1(a) It states that every algebraic expression deducible from the postulates of Boolean algebra remains valid if the operators & identity elements are interchanged. For example:

if $A+\bar{A} = 1$ is true then $A.\bar{A} = 0$ is also true

1.(b) $(\overline{ABC})(A+C)(A+\bar{C})$

$= \overline{(ABC)}\ (A + C\bar{C})$ [or distributed law]

$= \overline{(ABC)} . (A+0)$ [$C\bar{C} = 0$]

$= \overline{(ABC)} . A$ [$A+0 = A$]

$= (\bar{A} + \bar{B} + \bar{C}) . A$ [De Morgan law]

$= A.\bar{A} + A\bar{B} + A\bar{C}$ [distributed law]

$= A\bar{B} + A\bar{C}$ [$A.\bar{A} = 0$]

* Determine the single error correcting code (Hamming code) for '101110110'

Soln: The no of data bits in the message, n = 9. we can obtain the required no of parity bits (P) from the following formula: $2^P >= n + P + 1$

if P = 3 . $2^3 >= 9 + 3 + 1$ (false); if P = 4 , $2^4 >= 9+4+1$ (true)

so , P = 4 , we have to sent total n+P = 9+4 = 13 bit message.

Sending end:

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13
Values	P1	P2	1	P3	0	1	1	P4	1	0	1	1	0

To find P1, P2, P3, P4 we have to X-OR the binary values of bit positions holding '1's

3	0 0 1 1
6	0 1 1 0
7	0 1 1 1
9	1 0 0 1
11	1 0 1 1
12	1 1 0 0
X-OR	1 1 0 0
	P4 P3 P2 P1

Sending message code will be : 0011011110110

Receiving End: If there is no error, the receiver will get: 0011011110110

verification:

Bit position	1	2	3	4	5	6	7	8	9	10	11	12	13
Values	0	0	1	1	0	1	1	1	1	0	1	1	0

Binary values of the bit position holding 1's:

3	0 0 1 1
4	0 1 0 0
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
11	1 0 1 1
12	1 1 0 0
X-OR	0 0 0 0

So, there is no error.

2. (a)

Given, $F(A,B,C,D) = \Sigma(1,3,5,7,9,15)$

$= \Sigma(0001, 0011, 0101, 0111, 1001, 1111)$

And, $d(A,B,C,D) = \Sigma(4,6,12,13)$

$= \Sigma(0100, 0110, 1100, 1101)$

AB \ CD	00	01	11	10
00		1	1	
01	X	1	1	X
11	X	X	1	
10		1		

$F = C'D + A'D + BD$

(2)

(b) Simplify the following Boolean function F by using the Quine-McCluskey method:

$$F(w,x,y,z) = \Sigma(0,1,2,8,10,11,14,15)$$

$\Rightarrow F(w,x,y,z) = \Sigma(0000, 0001, 0010, 1000, 1010, 1011, 1110, 1111)$

Determination of prime implicants:

Column I	Column II	Column III
Group 0: 0 0000 ✓	0,1 000_ 0,2 00_0 ✓ 0,8 _000 ✓	0,2,8,10 _0_0 ~~0,8,2,10 _0_0~~
Group 1: 1 0001 ✓ 2 0010 ✓ 8 1000 ✓	2,10 _010 ✓ 8,10 10_0 ✓	
Group 2: 10 1010 ✓	10,11 101_ ✓ 10,14 1_10 ✓	10,11,14,15 1_1_ ~~10,14,11,15 1_1_~~
Group 3: 11 1011 ✓ 14 1110 ✓	11,15 1_11 ✓ 14,15 111_ ✓	
Group 4: 15 1111 ✓		

selection of Prime implicants -

PIs	w x y z		0	1	2	8	10	11	14	15
0,2,8,10	- 0 - 0	$x'z'$	X		X	X	X			
10,11,14,15	1 - 1 -	wy					X	X	X	X
0,1	0 0 0 -	$w'x'y'$	X	X						

