

## Assignment - 1

Q.1 Make a dual & Complement of

$$1. F = (a'b'c + abc + b')$$

$$F_D = (a' + b + c) \cdot (a + b + c) \cdot (b')$$

$$F_C = (a + b' + c') \cdot (a' + b' + c') \cdot (b)$$

$$2. F = x'y'z' + x'y'z$$

$$F_D = (x' + y + z') \cdot (x' + y' + z)$$

$$F_C = (x + y' + z) \cdot (x + y + z')$$

$$3. F = x(y'z' + yz) = xy'z' + xyz$$

$$F_D = (x + y' + z') \cdot (x + y + z)$$

$$F_C = (x' + y + z) \cdot (x' + y' + z')$$

$$4. F = xy + x'z + yz$$

$$F_D = (x + y) \cdot (x' + z) \cdot (y + z)$$

$$F_C = (x' + y') \cdot (x + z') \cdot (y' + z')$$

Q.2 Reduce the foll. Boolean expression using Postulates and theorems of Boolean Algebra.

$$1. bc + abc + a'b'c + abc$$

$$= bc + abc + a'b'c$$

$$= bc + bc(a + a')$$

$$= bc$$

$$\begin{aligned} 2. & (A'B'C')' + (ABC')' \\ = & (A'B'C')' + (AB'C)' \\ = & (A+B'+C) + (A'+B+C') \\ = & (A+A') + (B+B') + (C+C') \\ = & 1 \end{aligned}$$

$$\begin{aligned} 3. & B'D + A'BC' + ACD + A'B'C \\ = & B'D + A'BC(C'+C) + ACD \\ = & B'D + A'B + ACD \end{aligned}$$

$$\begin{aligned} 4. & x + x'y \\ = & (x+x') \cdot (x+y) \\ = & (x+y) \end{aligned}$$

$$\begin{aligned} 5. & x(x'+y) \\ = & xx' + xy \\ = & xy \end{aligned}$$

$$\begin{aligned} 6. & x'y'z + x'y'z + xy' \\ = & x'z(y'+y) + xy' \\ = & x'z + xy' \end{aligned}$$

$$7. xy + x'z + yz$$

$$\begin{aligned}
 & 8. (x+y)(x'+z)(y+z) \\
 &= (xx' + xz + yx' + yz)(y+z) \\
 &= (xz + x'y + yz)(y+z) \\
 &= xyz + x'y + yz + xz + x'y'z + yz \\
 &= yz + x'y + xz
 \end{aligned}$$

Q. 3 Find out the complement of fall. function by De Morgan's Law

$$\begin{aligned}
 F(a, b, c, d) &= [(abc + bcd)' + \\
 &\quad + (abc'd' + (ab)'d)]' \\
 F'(a, b, c, d) &= [F(abc + bcd) \cdot \\
 &\quad (abc'd' + (a+b))']
 \end{aligned}$$

Q. 4 Express the Boolean function in a

a) sum of min terms

$$\begin{aligned}
 1. F &= A + B'C + A'B \\
 &= A(B+B')C(c+c') + B'C(A+A') + \\
 &\quad A'B(Cc+c') \\
 &= ABC + ABC' + AB'C + AB'C' + AB'C + \\
 &\quad A'B'C + A'B'C + A'B'C' \\
 &= m_7 + m_6 + m_5 + m_4 + m_1 + m_3 + m_2 \\
 &= \Sigma m(1, 2, 3, 4, 5, 6, 7)
 \end{aligned}$$

$$2. F = a' + a'b + ab'c'$$

$$= a'(b+b')(cc+c') + a'b(cc+c') + ab'c'$$

$$= a'bc + a'bc' + a'b'c + a'b'c' + a'bc + a'b'c' + ab'c'$$

$$= m_3 + m_2 + m_1 + m_0 + m_4$$

$$= \sum m(0, 1, 2, 3, 4)$$

Q. 5 Co

1. F

= C

$$3. F = xy + x'z$$

$$= xy(z+z') + x'z(y+y')$$

$$= xyz + xyz' + x'yz + x'y'z$$

$$= m_7 + m_6 + m_5 + m_1$$

$$= \sum m(1, 3, 5, 7)$$

2. F

= C

$$4. F = (a+b'+c)(a'+d)(a+b'+c+d)$$

$$= ad + a'b' + b'd + a'c + cd$$

$$(a+b'+c+d)$$

$$= ad + a'b' + b'd + a'c + cd$$

$$(a+b'+c+d)$$

$$= ad + ab'd + accd + ad + a'b'b' +$$

$$a'b' + a'b'c + a'b'd + ab'd + b'd +$$

$$b'cd + b'd + a'ac + a'b'c + a'c' +$$

$$+ a'cd + accd + b'cd + ccd + cd$$

$$= ad(b+b')(c+c') + ab'd(c+c')$$

$$+ accd(b+b') + a'b'(cc+c')(cd+d')$$

Q. 6

1.

Q. 5 Convert the foll. to other canonical form.

$$1. F(x, y, z) = \Sigma(1, 3, 7)$$

$$= \Pi(0, 2, 4, 5, 6)$$

$$= (x+y+z) \cdot (x+y'+z) \cdot (x'+y+z) \cdot \\ (x'+y'+z) \cdot (x'+y'+z)$$

$$2. F(x, y, z) = \Pi(1, 3)$$

$$= \Sigma(0, 2, 4, 5, 6, 7)$$

$$= x'y'z' + x'y'z + x'yz' + xy'z + xyz' + \\ xyz$$

$$3. F(a, b, c, d) = \Sigma(1, 3, 5, 2, 11, 15)$$

$$= \Pi(0, 4, 6, 7, 8, 9, 10, 12, \\ 13, 14)$$

$$(2) = (a+b+c+d) \cdot (a+b'+c+d) \cdot (a+b'+c'+d) \\ (a+b'+c'd') \cdot (a'+b+c+d) \cdot (a'+b+c+d') \\ (a'+b+c'+d) \cdot (a'+b'+c+d) \cdot \\ (a'+b'+c+d') \cdot (a'+b'+c'+d)$$

Q. 6 Reduce foll. function by K-map

$$1. F(a, b, c) = a'b + ab'c$$

a \ bc	00	01	11	10
0	0	1	1	1
1	1	0	0	0

2.  $F(a, b, c) = \sum m(1, 2, 4, 7, 8, 11, 13, 14)$

$ab \backslash cd$	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	1	2	11	10

$$\begin{aligned}
 &= a'b'c'd + a'b'cd' + a'bc'd' + ab'cd \\
 &\quad + abc'd + abc'd' + ab'c'd' +
 \end{aligned}$$

$$ab'cd$$

$$\begin{aligned}
 &= a'b'(c(c \oplus d)) + a'b(c(c \oplus d))' + \\
 &\quad ab(c \oplus d) + ab'(c(c \oplus d))' \\
 &= (a \oplus b)'(c(c \oplus d)) + (a \oplus b)(c(c \oplus d))' \\
 &= a \oplus b \oplus c \oplus d
 \end{aligned}$$

3.  $F(a, b, c) = \prod M(0, 3, 5, 6, 9, 10, 12, 15)$

$$= \sum m(1, 2, 4, 7, 8, 11, 13, 14)$$

$ab \backslash cd$	00	01	11	10
00	0	1	2	3
01	4	5	7	6
11	12	13	15	14
10	1	2	11	10

4.  $F = (x+y) + (x'+y)$

$$= M_0 + M_2$$

$$= \prod M(0, 2)$$

$$= \sum m(1, 3)$$

$$= x + y + xc' + y \\ = y + 1$$

$$= 1$$

$$5. F = A'B'C' + A'CD' + A'BCD' + AB'C'$$

~~AB~~ \ CD 00 01 11 10

00	10 (D <sub>1</sub> )	3 (D <sub>2</sub> )	= A'B'C' + A'CD' + AB'C'
01	4	5	= B'C'(A' + A) + A'CD'
11	12	13	= B'C' + A'CD'
10	0 (D <sub>1</sub> )	11	

$$6. F(A, B, C, D) = \sum m(1, 3, 4, 5, 6, 7, 9, 12, 13)$$

~~AB~~ \ CD 00 01 11 10

00	0 (11) 1 (D <sub>1</sub> )	2	= a'd + a'b + c'b + c'd
01	1 (4) 1 (D <sub>2</sub> ) 1 (2) 1 (D <sub>3</sub> )	= a'(b+d) + c'(b+d)	
11	1 (12) 1 (D <sub>1</sub> ) 1 (5) 1 (D <sub>2</sub> )	= (a' + c')(b+d)	
10	1 (8) 1 (4) 1 (7) 1 (D <sub>3</sub> )		

$$7. F = A'B'C' + A'BC' + A'BC + AB'C$$

~~A~~ \ BC 00 01 11 10

0	1 (D <sub>1</sub> )	1 (D <sub>3</sub> ) (D <sub>2</sub> )	= a'b + a'c' + ab'c
1	4 (D <sub>1</sub> )	2 (D <sub>2</sub> )	

Q.7 Find out maxterm for the foll.

$$F_1 = (x+y)(y'+z)$$

$$= x + y + y' + z$$

$$= x + z + 1$$

$$= 1$$

Q.8 Simplify the Boolean function

$$\begin{aligned}
 1. F &= x'y'z' + x'y'z + xy'z' + xy'z \\
 &= x'y'(z' + z) + xy'(z' + z) \\
 &= x'y' + xy' \\
 &= (x' + x)y' \\
 &= y'
 \end{aligned}$$

2.  $F(w, x, y, z) = \Sigma(1, 3, 7, 11, 15)$  and  
 $d(w, x, y, z) = \Sigma(0, 2, 5)$

$$\begin{array}{c}
 = \cancel{wx}yz \\
 \begin{array}{ccccc}
 & 00 & 01 & 11 & 10 \\
 \begin{array}{|c|c|c|c|c|} \hline
 00 & 1 & 1 & 1 & 1 \\ \hline
 01 & 1 & X & 1 & 1 \\ \hline
 11 & 1 & 1 & 1 & 1 \\ \hline
 10 & 1 & 1 & X & 1 \\ \hline
 \end{array} & = w'x' + yz
 \end{array}$$

3.  $F = A'B'D' + A'CD + A'BC$  and  
 $d = A'BC'D + ACD + AB'D'$

$$\begin{array}{c}
 = \cancel{AB}CD \\
 \begin{array}{ccccc}
 & 00 & 01 & 11 & 10 \\
 \begin{array}{|c|c|c|c|c|} \hline
 00 & 1 & 0 & 1 & 1 \\ \hline
 01 & 1 & 1 & X & 1 \\ \hline
 11 & 1 & 1 & 1 & 1 \\ \hline
 10 & 1 & 1 & X & 1 \\ \hline
 \end{array} & = a'c + b'c'd'
 \end{array}$$

Q.9 Impl  
only

(i)  $F = A$

=  $A \bar{B}$

NAND

NOR

(ii)  $F$

NAND

A

C

D

B

E

(A B)'

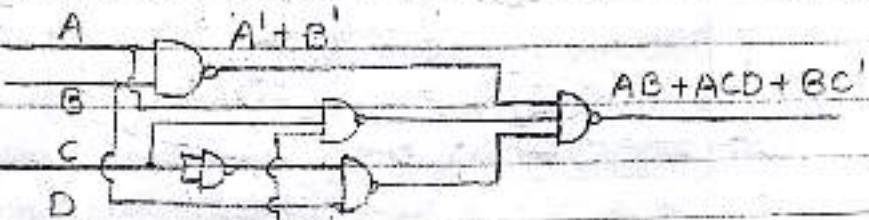
RIGHT/WAY  
Date / /  
Page No. /

Q.9 Implement the foll. Boolean fun" with  
only NOR gates and only NAND gates

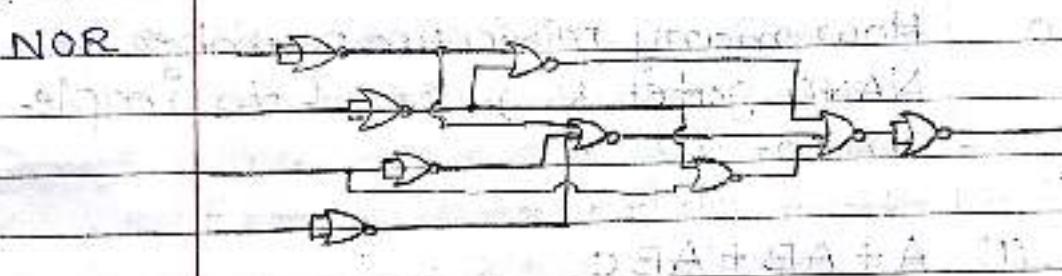
(i)  $F = A(B+CD) + BC'$

=  $AB + ACD + BC'$

NAND



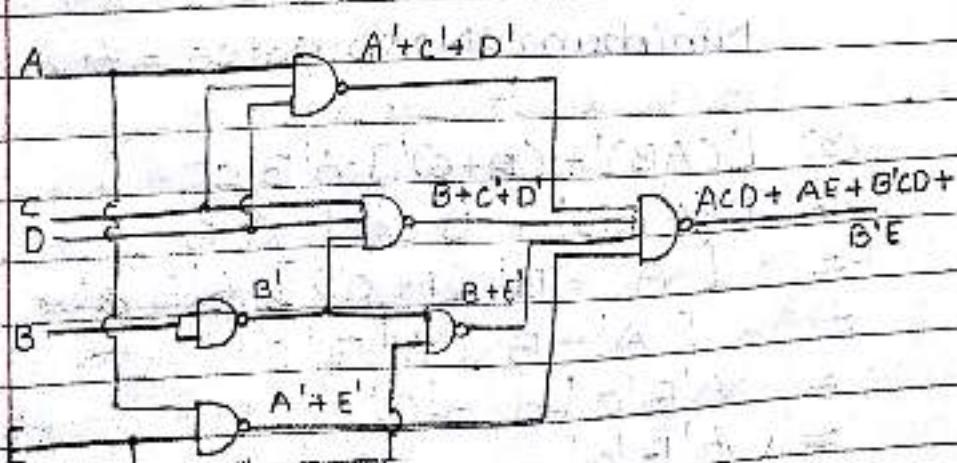
NOR



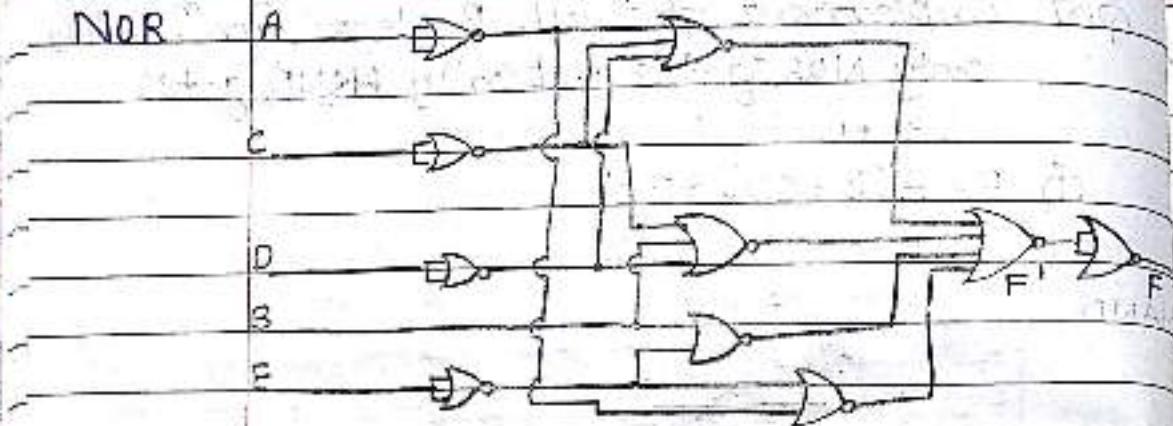
(ii)  $F = (A+B')(C'D+E)$

=  $ACD + AE + B'CD + B'E$

NAND



NOR



Q. 11

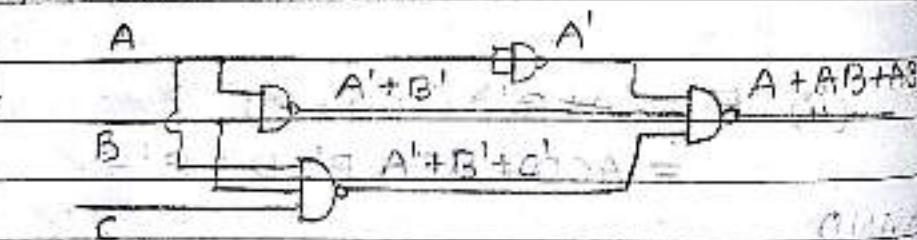
M  
F  
B  
m

Q. 10 How many minimum number of NAND gates required to implement

$$\begin{aligned} 1. & B' \\ = & \bar{B} \\ = & \bar{B}' \\ = & F \end{aligned}$$

(1)  $A + AB + ABC$

$$\begin{aligned} & = (A \\ & = C \\ & = C \\ & = C \end{aligned}$$



Minimum NAND gates = 4

(2)  $[(AB)' + (B+C')'] A'B'C'$

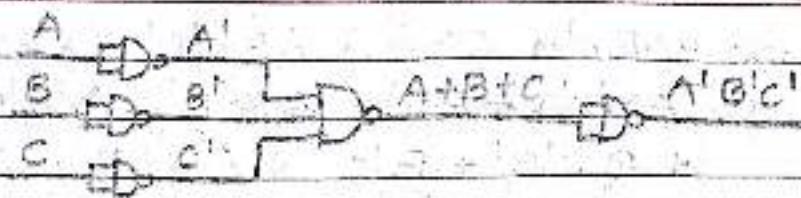
$$= [A' + B' + B'C'] A'B'C'$$

$$= [A' + B'(C+C')] A'B'C'$$

$$= [A' + B'] A'B'C'$$

$$= A'B'C' + A'B'C'$$

$$= A'B'C'$$



Minimum NAND gates = 5

Q. 11 Find Complement of the foll. Boolean fun<sup>n</sup> and reduce to minimum no. of literals.

$$\begin{aligned}
 & 1. B'D + A'B'C' + ACD + A'BC \\
 & = //B'D + A'B'C'C' + () + ACD \\
 & = //B'D + A'B + ACD \\
 & = F' = (B+D') \cdot (A+B'+C) \cdot (A'+C'+D') \cdot \\
 & \quad (A+B'+C') \\
 & = (AB + BB' + BC + AD' + B'D' + CD') \\
 & \quad (A'+C'+D') \cdot (A+B'+C') \\
 & = (AB + BC + AD' + B'D' + CD') (A'+C'+D') \cdot \\
 & \quad (A+B'+C') \\
 & = ABC' + ABD' + A'B'C + BCD' + AC'D' + \\
 & \quad AD' + A'B'D' + B'C'D' + B'D' + A'CD' + CD' \\
 & \quad (A+B'+C') \\
 & = ABC' + ABC' + ABD' + ABC'D' + ABCD' \\
 & \quad + AC'D' + AB'C'D' + AC'D' + AD' + ABD' \\
 & \quad + AC'D' + A'B'D' + A'B'C'D' + AB'C'D' + \\
 & \quad B'C'D' + AB'D' + B'D' + B'C'D' + A'B'CD \\
 & \quad + ACD' + BCD' +
 \end{aligned}$$

$$\begin{aligned}
 &= ABC' + ABD' + ABC'D' + ABCD' + ACD' \\
 &\quad AB'C'D' + AD' + AB'D' + A'B'D' + A'B'C'D' \\
 &\quad + B'C'D' + B'D' + A'B'C'D' + ACD' + \\
 &\quad B'C'D'
 \end{aligned}$$

Q. 12 obt  
in  
K-r  
1.  $x'z$

$$\begin{aligned}
 &= ABC' + ABD' + AD' + B'D' + B'C'D' + \\
 &\quad A'B'C'D'
 \end{aligned}$$

$$= ABC' + ABD' + AD' + B'D' + B'C'D'$$

$$= ABC' + ABD' + AD' + B'D'$$

$$= ABC' + AD' + B'D'$$

$$2. x'z + w'xy' + w(x'y + xy')$$

$$= x'z + w'xy' + wx'y + wxy'$$

$$\begin{aligned}
 F' = & (x+z')(w+x'+y)(w'+x+y') \\
 & (w'+x'+y)
 \end{aligned}$$

$$= (wx + xy + wz' + x'z' + yz')$$

$$(w'w' + w'x' + w'y + x'w' + xy + \\ y'w' + x'y' + )$$

$$\begin{aligned}
 = & wxy + w'xy + xyw' + xy + wxyz' \\
 & + wx'y'z' + w'x'z' + w'x'y'z' +
 \end{aligned}$$

$$x'y'w'z' + x'y'z' + w'yz' + w'x'z' \\ z' + w'yz' + w'xyz' + xyz' +$$

$$\begin{aligned}
 = & wxy + w'xy + xy + wxyz' + wx'y'z' \\
 & + w'x'z' + w'x'y'z' + x'y'w'z' + x'y'z' \\
 & + w'yz' + w'xyz' + (xyz')
 \end{aligned}$$

$$\begin{aligned}
 = & xy + x'y'z' + x'w'z' + xyz' + w'yz' \\
 = & xy + x'y'z' + x'w'z' + w'yz'
 \end{aligned}$$

obt  
in  
K-r  
1.  $x'z$

wx  
00

01

10

10

2.  $B'D$

AB

00

01

10

3.  $FCA$

00

01

10

10

10

10

10

10

10

10

10

10

Q.12 obtain the simplified expression in sum of products using K-map:

$$1. \quad x'z + w'xy' + w(x'y + xy')$$

		xyz	00	01	11	10				
		ab	00	01	11	10				
		cd	00	01	11	10				
			00	01	11	10				
			01	11	10	00				
			11	12	13	14				
			10	01	11	00				

$$F = xy' + x'z + wxy'$$

$$2. \quad B'D + A'B'C' + ACD + A'B'C$$

		CD	00	01	11	10				
		AB	00	01	11	10				
		00	01	11	10					
			00	01	11	10				
			01	11	10	00				
			11	12	13	14				
			10	01	11	00				

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

$$3. \quad F(a,b,c,d,e) = \Sigma(0,2,4,6,9,11,13,15, 17, 21, 25, 27, 29, 31)$$

		cde	000	001	011	010	110	111	101	100
		ab	00	01	11	10	11	10	01	11
		00	01	11	10	11	10	01	11	
			00	01	11	10	11	10	01	11
			01	11	10	00	11	10	01	11
			11	21	12	17	26	30	13	28
			10	15	17	19	18	22	23	16

$$F = be + ad'e + a'b'c'$$

Q.13 Obtain truth table for foll. fun?

Q.15

$$1. F = x(y'z' + yz)$$

x	y	z	$y'z'$	$yz$	$y'z' + yz$	$x(y'z' + yz)$
0	0	0	1	0	1	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	1	0
1	0	0	1	0	1	1
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	1	1	1

Q.14 3-Variable truth table has a high T.T. Output for the inputs: 010, 011 and 110. Write down Boolean expression for sum of Product

$$F = 010 + 011 + 110$$

$$= x'y'z' + x'y'z + x'yz'$$

$$= x'y + x'yz'$$

$$= y(x' + xz')$$

$$= y(x' + z')$$

$$= yx' + yz'$$

T.T.

Q.15 State and Prove :

• Duality principle

OR  $\rightarrow$  AND

$$\text{Dual} \Rightarrow (\bar{x}+y)(x+y)$$

AND  $\rightarrow$  OR

$$= \bar{x}y$$

0  $\rightarrow$  1

$$1 \cdot H \cdot S = (\bar{x}+y)(x+y)$$

1  $\rightarrow$  0

$$\begin{aligned} & -\bar{x}x + xy + \bar{x}y' + yy' \\ & = \bar{x} + \bar{x}y + \bar{x}y' \\ & = \bar{x}(1+y+y') \\ & = \bar{x} \end{aligned}$$

• De-Morgan Theorem

$$(i) \quad \overline{x+y} = \bar{x} \cdot \bar{y}$$

$$(ii) \quad \overline{x \cdot y} = \bar{x} + \bar{y}$$

	x	y	$x+y$	$\bar{x}+\bar{y}$	$\bar{x}$	$\bar{y}$	$\bar{x} \cdot \bar{y}$
	0	0	0	1	1	1	1
	0	1	1	0	1	0	0
	1	0	1	0	0	1	0
	1	1	1	0	0	0	0

$\therefore \overline{x+y} = \bar{x} \cdot \bar{y}$  proved

	x	y	$x \cdot y$	$\bar{x} \cdot \bar{y}$	$\bar{x}$	$\bar{y}$	$\bar{x}+\bar{y}$
	0	0	0	1	1	1	1
	0	1	0	1	1	0	1
	1	0	0	1	0	1	1
	1	1	1	0	0	0	0

$\therefore \overline{x \cdot y} = \bar{x} + \bar{y}$  proved

• Associative Law

$$(i) x + (y + z) = (x + y) + z$$

$$(ii) x(y \cdot z) = (x \cdot y)z$$

Q. 16

• Distributive Law

$$(i) x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

$$(ii) x + (y \cdot z) = (x + y) \cdot (x + z)$$

lets take example (i) to prove

x	y	z	y+z	x \cdot (y+z)	xy	xz	xy+xz
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	1
1	1	0	1	1	0	0	1
1	1	1	1	1	1	1	1

• Absorption Law

$$(i) x + xy = x \quad (ii) x \cdot (x+y) = x$$

$$\text{L.H.S.} = x + xy \quad \text{R.H.S.} = x \cdot x + x \cdot y$$

$$= x(1+y) \quad = x + xy$$

$$= x \quad = x(1+y)$$

$$= \text{R.H.S.} \quad = x = \text{R.H.S.}$$

• Commutative Law

$$(i) x \cdot y = y \cdot x$$

$$(ii) x+y = y+x$$

Q.16 Explain foll. terms :

- Minterms

A minterm is a Boolean expression resulting in 1 for the output of a single cell & 0s for all other cells in a K-map or Truth Table.

