ECE 3300L

Lab 2: Majority Encoder

Instructor: Dr. Mohamed Aly

Group A

Sami Elias	012546378	samielias@cpp.edu
Joseph Popoviciu	014357772	jdpopoviciu@cpp.edu

Objective:

The objective is to design a majority encoder using structural modeling with data flow. Furthermore, we incorporate the majority encoder with our FA and FA_nbits.

Procedure:

We first create a 3-bit majority encoder which calculates our carry out for a 1-bit full adder. Once we had the module for the majority encoder, we instantiated it within the 1-bit full adder which was already instantiated within the n-bit full adder.

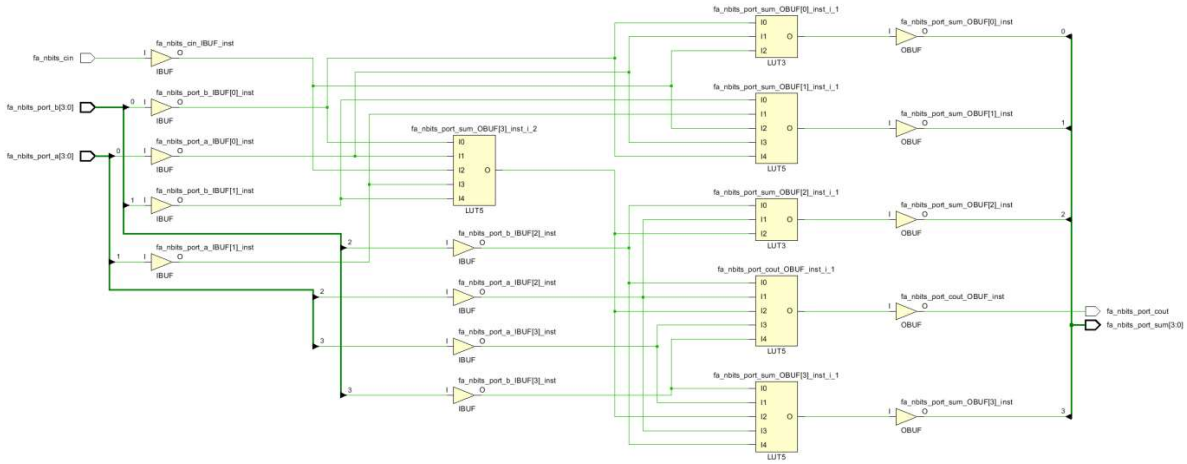

Figure 1- 4-bit Full Adder Schematic

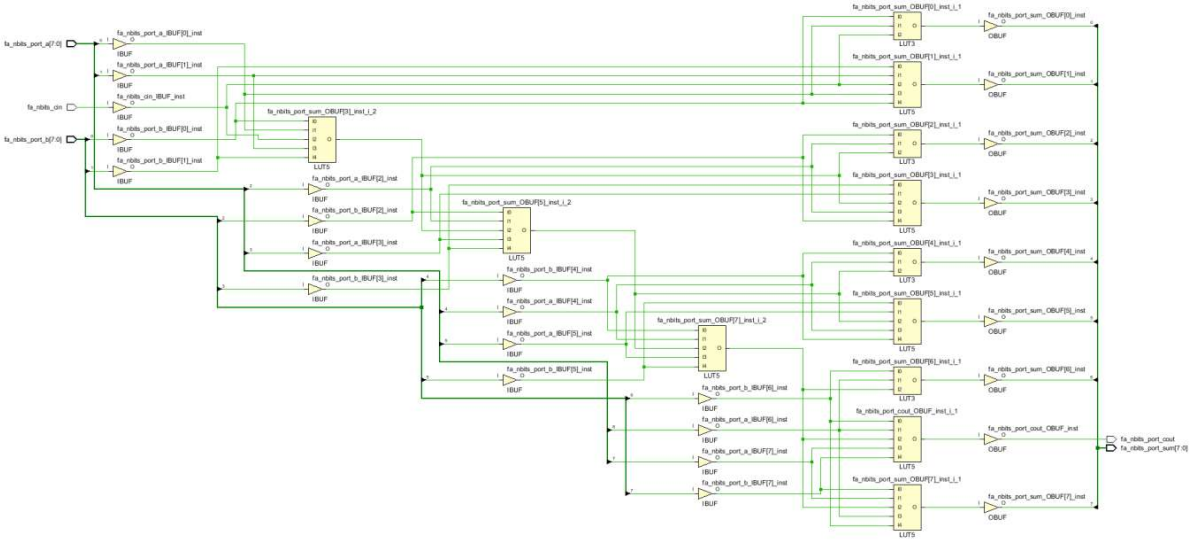

Figure 2- 8-bit Full Adder Schematic

After ensuring our code was correct with a testbench and simulation. Once the simulation was outputting the expected bits we began to synthesize, implement, and generate a bitstream. As a result, we obtained schematics shown in figures 1 and 2 above.

Results:

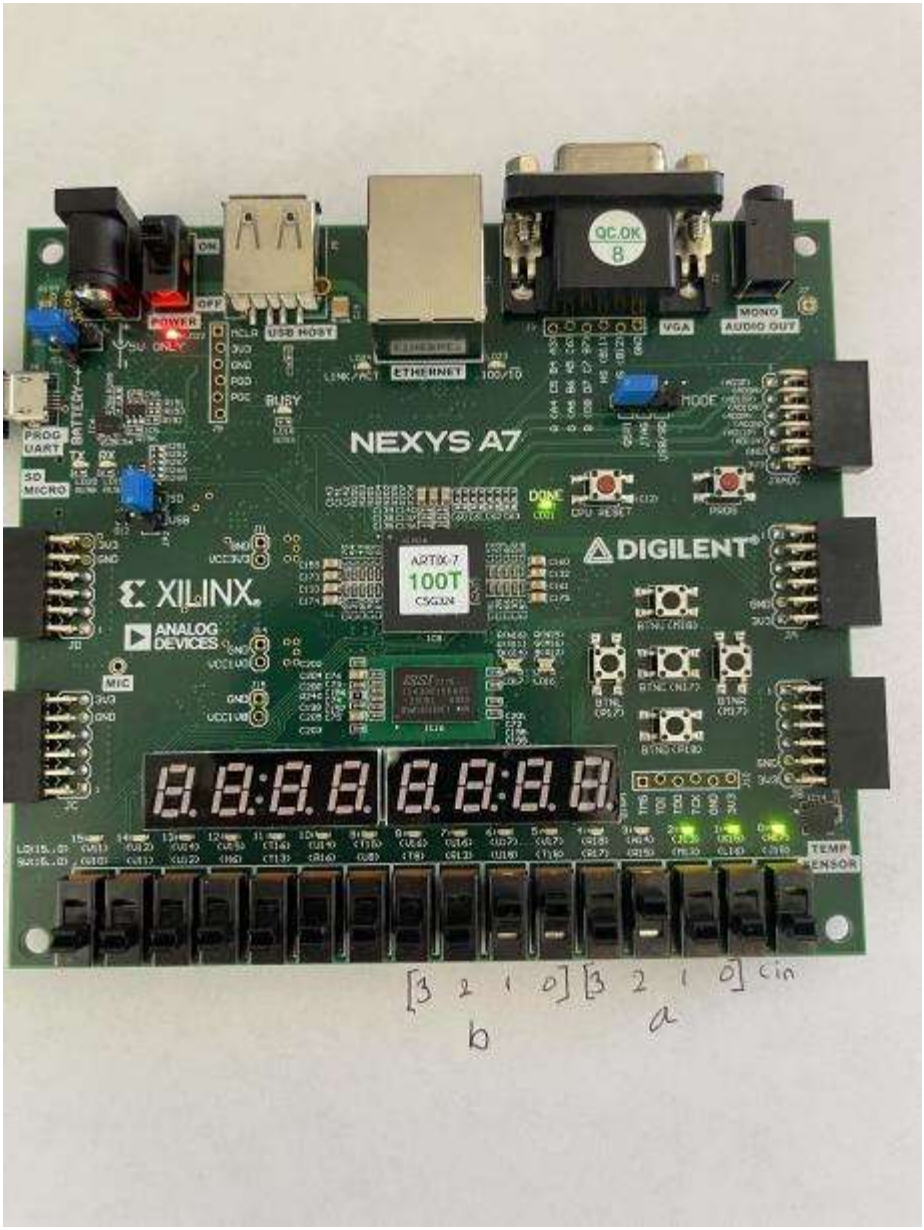

Figure 3- FPGA Board Displaying 4-bit addition

Image 3 shows 0100+0011, with the sum being 0111.