$$F = x'z' + wy + w'x'y'$$

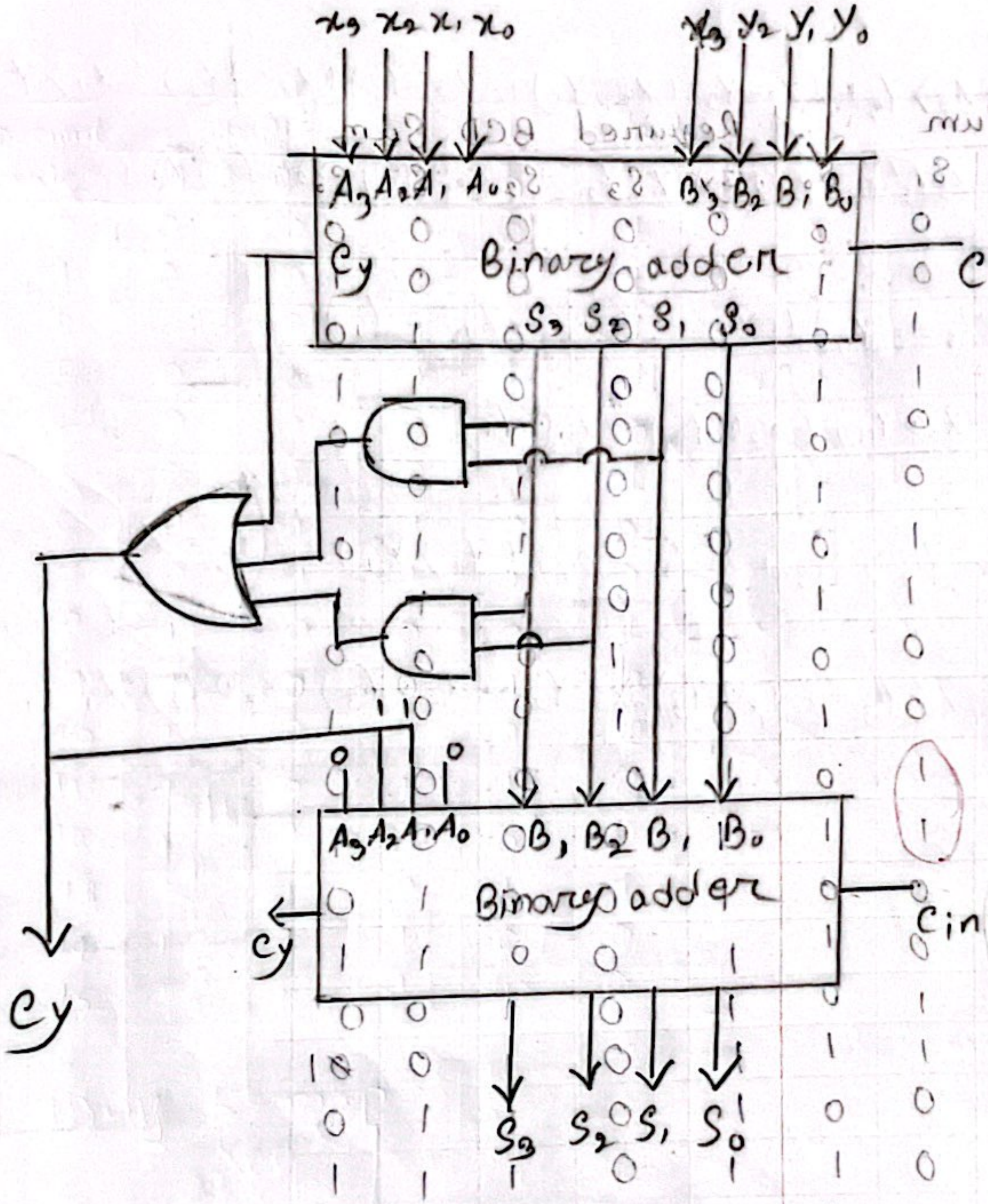
checking:

		$w'x'$	$w'x$	wx	wx'
	yz \ wx	00	01	11	10
$y'z'$	00	1			1
$y'z$	01	1			
yz	11			1	1
yz'	10	1		1	1

$$F = wy + z'x' + w'x'y'$$

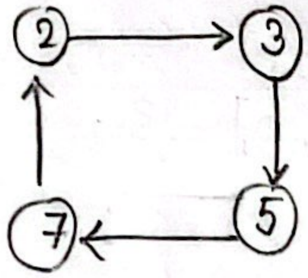
3. (a)

