CompArch Lab 0

Jacob Riedel, Jennifer Vaccaro, Jennifer Wei

October 2 2015

1 Introduction

Lab 0 was an introduction to FPGAs, where we took the one-bit Full Adder from HW2 to make a four-bit Full Adder, tested it on ModelSim, and loaded the tested four-bit Full Adder design onto the FPGA board using the provided lab wrapper to interface between our code and the Zybo board.

2 Waveforms

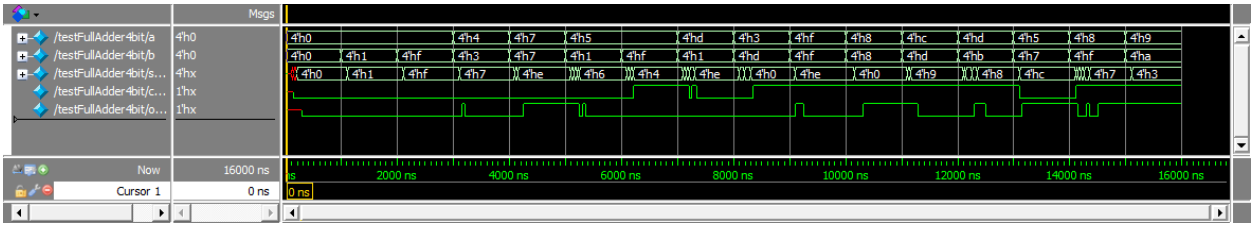

Figure 1: Four-Bit Full Adder Waveform (with added delay)

The waveform (Fig. 1) shows the full adder with delays of 50 units of time for each gate in the adder. The worst case delay would occur when there is an overflow as it has to cascade from the first adder to the fourth adder and also go through the overflow logic as well.

The worst case of each of our adders is a delay of $150ns$ (due to all the gates), and the worst case of the overflow logic is a delay of $100ns$ ($50ns$ for each gate with two that are in parallel). Thus, with four adders and the overflow, the worst case delay based on the statements above would be $700ns$ if we defined the delay for each gate to be $50ns$.

However, by delaying the display in our ModelSim test by $1000ns$, we avoid inaccuracies due to delays. In addition, rather than show each input into the full adder, we show the 4-bit wires. The values of the wires are expressed in Hex 1.

3 Test Case Strategy

A_3	B_3	S_3	Carryout	Overflow
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

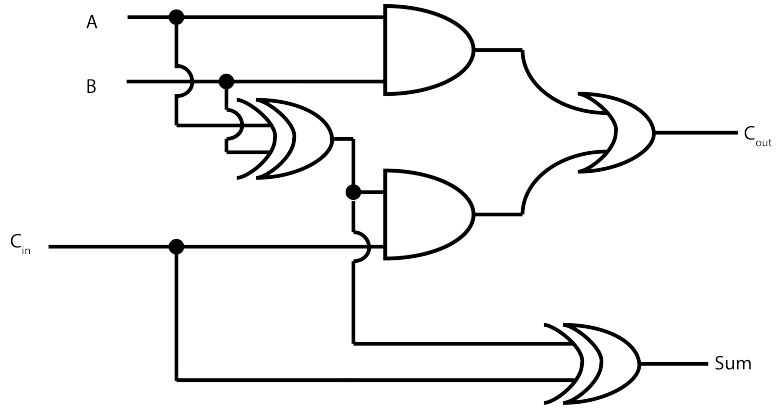

Figure 2: One-bit Full Adder Logic Diagram

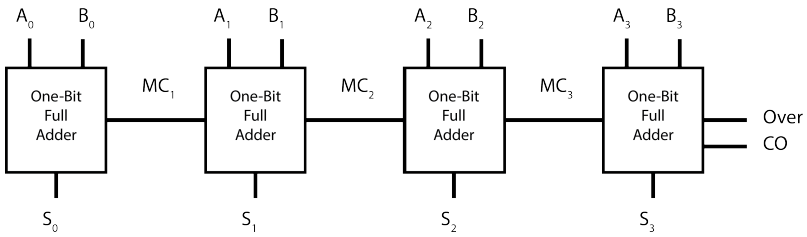

Figure 3: Four-bit Full Adder Block Diagram

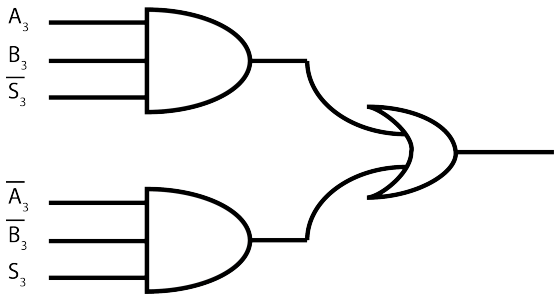

Figure 4: Four-bit Full Adder Logic Diagram for Overflow

Before going into the code, we first drew the four-bit full adder (by chaining four one-bit full adders (Fig. 2)) and then drew out the logic for the carryout and overflow values (Fig. 4). In order to determine the logic, we drew out a truth table (Fig. 3) and generated a function which yielded those values. The overflow value O_f was determined by