- Maxterms

Maxterm is a sum of  $m$  Boolean variables uncomplemented or complemented but not repeated in a Boolean function of  $n$  variables with  $n$  variables,  $2^n$  diff. maxterms are possible.

- Canonical Form

The truth table then converts to a digital logic circuits which is in the form of two level AND-OR circuit & when expressed in Boolean algebraic form is a sum of Product. In an effort to standardize the expression of the T.I the form used is termed as canonical form.

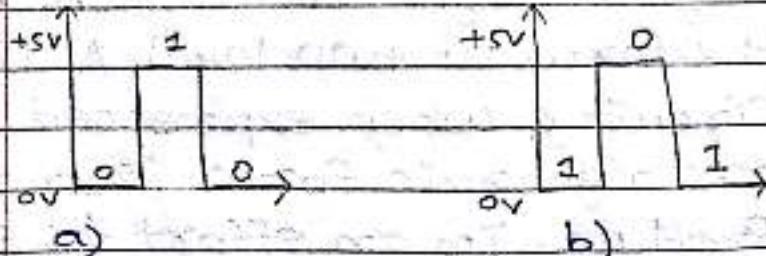
- Standard Forms

- Implementations of logic function based on canonical forms are called two-level implementation
- It is possible to simplify canonical forms into standard forms.
- Sum of minterms to sum of product (SOP) & product of maxterms to product of sum (POS)

- Positive logic System

A physical gate has two different interpretations depending on if positive or negative logic is used.

- a) Positive logic inputs & outputs are also called active high.
- b) Negative logic inputs & output are also called active low.



- Negative Logic System

Q.17 Define

- propos
- prop
- is th
- whe
- Conn
- the t
- gate

- Pow
- Ever
- am
- this
- car
- sta

(E) 15  
Dy

Q.18 H

Q.19 W

Q.17 Define foll. terms:

- Propagation Delay

Propagation Delay or Gate Delay, is the length of time which starts when the input to a logic gate becomes stable & valid to change to the time that the output of that logic gate is stable & valid to change.

- Power Dissipation

Every digital gate will require some amount of power. It must dissipate this power in the form of heat. We consider two types of power.

static  $P_0 \rightarrow P.D$  when gate is not changing state.

Dynamic  $P_0 \rightarrow P.D$  when gate is changing state.

$$\Rightarrow \text{Dynamic } P_0 \geq \text{Static } P_0$$

Q.18 How many function can be formed with  $n$  variable in Boolean algebra?

$$2^{2^n}$$

Q.19 What is the difference between Canonical and Standard form?

Canonical Forms  $\rightarrow$  Canonical form of Boolean Algebra are basic forms that obtains from reading a function from truth table. These contain least number of literals because each min & max terms must contains by definition, all variable either complemented or true form.

Standard form  $\rightarrow$  The terms that form the function contain one, two or any number of literals

Q.20 Prove that Sum of all minterms of Boolean function of 2 variable is 1.

2 Variable K-Map :-  $F = \Sigma(0, 1, 2, 3)$

A \ B    0    1

0    1<sub>0</sub>    1<sub>1</sub>

1    1<sub>2</sub>    1<sub>3</sub>

$F = 1$  = proved

Q.21 Prove that Product of all maxterms of Boolean function of 2 variable is 0.

A \ B    0    1

0    0    0

1    0    0

$F = 0$  = proved

Q.22 Show that dual of exclusive-OR gate is equal to its complement.

$$F = xy' + x'y$$

$$\therefore \text{Dual} = (x+y')(x'+y)$$

$$xx' + xy + x'y' + yy'$$

$$xy + x'y' \rightarrow \textcircled{1}$$

$$F = xy' + x'y$$

$$\therefore \text{complement} = (x'+y) \cdot (x+y')$$

$$= x'x + x'y' + xy + yy'$$

$$= x'y' + xy \rightarrow \textcircled{2}$$

$\therefore$  From  $\textcircled{1}$  &  $\textcircled{2}$

Dual of Ex-OR = Complement of Ex-OR

Q.23 State limitation of K-map.

$\Rightarrow$  The K-Map does not necessarily 'fail' for higher dimensions. The problem is that it is so difficult to visualize for more than 5 variables.

$\Rightarrow$  A 4 Variable K-Map is 2-D & easy to visualize.

$\Rightarrow$  Just getting equations correct with more than 5 variables is difficult enough using K-Map

Q.24 Which code is used in K-Map?

$\Rightarrow$  Gray Code is used in K-Map.

Q. 25 Why NAND and NOR gate is called universal gate.

Each & every gate, circuit is made using NAND & NOR gate which may not possible for other gates to make such circuits. That's why NAND & NOR gate is called Universal Gate.

## Assignment - 2

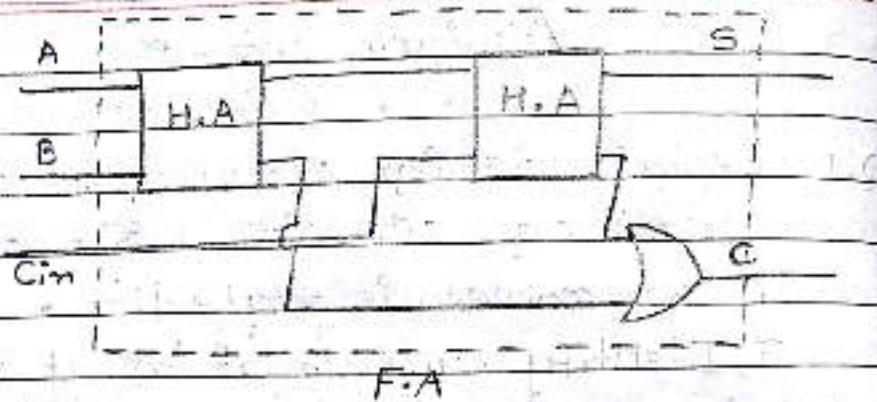
**Q.1** Explain design procedure of combinational circuit.

- i) The problem is stated.
- ii) Identify number of input or output required.
- iii) Assign letter symbol to each input and output.
- iv) Make a truth table which shows relationship between input and output.
- v) Obtain simplified boolean function for each output
- vi) Draw circuit diagram.

**Q.2** Design a full-adder with two-half adders and an OR gate.

i/P      o/P

A	B	C <sub>in</sub>	S	C	$S = A'B'C_{in} + A'B'C_{in}' + AB'C_{in}' + AB'C_{in}$
0	0	0	0	0	$= A'(B \oplus C) + A(B \oplus C)'$
0	0	1	1	0	$= A \oplus B \oplus C_{in}$
0	1	0	1	0	$C = A'B'C_{in} + A'B'C_{in}' + AB'C_{in}' + AB'C_{in}$
0	1	1	0	1	$= BC_{in} + A(B \oplus C_{in})$
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	
1	1	1	1	1	



Q.3 Design a full subtractor with two half subtractor and an OR gate.

I/P      O/P

A    B    Gnd    D    B

$$0 \ 0 \ 0 \ 0 \ 0 \quad D = A'B'C_{in} + A'Bc_{in} + AB'C_{in} +$$

$$0 \ 0 \ 1 \ 1 \ 1 \quad A'BC_{in}$$

$$0 \ 1 \ 0 \ 1 \ 1 \quad = A'(B \oplus C_{in}) + A(B \oplus C_{in})$$

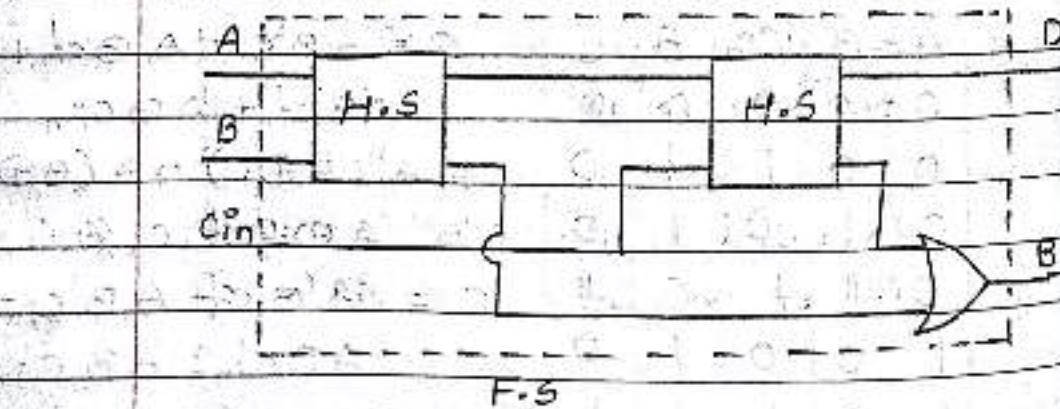
$$0 \ 1 \ 1 \ 0 \ 1 \quad = A \oplus B \oplus C_{in}$$

$$1 \ 0 \ 0 \ 1 \ 0 \quad B = A'B'C_{in} + A'BC_{in} + A'Bc_{in}$$

$$1 \ 0 \ 1 \ 0 \ 0 \quad ABC_{in}$$

$$1 \ 1 \ 0 \ 0 \ 0 \quad = A'(B \oplus C_{in}) + BC_{in}$$

$$1 \ 1 \ 1 \ 1 \ 1$$



Q.4 Design 4-bit binary to gray code converters.

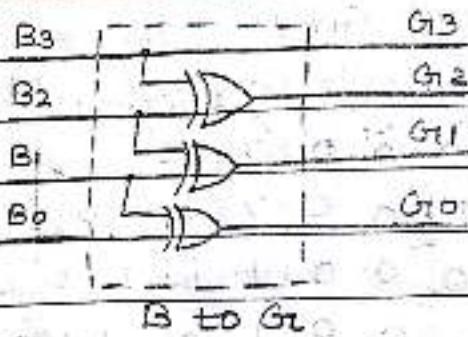
## Binary      Gray

	B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>	G <sub>3</sub>	G <sub>2</sub>	G <sub>1</sub>	G <sub>0</sub>
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	0	0	0
9	1	0	0	1	1	0	1	0
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	0	1
12	1	1	0	0	1	0	1	1
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	0	0	0	0

B <sub>3</sub> B <sub>2</sub> B <sub>3</sub> B <sub>2</sub> 00 01 11 10				B <sub>3</sub> B <sub>2</sub> B <sub>3</sub> B <sub>2</sub> 00 01 11 10				
00	0	1	13	10	00	0	1	3
01	14	15	7	6	01	4	15	7
11	12	13	15	14	11	12	13	15
10	8	9	11	10	10	8	9	11

$$G_1 = B_2 B_1' + B_2' B_1 \\ = B_2 \oplus B_1$$

$$G_0 = B_1 B_0' + B_1' B_0 \\ = B_1 \oplus B_0$$



Decimal  
Digits

B<sub>3</sub>  
B<sub>2</sub>

0 0

1 0

2 0

3 0

4 0 1

5 0

6 0

7 0

8 1

9 1

B<sub>3</sub>  
B<sub>2</sub>

Q. 5 What is disadvantage of Binary Parallel Adder? How it can be overcome?

Disadvantage of binary parallel adder is that next bit will not generate infinite the first bit calculate the answer and carry part to the next bit.

Hence for an N-bit parallel adder, the total delay time is equal to  $Nt_p$ .

The parallel adder has a larger propagation delay and hence low speed operation. This limitation is overcome in parallel adder.

Q. 6 Design Combinational circuit to Convert BCD number to Excess-3

Decimal  
Digits

	BCD	Excess-3	$B_3 B_0$	$B_3 B_1$	$B_0$	$E_0$	$E_1$	$E_2$	$E_3$
0	0 0 0 0	0 0 1 1	0 0	0 1	1	0 0	0 1	1 0	1 1
1	0 0 0 1	0 1 0 0	0 1	1 0	0	0 1	1 1	1 2	1 3
2	0 0 1 0	0 0 1 0	1 1	X <sub>13</sub>	X <sub>14</sub>	1 1	X <sub>13</sub>	X <sub>14</sub>	X <sub>15</sub>
3	0 0 1 1	0 1 1 0	1 0	X <sub>13</sub>	X <sub>14</sub>	1 0	X <sub>13</sub>	X <sub>14</sub>	X <sub>15</sub>
4	0 1 0 0	0 1 1 1	$E_3 = B_3 B_2 + B_3' B_2 B_1 + B_1' B_2 B_0$	$B_3 B_2$	$B_0$	0 0	0 1	1 1	1 0
5	0 1 0 1	1 0 0 0	$B_3 B_2$	$B_0$	0 1	1 1	1 0	1 0	1 0
6	0 1 1 0	1 0 0 1	0 0	0 1	1 1	1 0	1 0	1 0	1 0
7	0 1 1 1	1 0 1 0	0 1	1 0	1 1	1 0	1 0	1 0	1 0
8	1 0 0 0	1 0 1 1	1 1	X <sub>12</sub>	X <sub>13</sub>	X <sub>15</sub>	X <sub>14</sub>	1 0	1 0
9	1 0 0 1	1 1 0 0	1 0	X <sub>12</sub>	X <sub>13</sub>	X <sub>14</sub>	X <sub>15</sub>	1 0	1 0

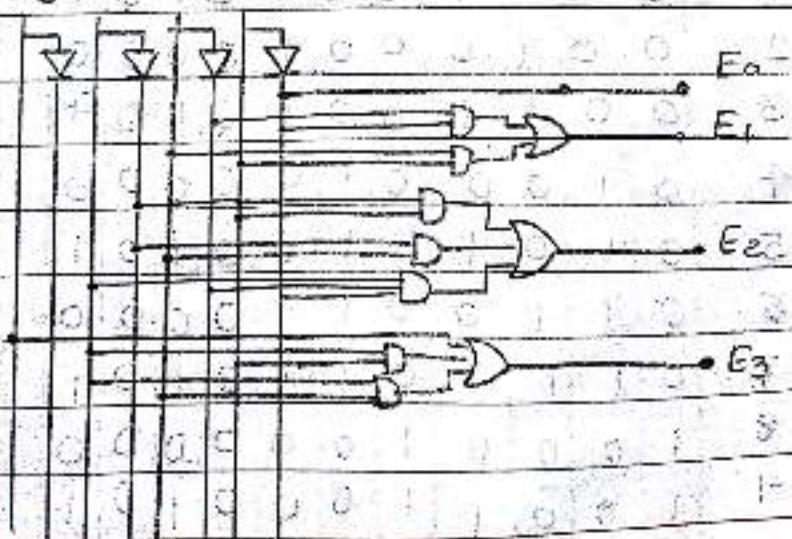
$$E_2 = B_2' B_0 + B_2' B_1 + B_2 B_1' B_0$$

$B_3 B_0$	0 0	0 1	1 1	1 0	$B_3 B_2$	0 0	0 1	1 1	1 0
0 0	0	1	3	2	0 0	0	1	3	2
0 1	4	5	12	6	0 1	4	5	12	6
1 1	X <sub>12</sub>	X <sub>13</sub>	X <sub>15</sub>	X <sub>14</sub>	1 1	X <sub>12</sub>	X <sub>13</sub>	X <sub>15</sub>	X <sub>14</sub>
1 0	8	9	X <sub>11</sub>	X <sub>10</sub>	1 0	8	9	X <sub>11</sub>	X <sub>10</sub>

$$E_0 = B_0$$

$$E_1 = B_1 B_0' + B_1 B_0$$

$B_3 \ B_2 \ B_1 \ B_0$



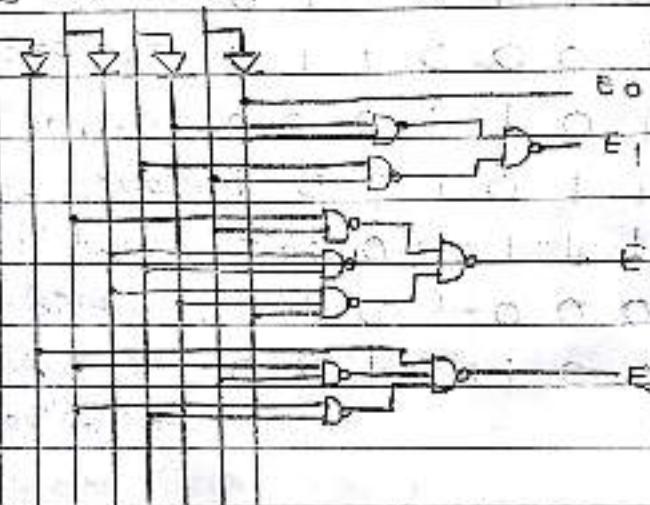
Q. 7 Design BCD to Excess-3 Code converter using min. no of NAND Gates.

$$E_0 = B_0', \quad E_1 = B_1 B_0' + B_1 B_0,$$

$$E_2 = B_2' B_0 + B_2' B_1 + B_2' B_1 B_0'$$

$$E_3 = B_3 + B_2 B_0 + B_2 B_1$$

B<sub>3</sub> B<sub>2</sub> B<sub>1</sub> B<sub>0</sub>



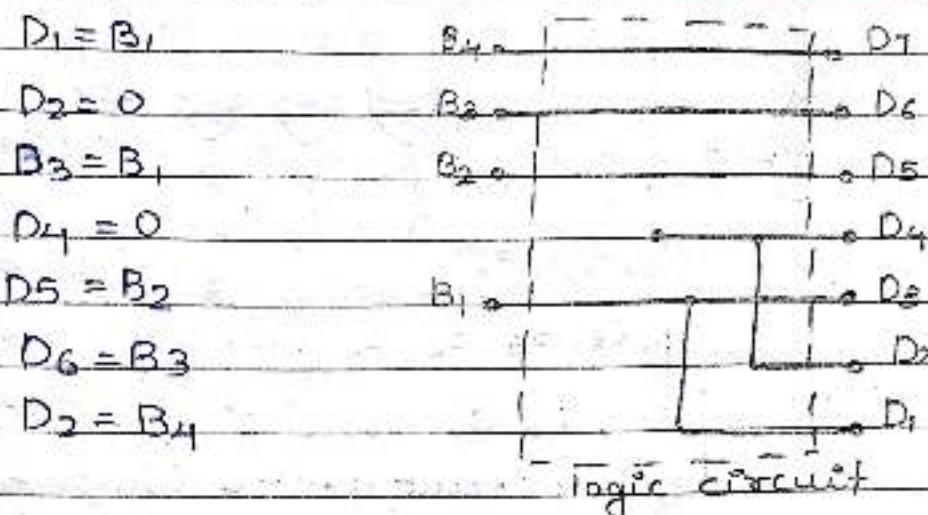
Q. 8 Design a Combinational Ckt which multiply a decimal digit by 5.

Decimal Input      Output

Digit      B<sub>4</sub> B<sub>3</sub> B<sub>2</sub> B<sub>1</sub> D<sub>1</sub> D<sub>2</sub> D<sub>3</sub> D<sub>4</sub> D<sub>5</sub> D<sub>6</sub>

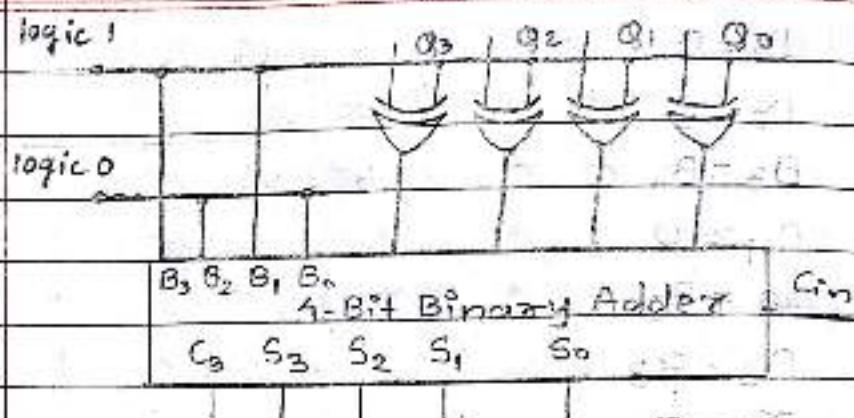
	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	1	0	1
2	0	0	1	0	0	0	1	0	0	0	0
3	0	0	1	1	0	0	1	0	1	0	1
4	0	1	0	0	0	1	0	0	0	0	0
5	0	1	0	1	0	1	0	0	1	0	1
6	0	1	1	0	0	1	1	0	0	0	0
7	0	1	1	1	0	1	1	0	1	0	1
8	1	0	0	0	1	0	0	0	0	0	0
9	1	0	0	1	1	0	0	0	1	0	1

Q. 9 Design the The ok from



Q.9 Design a Combinational CKT that generate the 9's Complement of a BCD digit.  
 The 9's Complement of any digit can be obtained by subtracting the digit from 9.

Digit	9's Complement
0	9-0=9
1	9-1=8
2	9-2=7
3	9-3=6
4	9-4=5
5	9-5=4
6	9-6=3
7	9-7=2
8	9-8=1
9	9-9=0



logic circuit of 9's Complement.

Q.10 Design a Combinational Ckt that accepts a three bit binary no. & generates an output binary no. equal to the square of the input number.

Deci. Digit	Input			Output						
	$B_3$	$B_2$	$B_1$	Squa	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	1
2	0	1	0	4	0	0	0	1	0	0
3	0	1	1	9	0	0	1	0	0	1
4	1	0	0	16	0	1	0	0	0	0
5	1	0	1	25	0	1	1	0	0	1
6	1	1	0	36	1	0	0	1	0	0
7	1	1	1	49	1	1	0	0	0	1
8										

$$\begin{aligned}
 D_1 &= B_1 \quad \therefore D_2 = 0, \quad D_3 = B_3' B_2 B_1' + \\
 &\quad B_3 B_2 B_1' \\
 &= B_2 B_1'
 \end{aligned}$$

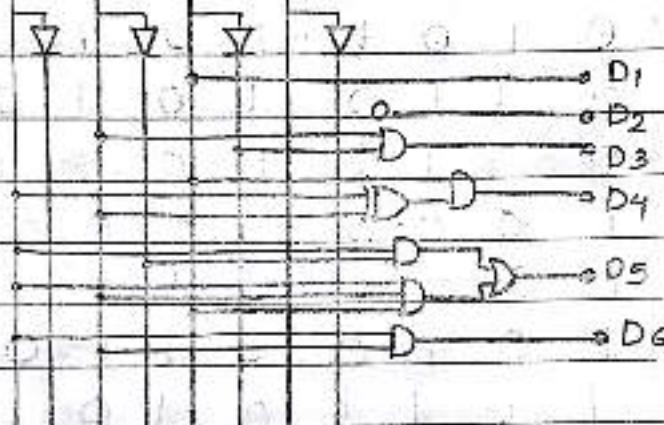
Q-11

$$D_4 = B_3'B_2B_1 + B_3B_2'B_1 \\ = B_1(B_3 \oplus B_2)$$

$$D_5 = B_3B_2'B_1' + B_3B_2'B_1 + B_3B_2B_1 \\ = B_3B_2' + B_3B_2B_1$$

$$D_6 = B_3B_2B_1' + B_3B_2B_1 \\ = B_3B_2$$

$B_3 \quad B_2 \quad B_1 \quad B_0$



Q.11 Design a Combinational circuit whose input is four bit binary no. & output is the 2's Complement.

