Name: Newman Ilgenfritz

CIT 593 Module 3 Assignment: Transistors to Logic

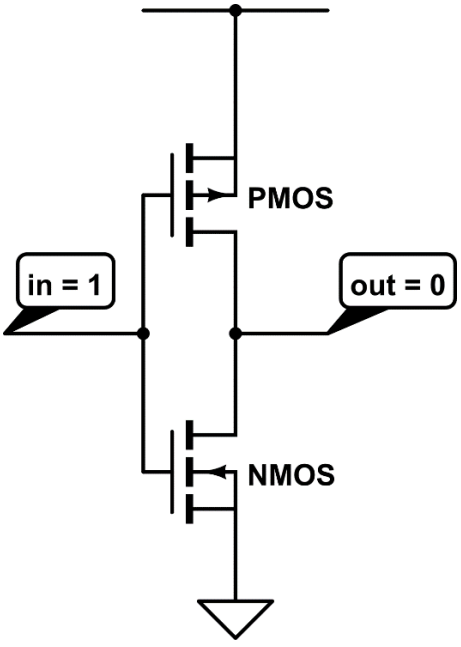
Remember that this is an individual assignment (NO group work).

Part I) Transistors & Gates

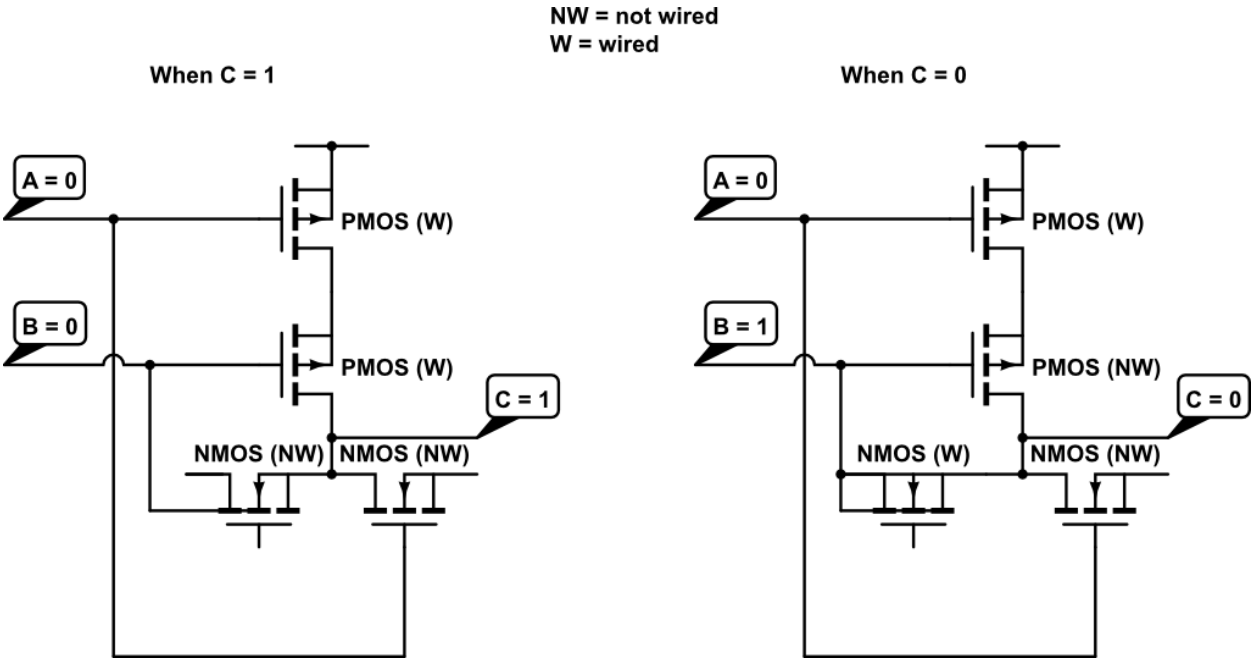
3.1:

	N-Type	P-Type
Gate=1	Closed	Open
Gate=0	Open	Closed

3.2:



3.4:



3.5:

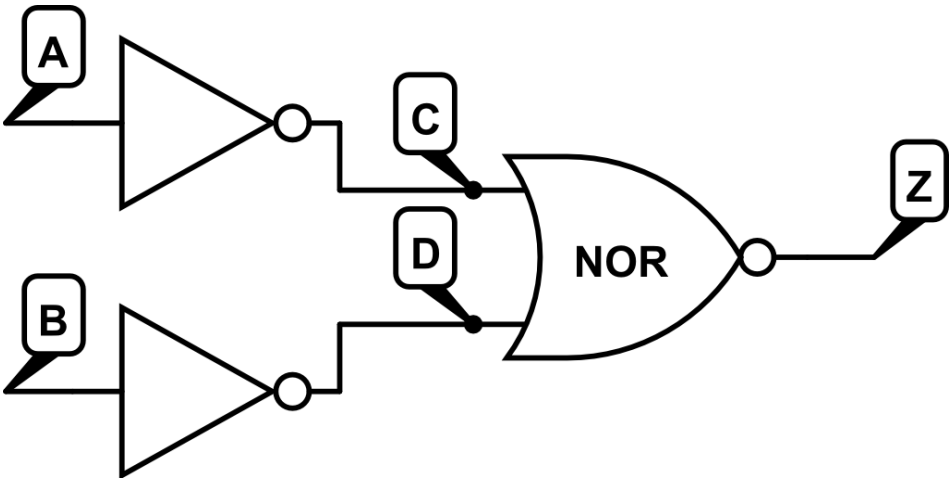
A	B	C		OUT
0	0	0		1
0	0	1		0
0	1	0		1
0	1	1		0
1	0	0		1
1	0	1		0
1	1	0		0
1	1	1		0

3.6:

Remember to draw the interconnected logic gates that the transistors are implementing – make certain your truth table still works when you are done!