$$O_f = A_3 B_3 \bar{S_3} + \bar{A_3} \bar{B_3} S_3$$

Informed by our truth table and mathematical identities, we selected a 16 element subset to represent the 256 possible inputs. We demonstrated the additive identity 0 summed with itself, negative and positive numbers, and then we gave multiple examples each of sums of two negative numbers, sums of two positive numbers, sums of negative and positive numbers, and sums which yielded a positive and negative overflow. Because we worked with two's complement negative numbers, we prioritized the accuracy of the sign of our sum.

4 Test Case Failures

When we first ran our test module, it failed to produce outputs. Our inputs displayed correctly, but all outputs were listed as "x". As we located errors in our code, we realized that we had failed to add a delay time between computations and displays. However, once we added delays in, our tests ran as we had expected. Throughout the test cases we chose, we did not find any unexpected values. This was because we had worked through the logic manually so that our implementation was robust.

5 FPGA Board Testing Summary

To test our 4-bit Full Adder, we uploaded our adder code and the wrapper to our FPGA board. After the code was successfully uploaded, we ran through out test bench with the physical switches on the board. By confirming that the board's output was the same as our test bench, we proved that our implementation of the 4-bit full adder was correct. We then took pictures of the board during one of the test case run through.

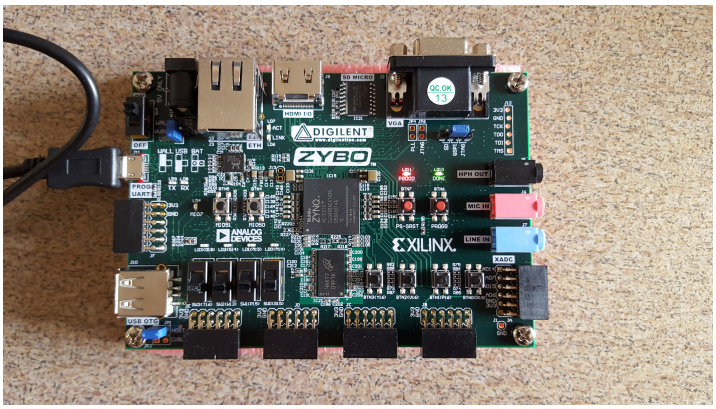

Figure 5: The Input +7 (0111)

Our test was the addition of +7 and +7, which we expected to return +14 if unsigned, but instead returned -2 due to overflow.

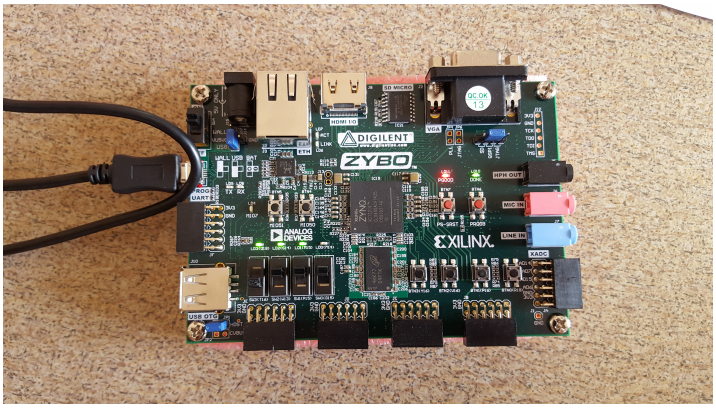

Figure 6: Sum of A and B returns -2 (1110)

