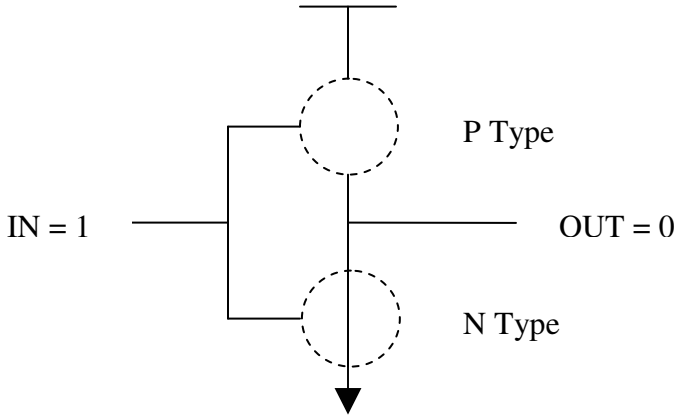
F.3 Chapter 3 Solutions

3.1

	N-Type	P-Type
Gate=1	closed	open
Gate=0	open	closed

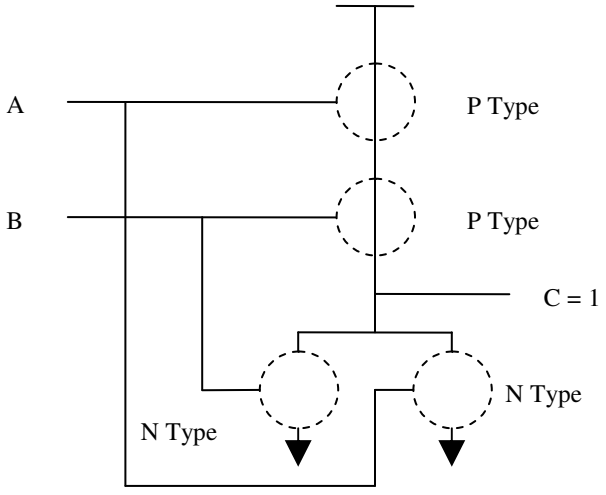
3.2

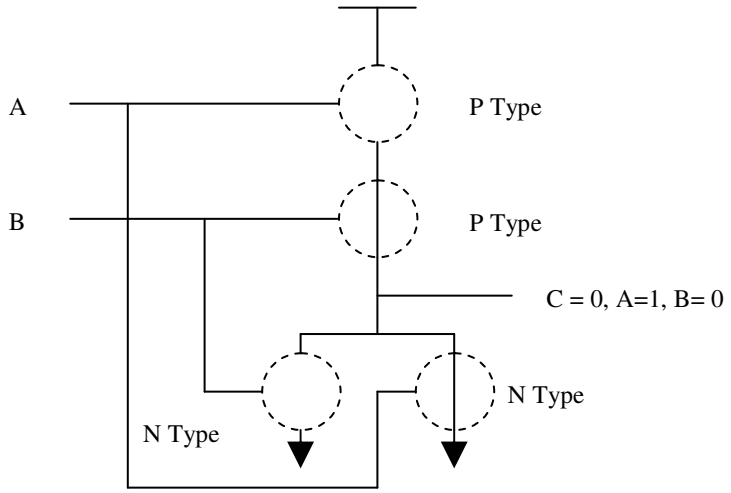


3.3 There can be 16 different two input logic functions.

3.4

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





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

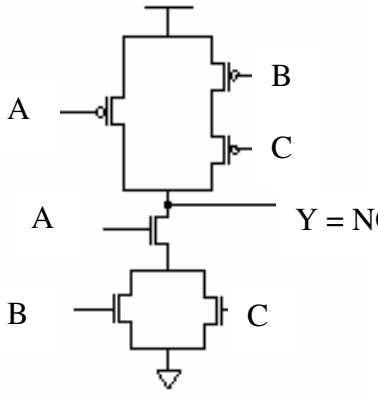
3.6 C = A'; D = B'; Z = (C+D)' = (A'+B')' = A . B

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

3.7 There is short circuit (path from Power to Ground) when either A = 1 and B = 0 or A = 0 and B = 1.

3.8 Correction: Please correct the logic equation to

Y = NOT (A AND (B OR C))



3.9

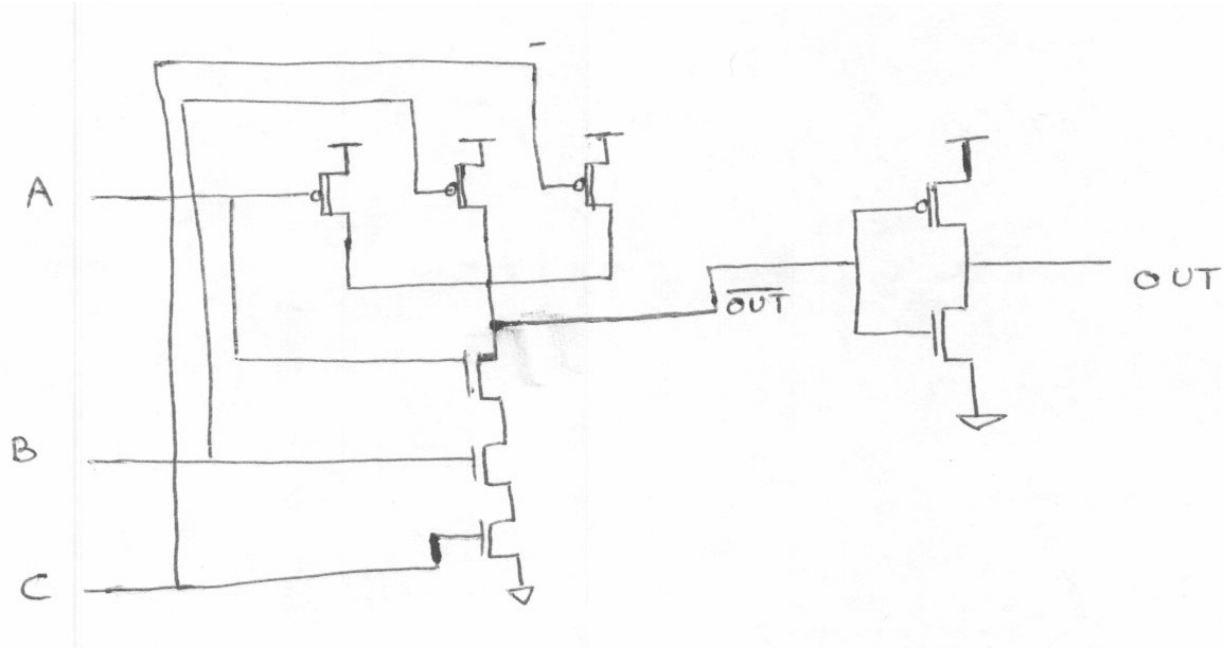
A	B		NOT(NOT(A) OR NOT(B))
0	0		0
0	1		0
1	0		0
1	1		1

AND gate has the same truth table.

