

## Digital

Boolean logical ideas :- Boolean logical ideas are characterised into 3-ways -

1. Producing the constt. (0,1) [null, identity operations]
2. Unary Operations : [Complement, Transfer].
3. Binary operations : [And, Or, Nand, Nor, Ex-or, Ex-Nor, Inhibition, Implication]

x	y	$f_0$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$	$f_9$	$f_{10}$	$f_{11}$	$f_{12}$	$f_{13}$	$f_{14}$	$f_{15}$
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

# #

$f_0 = 0 \Rightarrow$  Null-operation.

$f_1 = x \cdot y \Rightarrow$  AND-operation. [  $x \wedge y$  ].

$f_2 = x \cdot y' \Rightarrow$  Inhibition.

$\Rightarrow x/y \Rightarrow$  [  $x$  but not  $y$  ].

$f_3 \Rightarrow x$  Transfer op. (Buffer-gate)

$f_4 \Rightarrow x' \cdot y \Rightarrow$  Inhibition.

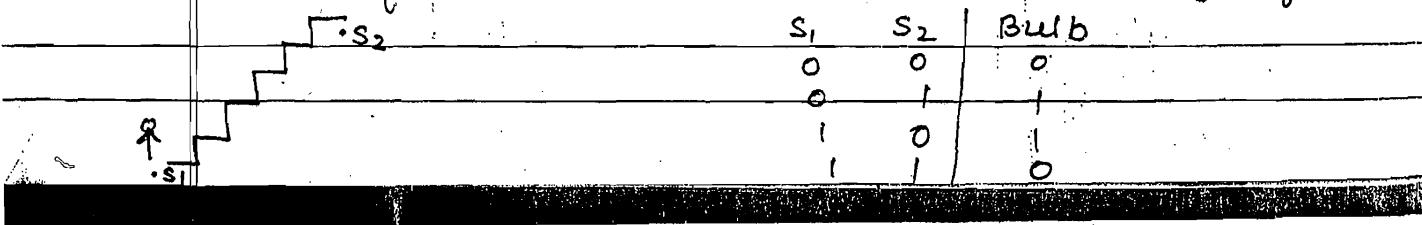
$y/x \Rightarrow$  [  $y$  but not  $x$  ].

$f_5 \Rightarrow y$  [ Transfer op. ] [ Buffer-gate ]

$f_6 \Rightarrow x \oplus y$  [ EX-OR ]

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

EX-OR gate is also known as stair-case logic gate.



$f_7 = x + y$  OR-gate

$$\rightarrow [x \vee y]$$

$f_8 = \overline{x+y}$  Nor-gate

$$\rightarrow [x \downarrow y]$$

$f_9 = x \oplus y$  Ex-Nor

$$= xy + x'y'$$

$f_{10} = \bar{y}$  [complementary operation] [Not-gate]

$f_{11} = x + y'$  [Implication op.]

$x \Rightarrow y$  [if  $y$  then  $x$ ].

$f_{12} = \bar{x}$  (complementary)

$f_{13} = x'y$  (Implication)

$x \Rightarrow y$  [if  $x$  then  $y$ ].

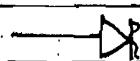
$f_{14} = \overline{x \cdot y}$  (Nand)

$$x \uparrow y$$

$f_{15} = 1$  [Identity].

Logic-Gate Symbols :-

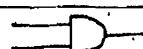
Buffer

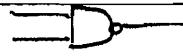


NOT



AND



NAND - 

OR - 

NOR - 

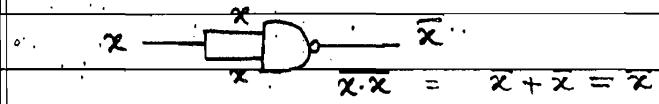
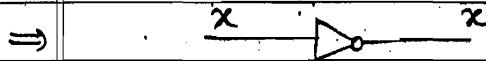
EX-OR - 

A.U.A

EX-NOR - 

### NOTE:

\* NAND, NOR known as "UNIVERSAL - LOGIC - GATES."



### NAND & NOR

NOT	1	1
-----	---	---

AND	2	3
-----	---	---

OR	3	2
----	---	---

EX-OR	4	5
-------	---	---

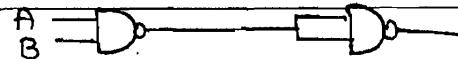
EX-NOR	5	4
--------	---	---

\* For  $n$ -variables, we get  $2^n$  combinations.  
and we get  $2^{2^n}$  possible - functions.

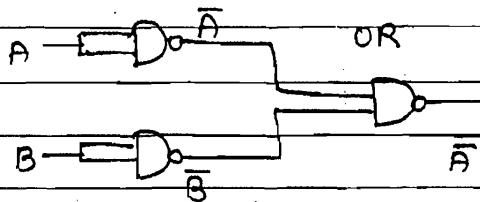
NOT



AND

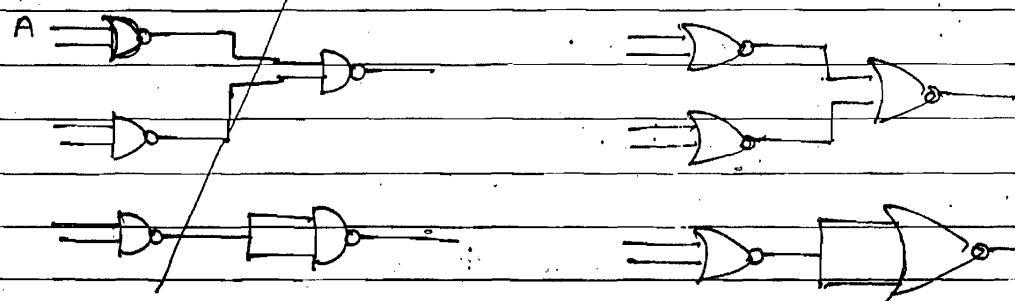


OR

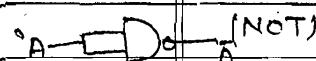


$$\bar{A} \cdot \bar{B} = A + B$$

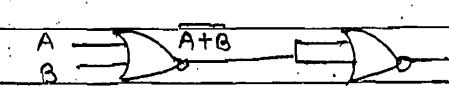
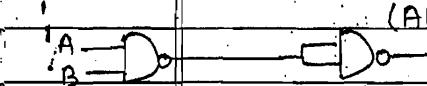
NOR



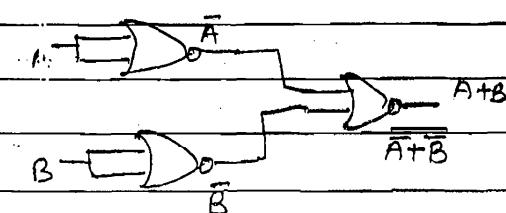
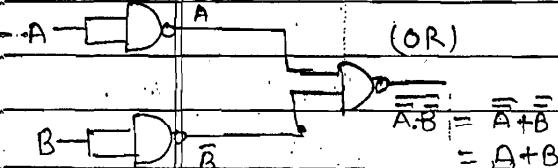
Practise :

NAND - Gates

(AND)



(OR)



$$A \cdot B = \bar{\bar{A}} + \bar{\bar{B}} = A + B$$

Duality:

Step: 1 Interchange the operators [ . , + ]

Step: 2 Interchange the identities [ 0, 1 ].

e AND-Laws

OR

eg. ①  $x \cdot \bar{x} = 0$

$\underline{x + \bar{x} = 1}$

A.U.A

eg. ②  $x + 0 = x$

$\underline{x \cdot 1 = x}$

E.C.S

eg. ③  $f = (A' \cdot B) + C$

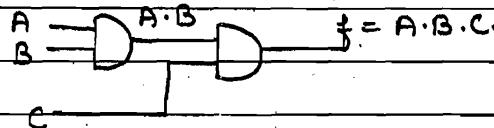
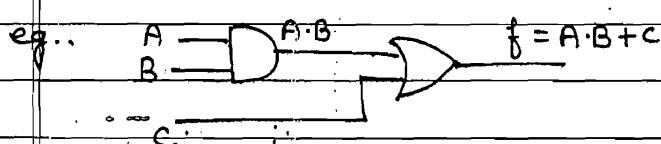
$f^D = ?$

$f^D = (A' + B) \cdot C$

$\Rightarrow (f^D)^D = f$

# Degenerative form :- When a 3-level logic gate's system output is expressed with a single logic-gate then the 3-level logic gate sys. is known as degenerative form for single logic gate.

e.g. AND - AND is degenerative form for the 'AND' - gate.

Non-degenerative form :-

E.G.S

A.U

Short-cut-

Duality

AND - OR

OR - AND

NAND - NAND

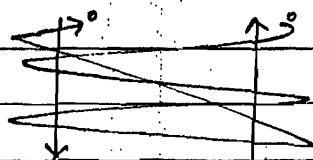
NOR - NOR

NOR - OR

NAND - FND

OR - NAND

AND - NOR



$\Rightarrow$  Combinations of the same-level in above representation are dual-forms.

Note:

doing the inversion before the binary operation is not same as that of doing the inversion after the binary operation." But for the unary operations it will be the same - results.

eg.. ①  $x \rightarrow \overline{x}$

$$x \rightarrow \text{NOT} \rightarrow \overline{x}$$

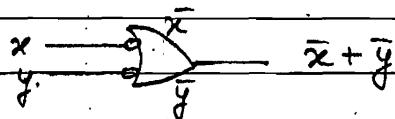
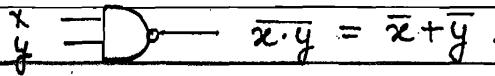
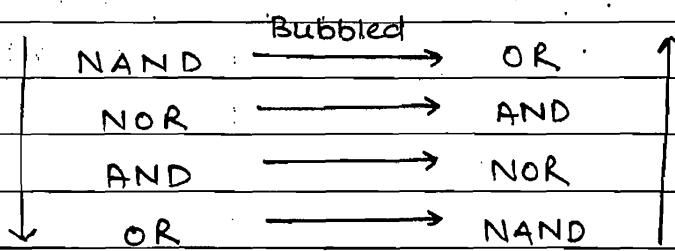
②  $x \rightarrow \overline{x+y} = \overline{x} \cdot \overline{y}$

$$\begin{array}{ccc} x & \rightarrow & \overline{x} \\ y & \rightarrow & \overline{y} \end{array} \rightarrow \overline{\overline{x} + \overline{y}}$$

Bubbled-OR

Alternative - logic - Gates :-

eg..

short-cut :-Positive , Negative - logics :-

+ve - logic

High  $\rightarrow$  1Low  $\rightarrow$  0

-ve - logic

High  $\rightarrow$  0Low  $\rightarrow$  1

eg..  $\begin{cases} -7V \\ -2V \end{cases} \Rightarrow$  +ve - logic

(+)'ve AND - logic

(-)'ve AND - logic = OR (+)ve logic

A	B	$f = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

A	B	$f$
1	1	1
1	0	0
0	1	1
0	0	0

∴ -ve AND = +ve OR (Vice-versa)

∴ -ve NAND = +ve NOR (Vice-versa)

### Ex-OR Specialities :-

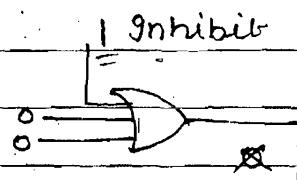
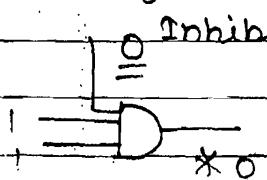
$x \oplus x$	=	0 ↑
$x \oplus x'$	=	1 ↑
$x \oplus 1$	=	$x'$ ↑
$x \oplus 0$ ↓	=	$x$

$$\begin{array}{c} \overline{x \oplus y} \\ \left. \begin{array}{c} x \oplus y \\ x' \oplus y \\ x \oplus y' \end{array} \right\} \Rightarrow x \odot y \end{array}$$

$$\begin{aligned} \text{eg.. } x' \oplus y &= \bar{x}y + x\bar{y} \\ &= xy + x'y' \\ &= x \odot y \end{aligned}$$

$$x \oplus y \oplus z \Rightarrow x \odot y \odot z$$

Inhibiting the logic-gates :-

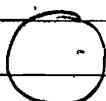


Short-Cut :-

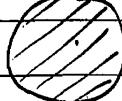
	Disable	Enable
NAND	L	H
NOR	H	L
AND	L	H
OR	H	L

## Venn-Diagramm :-

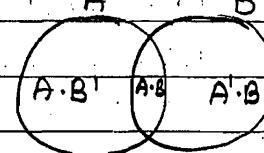
$$A = 0$$



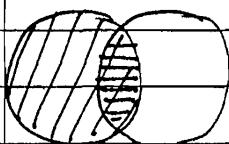
$$A = 1$$



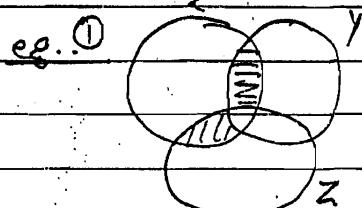
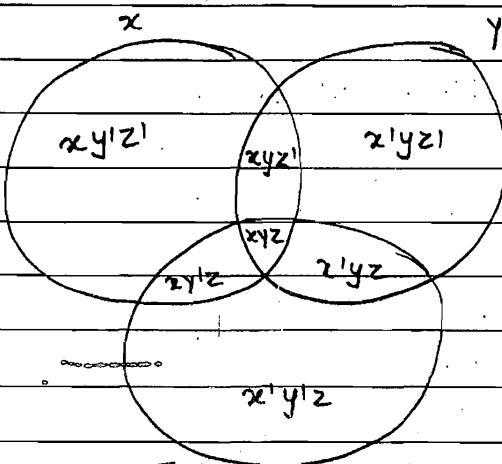
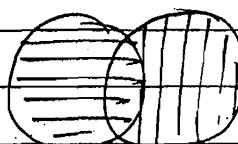
$$A + B = \bar{A} \cdot \bar{B}$$



$$\Rightarrow x + xy = x$$



$$\Rightarrow x + x'y = x + y$$



$$\Rightarrow f = xyz' + x'y'z$$

## LOGIC-MINIMIZATION :-

### Boolean-laws :-

NOT :

$$\text{if } x = 0 \quad \therefore \quad \bar{x} = 1$$

$$(x')' = x$$

AND :

$$x \cdot x = x$$

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

$$x \cdot \bar{x} = 0$$

OR :

$$x + x = x$$

$$x + 0 = x$$

$$x + \bar{x} = 1$$

$$x + 1 = 1$$

Commutative - law :-

$$x \cdot y = y \cdot x$$

$$x + y = y + x$$

Associative - law :-

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

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

Distributive laws :-

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

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

De-Morgan's - law :-

$$\overline{x \cdot y} = \overline{x} + \overline{y}$$

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

# Consensus - Theorem :-

A variable is associated with some variable and its complement is also with some other vari. and next term is followed formed by leftover variab.. then the term is redundant.

$$\rightarrow A \cdot B + \bar{A} \cdot C + BC = A \cdot B + \bar{A} \cdot C$$

$$\rightarrow (A+B) \cdot (\bar{A}+C) \cdot (B+C) = (A+B) \cdot (\bar{A}+C)$$

NOTE:

$\Rightarrow$  Consensus- The. can be extended to any no. of variable.

$$A \cdot B + \bar{A} \cdot C + BCDE = A \cdot B + \bar{A} \cdot C$$

...  
↑  
must be  
there

# Transposition - Theorem :-

operators and associations can be interchange.

$$\Rightarrow A \cdot B + \bar{A} \cdot C = (A+C) \cdot (\bar{A}+B)$$

$$\Rightarrow (A+B) \cdot (\bar{A}+C) = (A \cdot C) + (\bar{A} \cdot B)$$

# Redundant Lateral Rule (RLR) :-

ORing a variable with ANDing of its complement by another variable results in same ORing of those two variables.

$$x + x \cdot y = x + y$$

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

Absorption - law :-

Absor' n - law :-

ORing of a variable with ANDing of that variable by another variable results in same - variable

$$\rightarrow x + xy = x$$

$$\rightarrow x \cdot (x+y) = x$$

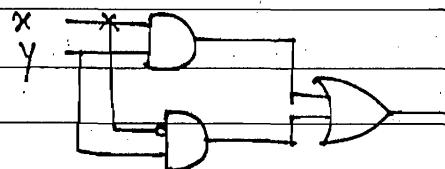
SOP :-

•- (Minterm)

•- (AND-OR)

x	y	minterm	f
0	0	$\bar{x} \cdot \bar{y}$	1
0	1	$\bar{x} \cdot y$	1
1	0	$x \cdot \bar{y}$	0
1	1	$x \cdot y$	1

$$f = x \cdot y + \bar{x} \cdot y$$



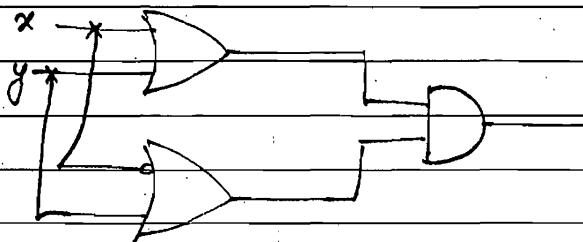
POS :-

(Maxterm)

(OR-AND)

x	y	maxterm	f'
0	0	$x+y$	0
0	1	$x+\bar{y}$	1
1	0	$\bar{x}+y$	0
1	1	$\bar{x}+\bar{y}$	1

$$f = (x+y) \cdot (\bar{x}+y)$$



Half-Adder :-

$$\begin{aligned} \text{sum} &= \bar{x} \cdot y + x \cdot \bar{y} \\ &= x \oplus y \end{aligned}$$

x y C sum

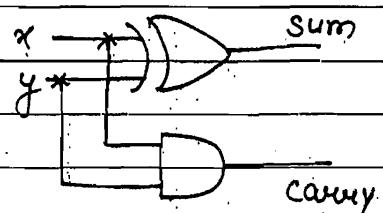
0 0 0 0

0 1 0 1

1 0 0 1

1 1 1 0

(Carry)  
(x · y)



Minterm = Maxterm

Complementrial Boolean - Expression :-

Step : ① duality.

Step : ② complement the individual variable.

e.g.  $f = A' \cdot B + C$

$\bar{f} = ?$

Step : ①  $(A' + B) \cdot C$

Step : ②  $(A + B') \cdot \bar{C} = \bar{f}$

x	y	f
0	0	0 → 1
0	1	1 → 0
1	0	0 → 1
1	1	1 → 0

Step:①  $f = x \cdot y + \bar{x} \cdot \bar{y}$

Step:②  $f = (x+y)(\bar{x}+y) \dots$  (dual form)

→  $\bar{f} = (\bar{x}+\bar{y})(\bar{\bar{x}}+\bar{y}) \dots$  complement<sup>n</sup> of andiv.

⇒ This can also be written from the table with 'new-zeroes' and i.e.

$$f = (x+\bar{y}) \cdot (\bar{x}+\bar{y})$$

Step:① ⇒  $f = (x \cdot \bar{y}) + (\bar{x} \cdot \bar{y}) \dots$  (dual form)

Step:③ ⇒  $f = (\bar{x} \cdot y) + (x \cdot y) \dots$  (complement<sup>n</sup> of andiv.)

Home-work :

Workbook - Problems : P.NO. 50

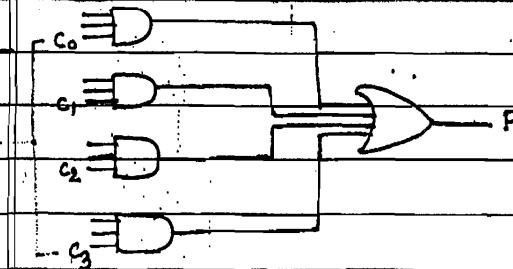
Q:12 Let  $f(A, B) = A \oplus B$ .

$$f(f(x \oplus y, z) \oplus w) = ?$$

$$\because f(A, B) = A \oplus B$$

$$f(f(x \oplus y, z) \oplus w) = (x \oplus y \oplus z) \oplus w$$

Q:13



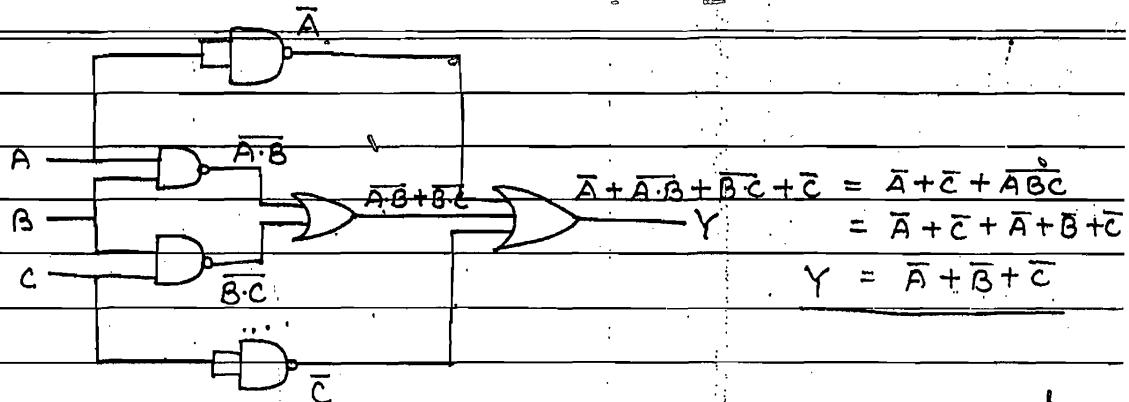
EX-OR is also known as  $A \oplus B$

$$\Rightarrow c_0 \oplus c_1 \oplus c_2 = ?$$

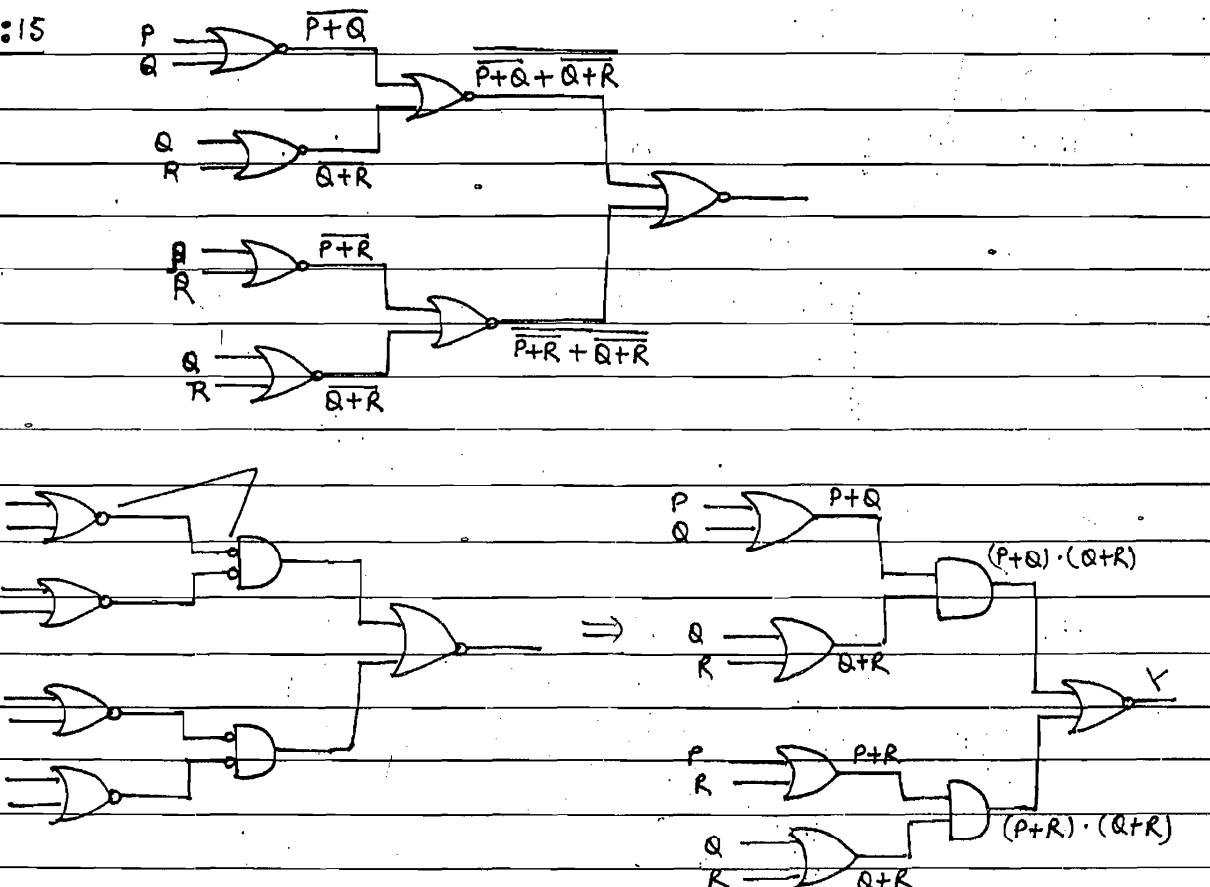
$$\Rightarrow c_3 \oplus c_0 \oplus c_1 \oplus c_2 = ?$$

$$(c_1, c_2, c_3)$$

Ques: 14



Ques: 15



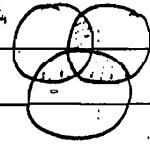
$$Y = (P+Q) \cdot (\bar{Q}+R) + (P+R) \cdot (\bar{Q}+R)$$

$$= Q \cdot \bar{Q} + P \cdot \bar{Q} + P \cdot R + Q \cdot R$$

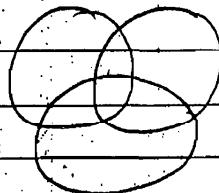
$$= P \cdot (Q + R) + Q \cdot (P + R)$$

$$= P \cdot Q + P \cdot R + Q \cdot P + Q \cdot R$$

Ques: 16



Ques: 18



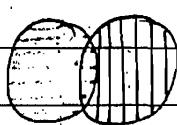
Ques: 20

$$\text{A} \cdot \overline{\text{B}} = \gamma$$

$$\text{Ques: 21 } [(A + A\bar{B})(A + \bar{A}\bar{B})] + [(\bar{C}D + \bar{C}\bar{D}) + (\bar{C}\oplus D)]$$