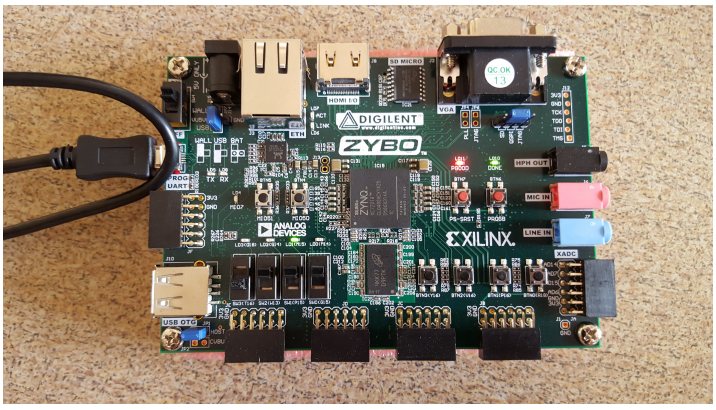

Figure 7: LED1 (Overflow) equals 1, LED0 (Carryout) equals 0

6 Summary Statistics

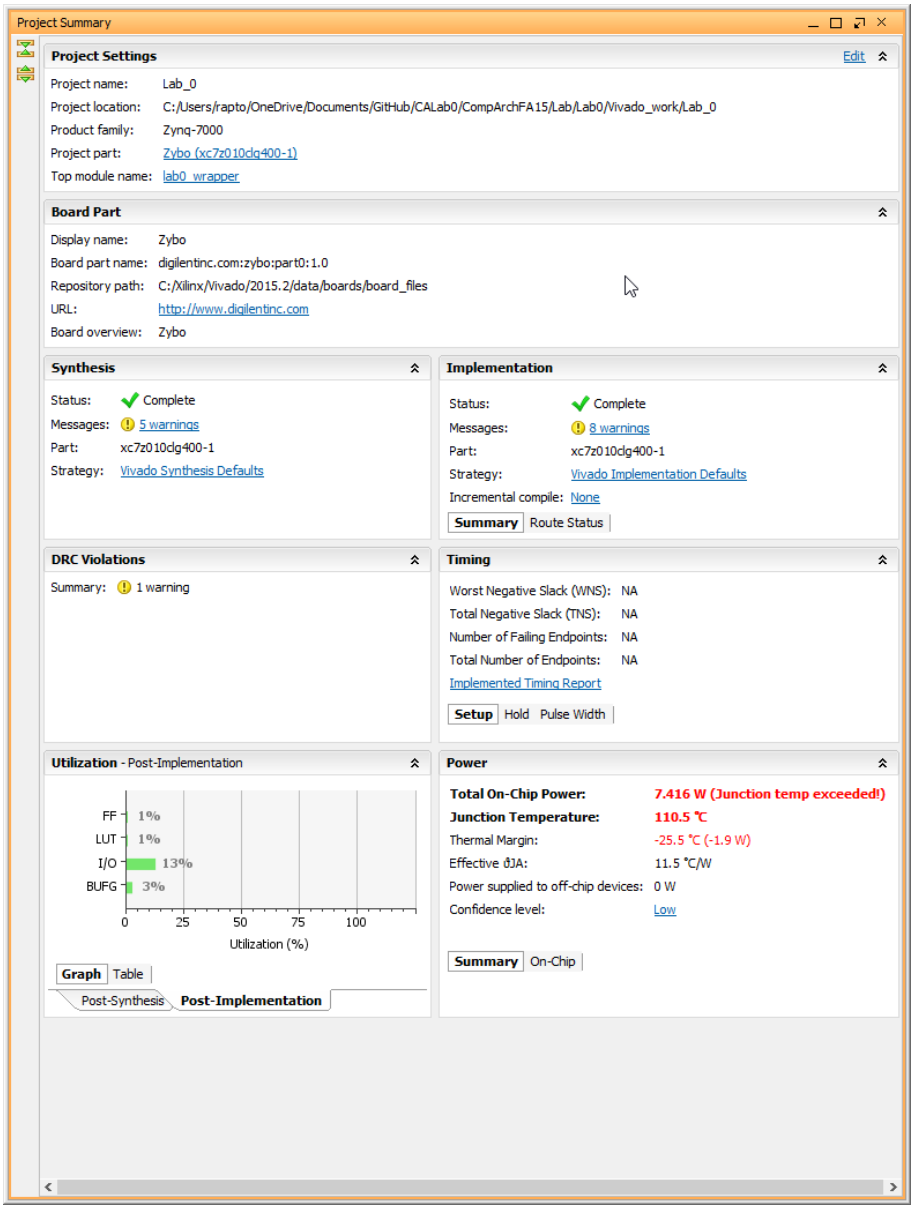

Figure 8: Project Summary Provided by Vivado

As shown in 8, the FPGA implementation used 1% flip-flops, 1% lookup tables, 13% input/output, and 3% buffers. The power section looks questionable with the red text, but we did not look into it this time around.

7 Acknowledgments

Thank you to the NINJAs (Sidd, Kyle, and Radmer) for answering all our questions about this lab and helping us better understand it.