**EECS 478 Project 2 report**

**Cheng-Fu Lin**

**Unique name:chengfu**

Implementation:

* 2-Input AND :

I implemented the TruthTable of 2-Input AND in the function createAND2Node.

* 3-Input XOR:

I implemented the TruthTable of 3-Input XOR in the function createXOR3Node.

* 4-Input MUX

I implemented the TruthTable of 4-Input Mux in the function createMUX4Node.

* Adder:

I implemented Sum bit using the XOR3 gate by S[i]=A[i] xor B[i] xor cin[i]. And implemented the cout bit using the AND2, XOR3 and OR2 nodes to perform the function cout[i] = A[i] and B[i] or cin[i] and A[i] xor B[i].

* Subtractor:

I implemented the subtract function by the same method of implementing adder. I use the XOR3 gate to obtain output bit S[i]=A[i] xor B[i] xor PBo[i]. I used AND2, OR2, XOR3 to compute borrow bit Bo[i] = PBo and not(A[i] xor B[i]) and not(A[i]) and B[i]. In computing borrow term, I need a not gate which can obtain by using XOR3 with input 1, 0, and A.

* Datapath:

1. I first extend two input into 19 bit by sign extension.
2. I use “Shifter” to generate 4x and 2y
3. I use “Adder” to generate 5x and 3y
4. I use “Subtractor” to generate -5x and -3y
5. I analyze the problem and find out I need the msb of 5x-3y, 5x and 3y
6. I use “Subtractor” to generate 5x-3y
7. By setting selector as following, I can use MUX4 to calculate the result.

sel1 = (5X or 3Y)

sel2 = 5X-3Y

1. Then I can get a truth table like this:

sel1 sel2 out

0 0 3y

0 1 5x

1 0 -3y

1 1 -5x

1. As a result, I can get z we need.

* Verification:

I have used the simulator to verify my design. In each module and datapath, I check 10 possible input with the combination of positive values and negative values. I also checked my blif files with Hsiu-Ting Hsieh.