Ques: 24

$$f = A + \bar{A}\bar{B}$$



## # CONVERSION FROM SOP - POS and POS - SOP :-

$$\Sigma m \Rightarrow SOP$$

$$\Pi m \Rightarrow POS$$

$$\textcircled{1} \quad \Sigma m(0, 2) = \bar{x} \cdot \bar{y} + x \cdot \bar{y}$$

↓      ↓  
00    10  
 $x \cdot y$     $\bar{x} \cdot y$

$$\textcircled{2} \quad \Pi M(0, 2) = (x+y) \cdot (\bar{x}+y)$$

↓      ↓  
00    10  
 $x+y$     $\bar{x}+y$

$$\textcircled{3} \quad \Sigma m(0, 2) = \Pi M(0, 3)$$

↓  
10

$$\textcircled{4} \quad \Pi M(0, 5) = \Sigma m(1, 2, 3, 4, 6, 7)$$

4-bit

$$\textcircled{5} \quad f_1 = \Sigma m(0, 1, 2, 3).$$

$$f_2 = \Sigma m(2, 3, 5).$$

$$f_3 = \Sigma m(3, 4, 6, 7)$$

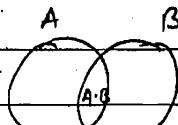
$f_1 \cdot f_2 = \Sigma m(2, 3)$

$f_1 \cdot f_2 \cdot f_3 = \Sigma m(2, 3, 5)$

$f = f_1 f_2 + f_3$

$= \Sigma m(2, 3) + \Sigma m(3, 4, 6, 7)$

$\Sigma m(2, 3, 4, 6, 7)$



$$A = \Sigma m(0, 1, 2, 4, 7), \quad B = \Sigma m(1, 2, 3, 5, 6, 7)$$

$A \rightarrow$   $B \rightarrow$   $f = ?$

$= A \oplus B = \bar{A}B + A\bar{B}$

$$\bar{A}B = \Sigma m(3, 5, 6)$$

$$A\bar{B} = \Sigma m(0, 4)$$

$$A \oplus B = \Sigma m(0, 3, 4, 5, 6)$$

**K-Map:** Modification of Venn-diag. is only known as K-Map.

- o- K-Map is the group of adjacent cell.
- o- each cell is represented by group of literals (Minterms)

$A'$	$A$	$A'$	$A$	$A'$
$B'$	$B$	$B'$	$B$	
		$B'$	$B$	

$k\text{-Map}:-$	$A'$	$A' \cdot B'$	$A' \cdot B$
	$A$	$A \cdot B'$	$A \cdot B$

#### Notes:

- \* Neighbouring cell should differ by only '1' literal.
  - \* Generalized procedure to Simplify a Boolean Exp.  
By K-Map:-
1. Plot the K-Map and place 1's corresponding to Minterm of SOP - Exp.
  2. Check the K-Map for 1's which are not adjacent to any other 1's. They are isolated minterms. They are to be read as they are becz they can't be combine.

3. Check for those 1's which are adjacent to neighbouring 1's.
4. Follow the order of grouping from higher order to lower order.
5. Identify the redundant - groups and get the Expression by leaving the redundant - groups.

\* Redundant Group should be consider only for the case of Hazard-free circuits.

### K-Map - Representation :

		$\bar{B}$	B	
(1)	$f \notin A, B$	$\bar{A}$	$\bar{A}\bar{B}$	$\bar{A}B$
	A	$A\bar{B}$	AB	

		$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$	
(2)	$f(A, B, C)$	$\bar{A}$	$\bar{A}\bar{B}\bar{C}_0$	$\bar{A}\bar{B}C_1$	$\bar{A}B\bar{C}_2$	$\bar{A}B\bar{C}_3$
	MSB      LSB	A	$A\bar{B}\bar{C}_4$	$A\bar{B}C_5$	$ABC_6$	$A\bar{B}\bar{C}_7$

		$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$	
(3)	$f(A, B, C)$	$\bar{C}$	$\bar{C}\bar{A}\bar{B}_0$	$\bar{C}\bar{A}B_1$	$\bar{C}AB_2$	$\bar{C}A\bar{B}_3$
	MSB      LSB	C	$C\bar{A}\bar{B}_4$	$C\bar{A}B_5$	$CA\bar{B}_6$	$CA\bar{B}_7$

		$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$	
4.	$f(A, B, C)$	$\bar{C}$	0	1	3	2
	MSB      LSB	C	4	5	7	6

Question:  $f = \sum m(0, 1, 2, 6)$

$\downarrow$	$\downarrow$	$\downarrow$	$\rightarrow$
000	001	010	110

$$f = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}$$

### 1st method

st method	$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$	
$\Rightarrow$	A	1 0	1	3	1 2
A	4	5	7	1 6	

$$f = \bar{A}\bar{B} + B\bar{C}$$

, and

	$\bar{A}\bar{B}$	$\bar{A}B$	$AB$	$A\bar{B}$	
$\bar{C}$	1 0	1 2	1 6		4
C	1 1	3	7		5

$$f = \overline{DB} + \overline{BC}$$

83

	$\overline{CD}$	$\overline{CD}$	$CD$	$CD$
$\overline{AB}$	10	11	3	12
$\overline{AB}$	4	15	17	6
$AB$	12	13	15	14
$A\overline{B}$	1	8	9	11

$$A\bar{C}\bar{D} + B\bar{D} \rightarrow \bar{B}\bar{D}$$

eg.

	B.C.	B.C.	B.C.	.B.C.
A'		1	1	
A		1	1	

$$f \equiv c$$

$$f = \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$$

\*\*

1 Pair

for Pair :  $(n-1)$ ;  $n \rightarrow$  variable no.

2 Quad

for Quad :  $(n-2)$ 

3 Octet

for Octet :  $(n-3)$ eg. In a 5-variable K-Map, octet gives:  $5-3 = 2$  literals

\* Redundancy - grouping :- With (or) Without the redundant group there's no effect on the result of the CLR.

eg.. ①

$$f = \sum m(1, 3, 6, 7)$$

	$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$
$\bar{A}$	0	1	1	2
A	4	5	1	6
...				

$$f = \bar{A}\bar{C} + AB + BC$$

EPI    EPI    X [R.P.I.]

$$f = \bar{A}\bar{C} + AB$$

eg.. ②

$$f = \sum m(1, 5, 6, 7, 11, 12, 13, 15)$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	1	2
$\bar{A}B$	4	1	5	7
AB	11	12	13	14
A $\bar{B}$	8	9	10	15

$$f = \begin{matrix} EPI & EPI & EPI \\ ABC & ACD & \bar{A}\bar{C}D \\ + BD & & R.P.I. \end{matrix}$$

\* Implicant: The minimum corresponding to a cell in K-Map which is having '1' in it.

Prime-implicant: All possible grouping in K-Map, are prime implicants.

Essential Prime Imp.: EPI is PI only in which atleast a single 1 is which is only 1-time grouped (Uniquely - grouped).

Non-Prime Imp.: Implicant which can't get the grouping is known as Non-Prime - Imp.

Redundant - P.I :- It is the prime implicant by which is in such a way that with (or) without that there's no effect on CKR - result known as R.P.I.

Selective Prime Imp.: It is the P.I. for which it was man-at decided for the selection.

Note:

- Max-term is known as false - Minterm.
- If the grouping is done by taking 0s then those can be treated as :-
  1. False Implicant
  2. False Prime Implicant.
  3. False Essential P.I.
  4. False R.P.I.
  5. False S.P.I.

Examples:

$\bar{B}\bar{C}$     $\bar{B}C$     $BC$     $B\bar{C}$

$$f = \Sigma m(0, 1, 2, 6, 7)$$

$\bar{A}$	1	0	1	1	3	1	0
A	4	5	7	6			

$\bar{A}\bar{B}$  (0, 1)   E.P.I. (P.I.)

$\bar{A}\bar{C}$  (0, 2)    $\boxed{P.I.}$    ( $P.I.$ )

$\bar{B}\bar{C}$  (2, 6)    $\boxed{P.I.}$    ( $P.I.$ )    $\rightarrow$  S.P.I.

$AB$  (6, 7)   E.P.I. - (P.I.)

Imp.  $\rightarrow$  SP.I.  $\rightarrow$  4Non-P.I.  $\rightarrow$  0E.P.I.  $\rightarrow$  2

eg. (2)

	$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$	
$\bar{A}$	1	0	1	0	3
A	4	5	6	7	1

 $\bar{A}\bar{B}\bar{C}$      $BC$      $AB$ Imp.  $\rightarrow$  4Non-Imp.  $\rightarrow$  1P.I.  $\rightarrow$  2E.P.I.  $\rightarrow$  2

eg. (3)

	$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$	$\bar{A}\bar{B}$	AB
$\bar{A}$	1	0	1	3	2	E.P.I.
A	4	5	6	7	1	(P.I.)

# DIAGONAL GROUPING :-

	$\bar{B}$	B
$\bar{A}$	1	0
A	1	0

	$\bar{B}$	B
$\bar{A}$	1	0
A	1	0

$f = \bar{A}\bar{B} + AB$

$f = \bar{A}B + A\bar{B}$

$f = A \odot B$

$f = A \oplus B$

	$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$
$\bar{A}$	1	0	1	0
A	1	0	1	0

	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$
$\bar{A}\bar{B}$	1	1	1	1
AB	1	1	1	1

$\bar{B}[\bar{A}\bar{C}] + B[A\oplus C]$

$f = A\bar{C}[B\oplus D] + \bar{A}C[B\oplus D]$

 $A \odot B \Rightarrow$  when on with same nature $+ B\bar{D}[\bar{A}\bar{C}]$  $A \oplus B \Rightarrow$  when on with opposite nature

#

OFFSETS in K-Map :-offset  $\bar{A}(B \oplus C)$ 

eg.. ①

	$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	$BC$
$\bar{A}$	1	0	1	0
A	4	1	5	7
	2	3	6	8

$$A[B \oplus C]$$

$$\bar{A}\bar{B}(C \oplus D)$$

eg.. ②

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	1	0	1	0
$\bar{A}B$	4	5	7	6
AB	12	13	15	14
A $\bar{B}$	8	9	11	10

$$\bar{C}D(A \oplus B)$$

$$\bar{A}\bar{B}(C \oplus D)$$

#

POS forms in K-Map :-

\* Step: ① Operator change.

\* Step: ② complement the given variable.

eg-

$$f = \pi M(0, 1, 3, 7)$$

	$B^tC$	$B^t\bar{C}$	$\bar{B}^t\bar{C}$	$\bar{B}^tC$
$\bar{A}$	0	0	0	3
A	4	5	0	7
	2	1	6	8

$$f = (A+B) \cdot (\bar{B} + \bar{C})$$

Ques:  $f = \pi M(0, 1, 3, 7)$

$f = \sum m(2, 4, 5, 6)$

	$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	$BC$	
$\bar{A}$	0	1	3	1	2
A	4	1	5	7	6

$$f = A\bar{B} + B\bar{C}$$

$f = \pi M(0, 1, 3, 7)$

$f = \sum m(2, 4, 5, 6)$

$f' = \sum m(0, 1, 3, 7)$

## # SOP to SSOP [Canonical-Form]:

Ques:

$$f = \bar{A}B + \bar{A}B\bar{C} + AC.$$

Method I

$$f = \bar{A}B(C + \bar{C}) + \bar{A}B\bar{C} + AC(B + \bar{B})$$

$$= \bar{A}BC + \bar{A}B\bar{C} + \bar{A}B\bar{C} + ACB + AC\bar{B}$$

$$f = \bar{A}BC + \bar{A}B\bar{C} + ACB + AC\bar{B}.$$

Method II

	$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	$BC$	
$\bar{A}$	0	1	3	1	2
A	4	1	5	7	6

$$\{ f = \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}C + ABC. \}$$

Ques:

$$f = A + BD$$

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$		$f = \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D$
$\bar{A}\bar{B}$	0	1	3	2		$+ A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D + ABCD$
$\bar{A}B$	4	1	5	7	6	$+ ABC\bar{D} + A\bar{B}\bar{C}\bar{D} +$
$A\bar{B}$	1	2	1	3	15	$A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D +$
$AB$	8	1	9	1	11	$A\bar{B}CD + A\bar{B}C\bar{D}$

**DON'T-CARE - Conditions :** There'll be some situations in the circuit designing so that for certain inputs, the output is unspecified such type of invalid inputs corresponding output treated as don't care (X).

eg.

$$f = \sum m(1) + \phi(3)$$

 $\bar{B}\bar{C} \quad \bar{B}C \quad BC \quad B\bar{C}$ 

$\bar{A}$	0	(1)	(1) X	0
A	0	0	0	0

without  $\phi \Rightarrow \bar{A}\bar{B}\bar{C}$

with  $\phi \Rightarrow \bar{A}\bar{C}$

eg.

 $\bar{C}\bar{D} \quad \bar{C}D \quad CD \quad C\bar{D}$ 

$\bar{A}\bar{B}$				
$\bar{A}B$		1	1	
AB	X	X 1	X 1	X
$A\bar{B}$			X	X

without  $\phi \Rightarrow \bar{A}\bar{B}D$

with  $\phi \Rightarrow BD$

eg.

$$f = \sum m(1, 2, 3, 5) + \phi(4) \leftarrow$$

$$f = \prod M(0, 6, 7) + \phi(4). \leftarrow$$

 $\bar{B}\bar{C} \quad \bar{B}C \quad BC \quad B\bar{C}$ 

$\bar{A}$	0	(1)	1	(1) 3	1	2
A	X	1 4	1 5	7	6	

eg.  $f = \sum m(1, 2) + \phi(3, 5, 7)$

$\bar{B}\bar{C}$     $\bar{B}C$     $BC$     $B\bar{C}$

$\bar{A}$	0	1	1	0
A	4	1	$x_5$	$x_7$

$$f = \bar{A}B + C$$

### # Variable-Entered-Mapping :-

①  $f = \bar{A}\bar{B}CD + \bar{A}BC\bar{D} + \bar{A}BCD + \bar{A}B\bar{C}\bar{D} + ABCD + A\bar{B}\bar{C}D$

$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$
$\bar{A}$	0	D	$D + D_3$
A	4	5	D

$$f = \bar{A}CD + \bar{A}B\bar{D} + BCD + A\bar{B}\bar{C}D$$

②

$\bar{B}$    B

$\bar{A}$	C	C
A		

$$f = \bar{A}\bar{B}C + \bar{A}BC$$

$$f = \bar{A}C$$

### # Steps involved in VEM method :-

Step: ① Keep all variable position as 0s and get the simplified eqn.

Step: ② Select any variable and keep 1 in it and other variables should be kept at 0.

Step: ③ In the place of given 1s keep don't cares (X)

and get the simplified egn.

Step:4 Repeat the above process with selection of other variables and get the simplified forms.

	$\bar{y}\bar{z}$	$\bar{y}z$	$yz$	$y\bar{z}$
$\bar{x}$	0 0	A 1	1 3	$\bar{A} \bar{a}$
x	B 4	0 5	x 7	C 6

	$\bar{y}\bar{z}$	$\bar{y}z$	$yz$	$y\bar{z}$
$\bar{x}$	0	0	1	0
x	0	0	x	0

→ so need to consider in final grouping ]

	$\bar{y}\bar{z}$	$\bar{y}z$	$yz$	$y\bar{z}$
$\bar{x}$	0	1	x	0
x	0	0	x	0

	$\bar{y}\bar{z}$	$\bar{y}z$	$yz$	$y\bar{z}$
$\bar{x}$	0	0	x	0
x	0	0	x	0

	$\bar{y}\bar{z}$	$\bar{y}z$	$yz$	$y\bar{z}$
$\bar{x}$	0	0	x	0
x	0	0	x	0

	$\bar{y}\bar{z}$	$\bar{y}z$	$yz$	$y\bar{z}$
$\bar{x}$	0	0	x	1
x	0	0	x	0

$$f = \underline{yz} + A\bar{x}\bar{y} + Bx\bar{y}\bar{z} + Cxy + \bar{A}\bar{x}\bar{y}$$

↳ Redundant

B    B

eg.	<u>A</u>		<u>I</u>	
	<u>A</u>		<u>C</u>	

B    B

method : step : ①  $\bar{A} \quad | \quad I$        $f = \bar{A}B$

A    O

Step : ②

<u>A</u>		<u>X</u>	
<u>A</u>		<u>I</u>	

$f = CB$

$f = \bar{A}B + CB$

5-variable K-MAP :-

Sunday

A $\bar{B}\bar{C}$  will form OctetA

	<u>D</u> <u>E</u>	<u>D</u> <u>E</u>	<u>D</u> <u>E</u>	<u>D</u> <u>E</u>		<u>D</u> <u>E</u>	<u>D</u> <u>E</u>	<u>D</u> <u>E</u>	<u>D</u> <u>E</u>
<u><math>\bar{B}\bar{C}</math></u>	1	0	1	1	1	9			
<u><math>\bar{B}C</math></u>	4	5	7	1	6				
<u><math>B\bar{C}</math></u>	13	(1) 13	1	15	1	14			
<u><math>BC</math></u>	8	9	10	1	10				

BCE

{will not form Octet as there are  
more than one literal changes}ADE    ADEIdentification of Prime and Essential Prime by  
Tabulation Method :

- First see (x) having single (x) in the columns (see which column has single (x)).
- Secondly see which Row has single having that Round (X).
- Then that row will be EPI and other Remaining will be PI.

\* Mcclusky Method (Tabulation - Method):

Step: ① Take the given CKF in form of SOP.

Step: ② Form the different groups with no. of 1s.

Step: ③ compare the successive groups for 1-literal change.

Step: ④ Repeat the above process until we get 1-literal change.

e.g.  $f = \sum m(0, 1, 3, 7, 8, 9, 11, 15)$

①	Group	Minterm	Variable			
			A	B	C	D
	0	0	0	0	0	0
	1	1	0	0	0	1
	8	-	1	0	0	0
1	2	3	0	0	1	1
		9	1	0	0	1
3		7	0	1	1	1
		11	1	0	1	1
4		15	1	1	1	1

(2) Group

Minterm

Variables

A B C D

0

0, 1, 8

0 0 0 -

0, 8

- 0 0 0

1

1, 3

0 0 - 1

1, 9

- 0 0 1

8, 9

1 0 0 - 1

2

3, 7

0 - 1 - 1

3, 11

- 0 1 1

9, 11

1 0 - 1

3

7, 15

- 1 1 1

11, 15

1 - 1 1

(3)

Group

Minterm

Variable

A B C D

0

0, 1, 8, 9

- 0 0 - 1

0, 8, 1, 9

- 0 0 -

1

1, 3, 9, 11

- 0 - 1

1, 9, 3, 11

- 0 - 1

2

3, 7, 11, 15

- - 1 1

3, 11, 7, 15

- - 1 1

$$f = \bar{B}\bar{C} + \bar{B}D + CD$$

0 1, 3, 7 8 9 11 15

 $\begin{matrix} \bar{B} \bar{C} & \textcircled{X} & \times & \textcircled{X} & \times & \times \\ \swarrow \text{EPI} & & & & & \\ \bar{B} D & X & X & & X & X \end{matrix}$ 
 $\begin{matrix} C D & & \textcircled{X} & & X & \textcircled{X} \\ \nwarrow \text{EPI} & & & & & \end{matrix}$

**\*\* Note:**

- If any pair (lower-order-grouping) is not coming under <sup>[Higher-Order-grouping]</sup> Quad-grouping, then those numbers won't be carried the <sup>4m</sup> second table to next table.

But their concern minterm also should be present in the final expression.

- Tabulation method for DON'T CARE COND:-

If the boolean expr. consists of a DON'T CARE Treat them as usual minterms and follow the usual procedure to obtain the set of prime implicants.

(Note: In final implicant chart all the don't care minterms are omitted)

① eg.  $f = \sum m(6, 7, 8, 9) + \phi(10, 11, 13, 12, 14, 15)$

$\Rightarrow$  In final chart we should have :

	6, 7, 8, 9	...
--	------------	-----

Prob: ②  $f = \sum m(0, 1, 6, 7, 8, 9, 13, 14, 15)$ .

Prob: ③  $f = \sum m(1, 2, 3, 5, 6, 7, 8, 9, 12, 13, 15)$ .

Prob: ④  $f = \pi M(2, 3, 8, 12, 13) \cdot \phi(10, 14)$ .

\* \* Procedure: Tab" Method for POS is exactly same as that for SOP form.

- o - Treat the Maxterms as if they are the Minterms and complete the process.
- o - While writing the minimal expression in POS-form treat the non-complemented variable as '0' and complemented variable as '1' and write the term in sum form (Max-term) and final term in Product

Prob: 5  $f = \bar{\Sigma} M(0, 1, 4, 5, 9, 11, 13, 15, 16, 17, 25, 27, 28, 29, 31)$   
 $\phi(20, 21, 22, 30)$ .

\* \* This Tab" Method is playing an imp. role for the ckt's involving higher no. of variables. Better than that of Karnaugh-Map.

# HAZARDS :- Bcoz of the different propagation-delays at a diff. parts of ckt the O/P of the ckt may not be accurate for small interval of time, known as Hazards in ckt.

- o - It can be eliminated by using redundant groups.

