Synthesizing an 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 the lab.

Design Description

The design consists of a uart receiver receiving the input typed on a keyboard and displaying the least significant 4 bits of binary equivalent of the typed character on the 4 LEDs. When a push button is pressed, the upper bits of the binary equivalent are displayed. The block diagram is as shown in **Figure 1**.

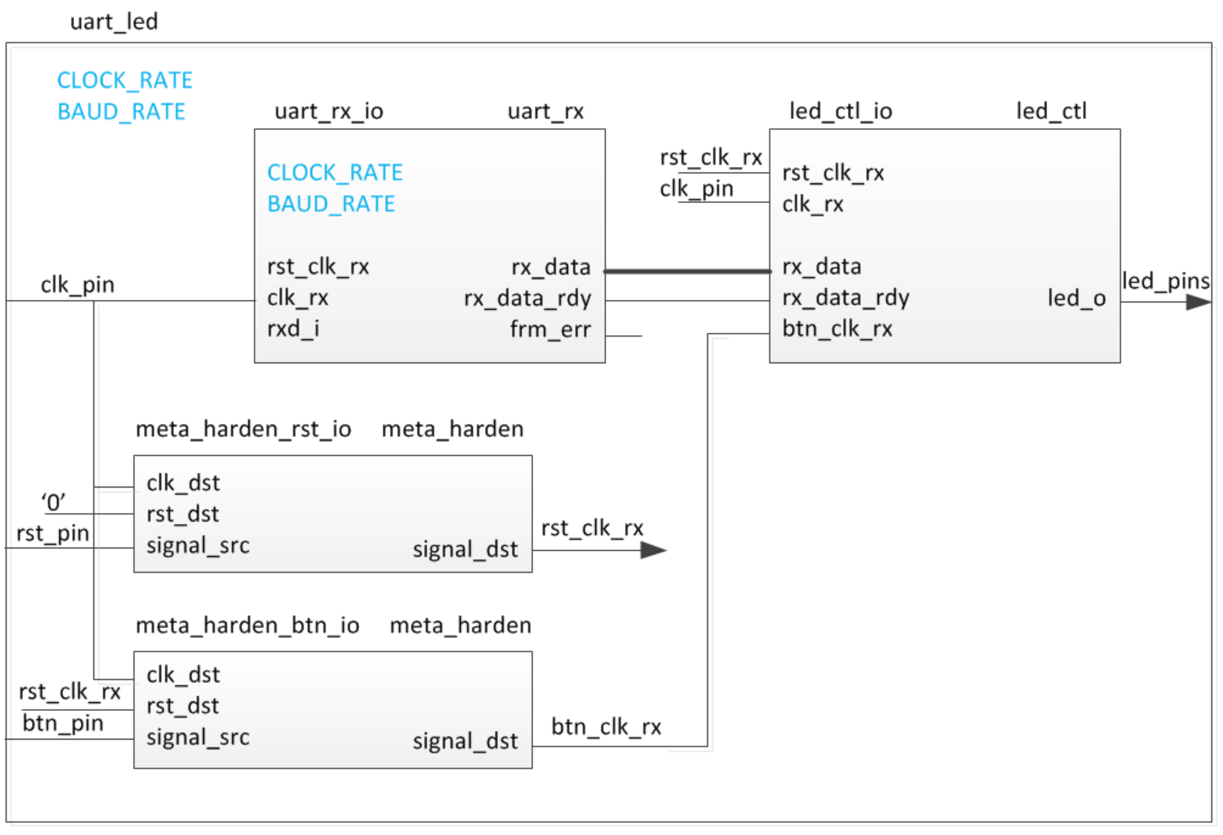
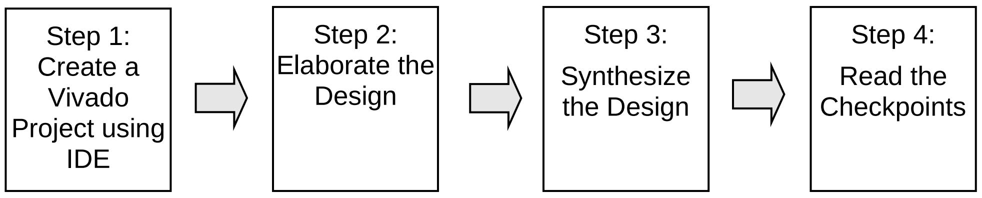