		$B_2B_1$				$B_4B_3$				$B_2B_1$				$B_4B_3$			
		00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
		00	011112	13	14	00	011112	13	14	00	011112	13	14	00	011112	13	14
		01	415165	7	6	01	415165	7	6	01	415165	7	6	01	415165	7	6
		11	12111314	15	16	11	12111314	15	16	11	12111314	15	16	11	12111314	15	16
		10	8191110	17	18	10	8191110	17	18	10	8191110	17	18	10	8191110	17	18

$$D_1 = B_1$$

$$D_2 = B_2'B_1 + B_2B_1'$$

		$B_2B_1$				$B_4B_3$				$B_2B_1$				$B_4B_3$			
		00	01	11	10	00	01	11	10	00	01	11	10	00	01	11	10
		00	011112	13	14	00	011112	13	14	00	011112	13	14	00	011112	13	14
		01	11415165	7	6	01	11415165	7	6	01	11415165	7	6	01	11415165	7	6
		11	12111314	15	16	11	12111314	15	16	11	12111314	15	16	11	12111314	15	16
		10	8191110	17	18	10	8191110	17	18	10	8191110	17	18	10	8191110	17	18

$$D_3 = B_3B_2'B_1' + B_3B_2 + B_3'B_2$$

Dig.	Input	2's comp. Output
	$B_4 \ B_3 \ B_2 \ B_1$	$D_4 \ D_3 \ D_2 \ D_1$

0	0 0 0 0	0 0 0 0
---	---------	---------

1	0 0 0 1	1 1 1 1
---	---------	---------

2	0 0 1 0	1 1 1 0
---	---------	---------

3	0 0 1 1	1 1 0 1
---	---------	---------

4	0 1 0 0	1 1 0 0
---	---------	---------

5	0 1 0 1	1 0 1 1
---	---------	---------

6	0 1 1 0	1 0 1 0
---	---------	---------

7	0 1 1 1	1 0 0 1
---	---------	---------

8	1 0 0 0	1 0 0 0
---	---------	---------

9	1 0 0 1	0 1 1 1
---	---------	---------

10	1 0 1 0	0 1 1 0
----	---------	---------

11	1 0 1 1	0 1 0 1
----	---------	---------

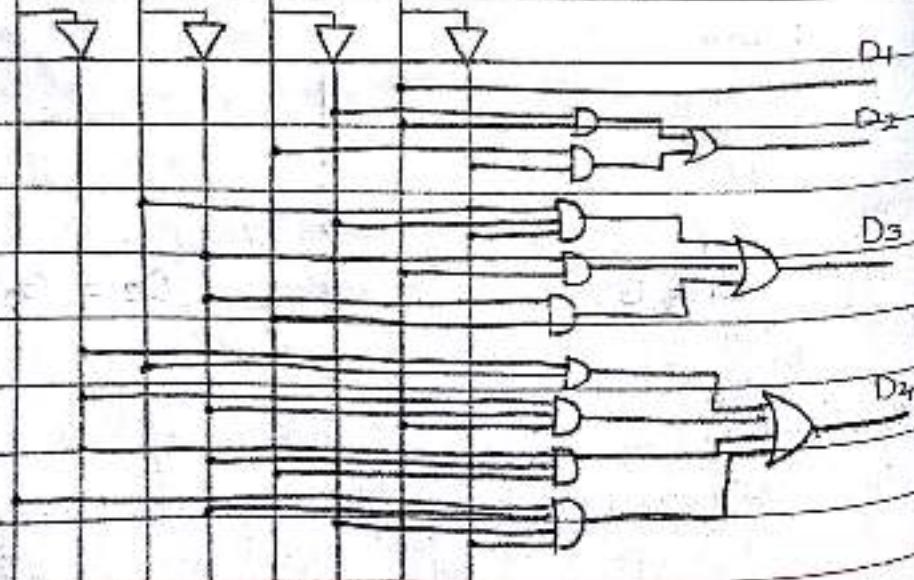
12	1 1 0 0	0 1 0 0
----	---------	---------

13	1 1 0 1	0 0 1 1
----	---------	---------

14	1 1 1 0	0 0 1 0
----	---------	---------

15	1 1 1 1	0 0 0 1
----	---------	---------

$B_4$	$B_3$	$B_2$	$B_1$
-------	-------	-------	-------



Q. 12 H

Q. 13 C

Q. 14 C

Q. 12 How many don't care condition are there in BCD adder?

6 condition

Q. 13 What is carry propagation?

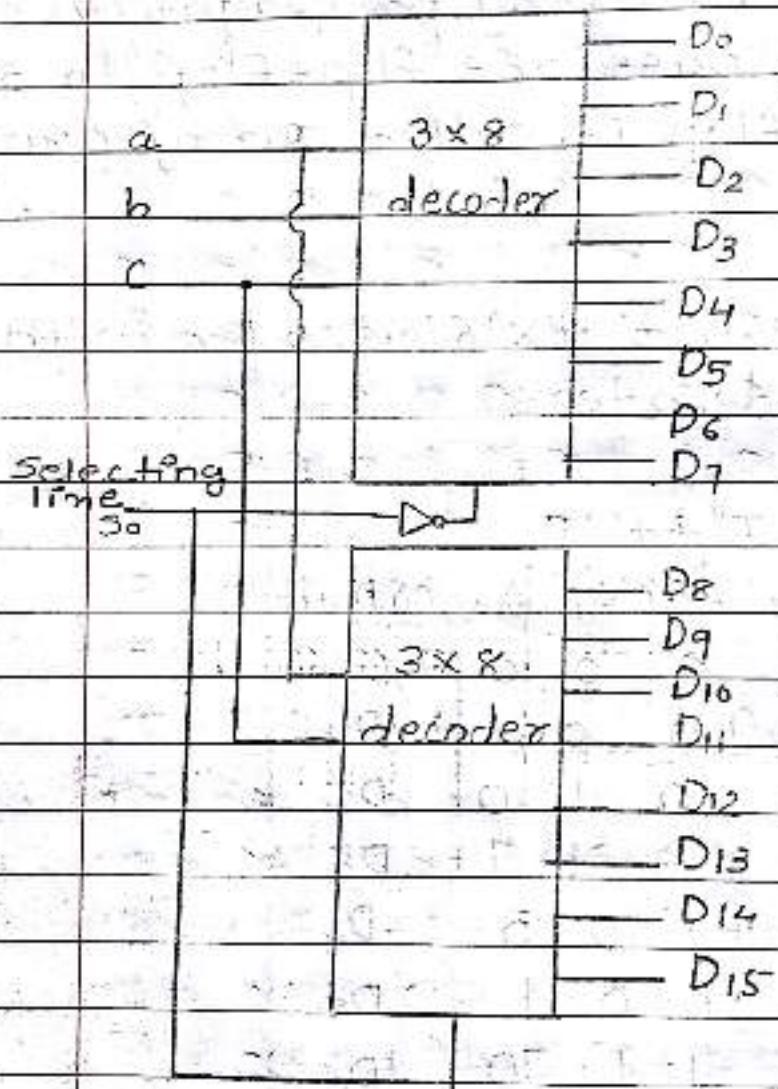
Carry pass one flip-flop to second flip-flop is called carry propagation.

Q. 14 Construct  $4 \times 16$  decoder with two  $3 \times 8$  decoders.

Digit	Input				Output
	$A_3$	$A_2$	$A_1$	$A_0$	
0	0	0	0	0	$D_0$
1	0	0	0	1	$D_1$
2	0	0	1	0	$D_2$
3	0	0	1	1	$D_3$
4	0	1	0	0	$D_4$
5	0	1	0	1	$D_5$
6	0	1	1	0	$D_6$
7	0	1	1	1	$D_7$
8	1	0	0	0	$D_8$
9	1	0	0	1	$D_9$
10	1	0	1	0	$D_{10}$
11	1	0	1	1	$D_{11}$
12	1	1	0	0	$D_{12}$
13	1	1	0	1	$D_{13}$

14	1	1	1	0	D14
15	1	1	1	1	D15

Logic circuit

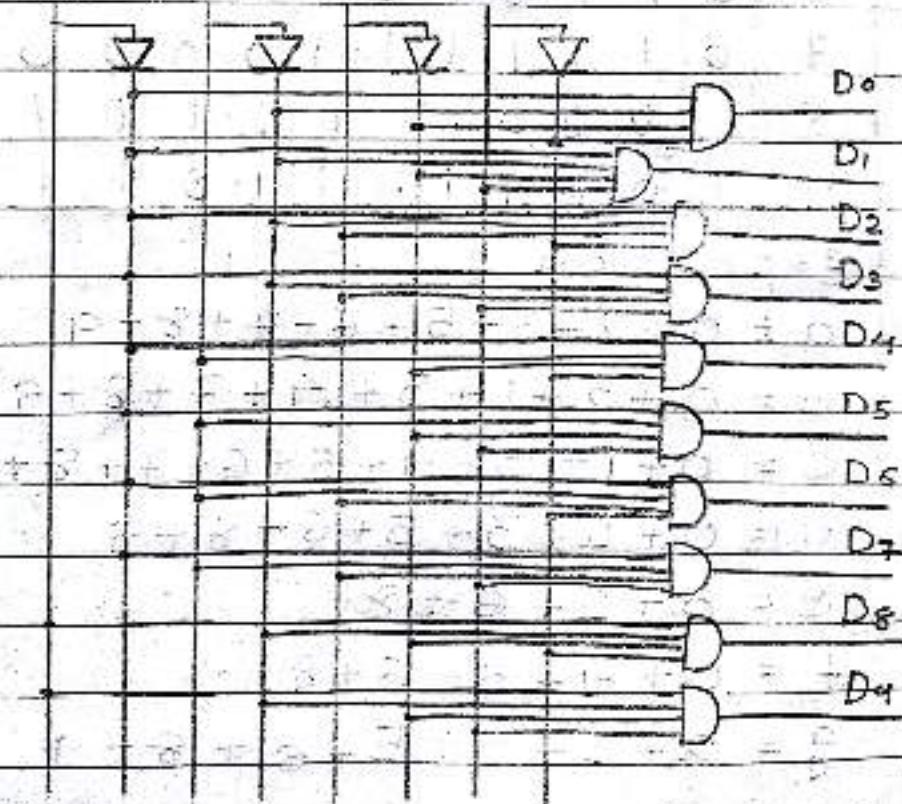


Q. 15 Design a BCD to decimal decoder

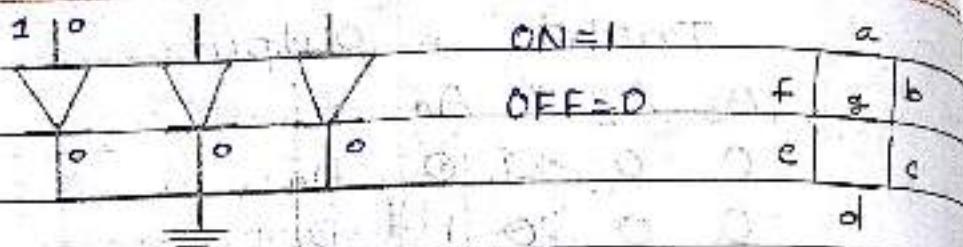
Q.

Input	Output
A <sub>3</sub> A <sub>2</sub> A <sub>1</sub> A <sub>0</sub>	
0 0 0 0	D <sub>0</sub>
0 0 0 1	D <sub>1</sub>
0 0 1 0	D <sub>2</sub>
0 0 1 1	D <sub>3</sub>
0 1 0 0	D <sub>4</sub>
0 1 0 1	D <sub>5</sub>
0 1 1 0	D <sub>6</sub>
0 1 1 1	D <sub>7</sub>
1 0 0 0	D <sub>8</sub>
1 0 0 1	D <sub>9</sub>

A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub> | D<sub>0</sub> | D<sub>1</sub> | D<sub>2</sub> | D<sub>3</sub> | D<sub>4</sub> | D<sub>5</sub> | D<sub>6</sub> | D<sub>7</sub> | D<sub>8</sub> | D<sub>9</sub>



Q.16 Explain BCD to 7-Segment display decoder in common Cathode Config.



BCD	B3	B2	B1	B0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Q.17

(a)

Expression

$$a = 0 + 2 + 3 + 5 + 6 + 7 + 8 + 9$$

$$b = 0 + 2 + 1 + 3 + 4 + 7 + 8 + 9$$

$$c = 0 + 1 + 3 + 4 + 5 + 6 + 7 + 8 + 9$$

$$d = 0 + 2 + 3 + 5 + 6 + 8 + 9$$

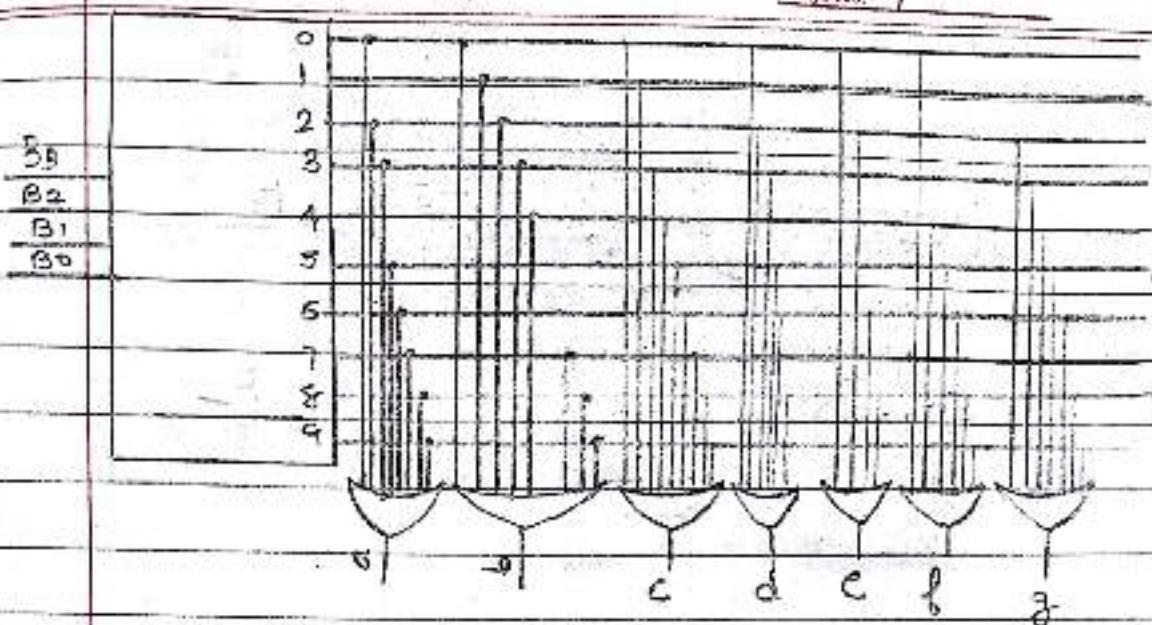
$$e = 0 + 2 + 6 + 8$$

$$f = 0 + 4 + 5 + 6 + 8 + 9$$

$$g = 2 + 3 + 4 + 5 + 6 + 8 + 9$$

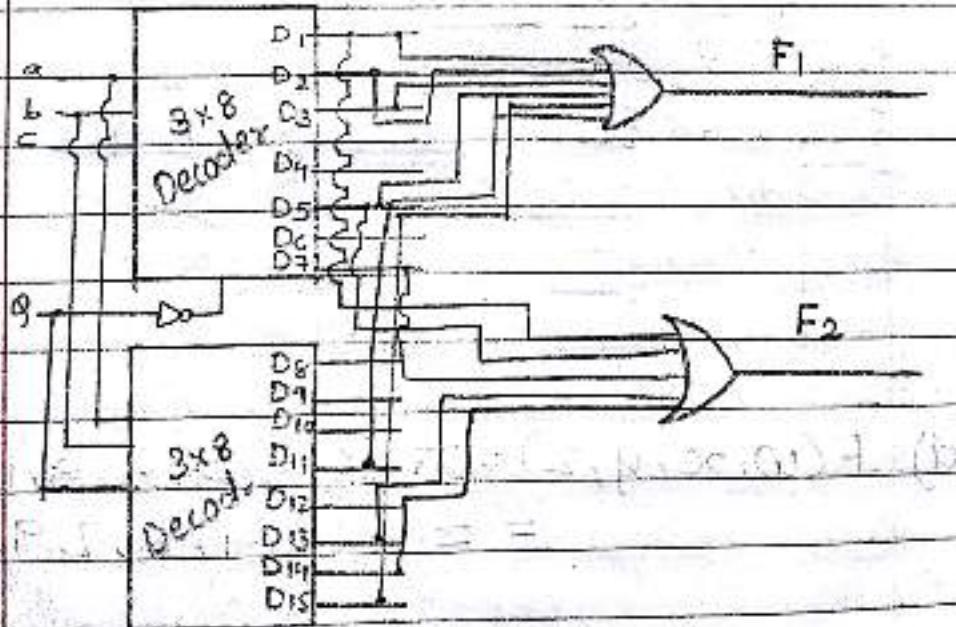
Transistor output logic design of 7 segment display

Date \_\_\_\_\_ Page No. \_\_\_\_\_

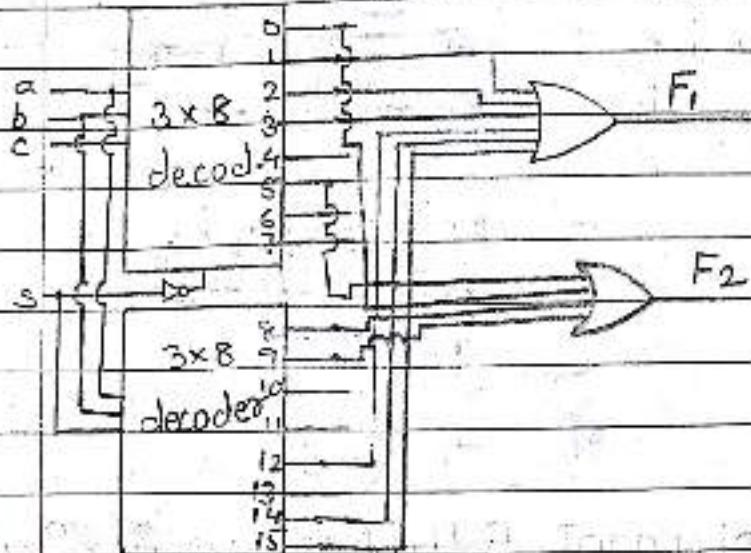


Q.17 Implement foll. Boolean fun<sup>n</sup> using Decoders.

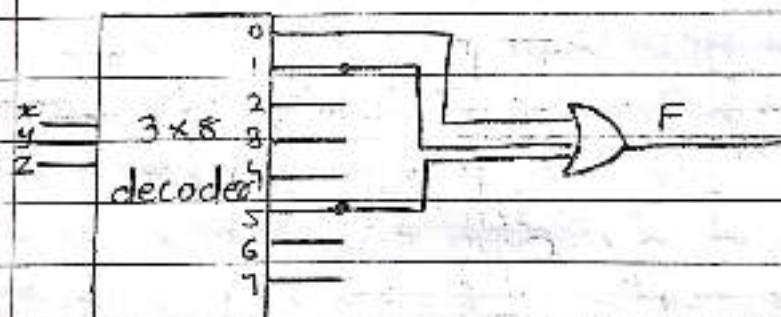
a)  $F_1(a, b, c, d) = \sum(1, 3, 5, 2, 11, 15)$  and  
 $F_2(a, b, c, d) = \sum(1, 5, 7, 14, 13)$



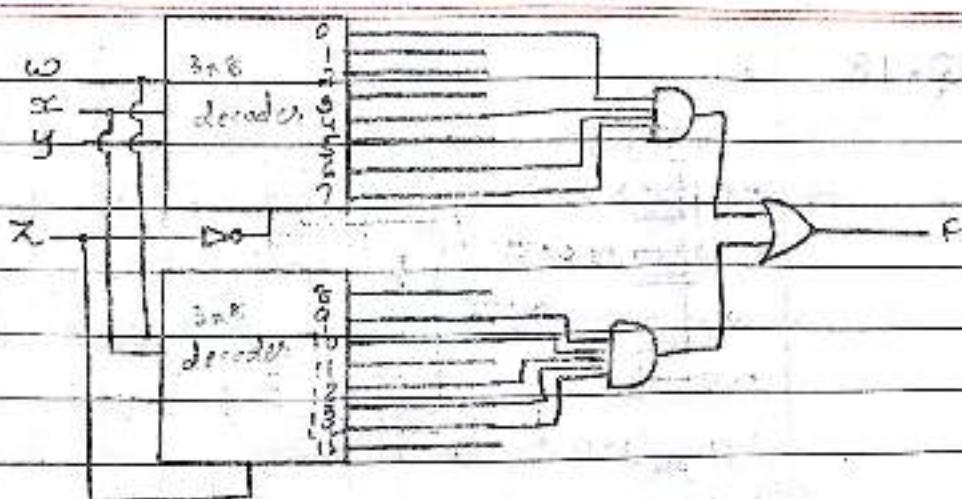
b)  $F_1(a, b, c, d) = \sum(1, 12, 3, 2, 14, 15)$  and  
 $F_2(a, b, c, d) = \sum(0, 5, 8, 9)$



c)  $F = \sum m(0, 2, 3, 5, 8, 11, 15)$



d)  $F(x, y, z) = \sum m(0, 4, 6, 7, 9, 10, 12, 13)$



e) full adder

Input      Output

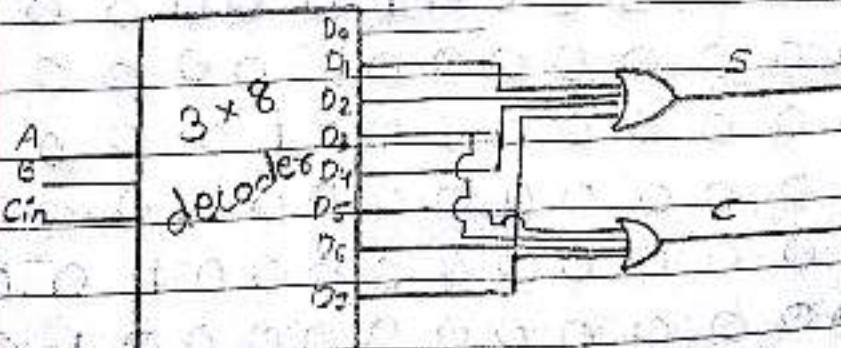
A	B	$Cin$	S	C
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	0	1	0
1	0	1	0	0
1	0	0	0	1
1	1	0	1	0
1	1	1	1	1

$$S = \sum m(1, 2, 4, 7)$$

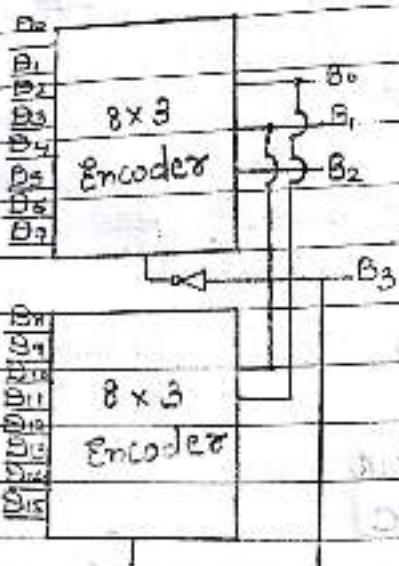
$$C = \sum m(3, 5, 6, 7)$$

5)

12, 13, 14)



Q-18



$B_0$	$B_1$	$B_2$	$B_3$	$B_4$	$B_5$	$B_6$	$B_7$	$B_8$	$B_9$	$B_{10}$	$B_{11}$	$B_{12}$	$B_{13}$	$B_{14}$	$B_{15}$	$B_2$	$B_2$	$B_1B_2$
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	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	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	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	0	0	0	0	0	1	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

