Implementing the Design

Introduction

This lab continues with the previous lab (lab2a). You will perform static timing analysis. You will implement the design with the default settings and generate a bitstream. Then you will open a hardware session and program the FPGA. You will use on-board UART of the ZYBO board to validate your design.

Objectives

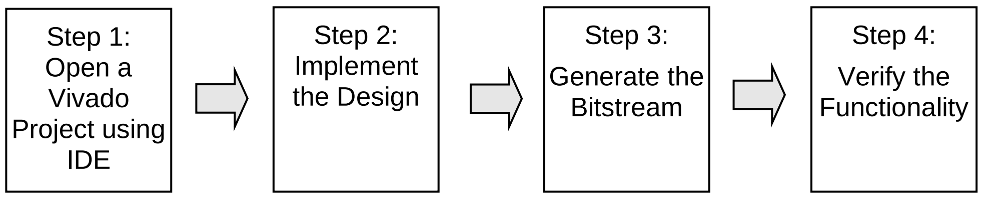
After completing this lab, you will be able to:

- 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

Implement the Design Step 2

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

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

1-1-2. Click **Ok**. If prompted to run the synthesis first before running the implementation process, click **Yes**.

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

1-1-3. Select the *Open Implemented Design* option and click **OK**

1-1-4. Click **Yes** to close the synthesized design, if prompted about it.

1-2. View the amount of FPGA resources consumed by the design using Report Utilization.

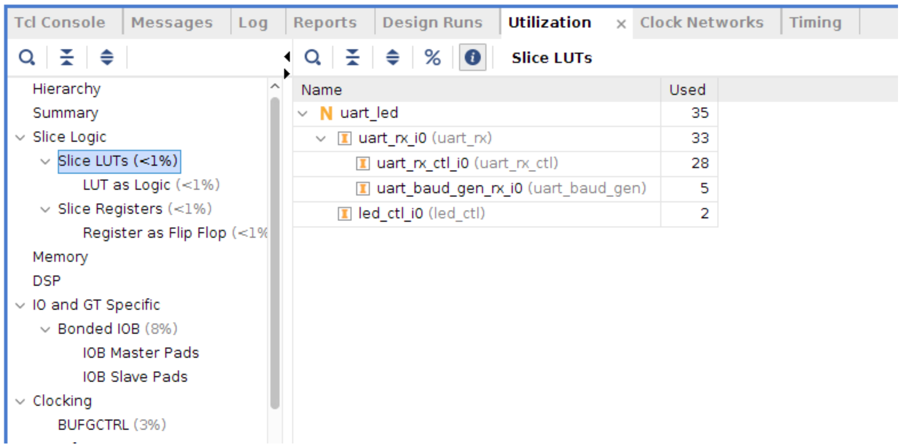
1-2-1. In the *Flow Navigator* pane, select ***Open Implemented Design*** > ***Report Utilization***.

The Report Utilization dialog box opens.

1-2-2. 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.

1-2-3. Select Slice LUTs to view how much and which module consumes the resource.



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

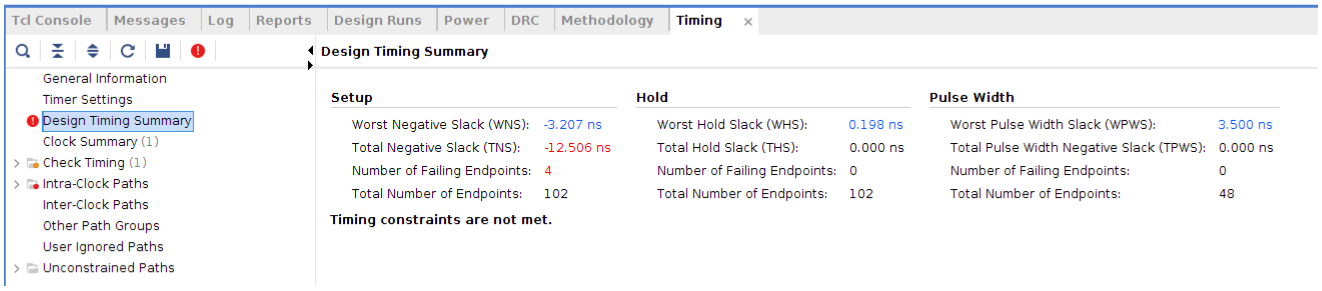
Figure 11. Resource utilization

1-3. Generate a timing summary report.

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

The Report Timing Summary dialog box opens.

1-3-2. Leave all the settings unchanged and click **OK** to generate the report.



Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	-3.207 ns	Worst Hold Slack (WHS):	0.198 ns	Worst Pulse Width Slack (WPWS):	3.500 ns
Total Negative Slack (TNS):	-12.506 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	4	Number of Failing Endpoints:	0	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.

Figure 12. The timing summary report showing timing violations

1-3-3. Expand the **Intra-clock** folder on the left, expand *clk_pin*, and select setup group to see the list of 10 worst case delays on the right side.

1-3-4. Double-click on the worst path (the first) to see how that is made up of. Also right-click on it and select **Schematic.**

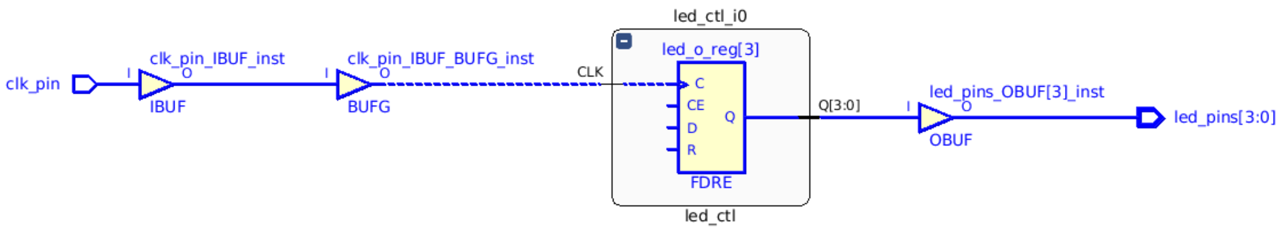


Figure 14. Schematic view of the worst path delay

1-3-5. Click on the **Device** tab and see the highlighted path in the view.

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

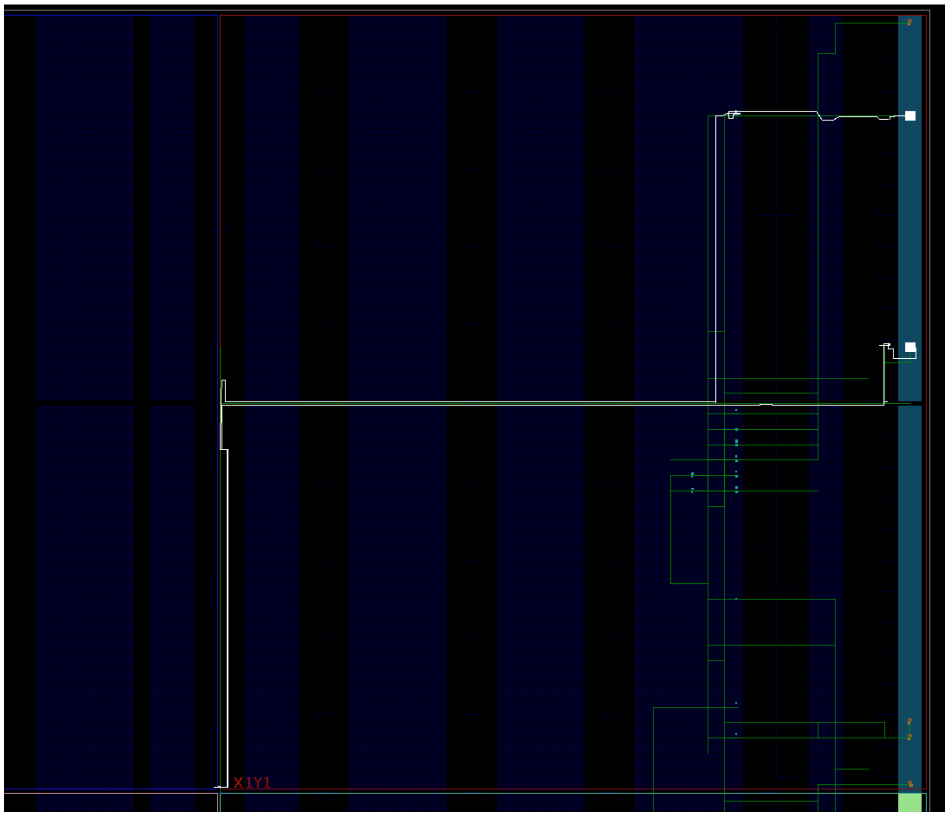


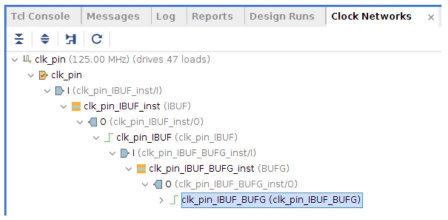
Figure 15. Path displayed in the Device view

1-3-6. Select ***Open Implemented Design > Report Clock Networks***.

