Implementation and Timing Analysis

PURPOSE - This lab continues through typical design steps. We continue by moving on to the Implementation and Timing Analysis steps of your design, which you started during the first lab.

Introduction

A short description of all the design steps was given in the Introduction section of the previous lab. You may want to revisit it. You will continue your design (which you Entered, Behaviorally Simulated, and Synthesized during the previous lab) with the Implementation and Timing Simulation steps.

Start Vivado by clicking on the Vivado icon on your desktop. Under Project click on Open Project. In the window that appears open the folder for the previous lab, lab1, and click on the project file (the entry with the Vivado icon in front of it.) The lab1 project should open. Now select File>Save Project As. In the window that appears name the project lab2 and click Save.

Alternatively, if you have Vivado Open you can go to Open Project, select the file corresponding to lab1, lab1.xpr, select File>Save Project As to bring up the Save Project As window, name the project lab2 and click Save.

Design Implementation

Design Implementation is the process of translating, mapping, placing, routing, and generating a Bit file for your design. In this step we optimize, place, and route the synthesized netlist generated in the previous lab using the available device resources on the FPGA.

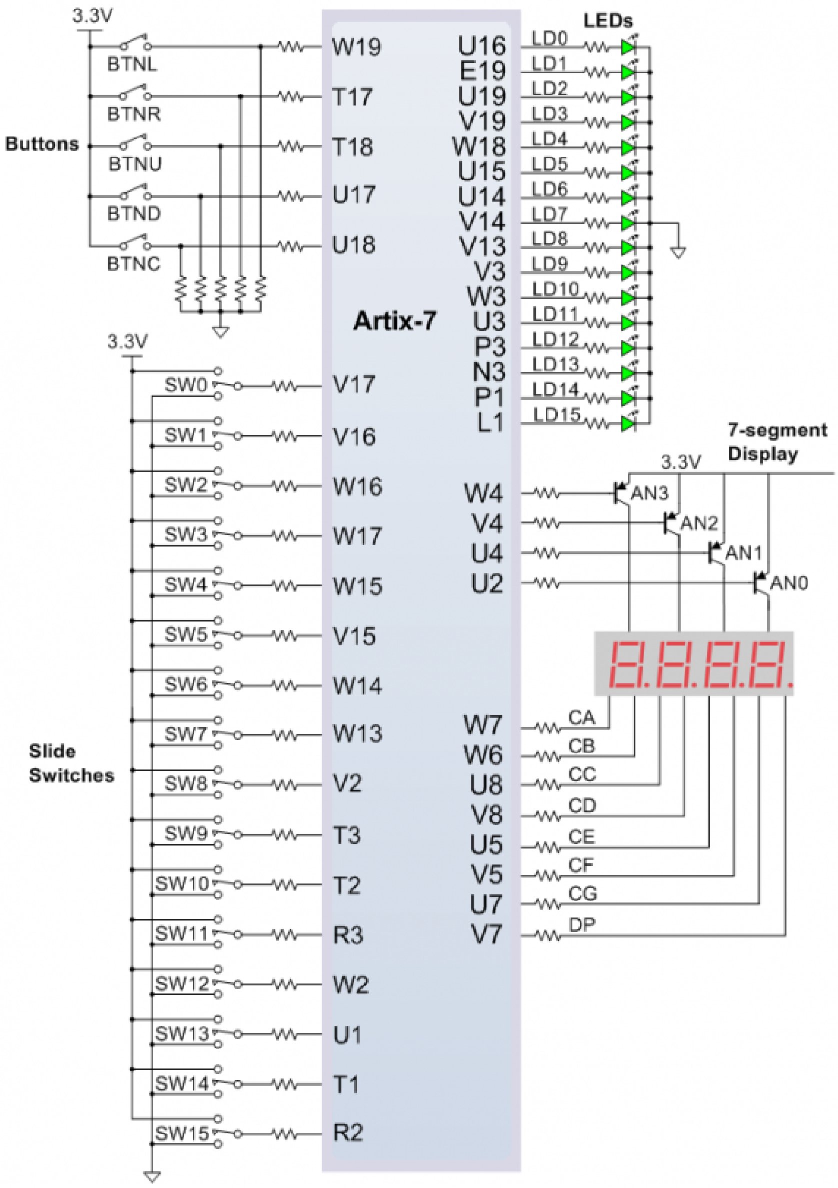
The Implementation process consists of a series of substeps which transform a logical netlist and constraints into a placed and routed design from which a bitstream is generated. It consists of the following sub-processes