Glitch :- Bcoz of the hazards in ckt, for a temporary interval of time there'll be wrong O/P, known as glitch in ckt.

e.g..  $f = \Sigma m(1, 3, 6, 7)$

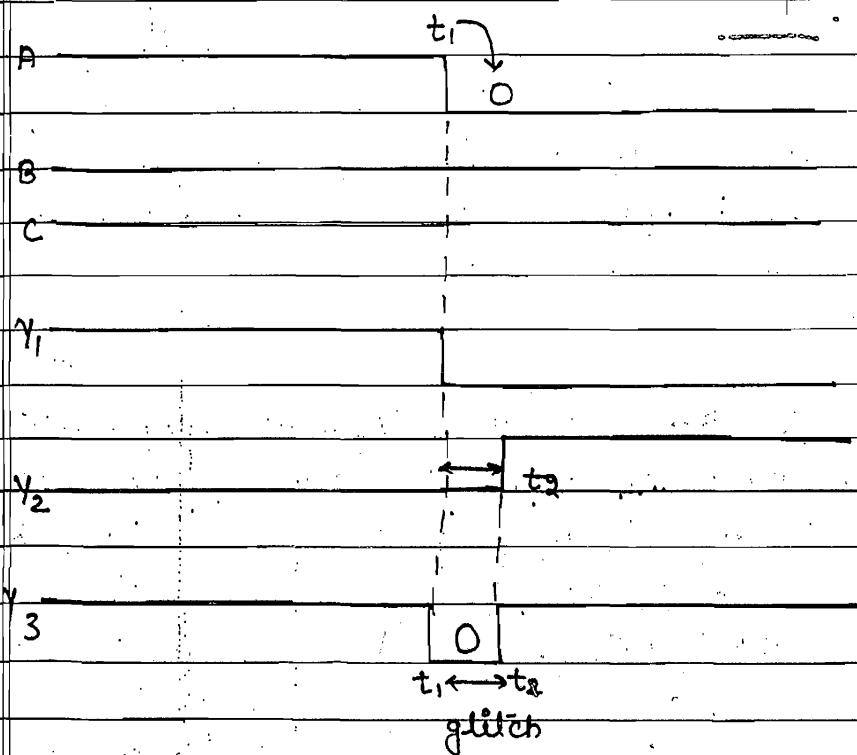
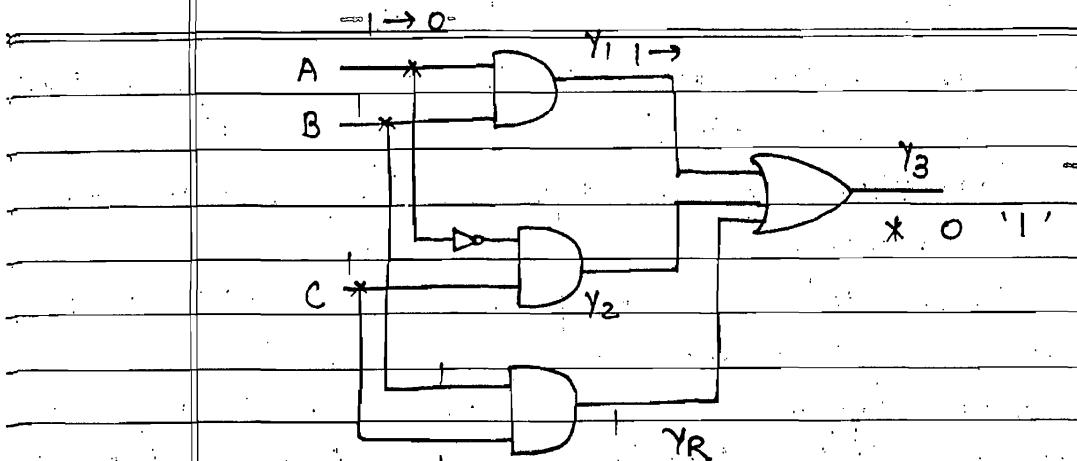
$\bar{B}\bar{C}$     $\bar{B}C$     $BC$     $B\bar{C}$

$\bar{A}$		(1)	(1)	
A			(1)	(1)

$f = \bar{A}\bar{C} + AB$ . — x

$f = \bar{A}\bar{C} + AB + BC$  — ✓

Redundant group



Types:

- ① static '1'
- ② static '0'
- ③ Dynamic

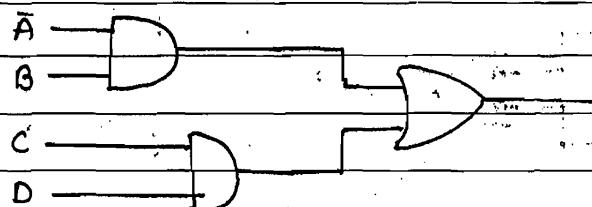
# Implementation of Boolean-functions by Using NAND,  
NOR - gates only :-

(1)

$$f = A'B + CD$$

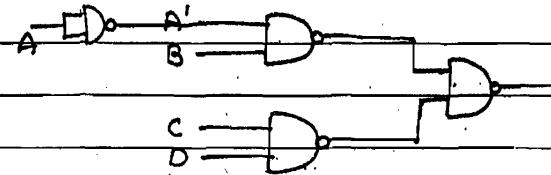
(NAND)

AND-OR



$$f = \overline{\overline{A'B} + \overline{CD}}$$

$$f = \overline{\overline{A'B} \cdot \overline{CD}}$$

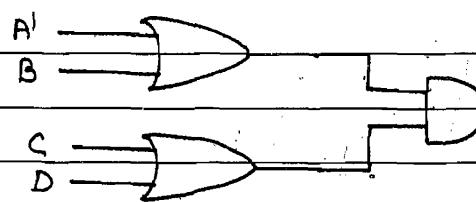


(2)

$$f = (A'+B) \cdot (C+D)$$

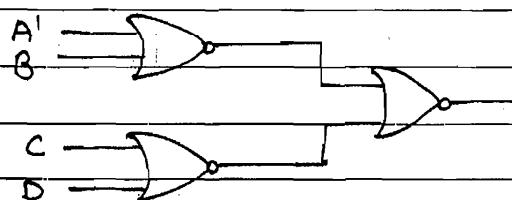
(NOR)

OR-AND



$$f = (A'+B) \cdot (C+D)$$

$$f = \overline{(A'+B)} + \overline{(C+D)}$$

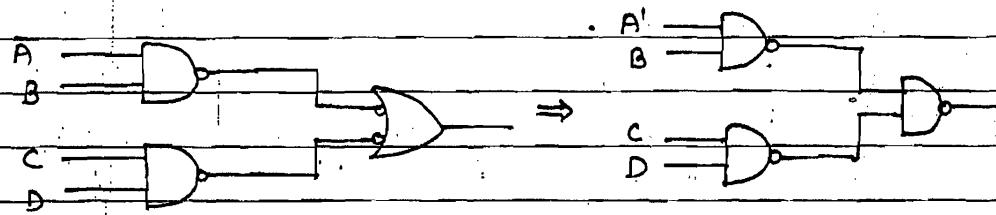


Method : ②

Step: ① AND (NAND)

Step: ② OR (Bubbled OR = NAND)

Step: ③ NOT (NAND)

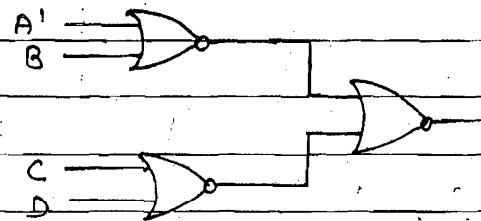


NOR :

Step : ① OR (NOR)

Step : ② AND (Bubbled AND = NOR)

Step : ③ NOT (NOR)



"Active - low":SR-latch  $\Rightarrow$ 

And-gate connect hai isle a CLK hmesha 'high'

Gated-SR-latch  $\Rightarrow$ 

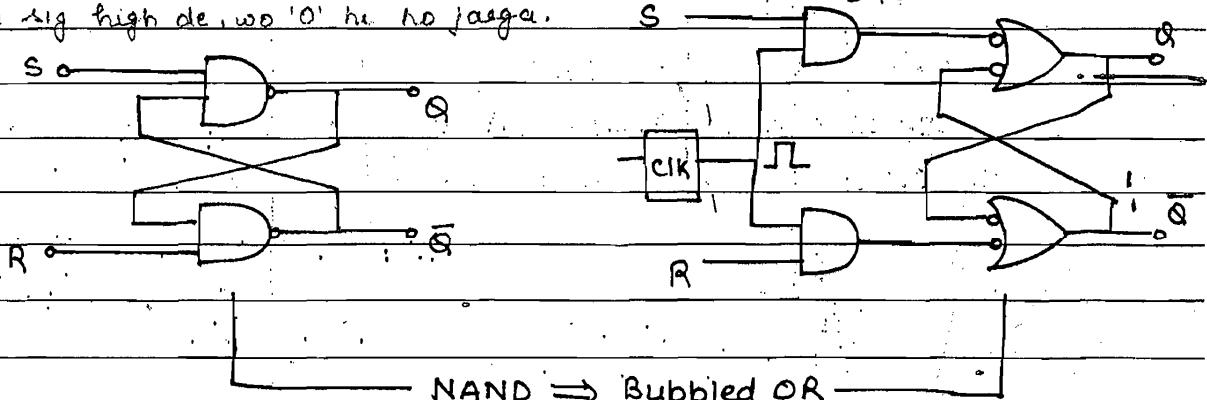
Nahega aur isie CLK ko hmesha high ke rakh-

clocked-SR-latch  $\Rightarrow$ 

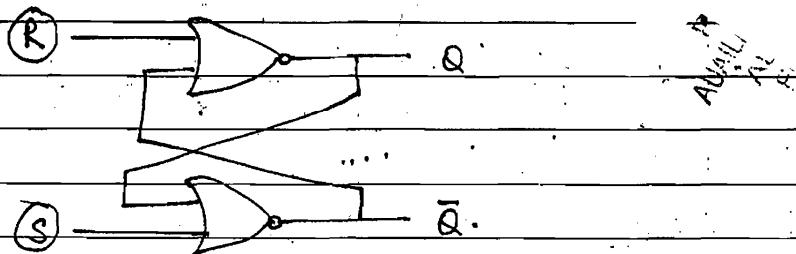
enge qki agar use ke low kar denge to kitna

te nig high de wo '0' hi ho jaega.

AND Gate

Table for Active - low latch :-

CLK	S	R	$Q^+$
Low	1	1	NC
High	1	0	0 (Reset)
Low	0	1	1 (Set)
High	0	0	X DON'T CARE

Active - High latch :-

CLK	S	R	$Q^+$
Low	0	0	NC
High	1	0	1 Set
Low	0	1	0 Reset
High	1	1	X DON'T CARE

## # Sequential circuits :-

Logic gates with feedback connection are known as sequential circuits.

e.g. latches :-

Latches are two - types -

1. NAND-GATE LATCH [Active-low] fig(a)
2. NOR-GATE LATCH [Active-high] fig(b)

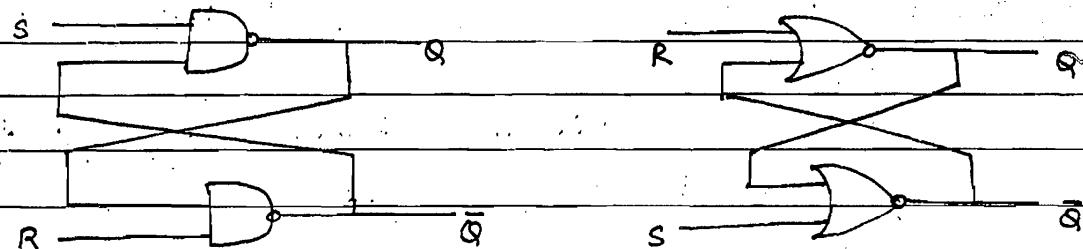


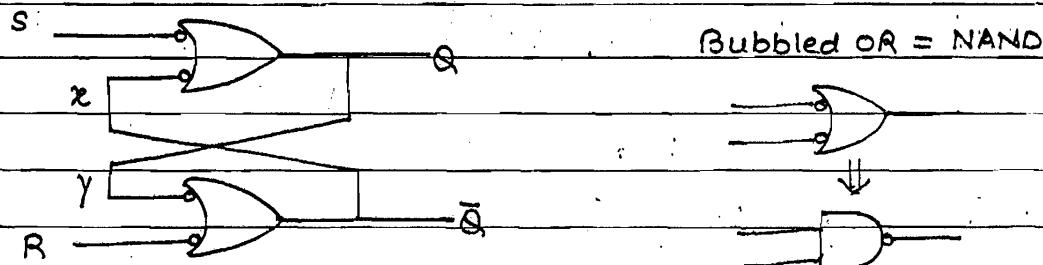
fig (a)

fig (b)

$S \rightarrow$  Set

$R \rightarrow$  Reset

SR-Latch operations : (ACTIVE-LOW)



Case: 1  $S=1, R=1$

If initially  $Q=0$ , then the feedback input will be  $x=1$  &  $y=0$ . Then the next state will be  $Q^+=0$

→ If initially  $Q=1$ , then the feedback s/p will be  $x=0$  &  $y=1$ . Then the output will be  $Q^+=1$   
So by the above expression, we can say when

when  $x=1, R=1$  we get  $\rightarrow Q^+ = NC$

Case: 2 when  $S=1, R=0 \Rightarrow$

→ If initially  $Q=0$ , then the feedback OIP will be  $x=1 \& y=0$  and we get the next-state  $Q^+ = 0$

→ If initially  $Q=1$ , then the feedback OIP will be  $x=0 \& y=1$  so we get  $\bar{Q} = 1 = Q$  only but it is the intermediate state and it is not stable.

so feedback OIP must be operated until we get the stable OIPs. By observations, we can say that when  $S=1 \& R=0$  we get  $Q^+ = 0$ , the Reset condition.

Case: 3 when  $S=0, R=1 \Rightarrow$

The same above process should be conduct for the stable OIPs. and by observations, we can say that when  $S=0, R=1$  we get  $Q^+ = 1$ , the Set condition.

Case: 4 when  $S=0, R=0 \Rightarrow$

By observation we get  $Q^+ = \bar{Q} = 1$  only which is the useless condition known as DON'T CARE condition.

\*\* Note:- The S-R latch can be controlled with the help of a switch known as Gated S-R Latch.

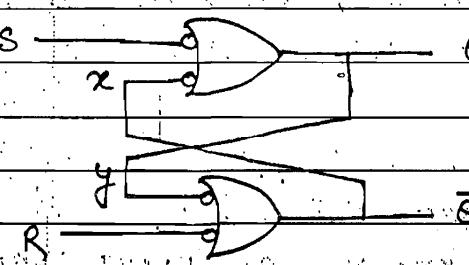
- S-R latch controlling can also be done by using clk generator. Then it is known as clocked S-R latch (or) clocked S-R flip-flop.

Saturday

Sequential Circuit  $\Rightarrow$  In the sequential ckt's present O/P not only depends on the present GIP's it also depends on the past output.

- o- logic-gates with feedback connections behaves as sequential ckt's
- e.g. latches
- o- Latches are of two types-
  1. NAND-Gate (Active-Low)
  2. NOR-Gate (Active-High)

\* Operation of S-R F.F.  $\Rightarrow$  (Active-Low)



Case: 1.  $S=1, R=1$

o- when initially ' $Q=0$ ', then the feedback GIP's will be  $x=0$  and  $y=1$  then you'll get the next state  $Q^+ = 0$

if initially ' $Q=1$ ', then the feedback GIP's will be  $x=0$ ,  $y=1$  then we'll get next-state  $Q^+ = 1$

so by the observation we can say that when  $S=1, R=1$   
we get  $Q^+ = \text{No-change}$ .

case: 2  $S=0, R=1$

- If initially ' $Q=0$ ',  $x=1, y=0$ , we'll get the next state

the feedback OIP's are operated, we get  $Q=\bar{Q}=1$ , only but is not stable OIP's so this is the intermediate state only so the feedback OIP's should be operated until we get the stable state then we get  $Q^+=1$  so by observation we can say  $S=0, R=1$ , we get  $Q^+=1$  [the SET-CONDITION].

Case: 3  $S=1, R=0$

- In the same above manner, the procedure should be conducted until we get the stable OIP. Feedback OIP's should be operated for both cases of initially  $Q=0$  and initially  $Q=1$  and by the observation we can say that when  $S=1, R=0$  we get next state  $Q^+=0$ , Reset-condition.

Case: 4  $S=0, R=0$

In this case by the observation we can say that  $Q^+=\bar{Q}^+=1$  only which is an unused condition known as DON'T CARE COND.

- S-R latch can be control by using a switch known as gated SR-latch.

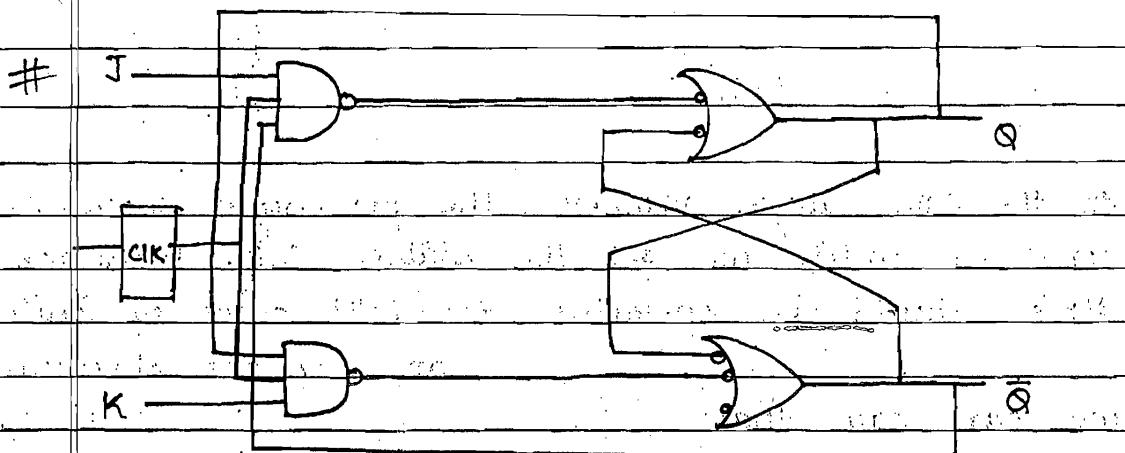
\*\* The controlling can also be done by using clock-generating known as clocked S-R FF.

X J-K Flip-Flop :- It is the modification of S-R F.F. which consists of external feedback also.

when  $J=1, K=1$ , we get the next-state  $Q^+ = \bar{Q}$  [the : (TOGGLE - CONDITION)]

\* Internal feedback should be operated until we get the stable output.

\* External feedback should be operated only 1 time for single clock-pulse.



C1K      J      K       $Q^+$

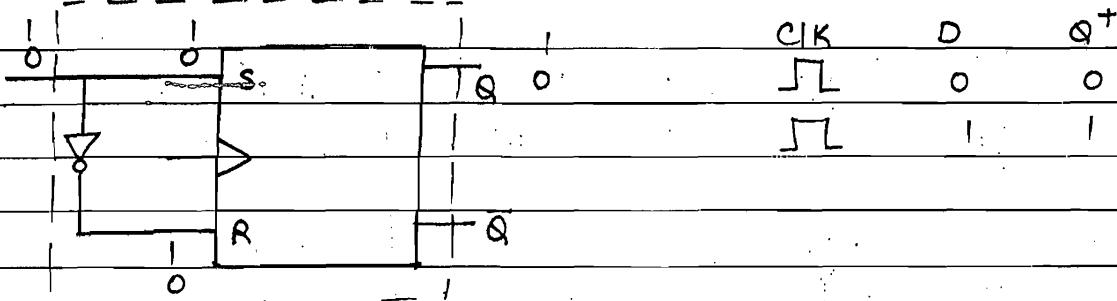
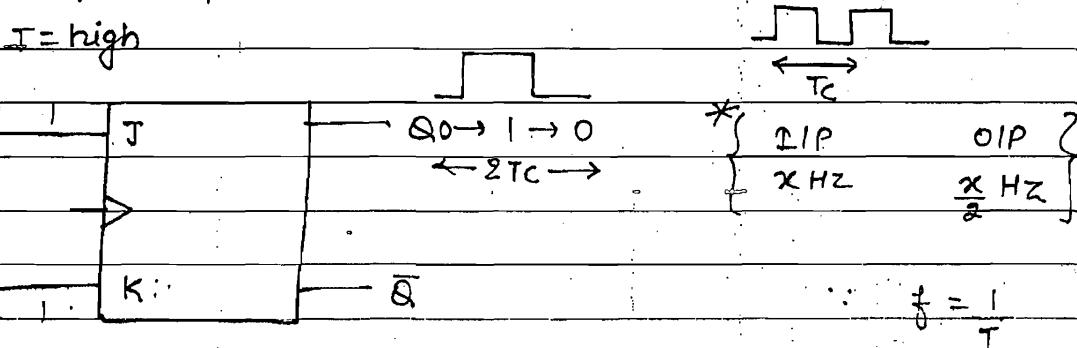
1      0      0      NC

1      0      1      Set

0      1      0      Reset

0      1      1       $\bar{Q}$  (Toggle)

\* Note: By using NOR-gate also J-K-FF can be design But I should have a feedback along with it from  $\bar{Q}$  only."

# D-Flip-Flop :-d. T-Flip-Flop :- $T = \text{high}$ 

$\Rightarrow$  If  $x$  Hz is i/p frequency of T-FF, then the o/p freq. will be  $\frac{x}{2}$  Hz.

Characteristics equations of J-K FF :-

Q	J	K	Q <sup>+</sup>		J'K'	J'K	JK	JK'
0	0	0	0	NC	0	1	1	0
0	0	1	0	Reset	0	0	0	1
0	1	0	1	Set	0	1	1	0
0	1	1	1	Q Toggle	1	0	0	1
1	0	0	1					
1	0	1	0					
1	1	0	1					
1	1	1	0					

$Q^+ = JQ' + K'Q$

S' R' S' R SR SR'

# Characteristics of SR:

Q		X	1
Q	1	X	1

Q S R Q<sup>+</sup>

$$\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array} \Rightarrow Q^+ = S + R'Q$$

0 1 0 1 Reset

0 1 1 X set

1 0 0 1

1 0 1 0

1 1 0 1

1 1 1 X

D D

# Char. of D :- FF

Q	1
Q	1

Q D Q<sup>+</sup>

$$\begin{array}{ccc} 0 & 0 & 0 \end{array} \Rightarrow Q^+ = D$$

0 1 1

1 0 0

1 1 1

T' T

# Char. of TFF :-

Q'	1
Q	1

Q T Q<sup>+</sup>

0 0 0

0 1 1

1 0 1

1 1 0

$$Q' = T \oplus Q$$

$$T'Q + TQ'$$

$$\Rightarrow Q' = T'Q + TQ'$$

# Truth-Table: It gives the info about O/P, for a corresponding G/P's.

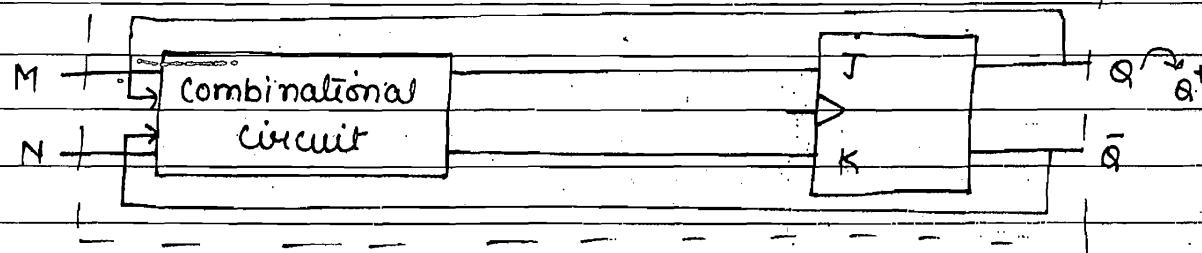
# Transition or Excitation - Table :- It gives the info about of the possible way of G/P's to get a particular O/P.