1-3-7. Click **OK**.

The Clock Networks report will be displayed in the Console pane showing one clock net entry.

1-3-8. Select *clk_pin* entry and observe the selected nets in the Device view. (The clock nets are located in the X1Y0 and X1Y1 clock regions.)



```
clk_pin (125.00 MHz) (drives 47 loads)
  clk_pin
    I (clk_pin_IBUF_inst/I)
      clk_pin_IBUF_inst (IBUF)
        O (clk_pin_IBUF_inst/O)
          clk_pin_IBUF (clk_pin_IBUF)
            I (clk_pin_IBUF_BUFG_inst/I)
              clk_pin_IBUF_BUFG_inst (BUFG)
                O (clk_pin_IBUF_BUFG_inst/O)
                  clk_pin_IBUF_BUFG (clk_pin_IBUF_BUFG)
```

1-3-9. Then select clk_pin_IBUF_BUFG to see details in routing

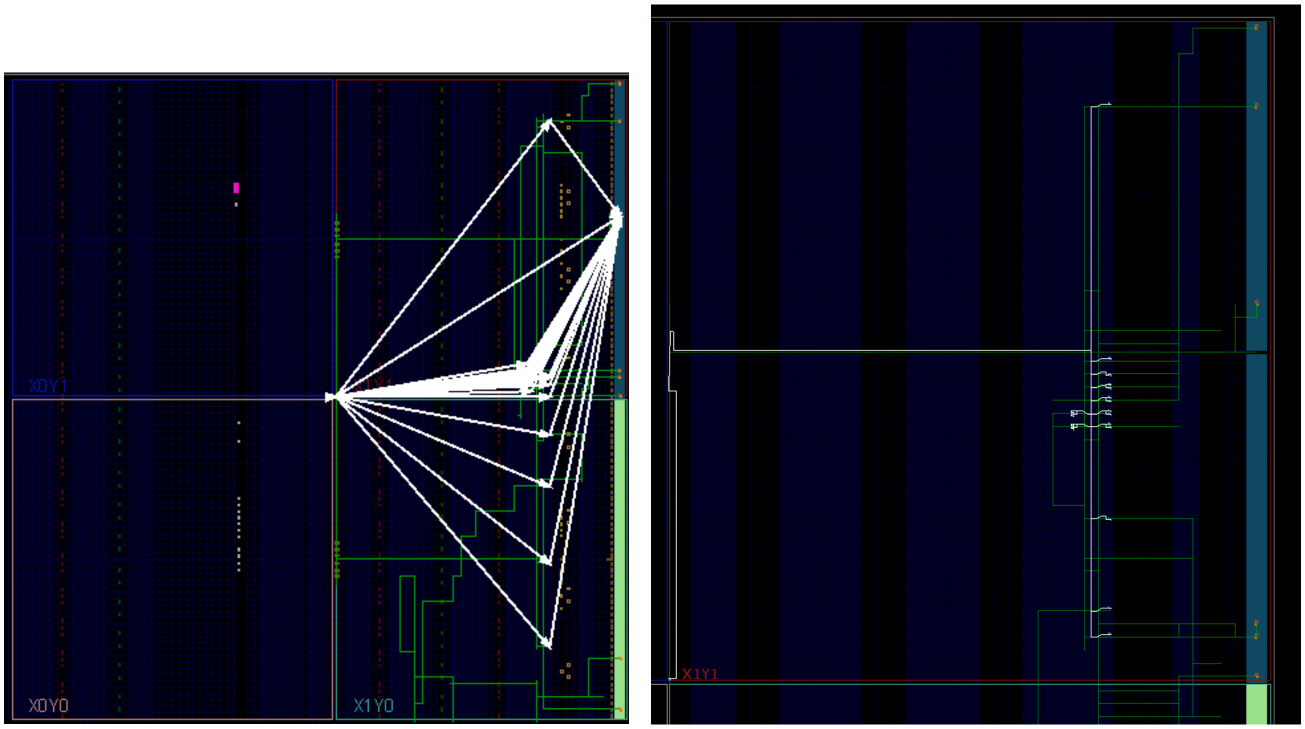


Figure 16. Clock nets

Generate the Bitstream Step 3

2-1. Generate the bitstream.

