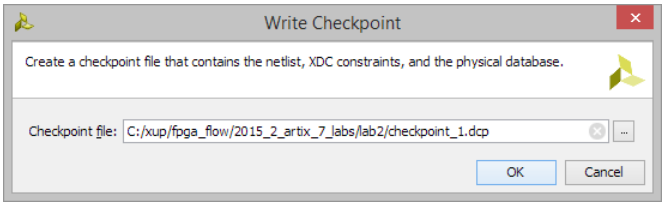


Figure 14. Writing checkpoint

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.

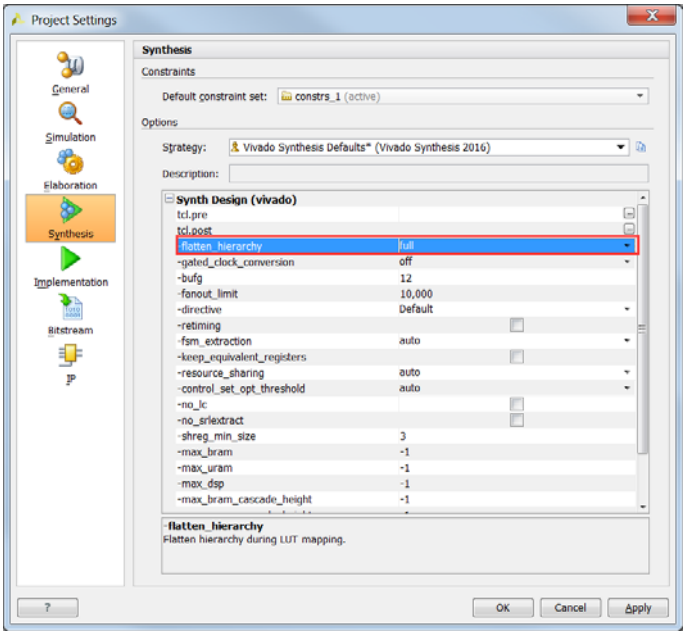


Figure 15. Selecting flatten hierarchy option

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.

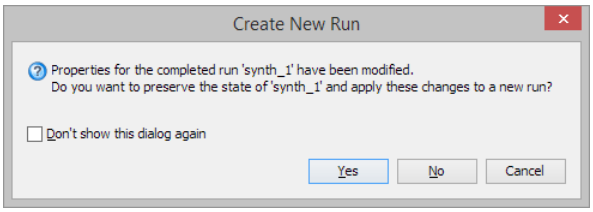


Figure 16. Create New Run dialog box

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

3-5-6. Change the name from **synth_2** to **synth_flatten** and click **OK**.

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

3-5-8. Click **Save**, **OK**, and again **OK** to save the synthesized design and save the constraints.

The Reload Design dialog box may re-appear. Click **Cancel**.

3-5-9. Click **OK** to open the synthesized design when synthesis process is completed.

3-5-10. Click on **Schematic** under the *Open 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.

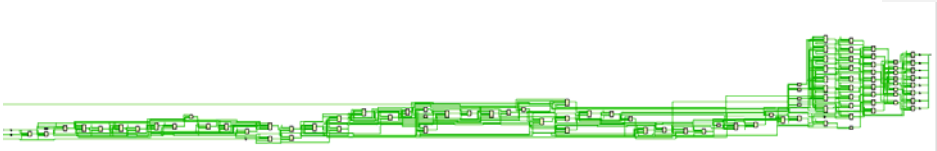


Figure 17. Flattened design

3-5-11. Click on **Report Utilization** and observe that the hierarchical utilization is no longer available. Also note that the number of **Slice Registers** is **48**.

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**.

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 **{labs}\lab2** and select **checkpoint_1.**

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

4-1-4. If the schematic isn't open by default, in the netlist tab, 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 tab, 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 **{labs}\lab2** and select **checkpoint_2.**

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

This will invoke second Vivado GUI.

4-1-10. If the schematic isn't open by default, 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. You can generate the desired reports on this checkpoint as you wish.

4-1-12. 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:

   FF: 48
   LUT: 42
   I/O: 12
   BUFG: 1

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

   FF: 48
   LUT: 42
   I/O: 12
   BUFG: 1

3. From the power report, find the % power consumption used by each of the following (Nexys4 DDR):

   Clocks: 31%
   Signals: 4%
   Logic: 3%
   I/O: 62%

   For the Basys3:
   Clocks: 30%
   Signals: 4%
   Logic: 3%
   I/O: 63%

   For the Nexys Video:
   Clocks: 45%
   Signals: 5%
   Logic: 4%
   I/O: 46%