- *Opt Design*: Optimizes the logical design to make it easier to fit onto the target Xilinx device.
- *Power Opt Design* (optional): Optimizes the design elements to reduce the power demands of the target Xilinx device.
- *Place Design*: Places the design onto the target Xilinx device.
- *Post-Place Power Opt Design* (optional): Additional optimization to reduce power after placement.
- *Post-Place Phys Opt Design* (optional): Optimizes logic and placement using estimated timing based on placement.
- *Route Design*: Routes the design onto the target Xilinx device.
- *Post-Route Phys Opt Design* (optional): Optimizes logic, placement, and routing using actual routed delays.

On completion of the Implementation process you should be ready to generate a bitstream.

Running the implementation

We will first create and add a constraints file to the project. The constraints file for this lab will be used to map the signals in the Verilog module `Add8` to available pins on the Artix-7 FPGA on the Basys 3 board. The connection to the FPGA I/O pins of the switches and leds on the Basys3 is shown in the following figure (taken from the Basys 3 reference manual)



Our constraints file will map the signals in the Verilog port list to the Artix-7 pins and the corresponding switches, buttons, and leds. To create this constraints file select **File > Add Sources** and in the window that appears select **Add or Create Constraints** and click **Next**. Now select **Create File**. This file should be an xdc file in the local directory, name it `adder8` and enter the following

```
# This file is the .xdc for the Basys3 rev B board used with lab1

# Switches
set_property PACKAGE_PIN V17 [get_ports {a[0]}]
       set_property IOSTANDARD LVCMOS33 [get_ports {a[0]}]
set_property PACKAGE_PIN V16 [get_ports {a[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {a[1]}]
set_property PACKAGE_PIN W16 [get_ports {a[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {a[2]}]
set_property PACKAGE_PIN W17 [get_ports {a[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {a[3]}]
set_property PACKAGE_PIN W15 [get_ports {a[4]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {a[4]}]
set_property PACKAGE_PIN V15 [get_ports {a[5]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {a[5]}]
set_property PACKAGE_PIN W14 [get_ports {a[6]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {a[6]}]
set_property PACKAGE_PIN W13 [get_ports {a[7]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {a[7]}]
set_property PACKAGE_PIN V2 [get_ports {b[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[0]}]
set_property PACKAGE_PIN T3 [get_ports {b[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[1]}]
set_property PACKAGE_PIN T2 [get_ports {b[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[2]}]
set_property PACKAGE_PIN R3 [get_ports {b[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[3]}]
set_property PACKAGE_PIN W2 [get_ports {b[4]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[4]}]
set_property PACKAGE_PIN U1 [get_ports {b[5]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[5]}]
set_property PACKAGE_PIN T1 [get_ports {b[6]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[6]}]
set_property PACKAGE_PIN R2 [get_ports {b[7]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {b[7]}]

# LEDs
set_property PACKAGE_PIN U16 [get_ports {s[0]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[0]}]
set_property PACKAGE_PIN E19 [get_ports {s[1]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[1]}]
set_property PACKAGE_PIN U19 [get_ports {s[2]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[2]}]
set_property PACKAGE_PIN V19 [get_ports {s[3]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[3]}]
set_property PACKAGE_PIN W18 [get_ports {s[4]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[4]}]
set_property PACKAGE_PIN U15 [get_ports {s[5]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[5]}]
set_property PACKAGE_PIN U14 [get_ports {s[6]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[6]}]
set_property PACKAGE_PIN V14 [get_ports {s[7]}]
      set_property IOSTANDARD LVCMOS33 [get_ports {s[7]}]

set_property PACKAGE_PIN V13 [get_ports {cout}]
      set_property IOSTANDARD LVCMOS33 [get_ports {cout}]

# Buttons
set_property PACKAGE_PIN U18 [get_ports cin]
      set_property IOSTANDARD LVCMOS33 [get_ports cin]
```

```
      # Timing Constraints
      create_clock -period 12.000 -name virtual_clock
      set_input_delay -clock [get_clocks virtual_clock] -add_delay 0.000
[get_ports -filter { NAME =~  "*" && DIRECTION == "IN" } ]
      set_output_delay -clock [get_clocks virtual_clock] -add_delay 0.000
[get_ports -filter { NAME =~  "*" && DIRECTION == "OUT" } ]
```