2-1-1. In the Flow Navigator, under Program and Debug, click **Generate Bitstream.**

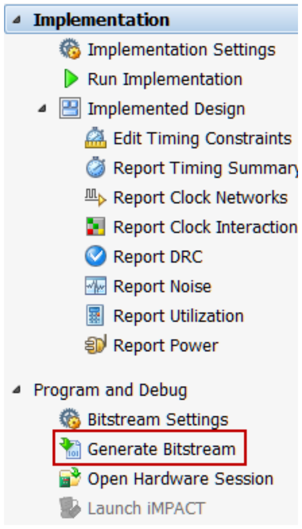


Figure 17, Generating the bitstream

2-1-2. Click **OK**. The `write_bitstream` command will be executed (you can verify it by looking in the Tcl console).

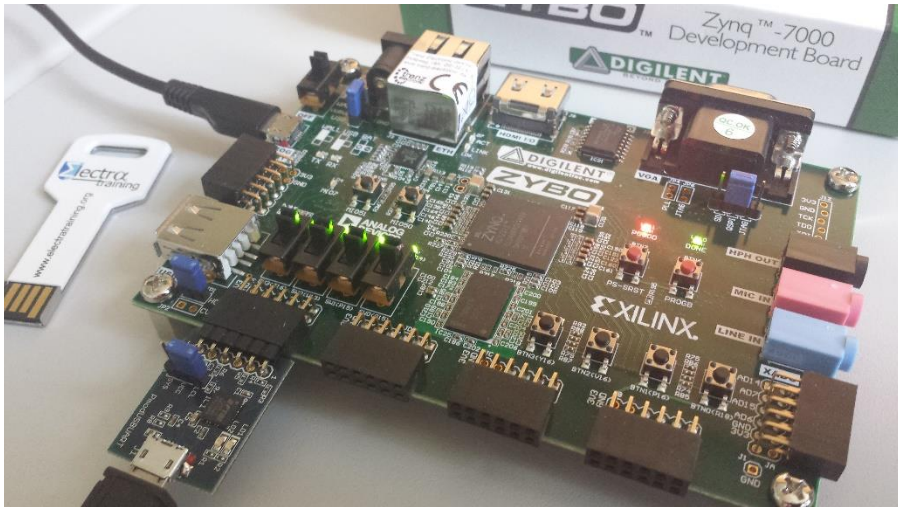
2-1-3. Click **Cancel** when the bitstream generation is completed.

Verify the Functionality Step 4

3-1. Connect a micro-usb cable in the PmodUSBUart module and insert the module into the top-row of the JE PMOD. Connect the board with another micro-usb cable and power it ON. Open a hardware session, and program the FPGA.

3-1-1. Connect a micro-USB cable between the PmodUSBUart module and the host PC USB port.

3-1-2. Plug-in the PmodUSBUart module into the top-row of the JE PMOD connector (below the slide-switch 3).



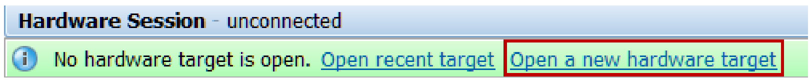
3-1-3. Make sure that another micro-USB cable is connected to the JTAG PROG connector (next to the power supply switch) and connect it to the PC. Make sure that the JP7 is set to select USB power. Turn ON the power.

3-1-4. Select the *Open Hardware Manager* option in the *Flow Navigator* and click **OK**.

The Hardware Manager window will open indicating "unconnected" status.

3-1-5. Click on the **Open target** or **Open a new hardware target** link. Then "*Open New Target…*".

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



Hardware Session - unconnected

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

Figure 18. Opening new hardware target

3-1-6. Click **Next** to see the Vivado CSE Server Name form.

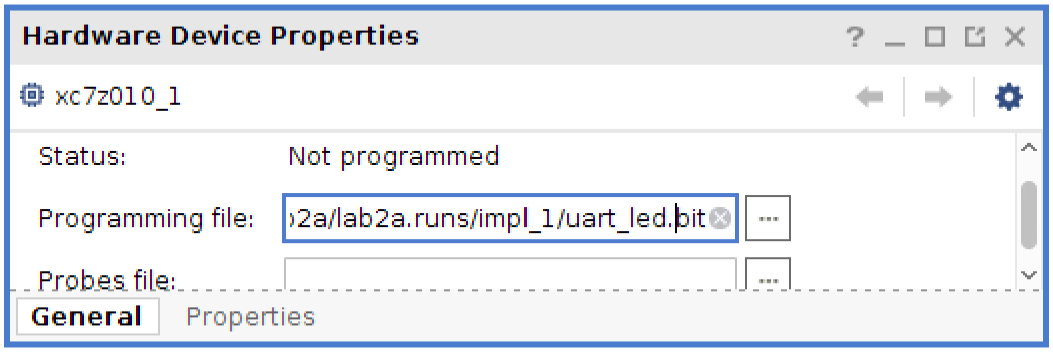
3-1-7. Click **Next** with the localhost port selected.