Decimal value	Binary Sum Cy	S_3	S_2	S_1	S_0	Required BCD Sum Cy	S_3	S_2	S_1	S_0
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	1	1
4	0	0	1	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	1	0	1
6	0	0	1	1	0	0	0	1	1	0
7	0	0	1	1	1	0	0	1	1	1
8	0	1	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	1	0	0	1
10	0	1	0	1	0	1	0	0	0	0
11	0	1	0	1	1	1	0	0	0	1
12	0	1	1	0	0	1	0	0	1	0
13	0	1	1	0	1	1	0	0	1	1
14	0	1	1	1	0	1	0	1	0	0
15	0	1	1	1	1	1	0	1	0	1
16	1	0	0	0	0	1	0	1	1	0
17	1	0	0	0	1	1	0	1	1	1
18	1	0	0	1	0	1	1	0	0	0
19	1	0	0	1	1	1	1	0	0	1



x3 x2 x1 x0
y3 y2 y1 y0
A3 A2 A1 A0
B3 B2 B1 B0
Cy
Binary adder
Cin = 0
S3 S2 S1 S0
A3 A2 A1 A0
B3 B2 B1 B0
Binary adder
Cy
Cin = 0
S3 S2 S1 S0
Cy

3 (b)



State diagram

State table

Present Stat			Next State		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1
0	1	0	0	1	d
0	1	1	1	0	d
1	0	1	1	1	d
1	1	1	0	1	0

we need 3 D Flip flop

Excitation Map

present state			Next state			FF's input		
Q_3	Q_2	Q_1	Q_3	Q_2	Q_1	D_3	D_2	D_1
0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	1	0	1
1	0	1	1	1	1	1	1	1
1	1	1	0	1	0	0	1	0

