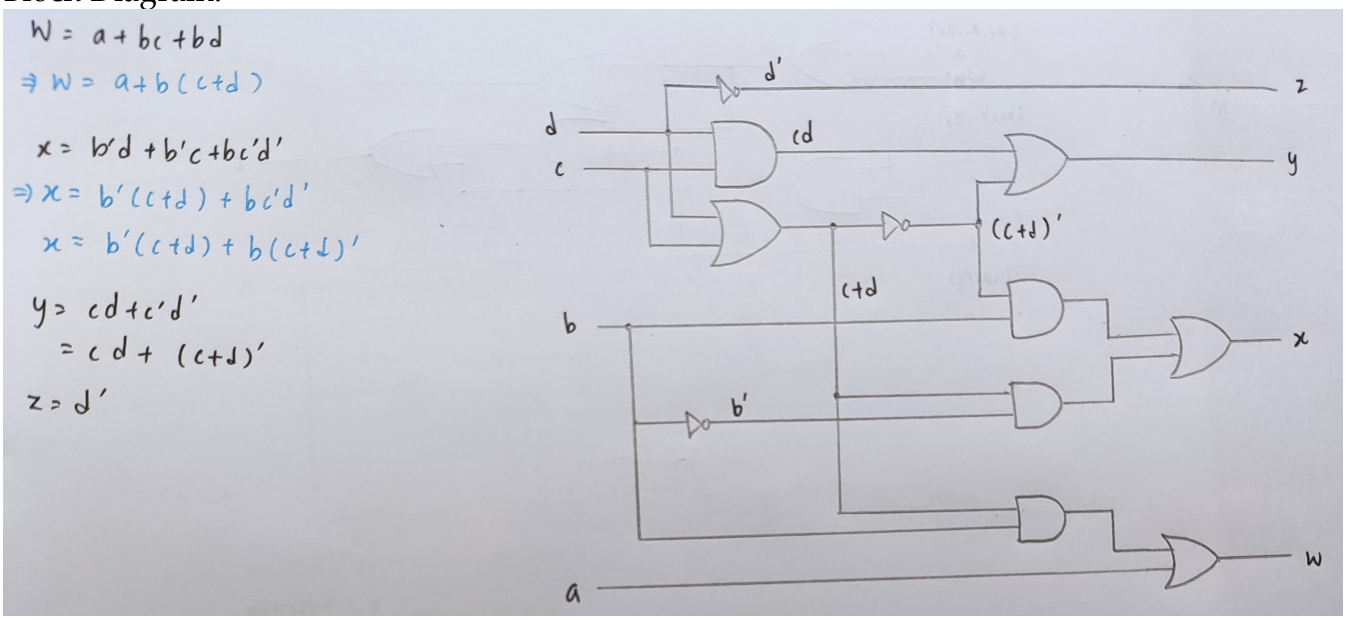
Report Lab 1

Design Specification

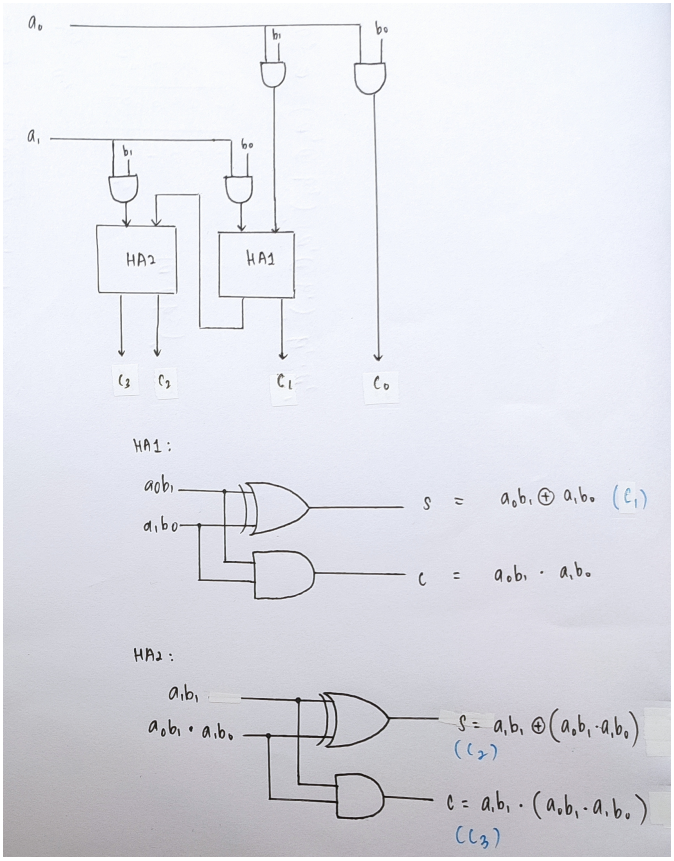
Lab1_1:
output w, x, y, z
input a, b, c, d