Excitation - Table: for "Active High"  
(short-cut)

	Q	$Q^+$	J	K			Q	$Q^+$	S	R		
0	0	0	0	0	X	↑	0	0	0	0	X	↑
1	0	1	1	1	X		1	0	1	1	0	
2	1	0	X	1			2	1	0	0	1	
3	1	1	X	0	↓		3	1	1	X	0	↓

	Q	$Q^+$	D			Q	$Q^+$	T		
0	0	0	0			0	0	0	0	
1	0	1	1			1	0	1	1	
2	1	0	0			2	1	0	1	
3	1	1	1			3	1	1	0	

# Designing of F.F.  $\Rightarrow$  CONVERSION OF F.F.  $\Rightarrow$



	Q	$Q^+$	J	K
0	0	0	0	X
0	1	1	1	X
-	0	0	X	1
1	1	1	X	0

Q	M	N	$Q^+$	J	K
0	0	0	0	0	x
0	0	1	1	1	x
0	1	0	1	1	x
0	1	1	1	1	x
1	0	0	0	x	1
1	0	1	1	x	0
1	1	0	0	x	1
1	1	1	x	x	x

M'N' M'N MN MN'

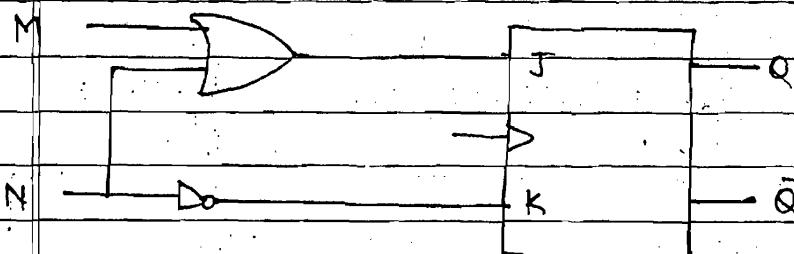
0	0	1	1	3	1	2
0	x	4	x	5	x	7

$$J = M + N$$

MN' M'N MN MN'

0	x	x	x	x
0	1	-	x	1

$$K = N'$$



# Using  $\cdot \overline{S}R - F \cdot F \Rightarrow$

I  
F  
A  
U  
A  
I

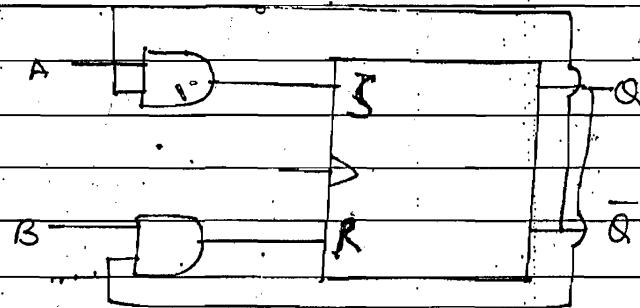
Q	A	I	B	$Q^+$	S	R	0	0	0	x
0	0	0	0	0	0	x	0	1	1	0
0	0	1	0	0	0	x	1	0	0	1
0	1	0	1	1	1	0	1	1	x	0
0	1	1	1	1	1	0	1	1	1	0
1	0	0	1	1	x	0	1	1	1	1
1	0	1	0	0	1	1	1	1	1	1
1	1	0	1	1	x	0	1	1	1	1
1	1	1	0	0	0	1	1	1	1	1

$\bar{Q}$	$S'R'$	$S'R$	$SR$	$SR'$
Q	0	*	(13)	D <sub>2</sub>
Q	X 4	5	*	6

$$S = A\bar{Q}' \quad | \quad S = JQ'$$

$\bar{Q}$	$S'R'$	$S'R$	$SR$	$SR'$
Q	X 0	1	3	4
Q	4	15	17	6

$$R = BQ \quad | \quad R = KQ$$



H.W.

$$\begin{aligned} Q.1 \quad S-R &\Rightarrow JK \\ &\Rightarrow T \\ &\Rightarrow D \end{aligned}$$

$$\begin{aligned} Q.2 \quad J-K &\Rightarrow SR \\ &\Rightarrow T \\ &\Rightarrow D \end{aligned}$$

$$\begin{aligned} Q.3 \quad D &\Rightarrow S-R \\ &\Rightarrow J-K \\ &\Rightarrow T \end{aligned}$$

$$\begin{aligned} Q.4 \quad T &\Rightarrow S-R \\ &\Rightarrow J-K \\ &\Rightarrow D \end{aligned}$$

### Race - Around - Problem (Repetition)

Repetition of Toggle for single CLK-pulse at Q1P is known as Race - Around Problem.

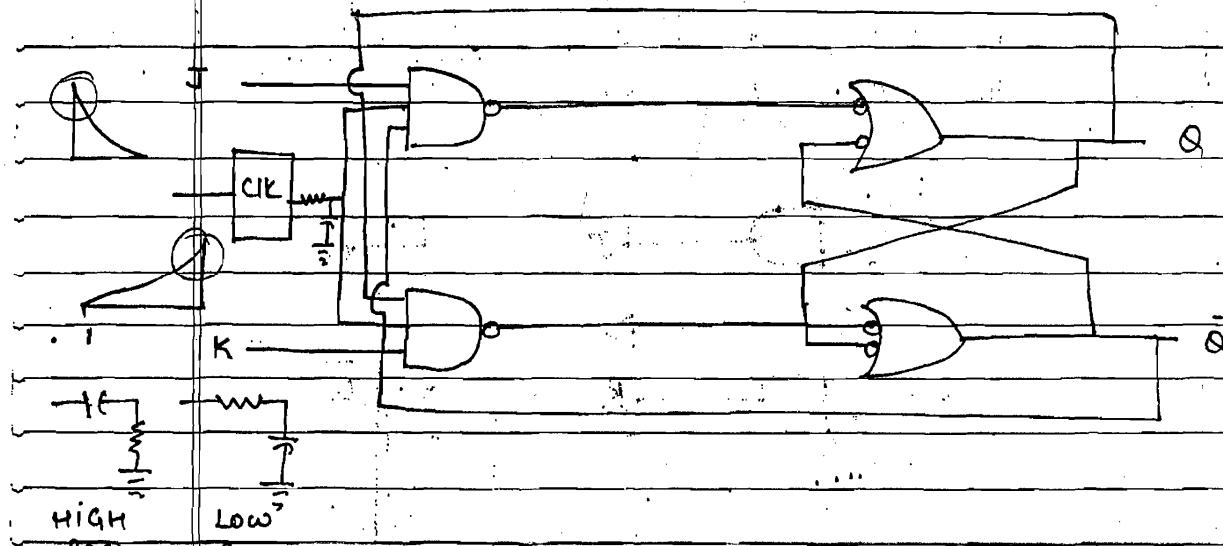
Reason: Q1P occurrence time Q1P CLK-pulse is in the ON-state only which causes one more pulse triggering - leads to repetition of Toggle known as RACE AROUND PROB.

Methods to avoid Prob. =

1. By reducing pulse width.
2. By using edge Triggering.

Note: Edge-Trigg. is two types -

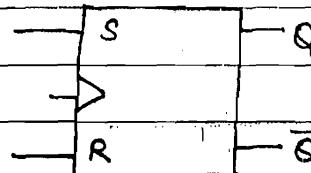
- A. Rising Edge Trigg. [Positive Edge Trigg.]
- B. Falling Edge Trigg. [Negative Edge Trigg.]



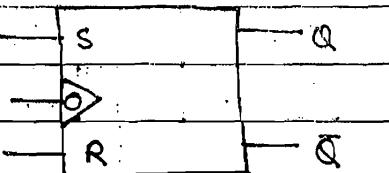
\* 3. By using Master slave flip-flop, Race-around problem can be avoided.

### TIMING DIAGRAMS :

Negative Trigg.

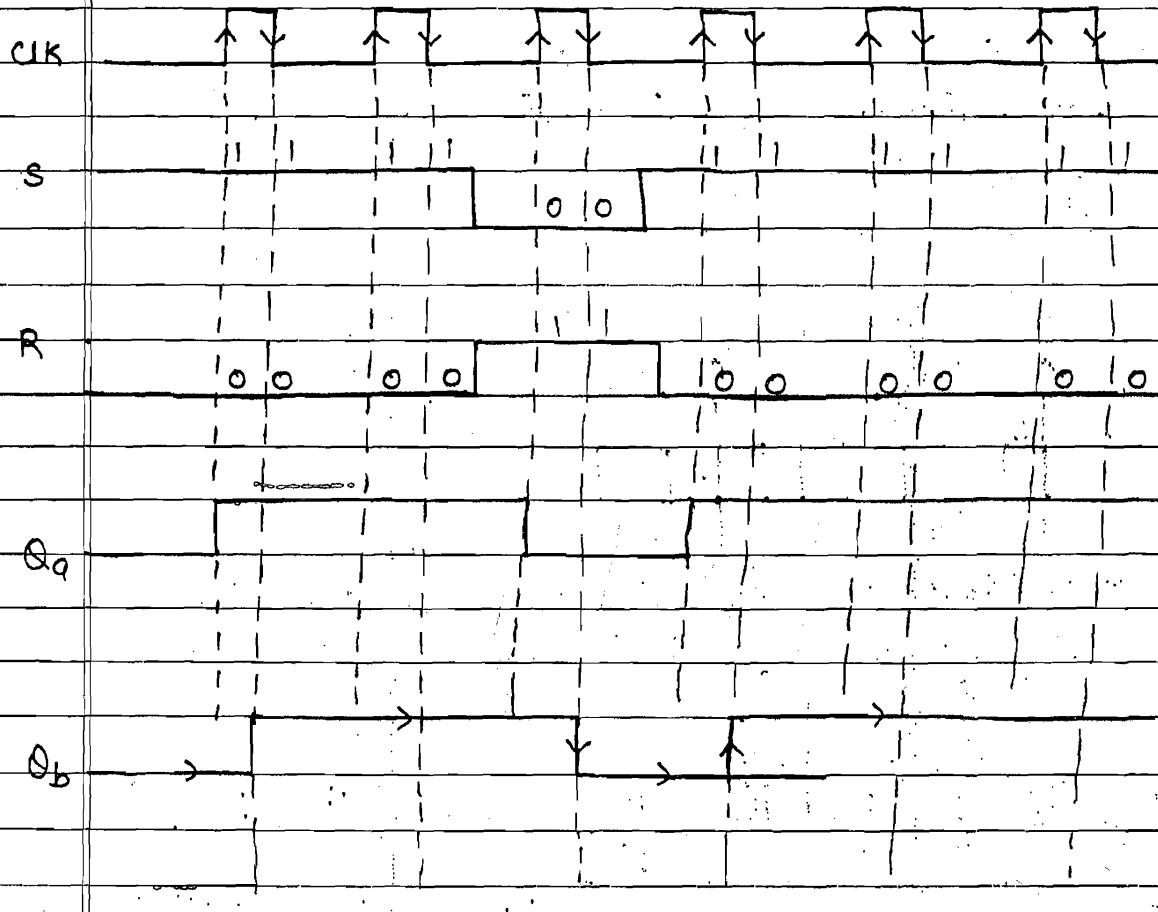


(a)

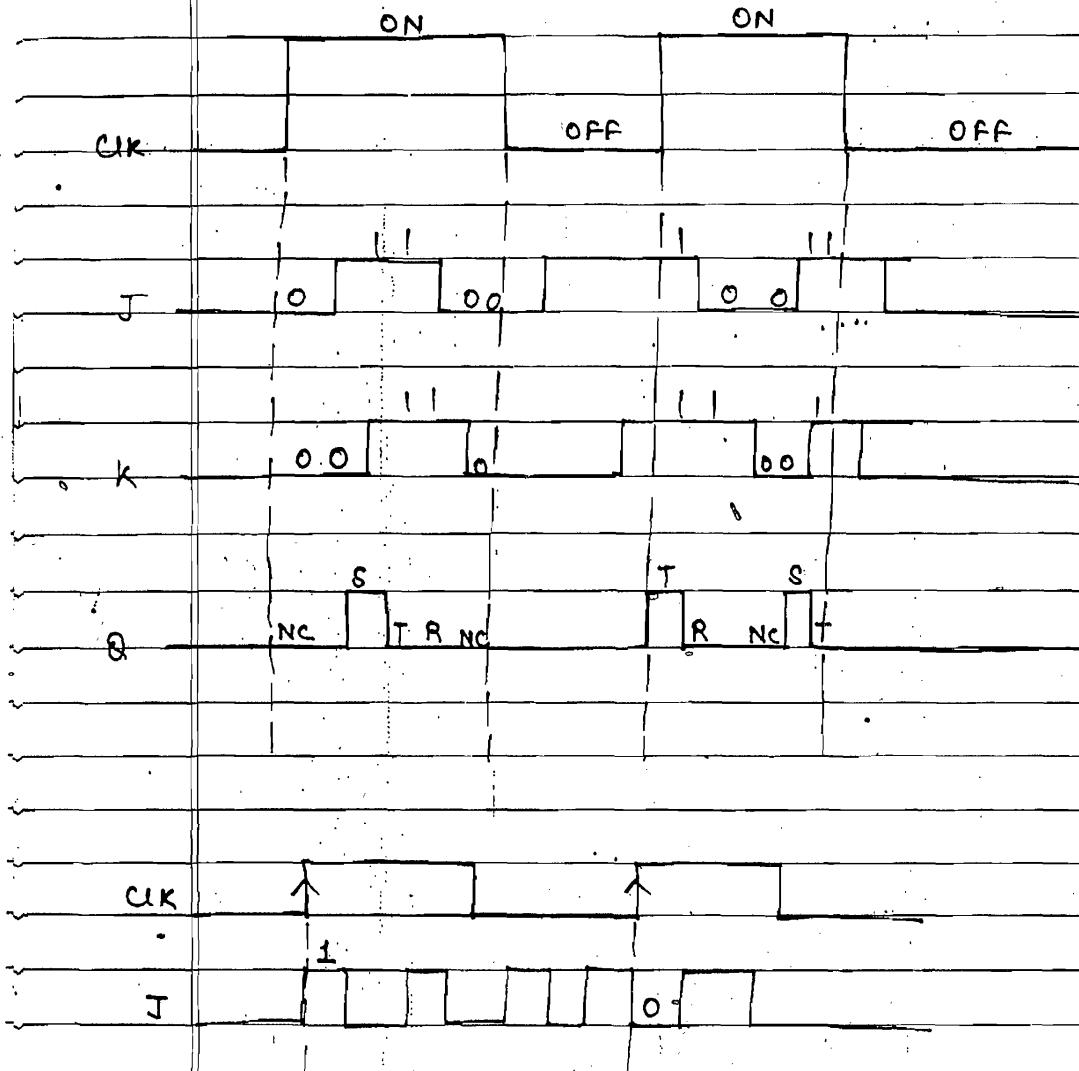


(b)

+ 've Triggered



Pulse-Trigg.  $\Rightarrow$



+ve Trigg. is not deciding the next state. what actually next state is that is going to be set.

	$\bar{B}C$	$\bar{B}C$	$Bc$	$\bar{B}C$
Ques: ①	A	I	D	E
	A	I	I	D

	$\bar{B}C$	$\bar{B}C$	$Bc$	$\bar{B}C$
Ques: ②	A	E	I	F
	A	D		F

$$f = \bar{A}C + A\bar{B}\bar{D} + BC\bar{E} + \bar{A}\bar{B}D + A\bar{C}\bar{E} + \bar{A}\bar{B}\bar{C}$$

classmate  
53

Date \_\_\_\_\_  
Page \_\_\_\_\_

A	I	0	D	E	3	E	2
A	4	1	S	1	7	D	6

$\bar{B}\bar{C}$   $\bar{B}C$   $BC$   $B\bar{C}$

A	1	0	0	3	0	2
A	4	1	5	1	7	0

$$\Rightarrow AC + \bar{A}\bar{B}\bar{C}$$

A	X	1	0	3	0	2
A	4	X	5	X	7	0

$$\Rightarrow \bar{A}\bar{B} + \bar{B}A$$

A	X	0	0	1	3	0	2
A	4	X	5	X	7	0	6

$$\Rightarrow \bar{E}BC$$

A	X	0	0	1	3	0	2
A	4	X	5	X	7	0	6

$$\Rightarrow E\bar{A}\bar{B}\bar{C}$$

A	X	0	0	1	3	0	2
A	4	X	5	X	7	1	6

$$\Rightarrow \bar{D}AB$$

A	X	0	0	1	3	0	2
A	4	X	5	X	7	1	6

$$f = \bar{A}C + \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}D + \bar{B}C + \bar{E}G\bar{C} + E\bar{A}\bar{B}\bar{C} + \bar{D}AB$$

A	E	I	F	3	1	2
A	4	D	5	7	F	6

$$\Rightarrow \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}$$

A	0	0	1	0	3	1	2
A	4	0	5	7	0	6	

$$\Rightarrow E\bar{A}\bar{B}$$

A	1	X	D	3	X	2
A	4	0	5	7	D	6

$$\Rightarrow F\bar{A}C$$

A	0	X	1	0	3	X	2
A	4	1	5	7	0	6	

$$\Rightarrow \bar{F}B\bar{C}$$

A	0	X	1	0	3	X	2
A	4	1	5	7	0	6	

$$\Rightarrow \bar{D}\bar{B}\bar{C}$$

A	0	X	1	0	3	X	2
A	4	1	5	7	0	6	

$$\Rightarrow \bar{A}\bar{B}C + (\bar{A}\bar{B}\bar{C}) + E\bar{A}\bar{B} + F\bar{A}C + \bar{F}B\bar{C} + \bar{D}\bar{B}\bar{C}$$

## Sheet Problems

S F N

### Answers

S

## Active-low SR latch

0 1  
1.0 0  
0 0 X

CK 2

3

1

9

Dues: \$

8

## ACTIVE-H

S R Q

0 0 146

० । ०

10

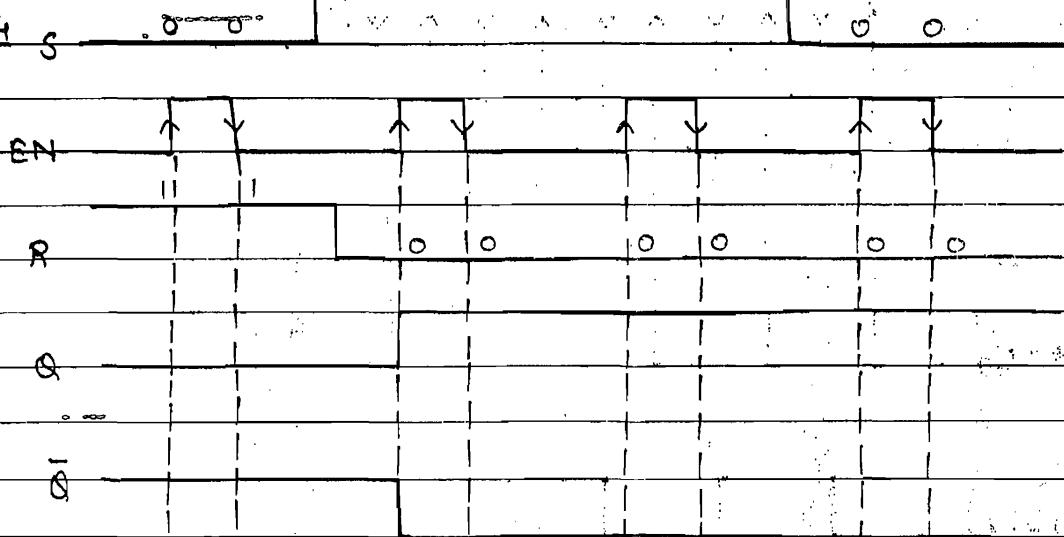
Ques: 3

6

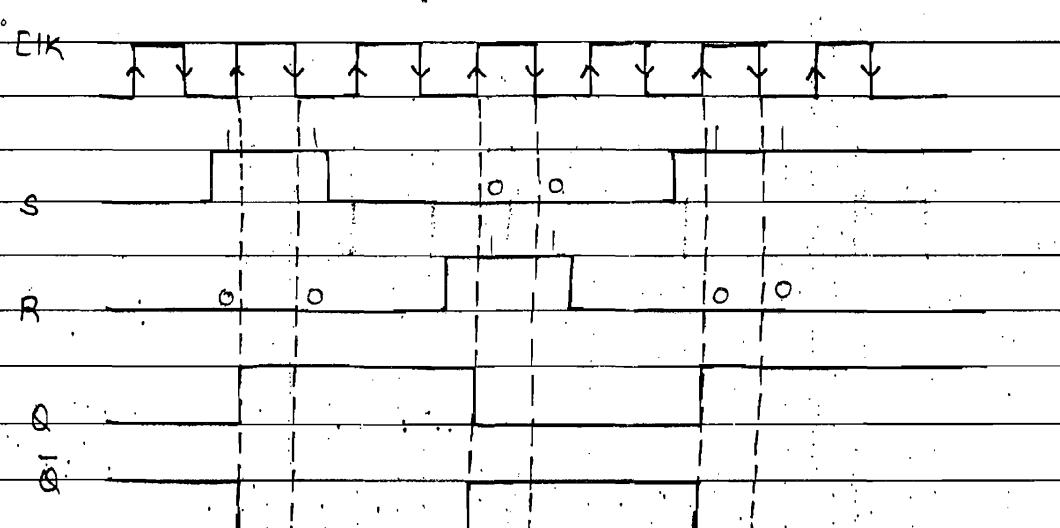
f

8

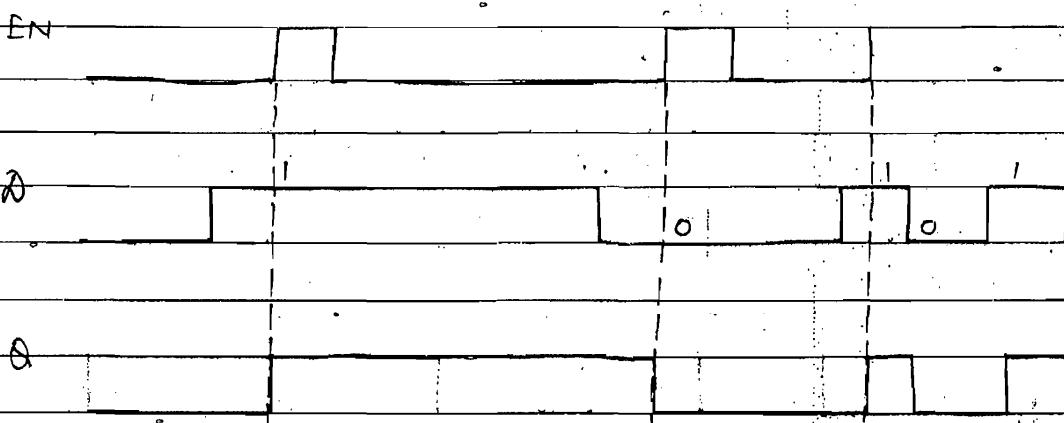
Ques 4



Ques 5



Ques: 54



clk is active  
as it is set as

Ques: 55

Clk: ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓

S

0 0 1 1 0 0

R

0 0 1 1 0 0 0 0

+ve trigger → Q<sub>a</sub>-ve trigger ← Q<sub>b</sub>

Ques: 56

Clk

+ve Trigger

Q

0 0 0 1 0 ... 0 0 0 0 0

57

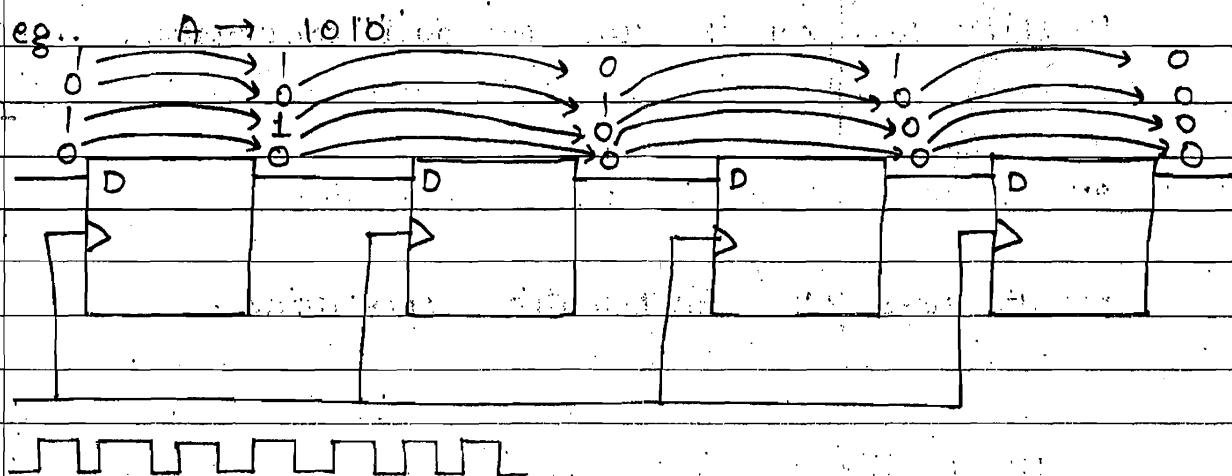
Clk

D

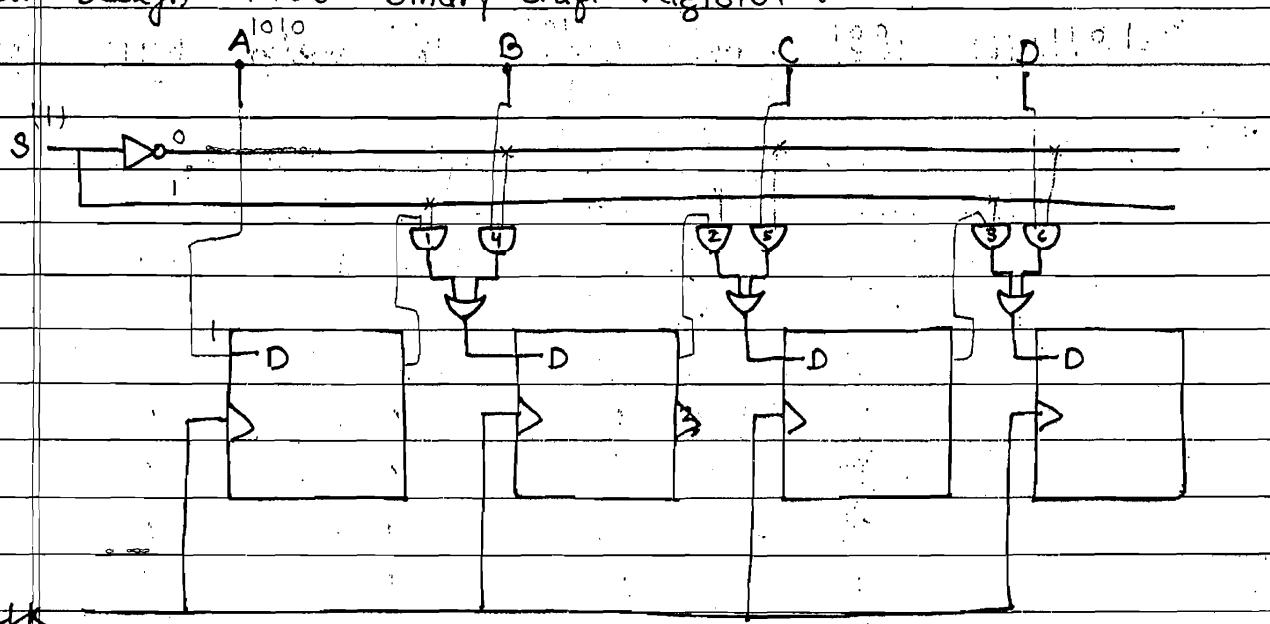
Q

BINARY SHIFT REGISTER :- When F.F. are connected in cascaded manner, binary shift registers. These are 4 - types.