FFs input equation

For D_3

$Q_3 \backslash Q_2Q_1$	00	01	11	10
0			1	0
1		1	0	

For D_2

$Q_3 \backslash Q_2Q_1$	00	01	11	10
0			0	1
1		1	1	

For D_1

$Q_3 \backslash Q_2Q_1$	00	01	11	10
0			1	1
1		1	0	

$D_3 = Q_3 Q_2' Q_1 + Q_3' Q_2 Q_1$

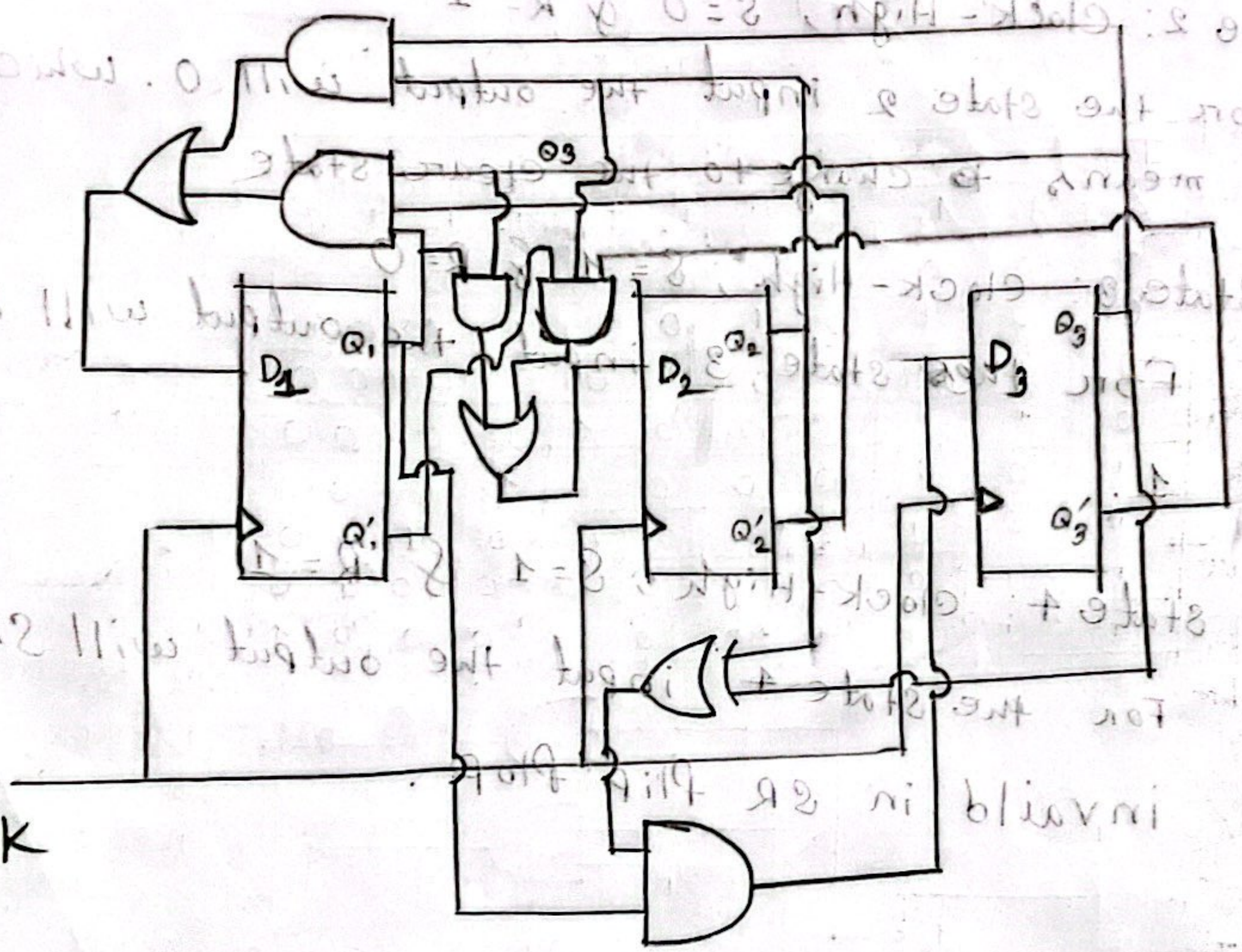
$= Q_1 (Q_3 Q_2' + Q_3' Q_2)$

$= Q_1 (Q_3 \oplus Q_2)$

$D_2 = Q_3 Q_1 + Q_3' Q_2 Q_1'$

$D_1 = Q_3 Q_2' Q_1 + Q_3 Q_2$

3(b)



Decimal value	input A	B	C	D	Output F
0	0	0	0	0	1
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

using K-map:

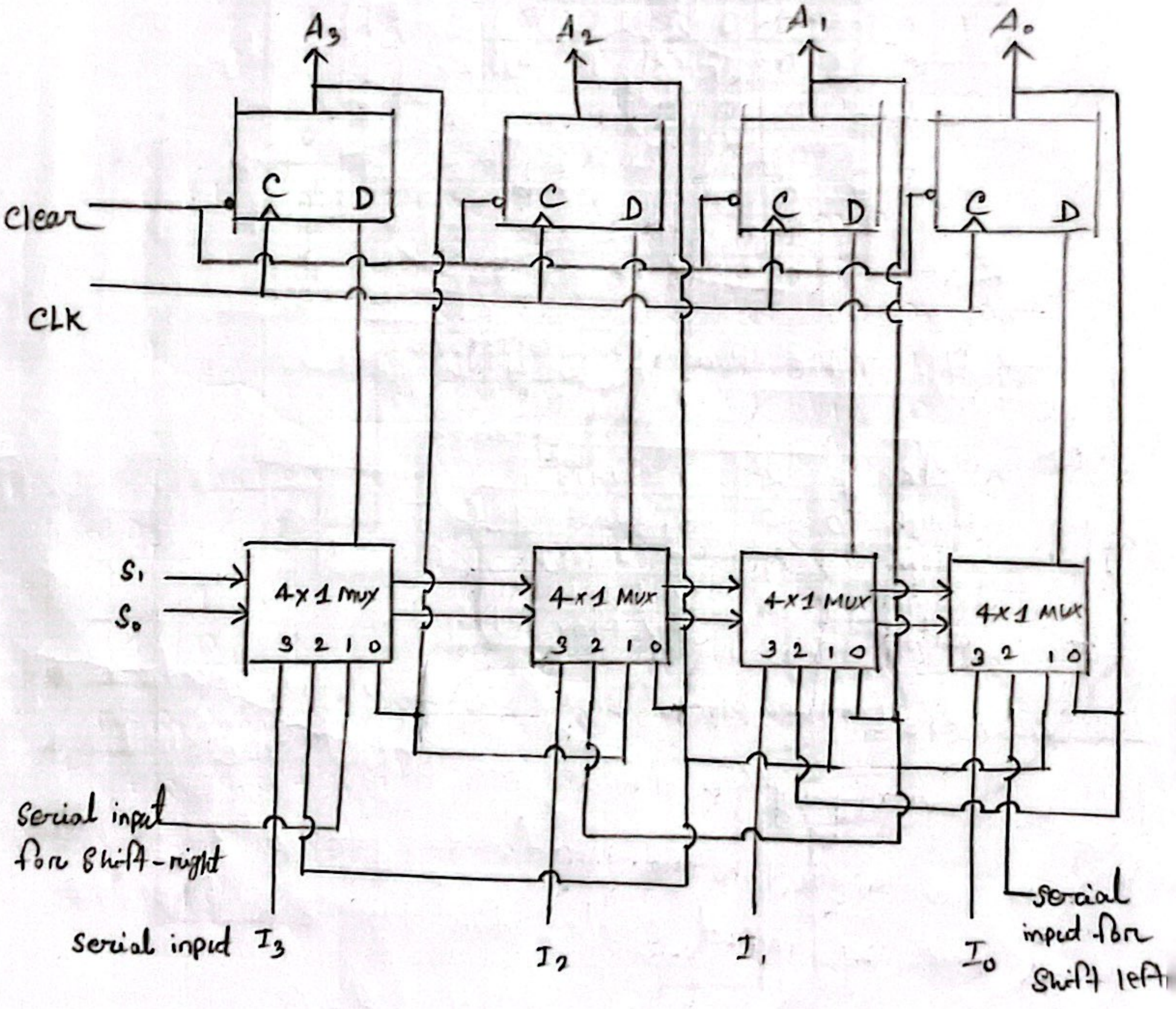
AB \ CD	00	01	11	10
00	1	1	1	1
01		1		
11		1		
10	1			

