Kyle Johnson and Francesca Narea

ELEN/COEN 21L 47363

Laboratory #5: 4-Bit Ripple Carry Adder

2-17-17

Introduction:

In this lab, we were asked to create a 4-bit ripple carry adder that produced the desired verilog code when particular switches were turned on high. From this lab, we learned how to create ripple carry adders using full adders. In addition, we learned the basics of verilog code and how to implement it into an adder. We also faced a more challenging concept of how to design our waveform since we had to think of numbers to add together.

Prelab:

Our prelab was to pre-make symbols for both the half and full-adder. We used the logic of XOR as well as AND for the half-adder, with inputs INA and INA and outputs SUM and CARRY. The next part of our prelab was to incorporate the half-adder symbol in order to create a full-adder, so we figured out how to connect INA, INB and Cin to two half-adders to obtain our desired output for a full-adder.

Figure 1: Prelab for Kyle Johnson (page 1)

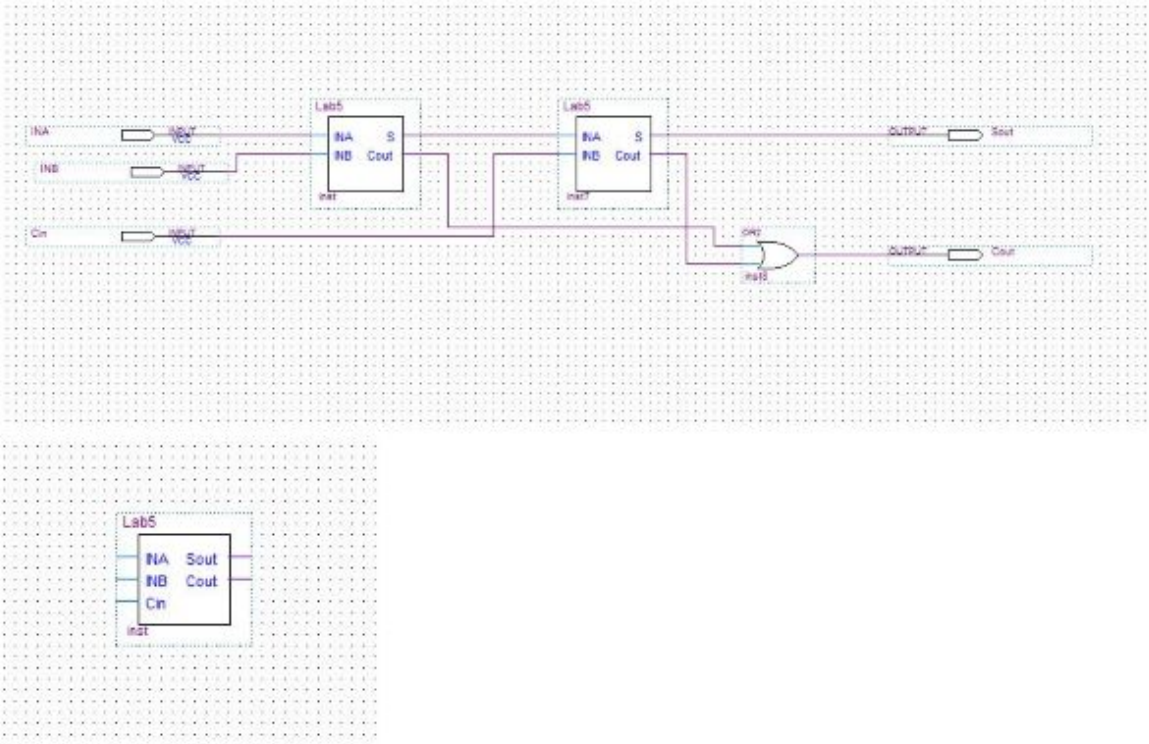
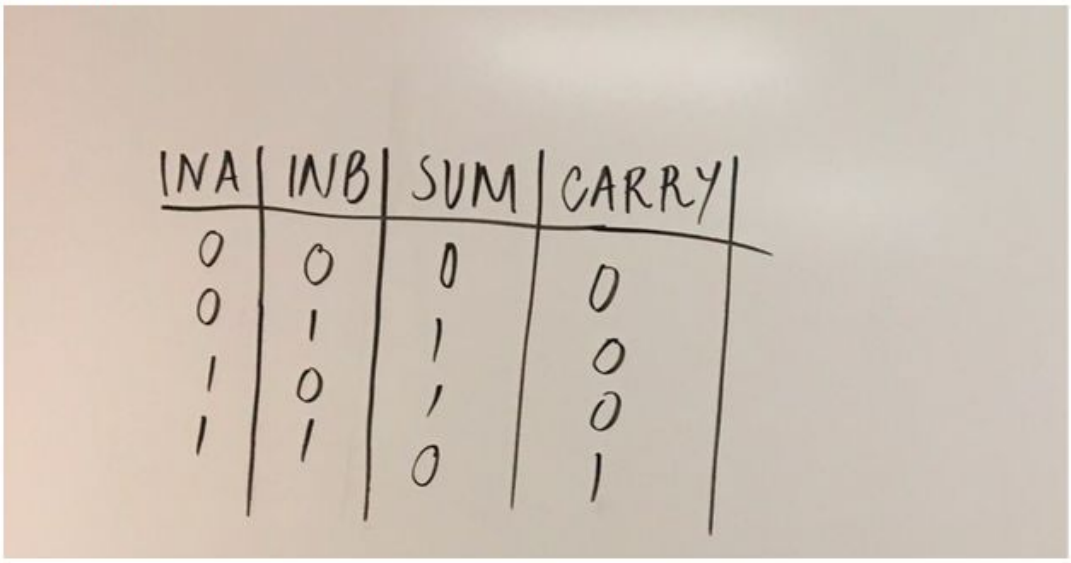