A	B		C	D	Z
0	0		1	1	0
0	1		1	0	0
1	0		0	1	0
1	1		0	0	1

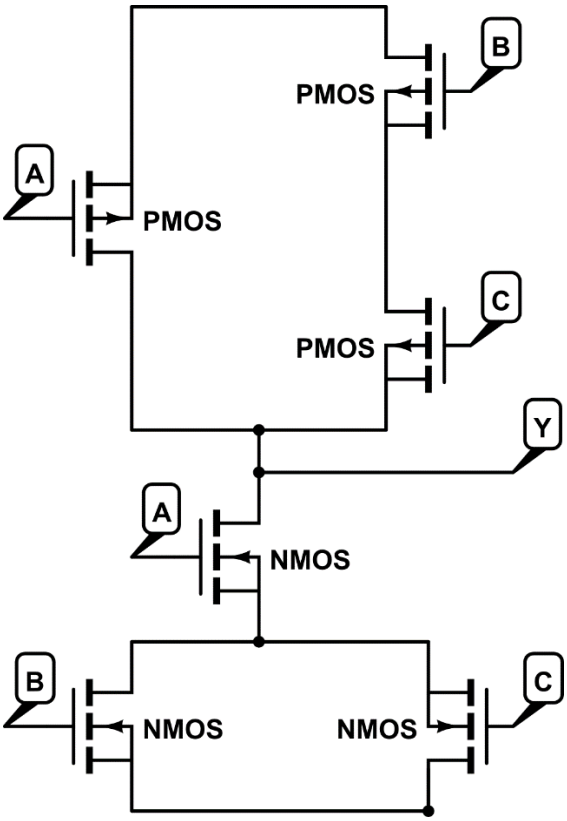
Based off this truth table, Z = A.B



3.7: The major flaw in this circuit is that it is shorted when A = 1 and B = 0 or vice versa.

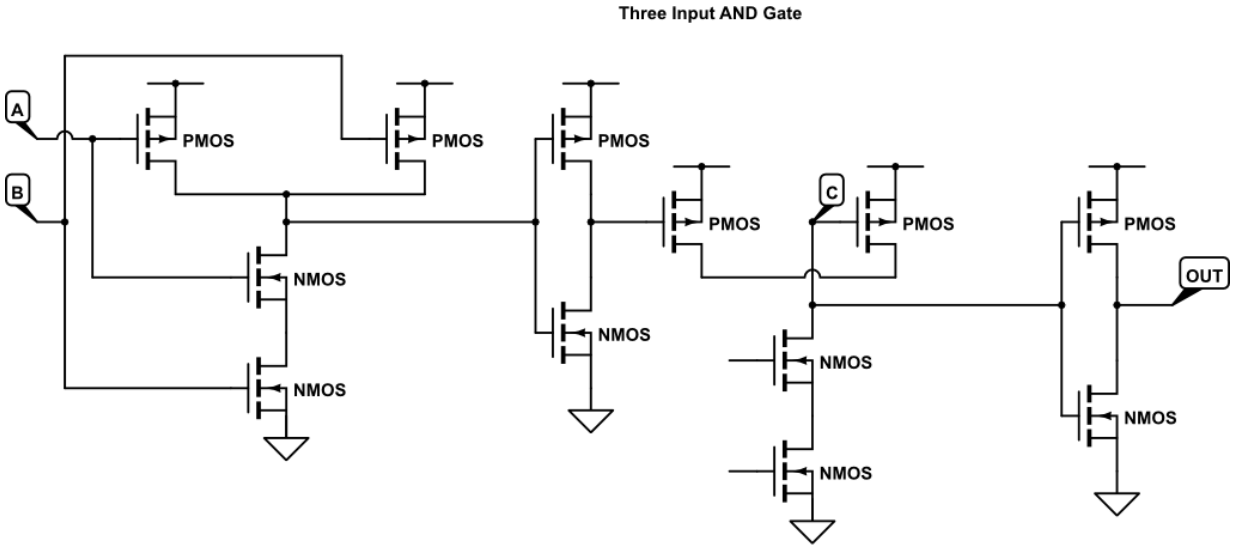
3.8:

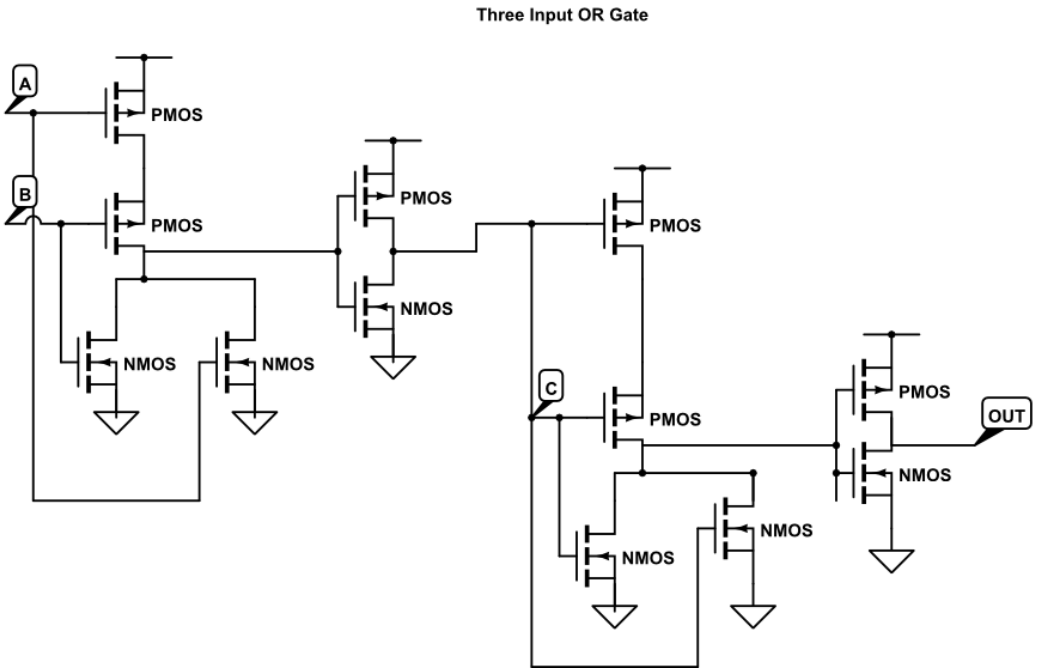
Remember: In the international edition of text, the function provided in the question is wrong. Use: Y=NOT(A AND (B OR C))



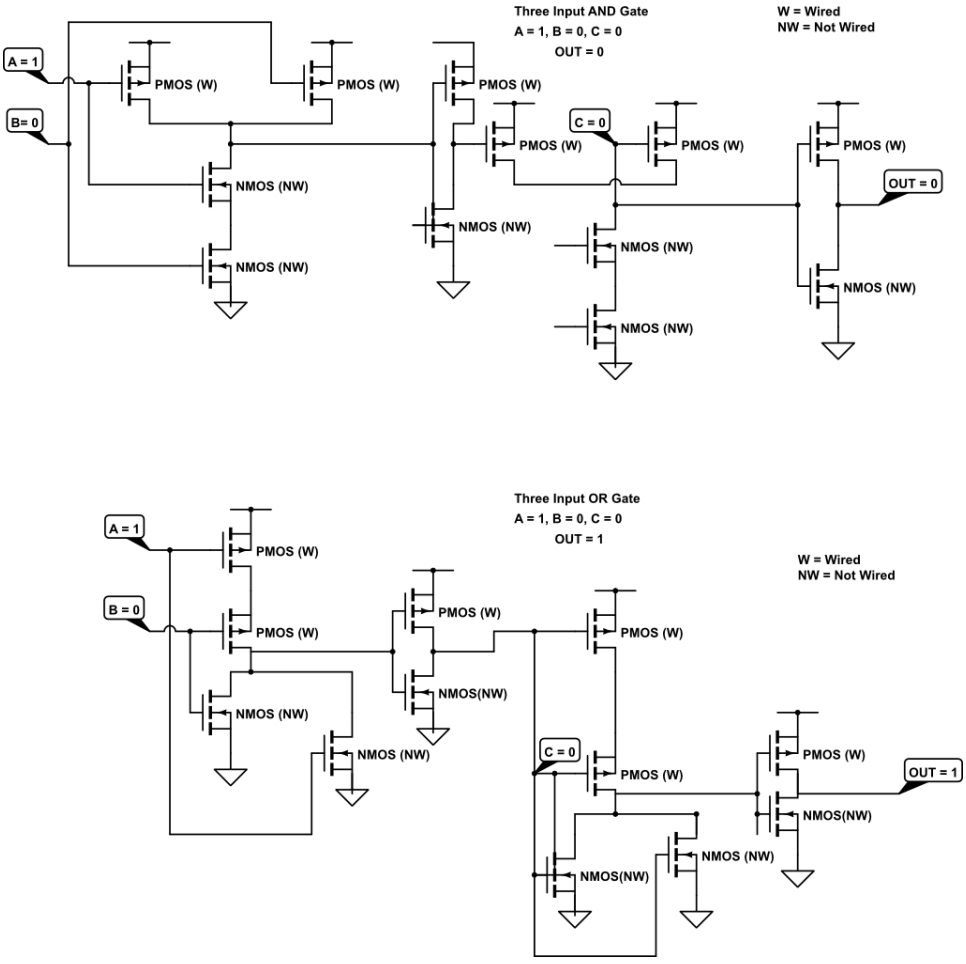
3.11:

(a)