$F = B'C'D' + A'B' + BC'D$

Logic diagram draw করতে হবে।

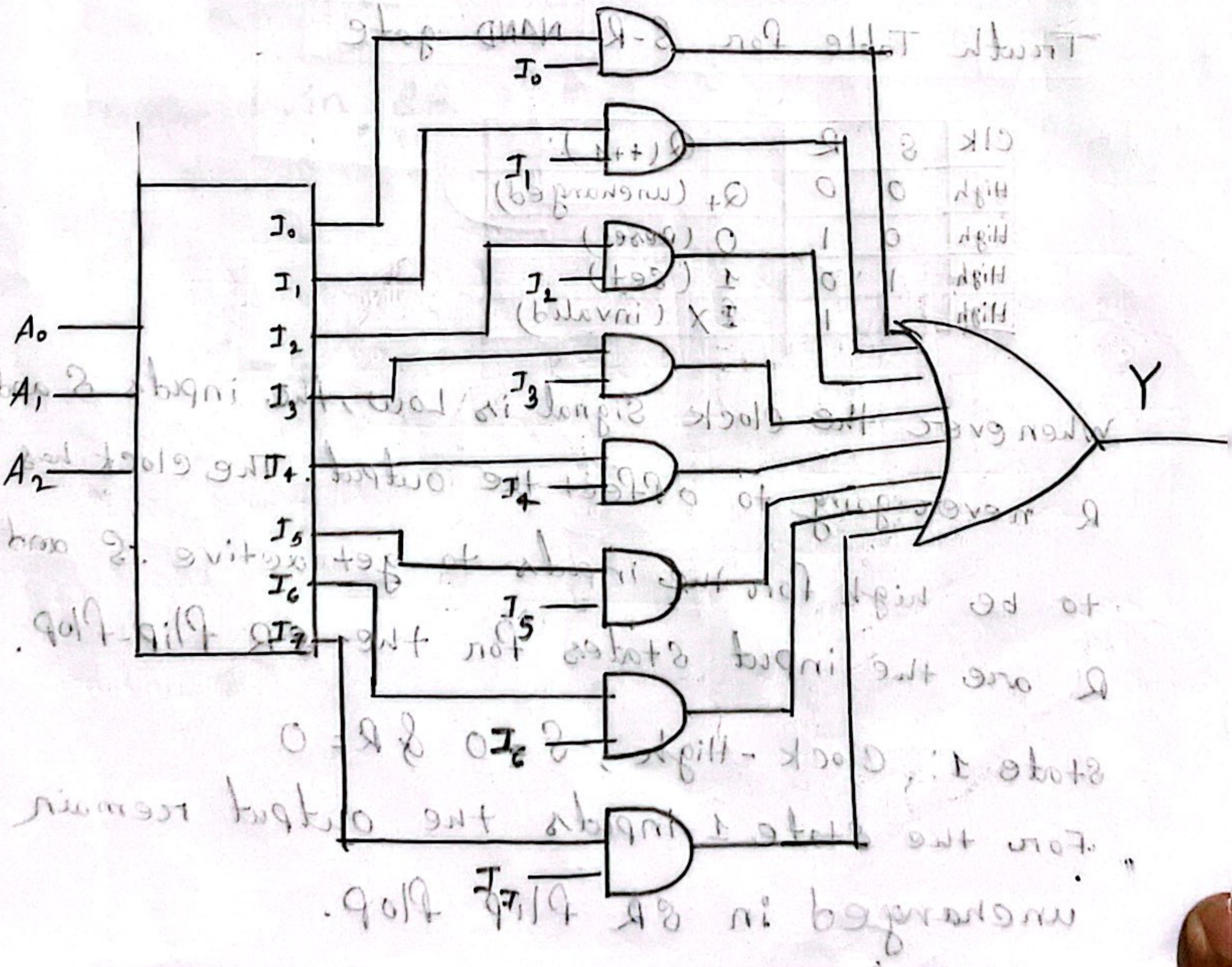
4.(b)