Figure 2: Prelab for Kyle Johnson (page 2)

Half Adder



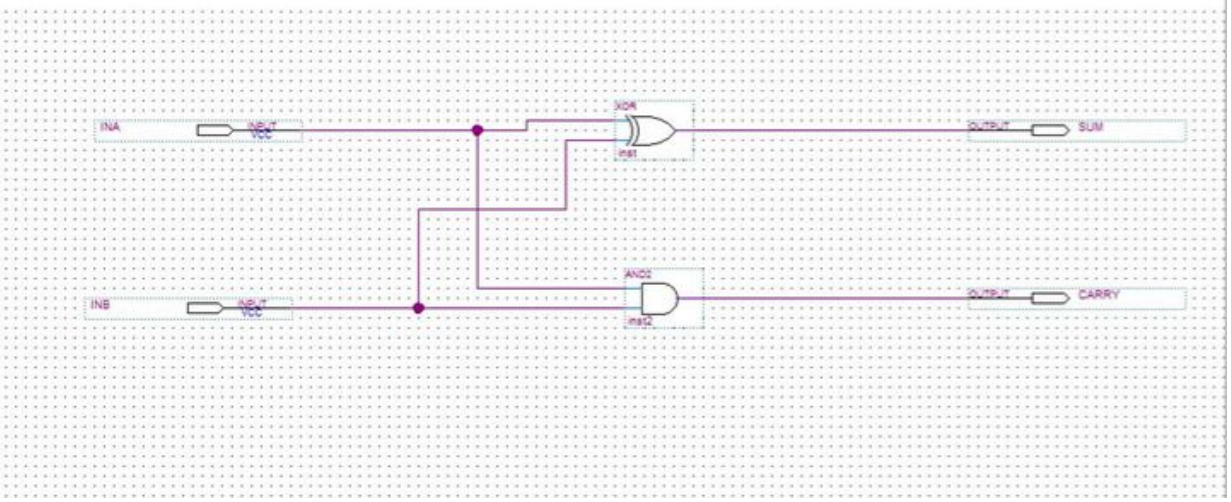
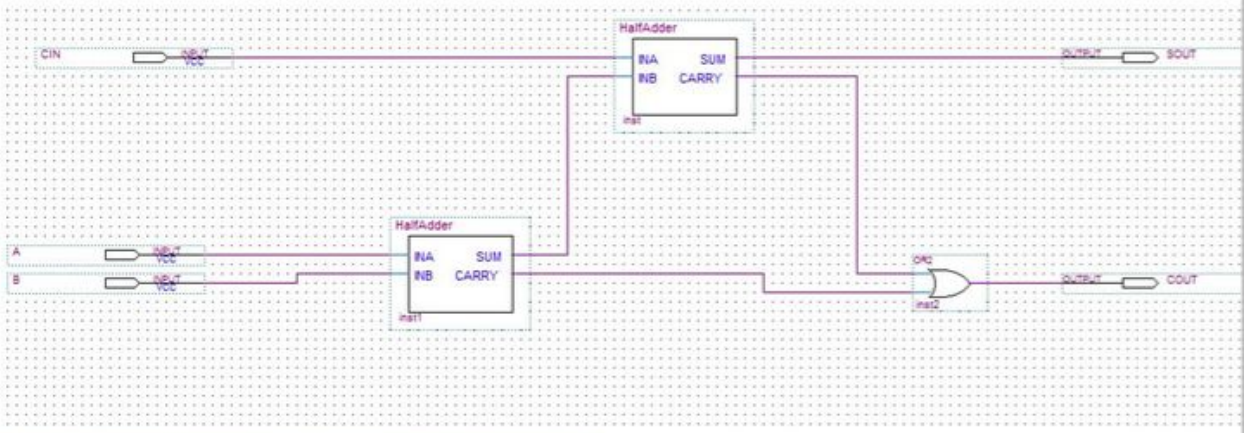
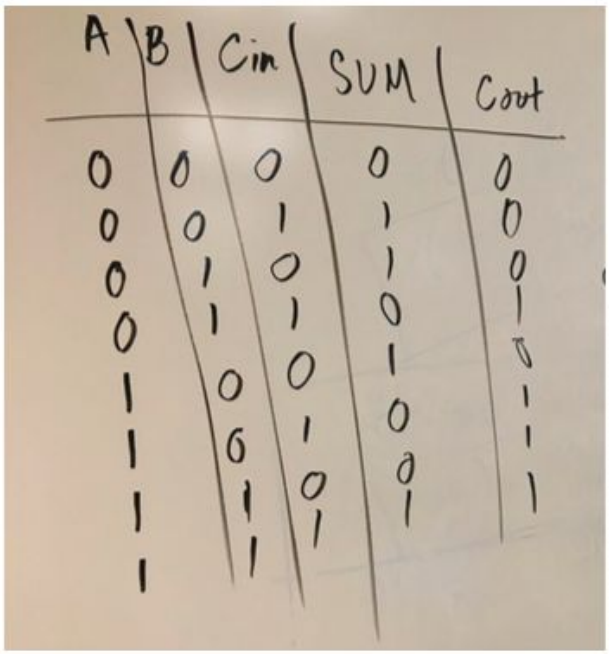