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

3-1-8. Click **Next** and **Finish**.

3-1-9. 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.

3-1-10. With the device selected in the *Hardware* section, verify that the **uart_led.bit** is selected as the "Programming file" in the *General* tab of the *Hardware Device Properties* section.


Hardware Device Properties

xc7z010_1

Status: Not programmed

Programming file: ›2a/lab2a.runs/impl_1/uart_led.bit

Probes file:

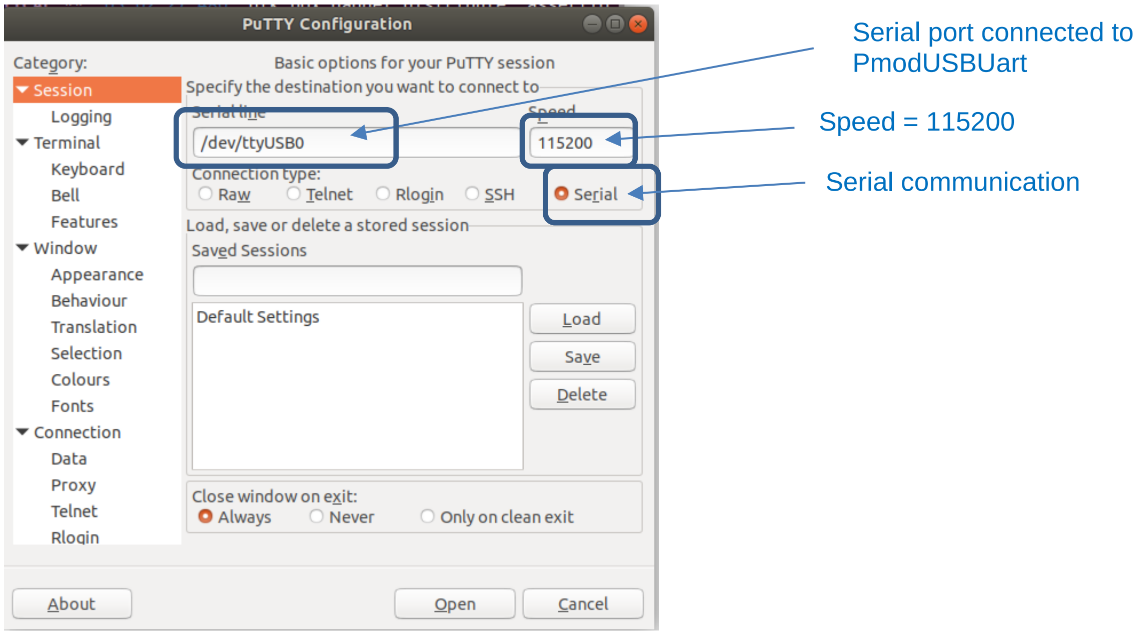
3-2. Start a terminal emulator. Set the COM port for 115200 baud rate communication. Program the FPGA and verify the functionality.

3-2-1. Start a terminal emulator program.

In linux you can use putty or minicom. The serial port is typically at /dev/ttyUSB[0-9]

3-2-2.

You can ask for connected serial ports using ls /dev/tty*. To check which one of the two /dev/tty* interfaces is the one connected to the PmodUSBUart module, you can plug and unplug the cable to see which ones are listed.


PuTTY Configuration

Basic options for your PuTTY session

Specify the destination you want to connect to

Serial line: /dev/ttyUSB0

Speed: 115200

Connection type: Raw Telnet Rlogin SSH [✓ Serial]

Close window on exit: [✓ Always] Never Only on clean exit

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

3-2-4. Right-click on the FPGA entry in the Hardware window and select Program Device…

3-2-5. 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.

3-2-6. Type in some characters in the terminal emulator window and see the corresponding ASCII equivalent bit pattern (least significant 4-bits) displayed on the LEDs.

3-2-7. Press and hold BTN0 and see the the upper four bits of the character.

3-2-8. When satisfied, close the terminal emulator program and power OFF the board.

3-2-9. Select **File > Close Hardware Manager**. Click **OK**.

3-2-10. Close the **Vivado** program by selecting **File > Exit** and click **OK**.