Block Diagram:



Lab1_2:
output c0, c1, c2, c3
input a0, a1, b0, b1

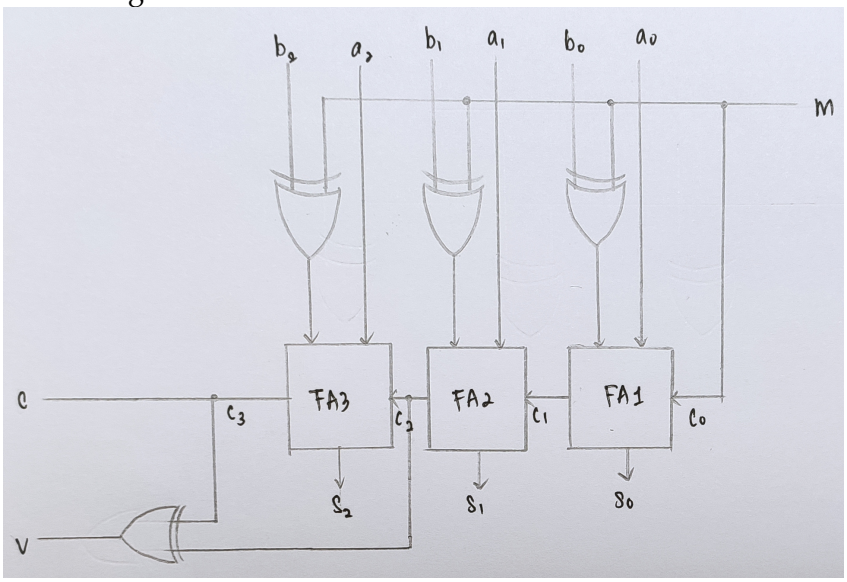
Block Diagram:

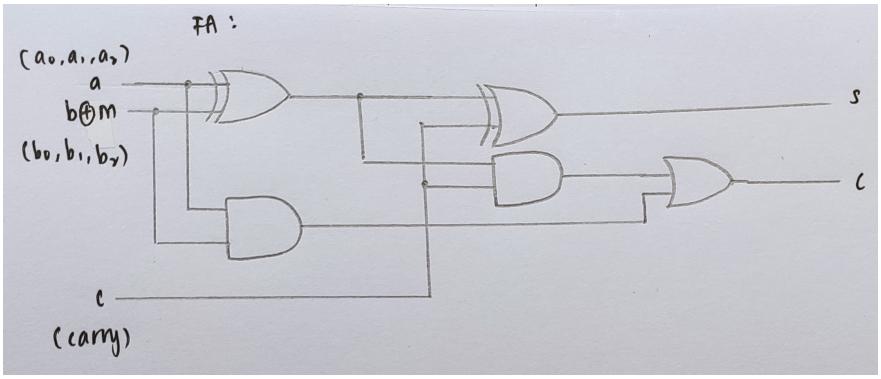


Lab1_3

input [2:0] a, [2:0] b, m
output [2:0] s, v

Block Diagram:

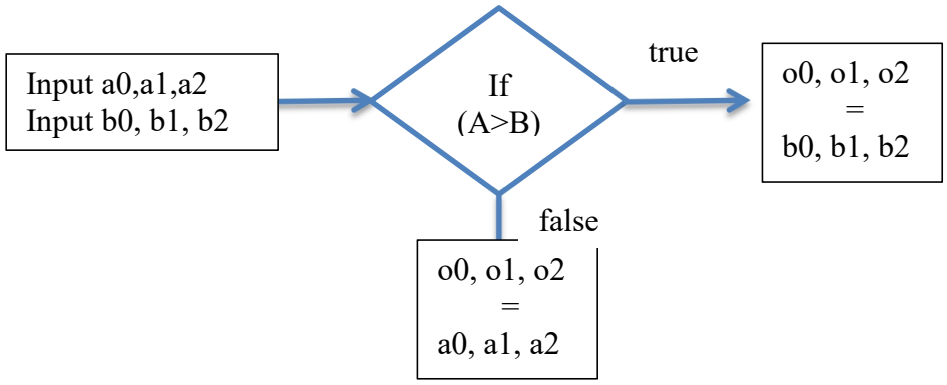




Lab1_bonus

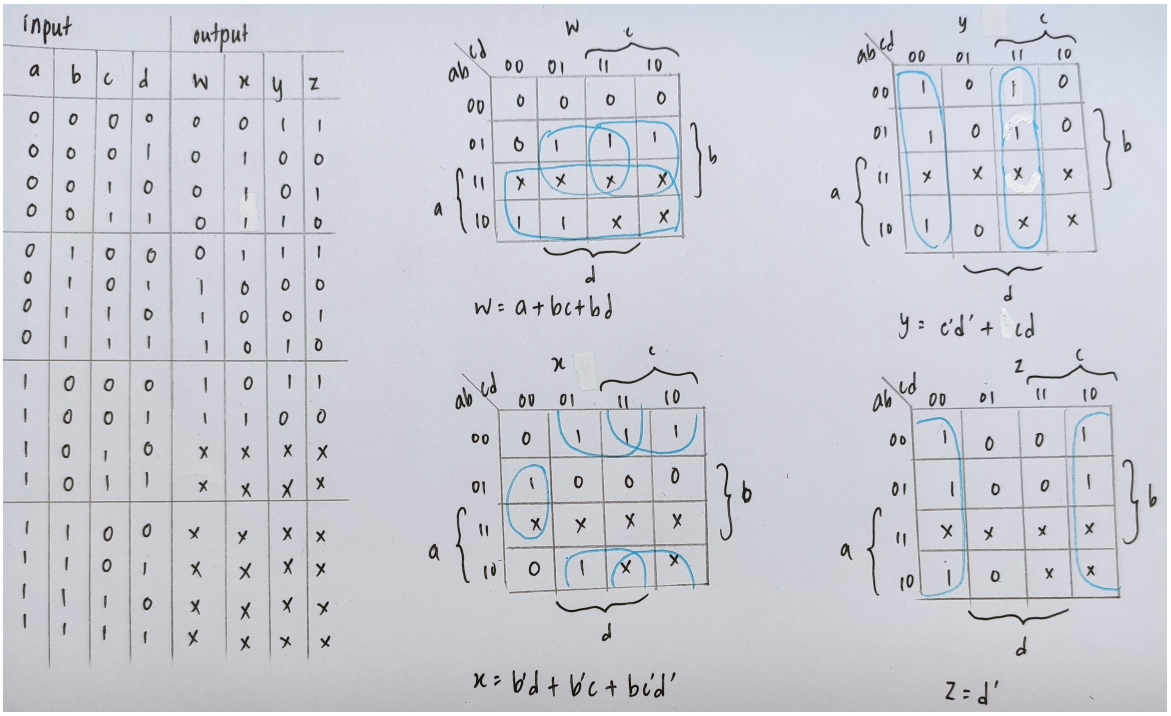
input a0, a1, a2, b0, b1, b2
output reg o0, o1, o2

Block Diagram:



Design Implementation

Lab1_1



w = a + bc + bd
  = a + b (c + d)

x = b'd + b'c + bc'd'
  = b' (c + d) + b (c + d)'

y = c'd' + cd

z = d'

Truth Table:

inputs				outputs			
a	b	c	d	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x

1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

Lab1_2

When 2-bit multiplicand a(a1,a0) x with 2-bit multiplier b(b1,b0), the results are the product c(c3,c2,c1,c0)
Which is:
C0 = a0b0
C1 = a0b1 *xor* a1b0
C2 = a1b1 *xor* (a0b1 *and* a1b0)
C3 = a1b1 *and* (a0b1 *and* a1b0)

Truth Table:

inputs				outputs			
a1	a0	b1	b0	c3	c2	c1	c0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

Lab1_3

The addition and subtraction operation is combined into a circuit with one common binary adder by including an exclusive-OR gate with full adder. When the m input is 1, the circuit is a subtractor, while m input is 0, it become an adder. B with *xor 0* is just B, while B with *xor 1* is the complement of B. the B inputs are all complemented and a 1 is added through the input carry. the circuit performs the operation of A plus the 2's complement of B.

