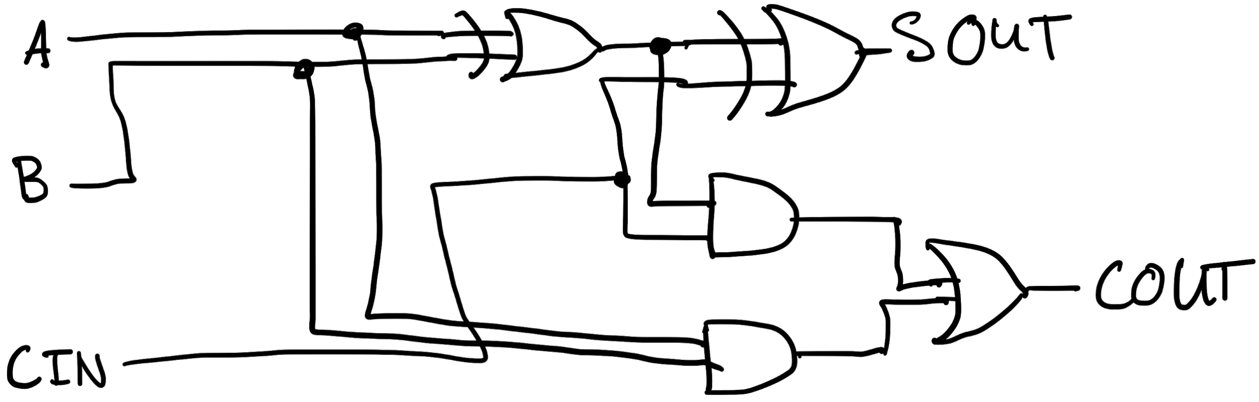
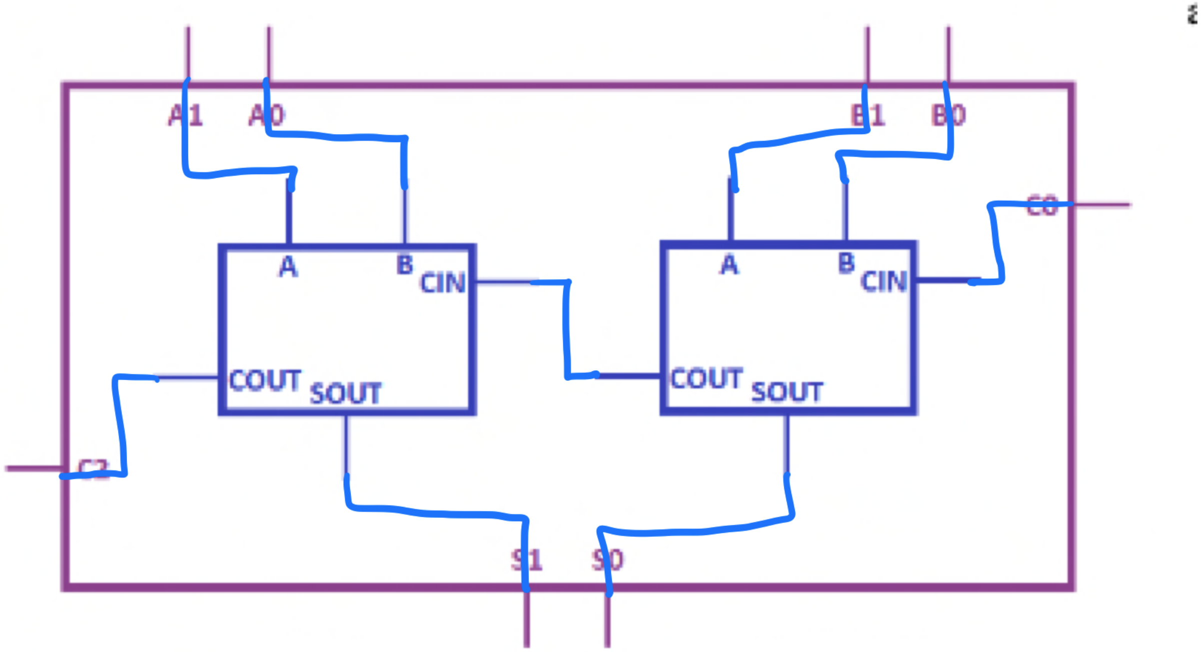
Lab 5

1. SOUT = A xor B xor CIN
   COUT = AB + (A xor B)CIN

2.



3.
   4 gates are on the path from inputs to C2



4.

```
module myfulladd(input a, b, cin, output sout, cout);
        assign sout = (a^b^cin);
        assign cout = ((a^b&cin) | (a&b));
endmodule
```

5.

```
module myadder2(input a1, a0, b1, b0, c0, output s1, s0, c2);
        wire c1;
        myfulladd(a1, a0, c0, s0, c1);
        myfulladd(b1, b0, c1, s1, c2);
endmodule
```

6.