Figure 3: Prelab for Francesca Narea (page 1)

Full Adder





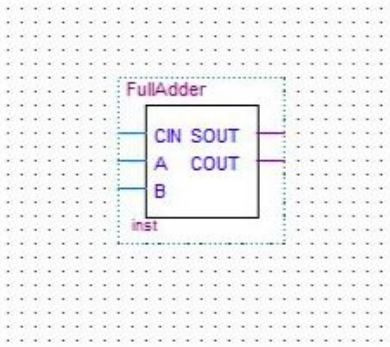


Figure 4: Prelab for Francesca Narea (page 2)

Procedure:

We first tested our prelab half-adder and full-adder to make sure that it was functioning properly. Then, using the two schematics, we created a 4-bit ripple carry adder. We created this using four of the ripple carry adders that all led to a carryout and a sum. We were introduced to a new bus tool in order to input 4 bits at a time, meaning our inputs were X[3..0], Y[3..0] and output was S[3..0]. We also had to label specific wires. Once this was completed, to test this, we ran a waveform on the schematic. The waveform tests we ran involved adding numbers together. We made sure to have a number over 10 to see if the COUT was correct. We then created a symbol for this adder. Next, we created a test circuit with output pins and the 4-bit ripple carry adder. This circuit contained two 7 segment displays with outputs of the 7 segment display pins on the board. In order to create the display board numbers, we filled in the verilog code. We then ran this through a waveform to ensure the circuit works. Finally, we uploaded this onto our board and successfully displayed our circuit.

