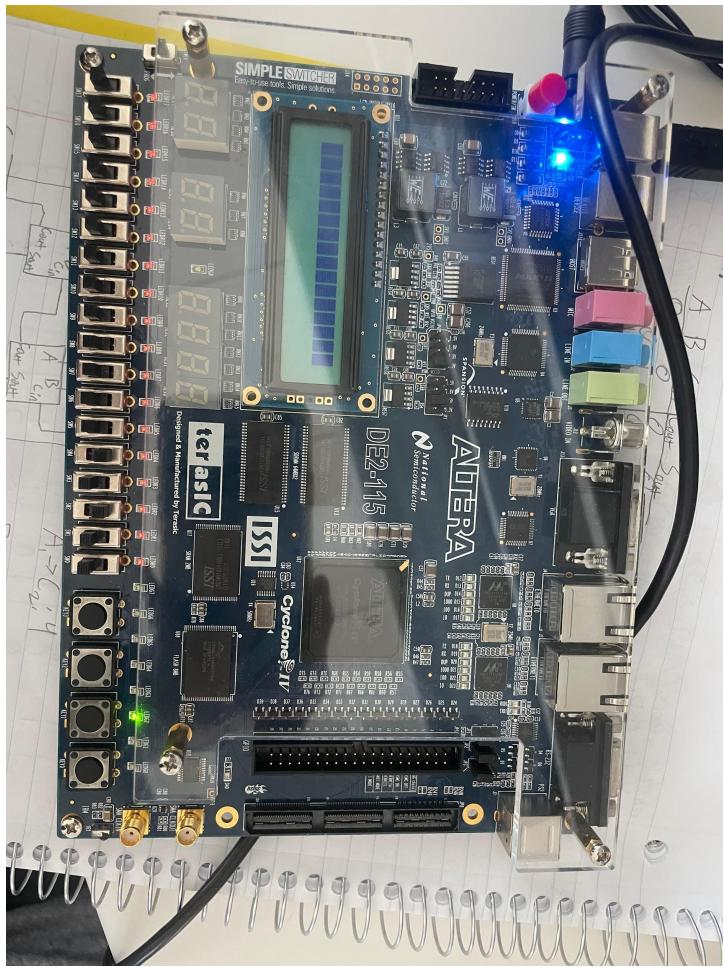


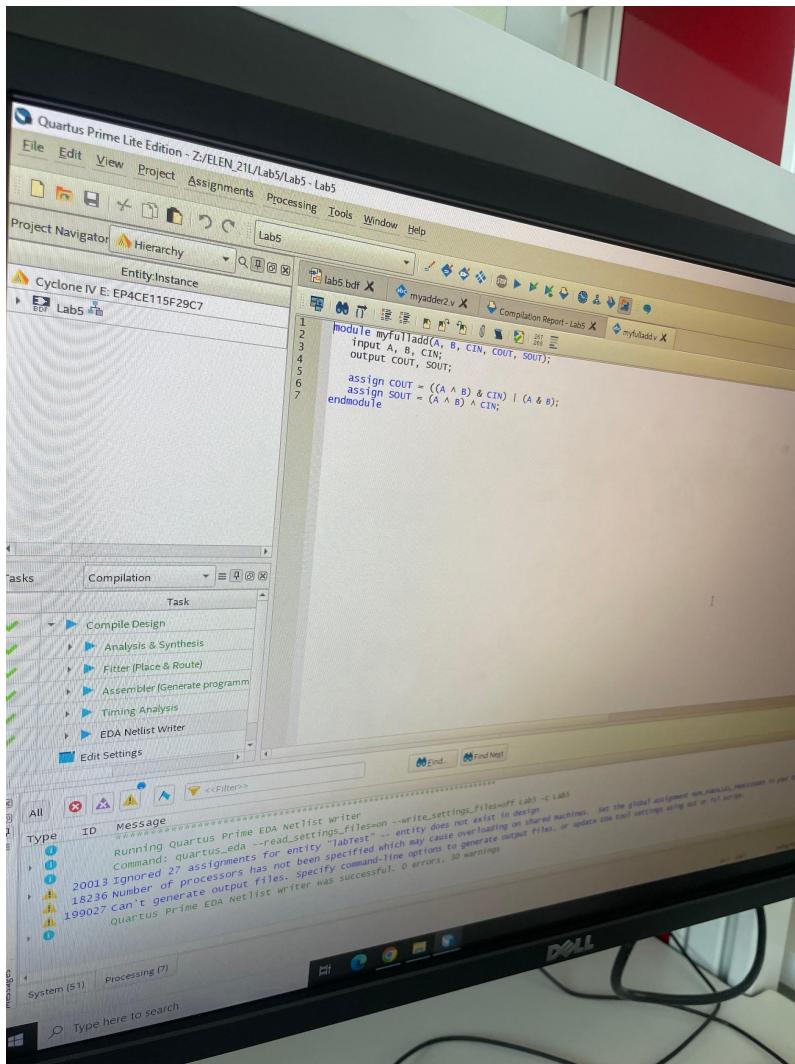
Ayden Dauenhauer

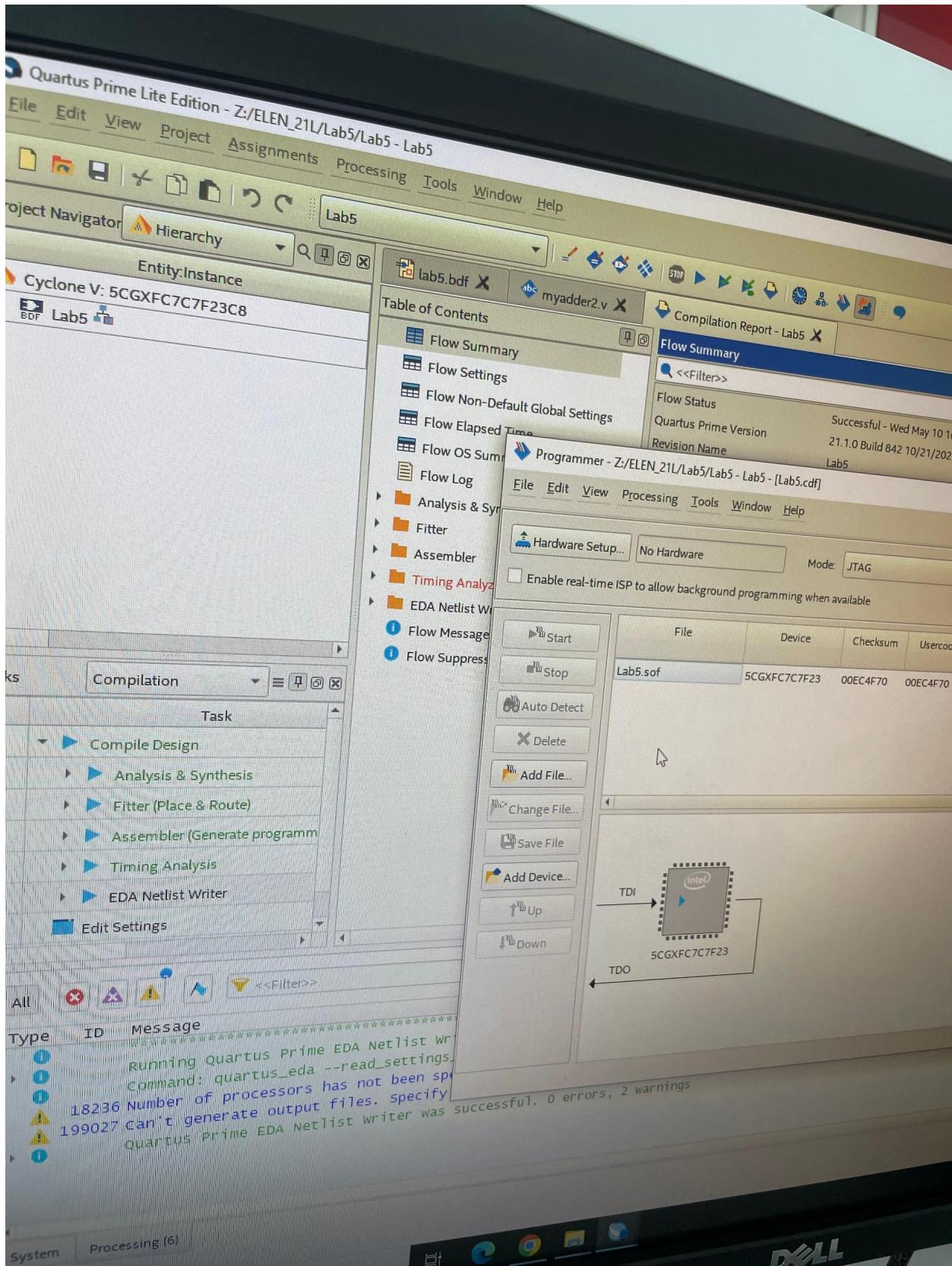
Seana Corners

Lab 5 Post Lab Report

1. A pair of X and Y inputs that would result in the following behavior:
 - a. C4 = 0 and V = 0
 - i. X = 0000
 - ii. Y = 0000
 - b. C4 = 0 and V = 1
 - i. X = 1111
 - ii. Y = 0001
 - c. C4 = 1 and V = 0
 - i. X = 0111
 - ii. Y = 0111
 - d. C4 = 1 and V = 1
 - i. X = 1111
 - ii. Y = 1111
2. If we had to make an 8-bit adder using the 4 bit MUX, we would connect the C0 into the first MUX and then connect C4 as input to a second MUX. C8 would be the C output of the second 4 bit adder and V would be determined by the overflow.
3. The initial testing of the circuit went smoothly.
4. If the X and Y inputs were switched then we would not be able to detect this based on observing the outputs during testing. In the Verilog code, the X and Y inputs are assigned the same values and the order of operations are determined by the parentheses. The number order does not matter because the operations are determined independently of what the actual numbers may be.







```
1 module myfulladd(A, B, CIN, COUT, SOUT);
2     input A, B, CIN;
3     output COUT, SOUT;
4     assign COUT = ((A & B) & CIN) | (A & B);
5     assign SOUT = (A & B) ^ CIN;
6 endmodule
7
```

- Lab5

File Tools Window Help

lab5.bdf X myadder2.v Compilation Report - Lab5 X

```
1 module myadder2(x, y, c0, s, v, c4);
2   input c0;
3   input [3:0]x, y;
4   output [3:0]s;
5   output v, c4;
6   wire [3:1]c;
7
8   myfulladd(x[0], y[0], c0, s[0], c[1]);
9   myfulladd(x[1], y[1], c[1], s[1], c[2]);
10  myfulladd(x[2], y[2], c[2], s[2], c[3]);
11  myfulladd(x[3], y[3], c[3], s[3], c4);
12  assign v = c[3] ^ c4;
13 endmodule
```