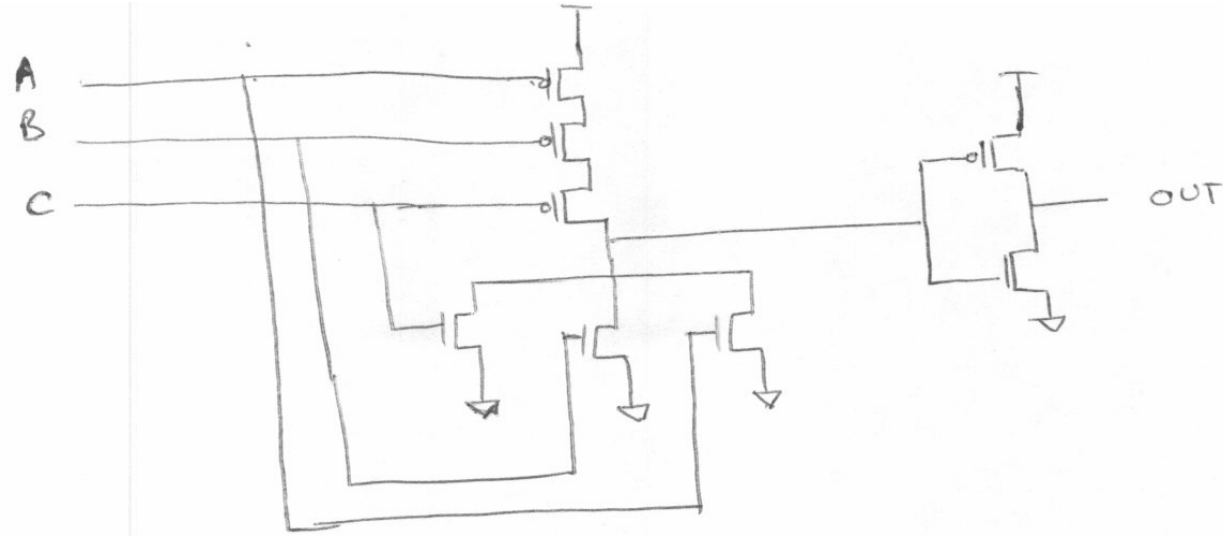
3.10

A	B		A NOR B
0	0		1
0	1		0
1	0		0
1	1		0

3.11 a. Three input And-Gate

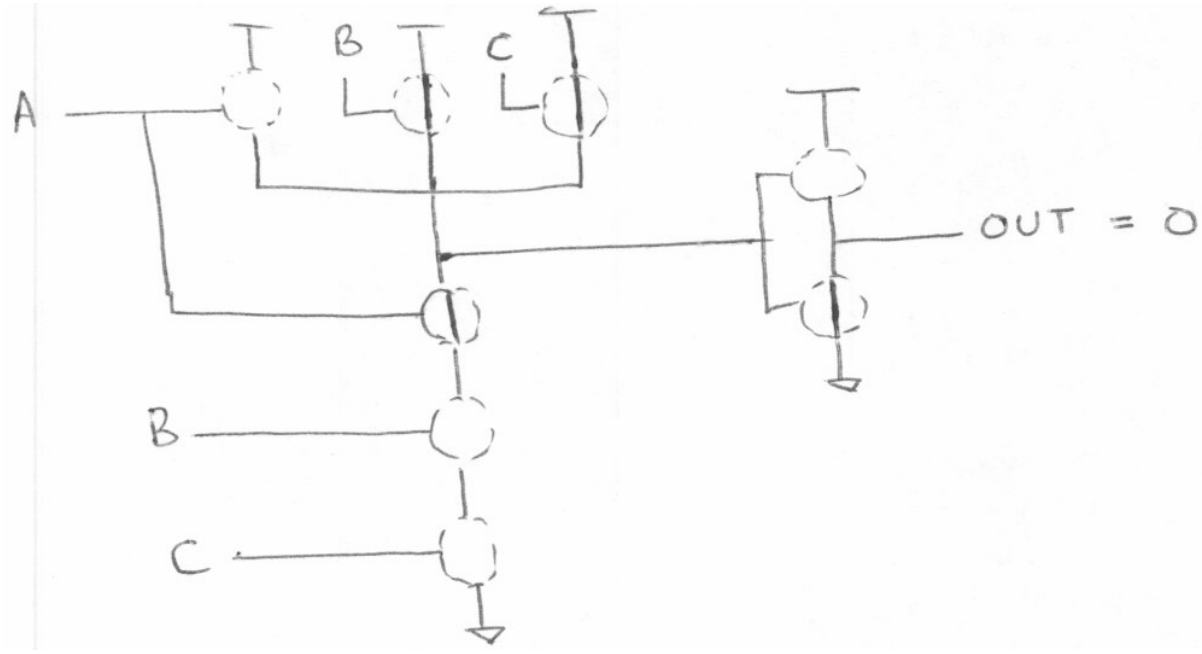


Three input OR-Gate

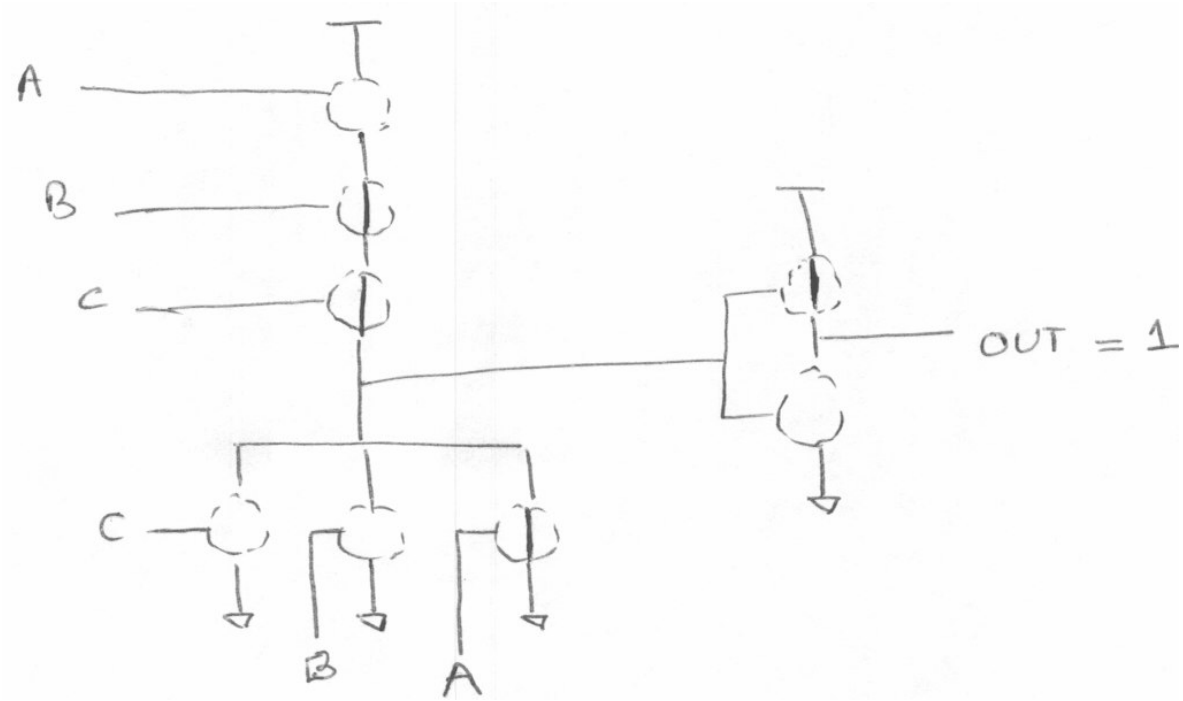


b. (1) A = 1, B = 0, C = 0.