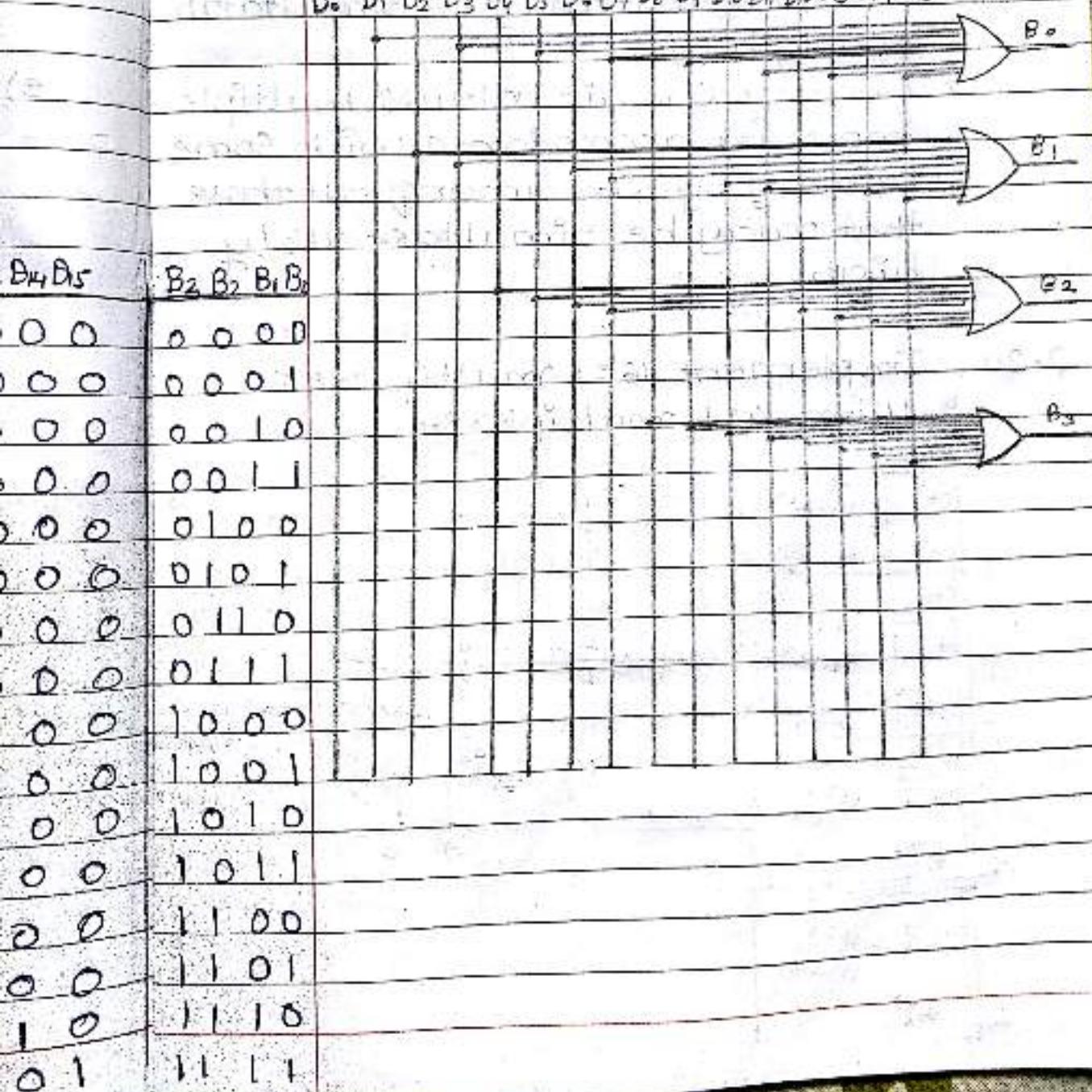
$$B_0 = D_1 + D_3 + D_5 + D_7 + D_9 + D_{11} + D_{13} + D_{15}$$

$$B_1 = D_2 + D_3 + D_6 + D_7 + D_{10} + D_{11} + D_{14} + D_{15}$$

$$B_2 = D_4 + D_5 + D_6 + D_7 + D_{12} + D_{13} + D_{14} + D_{15}$$

$$B_3 = D_8 + D_9 + D_{10} + D_{11} + D_{12} + D_{13} + D_{14} + D_{15}$$

D<sub>0</sub> D<sub>1</sub> D<sub>2</sub> D<sub>3</sub> D<sub>4</sub> D<sub>5</sub> D<sub>6</sub> D<sub>7</sub> D<sub>8</sub> D<sub>9</sub> D<sub>10</sub> D<sub>11</sub> D<sub>12</sub> D<sub>13</sub> D<sub>14</sub> D<sub>15</sub>

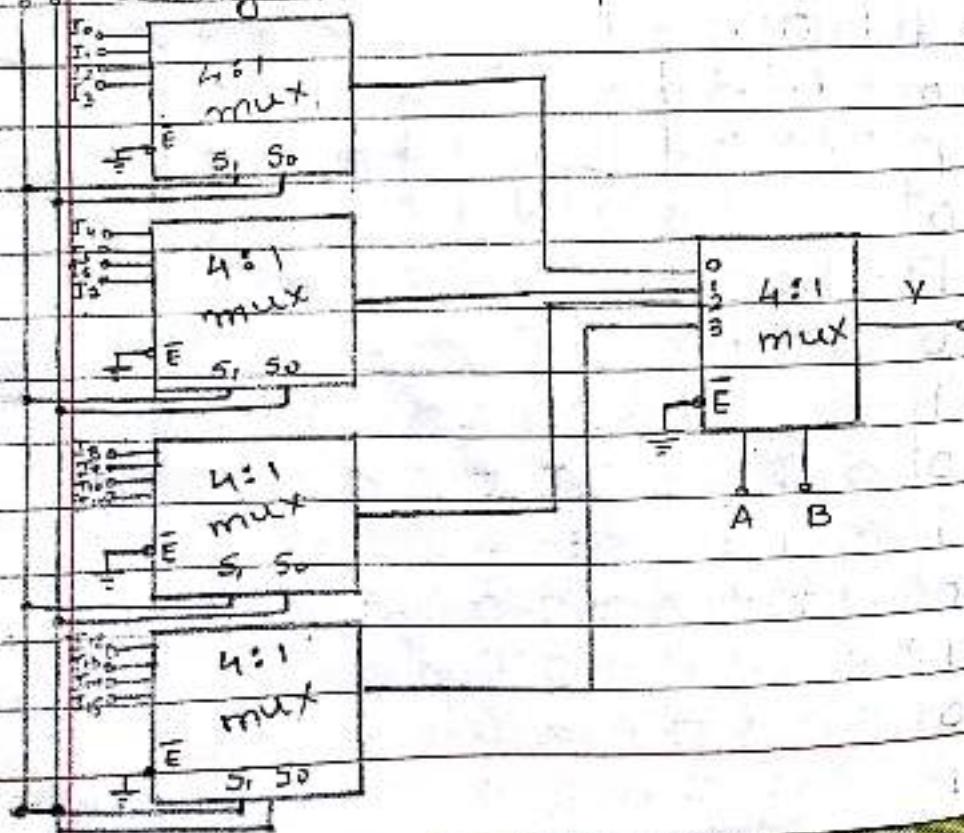


Q.19 What is difference between encoder and priority encoder.

→ A priority encoder is a logic circuit that respond to just one input in accordance with some priority system among all these that may be simultaneously high.

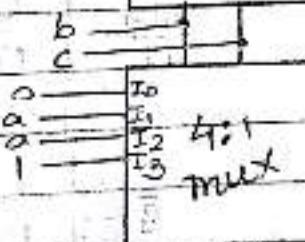
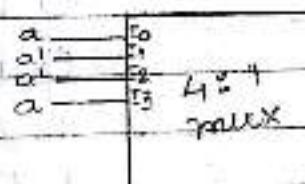
→ Encoder circuit allows multiple input in accordance with some priority system among all those that may be simultaneously high.

Q.20 Implement 16:1 multiplexer using 4:1 multiplexer.



Q.21 Implement a full adder circuit using two 4:1 mux

Input				
a	b	c	s	c
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
1	0	1	1	0
1	0	0	0	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1



For S

	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
a <sup>1</sup>	0	1	2	3
a	4	5	6	7
	a	a	a	a

For C

	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
a <sup>1</sup>	0	1	2	3
a	4	5	6	7
	0	a	a	1

Q.22. Implement foll. Boolean fun'ns using  
multiple fns.

a)  $F_1(a, b, c, d) = \sum(1, 3, 5, 2, 11, 15)$  and  
 $F_2(a, b, c, d) = \sum(1, 5, 7, 13, 14)$

Input	Output	Input	Output
0 0 0 0	0	0 0 0 0	0 0 0 0
0 0 0 1	1	0 0 0 1	0 0 0 1
0 0 1 0	1	0 0 1 0	0 0 1 0
0 0 1 1	1	0 0 1 1	0 0 1 0
0 1 0 0	0	0 1 0 0	0 1 0 0
0 1 0 1	1	0 1 0 1	0 1 0 1
0 1 1 0	0	0 1 1 0	0 1 1 0
0 1 1 1	0	0 1 1 1	1 0 0 0
1 0 0 0	0	1 0 0 0	1 0 0 0
1 0 0 1	0	1 0 0 1	1 0 0 1
1 0 1 0	0	1 0 1 0	1 0 1 0
1 0 1 1	1	1 0 1 1	1 0 1 0
1 1 0 0	0	1 1 0 0	1 1 0 0
1 1 0 1	0	1 1 0 1	1 1 0 1
1 1 1 0	1	1 1 1 0	1 1 1 0
1 1 1 1	0	1 1 1 1	1 1 1 0

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$a'$	0	1	2	3	4	5	6	7
$a$	8	9	10	11	12	13	14	15
	0	$a'$	$a'$	1	0	$a'$	0	$a$

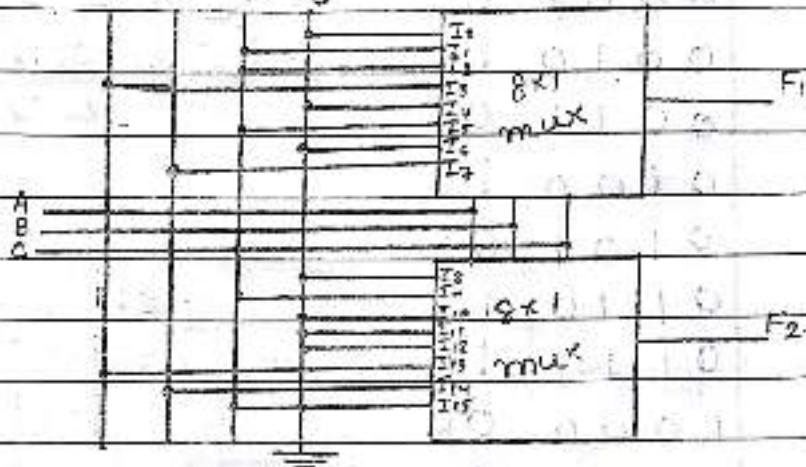
I<sub>0</sub> I<sub>1</sub> I<sub>2</sub> I<sub>3</sub> I<sub>4</sub> I<sub>5</sub> I<sub>6</sub> I<sub>7</sub>

a'	0	1	2	3	4	5	6	7
----	---	---	---	---	---	---	---	---

a	8	9	10	11	12	13	14	15
---	---	---	----	----	----	----	----	----

0	a'	0	0	0	1	a	a'
---	----	---	---	---	---	---	----

1 a a' 0

F<sub>1</sub>F<sub>2</sub>

b)  $F = x'y'z' + x'y'z + xy'z' + xy'z = \sum(0, 1, 4, 5)$

x	y	z	F
---	---	---	---

I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
----------------	----------------	----------------	----------------

0	0	1	1
---	---	---	---

a'	0	1	2	3
----	---	---	---	---

0	1	0	0
---	---	---	---

a	0	1	6	7
---	---	---	---	---

0	1	1	0
---	---	---	---

1	1	0	0
---	---	---	---

1	0	0	1
---	---	---	---

1	1	0	0
---	---	---	---

I <sub>0</sub>			
----------------	--	--	--

1	1	1	0
---	---	---	---

I <sub>1</sub>	4:1		
----------------	-----	--	--

1	1	1	0
---	---	---	---

I <sub>2</sub>	mux		
----------------	-----	--	--

1	0	1	1
---	---	---	---

I <sub>3</sub>			
----------------	--	--	--

0	1	1	1
---	---	---	---

--	--	--	--

0	0	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	1	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

1	0	0	1
---	---	---	---

--	--	--	--

1	1	1	1
---	---	---	---

--	--	--	--

</

c)  $F(w, x, y, z) = \pi(1, 2, 3, 5, 8, 11, 15)$   
 $= \sum m(0, 4, 6, 7, 10, 12, 13, 14)$

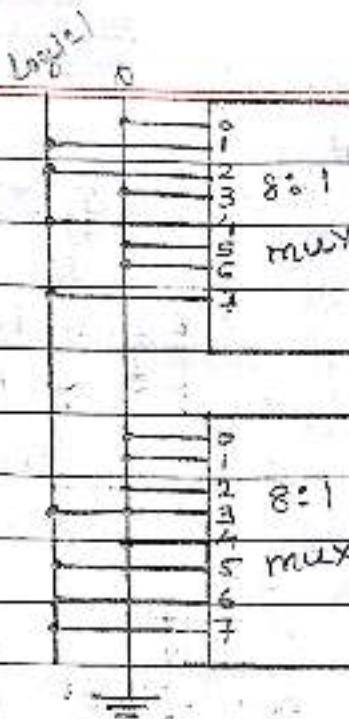
Input	Output		To $I_1, I_2, I_3, I_4, I_5, I_6, I_7$
0 0 0 0	1		
0 0 0 1	0	$a'$	② 1 2 3 ④ 5 ⑥ ⑧
0 0 1 0	0	$a$	⑧ ⑨ ⑩ 11 ⑫ ⑬ ⑭ 15
0 0 1 1	0	$a'$ $a$ $a$ 0 1 $a$ 1 $a$ 1	
0 1 0 0	1		
0 1 0 1	0	$a'$ $a$	$I_0$ $I_1$
0 1 1 0	1	$a$	$I_2$ 8:1
0 1 1 1	1	$a$ $I_3$	$I_4$ mux
1 0 0 0	0	$a$ $I_5$	$I_5$
1 0 0 1	Φ	$a'$ $I_6$	$I_7$
1 0 1 0	1		
1 0 1 1	0		
1 1 0 0	1		
1 1 0 1	1		
1 1 1 0	1		
1 1 1 1	0		

d) Full adder

a	b	c	s	c
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

$S = \sum m(1, 2, 4, 7)$

$C = \sum m(3, 5, 6, 7)$



Q.23 Implement foll. fun<sup>n</sup> using 1:1 multip.  
 $F(A, B, C) = \Sigma(1, 3, 4, 6)$

Input	Outp.	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>
a b c	F				
0 0 0	0	a'	0	①	2 ③
0 0 1	1	a	④	5 ⑥	7
0 1 0	0		a a'	a a'	
0 1 1	1		0	1	1
1 0 0	1				
1 0 1	0	a		I <sub>0</sub>	
1 1 0	1	a'		I <sub>1</sub>	4:1 F
1 1 1	0	a		I <sub>2</sub> MUX	
		a'		I <sub>3</sub>	

Q.24 Implement the foll. fun<sup>n</sup> using 8:1 multiplexer

a)  $F(A, B, C, D) = \Sigma(2, 4, 5, 7, 10, 14)$

Input	Output	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$
A B C D	F	$a'$	0 1	2 3	4 5	6 7		
0 0 0 0	0	$a$	8 9	10 11	12 13	14 15		
0 0 0 1	0							
0 0 1 0	1							
0 0 1 1	0							
0 1 0 0	1							
0 1 0 1	1	0						
0 1 1 0	0	0						
0 1 1 1	1	0						
1 0 0 0	0	$a'$						
1 0 0 1	0	$a$						
1 0 1 0	1	$a'$						
1 0 1 1	0							
1 1 0 0	0							
1 1 0 1	0							
1 1 1 0	1							
1 1 1 1	0							

b)  $F(A, B, C, D) = \Sigma(0, 2, 4, 6, 8, 10, 12, 14)$   
 $= \Sigma(1, 3, 5, 7, 9, 11, 13, 15)$

Q. 25

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$a'$	0 1	2 3	4 5	6 7				
$a$	8 9	10 11	12 13	14 15				
	0 1	0 1	0 1	0 1	0 1	0 1	0 1	0 1

Input	Out				
A	B	C	D	F	
0	0	0	0	0	$I_0$
0	0	0	1	1	$I_1$
0	0	1	0	0	$I_2$
0	0	1	1	1	$I_3$
0	1	0	0	0	$I_4$
0	1	0	1	1	$I_5$
0	1	1	0	0	$I_6$
0	1	1	1	1	$I_7$
1	0	0	0	0	
1	0	0	1	1	
1	0	1	0	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	0	1	1	
1	1	1	0	0	
1	1	1	1	1	

Q.25 Implement the foll. Boolean fun' with  
Rcm  $F_1(A, B, C) = \Sigma(1, 3, 4, 6)$

$$F_2(A, B, C) = \epsilon(0, 5, 6, 7)$$

$$F_3(A, B, C) = \Sigma(2, 3, 5, 6)$$

$T/P = 3$

TIP = 3

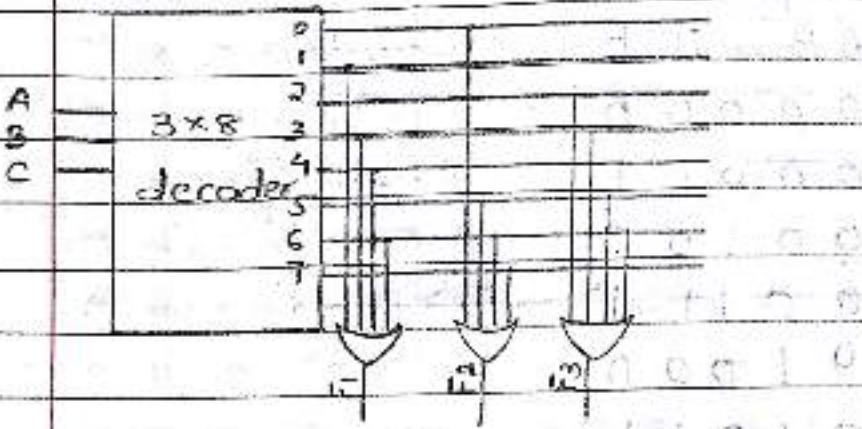
$\text{O/P} = 3$        $\text{Pulse rate} = 100 \text{ beats/min}$

$$ROM = 2^3 \times 3 = 8 \times 3$$

Decoder =  $3 \times 8$

OR = 3      No. of links = 2

Q. 28



Q. 26 What is the minimum size of ROM to implement full adder?

For sum  $F_1(A, B, C) = \Sigma(1, 2, 4, 7)$

For carry  $F_2(A, B, C) = \Sigma(3, 5, 6, 7)$

I/P = 3

O/P = 2

$$\text{ROM} = 2^3 \times 2 = 8 \times 2 = 16$$

No. of link = 16

Q. 27 Classification of ROM.

There are four types of ROM

- Mask Programmable read only memory (MROM)
- Programmable ROM (PROM)
- Erasable Programmable ROM (EPROM)
- Electrically Erasable Programmable ROM (EEPROM)

$$F \times 8 = 8 \times 8 = 64$$

$$8 \times 8 = 64$$

$$S = 40$$

$$20 \times 40 = 800$$

Q.28 Write short note on Read Only Memory (ROM).

A ROM contains permanently or semi-permanently stored data, which can be used from the memory but either cannot be changed at all or cannot be changed without specified equipment. A ROM stores data that are used repeatedly in system applications, such as tables, conversions, or programmed instruction for system initialization and operation. ROMs retain stored data when the power is off and are therefore non-volatile memories.

Q.29 Write short note on Programmable logic array (PLA)

A Programmable logic array device contains an array of AND & OR function whose configuration is done by the user. It typically consists of a set of fusible links which establish or remove the contact of a literal in the AND function or contact of a

product term in the OR function.

The PLA is used to implement a complex combinational circuit.

In design, PLA's are used because it reduce the Space requirement. Several PLAs include storage elements like flip-flops on a Silicon chip, hence it can also be used for Sequential applications.

Q. 30 Implement foll. Boolean fun<sup>n</sup> using PLA

a)  $F_1(A, B, C, D) = \Sigma(1, 3, 5, 2, 11, 15)$

$F_2(A, B, C, D) = \Sigma(1, 5, 7, 14, 13)$

→ For  $F_1$

ab	cd	00	01	11	10
00	0	1	1	1	1
01	1	1	1	1	0
11	1	1	1	1	1

For  $F_2$

ab	cd	00	01	11	10
00	0	1	1	1	1
01	1	1	1	1	0
11	1	1	1	1	1

$$F_1 = A'C'D + A'B'C + ACD : F_2 = A'C'D + BC'D$$

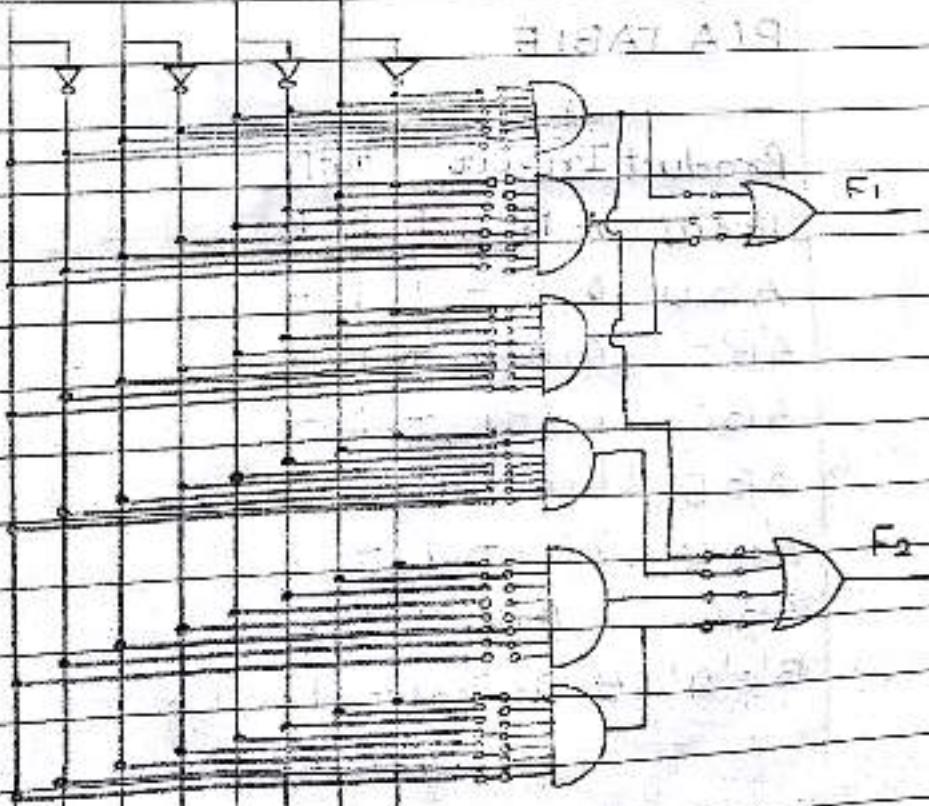
$$= (A'C' + AC)D + A'B'C + A'BD + ABCD$$

$$= (A \oplus C)'D + A'B'C$$

## (a) PLA Table

PRODUCT TERM	A	B	C	D	Input	Output
	F <sub>1</sub>	F <sub>2</sub>				
A'C'D	0	-	0	1	1	1
A'B'C	0	0	1	-	1	-
ACD	1	-	1	1	-	-
BC'D	-	1	0	1	-	1
A'BD	0	1	-	1	-	1
ABCD'	1	1	1	0	-	1

A B C D T T



b)  $F_1(a, b, c, d) = \{1, 12, 3, 2, 14, 15\}$

$$F_2(a, b, c, d) = \{0, 5, 7, 8, 9\}$$

Egor Egorov

F<sub>0</sub> or F<sub>2</sub>

ab	cd	00	01	11	10	11	10	11	10	ab	cd	00	01	11	10
00	0	1	1	1	1	1	1	1	1	00	10	1	4	8	2
01	1	4	5	3	6	7	6	7	6	01	1	15	13	6	5
11	12	13	15	14	14	14	14	14	14	11	12	13	15	14	14
10	8	9	11	10	10	10	10	10	10	10	14	14	14	14	10