Notice how each two lines are associate an element from the port list of `adder8` to an FPGA pin. The first line identifies a particular pin with a port element and the second line sets the properties. When you have entered this file save it and return to the main Vivado window.

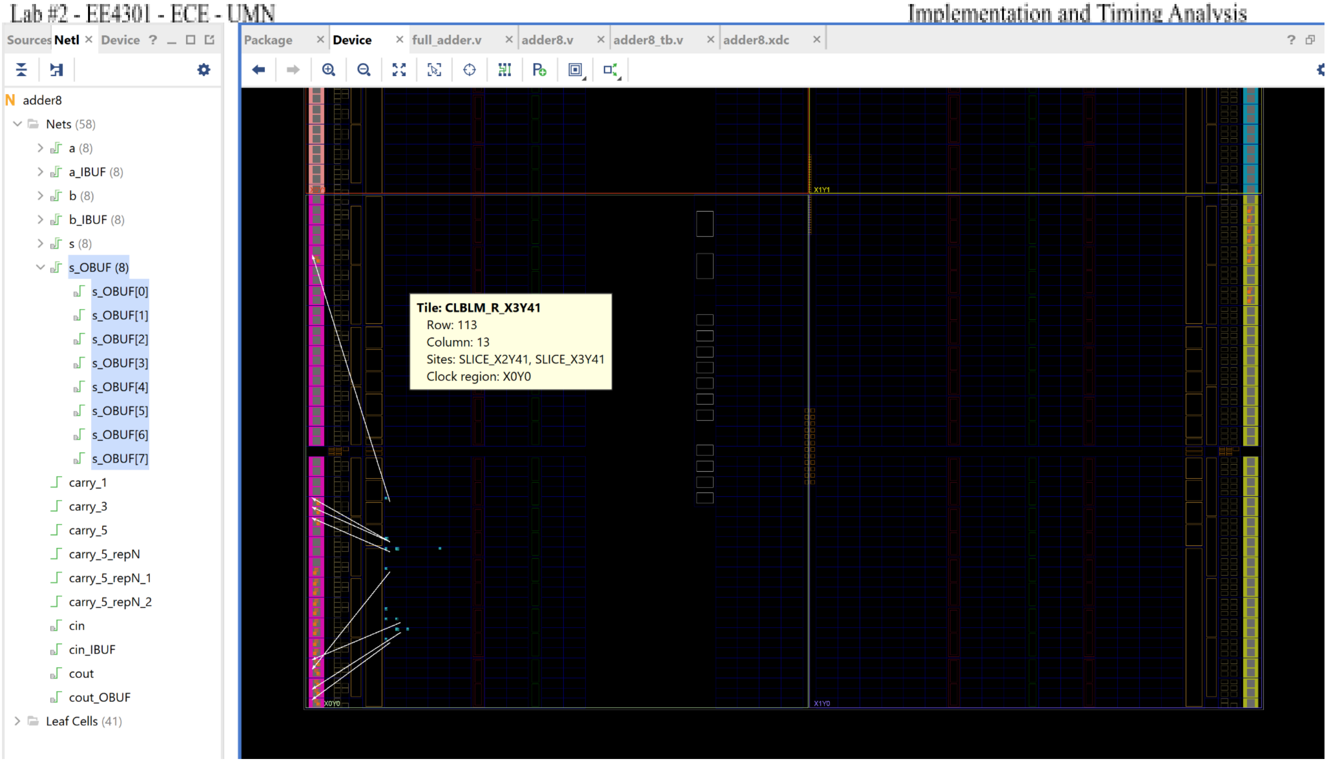
Now run the synthesis step by selecting **Run Synthesis** in the **Flow Navigator**. When synthesis is complete we are ready to proceed to implementation. Implementation is initiated by selecting Run Implementation and clicking **OK** in the pop-up window that appears when synthesis completes, clicking on Run Implementation under **Implementation** in the Design Flow window, or clicking on **Flow > Run Implementation**. Note that the implementation step may take some time to run. Be Patient!