Figure 1. The Completed Design

General Flow



Create a Vivado Project using IDE — Step 1

1-1. Launch Vivado and create a project targeting the XC7Z010CLG400-1 device and using the Verilog HDL. Use the provided Verilog source files and uart_led_timing.xdc file from the *lab2a_files* directory.

1-1-1. Open Vivado by selecting **Start > All Programs > Xilinx Design Tools > Vivado 201X.Y > Vivado 201X.Y**

*** In Linux. Double click the Icon in desktop or in a shell type "vivado"**

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 your **P1** directory and click **Select**.

1-1-4. Enter **lab2a** 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 **Add Files…** button, browse to the **lab2a_files** 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 Constraints* form.

1-1-8. Add only the *uart_led_timing.xdc* file, do not add uart_led_pins.xdc. Click **OK** and **Next.**

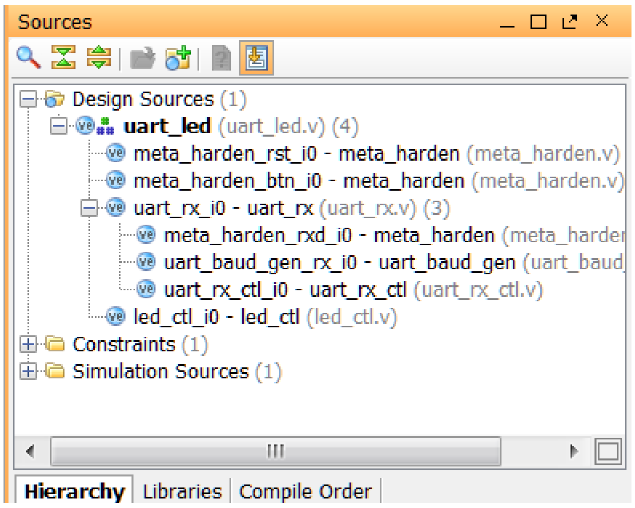
This Xilinx Design Constraints file assigns the basic timing constraints (period, input delay, and output delay) to the design.

1-1-9. In the *Default Part* form, using the **Parts** option and various drop-down fields of the **Filter** section (*Family=Zynq-7000, Package=clg400, Speed=-1)*, select the **XC7Z010CLG400-1** part. Click **Next**.

1-1-10. 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.



```
Sources
Design Sources (1)
  uart_led (uart_led.v) (4)
    meta_harden_rst_i0 - meta_harden (meta_harden.v)
    meta_harden_btn_i0 - meta_harden (meta_harden.v)
    uart_rx_i0 - uart_rx (uart_rx.v) (3)
      meta_harden_rxd_i0 - meta_harden (meta_harde
      uart_baud_gen_rx_i0 - uart_baud_gen (uart_baud
      uart_rx_ctl_i0 - uart_rx_ctl (uart_rx_ctl.v)
    led_ctl_i0 - led_ctl (led_ctl.v)
Constraints (1)
Simulation Sources (1)
```

