Digital Design Lab 6

Tara Moses

Section L001

November 10, 2015

tsmoses@uark.edu

**Abstract**

The purpose of this lab was to write VHDL code that implements four different circuits. Four circuits (a combinational function, an adder/subtractor, a decoder, and a multiplexer) were created using VHDL code and simulated in Quartus II.

**Introduction**

VHDL is a language that is used to create circuits with binary input. It is widely used among commercial production companies.

A decoder is a circuit that takes in an input En and an input bus W and, depending on whether En is a 0 or 1, decodes W. The decoded W is outputted as a bus. The bus sizes of W and Y depend on what type of decoder it is. In this lab, a 2-to-4 decoder was implemented, meaning that the input bus W was a 2-bit bus and Y, the output, was a 4-bit bus.

A multiplexer is a circuit that takes in two input buses: input bus W, which holds values that might possibly be the output, and input bus S, which essentially decides which part of W to output. The multiplexer outputs the value of one of the parts of W; which part of W depends on S. The multiplexer (or mux) implemented in this lab was a 4-to-1 mux. This means that the input bus W was a 4-bit number, and the output was a 1-bit number (0 or 1).

This lab was designed to familiarize the user with writing VHDL code for various functions. The four functions coded for were a simple combinational function, a 5-bit adder/subtractor, a 2-to-4 decoder, and a 4-to-1 mux. All code was written and compiled successfully, and simulation results met expectations.

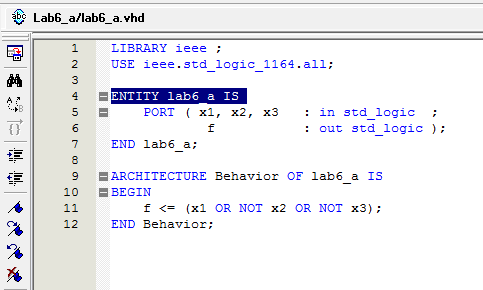
**Design and Implementation**

The first part of the lab involved coding a VHDL program that represented the combinational equation

. This equation was first simplified to

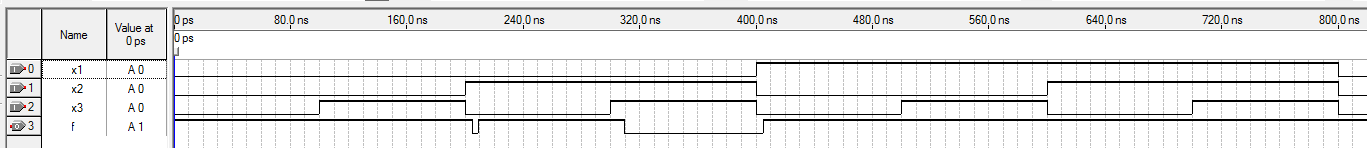
, and after writing a truth table for it, the function was rewritten as

. After the function was written this way, VHDL code was written. The VHDL code is shown in Figure 1 below.



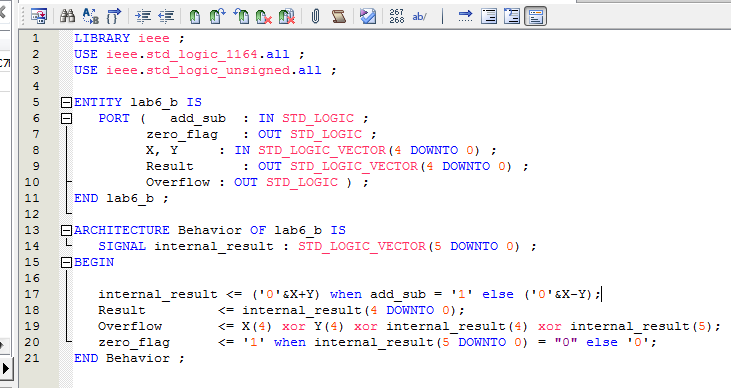
**Figure 1**: VHDL code for a combinational function.

The code was compiled, and a simulation was run to test the code’s functionality. Simulation results are shown in Figure 2 below.



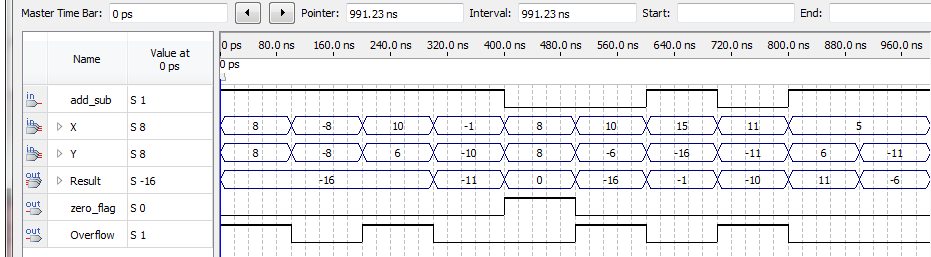
**Figure 2**: Simulation results for a combinational circuit.