Analyzing the Implementation

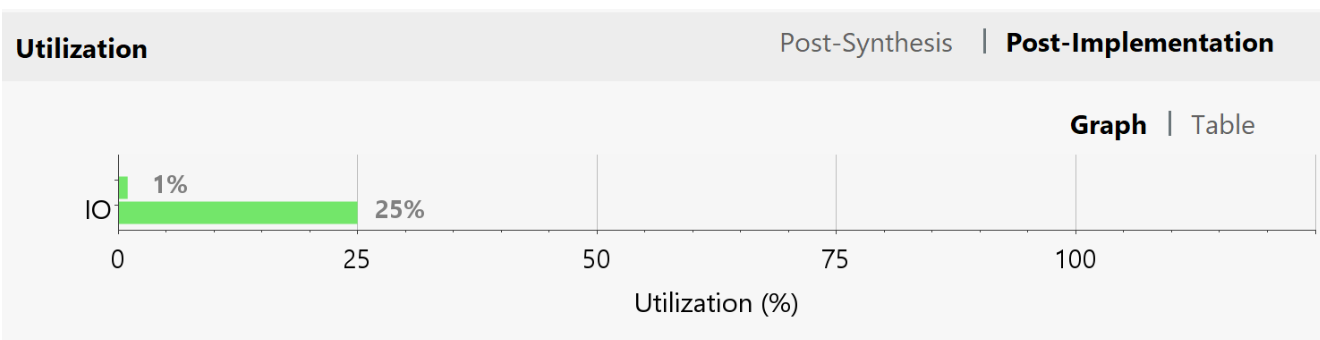
After running the implementation and opening the implementation Vivado provides a number of useful reports and other information that enables you to explore your design.

If you didn't select at the end of the implementation step you can open the implemented design by clicking on Open Implemented design under Implementation in the Design Flow pane. This will open the implemented design. Click on the Device View tab, the design will be opened. In the Netlist pane select one of the nets, Notice that the device has six clock regions, X0Y0, X0Y1 etc. The nets corresponding to your design are displayed in the device.

You can identify particular nets by clicking on that net in the netlist pane. Select s_OBUF(8) and you should see the figure zoom in on the X0Y0 clock region and identify the 8 connections for the output buffers on the sum bits to the output pins on the FPGA.



