ECE 4250 Lab 1

4-Bit Full Adder

Sterling LaBarbera

14096504

Spl2q2

# Objective

In this lab, we designed and simulated a 4-bit ripple carry adder to familiarize ourselves with Modelsim and VHDL basics.

# Lab Work

To implement a 4-bit full adder, we created a single full adder entity and connected four of them together to add two 4-bit values. Each value is represented as a std\_logic\_vector. The full adder entity has three inputs and two outputs, and from the truth table, we find the equations for each output.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | S | Cout |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Figure 1. Truth Table for Full Adder

The equation for S is:

And the equation for Cout is:

Port-mapping statements map each bit from the inputs, outputs, and internal carries for the 4-bit full adder to the appropriate input or outputs for each internal full adder. The order of bits corresponds to the order set in the component creation for the full adder. The bits need to be properly ordered to make sure the correct inputs are being summed at each step. In this case, the inputs are listed in the order X, Y, Cin, and outputs listed Cout, S. Therefore, the port-map must be ordered A, B, Cin, Cout, S to map correctly. Since each full adder takes 1-bit signals, the bit vector elements must be mapped to the appropriate full adder inputs by element.

# Conclusion

The only issues I had were understanding some of the syntax since we were using std\_logic instead of just bit values. However, after a quick explanation by the lab instructor, it made more sense. Reading the textbook also helped clear up some confusion on the VHDL syntax.

Reference:

C. H. Roth, L. K. John. “Introduction to VHDL,” in *Digital System Design Using VHDL*. 3rd Ed. Boston MA, United States: Cengage, 2016, ch. 2.

# Figures

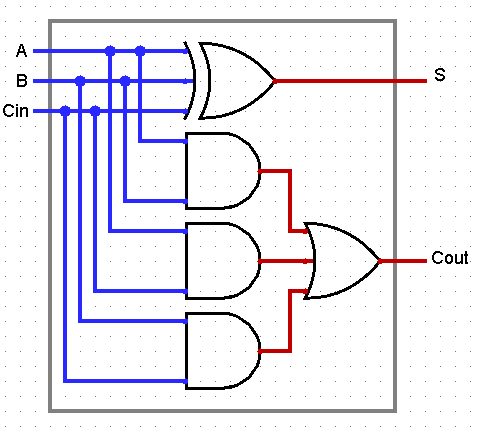


Figure . Full Adder Logical Circuit

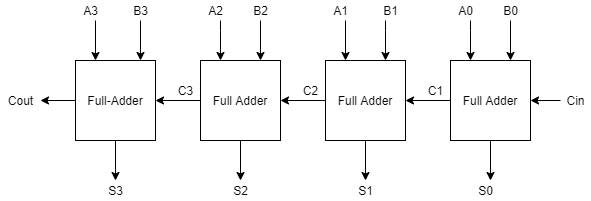


Figure . 4-bit Full Adder

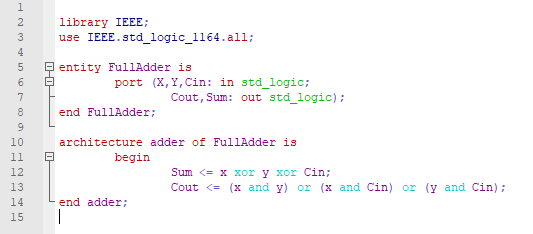


Figure 4. Code for Single Full Adder

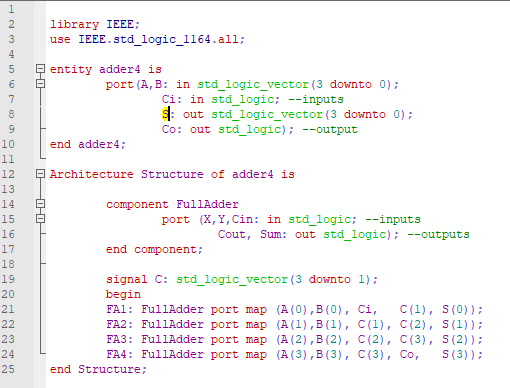


Figure 5. Code for 4-Bit Full Adder

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Simulation Data in Hexadecimal | | | | |
| A | B | Cin | S | Cout |
| 3 | 4 | 0 | 7 | 0 |
| A | 3 | 1 | E | 0 |
| B | 5 | 1 | 1 | 1 |

Figure 6. Simulation Values

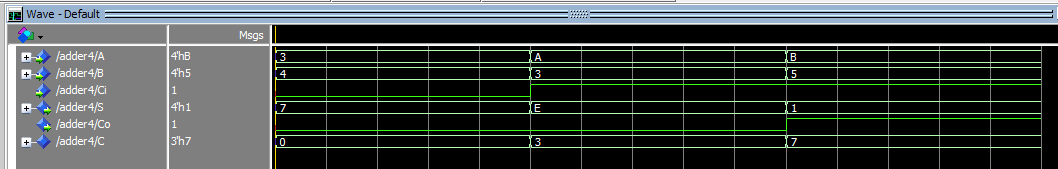


Figure 7. Simulation results