For the second part of the lab, VHDL code was written to implement a 5-bit adder/subtractor. This add/sub added or subtracted numbers between -15 and +16 (because signed 5-bit numbers can only go that low/high). VHDL code for the circuit is shown in Figure 3 below.



**Figure 3**: VHDL code for adder/sub.

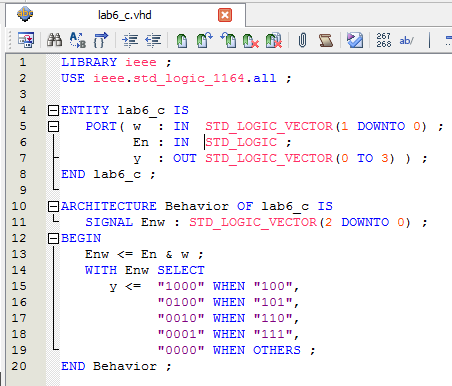
Once the code compiled correctly, it was tested using a simulation in Quartus II. The simulation results are shown in Figure 4 below.



**Figure 4**: Simulation results for adder/sub.

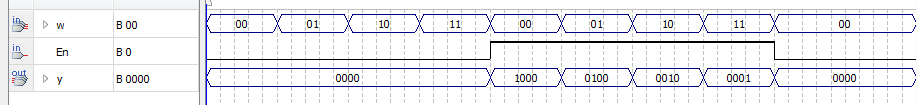
When input add\_sub equals 1, X and Y are added together, and vice versa when add\_sub equals 0. When output Overflow is 1, the sum or difference between X and Y was out of range (that is, either as low as -15 or as high as +16) For example, from 0 to 100 ns, the add/sub outputted the sum of X and Y, which was +16. The Result for that time frame, though, was -16 due to overflow. When output zero\_flag is 1, Result must have been 0.

For the third part of the lab, VHDL code was written to implement a 2-to-4 decoder. VHDL code for the decoder is shown in Figure 5 below.



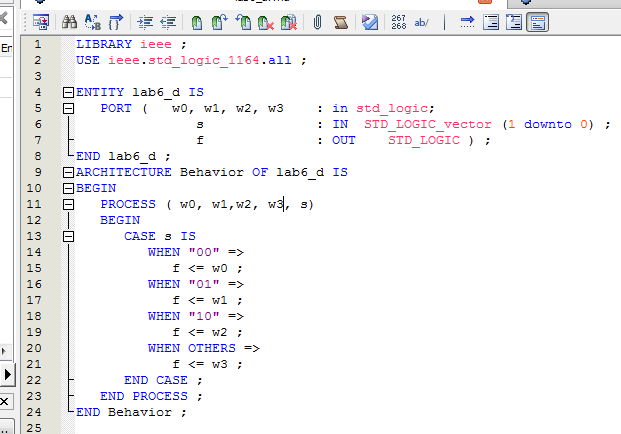
**Figure 5**: VHDL code for a 2-to-4 decoder.

The VHDL code was compiled, and a simulation was run. The results of the simulation are shown in Figure 6 below.



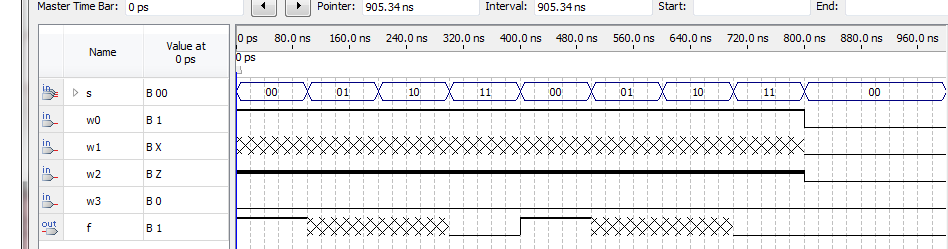
**Figure 6**: Simulation results for 2-to-4 decoder.

The fourth part of the lab involved writing VHDL code for a 4-to-1 multiplexer. The VHDL code for this multiplexer is shown in Figure 7 below.



**Figure 7**: VHDL code for 4-to-1 multiplexer.

The VHDL code was compiled and then tested using a simulation. Simulation results are shown in Figure 8 below.



**Figure 8**: Simulation results for 4-to-1 multiplexer.

The output of f was supposed to show high impedance (a bold line) from the time ranges of 200-300 ns and 600-700 ns. However, Quartus II doesn’t show high impedance from the output, but an unknown range (the grey x’s).

**Results**

All four instances of VHDL code did what they were supposed to do. Each simulation gave the expected results except for the adder/subtractor, which gave incorrect Result outputs due to overflow, and the multiplexer, which didn’t correctly show high impedance when it should have.

**Conclusion**

The lab was designed to give students a hands-on experience with writing VHDL code for various simple functions. Using Quartus II, four circuits were implemented using VHDL and tested with simulations. One large source of error was the fact that the adder/subtractor, while working with binary signed integers, could not account for overflow. This resulted with the sum of two positive numbers being negative and vice versa. Overall, the project was successful. The lab took about three hours to complete.