EELE 466

**Lab 2 Report:**

Implementing an 8-bit Carry Look Ahead Adder/ Subtractor

Austin Halverson,

Johnny Gaddis

January 30, 2018

1. Summary

The purpose of this lab will be to implement an 8-bit carry look ahead Adder and Subtractor. Ripple carry adders are slow adders and we can speed up the addition process by determining what the carries will be ahead of time.

2.0 Lab Design Components

The logic design for this adder is essential to the functionality of the code. Steps were taken to analyze the logic expressions at each step of the assignment. This ensures that as we move forward on the assignment and troubleshooting, we will not have to test more than one problem at a time. In order to design this 8-bit carry look ahead adder, we must create the logic for the full adder. The logic for the full adder is shown in figure 1 below.

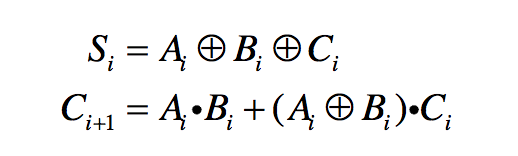
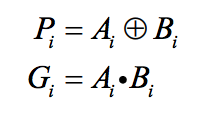


Figure 1: Full Adder Logic



We know that,

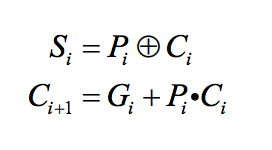
 We call Pi the carry propagate. If Pi = 1 then Ci will propagate to Ci+1 and Gi is the carry generate since if Gi = 1 then a carry is generated regardless of what Ci is. Therefore the equations in figure 1 can be rewritten as:

Figure 2: Simplified Full Adder Logic

Next, we use the equation from figure 1 to create a 4-bit carry look ahead component in VHDL.

After we wrote the 4-bit carry look ahead adder, we created an 8-bit carry look ahead adder by instantiating the two 4-bit carry look ahead adders. It was essential that we implemented some method for dealing with a subtraction signal. This was accomplished by creating a subtraction signal bit that determined whether we add or subtract. This will be handled with one bit that is either a 1 or a 0. If the subtraction signal were a 1 then the logic would do subtraction and if the signal is a 0 then addition was initiated. Then, we XOR the Subtraction bit with all the B inputs and then fed ‘Subtraction’ into Cin of the Full Adder 0.

We used a Mathlab test bench to verify your design using the HDL Verifier Toolbox for the same eight cases we used it for in Lab 1. We compiled each design in Quartus and found the value of Fmax for the ripple carry adder from Lab 1 and the carry look ahead adder from Lab 2 using the Time Quest Timing Analyzer under the Slow Model. The results of the timing for Labs 1 and 2 are shown below in figures 3 and 4 below.

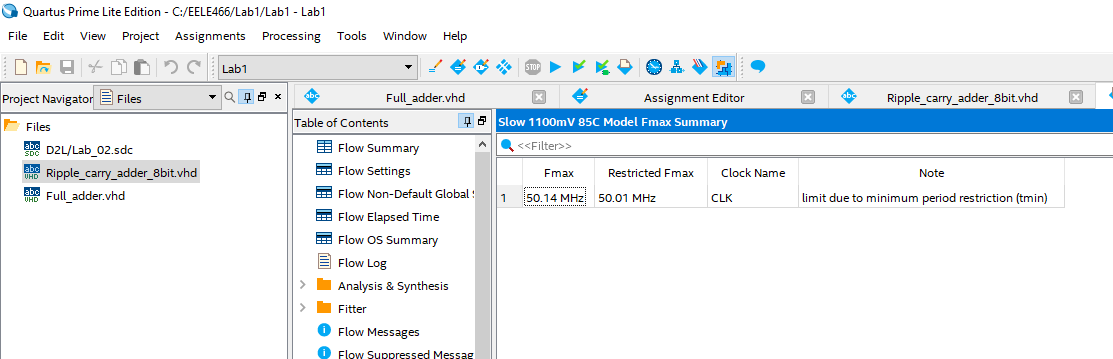


Figure 3: Timing for ripple carry adder from Lab 1

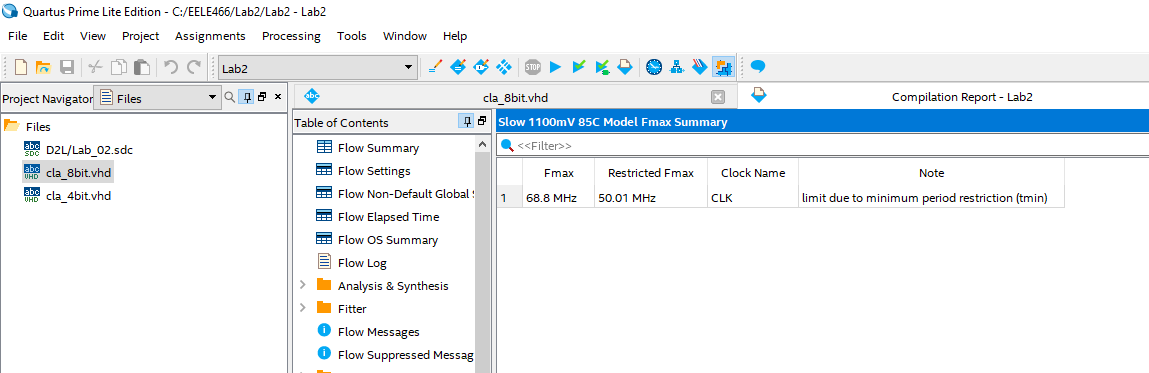


Figure 4: Timing for carry look ahead adder from Lab 2

As seen in figures 3 and 4, we determined that the timing was improved for the carry look ahead adder on Lab 2 with an Fmax of 68.8 MHz. This is what we expected to see in our results. The ability for the adder to look ahead and determine what the carry will be ahead of time has benefits for increasing time efficiency.

3.0 Conclusion

The objective of this lab was to design and test an 8-bit carry look ahead adder using two 4-bit adders and measure the timing of the program. The logic for the VHDL code allows for the adder to look ahead and determine whether there will be a carry before the addition/subtraction is initiated. This increases the speed in which the program can execute. We found that this type of adder is more efficient than the ripple carry adder from Lab 1. This is because by using a look ahead adder, we can save time on the addition process by knowing what the carries will be ahead of time.