AND Gate

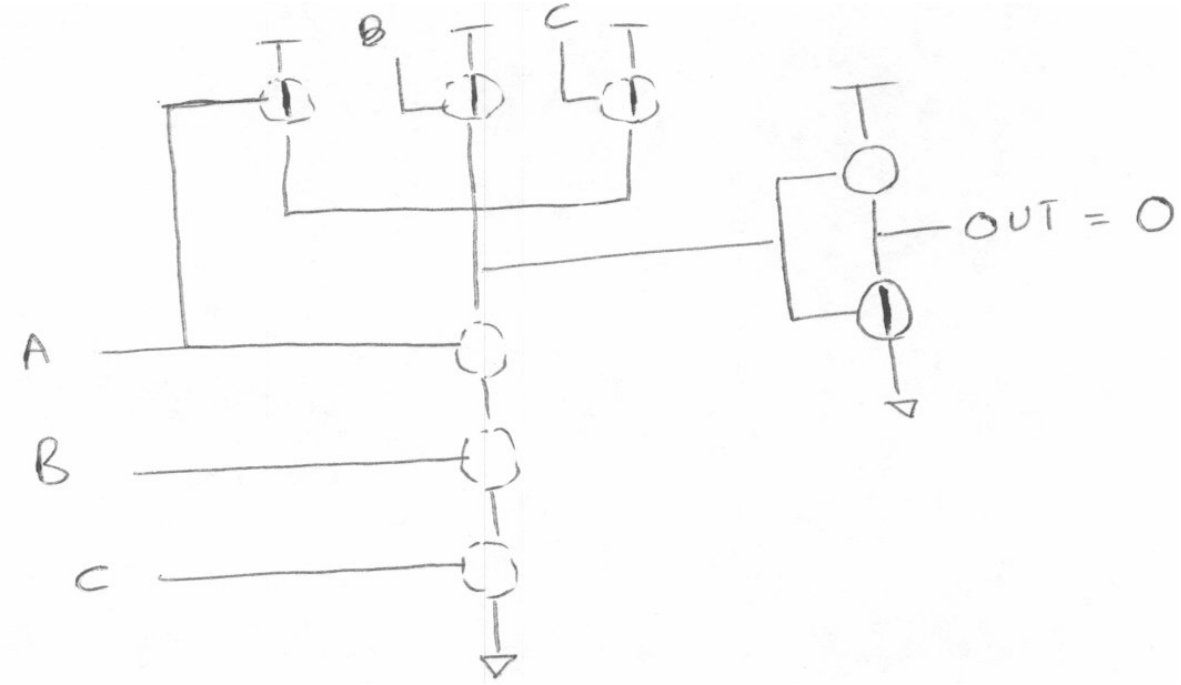


OR Gate

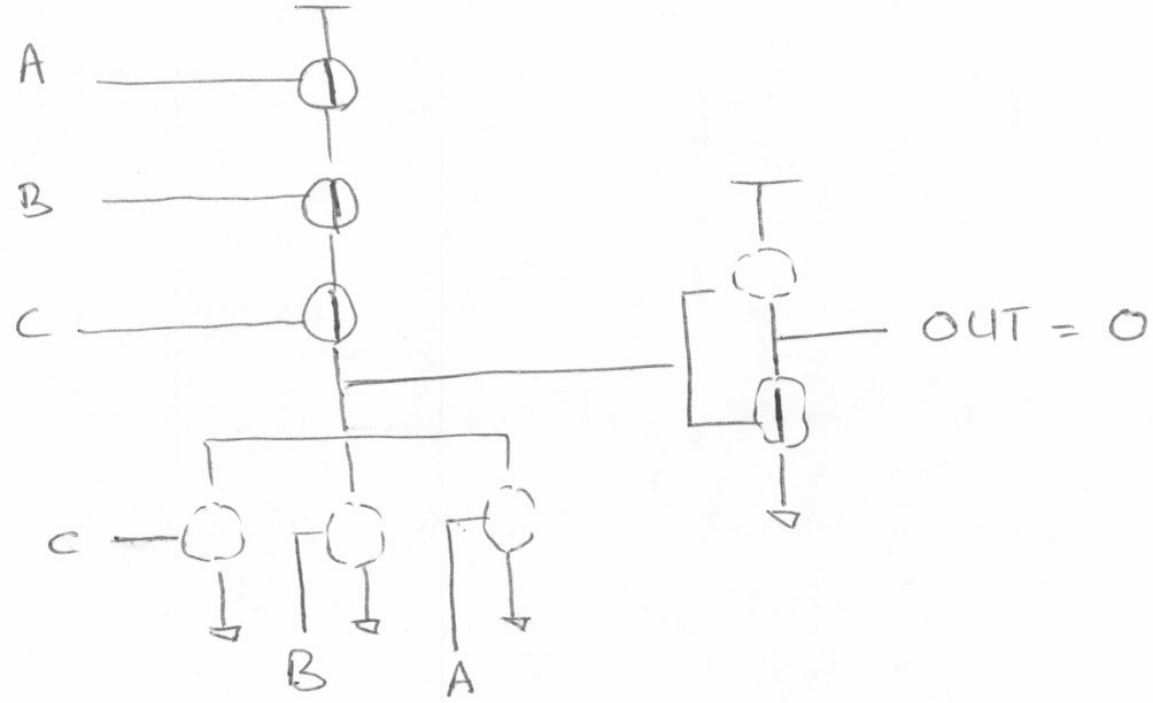


b. (2) A = 0, B = 0, C = 0

AND Gate

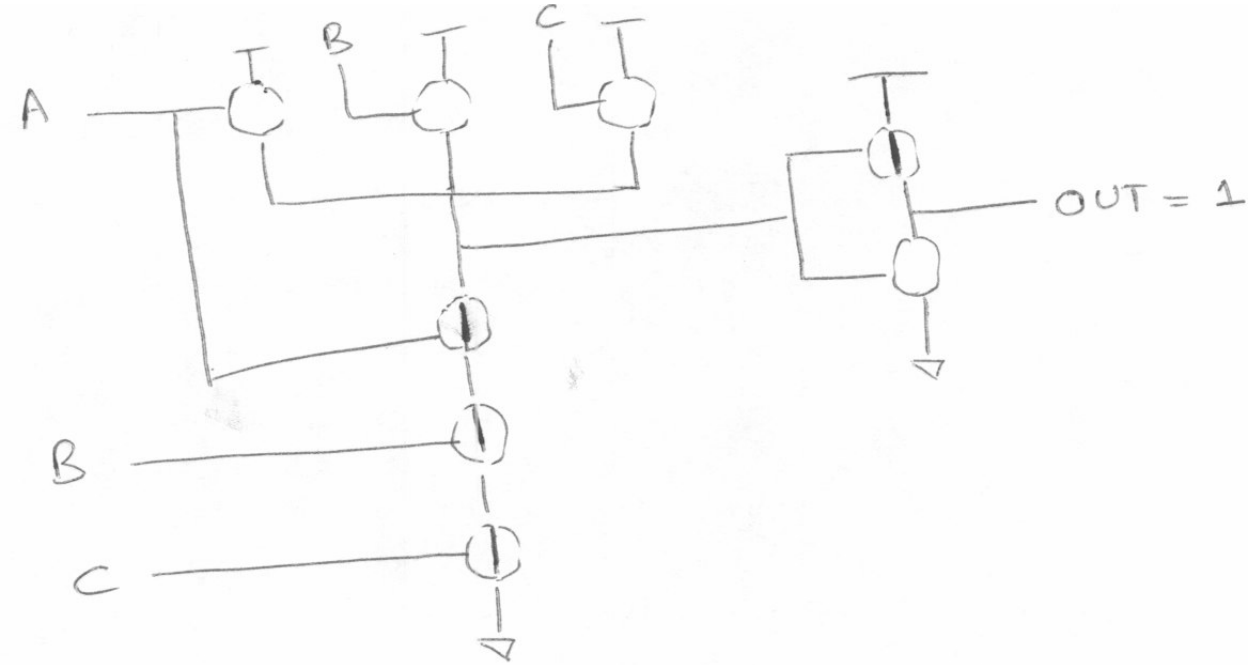


OR Gate

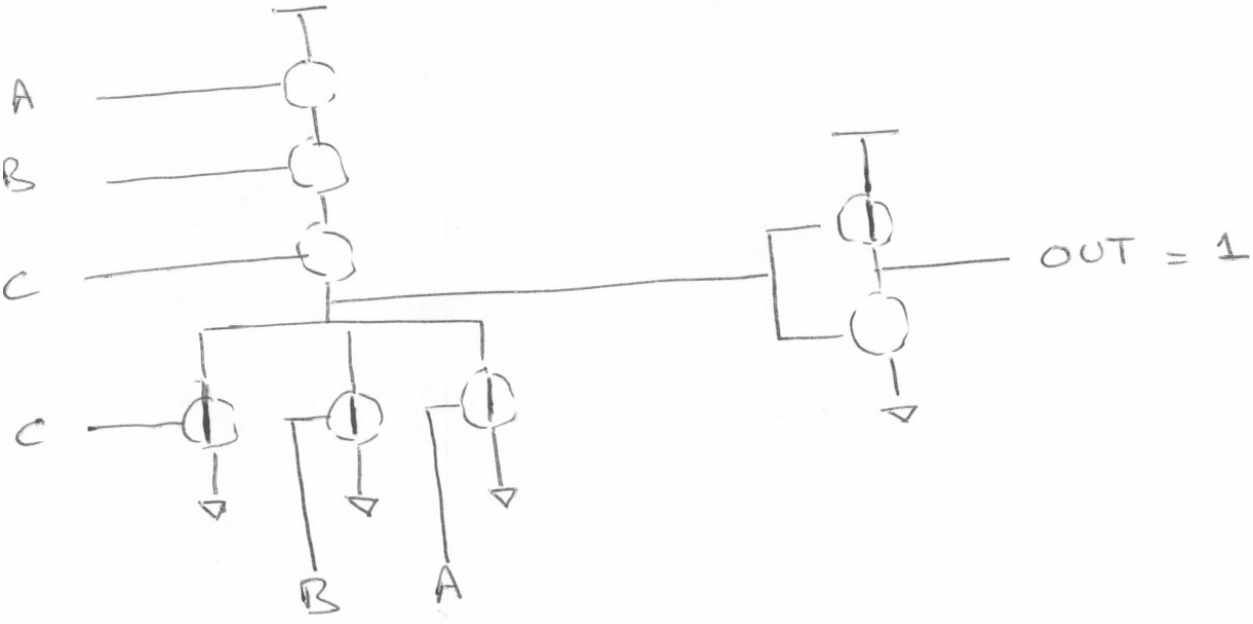


b. (3) A = 1, B = 1, C = 1

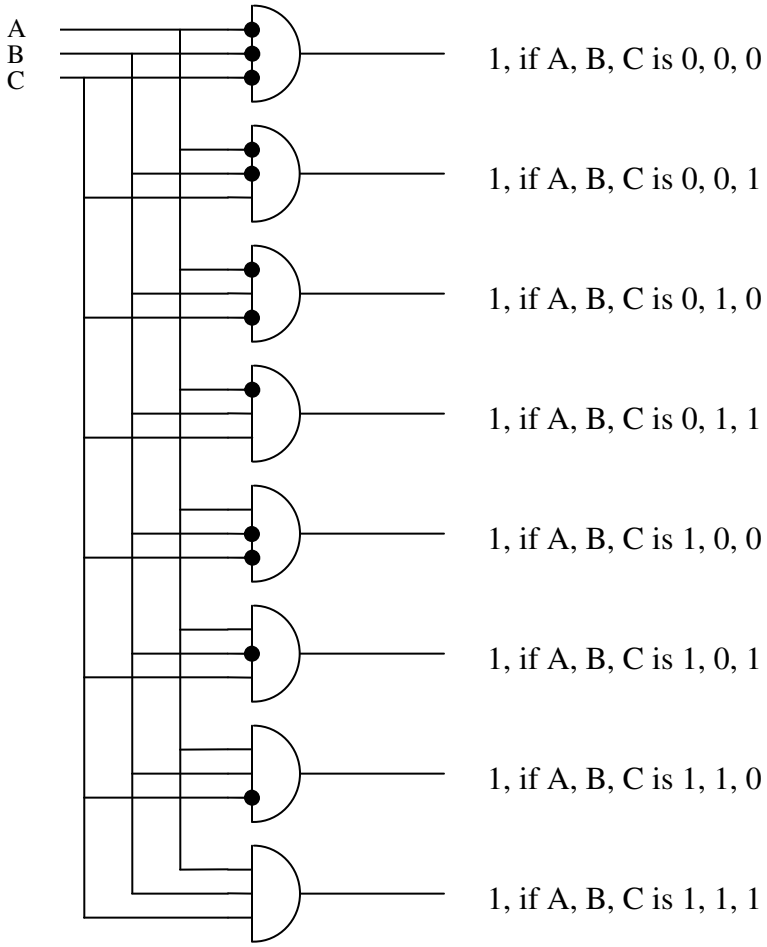
AND Gate



OR Gate



3.12