Figure 2. Opening the source file

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

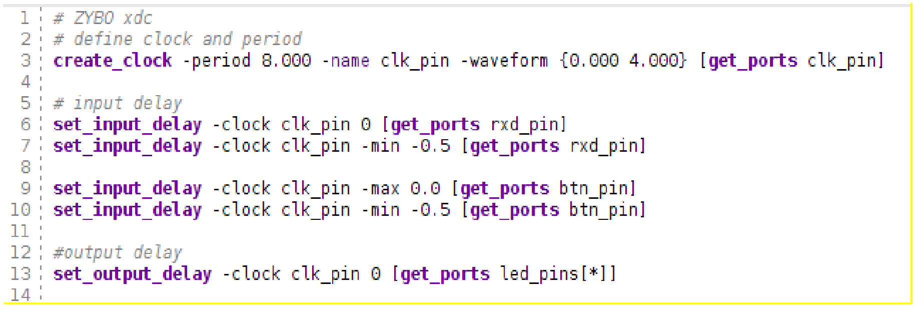
Notice in the Verilog code that the BAUD_RATE and CLOCK_RATE parameters are defined to be 115200 and 125 MHz respectively as shown in the design diagram (Figure 1). Also notice that the lower level modules are instantiated. The meta_harden modules are used to synchronize the asynchronous reset and push-button inputs.