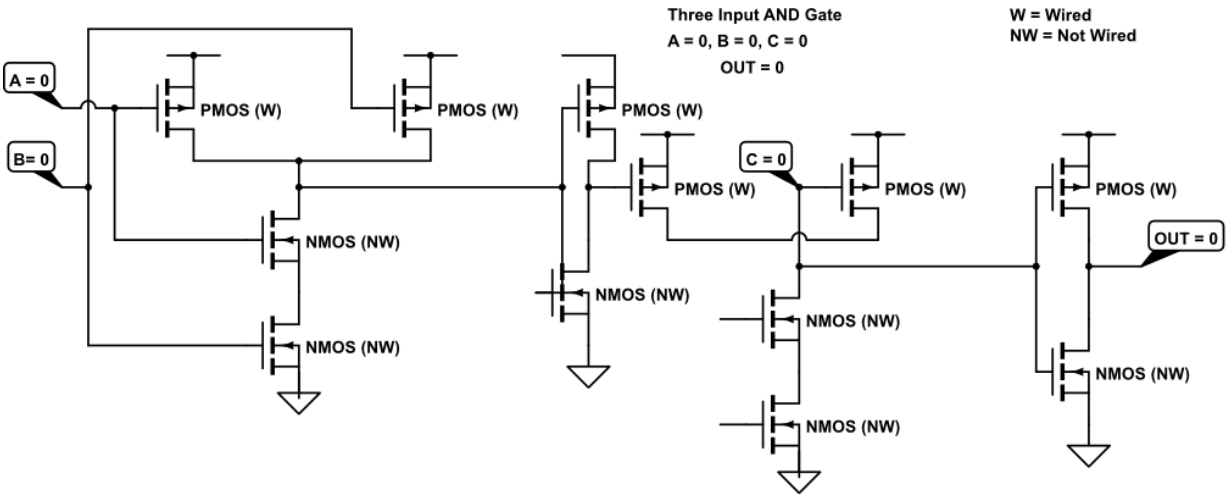

Three Input AND Gate
A
B
C
PMOS
NMOS
OUT

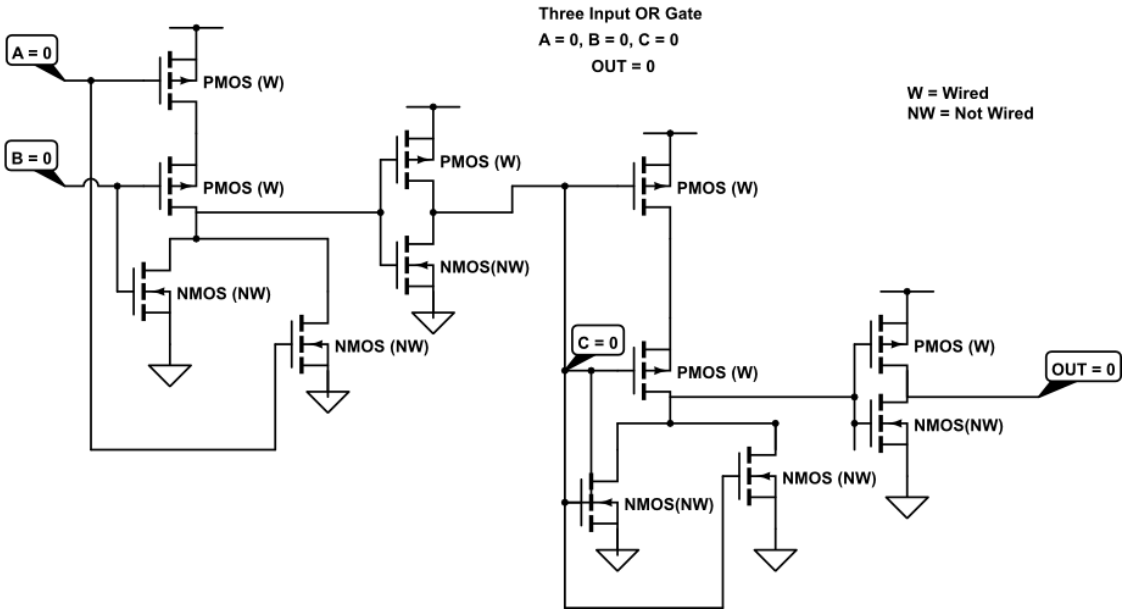

Three Input OR Gate
A
B
C
PMOS
NMOS
OUT

(b1)