3.13 A five input decoder will have 32 output lines.

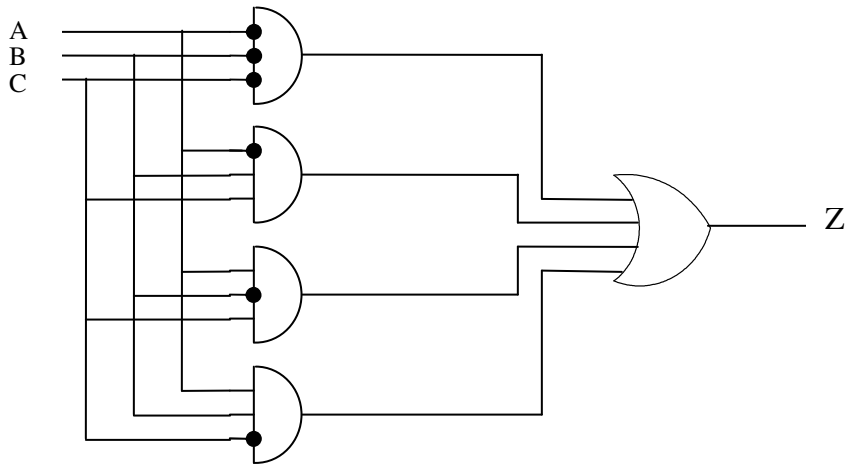
3.14 A 16 input multiplexer will have one output line (ofcourse!). It will have 4 select lines.

3.15

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

A = 7, B = 11, A + B = 18.
In the above calculation, the result (S) is 2 !! This is because 18 is