- A). SISO  $\Rightarrow$  n and  $(n-1)$
- B). SIPO  $\Rightarrow$  n and o
- C). PISO  $\Rightarrow$  l and  $(n-1)$ .
- D). PIPO  $\Rightarrow$  l and o



Ques: Design 'PISO' binary shift register :



$S=0$

Parallel io  $\Rightarrow$   
4, 5, 6 v

$S=1$

1, 2, 3 v (operable)

## \* Application of Binary - shift - Register :

i. o- Temporary data storage

o- we can provide time delay :  $\{ N T_c = \Delta t \}$

$$\{ \Delta t = N \}$$

N: no. of f.f.

T<sub>c</sub>: clock period

o- shift Reg. can be used for data conversion. [ SISO  $\rightarrow$  comm sys.]

eg.. SIPO converts serial form of data [ SIPO ]  
into Parallel form.

o- PISO  $\rightarrow$  converts Parallel  $\rightarrow$  Series.

o- It uses for Arithmetic operations.

eg. Left shift Reg. is multiplication by a network

Right- shift Reg. is division by a Network

Note: Right shift having a error of 0.5 for  
Odd no. of f.f.

\* o- Shift Regi. are used to design RING - COUNTER

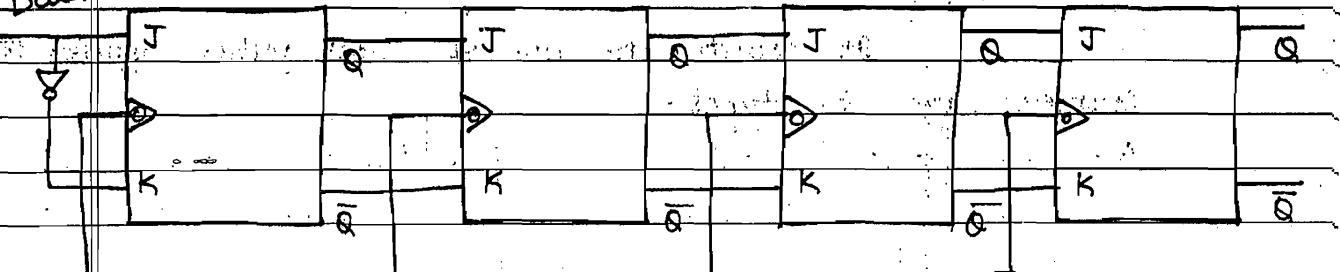
\* IES

Ques: Design 4-Bit BINARY - SHIFT- REGI. By using

J K. F.F. with -'ve edge Trigg. and draw the  
wave form diagrams for all F.F. O/P's for  
given data g/p.

CK	QA	QB	QC	QD
1	1	0	0	0
2	1	1	0	0
3	0	1	1	0
4	1	0	1	1
5	1	1	0	1
6	1	1	1	0
7	1	1	1	1

Data 91P



CLK

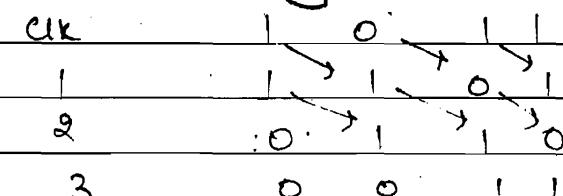
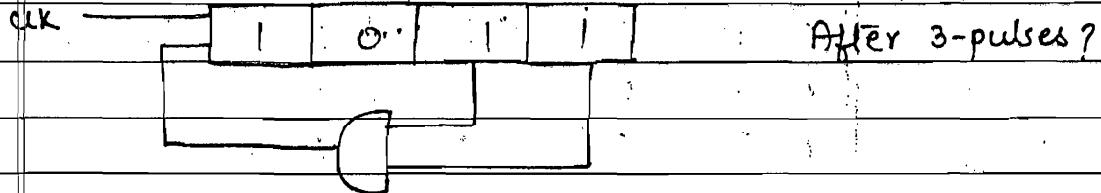
0 1 2 3 4 5 6 7

CLK

data  
91PQ<sub>A</sub>Q<sub>B</sub>Q<sub>C</sub>Q<sub>D</sub>

Bi-directional

Ques: Design a 4-bit Binary shift Reg.



## COUNTERS :

It counts the no. of clock pulses applied to it.

Counters are 2 - types -

A) Synchronous Counter  
(Parallel - counters)

B) Asynchronous Counters  
(Ripple - counters)

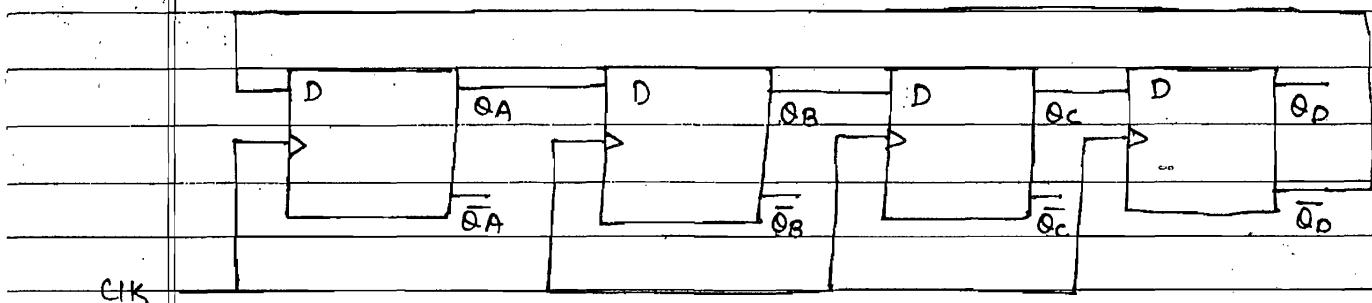
⇒ Ring counters are special category of synchronous C.

Ring C. are two - types -

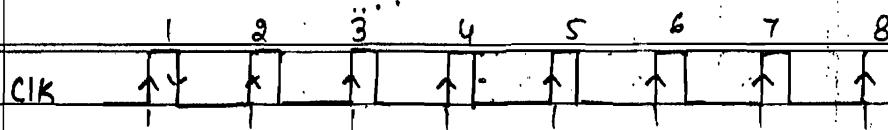
1. Johnson's Ring C. (twisted Ring C.) :-

Feedback is given from complementary QIP of final F.F.

2. Ordinary Ring C.

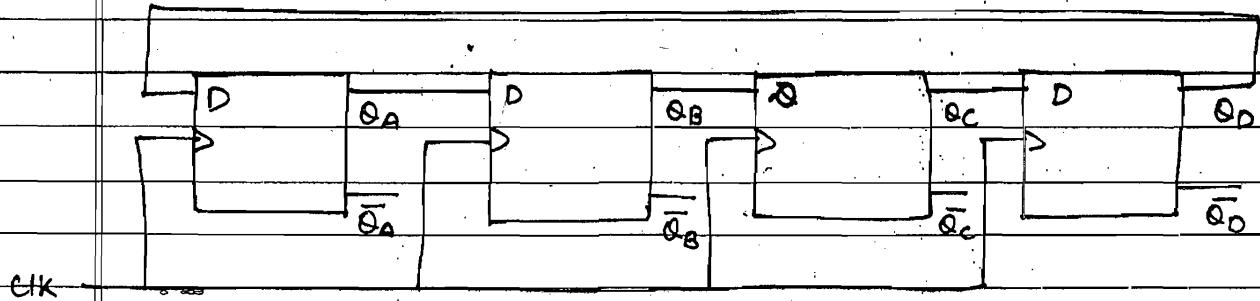


CLK	QA	QB	QC	QD	
1	1	0	0	0	
2	1	1	0	0	
3	1	1	1	0	
4	1	1	1	1	mod 8
5	0	1	1	1	
6	0	0	1	1	
7	0	0	0	1	
8	0	0	0	0	
9	1	0	0	0	X

 $Q_A$  $Q_B$  $Q_C$  $Q_D$ 

$n \rightarrow 2n \rightarrow$  Possible states  
 ↘ CLK pulses

2. Ordinary Ring counter: [not self-start]  
 feedback is given by uncomplementry O/P of final F.F.



CLK : Q<sub>A</sub>    Q<sub>B</sub>    Q<sub>C</sub>    Q<sub>D</sub>

1    0    0    0    0

2    0    0    0    1

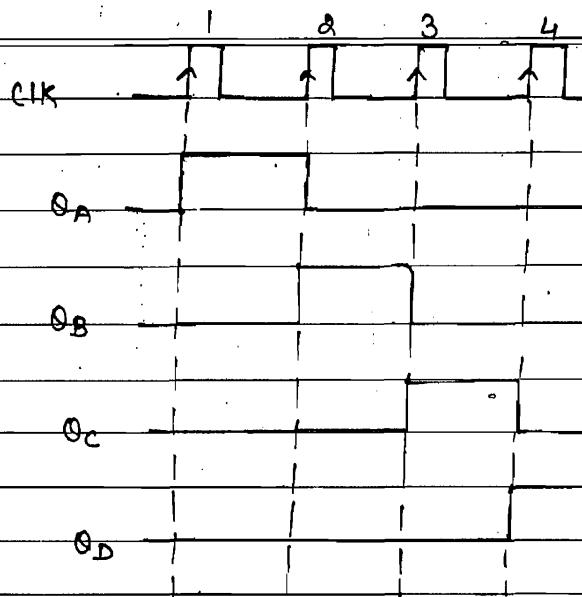
3    0    0    1    0

4    0    1    0    0

5    1    0    0    0

} mod 4

X



$n \rightarrow \textcircled{n}$  → Possible states.  
→ CLK pulses.

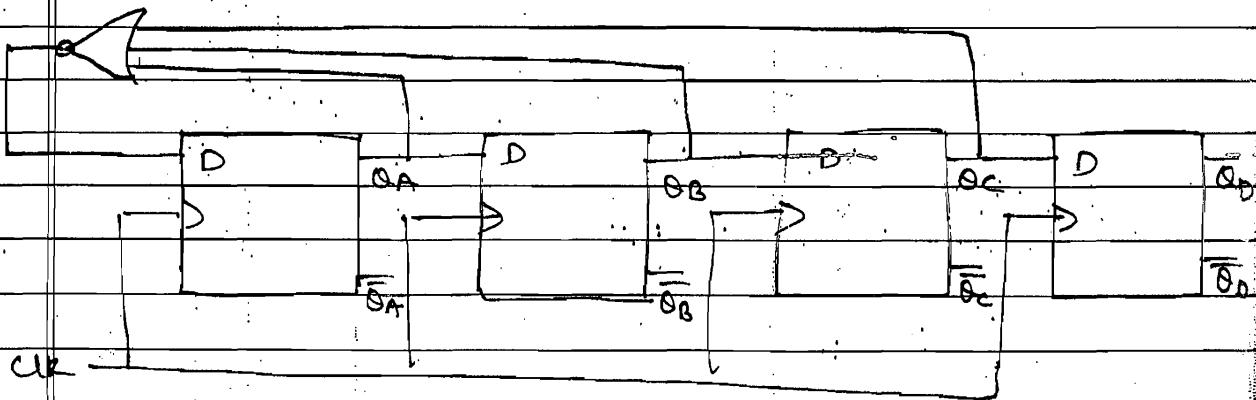
⇒ State identification (Decoding) is easy in RING-C.

#### \* Unused states :

$$\text{Johnson's C} \Rightarrow 2^n - 2n$$

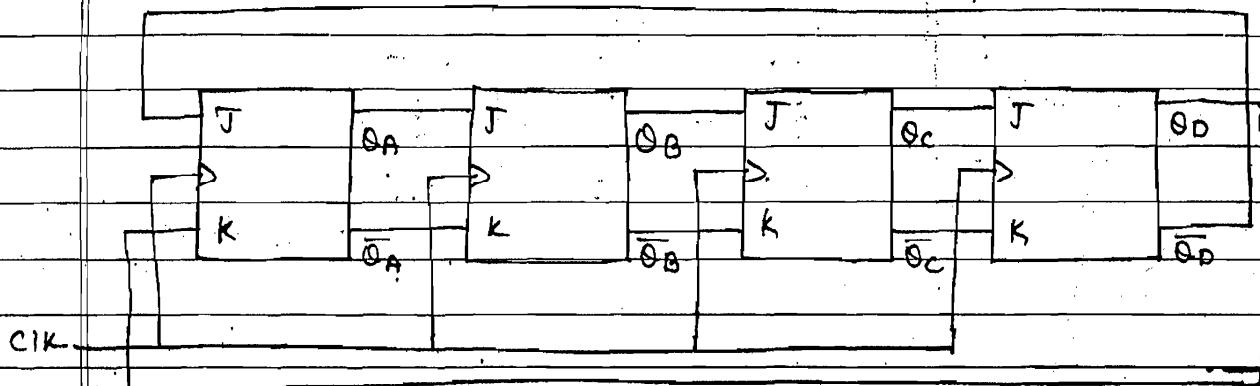
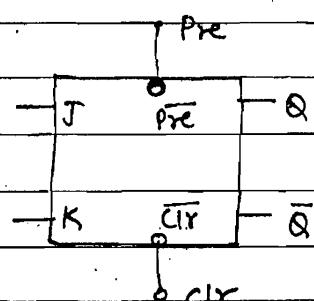
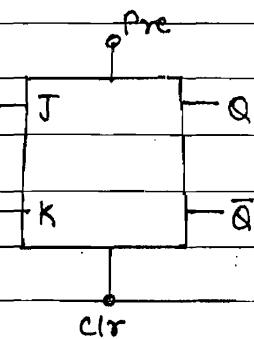
$$\text{RING C} \Rightarrow 2^n - 3^1$$

#### \* SELF-STARTED RING-C ⇒



Johnson's RING C.

- A. Creeping C.  
 B. Walking C.  
 C. Switch Tail C.

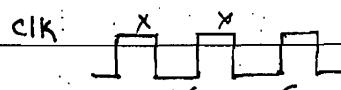
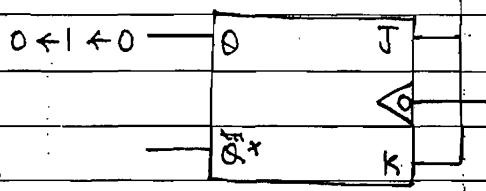
ASYNCHRONOUS INPUTS

Pre	clr	$Q^+$
0	0	ff
0	1	0
1	0	1
1	1	X

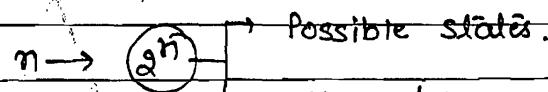
Pre	clr	$Q^+$
1	1	ff
1	0	0
0	1	1
0	0	X Don't Care Unspecified open

-ive trigger:

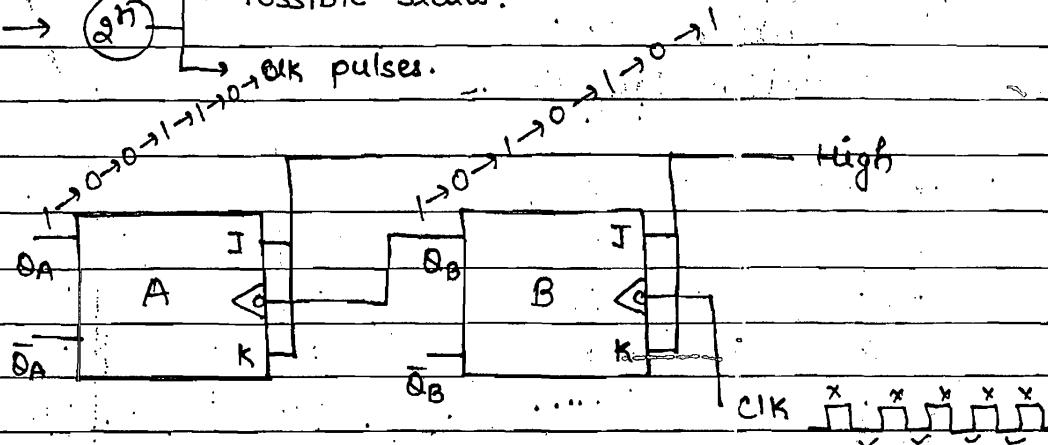
Toggle(1)



# Asyn. C  $\Rightarrow$  F.F. are not triggered simultaneously.



$\rightarrow$  CLK pulses.

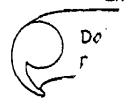


CLK	$Q_A$	$Q_B$	
0	0	0	
1	0	1	{ mod '4'
2	1	0	
3	1	1	
4	0	0	x

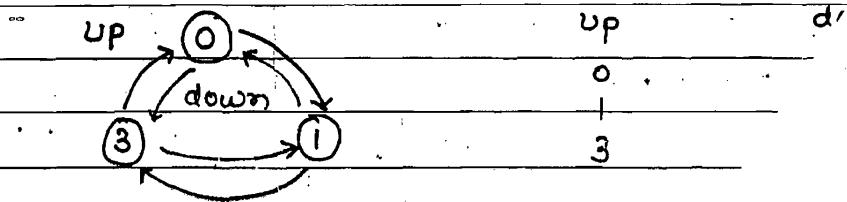
CLK      ↓      ↓      ↓      ↓

$Q_B$       ↑      ↓      ↑      ↓

$Q_A$       0      0      .      .



### Synchronous up-down count - design :-



$y=0$

	Present State		Next State					
	$Q_1$	$Q_0$	$Q_1^+$	$Q_0^+$	$J_0$	$K_0$	$J_1$	$K_1$
up	0	0	0	1	1	x	0	x
	0	1	1	1	x	0	1	x
	1	1	0	0	x	1	x	1

$Q \ Q^+ \ J \ K$

$\bar{Q}_1 \bar{Q}_0 \ \bar{Q}_1 Q_0 \ Q_1 Q_0 \ Q_1 \bar{Q}_0$

0 0 0 x

$\bar{y}$  | 1 0 x 2 x 3 x 4

0 1 1 x

$y$  | 1 4 x 5 x 7 x 6

1 0 x 1

1 1 x 0

$J_0 = 1$

$y=1$

	Present State		Next State					
	$Q_1$	$Q_0$	$Q_1^+$	$Q_0^+$	$J_0$	$K_0$	$J_1$	$K_1$
down	1	1	0	1	x	0	x	1
	0	1	0	0	x	1	0	x
	0	0	1	1	1	x	1	x

$Q \ Q^+ \ J \ K$

$\bar{Q}_1 \bar{Q}_0 \ \bar{Q}_1 Q_0 \ Q_1 Q_0 \ Q_1 \bar{Q}_0$

0 0 0 x

$\bar{y}$  | x 0 1 x

0 1 1 x

$y$  | x 0 1 x

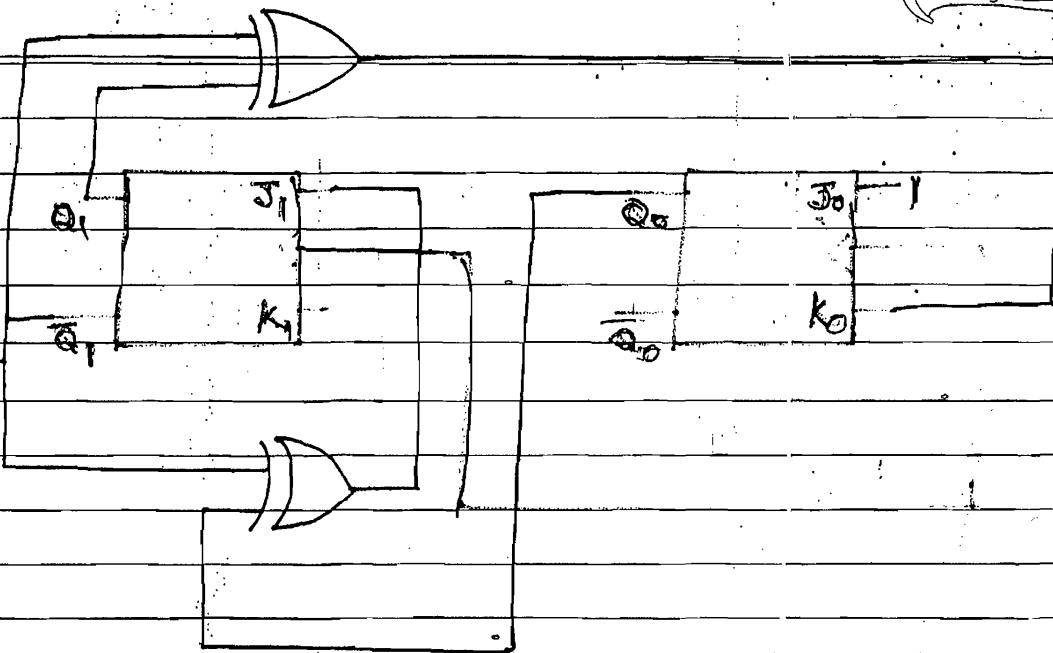
1 0 x 1

1 1 x 0

$y=1$

$K_0 = \bar{y}Q_1 + yQ_0$

$y \oplus Q_1$



$$J_1 = y \oplus Q_0$$

$$K_1 = 1$$

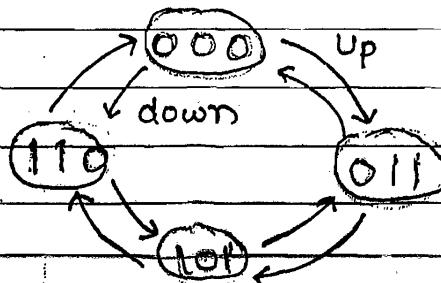
Ques: wanted Sequence By using T-flip-flop

0

3

5

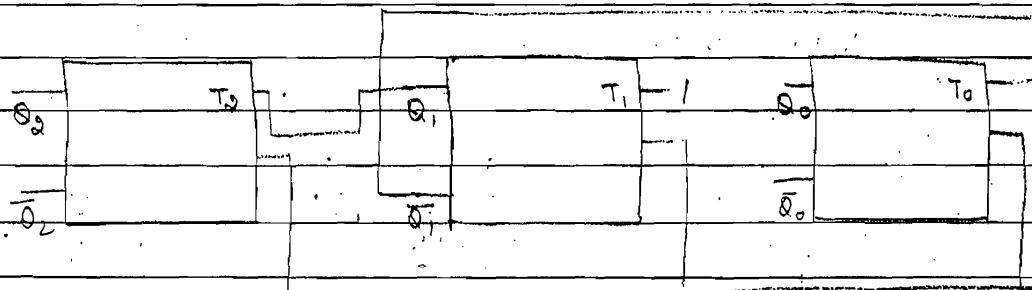
6



Present-state / Next-state						T <sub>0</sub>	T <sub>1</sub>	T <sub>2</sub>
Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub> <sup>+</sup>	Q <sub>1</sub> <sup>+</sup>	Q <sub>0</sub> <sup>+</sup>			
0	0	0	0	1	1	1	1	0
0	1	1	1	0	1	0	1	1
1	0	1	1	1	0	1	1	0
1	1	0	0	0	0	0	1	1
			Q <sub>2</sub> 0	Q <sub>1</sub> 0	Q <sub>0</sub> 0	Q <sub>2</sub> 0	Q <sub>1</sub> 0	Q <sub>0</sub> 0

T <sub>0</sub> = Q <sub>1</sub>	T <sub>0</sub> = Q <sub>1</sub> Y	Y	1	X	0	X		
T <sub>1</sub> = 1		Y	X	1	X	0		
T <sub>2</sub> = Q <sub>1</sub>	T <sub>1</sub> = 1		Q <sub>1</sub> 0	Q <sub>0</sub> 0	Q <sub>1</sub> 0	Q <sub>0</sub> 0	Q <sub>1</sub> 0	Q <sub>0</sub> 0
		Y	0	X	1	X		
	T <sub>2</sub> = Q <sub>1</sub> Y	X	0	X	1			



Q2

Procedure: corrected

Self-corrected counters: Bcoz of sig fluctuations if the counter went to some unwanted state initially and if it is coming back to original sequence (wanted sequence) after some clock pulse then it is known as self corrected counters.

lock-out condition : if the counter is not coming back from unwanted seq.. to wanted seq. it is known as lock out condition.

