EE 231 Lab 3

Adder/Subtractor

In the previous lab you have defined a half adder by using primitive gates. As the design becomes more and more complex, it is convenient to create separate modules and then combine them in one file. For example, in the previous lab you have created a half adder module and in this lab you will create a full adder/subtractor which can be designed by using the half adder module. So, in this lab you will instantiate two half adders to form the full adder, then instantiate four full adders to create the 4-bit adder/subtractor. Program 1 illustrates this concept.

In the following sample program, the module sample_moduleA is instantiated twice as SA1, and SA2 in the higher level module sample_moduleB.

```
module sample_moduleA(ouput C, input A, B);
    wire w1;
    XOR (w1, A, B);
    AND (C, A, w1);
endmodule

module_sample_moduleB(output D, input E, F);
    wire G, H;
    sample_moduleA SA1(G, E, F);
    sample_moduleA SA2(D, G, E);
endmodule
```

Program 1: An Example of a Module Instantiating Other Modules

1.Lab

1.1.Design a 4-bit adder using the half adders.

1.2.Using a decode, an add, a sel, and the needed components to be able to select whether your design acts as a 4-bit adder or a 4-bit subtractor.

1.3.Simulate your circuit and verify its operation. The Logic Analyzer can also be used for the simulation. You will need to try enough combinations to verify that the results of the addition and subtraction along with *C* and *V* are correct.

1.4.Add the BCD decoder you designed in last week's lab to output the adder/subtractor so you can display the output on your 7-segment display.

1.5.Connect your *sel* and your inputs to hardware switches, and connect the output to the 7-segment display.