The detection of the overflow after the addition of 2 numbers are done by *xor* the c2 and c3. When 2 unsigned numbers are added, an overflow is detected from the end carry out of the most significant

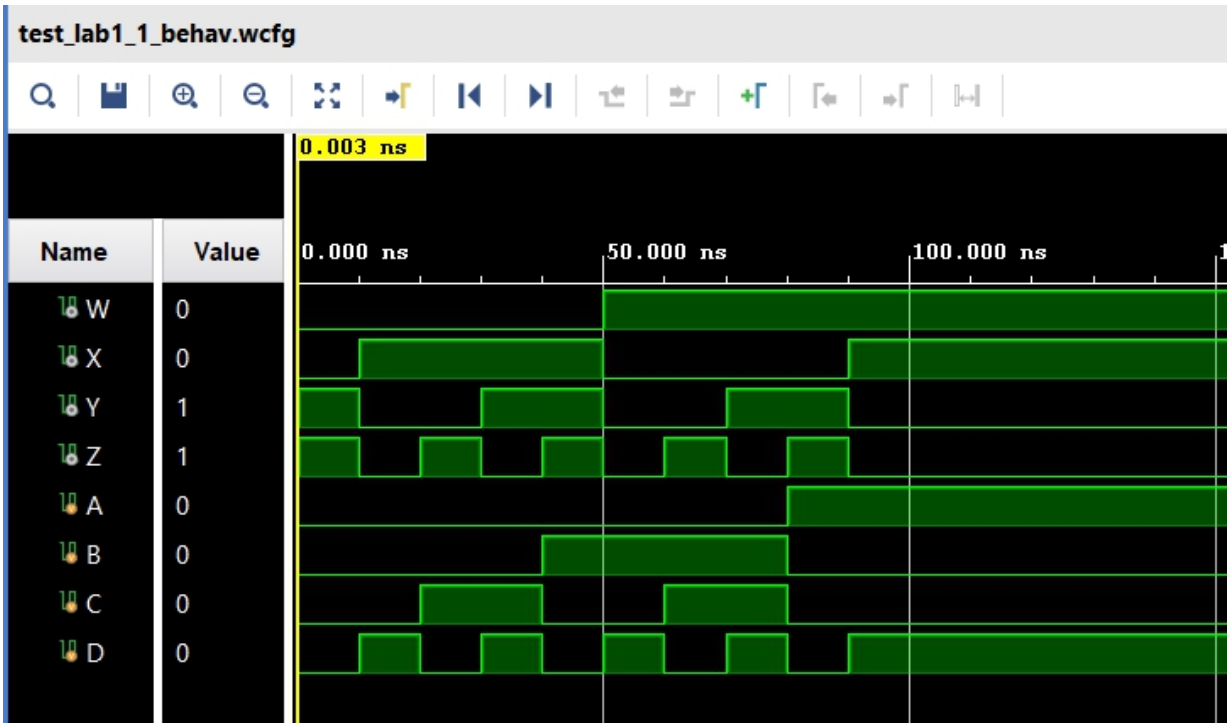
position. In the case of signed numbers, the leftmost bit represent the sign, nd the negative number are in 2's complement form, and there is no overflow for signed bit.