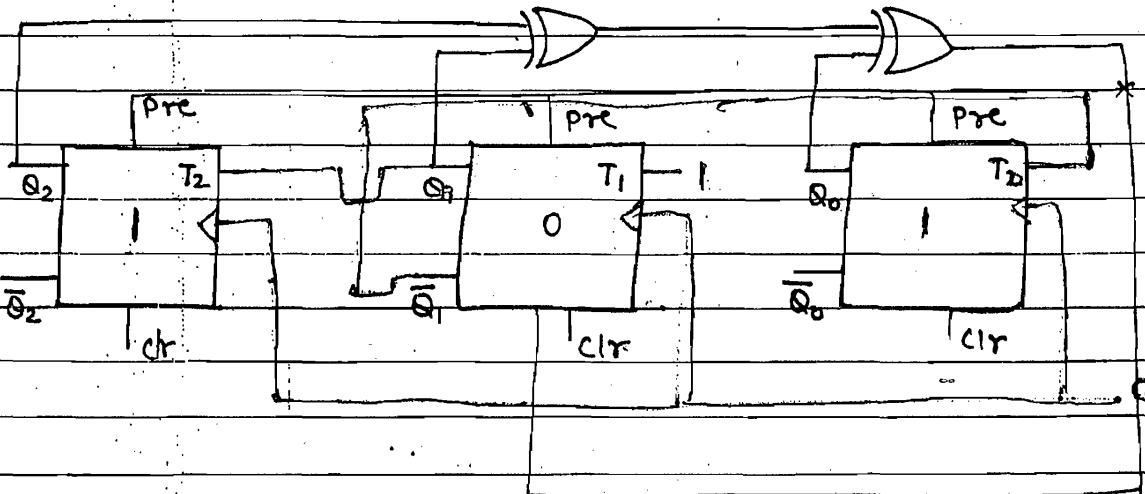
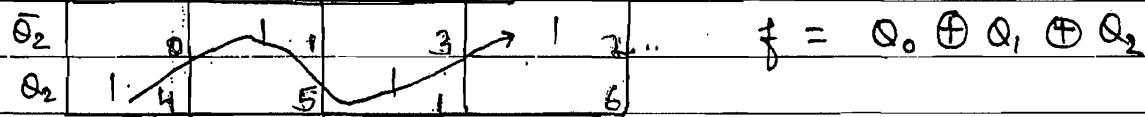
method to avoid lock-out condition : Get a circuit expression by unwanted state with the help of k-map

and design it as an additional circuit to the counter them with the help of proper preset and clear, the designed wanted state can be achieve without any loss of clock-pulse.

CLK	$Q_1$	$T_1$	$\bar{T}_1$	$Q_2$	$Q_1$	$Q_0$	...
					1	0	$0 \rightarrow 4$
	0	1	1	.	1	1	$1 \rightarrow 7$
	1	1	0	0	0	1	$1 \rightarrow 1$
	0	1	1	.	0	1	$0 \rightarrow 2$
	1	1	0	1	0	0	$0 \rightarrow 4$

→ It is not self-corrected.

$\bar{Q}_1\bar{Q}_0, \bar{Q}_1Q_0, Q_1Q_0, Q_1\bar{Q}_0$



\* State - deduction methods :

Step : 1 draw the state diagram for given CKT.

state diagram : it is the graphical representation which consists of :

1. Present state .
2. Input .
3. Next state .
4. Output .

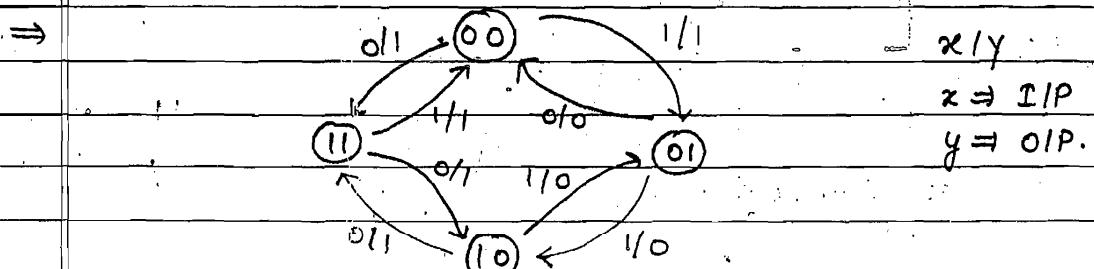
Step : 2 prepare the state table from state diagram.

Step : 3 identify the equivalent states .

equivalent states : Two states are said to be equiv. if their next state and O/P are identical.

Step : 4 when two states are equivalent then eliminate one of them.

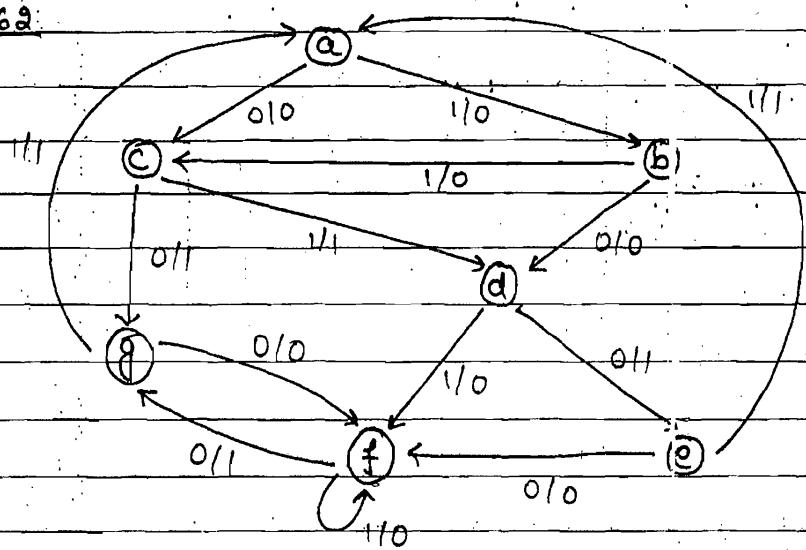
Step : 5 Repeat above process until we get all different states .



Present state	Next state $x=0$	Next state $x=1$	Output $x=0$	Output $x=1$
0 0	1 1	0 1	1	1
0 1	0 0	1 0	0	0
1 0	1 1	0 1	1	0
1 1	1 0	0 0	1	1

P.N. 6a

Ques:



Present state

Next-state

Output

	$x=0$	$x=1$	$x=0$	$x=1$
a	c	b	0	0
b	d	c	0	0
c	f, e	d	1	1
d	e	f, d	1	0
$d \rightarrow f$	$\cancel{f}, d$	<u>a</u>	0	1
$e \rightarrow g$	$\cancel{g}, e$	f	1	0
$\rightarrow g$	<u>f</u>	<u>a</u>	0	1

 $\Rightarrow 5$ -states

Ques: 33

 $x=0$      $x=1$  $x=0$      $x=1$ 

a

c

b

d

c

g

d

e

e

f

f

g

g

f

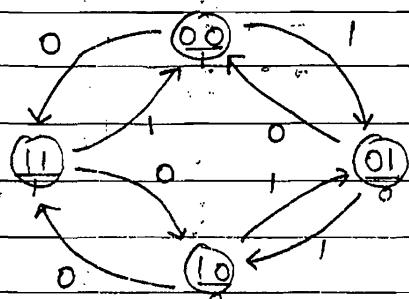
state - deduction methods are 2-types :

1. Meley method :
2. Moore method .

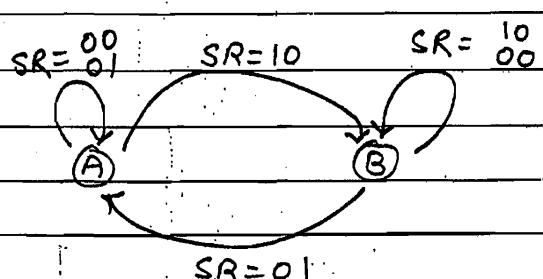
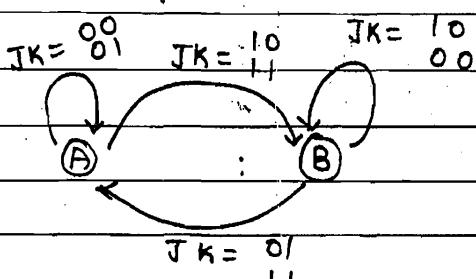
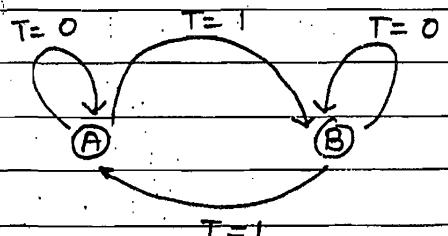
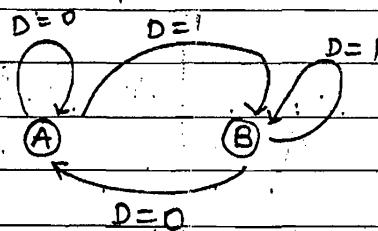
1. Meley-Method : In this method present O/P depends on present S/I/P and present - state .

Note: above discussed examples are Meley method only.

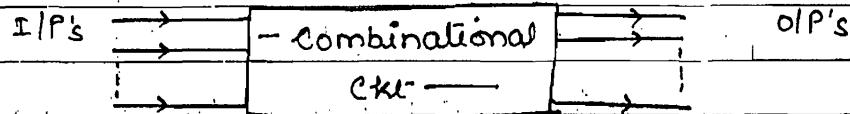
2. Moore-Method : In this method the present O/P depends only on present state .



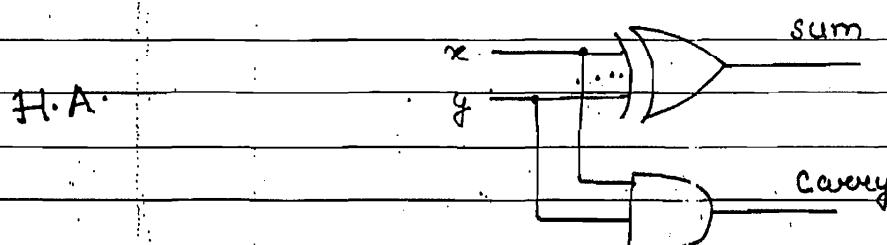
state diagram for f.f.  $\Rightarrow$



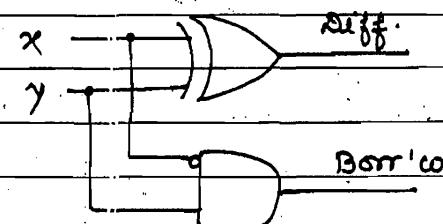
\* Combinational Circuit :-



x	y	C	sum	sum = $\bar{x}y + \bar{x}\bar{y}$ = $x \oplus y$
0	0	0	0	
0	1	0	1	
1	0	0	1	$C = xy$
1	1	1	0	



H.S.	x	y	B.	Diff.	Diff. = $\bar{x}y + x\bar{y}$ = $x \oplus y$
	0	0	0	0	
	0	1	1	1	
	1	0	0	1	$B = \bar{x}y$
	1	1	0	0	



F.A.

$x \ y \ z \ c \text{ sum}$

0 0 0 0 0

0 0 1 0 1

0 1 0 0 1

0 1 1 0 0

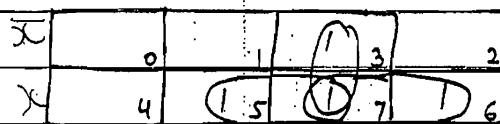
1 0 0 0 1

1 0 1 1 0

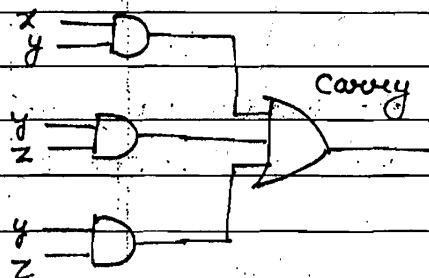
1 1 0 1 0

1 1 1 1 1

$$\text{Carry} = \bar{x}yz + \bar{x}\bar{y}z + xy\bar{z} + xyz.$$



$$\text{Carry} = xy + yz + yz$$

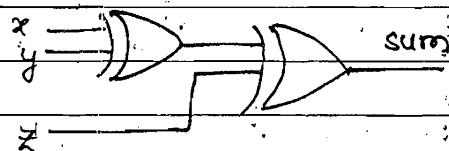


$$\text{sum} = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}\bar{z} + xy\bar{z}$$

$$\bar{x}\bar{y}z \quad \bar{x}y\bar{z} \quad y\bar{z} \quad y\bar{z}$$

$$\text{sum} = x \oplus y \oplus z$$

$$\bar{x} \quad | \quad 1 \quad | \quad 1 \quad | \quad 1 \quad | \quad 1$$



$x \ y \ z \ B \text{ Diff.}$

0 0 0 0 0

0 0 1 1 1

0 1 0 1 1

0 1 1 1 0

1 0 0 0 1

1 0 1 0 0

1 1 0 0 0

1 1 1 1 1

$$\text{Diff.} = x \oplus y \oplus z$$

$$\text{Borrow} = \bar{x}\bar{y}z + \bar{x}y\bar{z} + x\bar{y}z + xy\bar{z}$$

$$\bar{x}\bar{y}z \quad \bar{x}y\bar{z} \quad x\bar{y}z \quad xy\bar{z}$$

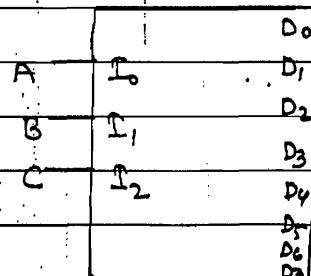
$$\text{Borrow} = \bar{x}y + y\bar{z} + \bar{x}\bar{z}$$

$$= \bar{x}y + y\bar{z} + \bar{x}\bar{z}$$

## Decoders :

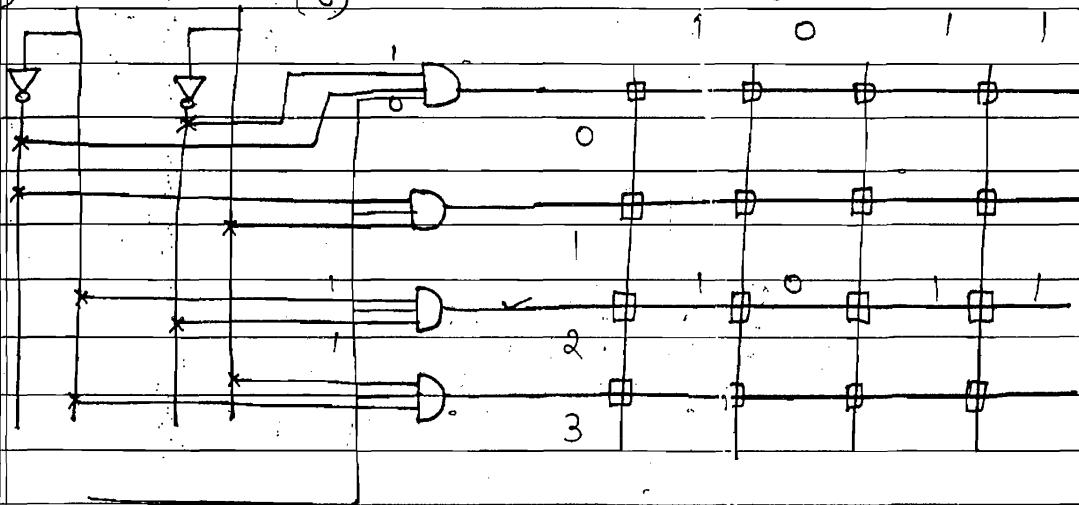
P.N. 59

Ques: 32



- decoders are widely used in memory.
- for  $n$ -select lines,  $2^n$  locations can be activated.

(1) A      B      (O)



Enable '1'  
[Demux]  
1 → many

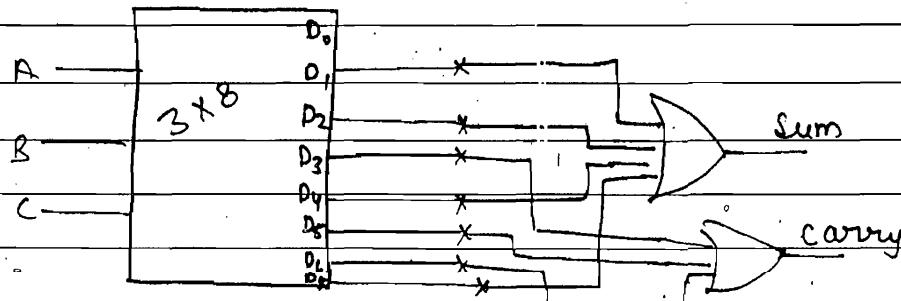
 $n \rightarrow 2^n$ 

	D <sub>0</sub>	$\bar{A}B$
3x4	D <sub>1</sub>	$\bar{A}B$
Deco.	D <sub>2</sub>	$A\bar{B}$
	D <sub>3</sub>	$AB$

$$\text{Sum} = \sum m(1, 2, 4, 7)$$

$$\text{Carry} = \sum m(3, 5, 6, 7)$$

F.A.



H.A.

A B C sum

0 0 0 0 0

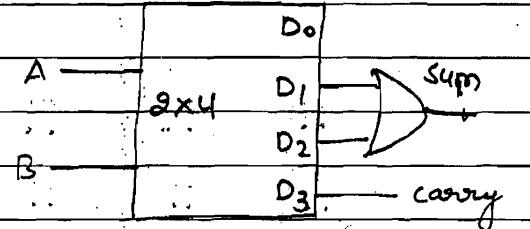
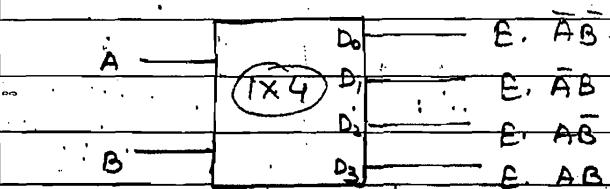
$$\text{Sum} = \sum m(1,2)$$

1 0 1 0 1

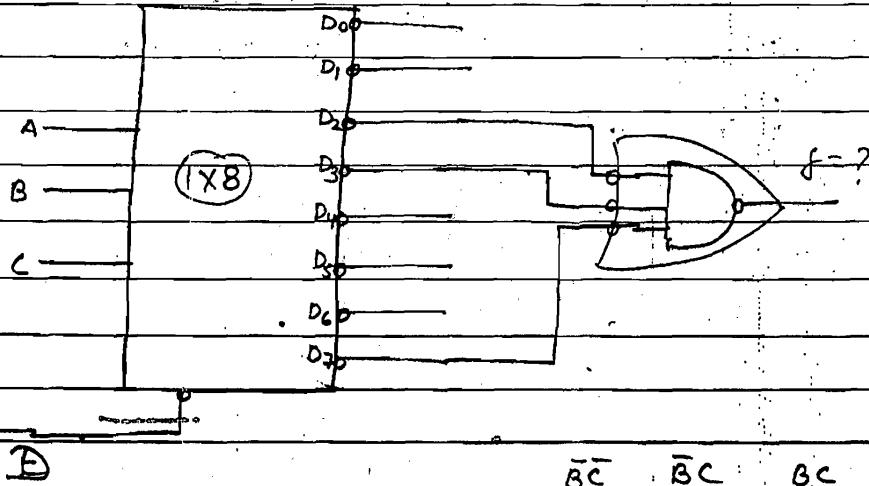
$$C = \sum m(3)$$

2 1 0 0 1

3 1 1 1 0

Demux :-

E



$$x = \sum m(2,3,7)$$

$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	$BC$
A	0	1	2

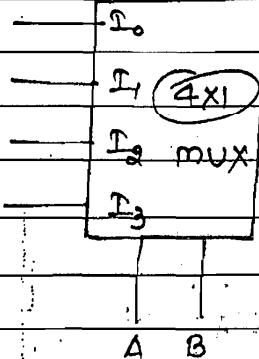
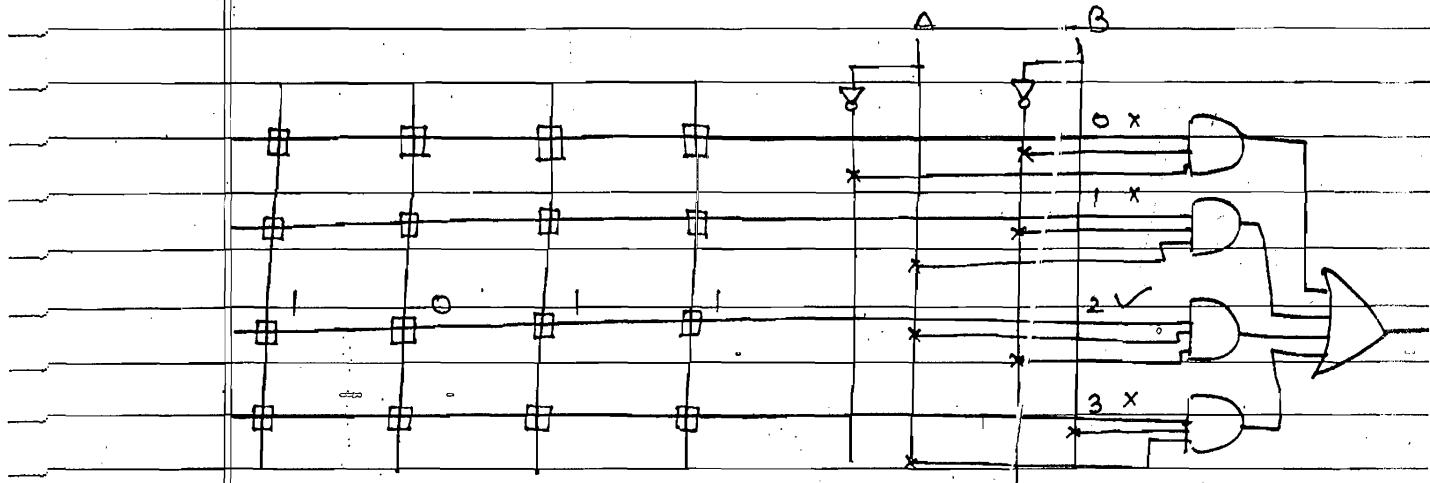
$$x = A.B + B.C$$

$$\Rightarrow f = E \cdot x$$

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

Multiplexer :: Many to one circuit.

