Lab 06

Adder Design Tradeoffs

ECE 380-002

University of Alabama

Yichen Huang

Thomas Dillman

2019/10/7

**Introduction**

In this lab, we use the Quartus prime software package to design and simulate one 4-bit subtractor and one 4-bit adder. The requirements for the lab consist of completing Quartus Prime designs, downloading designs to the Altera DE 1 board, printing circuit diagrams, VHDL files, simulating results, and the laboratory report.

**Procedure**

1. Prelab
   1. Design A

In the design A, we created a 4-bit subtractor using structure VHDL. The circuit has two 4-bit data input A and B, a 4-bit output S, a carry-out bit and an overflow flag. We create the circuit by using two VHDL files, fulladd.vhd and sub4b.vhd.

A screenshot of a social media post

Description automatically generatedAfter coding, we compile the project and run the simulation by test vectors.

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

* 1. Design B

A screenshot of a cell phone

Description automatically generatedA screenshot of a social media post

Description automatically generatedIn the design B, we design a 4-bit adder by carry look ahead adders. We implemented the design by dataflow VHDL method to the file. This 4-bit adder have two 4-bit inputs, a 4-bit data outputs, and a bit for carryout. After implementation and compiling process, we run the stimulation, the result is same as we aspect.

* 1. Design C

A screenshot of a social media post

Description automatically generatedIn the design C, we use structure VHDL code to display seven segment LED to display 2 input and 1 output.

A screenshot of a social media post

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a social media post

Description automatically generated

* 1. Testing vectors

For the testing part, this lab is focusing on 4-bits adder circuit and 4-bits subtractor. In general, we have three outputs: a 4-bits output for addition or subtraction, a carry-out bit and a bit indicate overflow. For the test case, I will choose the case which will lead a correct 4-bits output with a carryout bit and without carryout bit(Like B - 2 and B – E) and other case which will lead overflow with carryout bit and without carryout bit (Like 7 – E and C – 5)

1. During the lab
   1. Design A

A screenshot of a cell phone

Description automatically generatedIn lab for design A, we set the switches on the board and compiled again. After success, we set the DE1 board and do the test again, the result is same as simulated result.

A screenshot of a social media post

Description automatically generated

* 1. Design B

In the design B, we repeat the same process in design A. The result is also same as we expected and simulated result.

A screenshot of a social media post

Description automatically generatedA screenshot of a computer

Description automatically generated

* 1. Design C

For the design C, we also repeat the same process in design B. After setting, we run the test on the board. The LED display the number of inputs and output. The number is also correct our simulation.

A screenshot of a social media post

Description automatically generated

**A screenshot of a social media post

Description automatically generated**

**Result**

The result of design A, B and C are the same as the waveform file in the prelab. For the design C the LED display the number of input and output are also correct.

A close up of text on a white background

Description automatically generated

**Conclusion**

In this lab, we learned how to create adder and subtractor in two different ways structure VHDL and dataflow VHDL. We also understand how to create carry-out look ahead adder. In the design C, we learned how to use led to display varibles.