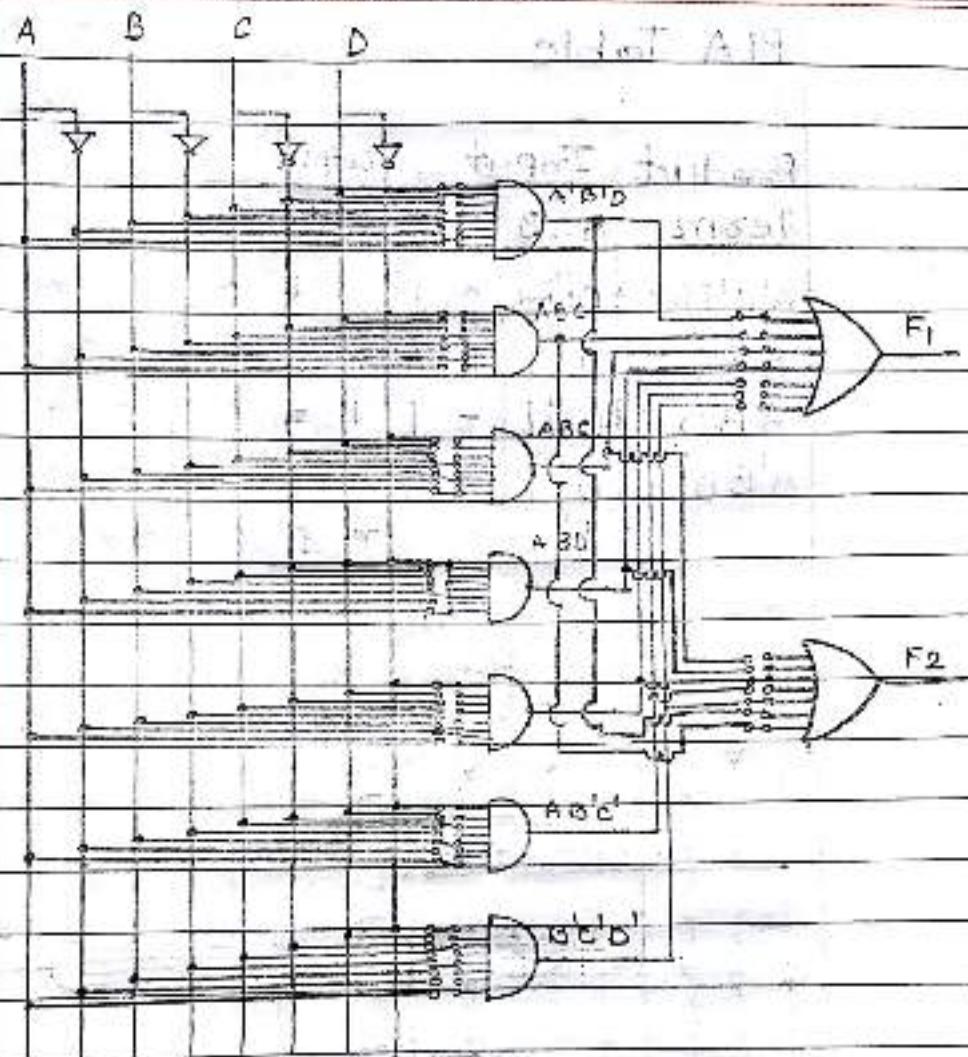
$$F_1 = A'G'D + A'B'C + \dots \quad F_2 = A'B'D + AB'C' + \dots$$

$$ABC + ABD' + C'D = B'C'D'$$

$$= A'B'D + A'B'C + ABC + ABPD'$$

## PLA TABLE

Product	Input				outp.	
Term	A	B	C	D	F <sub>1</sub>	F <sub>2</sub>
A'B'D	0	0	-	1	1	-
A'B'C	0	0	1	-	1	-
ABC	1	1	1	-	1	-
ABD'	1	1	-	0	1	-
A'B'D	0	1	-	1	-	1
AB'C'	1	0	0	-	-	1
B'C'D'	0	0	0	-	1	
					T	T



$$c) F_1(a, b, c) = \Sigma(1, 4, 5, 7)$$

$$F_2(a, b, c) = \Sigma(1, 4, 5, 6)$$

For F<sub>1</sub>

ab	cd	00	01	11	10
00	1	1	0	1	0
01	1	1	1	1	0
11		1	1	1	1
10		0	1	1	0

For F<sub>2</sub>

ab	cd	00	01	11	10
00	1	1	0	1	0
01	1	1	1	1	1
11		1	1	1	1
10		0	1	1	0

$$F_1 = A'C'D + A'BC' + A'BD'$$

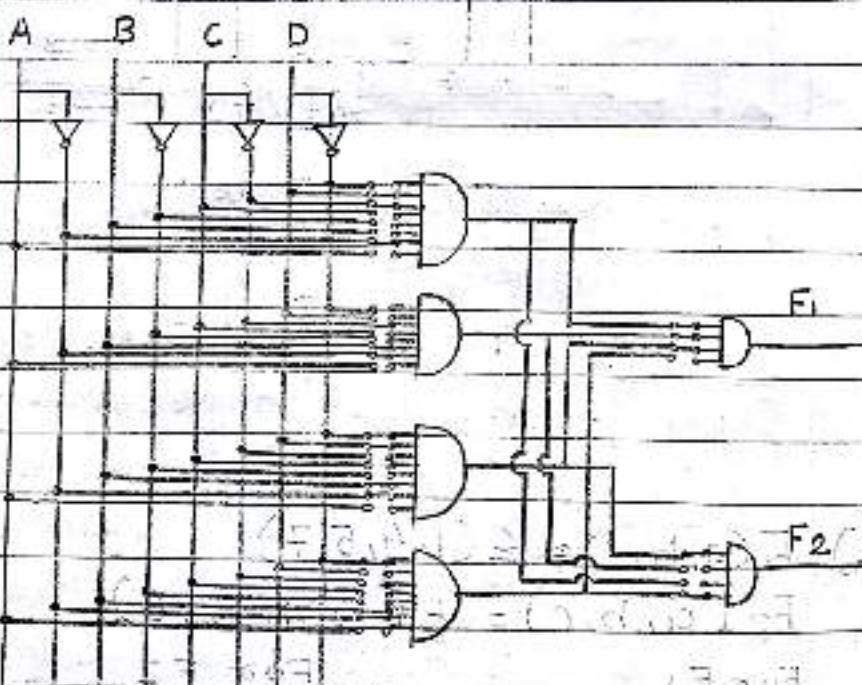
$$A'BD$$

$$F_2 = AC'D + A'BD'$$

## PLA Table

d)

Product Term	A	B	C	D	$F_1$	$F_2$
$A'C'D$	0	-	0	1	1	1
$A'BC'$	0	1	0	-	1	-
$A'BD$	0	1	-	1	1	-
$A'B'D'$	0	1	-	0	-	1
					T	T



d)  $F = \bar{x}y'z' + \bar{x}y'z + \bar{x}yz' + xyz$   
 $F = \Sigma(0, 1, 4, 5)$

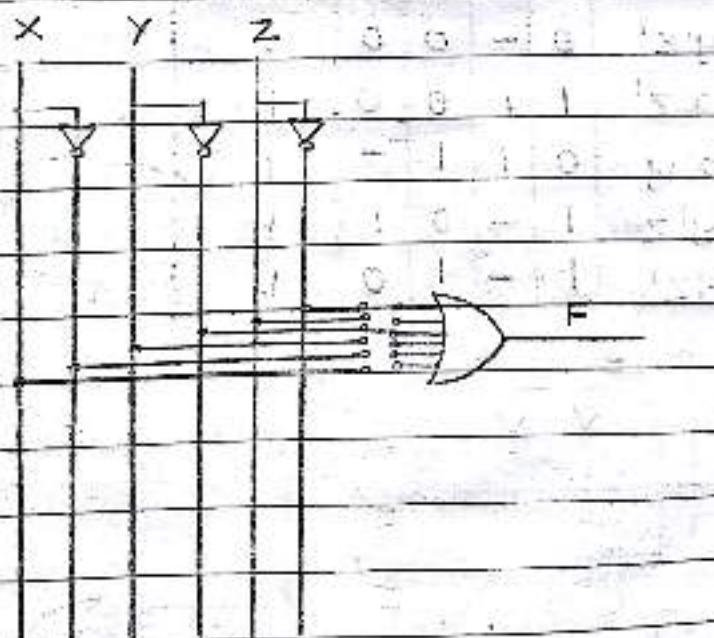
d)

x	yz	00	01	11	10
0	1 0	1		3	2
1	1 4	1 5	7	6	

$$F = Y'$$

PLA Table

Product	Input			Output
Term	x	y	z	F
$Y'$	-	0	-	1



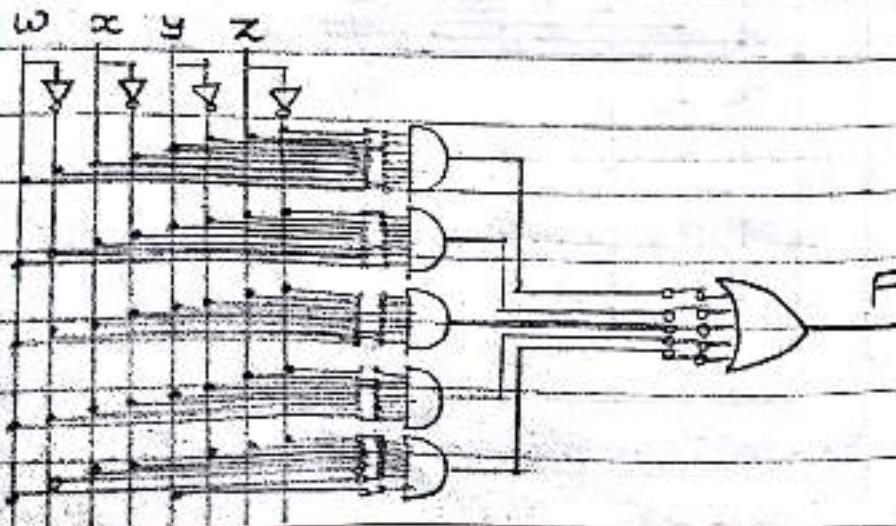
e)  $F(w, x, y, z) = \pi(1, 2, 3, 5, 8, 11, 15)$   
 $= \sum(0, 4, 6, 7, 9, 10, 12, 13, 14)$

wx	yz	00	01	11	10	1
00	π <sub>6</sub>	1	3	2		
01	π <sub>4</sub>	5	7	16		
11	π <sub>2</sub>	13	15	14		
10	π <sub>8</sub>	11	12	10	9	

$$F = w'y'z' + wxz' + w'xy + wy'z + w'yz'$$

PLA Table

Product Term	w	x	y	z	Output F
w'y'z'	0	-	0	0	1
wxz'	1	1	-	0	1
w'xy	0	1	1	-	1
wy'z	1	-	0	1	1
w'yz'	1	-	1	0	1



f) Full Adder  $S = \Sigma (1, 2, 4, 7)$

$$C = \Sigma (3, 5, 6, 7)$$

A \ BC		00	01	11	10
0	0	1	1	1	1
1	1	0	0	0	0
		s	t	g	6

A \ BC		00	01	11	10
0	0	1	1	1	1
1	1	0	0	0	0
		o	o	13	9

$$= AB'C' + ABC + A'B'C + A'BC' = A' + At - A'BC' = B.C + ACT + AB$$

## Assignment - 3

**Q.1** What is Sequential circuit?

→ Sequential logic is a type of logic circuit whose output depends not only on the present value of its input signals but on the sequence of past inputs, the input history. Sequential circuit has memory so output can vary based on input. This type of circuit uses previous input, output, clock and a memory element.

**Q.2** Differentiate Combinational and Sequential circuit.

Combinational Ckt	Sequential Ckt
-------------------	----------------

1. In Combinational Ckt, the output variables at any instant of time are dependent only on the present input variables.

In Sequential Ckt, the output variables at any instant of time are dependent only on the present input variables, but also on the present state, i.e. on the past history of the system.

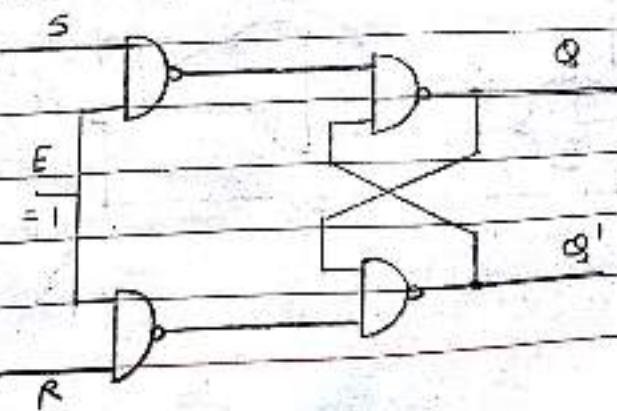
2. Memory unit is not required.

Memory unit is required to store past

- history of input Variable  
Sequential Circuits
3. Combinational CKT are faster are slower than because the delay between I/P & O/P is due to prop.
- delay of gate only Sequential CKT are
4. Combinational CKT comparatively harder are easy to design to design.

Q. 3 Explain SR flip-flop using logic diagram.

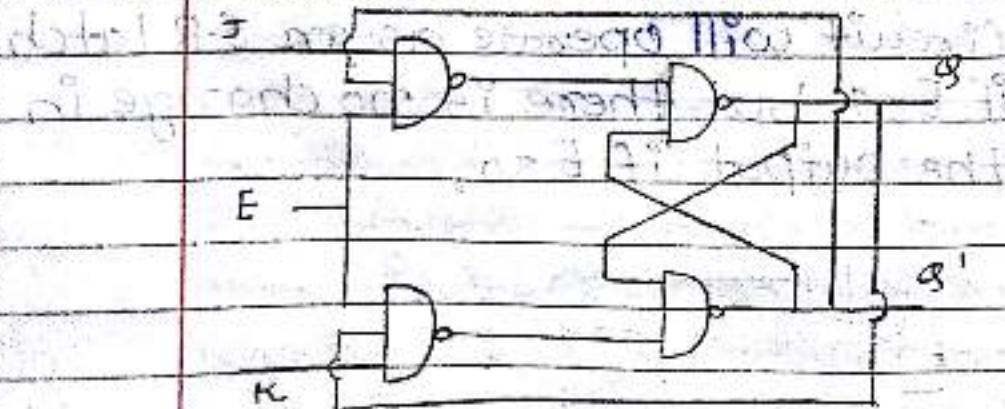
It is basically S-R latch using NAND gates with an additional enable input. It is also called as level triggered SR-FF. For this, circuit in output will take place if and only if the enable input ( $E$ ) is made active. In short this circuit will operate as an S-R Latch if  $E=1$  but there is no change in the output if  $E=0$ .



Q.4 Explain JK flip-flop with logic diagram

Q.5

JK flip flop is the ideal and important memory element which behaves the same fashion as RS flip flop except the condition where R & S equals to 1. It is a forbidden in RS flip flop, the JK flip flop is an improved version which avoids this prohibited or impracticable state and converts it to toggle state i.e. when  $J=1$  &  $K=1$  the output is the inversion of the last state. J and K in the JK flip flop means Jack and Kilby who invented this flip flop combination. This toggling condition is mostly used in the counters.



Q.6

Q.5 Derive characteristic equation for SR flip-flop

S	R	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

S \ R	00	01	11	10
0	0	1	1	0
1	1	X	X	X

$$Q_{n+1} = S + R'Q_n$$

Q.6 Obtain excitation table for SR flip-flop

Step 1. Number of flip-flops

Step 2. State diagram

Step 3. choice of flip-flops  
and excitation table :

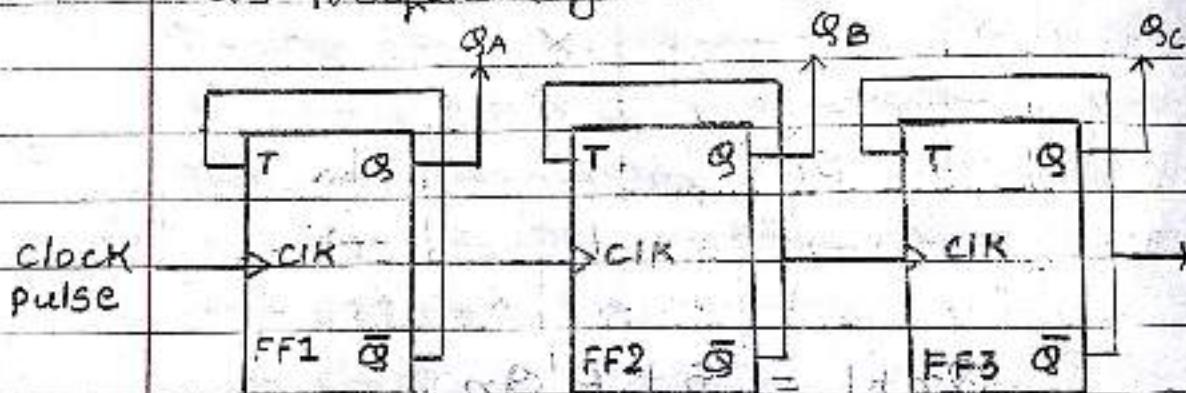
Step 4. minimal expressions  
for excitations

Step 5. logic diagram.

Q. 8

PS	NS	Required I/P	
Qm	Gndl	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

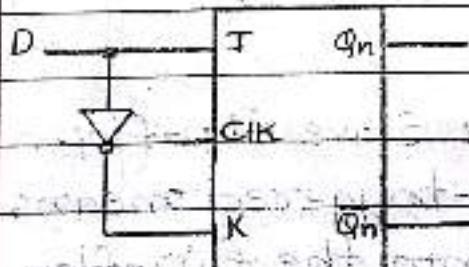
Q. 7 Explain how T flip flop can be used as frequency divider.



One benefit of using toggle flip-flops for frequency division is that the output at any point has an exact 50% duty cycle. The final output clock signal will have a frequency value equal to the input clock frequency divided by the mod number of the counter. Such circuits are known as "divide-by-n" counters.

Q.8 Convert JK flip-flop to D flip-flop.  
 D is the external input and J and K are the actual inputs of the flip-flop. D and  $Q_p$  make four combinations. J and K are expressed in terms of D and  $Q_p$ . The four combination conversion table, the K-maps for J and K in terms of D &  $Q_p$ , and the logic diagram showing the conversion from JK to D are given below.

D	$Q_n$	$Q_{n+1}$	J, K		D	$Q_n$	$Q_{n+1}$	J = D
			0	1				
0	0	0	0	x	1	1	$X_1$	
0	1	0	x	1			$X_2$	
1	0	1	1	x	D	0	$X_3$	
1	1	1	x	0		0	$X_4$	$K = D'$
						1	$X_2$	
							3	



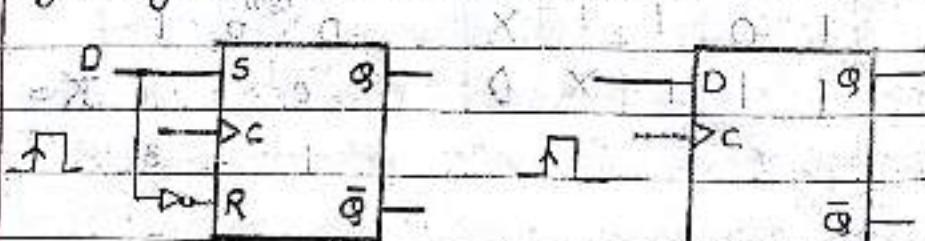
Q.9 Explain working of Edge triggered D flip-flop with logical diagram.

pg 471

(A.K)

The edge-triggered D flip-flop has only one input terminal. The D flip-flop may be obtained from an S-R flip-flop by just putting one inverter between the S and R terminals. This flip-flop has only one synchronous control input in addition to the clock. This is called the D(data) input. The operation of the D flip flop is very simple. The output Q will go to the same state that is present on the D input at the positive-going transition of the clock pulse. In other words, the level present at D will be stored in the flip-flop at the instant the positive-going transition occurs.

Q. 11.



Q. 10 Explain Master Slave flip-flop.

The term pulse-triggered means that data are entered into the flip-flop on the rising edge of the clock pulse, but the output does not reflect the input state until the falling edge of the clock.

pulse. As this kind of flip-flops are sensitive to any change of the input levels during the clock pulse is still HIGH, the inputs must be set up prior to the clock pulse's rising edge & must not be changed before the falling edge. Otherwise, ambiguous results will happen.

The three basic types of pulse-triggered flip-flops are S-R, J-K & D. Their logic symbols are shown below.

Q. 11. What is race-around condition and how to avoid it?

Race is a hold time violation. Flipflop samples its input at a clock edge and passes it on to the output. If the logic cone that generates the value to be sampled changes it too soon (race!) after the sampling edge, this new value may corrupt the correct data that was sampled. Then you have a race. In order to ensure correct sampling, the input data has to be stable during the setup time and hold time

windows. Hold time violations are flagged by static timing analysis and are typically fixed by adding delay in the violating path. More complex fix is needed if the path violates both setup time and hold time.

Q.13

Q.12 List out steps for designing of sequential circuit.

The steps for designing of Sequential Circuit can be expressed as

The first step is to do the state diagram and the second step is to make the excitation table.

The third step is to form the boolean expression for the given inputs before that we have to state the map for the table so the four steps are

Q.14

i) State Diagram (LSD)

ii) Excitation Table

iii) State Map

iv) Boolean expression

v) Implementation

Ans: Standard form and quick logic

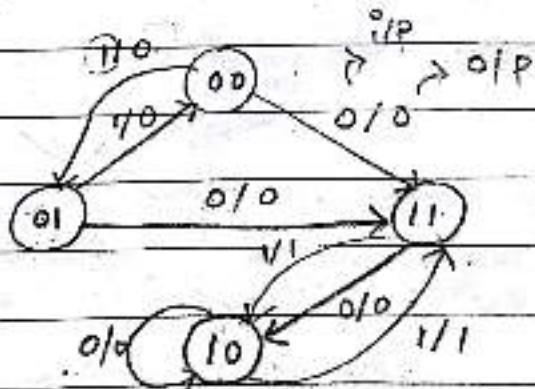
Q.13 Definitions : State diagram, State table, flip-flop.

**State diagram :** The state diagram or state graph is a pictorial representation of the relationships between the present state, the input, the next state, and the output of a sequential circuit.

**State Table :** The state table is a tabular representation of the state diagram.

**Flip-flop :** The most important memory element is the flip-flop, which is made up of an assembly of logic gates.

Q. 14 Design Sequential circuit from foll. state diagram.



P.S.	W.P.	N.S.		W.P.
Q <sub>1</sub> Q <sub>2</sub>	A	Q <sub>1</sub> Q <sub>2</sub>	D <sub>1</sub> D <sub>2</sub>	Z
0 0 0	0	1 1 1	0 0 0	0
1 0 0	1	0 1 0	1 0 1	0
2 0 1 0	1	1 1 1	1 0 0	0
3 0 1 1	0 0 0	0 0 1	0 1 0	1
4 1 0 0	1 0 0	0 0 0	0 0 0	0
5 1 0 1	1 1 1	0 0 0	1 1 1	1
6 1 1 0	1 0	0 0 0	1 0 0	0
7 1 1 1	1 0 1 0	1 0 1 0	1 1 1 1	Q.2

D<sub>1</sub>

$$\begin{array}{l} \cancel{A} Q_1 \\ Q_1 \quad 0 0 0 1 1 1 1 0 \\ \hline 0 \boxed{D_1} \quad 1 \quad 3 \boxed{1} 2 \end{array}$$

$$= Q_1 Q_1$$

$$1 \quad 4 \quad 5 \quad 7 \quad 6 \quad = 1$$

## Assignment : 5

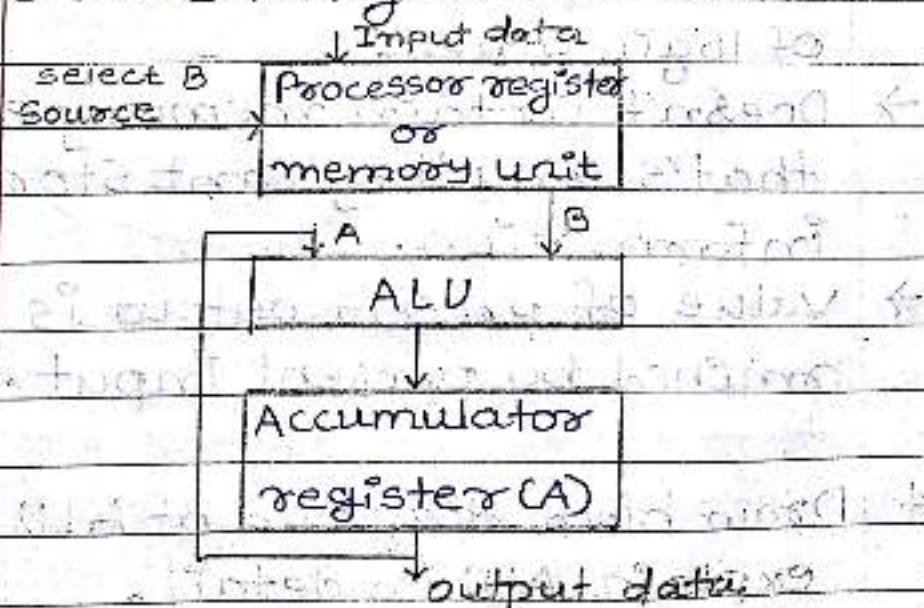
Q.1 What is Arithmetic and Logical Unit?

- The Arithmetic Logic Unit (ALU) is a widely used combinational circuit, which is capable of performing arithmetic as well as logical operations.

Q.2 Explain Accumulator briefly.