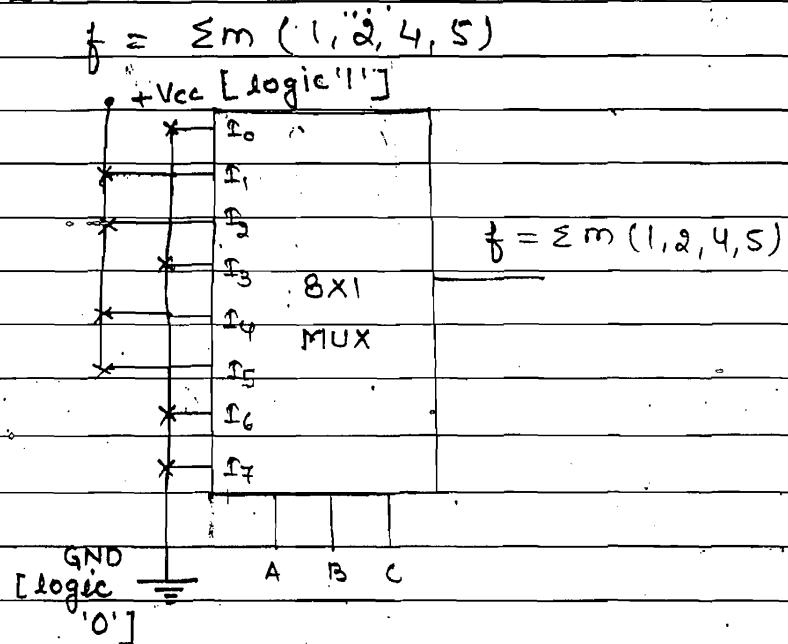
- Data selector.
- Parallel to serial converter.
- waveform generator.



$$f = \bar{A}\bar{B}(I_0) + \bar{A}B(I_1) \\ + A\bar{B}(I_2) + AB(I_3)$$

Designing the combinational ckt by MUXs :-

I:



II MSB :

$f(A B C)$			A	$\bar{A}$	$f_0$	$f_1$	$f_2$	$f_3$	$f = \sum m\{1, 2, 4, 5\}$
↓	↓	MSB LSB							
0	0	0 0							
1	0	0 1							
2	0	1 0							
3	0	1 1							
4	1	0 0							
5	1	0 1							
6	1	1 0							
7	1	1 1							

3. LSB:  $f(A B C)$

C	$f_0$	$f_1$	$f_2$	$f_3$
$\bar{C}$	0	2	4	6
C	1	3	5	7
R				
O				

$f = \sum m(1, 2, 4, 5)$

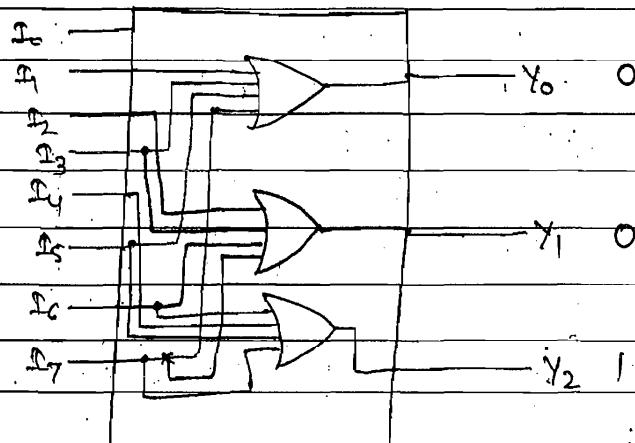
## Encoder:

$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	1	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$$Y_0 = I_1 + I_3 + I_5 + I_7.$$

$$Y_1 = I_2 + I_3 + I_6 + I_7,$$

$$Y_2 = I_4 + I_5 + I_6 + I_7.$$



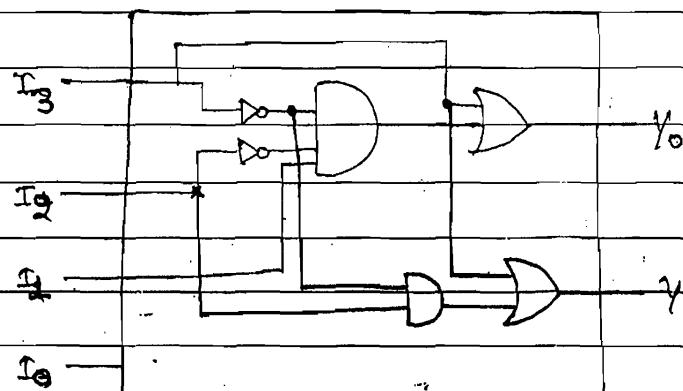
8x3 (Dcial to Binary Encoder).

## Priority Encoder:

$I_3$	$I_2$	$I_1$	$I_0$	$Y_1$	$Y_0$
0 - - 0	0	1		0	0
0	0	1	*	0	1
0	1	*	*	1	0
1	*	*	*	1	1

$$Y_0 = I_1 \cdot I_2' \cdot I_3' + I_3.$$

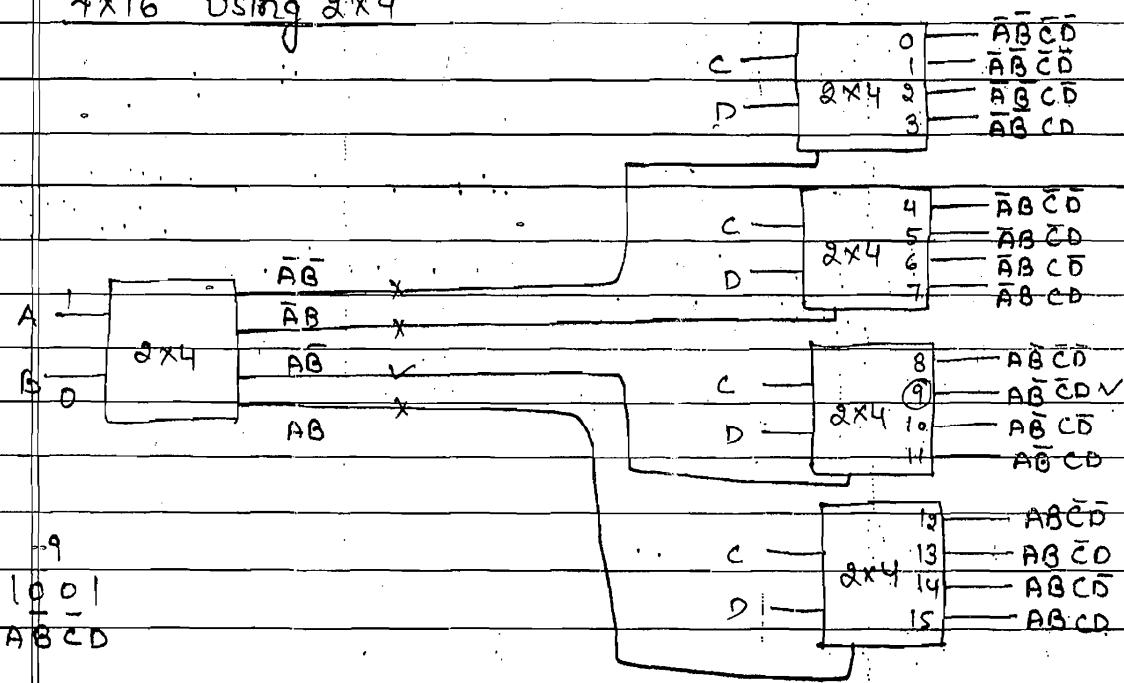
$$Y_1 = I_2 \cdot I_3' + I_3.$$



4x2 MUX

\* Higher ordered circuits by using lower ordered circuits

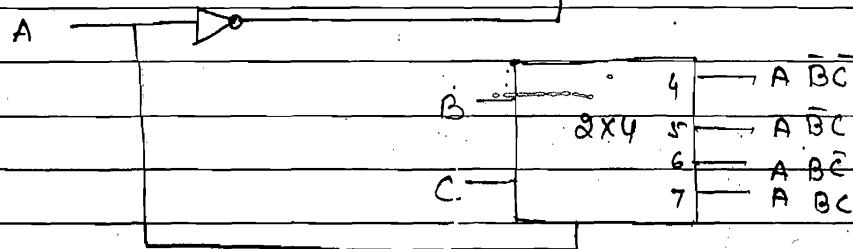
4x16 using 2x4



⑨ 3x8 using 2x4

$$\frac{8}{4} = 2$$

B -	0	A BC
	1	A $\bar{B}$ C
C -	2	$\bar{A}$ B C
	3	$\bar{A}$ $\bar{B}$ C

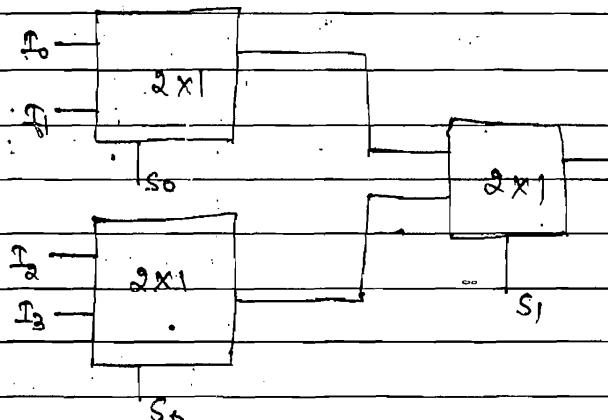


③# 4x1 MUX By using 2x1 + ...

$$\frac{4}{2} = 2$$

$$\frac{2}{2} = 1$$

(3)



short-cut:

① 4x16 using 2x4

$$\frac{16}{4} = 4$$

$\frac{4}{4} = 1$  → should fully divisible

(5) → so we need (5), 2x4 Demux's

② 3x8 using 2x4

$$\frac{8}{4} = 2$$

↓ not divisible

- kle b6

Points:

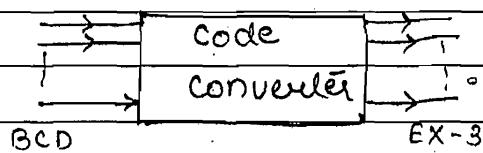
For eg: ① :

No. of decoders = ⑤

No. of enable decoders = ④.

NUMBER - SYSTEM :

Codes and Codes Converter :



BCD (8·4·2·1)  
7/5/9/3/6  
0111 / 0101 / 1001 / 0011 / 0110      BCD

8 4 2 1  
0 1 1 1 ← 7  
0 1 0 1 ← 5  
1 0 0 1 ← 9

10 → 1010 → invalid in BCD

11 → 1011      these are Binary - codes.

12 → 1100

13 → 1101

14 → 1110

15 → 1111

→ 110

000 | 0000 ← BCD

→ 111

0001 0001 ← BCD

→ 114

000 | 0100 ← BCD

2 4 2 1

0 0 0 0 0

1 0 0 0 1

2 0 0 1 0

3 0 0 1 1

4 0 1 0 0

5 0 1 0 1

6 0 1 0 0

7 0 1 0 1

8 1 1 1 0

9 1 1 1 1

2 4 2 1

5 → 0 1 0 1 X

5 → 1 0 1 1 V

⇒ add '3' in BCD will get Ex-3

0 → 0 0 0 0

+ 1 1 ← '3'

0 0 1 1 ← Ex-3 of '0'

1 → 0 0 0 1

+ 1 1 ← '3'

0 1 0 0 ← Ex-3 of '1'

Ex-3

self-complement

0 → 0 0 1 1

1 → 0 1 0 0

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

1

BCD to Ex-3 conversion :-BCD  
I/PEx-3  
O/P

for 'W'

A	B	C	D	W	X	Y	Z	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$	
0	0	0	0	0	0	1	1	$\bar{A}\bar{B}$	0	1	3	2
1	0	0	0	1	0	1	0	$\bar{A}B$	4	1	5	1
2	0	0	1	0	0	1	0	AB	$X_{12}$	$X_{13}$	$X_{15}$	$X_{14}$
3	0	0	1	1	0	1	0	$A\bar{B}$	1	8	1	9
4	0	1	0	0	0	1	1		1	0	$X_{11}$	$X_{10}$
5	0	1	0	1	1	0	0					
6	0	1	1	0	1	0	0					
7	0	1	1	1	1	0	1					
8	1	0	0	0	1	0	1					
9	1	0	0	1	1	1	0					
				1	0	1	0					

$$W = A + BC + BD$$

I/P's

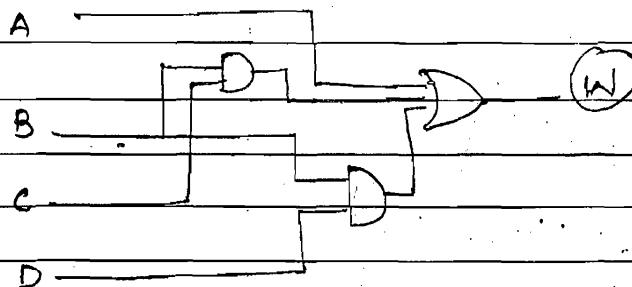
BCD

O/P's

EX-3

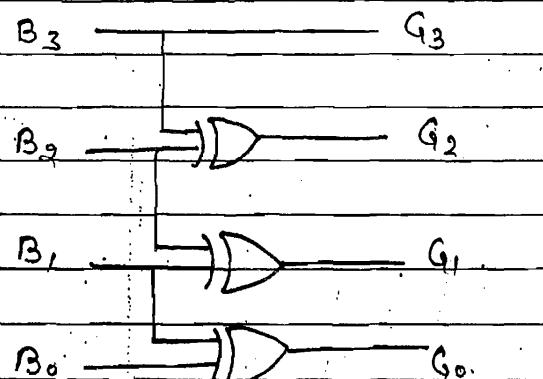
HW: Make for

x, y, z as well.



## Binary to Gray Conversion

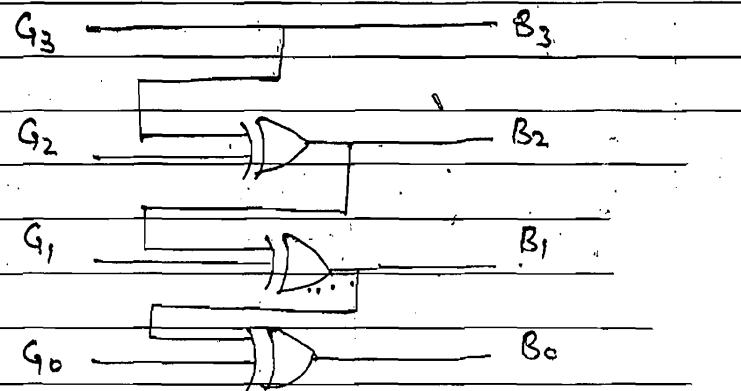
	MSB				
Binary	$B_3$	$B_2$	$B_1$	$B_0$	
		⊕		⊕	
	↓	↓	↓	↓	↓
Gray	1	0	1	1	
	$G_3$	$G_2$	$G_1$	$G_0$	
	MSB				
$G_3 = B_3$			$G_1 = B_2 \oplus B_1$		
$G_2 = B_3 \oplus B_2$				$G_0 = B_1 \oplus B_0$	



## \* Gray to Binary Convrs

	$G_3$	$G_2$	$G_1$	$G_0$	
Gray	1	0	1	1	
		⊕		⊕	
	↓	↓	↓	↓	↓
Binary	1	1	0	1	
	$B_3$	$B_2$	$B_1$	$B_0$	
	MSB				

$$\begin{aligned}
 B_3 &= G_3 \\
 B_2 &= B_2 \oplus G_2 \\
 B_1 &= B_2 \oplus G_1 \\
 B_0 &= B_1 \oplus G_0
 \end{aligned}$$



## SEVEN SEGMENT :-

	I/P B.C.D.	O/P Seven segment
a b c d	0 0 0 0	1 1 1 1 1 1 0
e f g	1 0 0 0 1	0 1 1 0 0 0 0
1 2 3	0 0 1 0	1 1 0 1 1 0 1
4	0 1 0 0	0 0 1 0 0 1 1
5	0 1 0 1	1 0 1 0 1 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 0 0 1 1

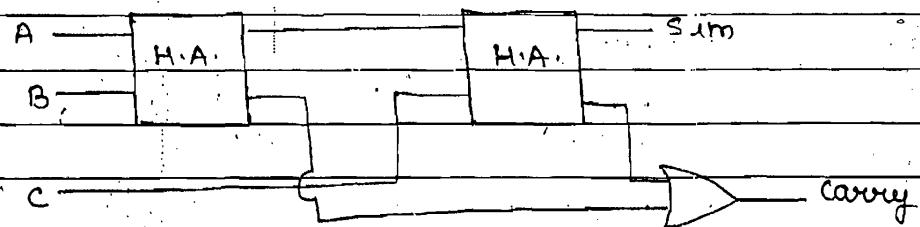
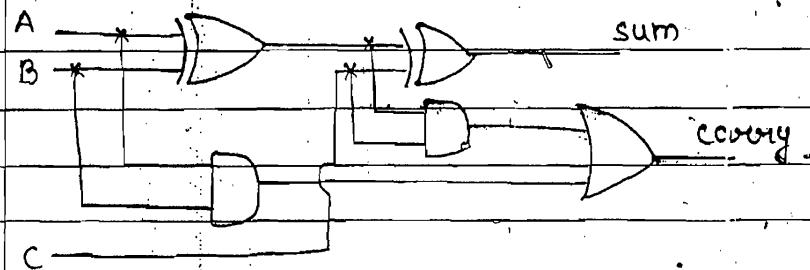
H.W.  $\Rightarrow$  34, 35 P.N. 60

P <sub>2</sub>	P <sub>1</sub>	a	b	c	d	e	f	g	
0	0	1	1	1	1	1	1	0	'0'
0	1	1	1	0	1	1	0	1	'2'
1	0	1	0	1	1	0	1	1	'5'
1	1	1	0	0	1	1	1	1	'E'

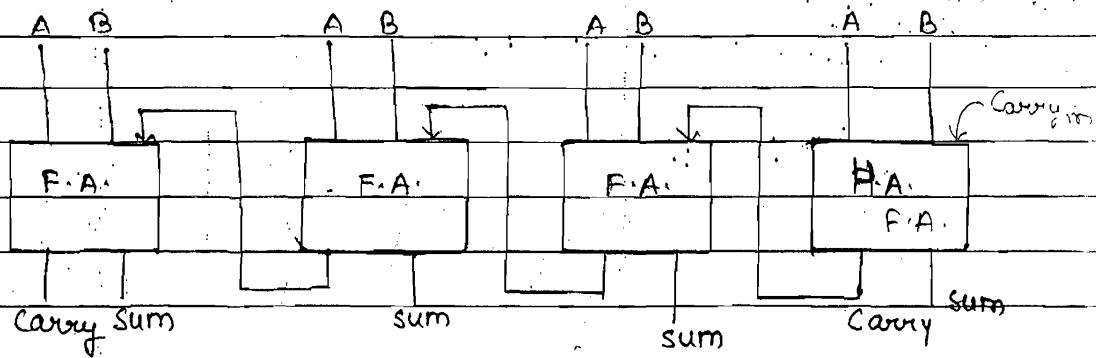
$$\overline{P_2} \quad \overline{P_1} \quad b = \overline{P_2}$$

$\overline{P_2}$	1	1
P	1	0

## Full-Adder Using Half-adder:



## \* Parallel Binary Adder (or) Ripple carry adder:-



Note:

o -  $N(F.A.)$  when there is a carry in is given.o -  $(N-1) F.A + 1 H.A.$ 

$$(N-1) [2H.A. + 1 \text{OR gate}] + 1H.A.$$

$$= \left\{ 2(N-1)H.A. + (N-1) \text{OR gates} \right\}$$

$$\therefore N = 4$$

$$(N-1)$$

$$(2 \cdot 4 - 1)$$

$$4 - 1$$

$$(8 - 1)$$

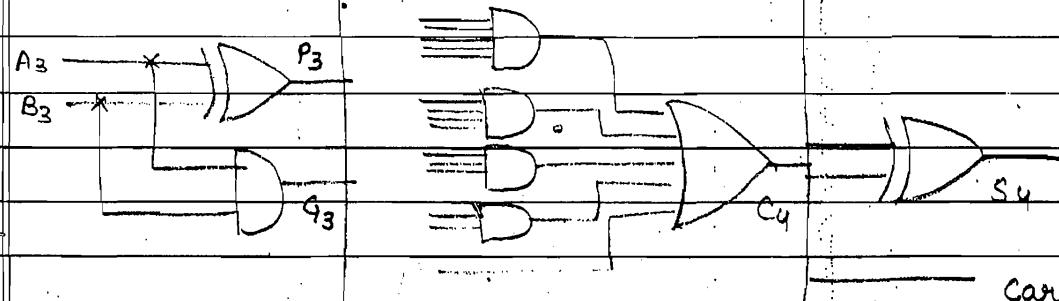
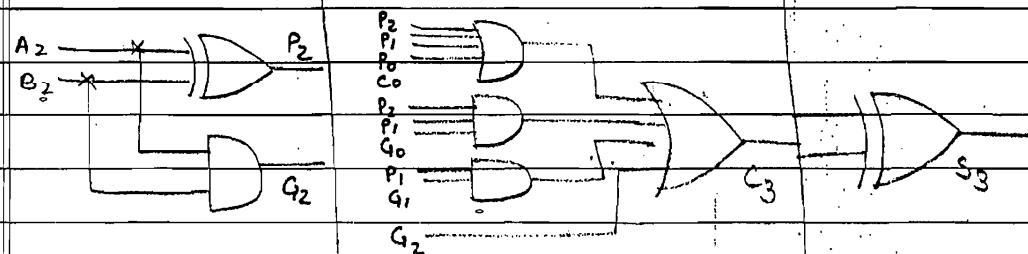
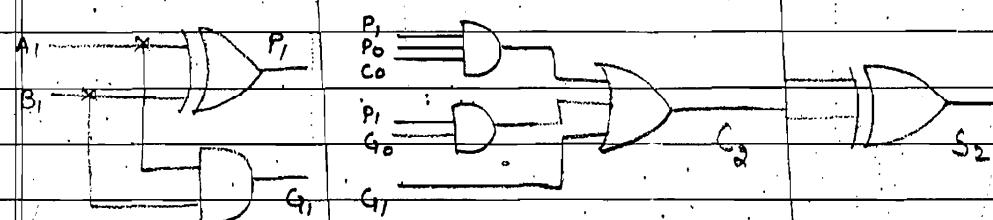
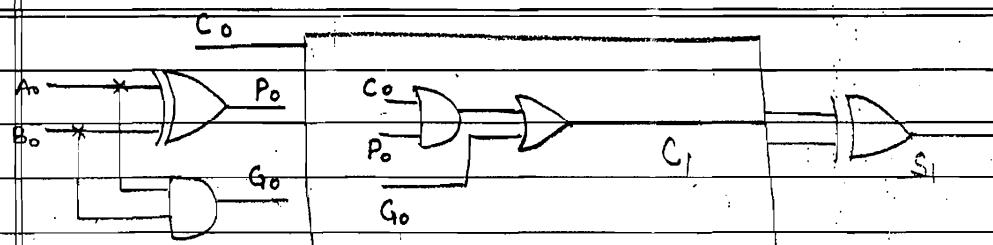
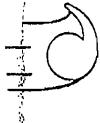
$$= 3 \text{ OR gates}$$

$$1 = H.A.$$

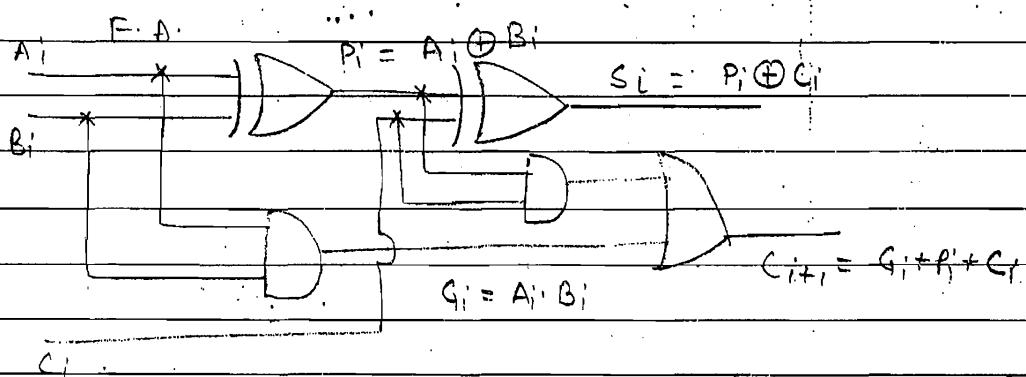
A'