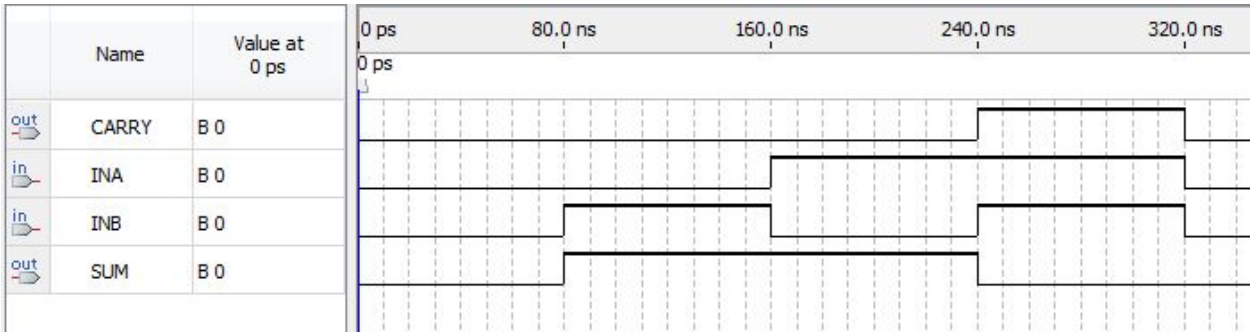


Figure 5: Waveform for Half-Adder

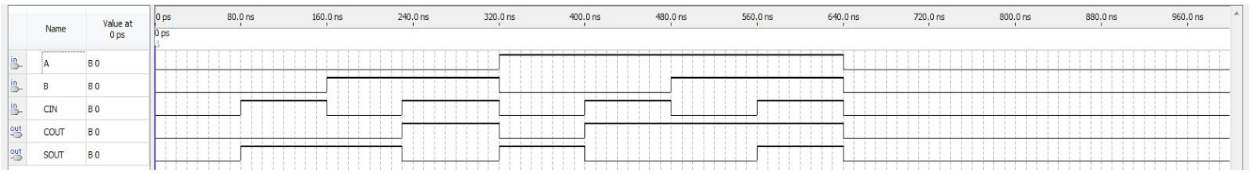


Figure 6: Waveform for Full-Adder

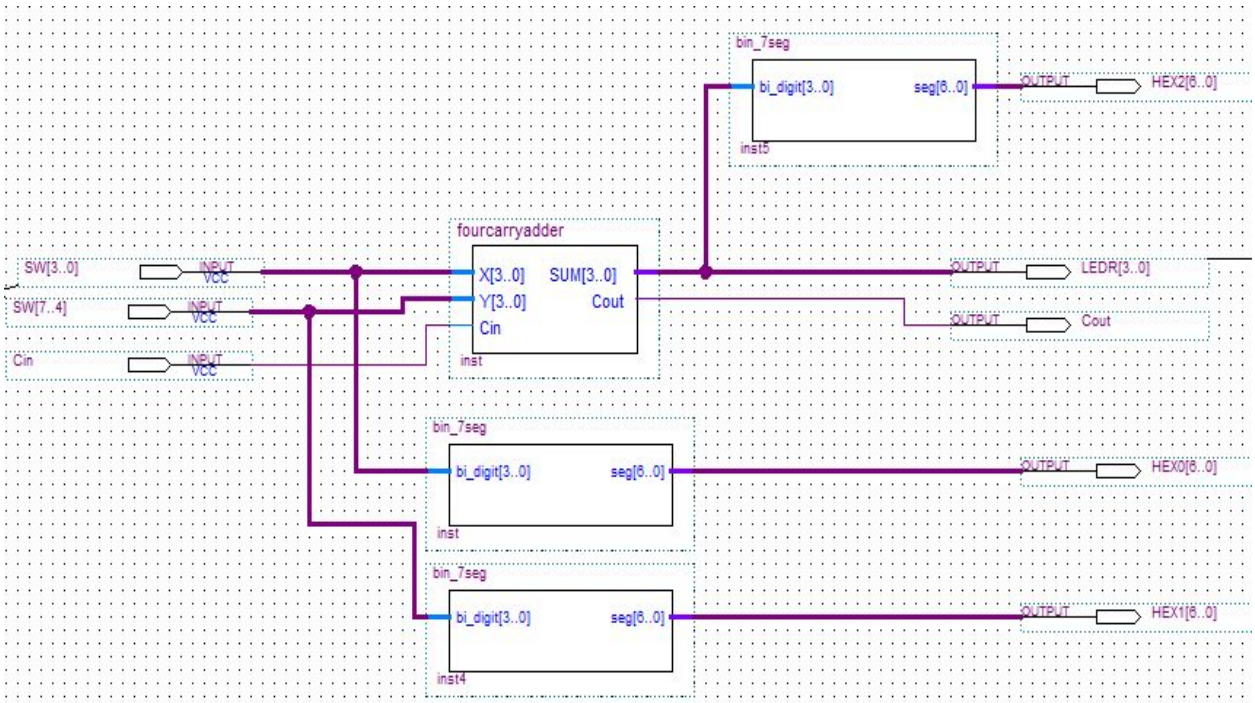


Figure 7: Schematic #1

```
1   module bin_7seg(bi_digit,seg);
2   input [3:0] bi_digit;
3   output [6:0] seg;
4   reg [6:0] seg;
5   // seg = {g,f,e,d,c,b,a};
6
7
8   always @ (bi_digit)
9   case (bi_digit)
10          4'h0: seg = ~7'b0111111;
11          4'h1: seg = ~7'b0000110;            // ---a----
12          4'h2: seg = ~7'b1011011;            // |      |
13          4'h3: seg = ~7'b1001111;            // f      b
14          4'h4: seg = ~7'b1100110;            // |      |
15          4'h5: seg = ~7'b1101101;            // ---g----
16          4'h6: seg = ~7'b1111101;            // |      |
17          4'h7: seg = ~7'b0000111;            // e      c
18          4'h8: seg = ~7'b1111111;            // |      |
19          4'h9: seg = ~7'b1100111;            // ---d----
20          4'ha: seg = ~7'b1110111;
21          4'hb: seg = ~7'b1111100;
22          4'hc: seg = ~7'b1011000;
23          4'hd: seg = ~7'b1011110;
24          4'he: seg = ~7'b1111001;
25          4'hf: seg = ~7'b1110001;
26  endcase
27
28  endmodule
29
```