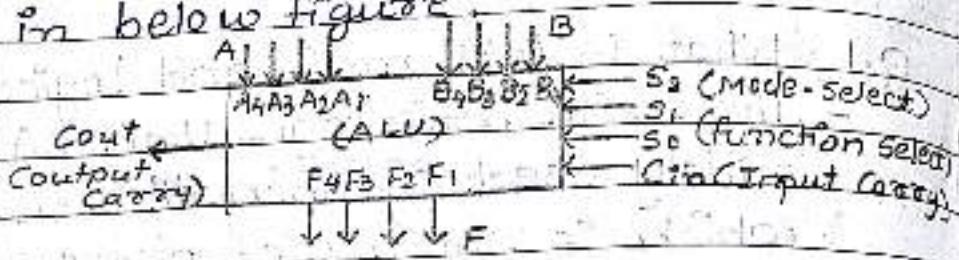
- Some processor units distinguish one register from all others and call it an accumulator register.

The Organization of a processor unit with an accumulator register is shown in figure.



- The ALU associated with the register may be constructed as a combinational

onal circuit of the type discussed in below figure.



Q.3 Which is the fundamental Combinational circuit in Arithmetic unit?

- The fundamental combinational circuits used in arithmetic unit are half adders, full adders, Half subtractor, full subtractor, multiplexer, Demultiplexer, encoder, decoder
- Combinational Circuits are made of logic gates.

→ Doesn't contain memory element, that's why they can't store any information.

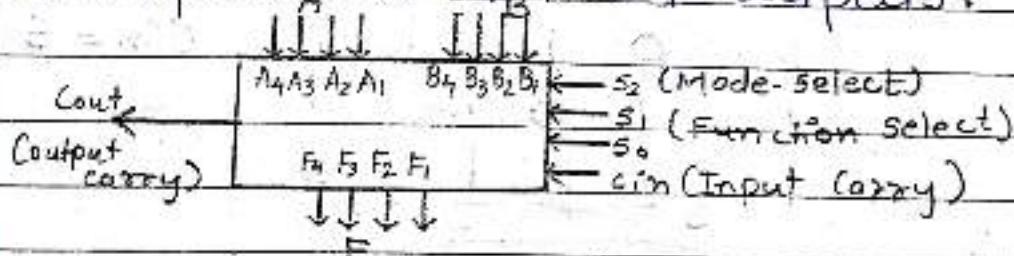
→ value of present output is determined by present input.

Q.4 Draw block diagram of ALU and explain ALU in details.

An arithmetic logic unit (ALU) is a multi operation, combinational

logic digital function. It can perform a set of base arithmetic operations and a set of logic operations.

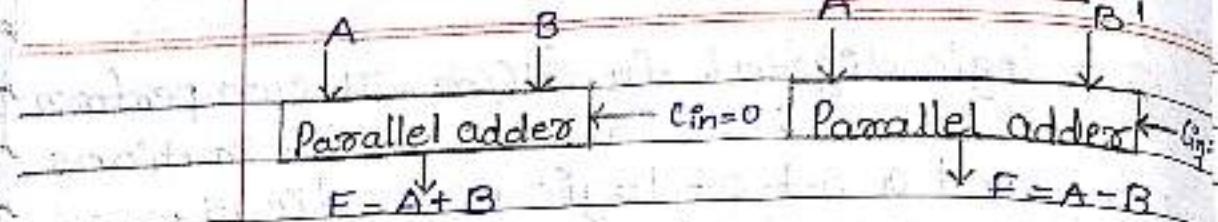
- The ALU has a number of Selection lines to select a particular operation in the unit. The Selection lines are decoded within the ALU so that  $K$  selection variables can specify up to  $2^K$  distinct operations.
- Figure shows the block diagram of a 4-bit ALU. The four data inputs from A are combined with the four inputs from B to generate an operation at the F outputs.



- Q.5 Design an adder/subtractor circuit with one selection variable  $S$  & 2 inputs A and B. When  $S=0$  the circuit performs the addition operation  $F=A+B$ . When  $S=1$  the circuit performs subtraction  $F=A-B$  by taking 2's complement of B.

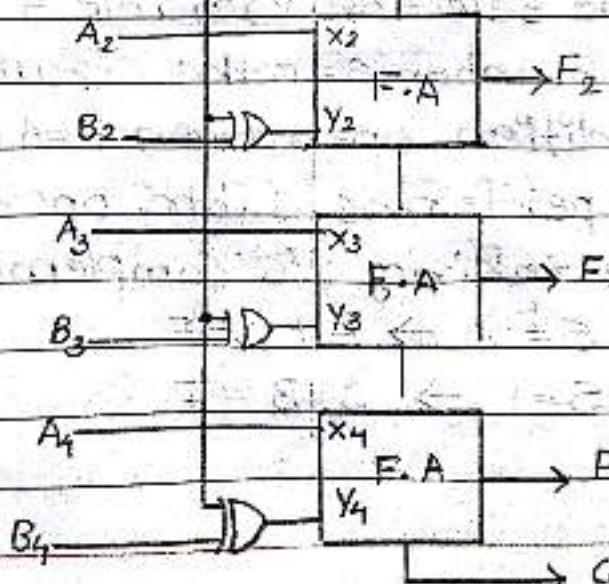
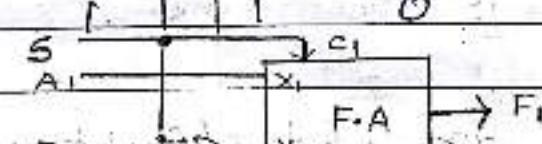
$$\text{If } S=0 \rightarrow A+B=F$$

$$S=1 \rightarrow A-B=F$$



S	X	Y	$C_{in}$	$S=0 \Rightarrow X=A$
0	A	B	0	$S=1 \Rightarrow X=A$
1	A	$B^1$	1	$S=0 \Rightarrow Y=B$ $S=1 \Rightarrow Y=B^1$

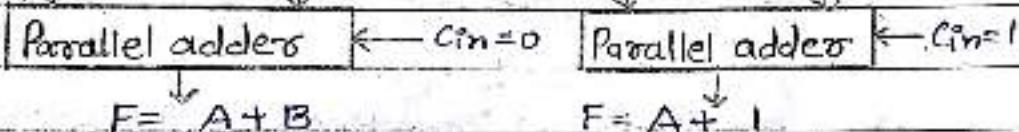
S	A	B	X	Y	$X = A$
0	0	0	0	0	$y = S'A'B + S'AB +$
0	0	1	0	1	$S'A'B^1 + SAB^1$
0	1	0	1	0	$= S'B(A^1 + A) + SB^1$
0	1	1	1	1	$(A^1 + A)$
1	0	0	0	1	$= S \oplus B$
1	0	1	0	0	$C_{in} = S$
1	1	0	1	1	
1	1	1	1	0	



Q.6 Design an arithmetic ckt with one selection variable  $S$  and two inputs  $A$  &  $B$ . When  $S=0$  the ckt performs the addition operation  $F=A+B$ . When  $S=1$  the ckt performs the increment operation  $F=A+1$ .

$$\text{If } S=0 \rightarrow F = A+B$$

$$\begin{array}{ccc} S=1 & \rightarrow & F = A+1 \\ \downarrow & & \downarrow \\ A & B & A \end{array}$$



$$\begin{array}{|c|c|c|c|c|} \hline S & X & Y & C_{in} & S=0 \rightarrow x=A \\ \hline 0 & A & B & 0 & \\ \hline 1 & A & 0 & 1 & \\ \hline \end{array}$$

$$S=1 \rightarrow x=A$$

$$S=0 \rightarrow y=B$$

$$S=1 \rightarrow y=0$$

$$\begin{array}{|c|c|c|c|c|c|} \hline S & A & B & X & Y & \\ \hline 0 & 0 & 0 & 0 & 0 & x=A \\ \hline 0 & 0 & 1 & 0 & 1 & y=S'A'B + S'AB \\ \hline 0 & 1 & 0 & 1 & 0 & = S'B \\ \hline 0 & 1 & 1 & 1 & 1 & \\ \hline 1 & 0 & 0 & 0 & 0 & \\ \hline 1 & 0 & 1 & 0 & 1 & \\ \hline 1 & 1 & 0 & 1 & 0 & \\ \hline 1 & 1 & 1 & 1 & 1 & \\ \hline \end{array}$$

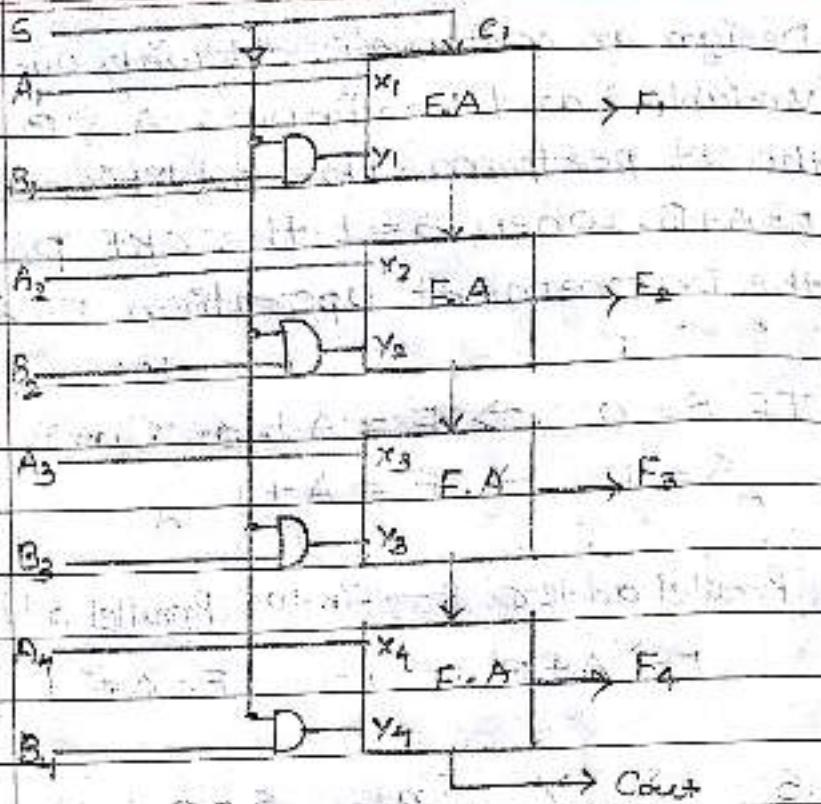
$$x=A$$

$$y=S'A'B + S'AB$$

$$= S'B$$

$$\begin{array}{l} S'AB + S'AB = 0 \\ S'AB + A = 1 \end{array}$$

$$\begin{array}{l} S'AB + A = 1 \\ S'AB = 1 \\ AB = 1 \end{array}$$



Q. 8

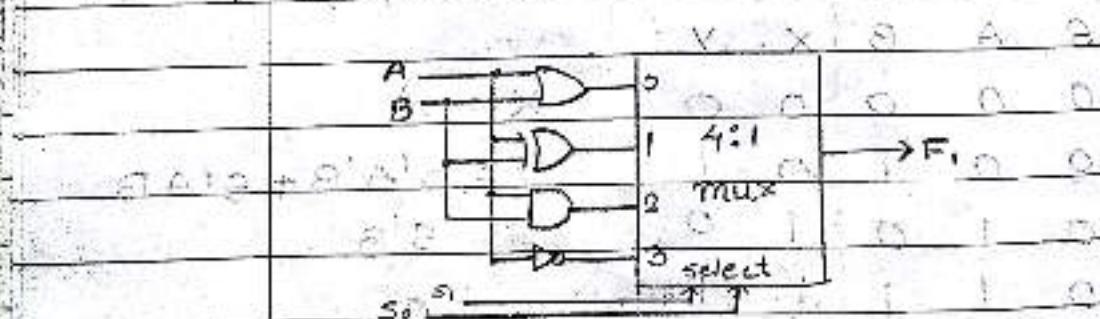
i)

ii)

iii)

iv)

Q. 7 Design logical unit perform a) AND b) OR  
c) EX-OR d) NOT operation on number  
A and B.

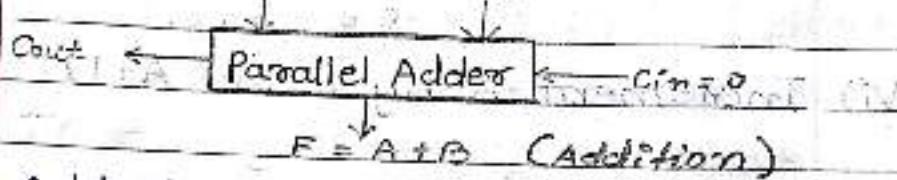


Q. 9

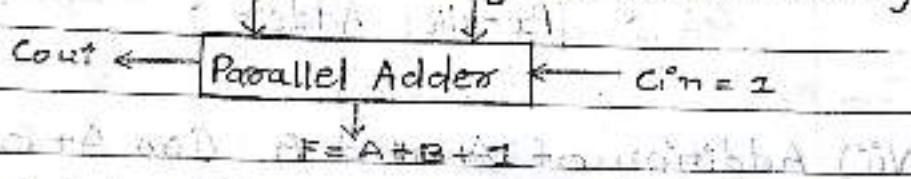
$S_1$	$S_0$	output	operation
0	0	$F = A + B$	OR
0	1	$F = A \oplus B$	XOR
1	0	$F = A \cdot B$	AND
1	1	$F = A'$	NOT

Q. 8 Design Arithmetical unit which perform foll. operations on two bits no. A & B.

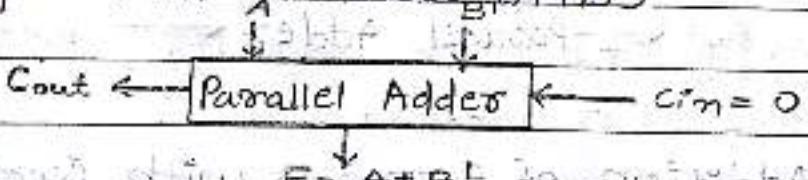
i) Addition of A and B ( $A+B$ )



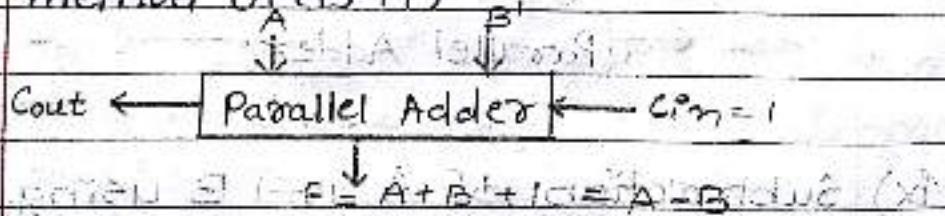
ii) Addition of A and B with carry ( $A+B+1$ )



iii) Subtraction of A and B with 2's complement method ( $A+B'$ )



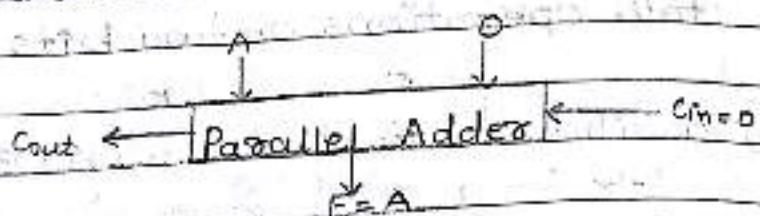
iv) Subtraction of A & B using 2's complement method ( $A+B'+1$ )



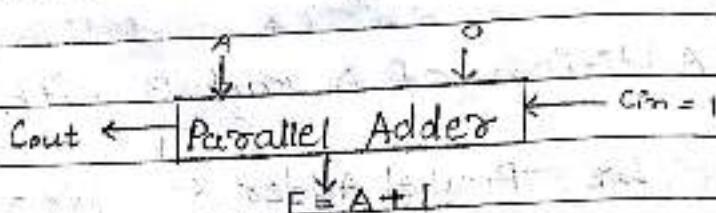
Q. 9 Design Arithmetical unit which perform foll. eight operations on two numbers A and B of two bits.

The operations are :

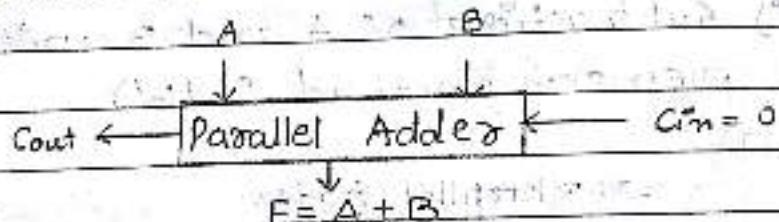
v) Transfer of A (or A)



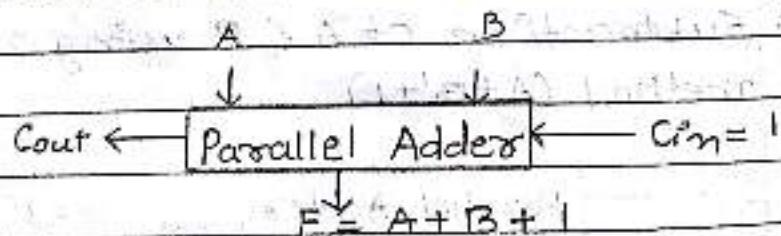
vi) Incrementing A by 1 (or A+1)



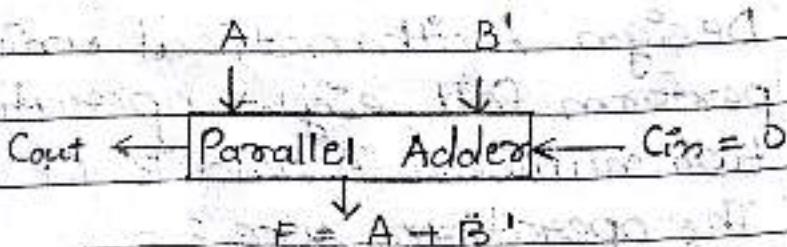
vii) Addition of A and B (or A+B)



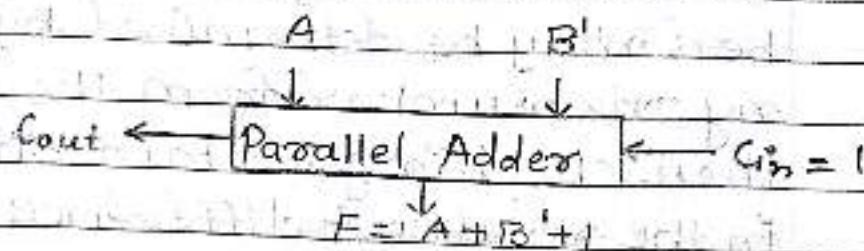
viii) Addition of A and B with carry



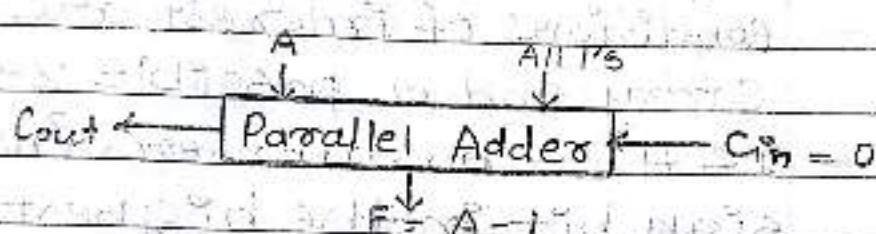
ix) Subtraction of A and B using 1's complement method (or A+B')



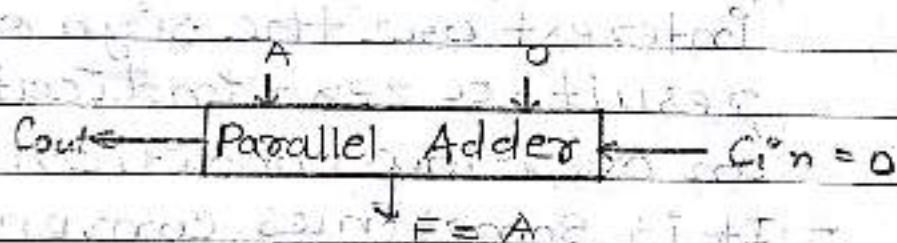
x) Subtraction of A and B using 2's complement method (or  $A+B'+1$ )



xii) Decrement A by 1 (or  $A-1$ )



xiii) Transfer of A (or A)



Q.10 Design an arithmetic ckt with two selection variables,  $S_1$  and  $S_0$ , that generates the following arithmetic operations. Draw the logic diagram of one typical stage.

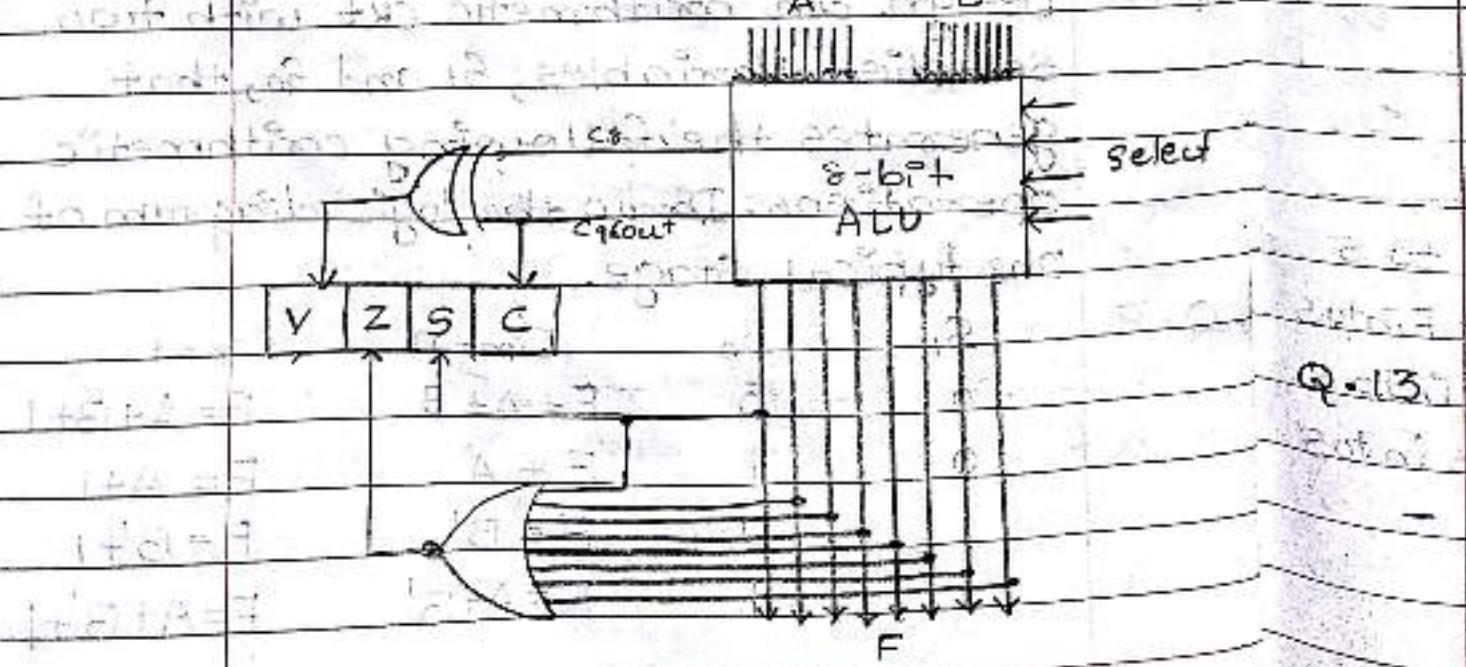
$S_1$	$S_0$	$C_{in}=0$	$C_{in}=1$
0	0	$F = A + B$	$F = A + B + 1$
0	1	$F = A$	$F = A + 1$
1	0	$F = B'$	$F = B' + 1$
1	1	$F = A + B'$	$F = A + B' + 1$

Q.11 Explain status register in detail.

- The relative magnitude of two numbers may be determined by subtracting one number from the other & then checking certain bit conditions in the resultant difference. If the two numbers are unsigned, the bit conditions of interest are the output carry and a possible zero result. If the two numbers include a sign bit in the highest order position, the bit condition of interest are the sign of the result, a zero indication, and an overflow condition.