S_1	S_0	Register operation
0	0	No change
0	1	Shift right
1	0	Shif left
1	1	Parallel load



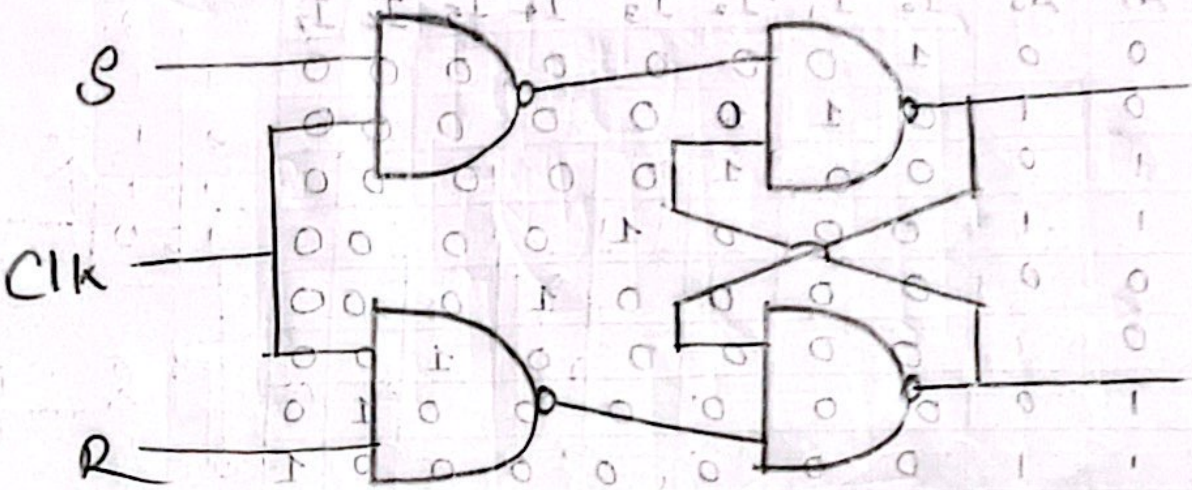
5. (a)

Input										
A_2	A_1	A_0	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



5.(b)

SR flip flop using Nand gate



Truth Table for S-R NAND gate

Clk	S	R	Q(t+1)
High	0	0	Q_t (unchanged)
High	0	1	0 (Reset)
High	1	0	1 (set)
High	1	1	X (invalid)

whenever the clock signal is Low, the inputs S and R never going to affect the output. The clock has to be high for the inputs to get active. S and R are the input states for the SR flip-flop.

State 1: Clock - High, S = 0 & R = 0

For the state 1 inputs the output remain unchanged in SR flip flop.

State 2: Clock - High, S = 0 & R = 1

For the state 2 input the output will 0. which means ~~to~~ change to the clear state