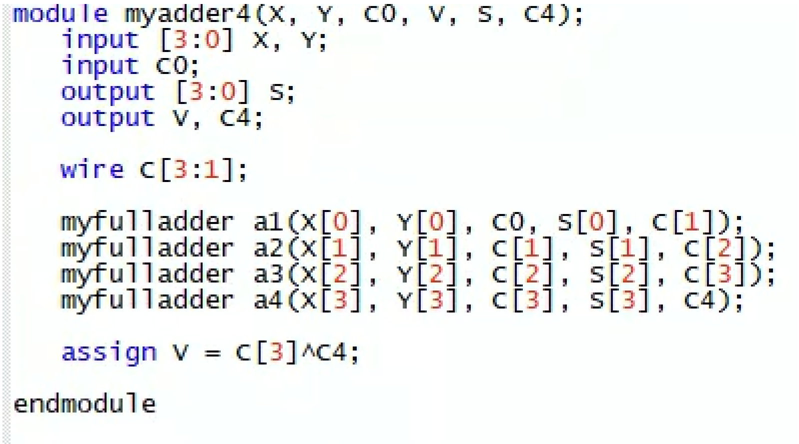
i. Test A1 = 0 and A0, B1, B0 = 1. This is the only time where C2 should equal 1 when A1 = 0.

ii. To test the individual carry connections, use an intermediate output C1 and set it equal to COUT. Observe if C1 is the same as CIN.

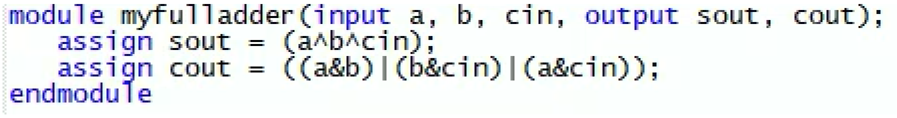
iii. Basic test for functionality is to set all input to 1. Expect to see S1 and C2 equal to 1 and S0 equal to 0.

iv. Keep one of the 2 sets of inputs constant. Vary the other two to observe if the output is correct. If not, change the wiring on the affected adder.

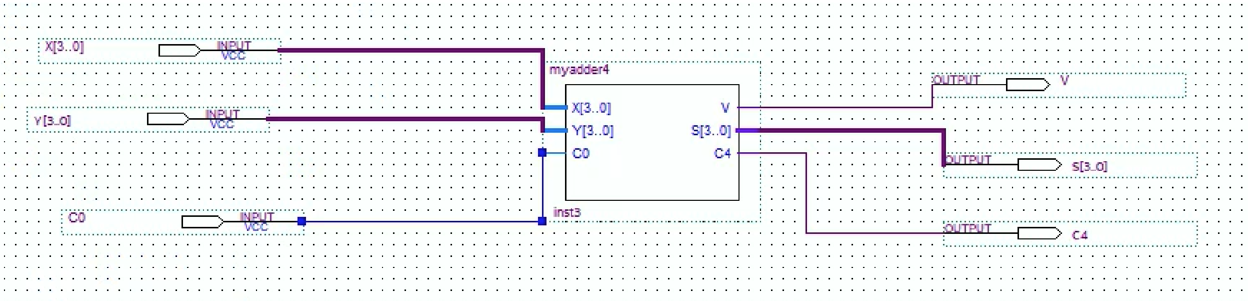
Lab

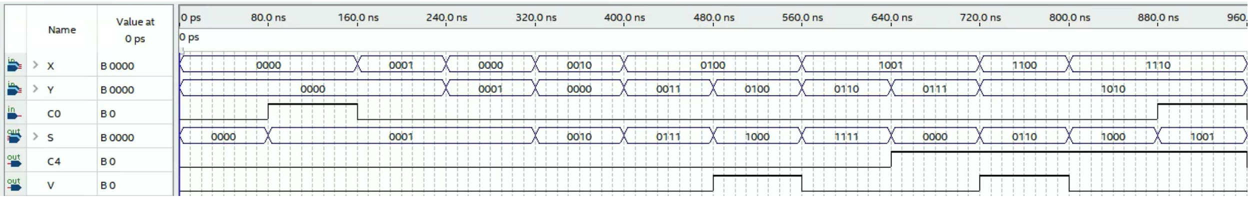


```
module myadder4(X, Y, C0, V, S, C4);
   input [3:0] X, Y;
   input C0;
   output [3:0] S;
   output V, C4;

   wire C[3:1];

   myfulladder a1(X[0], Y[0], C0, S[0], C[1]);
   myfulladder a2(X[1], Y[1], C[1], S[1], C[2]);
   myfulladder a3(X[2], Y[2], C[2], S[2], C[3]);
   myfulladder a4(X[3], Y[3], C[3], S[3], C4);

   assign V = C[3]^C4;

endmodule
```



```
module myfulladder(input a, b, cin, output sout, cout);
   assign sout = (a^b^cin);
   assign cout = ((a&b)|(b&cin)|(a&cin));
endmodule
```





1. C4 = 0 and V = 0 when X = 0 and Y = 0
   C4 = 0 and V = 1 when X = 4 and Y = 4
   C4 = 1 and V = 0 when X = 9 and Y = 7
   C4 = 1 and V = 1 when X = C and Y = A
2. To make an 8 bit adder, I would connect the C4 of one 4 bit adder to C0 of the other 4 bit adder. The new V output would be C8^C7 and the C8 output would be created from the second 4 bit adder as well as the carry in of the first 4 bit adder.
3. The initial testing of the circuit went smoothly.
4. No this would not be caught because since it is an adder, these bits would still be added anyways since they are of the same magnitude.
5. i. The incorrect results are shown when adding A=2 & B = 1, A = 0 & B = 11, A = 3 & B = 9, A = 1 & B = 9, A = 7 & B = 4, and A = 11 & B = 0
   ii. It looks like the error is in the wiring of A as shown by tests 6 & 7 (A = 3, 1 and B = 9). Since for the same B input, the outputs are incorrect.
   iii. To address this, more extensive tests on A would need to be done. To identify which bit is misbehaving, I would test all possible combinations of A (16 tests total).