Now select the **Project Summary** tab. The project summary summarizes the result of the implementation, it contains several windows. The Project Settings window, the Synthesis window, the Implementation, the DRC Violations window, the Timing window, the Utilization window, and the Power Window. Under Utilization select Post-Implementation and click on Graph. This provides a graphical report of the resource utilization

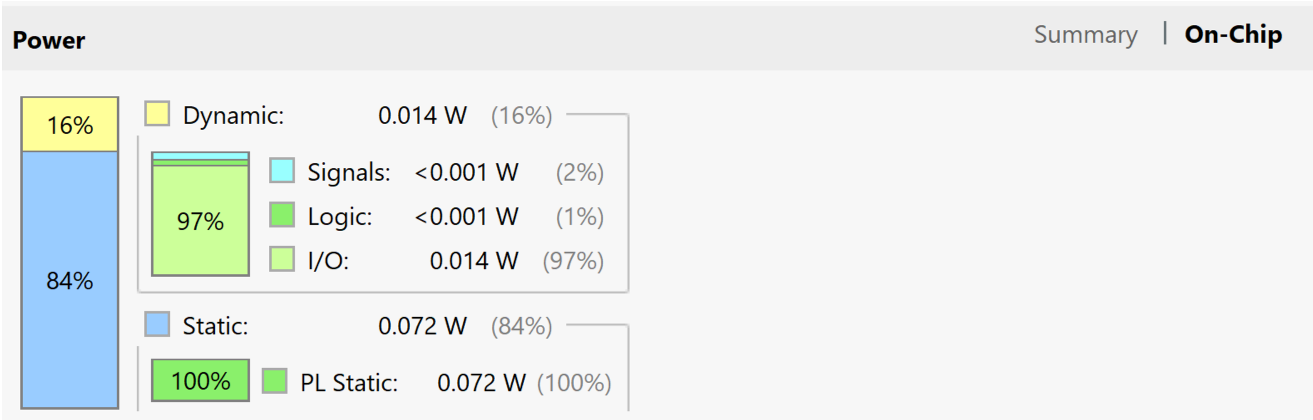


More detail provided by selecting table. Our design uses 8 of the 20800 LUTs and 26 of the 106 IO pins. Similarly we can view the estimated power related values for your design. Selecting On-Chip represents this graphically, selecting summary give a table.

Utilization Post-Synthesis | **Post-Implementation**

Graph | **Table**

Resource	Utilization	Available	Utilization %
LUT	12	20800	0.06
IO	26	106	24.53



These reports can be used to assess different implementations of our design.

Alternative Implementations

The implementation we used was the Vivado Default Implementation. To explore other possibilities, select **Implementation Setting** under Implementation in the Design Flow window. Under Strategy select **Area Explore** and click **OK**. A pop-up window **Create New Run** appears and asks if you want to save the completed run, click **OK** to create a new implementation. A new pop-up window, **Create Run**, appears. Use the default name `impl_2` and click **OK**. We are now ready to explore this alternative implementation.

We run this alternative implementation as before. When it completes examine the reports, has anything changed?

Timing Simulation

Select **Run Simulation** and select **Run Post-Implementation Timing Simulation**

SUMMARY -- This laboratory taught you how to perform the Implementation design step, analyze the placed and routed design, and explore alternative implementation strategies.

Lab Notebook deliverables:

- Brief description of what you did in the lab, what steps were followed
- Discussion of results:
    a. Attach snippet of the reports of the tool.
        i. Timing report
        ii. Utilization report from implementation
        iii. Power report

    NOTE: Only attach relevant sections, not the entire report.
- Summary or Conclusion:
    a. Were all design goals met?
    b. What were the difficulties encountered?
    c. Any improvement you would like to make time permitting? Etc.…
- All files must be labeled (ex. Figure 1: Simulation of eight-bit adder…). You should also highlight more interesting parts of your attachments.

REFERENCES

[1] Xilinx, *Vivado Design Suite User Guide: Using the Vivado IDE*, UG893(v.2016.2) June 8, 2016.

[2] Xilinx, *Vivado Design Suite User Guide: Using Contraints, UG903* (v.2016.2) June 8, 2016.

[3] Xilinx, *Vivado Design Suite User Guide: Implementation*, UG904 (v.2016.2) June 8, 2016.