Lab1_bonus

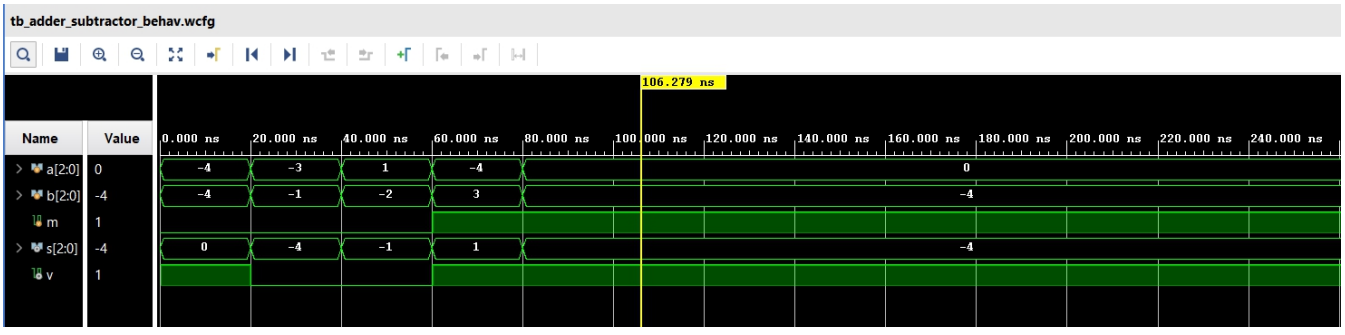
This question is more directly by comparing only the A and B and output the smaller result.

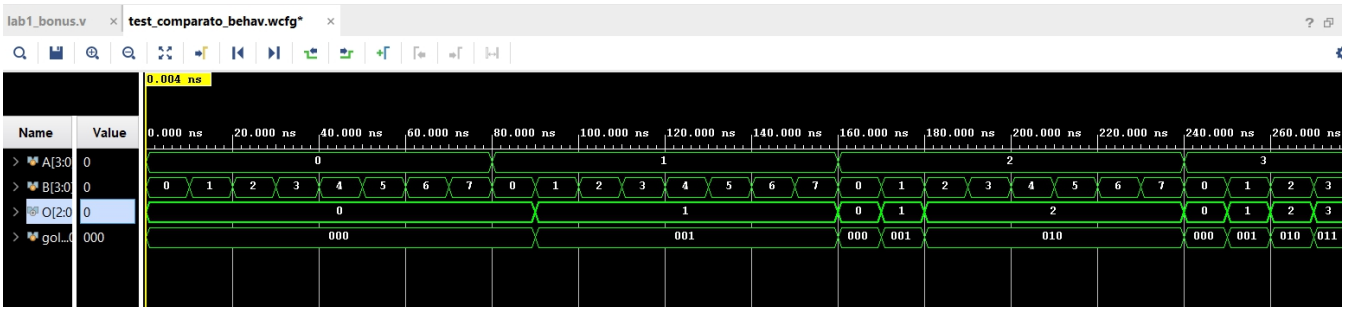
inputs		outputs
A	B	O
0	0	-
0	1	A
1	0	B
1	1	-

Discussion


test_lab1_1_behav.wcfg
0.003 ns
Name
Value
0.000 ns
50.000 ns
100.000 ns
W 0
X 0
Y 1
Z 1
A 0
B 0
C 0
D 0


test_lab1_2_behav1.wcfg
0.006 ns
Name
Value
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
140.000 ns
160.000 ns
out0 0
out1 0
out2 0
out3 0
in_a0 0
in_a1 0
in_b0 0
in_b1 0


tb_adder_subtractor_behav.wcfg
106.279 ns
Name
Value
0.000 ns
20.000 ns
40.000 ns
60.000 ns
80.000 ns
100.000 ns
120.000 ns
140.000 ns
160.000 ns
180.000 ns
200.000 ns
220.000 ns
240.000 ns
a[2:0] 0
b[2:0] -4
m 1
s[2:0] -4
v 1



For this lab 1, these experiments make me have a better understanding about the conversion of BCD-to-Excess-3 Code, full adder, how to design the verilog code for the adder and subtractor and comparator. The results as expect are due to the correctness of the design for the function, including the connection of the inputs as to get the final outputs are high. I think this experiment are the basic for the verilog code, and I learned more about verilog than before.

Conclusion

I think I can now have a better understanding about what I had learned from the last semester, because I can now write verilog by myself and I can implement the knowledge I learned to help me get the expected result from the experiment.

References

Digital Design With An Introduction To The Verilog HDL Fifth Edition by M.Morris Mano and Micheal D. Ciletti
This book help me to correct my block diagram for the question 1, for rducing the amount of logic gates.