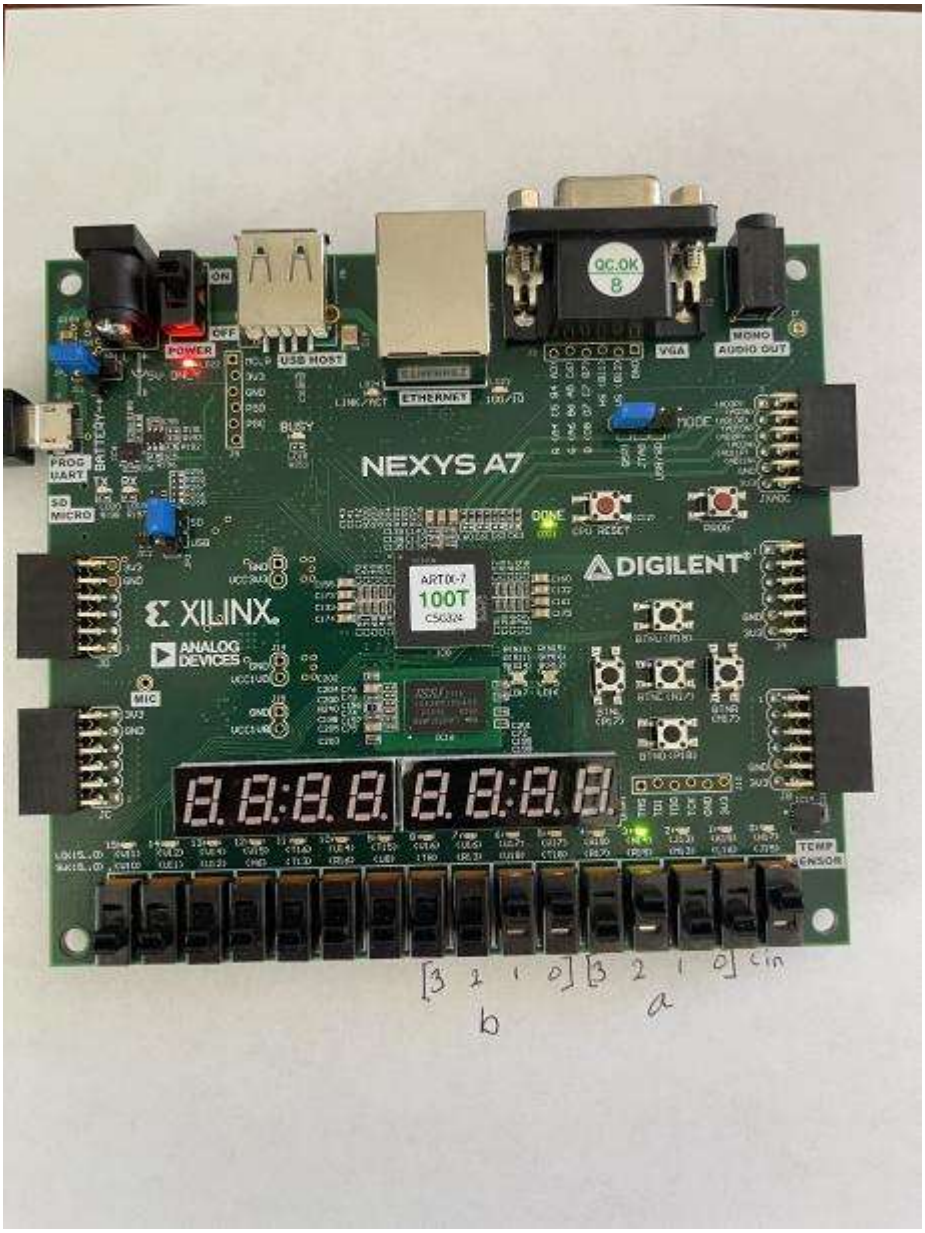

Figure 4- FPGA Board Displaying 4-bit addition

Image 4 shows the same as image 3, but there is a carry in of 1 which results with the sum being 1000.