Figure 8: Verilog Code

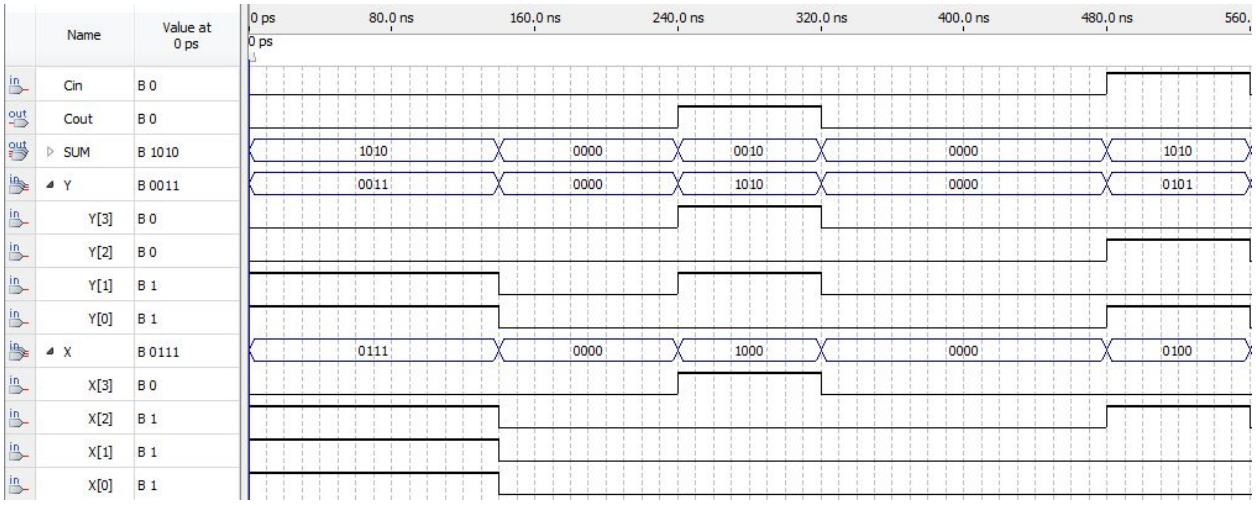


Figure 9: Waveform for Ripple Carry

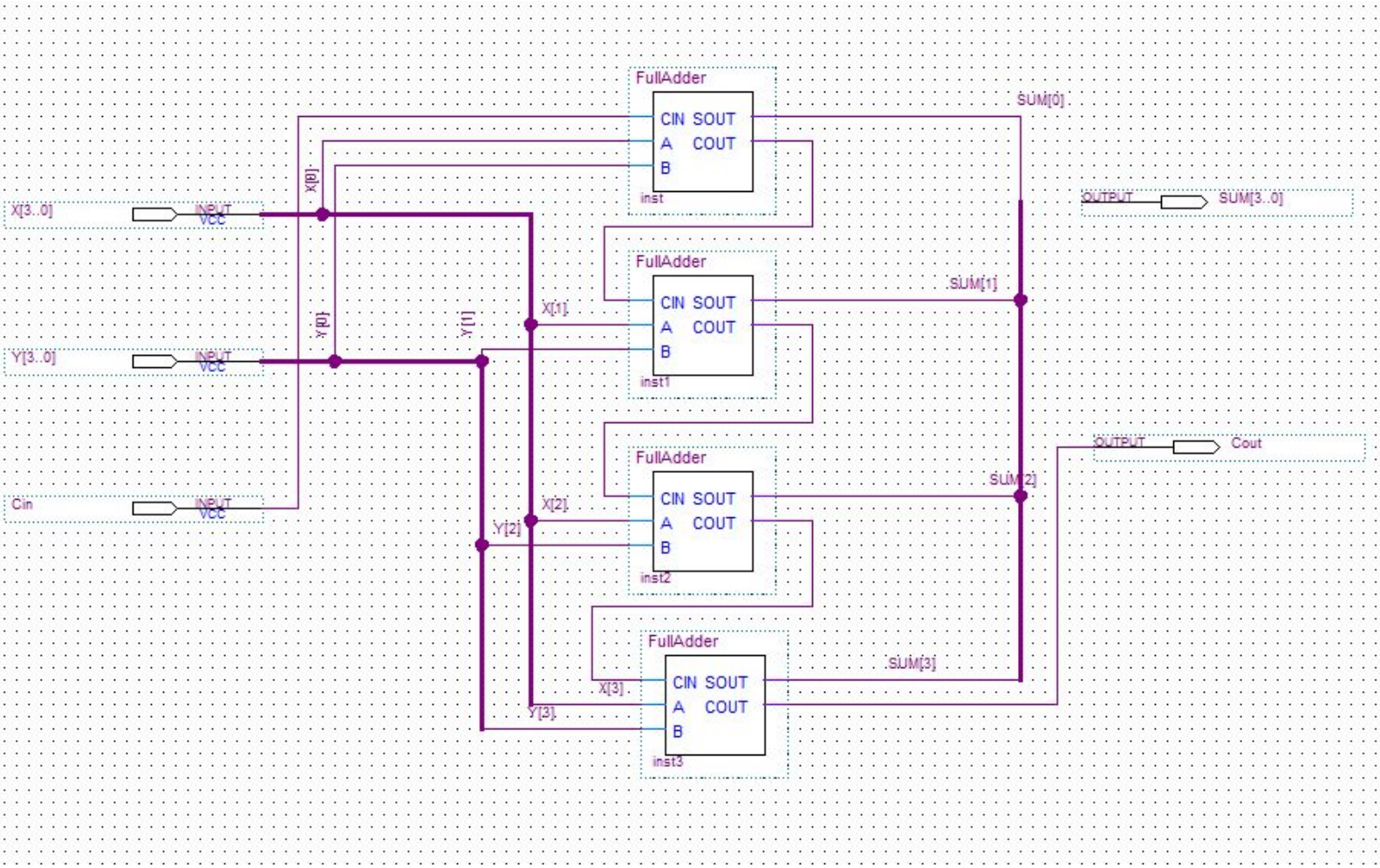


Figure 10: Schematic #2

Report:

1. If the result was greater than 15, then the circuit would overload and the analog display would show an error and the result would be incorrect.
2. The current passes through 12 gates before reaching the Cout of the ripple carry adder.
3. The schematic would be the same except for there would be 8 full adders and the switches would be 0 through 7 instead of 0 through 3.
4. There would be 24 gates to pass through because each full adder contains 3 gates and there are 8 full adders.

Conclusion:

Some of the skills that we learned from this lab are how to create a half-adder, a full-adder, and a ripple carry adder. While the prelab involved the more familiar half-adder and full-adder, the ripple carry adder was a more challenging schematic to complete conceptually but we were able to successfully work through this challenge. Another skill that we learned is how to implement verilog code into a ripple carry adder-- which also tested our understanding of the verilog code from the lecture. At first, one of our numbers on the display board was incorrect because one of the verilog codes was incorrect but this was an easy debugging process since we were able to immediately determine which part of our lab needed to be checked over. The most challenging part to this lab was creating the schematic for the test circuit as well as the waveform for the verilog code because we had to determine which numbers were the best to test the circuit.