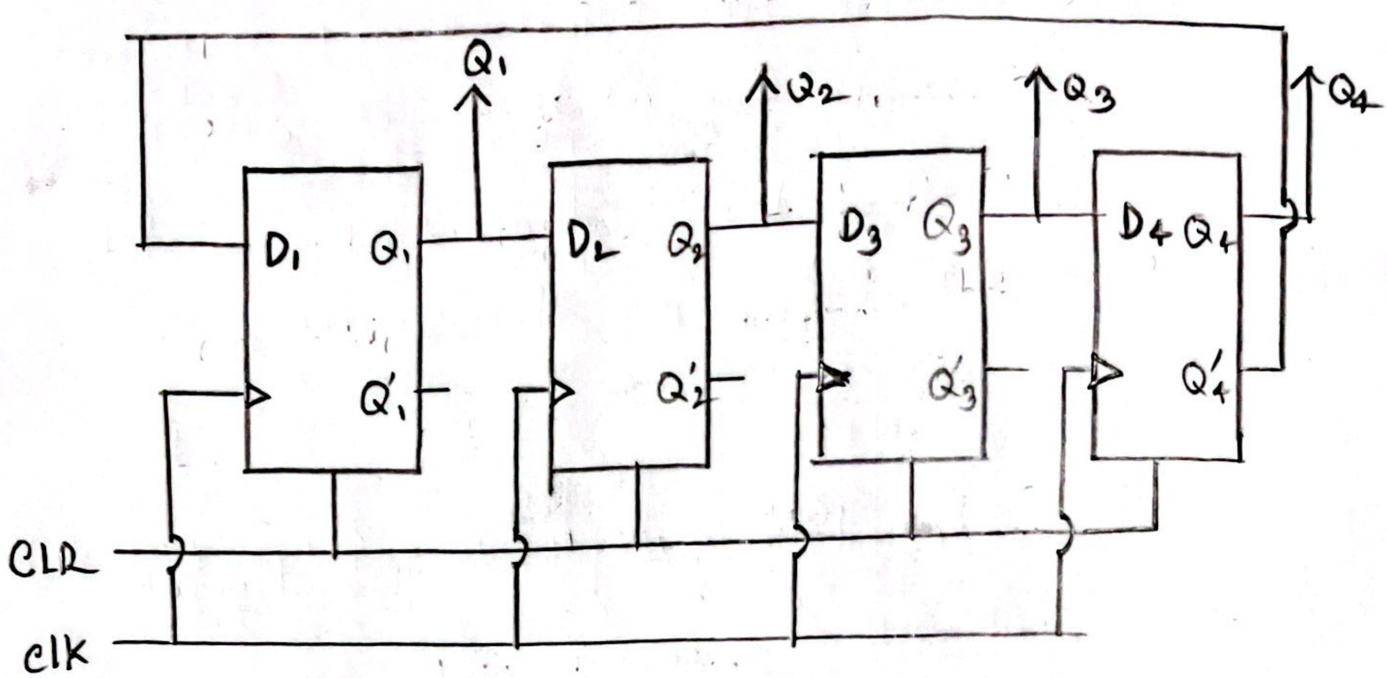
State 3: Clock - High, S = 1 & R = 0

For ~~thes~~ state 3 input the output will be 1.

State 4: Clock - High, S = 1 & R = 1

For the state 4 input the output will shown invaild. in SR flip flop.

6. (a)



Clock pulse No	Q_1	Q_2	Q_3	Q_4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	1

6. (b)

~~$S_1 = A_1 \oplus B_1$~~

$S_1 = P_1 \oplus C_1$

$= P_1 \oplus 0$ [intial carry $C_1 = 0$]

$= A_1 \oplus B_1$

$C_1 = 0$

$S_2 = P_2 \oplus C_2$

$= (A_2 \oplus B_2) \oplus A_1 B_1$

$C_2 = G_1 \overset{+}{\cancel{\otimes}} P_1 C_1$

$= A_1 B_1$ [intial carry $C_1 = 0$]

$C_3 = G_2 + P_2 C_2$

$= A_2 B_2 + [(A_2 \oplus B_2) A_1 B_1]$

$S_3 = P_3 \oplus C_3$

$= (A_3 \oplus B_3) \oplus [A_2 B_2 + (A_2 \oplus B_2) A_1 B_1]$

$C_4 = G_3 + P_3 C_3$

$= A_3 B_3 + (A_3 \oplus B_3) [A_2 B_2 + (A_2 \oplus B_2) A_1 B_1]$

$S_4 = P_4 \oplus C_4$

$= (A_4 \oplus B_4) + [A_3 B_3 + (A_3 \oplus B_3) [A_2 B_2 + (A_2 \oplus B_2) A_1 B_1]$

$C_5 = G_4 + P_4 C_4$

$= A_4 B_4 + (A_4 \oplus B_4) [A_3 B_3 + (A_3 \oplus B_3) [A_2 B_2 + (A_2 \oplus B_2) A_1 B_1]$

7. (b)