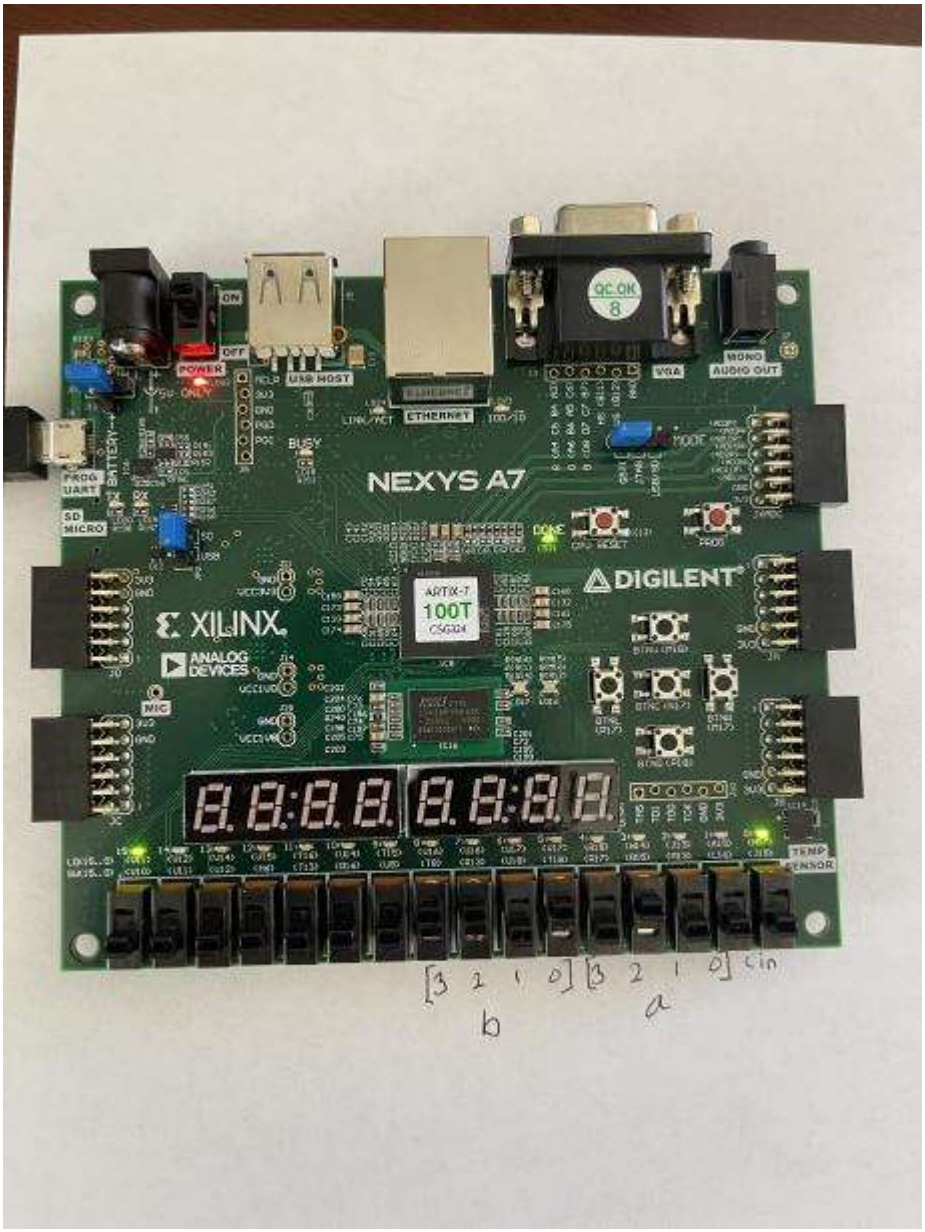

Figure 5- FPGA Board Displaying 4-bit addition

In figure 5 we test the carry out. The LED on the far-left side of the FPGA board represents the carry out. To test our carry out we added 1011 with 0100 which resulted with a sum of 0001 and a carry out of 1.