too large a number to be represented in 4 bits. Hence there is an overflow - Cout[3] = 1.

3.16 Z = XNOR(A, B, C)

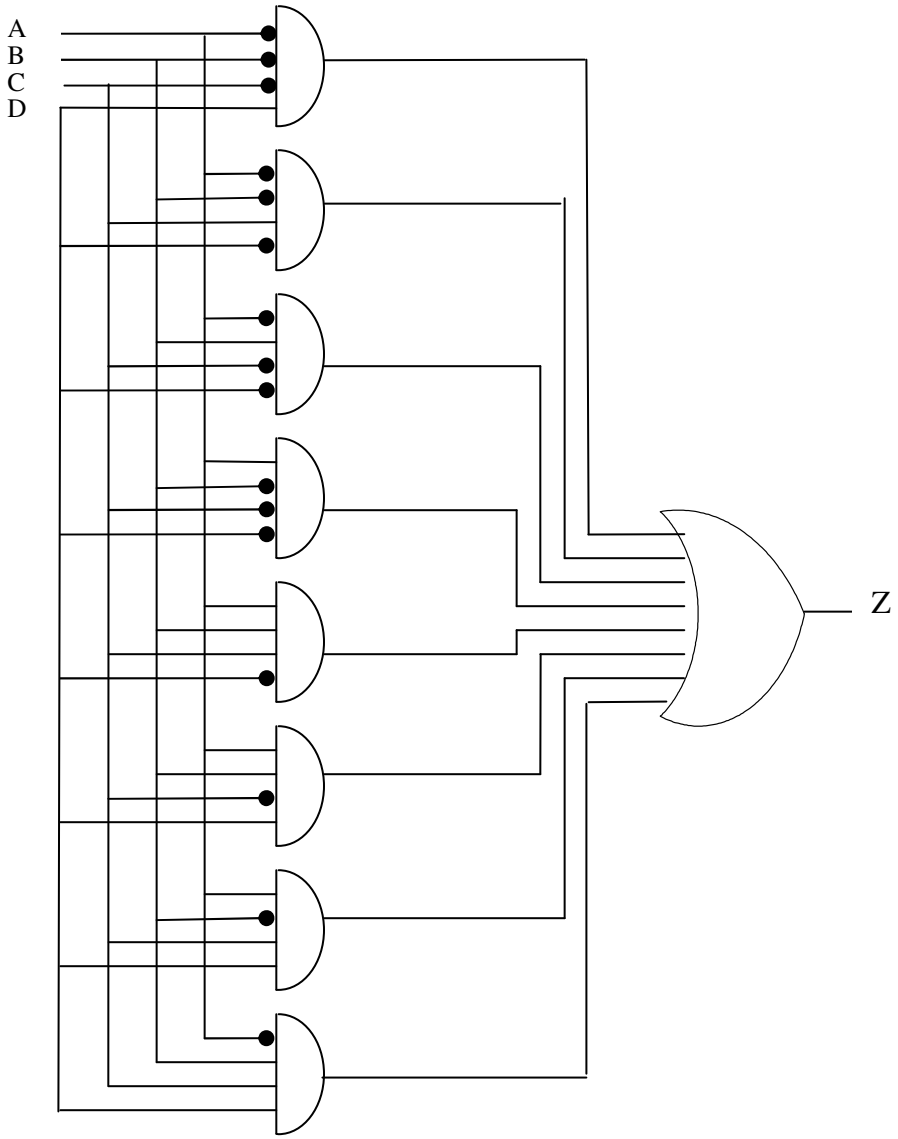


3.17 (a) The truth table will have 16 rows. Here is the truth table for Z = XOR (A, B, C, D). Any circuit with at least seven input combinations generating 1s at the output will work.