F_1 : $X > Y = X_3 > Y_3 + (X_3 == Y_3)(X_2 > Y_2) + (X_3 == Y_3)(X_2 == Y_2)(X_1 > Y_1)$

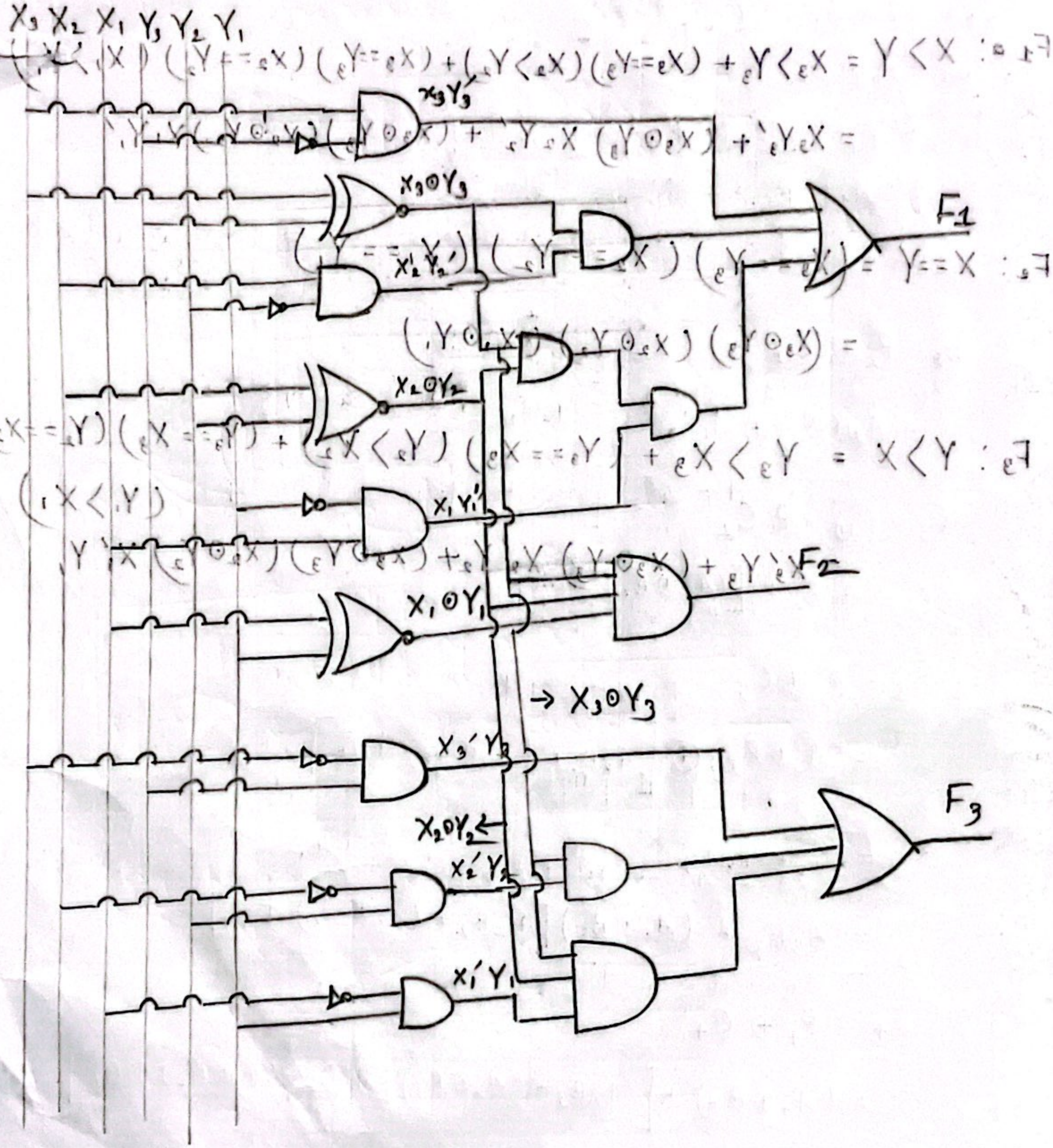
$= X_3 Y_3' + (X_3 \odot Y_3) X_2 Y_2' + (X_3 \odot Y_3)(X_2 \odot Y_2) X_1 Y_1'$

F_2 : $X == Y = (X_3 == Y_3)(X_2 == Y_2)(X_1 == Y_1)$

$= (X_3 \odot Y_3)(X_2 \odot Y_2)(X_1 \odot Y_1)$

F_3 : $Y > X = Y_3 > X_3 + (Y_3 == X_3)(Y_2 > X_2) + (Y_3 == X_3)(Y_2 == X_2)$
$(Y_1 > X_1)$

$= X_3' Y_3 + (X_3 \odot Y_3) X_2' Y_2 + (X_3 \odot Y_3)(X_2 \odot Y_2) X_1' Y_1$



X3 X2 X1 Y3 Y2 Y1
X3Y3'
X3⊙Y3
X2'Y2'
F1
X2⊙Y2
X1Y1'
F2
X1⊙Y1
→ X3⊙Y3
X3'Y3
F3
X2⊙Y2 ←
X2'Y2
X1'Y1