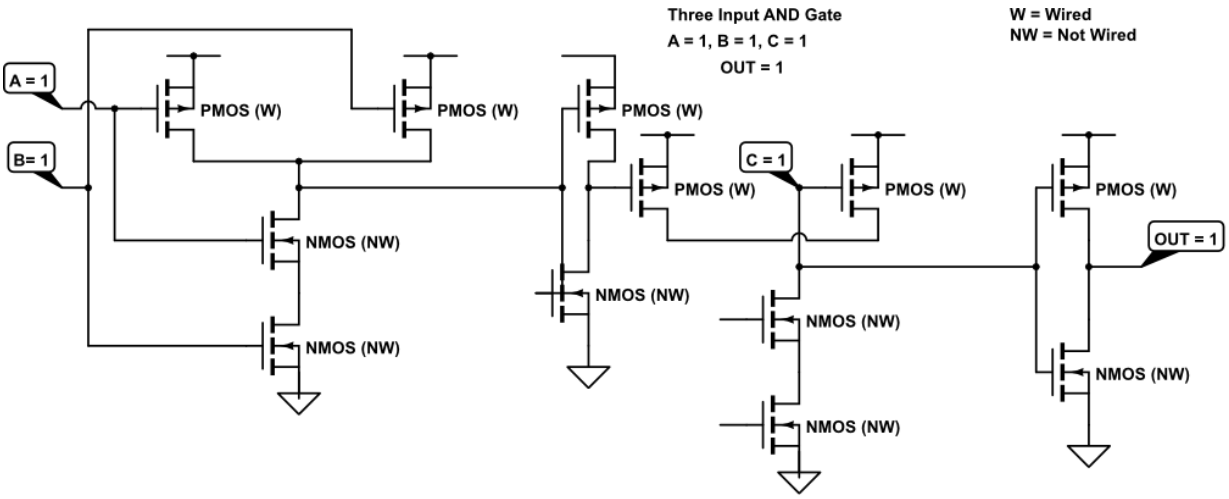

Three Input AND Gate
A = 1, B = 0, C = 0
OUT = 0
W = Wired
NW = Not Wired
A = 1
B= 0
C = 0
OUT = 0
PMOS (W)
NMOS (NW)
Three Input OR Gate
A = 1, B = 0, C = 0
OUT = 1
W = Wired
NW = Not Wired
A = 1
B = 0
C = 0
OUT = 1
PMOS (W)
NMOS(NW)
NMOS (NW)

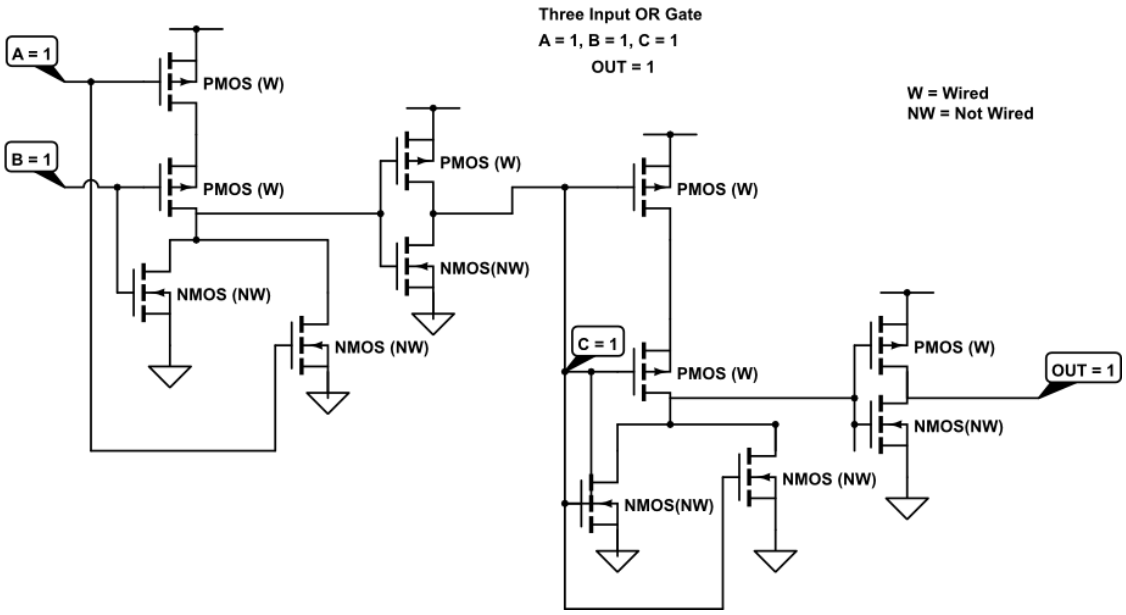
(b2)


Three Input AND Gate
A = 0, B = 0, C = 0
OUT = 0
W = Wired
NW = Not Wired
A = 0
B= 0
C = 0
PMOS (W)
NMOS (NW)
OUT = 0


Three Input OR Gate
A = 0, B = 0, C = 0
OUT = 0
W = Wired
NW = Not Wired
A = 0
B = 0
C = 0
PMOS (W)
NMOS (NW)
NMOS(NW)
OUT = 0

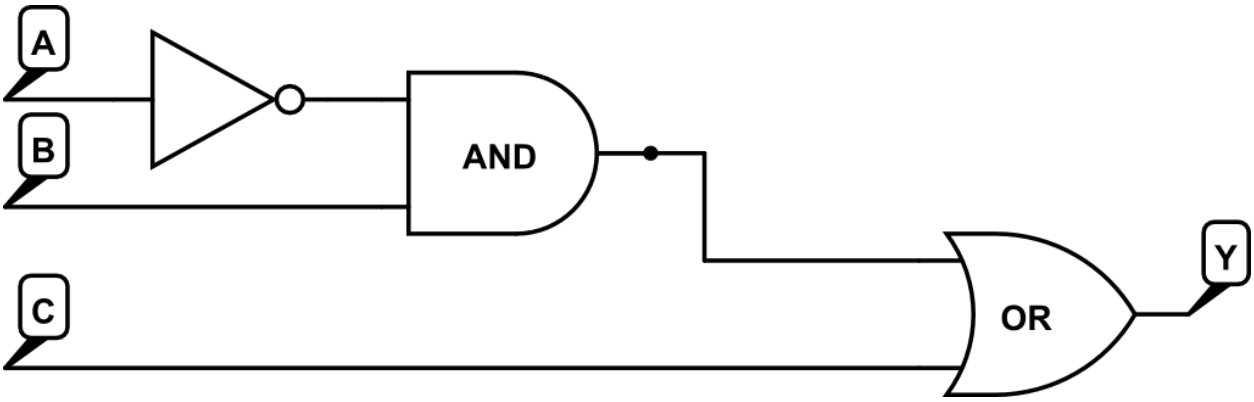
(b3)