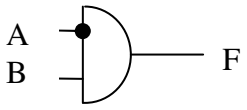
A	B	C	D		Z
0	0	0	0		0
0	0	0	1		1
0	0	1	0		1
0	0	1	1		0
0	1	0	0		1
0	1	0	1		0
0	1	1	0		0
0	1	1	1		1
1	0	0	0		1
1	0	0	1		0
1	0	1	0		0
1	0	1	1		1
1	1	0	0		0
1	1	0	1		1
1	1	1	0		1
1	1	1	1		0

Z = XOR (A,B,C,D)

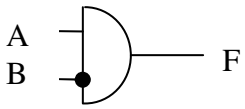
(b)



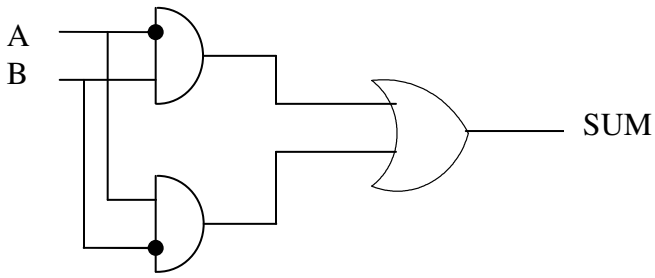
3.18. (a)



(b)



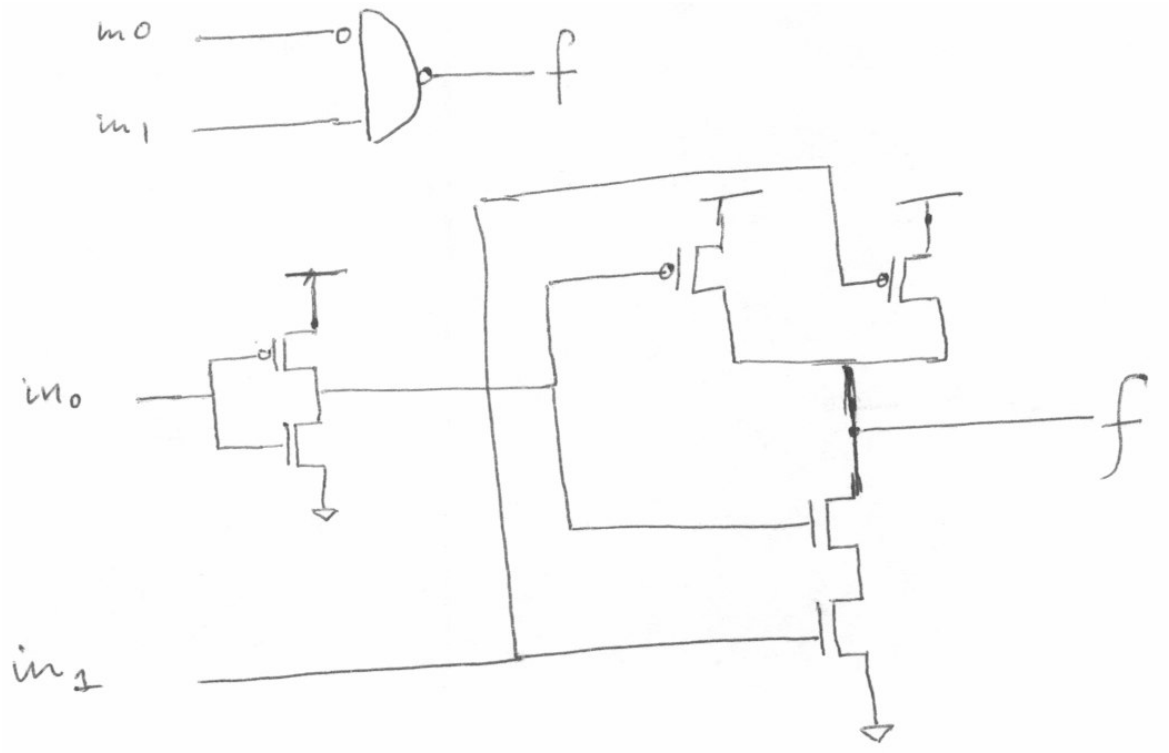
(c)



(d) No. The carry is not being generated/propagated.

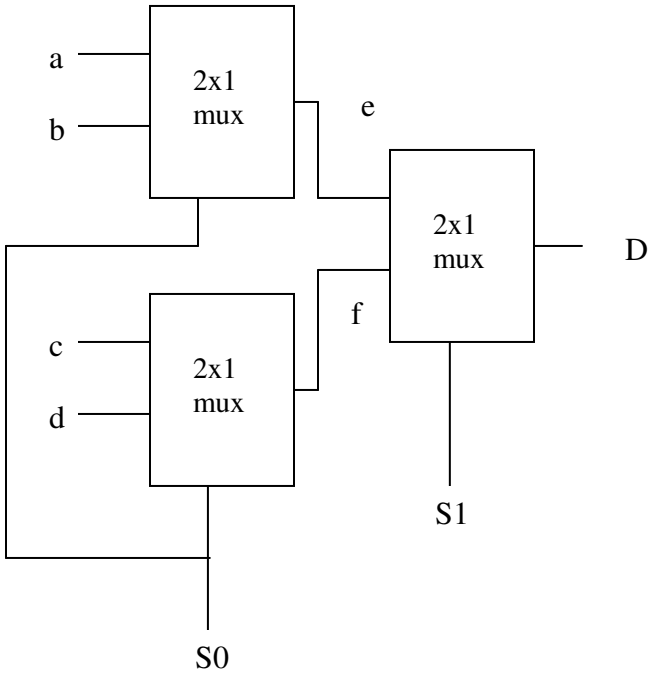
3.19 Figure 3.36 is a simple combinational circuit. The output value depends ONLY on the input values as they currently exist. Figure 3.37 is an R-S Latch. This is an example of a logic circuit that can store information. That is, if A, B are both 1, the value of D depends on which of the two (A or B) was 0 most recently.

3.20



3.21 $2 * 2^{14} = 2^{15} = 32768$ nibbles

3.22



S1	S0		e	f	D
0	0		a	c	a
0	1		b	d	b
1	0		a	c	c
1	1		b	d	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

3.24 (a) X=0 => S = A+B, X=1 => S = A+C

3.24 (b) Circuit diagram is same as Figure 3.39 with the following modifications: C = NOT (B), Carry-in = X

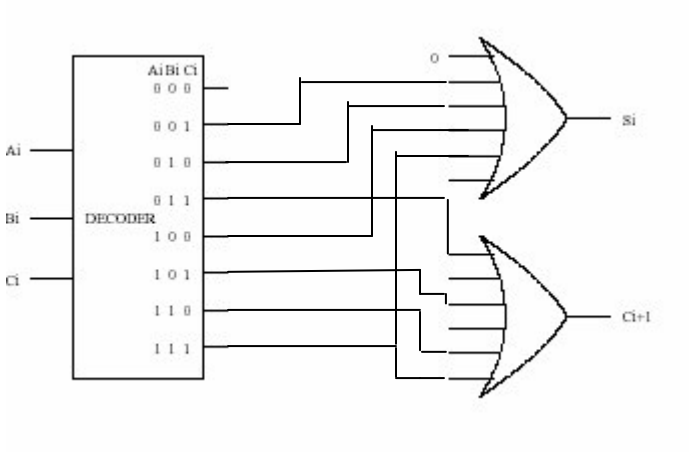
3.25 (a) 3 gate delays

3.25 (b) 3 gate delays

3.25 (c) 3*4 = 12 gate delays

3.25 (d) 3*32 = 96 gate delays

3.26



A	B	C		Si	Ci
0	0	0		0	0
0	0	1		1	0
0	1	0		1	0
0	1	1		0	1
1	0	0		1	0
1	0	1		0	1
1	1	0		0	1
1	1	1		1	1

3.27(a) When S=0, Z = A

3.27(b) When S=1, Z retains its previous value.