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

It uses the baud rate generator and the 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  # 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  # input delay
 6  set_input_delay -clock clk_pin 0 [get_ports rxd_pin]
 7  set_input_delay -clock clk_pin -min -0.5 [get_ports rxd_pin]
 8
 9  set_input_delay -clock clk_pin -max 0.0 [get_ports btn_pin]
10  set_input_delay -clock clk_pin -min -0.5 [get_ports btn_pin]
11
12  #output delay
13  set_output_delay -clock clk_pin 0 [get_ports led_pins[*]]
14
```

Figure 3. Timing constraints

Line 3 creates the period constraint of 8 ns with a duty cycle of 50% (125 Mhz). The rxd_pin is constrained with respect to the design clock (lines 6, and 7). The btn_pin is constrained in lines 9, 10. The led_pins are constrained in line 13.

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.

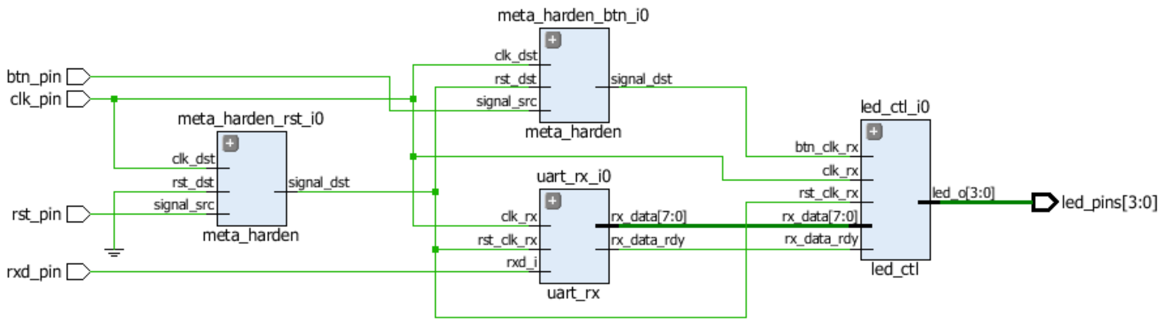


Figure 4. A logic view of the design

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.

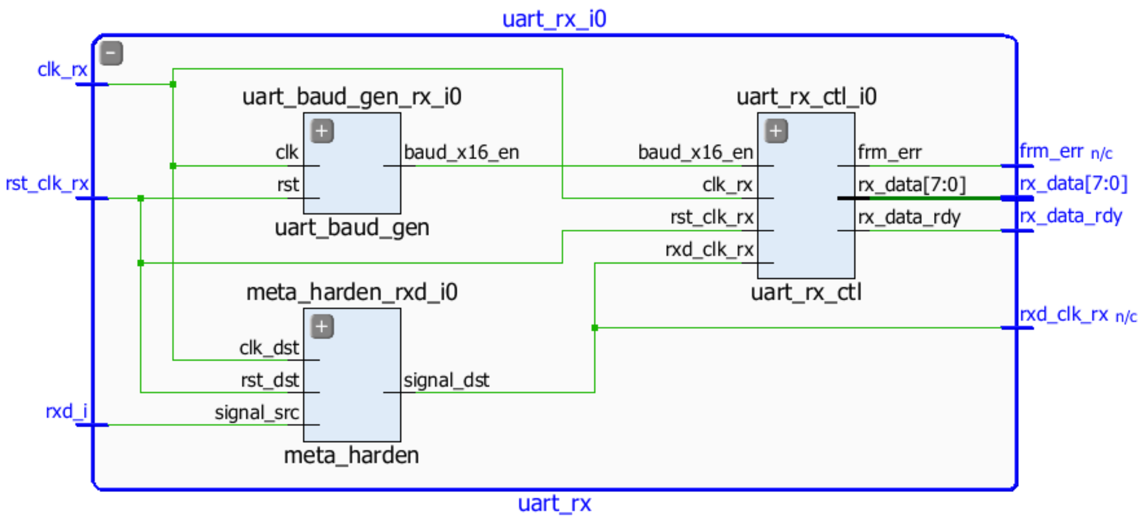
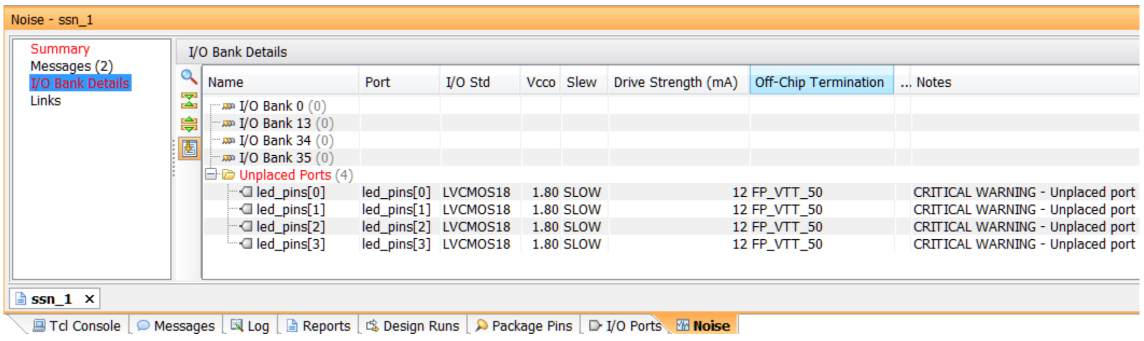


Figure 5. Lower level components of the uart_rx_i0 module

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.


Noise - ssn_1

Summary
Messages (2)
I/O Bank Details
Links

I/O Bank Details

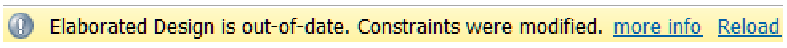
Name	Port	I/O Std	Vcco	Slew	Drive Strength (mA)	Off-Chip Termination	...	Notes
I/O Bank 0 (0)								
I/O Bank 13 (0)								
I/O Bank 34 (0)								
I/O Bank 35 (0)								
Unplaced Ports (4)								
led_pins[0]	led_pins[0]	LVCMOS18	1.80	SLOW	12	FP_VTT_50		CRITICAL WARNING - Unplaced port