- It is sometimes convenient to supplement the ALU

with flags to indicate A = B or A > B



Q.12 Find C, S, Z and V

1.  $A = 11101000, B = 10100100$

$\begin{array}{r} & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ - & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \hline & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}$  C=1

$\begin{array}{r} & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ - & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$  Z=0

$\begin{array}{r} & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ - & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}$  S=1

$\begin{array}{r} & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ - & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ \hline & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{array}$  V=0

C=1 F6 C=0

2.  $A = 11111111, B = 00000001$

$\begin{array}{r} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array}$  C=1

$\begin{array}{r} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{array}$  Z=1

$\begin{array}{r} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{array}$  S=0

$\begin{array}{r} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{array}$  V=0

3.  $A = 10000010, B = 11000000$

$\begin{array}{r} & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ - & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$  C=1

$\begin{array}{r} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ - & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$  Z=0

$\begin{array}{r} & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ - & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$  S=0

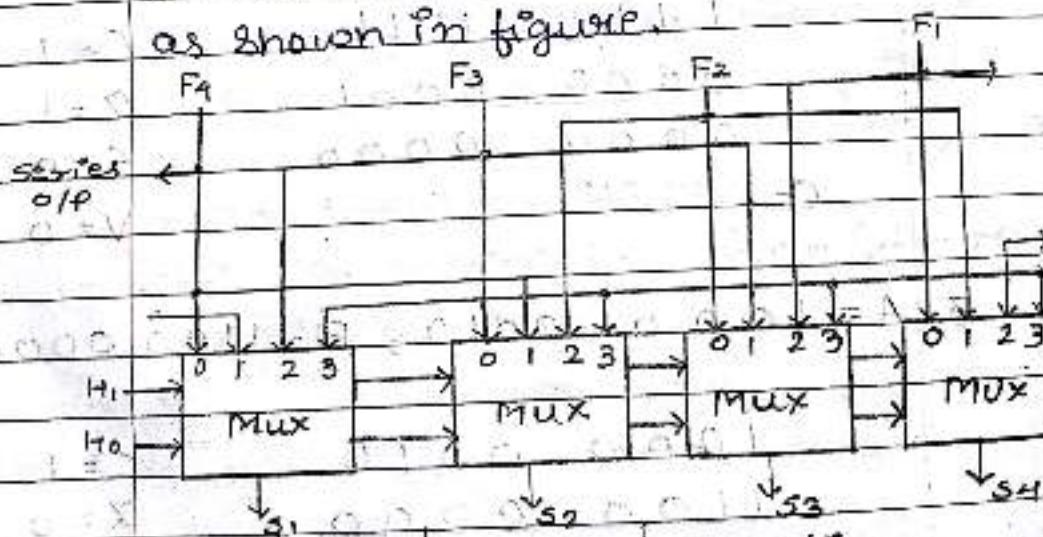
$\begin{array}{r} & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ - & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}$  V=0

Q.13 Draw and explain logic diagram of Shifter.

- The shift unit attached to a process or transfer the output of the ALU

onto the output bus. The shifter may transfer the information directly without a shift or it may shift the information to the right or left. Provision is sometimes made for no transfer from the ALU to the output. Thus shifter provides the shift micro operators commonly not available in an ALU.

- A combinational logic shifter can be constructed with multiplexers as shown in figure.



H<sub>1</sub>    H<sub>0</sub>    Operation    Function

0	0	$S \leftarrow F$	Transfer F to S
0	1	$S \leftarrow \text{sh}r F$	Shift right F into S
1	0	$S \leftarrow \text{sh}l F$	Shift left F into S
1	1	$S \leftarrow D$	Transfer 0's into S

Q.14 Draw the unit and unit in The Select unit connected with any given variable the shifter.

means variable in a sequence Selection

the choice typical

A block is Sh

Seven

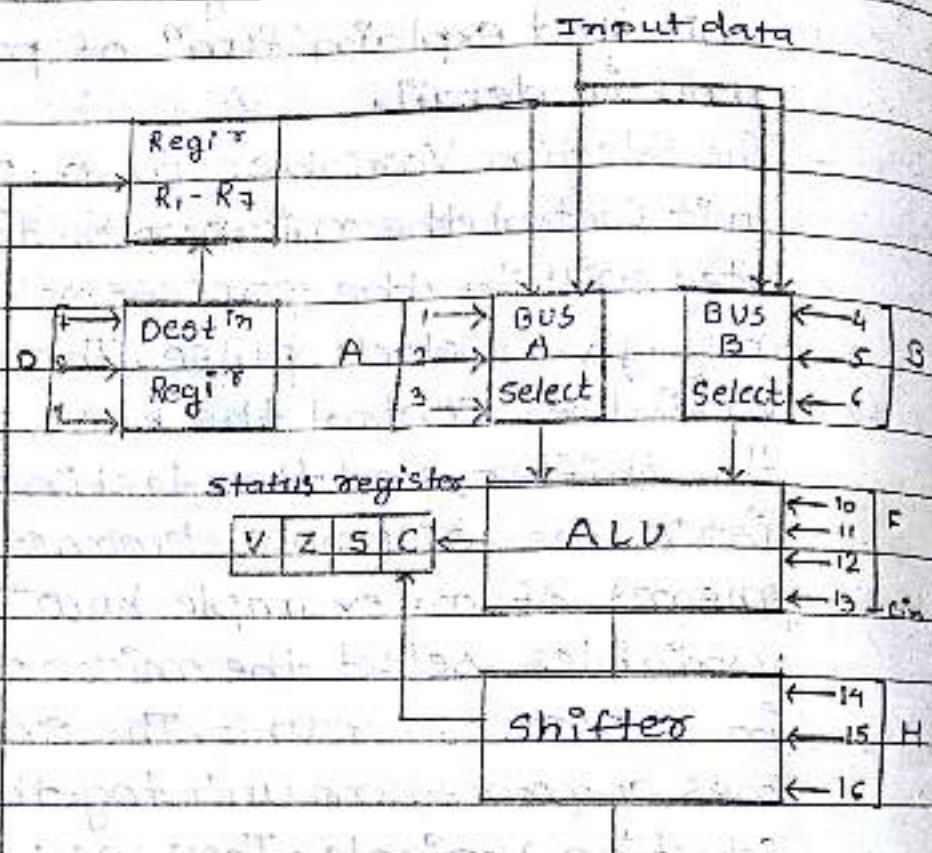
Seven

plex

Input are mu

Q.14. Draw the block diagram of processor unit and explain fun<sup>n</sup> of processor unit in detail.

- The Selection Variables in a processor unit control the microoperations executed within the processor during any given clock pulse. The Selection Variables control the buses, the ALU, the shifter, and the destination register. We will now demonstrate by means of an example how the control variables select the microoperations in a processor unit. The example defines a processor unit together with all Selection Variable. Then we will discuss the choice of control variables for some typical microoperations.
- A block diagram of a processor unit is shown in figure. It consists of seven registers  $R_1$  through  $R_7$  and a status register. The output of the seven registers go through the multiplexers to select the input to the ALU. Input data from an external source are also selected by the same multiplexer.



a) Block diagram

→ output data

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

A B D F G<sub>n</sub> H.

### b) Control word

**Q.15** Find the Control word for the following operations:

$R1 \leftarrow R1 - R2$

A ← A ∪ B

~~001 001 010 010 000~~

Cal : 0010100010100000  
          2   8   A   0  
= 28AOH o/p data

2.  $R_5 \leftarrow R_3 + R_4$

D  $\leftarrow A + B$

101 011 100 + 011 100 = 011 111

Cal : 0111001010010000  
          7   4   2   9   0  
= 729OH o/p data

3.  $R_7 \leftarrow R_5 \text{ AND } R_6$

D  $\leftarrow A \text{ AND } B$

Cal : 1011101111000000 (A)  
          B   B   C   0

= BBCOH o/p data

the output to maximum value with B-C  
register write of below

BC = 01010101 (D)

BC = 00100000 (D)

= 00100000 = 16H (D)

BC = 00000000 (D)

# Assignment - 4

Q. 1 Define

- Register

The group of flip-flop is register.  
Stack of flip-flop use to store  
binary data.

- flip-flop - It can be store 2-bit.

- Counter

It use to count pulses

Q. 2 List out applications of Shift regis.  
too.

(1) Time delays

(2) Serial or parallel data conversion

(3) Ring Counters

(4) Universal asynchronous receiver  
transmitter.

Q. 3 How many number of flip-flop  
needed to store number

$$(104)_{10} = (1101000)_2$$

$$n \text{ bit} = 2^n \text{ mod}$$

$$(1101000)_2$$

$$\therefore 7 \text{ bit} = n = n \cdot \text{no of FF}$$

$$\therefore 7 \text{ FF}$$

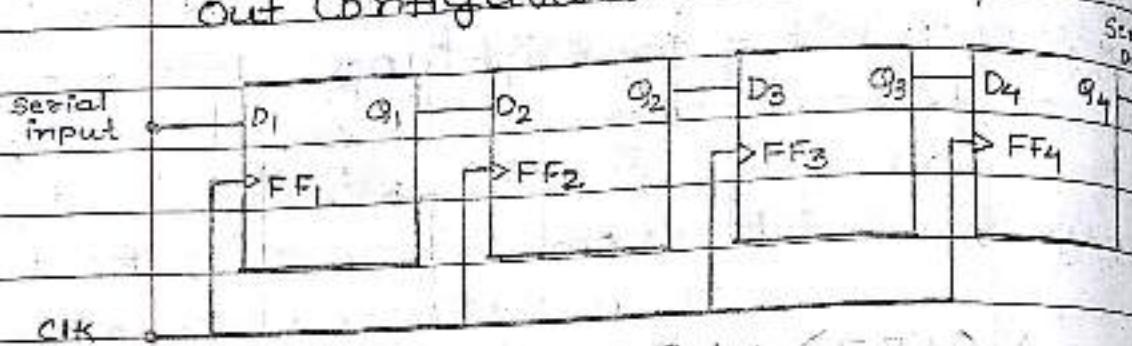
$$\begin{aligned}
 \text{(b)} \quad (ACF=2)_8 &= [ \underline{0} \underline{1} \underline{0} \underline{1} \underline{0} \underline{0} \underline{1} \underline{1} \underline{1} \underline{0} \underline{0} \underline{1} \underline{0} ], \\
 &= n \text{ bit} = 2^n \text{ Mod} \\
 &= (101011001110010)_2 \\
 &= 16 \text{ bit} \\
 &n \text{ bit} = n \text{ Flip-Flop} \\
 &\therefore 16 \text{ F.F}
 \end{aligned}$$

$$\begin{aligned}
 \text{(c)} \quad (457)_8 &= (10010111)_2 \\
 &= n \text{ bit} = 2^n \text{ Mod} \\
 &= (10010111)_2 \\
 &= 9 \text{ bit} \\
 &n \text{ bit} = n \text{ Flip-Flop} \\
 &\therefore 9 \text{ F.F}
 \end{aligned}$$

$$\begin{aligned}
 \text{(d)} \quad (1010111011)_2 & \\
 &= n \text{ bit} = 2^n \text{ Mod} \\
 &= (1010111011)_2 \\
 &= 10 \text{ bit} \\
 &n \text{ bit} = n \text{ Flip-Flop} \\
 &\therefore 10 \text{ F.F}
 \end{aligned}$$

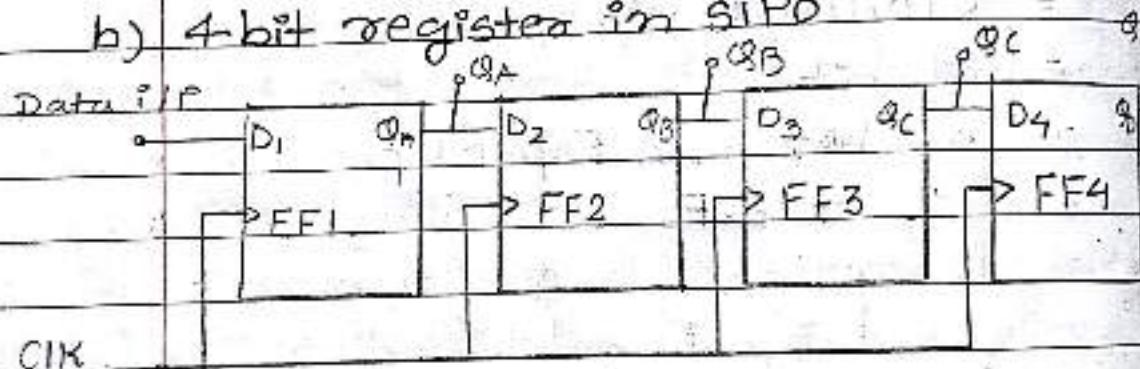
Q. 5 Draw logical diagram of

a) 4 bit register in Serial in Serial out Configuration (SISO)

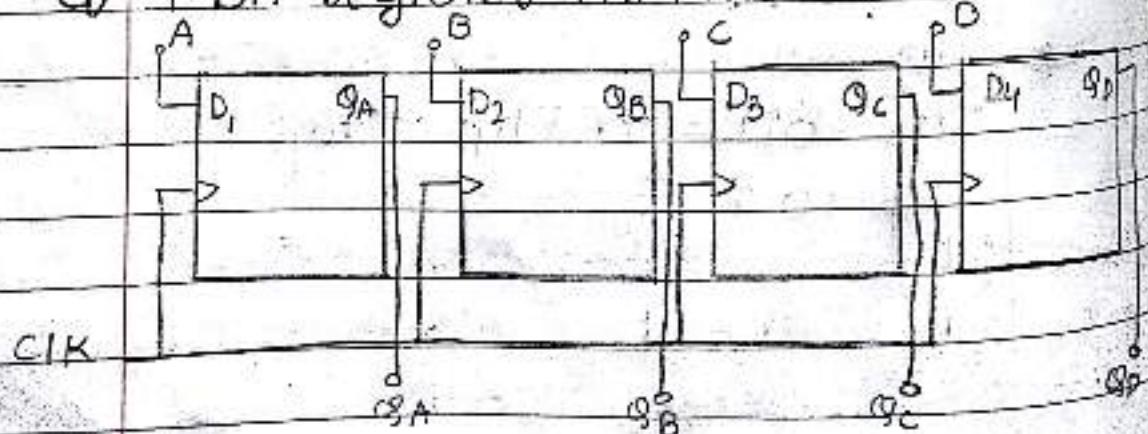


shift right (S = 1)

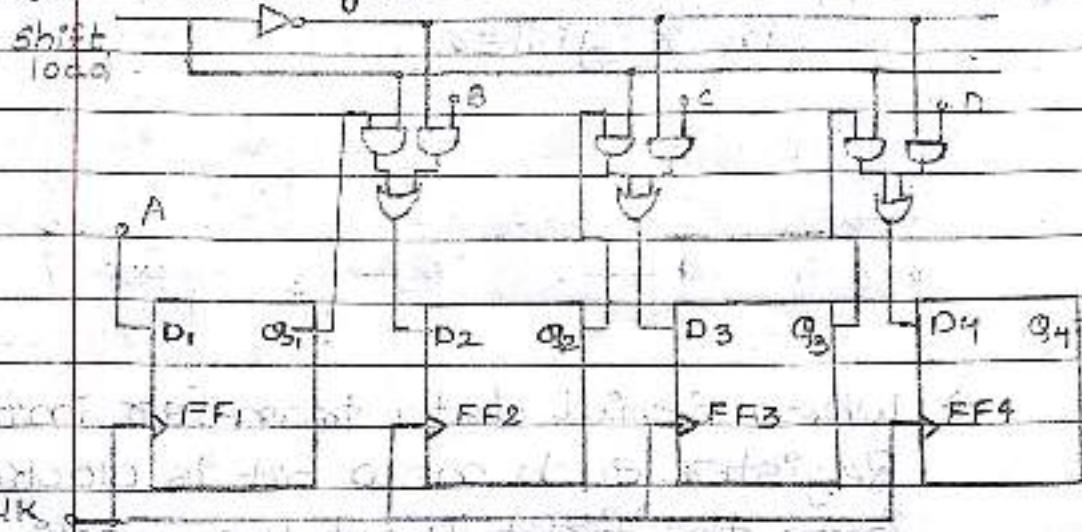
b) 4-bit register in SIPO



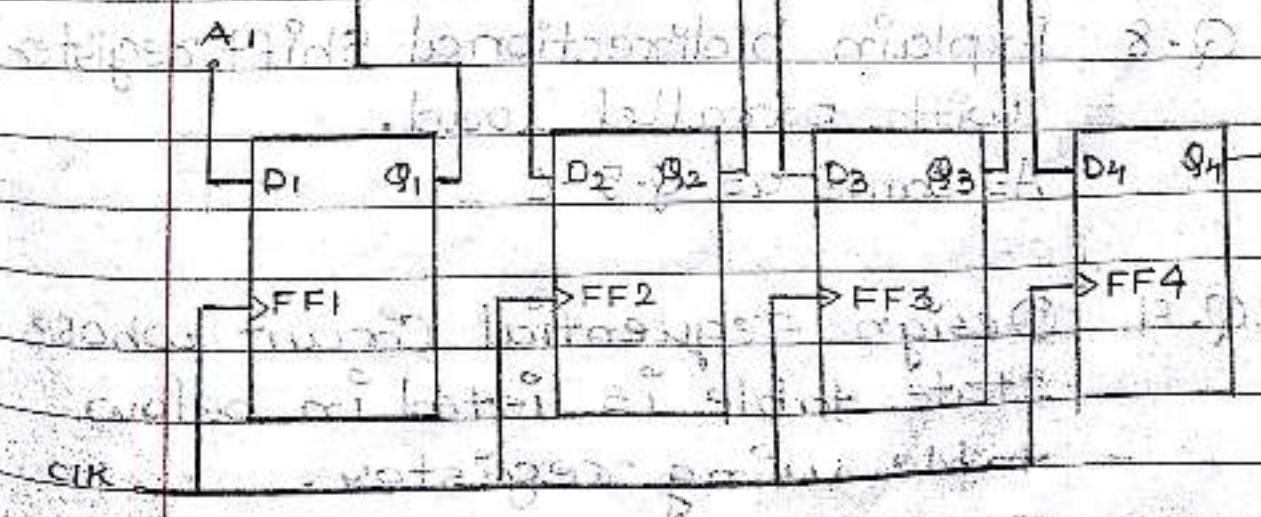
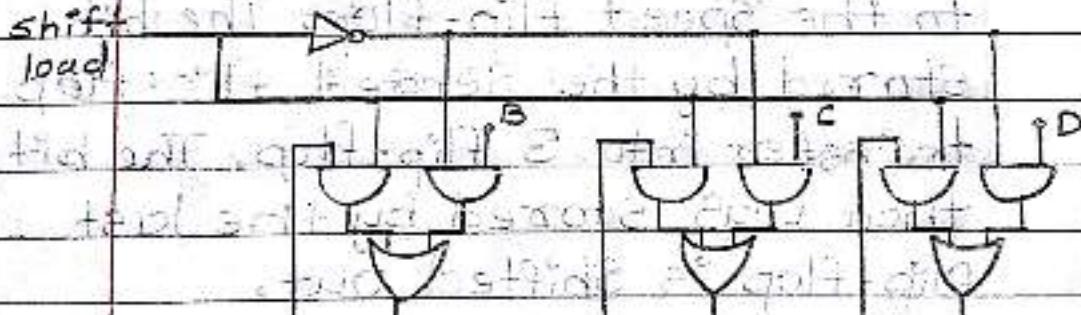
c) 4 bit register in PIPD



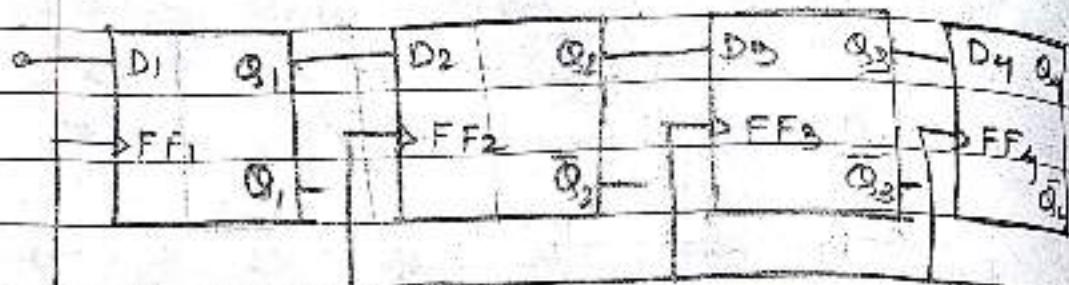
c) 4 bit register in PISO



Q.6 Explain 4 bit register with parallel load using SR flip-flop



Q-7 Explain Serial transfer operation in register.



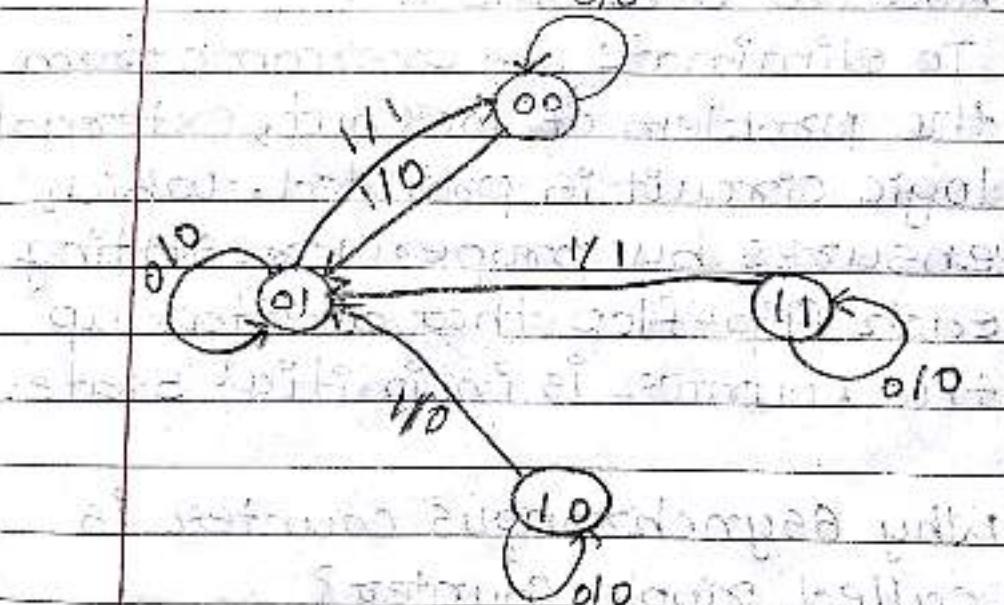
- When Serial data transfer into Register each new bit is clocked into the first FF at the positive going edge of each clock pulse.
- The bit was previously stored by the first flip-flop is transferred to the second flip-flop. The bit stored by the second flip-flop transfers into 3 flip-flop. The bit that was stored by the last flip-flop is Shifted out.

Q-8 Explain bidirectional Shift register with parallel load.

As same as Q-5-C

Q-9 Design Sequential Circuit whose state table is listed in below table using register.

Q.	P.S.		I/P	N.S.		O/P
	A <sub>1</sub>	A <sub>2</sub>	M	A' <sub>1</sub>	A' <sub>2</sub>	M'
	0	0	0	0	0	0
	0	0	1	0	1	0
	0	1	0	0	1	0
	0	1	1	0	0	1
	1	0	0	1	0	0
	1	0	1	0	1	0
	1	1	0	1	1	0
	1	1	1	0	1	1



Q.10 What is lock-out condition in counter?  
How to overcome it?

- Sometime when the Counter is on because of noise spikes the Counter may find it self in some unused state.
- The Counter whose unused states have this feature is said to suffer from the problem of lock out.

→ How to overcome it  
To eliminate or overcome from the problem of lock out, external logic circuit is provided which ensures by properly resetting each flip-flop that at start up the counter is in initial state.

Q.11 Why Asynchronous counter is called ripple counter?

- It called Ripple Counter because in their counter the transition of the first stage ripples through the last stage.

Q.12 How many flip-flops needed for twisted-ring (Johnson) counter having 8 different states?

(ii)

→ For 8 different state 4 flip-flops are needed for twisted-ring (Johnson) counter as in A n FF Johnson counter can have  $2^n$  unique states and can count up to  $2^n$  pulses. So, For 8 different states (i.e  $2 \times 4 = 2 \times n = 8$ ) we need ( $n=4$ ) 4 FF.

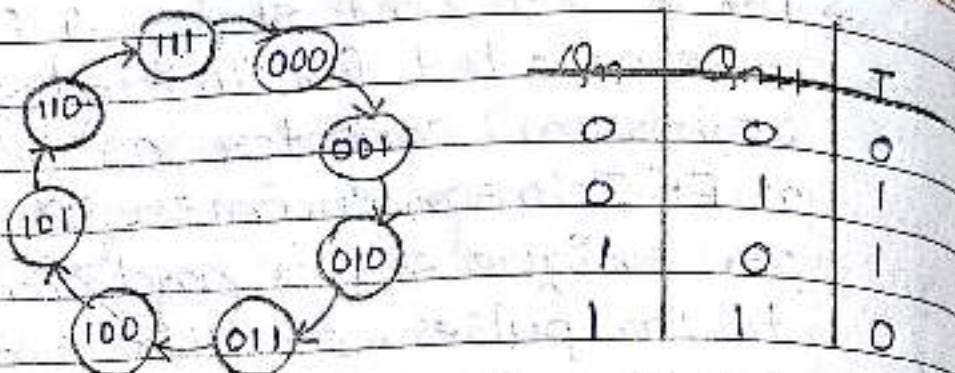
**Q.13** How many flip-flops needed for ring counter having 8 different states?