Test bench:

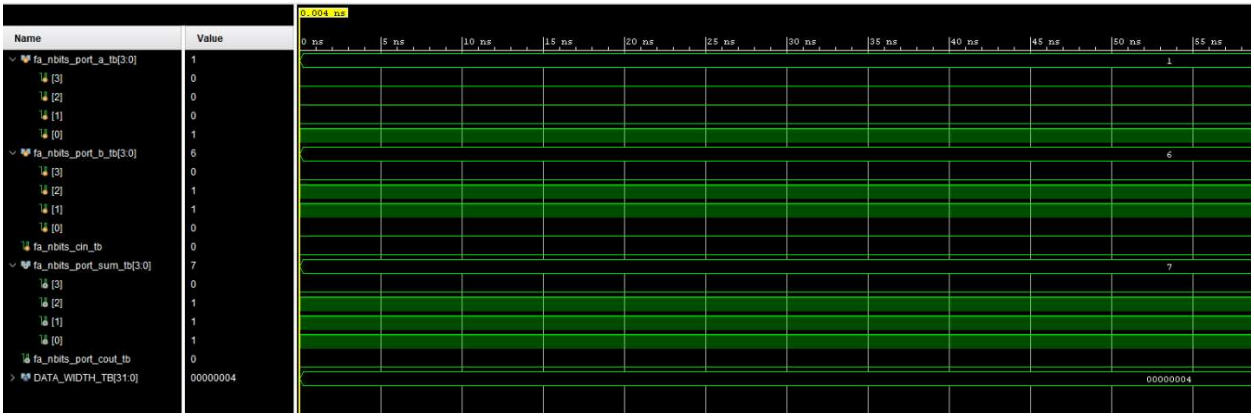

Figure 6 - Vivado Behavioral Testbench

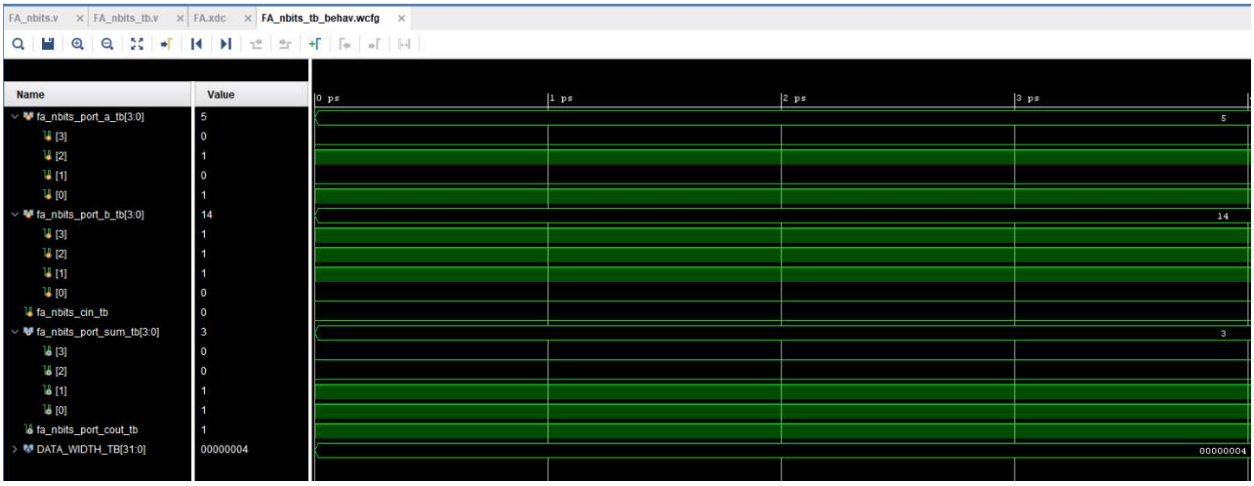

Figure 7 - Vivado Behavioral Testbench

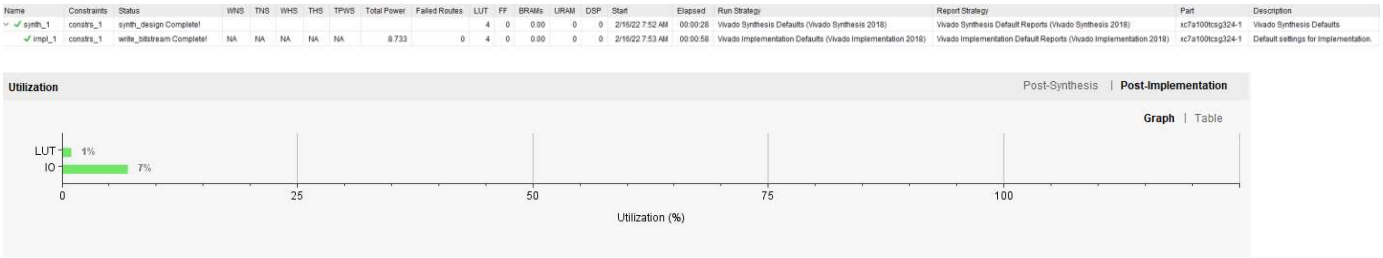

Figure 8 - FPGA Utilization

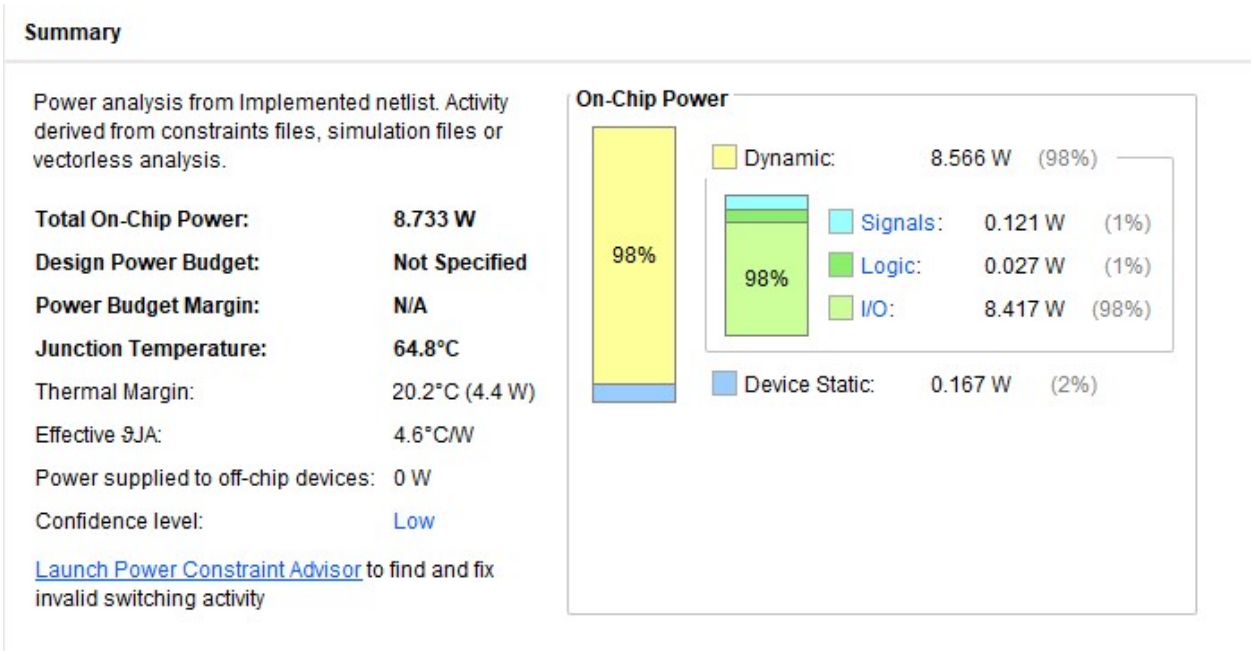

Figure 9 - FPGA Power Summary

Contributions:

Joseph Popoviciu:

I primarily worked on the code, programmed the FPGA board, and created the video demo. I also helped with the lab report.

Sami Elias:

I assisted with the code, wrote the main points of the report, and did the majority of the PowerPoint.