led_pins[1]	led_pins[1]	LVCMOS18	1.80	SLOW	12	FP_VTT_50		CRITICAL WARNING - Unplaced port
led_pins[2]	led_pins[2]	LVCMOS18	1.80	SLOW	12	FP_VTT_50		CRITICAL WARNING - Unplaced port
led_pins[3]	led_pins[3]	LVCMOS18	1.80	SLOW	12	FP_VTT_50		CRITICAL WARNING - Unplaced port

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 **Add Files…** button, browse to the **lab2a_files** directory, select **uart_led_pins.xdc** file, 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.


Elaborated Design is out-of-date. Constraints were modified. more info Reload

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

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. Click **Ok**.

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 Σ.

3-1-4. Click on the **Table** tab in the Utilization section of 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.

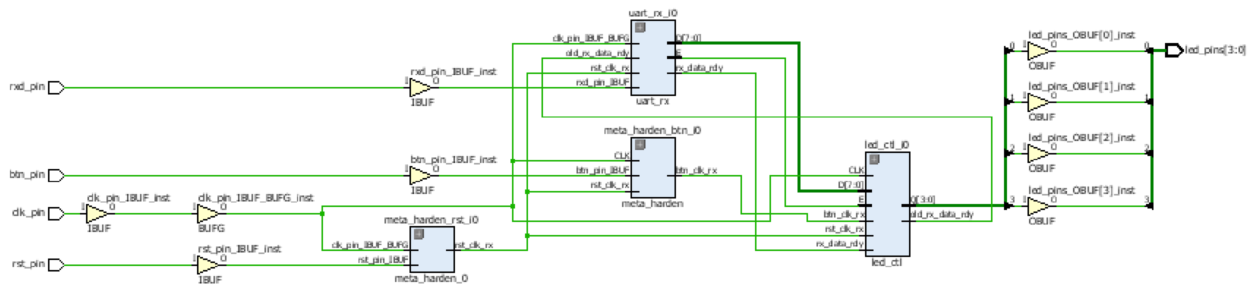


Figure 7. Synthesized design's schematic view

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

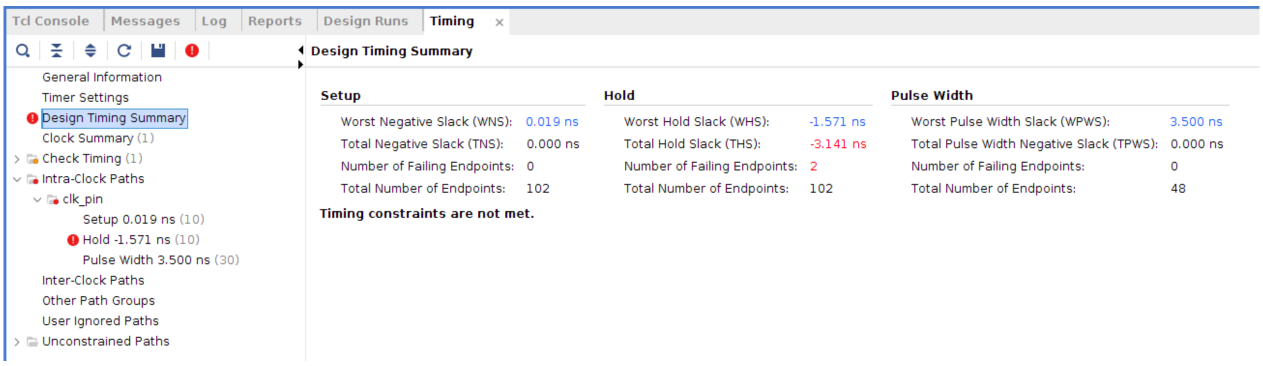
3-1-8. Double-click on the **meta_harden_rxd_io** instance to see how the synchronization circuit is being implemented using two FFs. This synchronization is necessary to reduce the likelihood of meta-stability.

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 *Open Synthesized Design* tasks of the *Flow Navigator* pane.

3-2-2. Click **OK** to generate the *timing_1* report.



Tcl Console | Messages | Log | Reports | Design Runs | Timing ×

Design Timing Summary

- General Information
- Timer Settings
- Design Timing Summary
- Clock Summary (1)
- Check Timing (1)
- Intra-Clock Paths
  - clk_pin
    - Setup 0.019 ns (10)