→ For 8 different state 8 flip-flops are needed for ring counter as in A n ring counter FF can have  $n$  unique states and can count up to  $n$  pulses. So, for 8 different state (i.e  $8 = n$ ) we need ( $n=8$ ) 8

Flip Flop

**Q.14** Design 3-bit synchronous Up counter using T flip-flop:

→ 3-bit up counter required 3-FF and  
 (i) Counting Sequence is 000, 001, 010,  
 011, 100, 101, 110, 111, 000, -- like this  
 (ii) state diagram



Q. 15

→

(ii)

(iii)

iii) PS NS Required excitations

Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>	T <sub>3</sub> T <sub>2</sub> T <sub>1</sub>
0 0 0	0 0 1	0 0 1
0 0 1	0 1 0	0 1 1
0 1 0	0 1 1	0 0 1
0 1 1	1 0 0	1 1 1
1 0 0	1 0 1	0 0 1
1 0 1	1 1 0	0 1 1
1 1 0	1 1 1	0 0 1
1 1 1	0 0 0	1 1 1

Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>
0 0 0 0 1 1 1 0	0 0 0 0 1 1 1 0
0 0 1 1 1 1 2	0 0 1 1 1 1 2
1 1 4 5 6	1 1 4 5 6

$$T_3 = Q_2 Q_1 \quad T_2 = Q_1$$

Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>
0 0 0 0 1 1 1 0	0 0 0 0 1 1 1 0
0 1 1 1 1 2	0 1 1 1 1 2
1 1 4 5 6	1 1 4 5 6

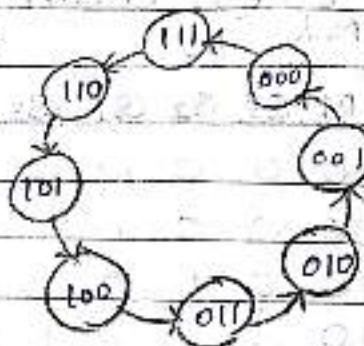
$$T_1 = 1$$

extra

T  
O  
I  
O  
Q. 15 Design 3-bit synchronous down counter  
Using T flip-flop.

→ 3 bit down counter required 3-FF and

- (i) Counting sequence is 000, 111, 110, 101,  
100, 011, 010, 001, 000, and like this
- (ii) State Diagram



iii)

PS	NS	Required exc
$Q_3\ Q_2\ Q_1$	$Q_3\ Q_2\ Q_1$	$T_3\ T_2\ T_1$
0 0 0	1 1 1	1 1 1
1 1 1	1 1 0	0 0 1
1 1 0	1 0 1	0 1 1
1 0 1	1 0 0	0 0 1
1 0 0	0 1 1	1 1 1
0 1 1	0 1 0	0 0 1
0 1 0	0 0 1	0 1 1
0 0 1	0 0 0	0 0 1

extra

$Q_2\ Q_1$		00 01 11 10			
0	1	1	0	1	0
1	0	1	1	0	1

$Q_2\ Q_1$		00 01 11 10			
0	1	1	0	1	0
1	0	1	1	0	1

$$T_3 = Q_2' Q_1'$$

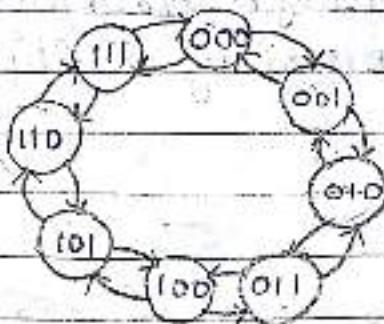
$$T_2 = Q_1'$$

$Q_3$	$Q_2$	$Q_1$	11	10
0	1	0	11	13
1	1	1	14	15

Q.16 Design 3-bit Synchronous Up/Down counter using T flip-flop

Mode PS NS Required excitation

M	$Q_3$	$Q_2$	$Q_1$	$Q_3$	$Q_2$	$Q_1$	T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>
0	0	0	0	0	0	1	0	0	1
0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	1	1	0	0	1
0	0	1	1	1	0	0	1	1	1
0	1	0	0	1	0	1	0	0	1
0	1	0	1	1	1	0	0	1	1
0	1	1	0	1	0	0	1	1	1
1	0	0	0	1	0	1	1	0	1
1	0	0	1	0	0	0	0	1	1
1	0	1	1	0	0	1	0	0	1
1	1	0	0	0	1	0	1	0	1
1	1	0	1	1	0	0	0	0	1
1	1	1	0	1	0	1	0	1	1
1	1	1	1	1	1	0	0	0	1



$M = 0 \rightarrow$  Up Counter

$M = 1 \rightarrow$  Down Counter

UP/  
Down  
Counter

		$Q_2 Q_1$	00	01	11	10
$Q_3$	00	0	1	13	2	
01	4	5	17	6		
11	11	12	13	15	14	
10	16	9	11	10		

$$T_2 = Q_3' Q_2 Q_1 + Q_3 Q_2' Q_1'$$

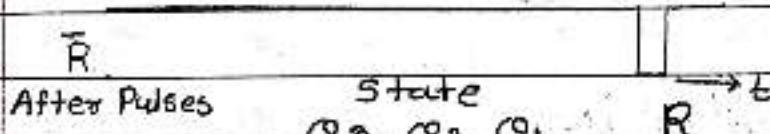
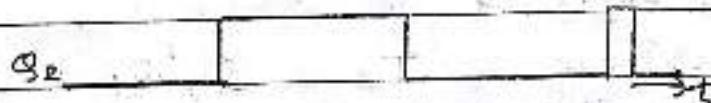
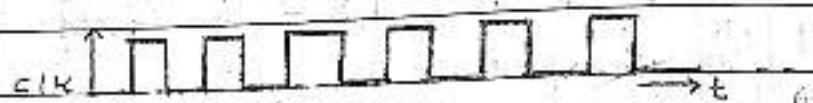
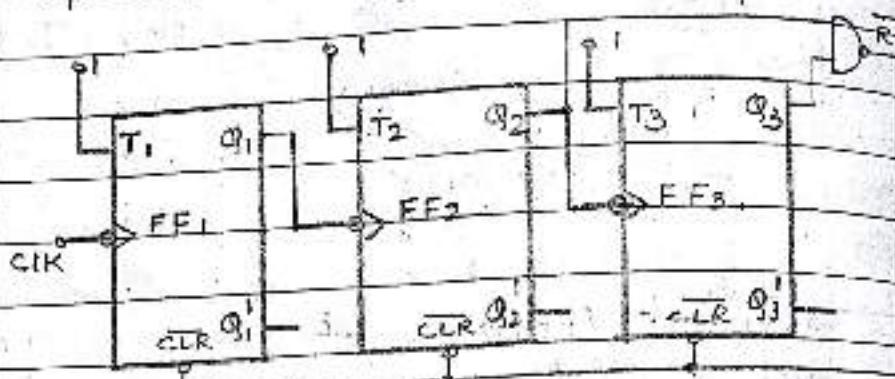
		$Q_2 Q_1$	00	01	11	10
$Q_3$	00	0	1	13	2	
01	4	5	17	6		
11	11	12	13	15	14	
10	16	9	11	10		

$$T_3 = Q_3' Q_1 + Q_3 Q_1' \\ = Q_3 \oplus Q_1$$

		$Q_2 Q_1$	00	01	11	10
$Q_3$	00	0	1	13	12	
01	14	15	17	16		
11	12	13	15	14		
10	5	6	11	10		

$$T_1 = 1$$

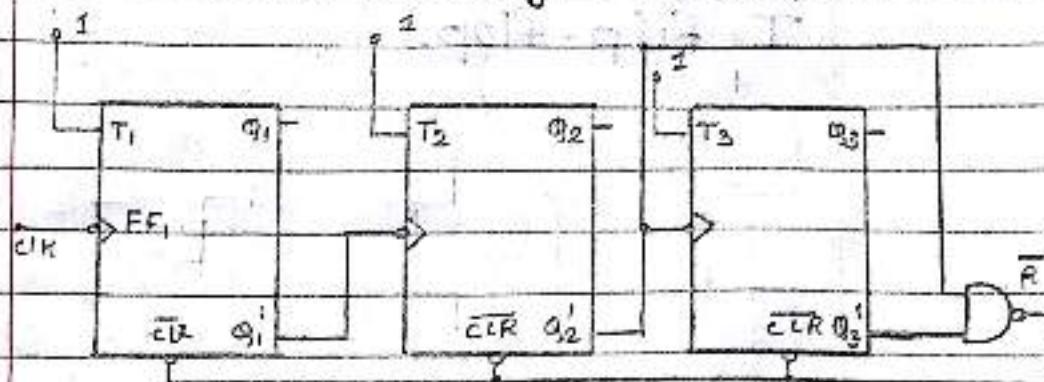
Q.17 Design 3 bit asynchronous ripple Up Counter using T flip-flop



	After Pulses	State	$Q_3$	$Q_2$	$Q_1$	R
0			0	0	0	0
1			1	0	0	0
2			0	1	0	0
3			0	1	1	0
4			1	0	0	0
5			1	0	1	0
6			1	1	0	1
7			0	0	1	0

Q.18

Q.18 Design 3 bit asynchronous ripple up Down Counter Using T flip-flop



After Pulses	Count
--------------	-------

$Q_3 \cdot Q_2 \cdot Q_1$

0 0 0 0

1 0 0 1

2 0 1 0

3 0 1 1

4 1 0 0

5 1 0 1

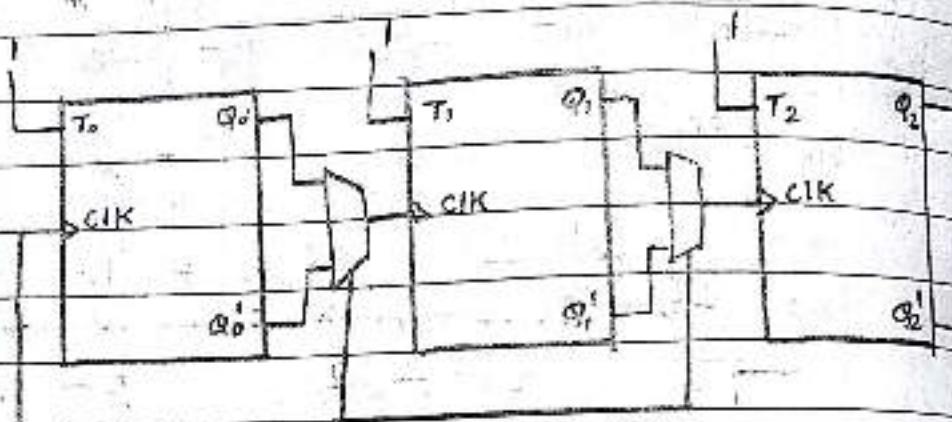
a) count Table

		$Q_2 Q_1$			$Q_3$		
		00	01	11	10	0	1
		0	0	1	3	2	1
0	0	0	0	1	3	2	1
1	1	4	15	X <sub>5</sub>	X <sub>6</sub>	0	1

b) K-Map

$$R = Q_3' Q_2'$$

**Q.19 Design 3-bit asynchronous/  
ripple UP/DOWN Counter using  
T flip-flop.**



**Q.20 Design Synchronous Mod-5 Up  
Counter**

PS	NS	Required exci.
Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>	Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>	J <sub>3</sub> K <sub>3</sub> J <sub>2</sub> K <sub>2</sub> J <sub>1</sub> K <sub>1</sub>
0 0 0 0	0 0 0 1	X X X X X X X X
1 0 0 1	0 1 0 0	X X X X X X X X
0 1 0 0	0 1 1 1	X X X X X X X X
1 0 0 0	1 0 0 0	X X X X X X X X

Q <sub>3</sub> Q <sub>2</sub> Q <sub>1</sub>		T <sub>3</sub>	T <sub>2</sub>	T <sub>1</sub>
0 0	0 1	0	0	1
0 1	1 1	0	1	1
1 0	0 0	1	0	0
1 1	1 0	X	X	X
		a		
		b		
		c		
		d		

= Q<sub>3</sub> Q<sub>2</sub>' Q<sub>1</sub>' + Q<sub>3</sub>' Q<sub>2</sub> Q<sub>1</sub>

= Q<sub>3</sub> ⊕ Q<sub>2</sub> Q<sub>1</sub>

Using

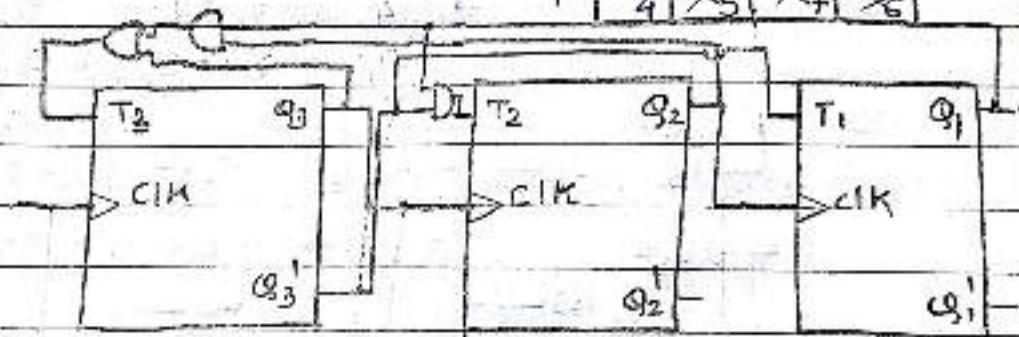
$$T_3 = Q_3 + Q_2 Q_1$$

		$Q_3$	$Q_2$	$Q_1$	11	10
		0	0	1	1	2
		1	1	X <sub>5</sub>	X <sub>7</sub>	X <sub>6</sub>

$$T_2 = Q_3' Q_1$$

$$T_1 = Q_3'$$

		$Q_3$	$Q_2$	$Q_1$	11	10
		0	1	1	1	2
		1	1	X <sub>5</sub>	X <sub>7</sub>	X <sub>6</sub>



Q.21 Design Synchronous Mod-5 down Counter

	P.S	N.S	Required excita.
	$Q_3$ $Q_2$ $Q_1$	$Q_3$ $Q_2$ $Q_1$	$T_3$ $T_2$ $T_1$
0 X	0 0 0	1 0 0	1 0 0
T <sub>1</sub>	1 0 0	0 1 1	1 1 1
	0 1 1	0 1 0	0 0 1
	0 1 0	0 0 1	1 0 1
0	0 0 1	0 0 0	0 0 1

		$Q_2 Q_1$			
		00	01	11	10
$Q_3$	0	1	X	3	2
	1	4	$X_5$	$X_7$	$X_6$

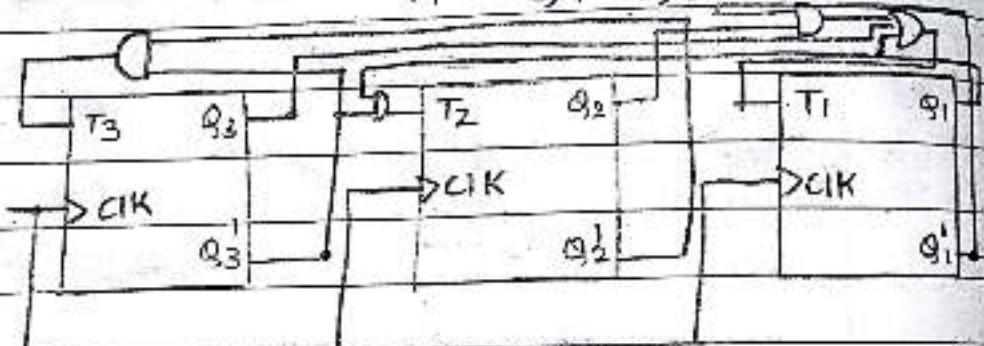
$$T_3 = Q_3' Q_2'$$

		$Q_2 Q_1$			
		00	01	11	10
$Q_3$	0	0	1	3	2
	1	4	$X_5$	$X_7$	$X_6$

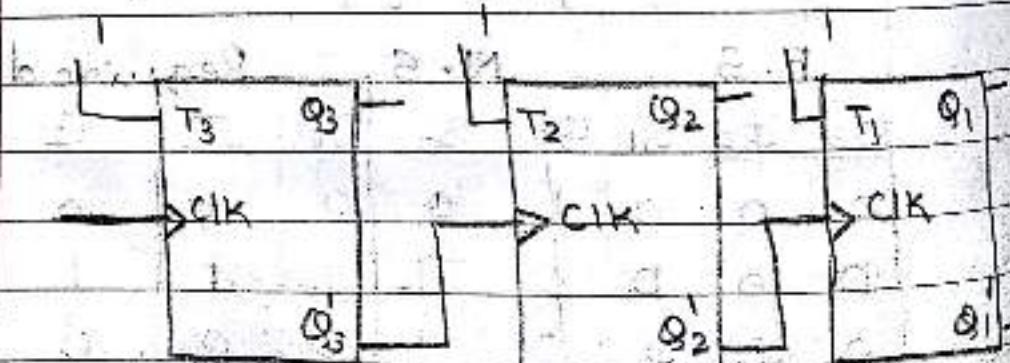
$$T_2 = Q_3' Q_1$$

		$Q_2 Q_1$			
		00	01	11	10
$Q_3$	0	0	1	3	2
	1	1	4	$X_5$	$X_7$

$$T_1 = Q_1 + Q_3 + Q_2 Q_1$$



Q.22 Design a Synchronous Mod-5 Up Counter



a) Logic diagram

$Q_3$	$Q_2$	$Q_1$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1

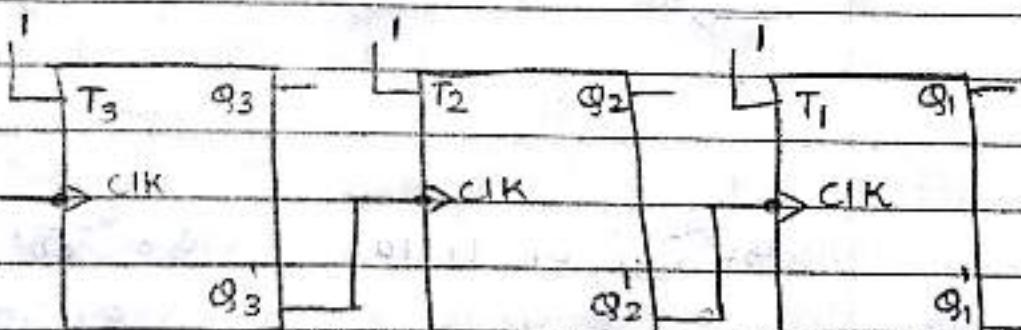
$Q_3 Q_2 Q_1$   
 $000 \ 011 \ 10$

0	0	1	3	e
1	4	5	x	x

c) K Map

b) Count Table

Q.23 Design asynchronous Mod-5 Down Counter



a) Logic diagram

$Q_3$	$Q_2$	$Q_1$
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0

$Q_3 Q_2 Q_1$   
 $000 \ 011 \ 10$

0	1	3	e
1	4	5	x

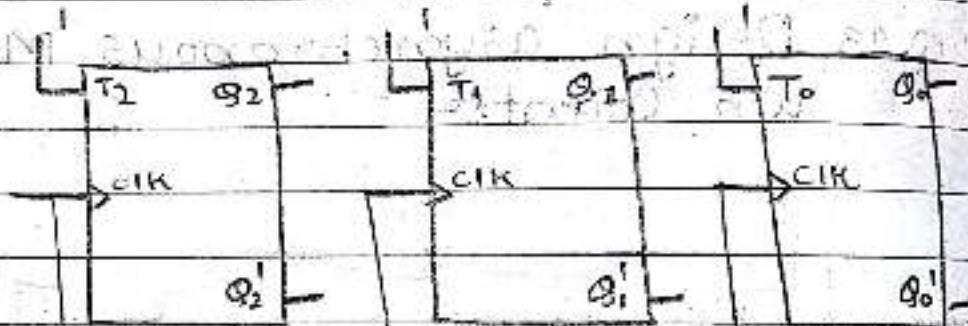
c) K Map

b) Count Table

Q.24 Design Counter which Counts binary Sequence 1, 4, 2, 7 & repeat.

PS	NS						I/Ps		
$Q_2$	$Q_1$	$Q_0$	$Q_2$	$Q_1$	$Q_0$	$T_2$	$T_1$	$T_0$	

1	0	0	1	1	0	0	1	0	$T_2 = 1$
4	1	0	0	0	1	0	1	1	D
2	0	1	0	1	1	1	1	0	I
7	1	1	0	0	1	1	1	0	O



$Q_3$	$Q_2$	$Q_1$	00	01	11	10	$Q_3$	$Q_2$	$Q_1$	00	01	11	10
0	0	1	3	2	-	-	0	0	1	3	2	-	-
1	1	4	5	6	7	6	1	4	5	7	6	7	6

$$\begin{aligned}
 & - Q_3(Q_2'Q_1 + Q_2Q_1') \\
 & = Q_3(Q_2 \oplus Q_1)' \\
 & = Q_3(Q_2 \odot Q_1)
 \end{aligned}$$

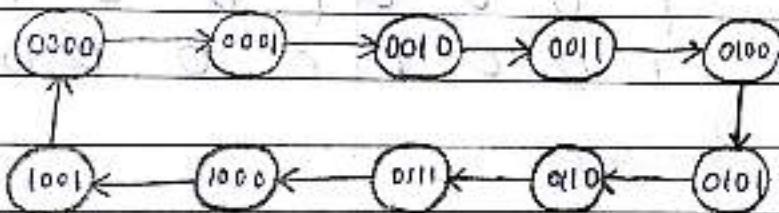
Q.25 Design/Explain BCD Counter

Step-1 The number of flip-flops: A BCD Counter is nothing but a mod 10 Counter. It is a decade Counter.

3 bits required

It has 10 states (0000 through 1001). It requires  $m=4$  FFs. Four FFs can have 16 states. After the tenth clock pulse, the counter resets. So, states 1010 through 1111 are invalid. The entries for excitations corresponding to invalid states are don't cares.

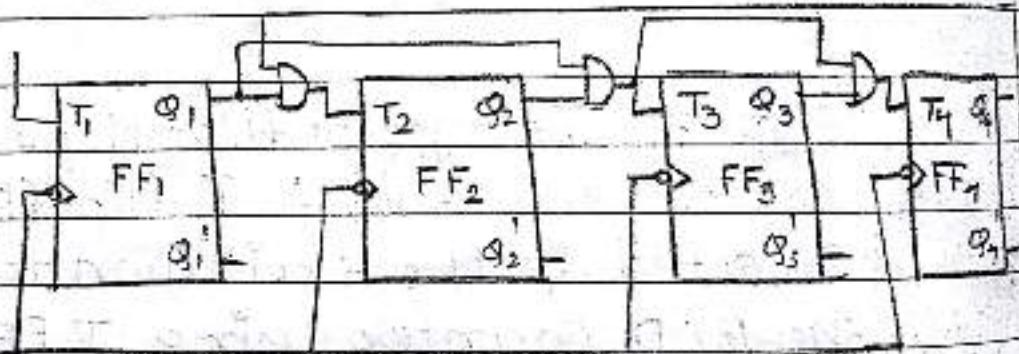
**Step-2** The state diagram: The state diagram of BCD counter is drawn below



**Step-3** The type of flip-flops & the excitation table : T flip-flops are selected & the excitation table of the BCD counter using T-FF

**Step-4** The logic diagram based on those minimal expression

P.S	N.S	Required excitation
$Q_4\ Q_3\ Q_2\ Q_1$	$Q_4\ Q_3\ Q_2\ Q_1$	$T_4\ T_3\ T_2\ T_1$
0 0 0 0	0 0 0 0	1 0 0 0
0 0 0 1	0 0 1 0	0 0 0 1
0 0 1 0	0 0 0 1	0 0 0 0
0 0 1 1	0 1 0 0	0 0 1 1
0 1 0 0	0 1 0 1	0 0 0 0
0 1 0 1	0 1 1 0	0 0 0 1
0 1 1 0	0 1 1 1	0 0 0 0
0 1 1 1	1 0 0 0	1 1 1 1
1 0 0 0	1 0 0 1	1 0 0 1
1 0 0 1	0 0 0 0	1 0 0 1



Q.26 Explain Ring Counter or Twisted Ring

Ring Counter: A 10-bit ring counter will produce a sequence of 10-bit groups having the property that each group has a single

The ring-counter code shown obtained by assigning a decimal digit to each of these ten patterns.