3.27(c) Yes; the circuit is a storage element.

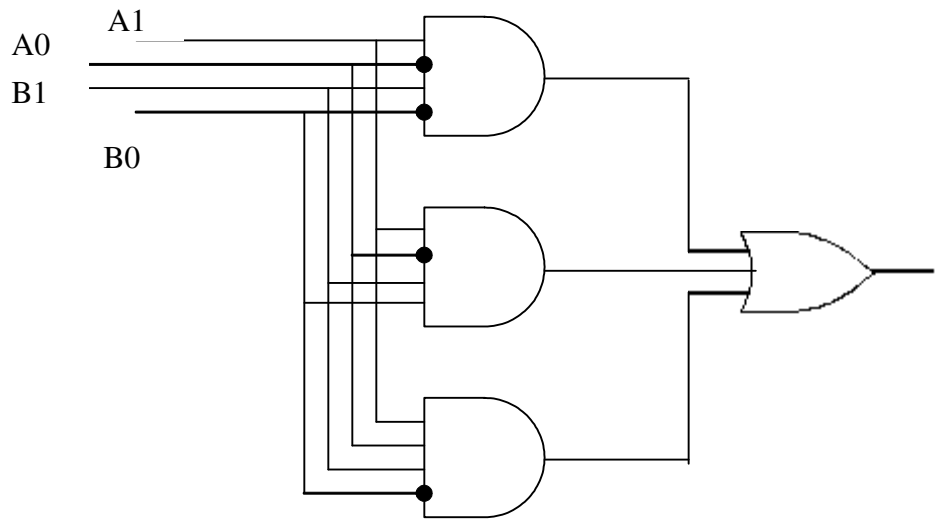
3.28

a) 3
b) 3
c) 9
d) 4

e)

A[1]	A[0]	B[1]	B[0]		Y[3]	Y[2]	Y[1]	Y[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	1		0	0	1	0
1	0	1	0		0	1	0	0
1	0	1	1		0	1	1	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

f) $Y_2 = A_1.A_0'.B_1.B_0' + A_1.A_0'.B_1.B_0 + A_1.A_0.B_1.B_0'$

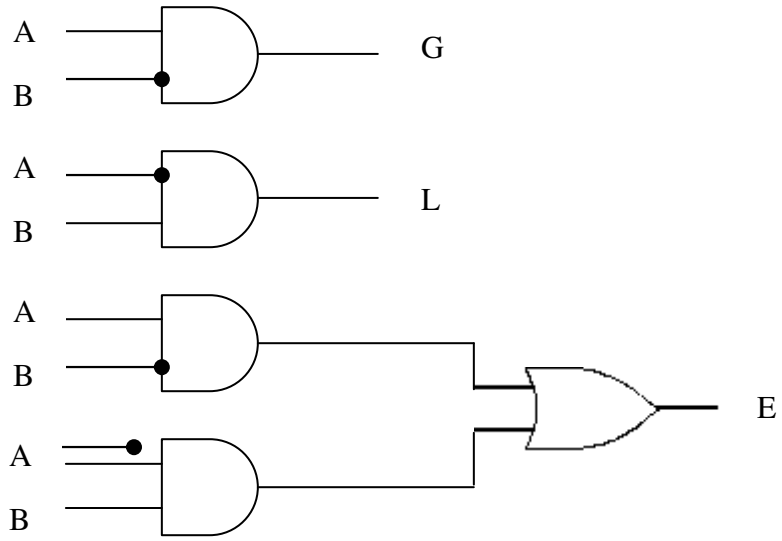


3.29 No. The original value cannot be recovered once a new value is written into a register.

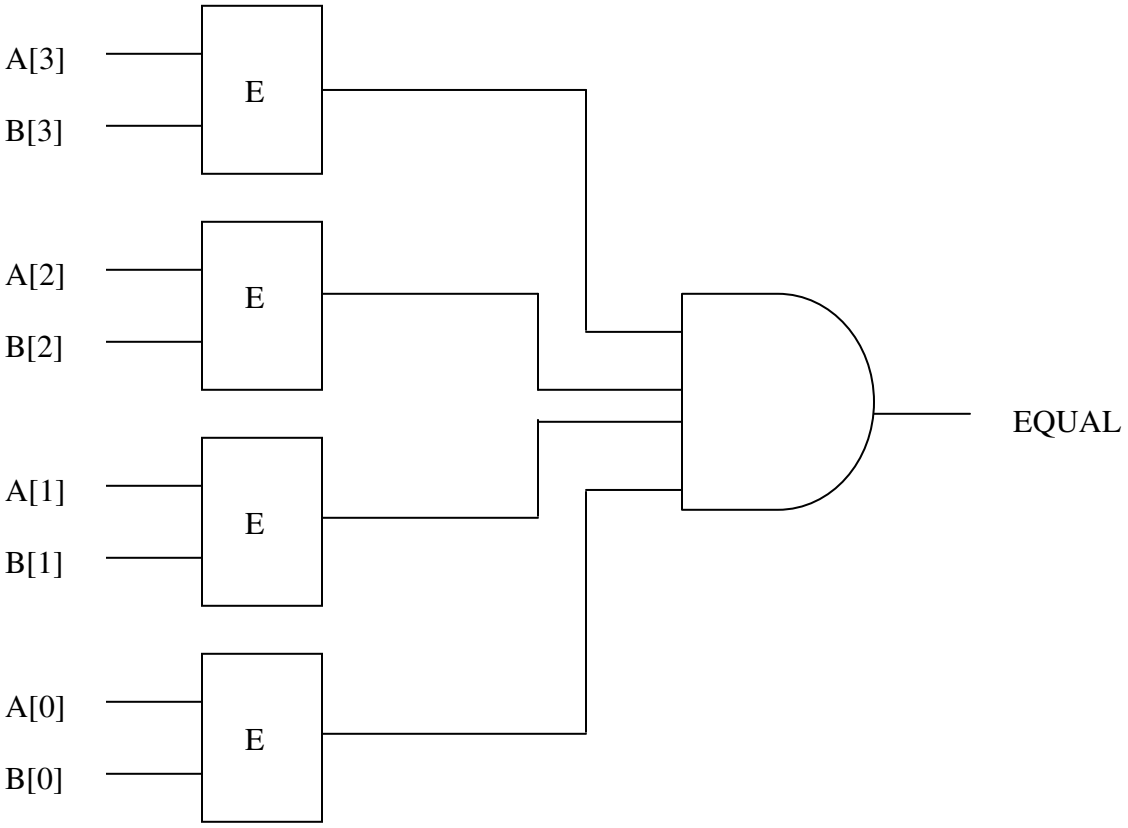
3.30
a)

A	B		G	E	L
0	0		0	1	0
0	1		0	0	1
1	0		1	0	0
1	1		0	1	0

b)



c)



3.31. 8 * (2^3) = 64 bytes

3.32 A memory address refers to a location in memory. Memory's addressability is the number of bits stored in each memory location.