Custom Problem #1 (logic down to transistors):

(a)



(b)

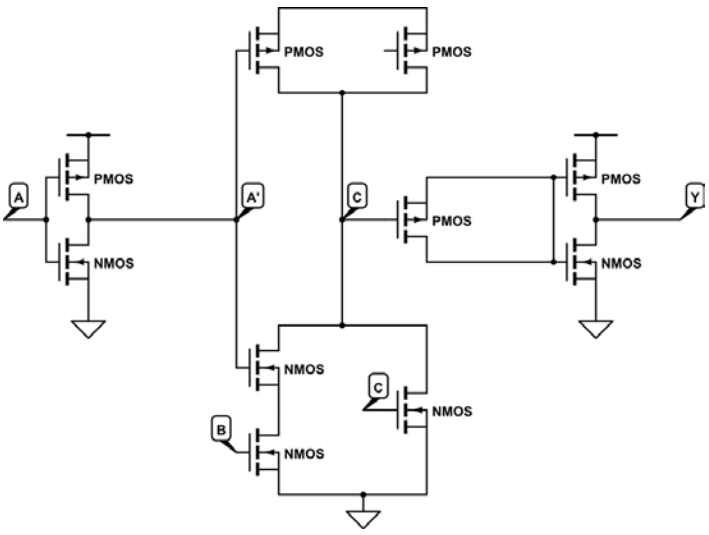
Or gate delay = 3 ns

AND gate delay = 2.5 ns

Inverter = 1 ns

Total delay = 1 + 2.5 + 3 = 6.5 ns

(c)



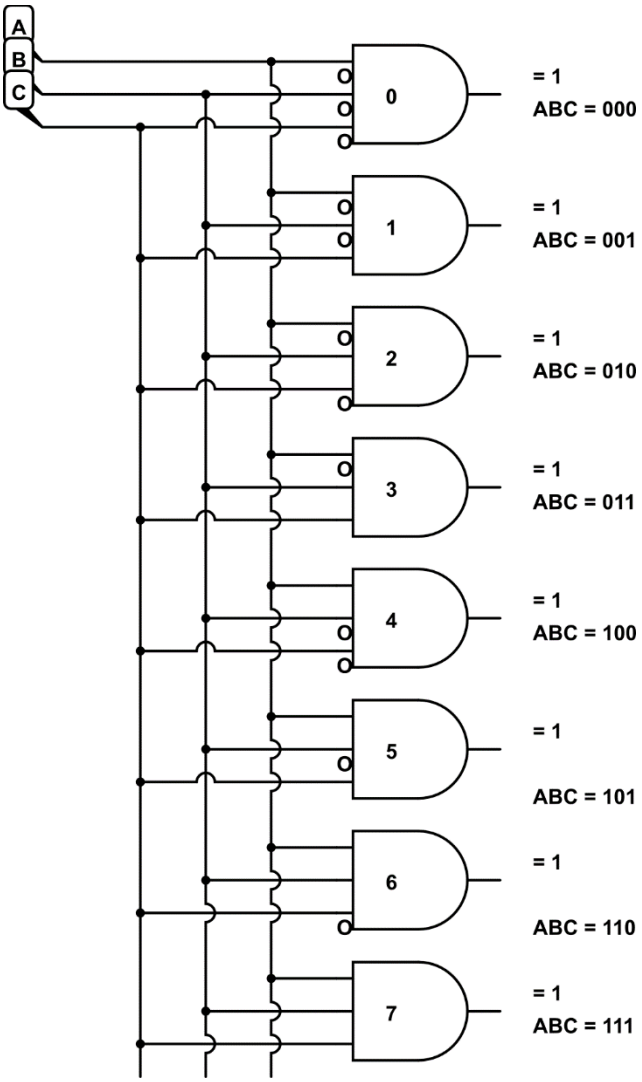
(d)

Input A	Input B	Input C	Output Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Part 2) Combinational Logic Circuits

3.12:

2^3 = 8 inputs



3.13: 2^5 = 32

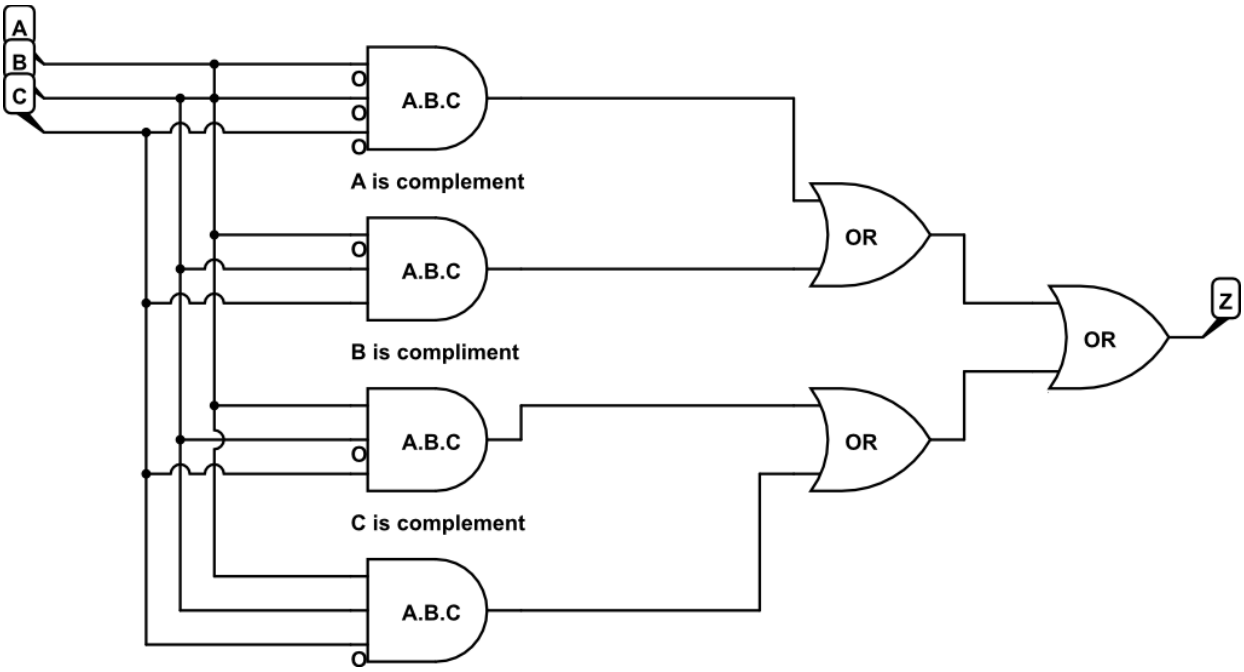
3.14: One output with four select lines.

3.15:

C_{in}				0
A	0	1	1	1
B	1	0	1	1
S	0	0	1	0
C_{out}	1	1	1	1

The sum is not equal because the addition requires 5 bits instead of 4 bits.

3.16:



3.22:

mux A C.S	Mux C C.S	A mux output	B mux output	Overall output
0	0	C	A	A
1	0	D	B	B
0	1	C	A	C
1	1	D	B	D

Inputs = A,B,C,D.

3.23:

A	B	C		Z
0	0	0		0
0	0	1		0
0	1	0		0
0	1	1		0
1	0	0		0
1	0	1		0
1	1	0		0
1	1	1		0

3.25:

(a) 3

(b) 3

(c) 4*3 = 12

(d)8*12 = 96

3.28:

(a) 3

(b) 3

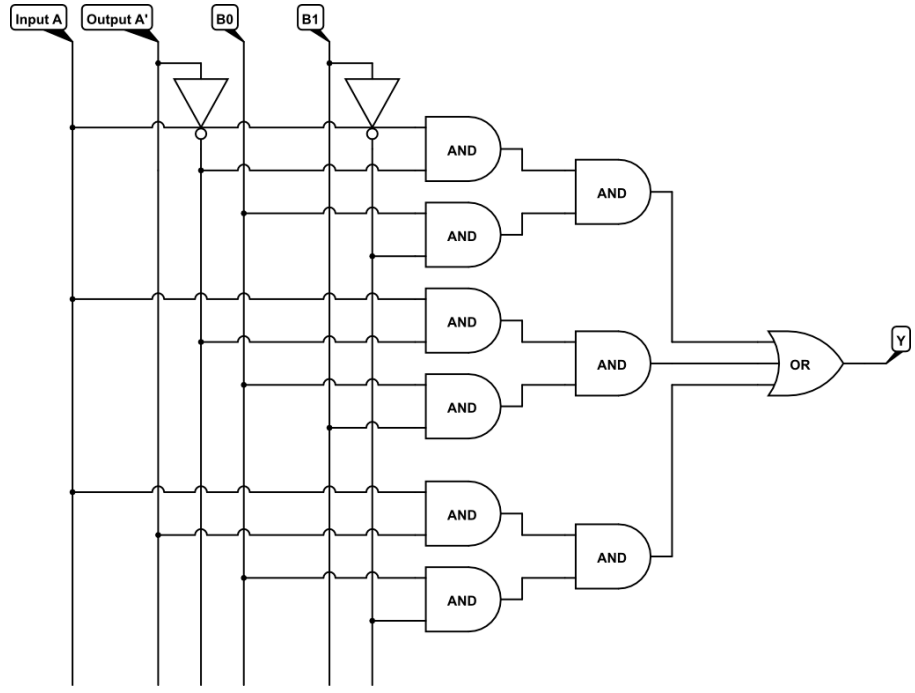
(c) 9

(d) 4

(e)