    - Hold -1.571 ns (10)
    - Pulse Width 3.500 ns (30)
- Inter-Clock Paths
- Other Path Groups
- User Ignored Paths
- Unconstrained Paths

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.019 ns	Worst Hold Slack (WHS):	-1.571 ns	Worst Pulse Width Slack (WPWS):	3.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	-3.141 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:	102	Total Number of Endpoints:	102	Total Number of Endpoints:	48

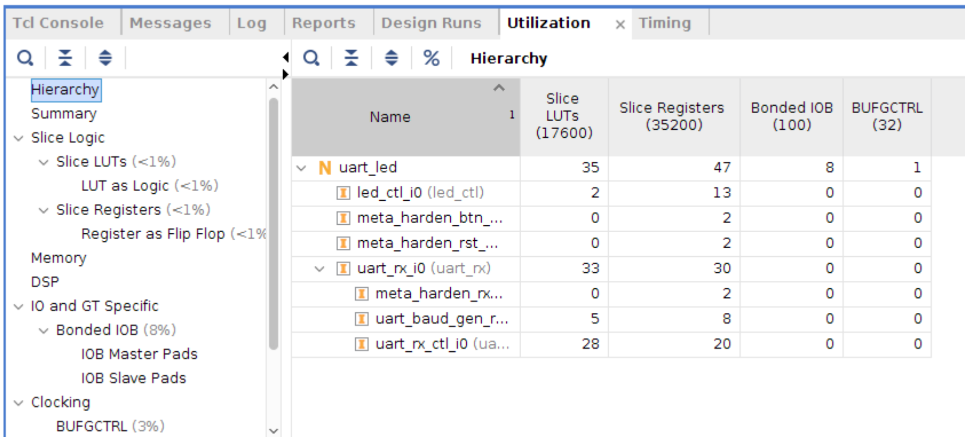
Timing constraints are not met.

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

We will come back to timing issues later on during the course.

3-3. Generate the utilization and power reports.

3-3-1. Click **Report Utilization** under *Open Synthesized Design*, and click **OK** to generate the *utilization_1* report.


Tcl Console | Messages | Log | Reports | Design Runs | Utilization | Timing

Hierarchy
- Hierarchy
- Summary
- Slice Logic
  - Slice LUTs (<1%)
    - LUT as Logic (<1%)
  - Slice Registers (<1%)
    - Register as Flip Flop (<1%
- Memory
- DSP
- IO and GT Specific
  - Bonded IOB (8%)
    - IOB Master Pads
    - IOB Slave Pads
- Clocking
  - BUFGCTRL (3%)

Name	Slice LUTs (17600)	Slice Registers (35200)	Bonded IOB (100)	BUFGCTRL (32)
uart_led	35	47	8	1
led_ctl_i0 (led_ctl)	2	13	0	0
meta_harden_btn_...	0	2	0	0
meta_harden_rst_...	0	2	0	0
uart_rx_i0 (uart_rx)	33	30	0	0
meta_harden_rx...	0	2	0	0
uart_baud_gen_r...	5	8	0	0
uart_rx_ctl_i0 (ua...	28	20	0	0

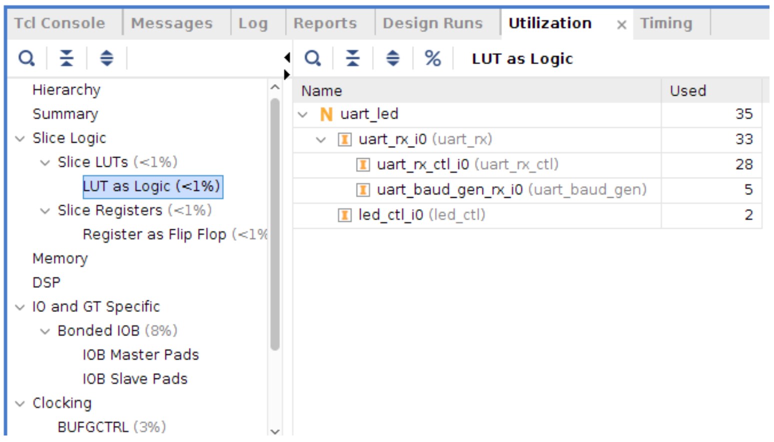
Figure 11. Utilization report

Question 2

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

FF (Flip Flops):	____________________