3.33.(a) To read the 4th memory location, A[1,0] = 11, WE = 0

3.33.(b) A total of 6 address lines are required for a memory with 60 locations. The addressability of the memory will remain unchanged.

3.33.(c) A program counter of width 6 can address 2^6 = 64 locations. So without changing the width of the program counter, 64-60 = 4 more locations can be added to the memory.

3.34
a) 4 locations
b) 4 bits
c) 0001

3.35 Total bits of storage = 2^22 * 3 = 12582912

3.36 No effect, since it is a combinational logic circuit.

3.37 There are a total of four possible states in this lock. Any other state can be expressed as one of states A, B, C or D. For example, the state performed one correct followed by one incorrect operation is nothing but state A as the incorrect operation reset the lock.

3.38 Yes. We can have an arc from a state where the score is Texas 30, Oklahoma 28 to a state where the score is tied. This transition represents a Oklahoma player scoring a two-point shot.

3.39 No. An arc is needed between the two states.

(a) Game in Progress:

Texas *	Oklahoma
Fouls:4	Fouls: 4
73	68
First Half	
7:38	
Shot Clock : 14	

(b) Texas Win:

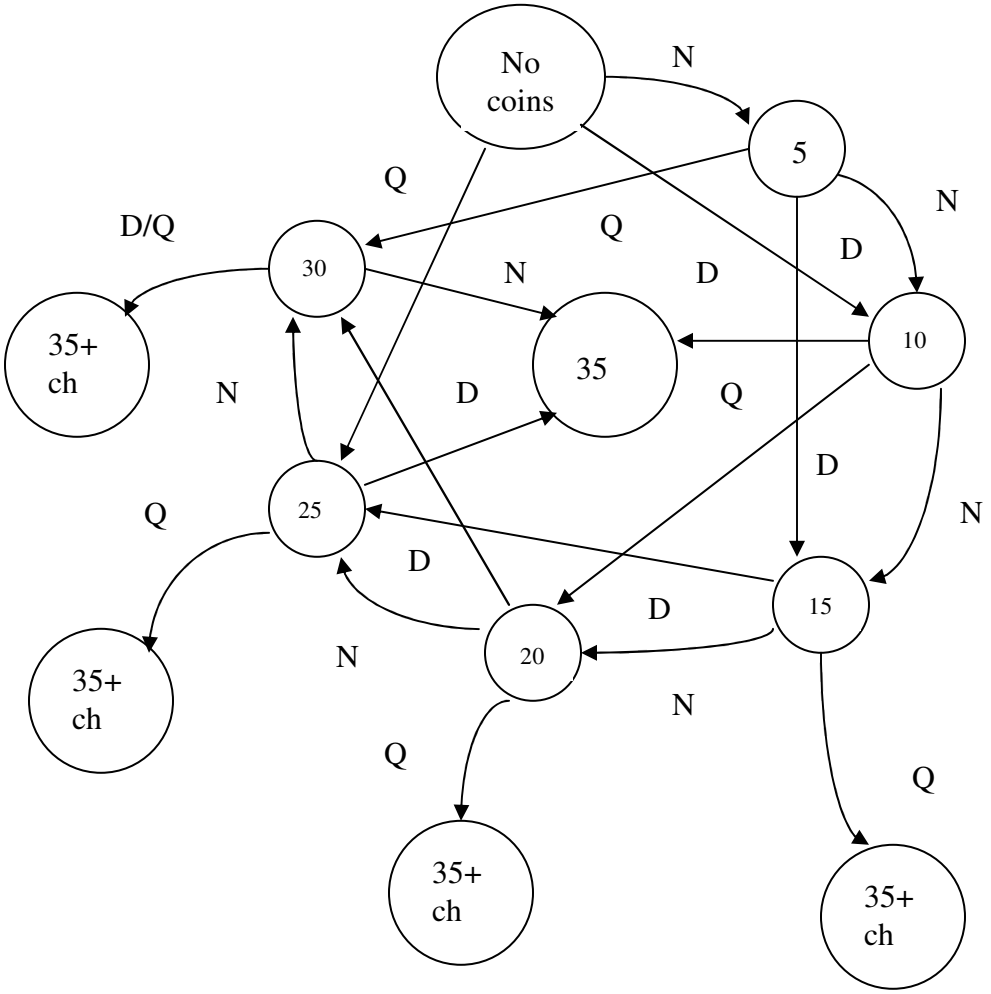
Texas *	Oklahoma
Fouls:10	Fouls: 10
85	70
Second Half	
0:00	
Shot Clock : 0	

(c) Oklahoma Win:

Texas *	Oklahoma
Fouls:10	Fouls: 10
81	90
First Half	
7:38	
Shot Clock : 0	

3.40 Left as an exercise. For each board state, come up with a transition to the best possible next state.

3.41



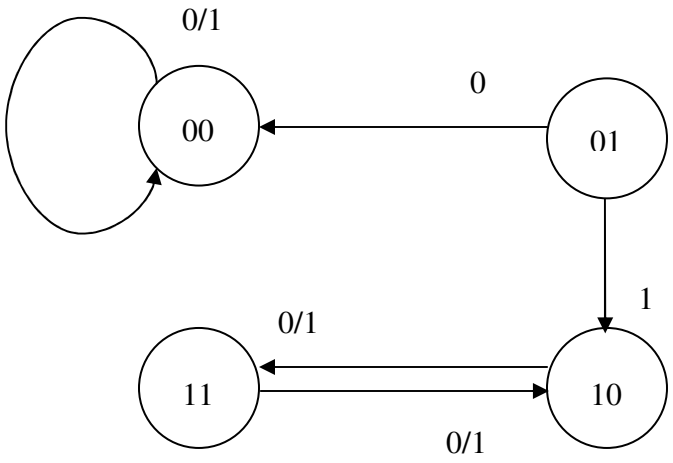
3.42 Since there are 3 states (states 01, 10 and 11) in which lights 1 and 2 are on, these lights are controlled by the output of the OR gate labeled U.

Storage element 2 should be set to 1 for the next clock cycle if the next state is 01 or 11.This is true when the current state is 00 or 10. So it is controlled by the output of the OR gate labeled U.

3.43
a)

S1	S0	X		D1	D0	Z
0	0	0		0	0	0
0	0	1		0	0	0
0	1	0		0	0	1
0	1	1		1	0	1
1	0	0		1	1	1
1	0	1		1	1	1
1	1	0		1	0	1
1	1	1		1	0	1

b)



3.44