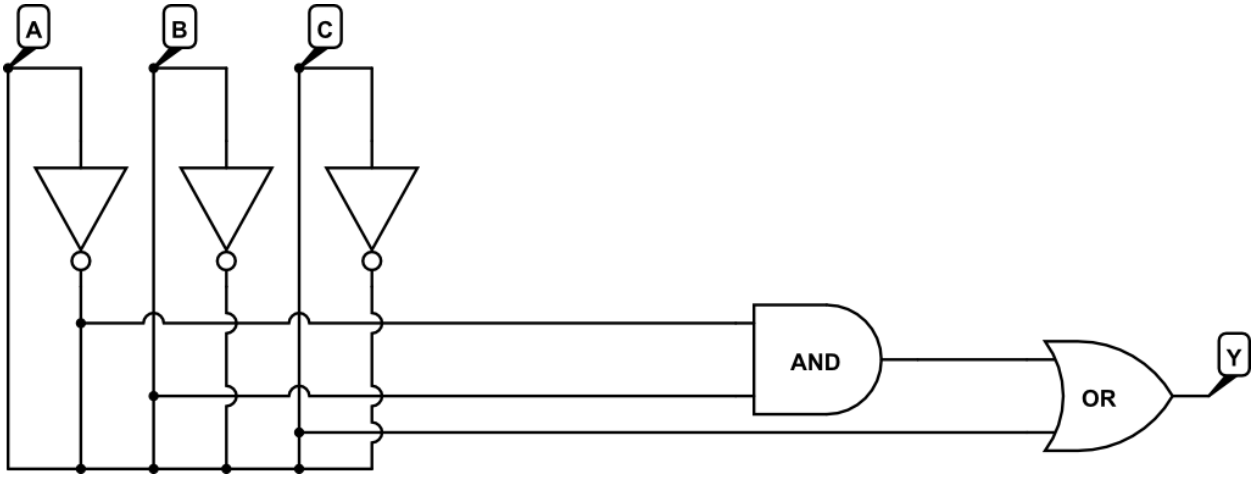
Input A	output A’	B0	B1	Y0	Y1	Y2	Y3
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	1	0	0	0
0	1	1	0	0	1	0	0
0	1	1	1	1	1	0	0
1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	0	1	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	1	1	0	0
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

(f)



Custom Problem #2 (PLA vs. Boolean Algebra)

(a)



(b)

Or gate delay = 3 ns

AND gate delay = 2.5 ns

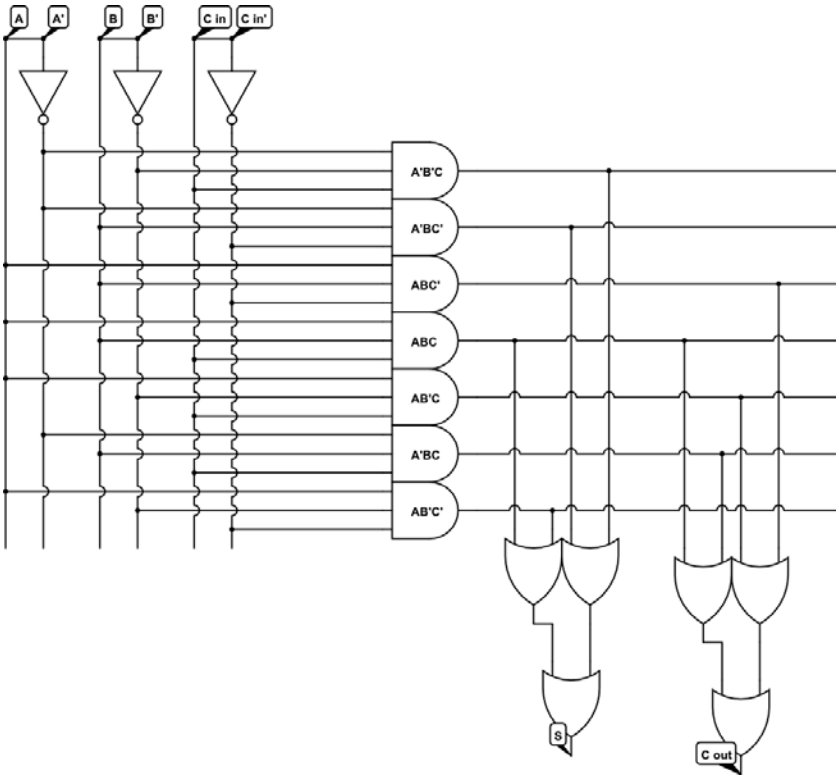
Total Gate Delay = 2.5 + 3 = 5.5 ns

(c) The PLA implementation is better because is utilizes less gate delay than the three basic gates implementation.

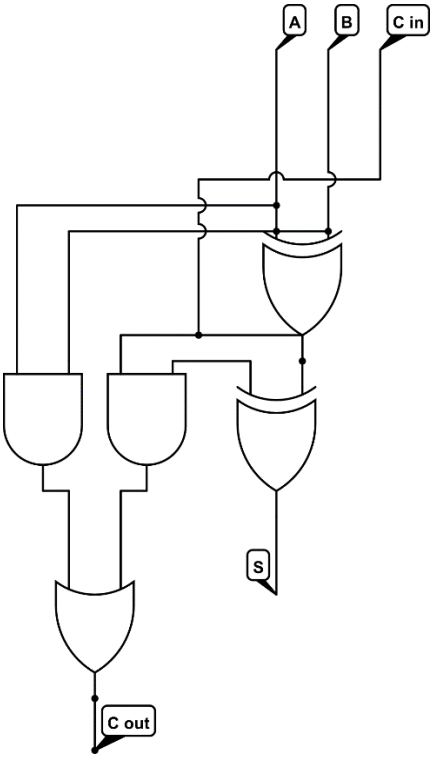
Custom Problem #3 (Multiple Output PLAs)

Using the lecture notes, locate the "Full Adder circuit" truth table with inputs A, B, C and outputs S and C-out.

(a)



(b)



I chose to implement this more efficient design utilizing XOR gates because it is the logical equivalent of two individual AND gates plus one OR gate. Also, this design does not require the use of digital NOTs.