LUT (Slice LUTs):	____________________
I/O (Bonded IOBs):	____________________
BUFG (BUFGCTRL):	____________________

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.


Tcl Console | Messages | Log | Reports | Design Runs | Utilization | Timing

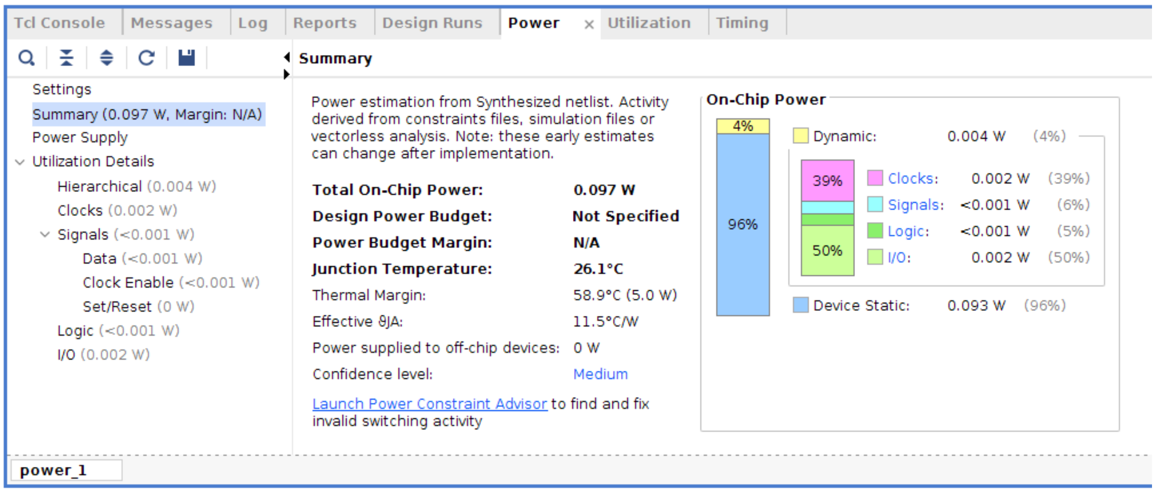
LUT as Logic
- Hierarchy
- Summary
- Slice Logic
  - Slice LUTs (<1%)
    - LUT as Logic (<1%)
  - Slice Registers (<1%)
    - Register as Flip Flop (<1%
- Memory
- DSP
- IO and GT Specific
  - Bonded IOB (8%)
    - IOB Master Pads
    - IOB Slave Pads
- Clocking
  - BUFGCTRL (3%)

Name	Used
uart_led	35
uart_rx_i0 (uart_rx)	33
uart_rx_ctl_i0 (uart_rx_ctl)	28
uart_baud_gen_rx_i0 (uart_baud_gen)	5
led_ctl_i0 (led_ctl)	2

Figure 12. Utilization of lower-level modules

3-3-3. Click **Report Power** under *Open 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.



Settings
Summary (0.097 W, Margin: N/A)
Power Supply
Utilization Details
  Hierarchical (0.004 W)
  Clocks (0.002 W)
  Signals (<0.001 W)
    Data (<0.001 W)
    Clock Enable (<0.001 W)
    Set/Reset (0 W)
  Logic (<0.001 W)
  I/O (0.002 W)

Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

Total On-Chip Power:	**0.097 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**26.1°C**
Thermal Margin:	58.9°C (5.0 W)
Effective ϑJA:	11.5°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Medium

Launch Power Constraint Advisor to find and fix invalid switching activity

On-Chip Power

Dynamic:	0.004 W	(4%)	4%
Clocks:	0.002 W	(39%)	39%
Signals:	<0.001 W	(6%)	
Logic:	<0.001 W	(5%)	
I/O:	0.002 W	(50%)	50%
Device Static:	0.093 W	(96%)	96%

Figure 13. Power consumption estimation