AL  
FC

de'smat™



Carry



$$C_{i+1} = C_{0+1} = \underline{C_1} = G_0 + P_0 C_0$$

$$\begin{aligned} C_{i+1} &= C_{1+1} = \underline{C_2} = G_1 + P_1 C_1 = G_1 + P_1 [G_0 + P_0 C_0] \\ &= G_1 + P_1 G_0 + P_1 P_0 C_0. \end{aligned}$$

$$\begin{aligned} C_{i+1} &= C_{2+1} = \underline{C_3} = G_2 + P_2 C_2 = G_2 + P_2 [G_1 + P_1 G_0 + P_1 P_0 C_0] \\ &= G_2 + P_2 G_1 + P_1 P_2 G_0 + P_1 P_2 P_0 C_0. \end{aligned}$$

$$\begin{aligned} C_{i+1} &= C_{3+1} = \underline{C_4} = G_3 + P_3 C_3 = G_3 + P_3 [G_2 + P_2 G_1 + P_1 P_2 G_0 \\ &\quad + P_1 P_2 P_0 C_0] \\ &= G_3 + P_3 G_2 + P_2 P_3 G_1 + P_1 P_2 P_3 G_0 + P_0 P_1 P_2 P_3 C_0 \end{aligned}$$

Short-cut:

$$\text{AND } \Rightarrow \frac{n(n+1)}{2} = \frac{4(4+1)}{2} = 10.$$

$$\text{OR } \Rightarrow n \rightarrow 4.$$

Complement - Methods :

dec	bin	oct	hexa
4's $\Rightarrow$	10's	2's	8's
(4-1)'s $\Rightarrow$	9's	1's	7's

$\rightarrow$  radix  
(or)  
base

9's Complement :

$$\begin{array}{r} 9 \\ 7_{(10)} \\ \hline 2 \end{array} \qquad \begin{array}{r} 9 & 9 & 9 \\ 9's \leftarrow | 7 \cdot 3 \\ \hline 8 & 2 & 6 \end{array}$$

$$1's \text{ complement} \Rightarrow 1's \leftarrow 0$$

↓

(1)

$$1's \leftarrow 1$$

(0)

$$2's \text{ complement} \Rightarrow$$

$$2's \leftarrow 0$$

↓

1

+ 1

↓

(0)

$$2's \leftarrow 1$$

0

↓

+ 1

↓

(1)

$$10's \text{ complement} \Rightarrow$$

$$10's \leftarrow (7)_{10}$$

↓

2

+ 1

↓

(3)

$$10's \leftarrow 17 - 3$$

↓

82 - 6

+ 1

↓

82 - 7

$$\begin{array}{r} 10100 \\ \downarrow \\ 01100 \end{array}$$

2's

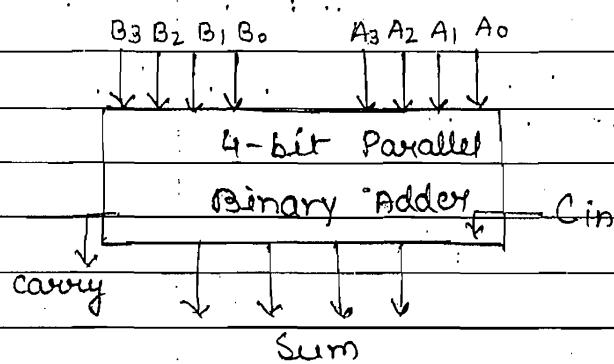
$$\begin{array}{r} 1010 \\ \downarrow \\ 01011 \end{array}$$

2's

$$\begin{array}{r} 1000 \\ \downarrow \\ 1000 \end{array}$$

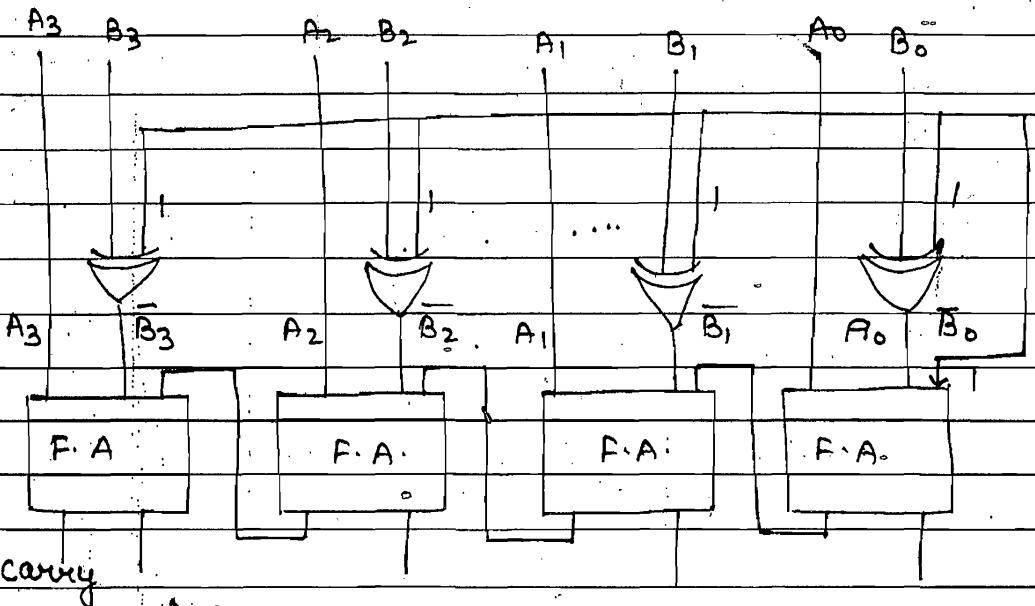
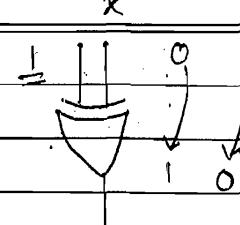
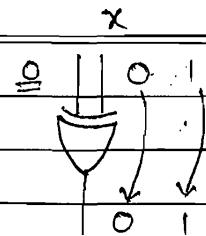
2's

$2's$  complement Adder / Subtractor :



$$A \rightarrow A_3 A_2 A_1 A_0$$

$$B \rightarrow B_3 B_2 B_1 B_0$$



$$\text{control} \Rightarrow 0 \Rightarrow A + B$$

$$\Rightarrow 1 \Rightarrow A - B$$

$$A_0 + [2\text{'s comp. of } B]$$

$$A_0 + [-B_0] \quad \text{Generally 2's comp is 'negative'}$$

$$\Rightarrow A_0 - B_0$$

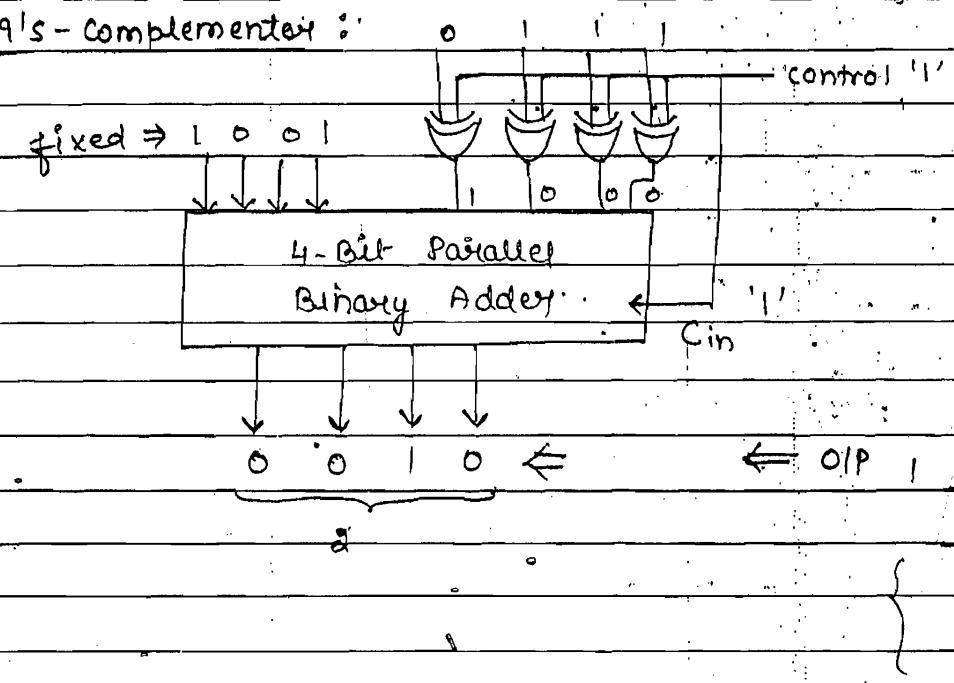
1 wise branching and control logic  
3 live within and exist without

class 9/3

Date \_\_\_\_\_

Page \_\_\_\_\_

\* 9's-complementor :



\* BCD - Arithmetic :

I. BCD - addition: When two valid BCD no are added, if the result is invalid or carry is generated then the '6' should be added for the final result.

Case(i)  $\Rightarrow$

$$\begin{array}{r} 5 \rightarrow 0101 \\ 4 \rightarrow 0100 \\ \hline 9 \rightarrow 1001 \text{ valid} \end{array}$$

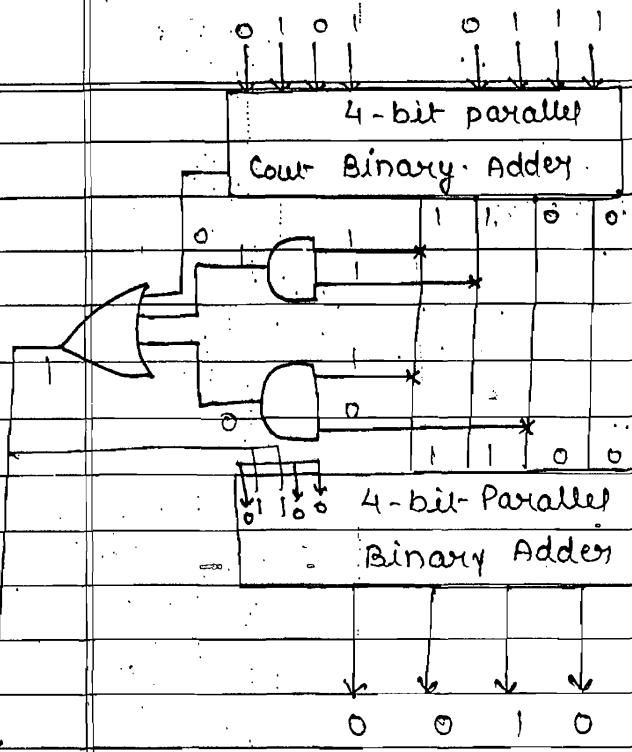
Case(ii)  $\Rightarrow$

$$\begin{array}{r} 5 \rightarrow 0101 \\ 7 \rightarrow 0111 \\ \hline 1100 \text{ invalid} \\ 0110 \leftarrow +6 \\ \hline 0001/0010 \end{array} \quad \left. \begin{array}{l} 10 \\ 11 \\ 12 \\ 13 \\ 14 \\ 15 \end{array} \right\} \begin{array}{l} \text{BCD} \\ \text{invalid} \\ \text{six no's} \end{array}$$

12

Case(iii)  $\Rightarrow$

$$\begin{array}{r} 8 \rightarrow 1000 \\ 9 \rightarrow 1001 \\ \hline \text{carry } 10001 \leftarrow \text{invalid} \\ 0001/0111 \leftarrow 17 \end{array}$$



### B) BCD Subtractor :

case(i) Result is +ve : 9's complement of "substrahend" is added with "minuend" then if the result is invalid then '6' should be added along with the endround carry.

case(ii) if the result is -ve : when the minuend is added with 9's complement of substrahend if the result is valid and no carry generated it indicates -ve result and 9's complement should be taken for final answer.

$$\begin{array}{r} 6 \rightarrow 0110 \\ - 4 \rightarrow 0101 \quad [9's \text{ comp. of } 4] \end{array}$$

1011 invalid

$$0110 \rightarrow 6$$

$$\begin{array}{r} 0001 \\ \text{endround} \quad +1 \\ \hline 0010 \\ \text{carry} \quad \underline{-} \\ \hline \text{(+9)} \end{array}$$

$$\begin{array}{r} 4 \rightarrow 0100 \\ - 6 \rightarrow 0011 \quad [9's \text{ comp. } 6] \end{array}$$

$$\hline \quad \quad \quad 0111 \leftarrow \text{valid}$$

No-carry

1. Valid } Result (-)ve  
2. No-carry }

for final answer Take 9's comp.

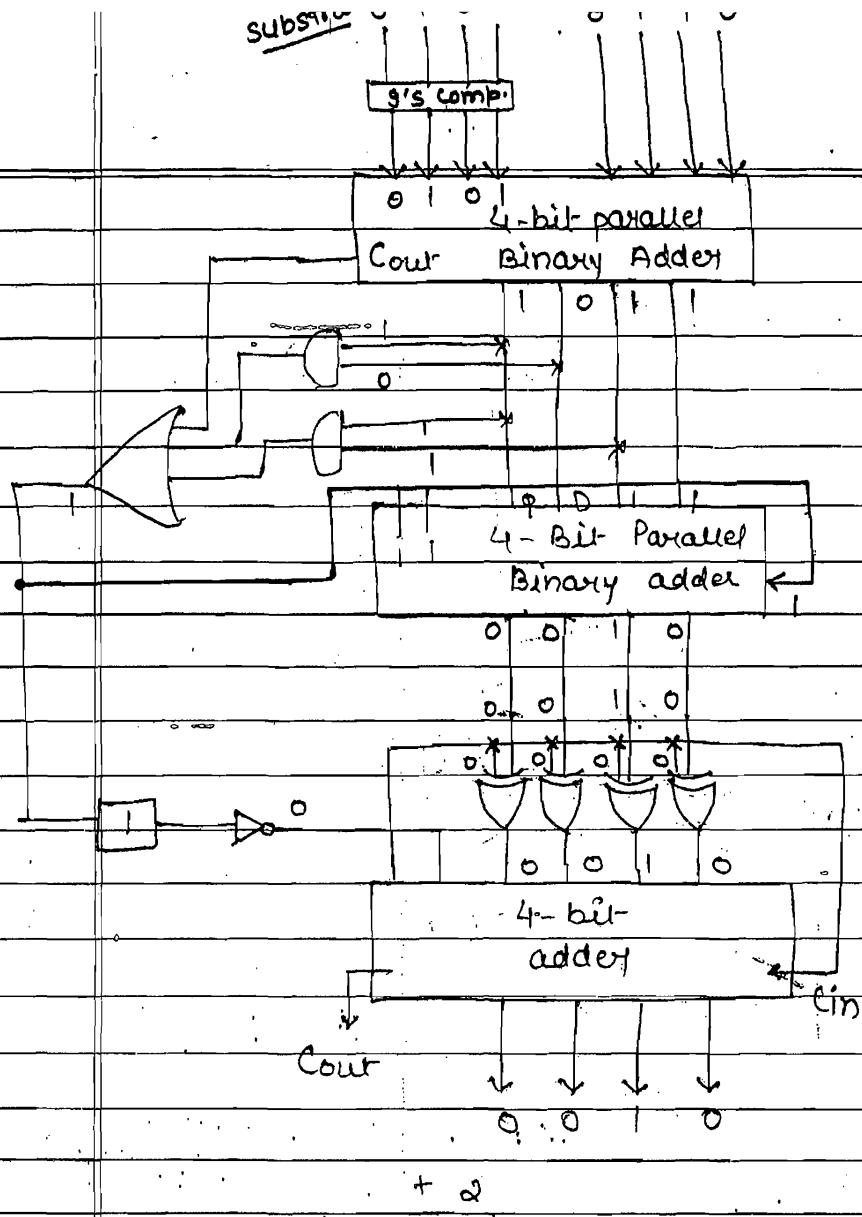
$$0111 \rightarrow 7 = 9_{10}$$

subtraction

classmate 95

Date \_\_\_\_\_

Page \_\_\_\_\_



+ 2

## No. Representation :

Case(i) (+)ve number

+51

32 16 8 4 2 1  
+ + 0 0 +

0 110011  
 ↓ sign bit      ← magnitude

Case(ii) -ive number

-51

1 110011 ⇒ sign-magnitude  
 ↓ sign bit      ← magnitude

1 001100 ⇒ 1's comp.  
 ↓ sign bit      ← 1's comp.

001100

+1

1 001101 ⇒ 2's complement  
 ↓ sign bit      ← 2's comp

Given	sign	1's	2's
0101	+5	+5	+5
1101	-13	(-2)	(-3)

1 1101 sign.  
 13

1 1101  
 ↓ 0010  
 -2 → 1's comp.

1 110  
 ↓ 0010  
 +1  
 0011  
 (-3)

sign

Range - ↗ I's comp.

$$\Rightarrow -[2^{n-1} - 1]_{\text{lo}} + [2^{n-1} - 1] \text{ for 1's complement}$$

Ex:-  $n = 3$  for  $n=3$ , range  $\Rightarrow -3 \leq x \leq +3$

sign	i's comp.
0 0 0 + 0	0 0 0 + 0
0 0 1 + 1	0 0 1 + 1
0 1 0 + 2	0 1 0 + 2
0 1 1 + 3	0 1 1 + 3
1 0 0 - 0	1 0 0
1 0 1 - 1	1 0 1
1 1 0 - 2	1 1 0
1 1 1 - 3	1 1 1

$\Rightarrow$  Rang :  $[-2^{n-1} \text{ to } +2^{n-1} - 1]$  for 2's complement.

Sign	2's comp. for n=3
000 +0	000 +0 Range: -7 to 4 to +3
001 +1	001 +1
010 +2	010 +2
011 +3	011 +3
100 -0	100 -
101 -1	101
110 -2	110
111 -3	111

$$[\alpha^{(3-)} - 1] \quad 10 \quad + \quad [\alpha^{3-} - 1]$$

$$= (2^2 - 1) \text{ to } (2^2 - 1)$$

-3 to +3 for first table above fig(i)

- - if MSB 1

$$\text{all other bits '0'} \Rightarrow -2^{n-1}$$

Overflow - Condition :

$$\text{Cin} \rightarrow \text{f} = \text{Cin} \oplus \text{Cout}$$

if ;  $f = 0$ , No-overflow.

$f = 1$ , overflow.

eg...  $5 \rightarrow \begin{array}{r} x \\ 0 \end{array} | 0 1 0 1$

$4 \rightarrow \begin{array}{r} y \\ 0 \end{array} | 0 1 0 0$

$\underline{-}$

$= 1 0 0 1$

$$\begin{aligned} f &= \bar{x}\bar{y}\bar{z} + xy\bar{z} \\ &= \bar{0}\bar{0}1 + 00\bar{1} \\ &= 1 + 0 \\ &= 1 \\ &= \text{overflow.} \end{aligned}$$

→ additional bit to overcome the overflow.

$$\begin{array}{r} 5 \rightarrow \boxed{0} 0 1 0 \\ 4 \rightarrow 0 0 1 0 0 \\ \hline 0 1 0 0 1 \end{array}$$

Tx:

Rx:

↓ error

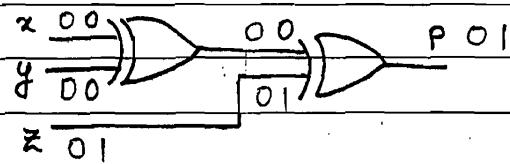
Even-parity

eg.. 0101

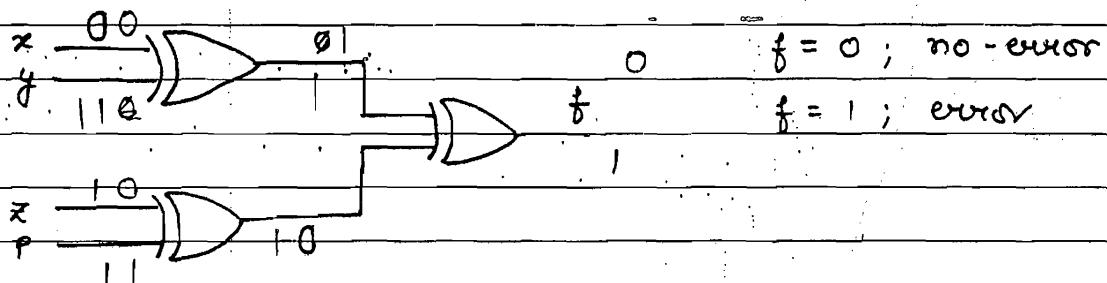
0111

$$\begin{array}{cccc} x & y & z & p \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array}$$

## \* Parity Generator



## \* Parity checker:



## \* Hamming-Code : (7-bit)

given - data : 1101

Tx :	P <sub>1</sub>	P <sub>2</sub>	D <sub>3</sub>	P <sub>4</sub>	D <sub>5</sub>	D <sub>6</sub>	D <sub>7</sub>
	1	0	1	0	1	0	1

eisen-Parity

P <sub>4</sub>	P <sub>2</sub>	P <sub>1</sub>
2 <sup>3</sup>	2 <sup>1</sup>	2 <sup>0</sup>
x	y	z

7-bit

P <sub>1</sub>
P <sub>2</sub>
P <sub>4</sub>

12-bit

P <sub>1</sub>
P <sub>2</sub>
P <sub>4</sub>
P <sub>8</sub>

15-bit

P <sub>1</sub>
P <sub>2</sub>
P <sub>4</sub>
P <sub>8</sub>
P <sub>16</sub>

16-bit

P <sub>1</sub>
P <sub>2</sub>
P <sub>4</sub>
P <sub>8</sub>
P <sub>16</sub>

0 0 0 0

odd

1 0 0 1

$$P_1 [1, 3, 5, 7] \Rightarrow [P_1, \underline{111}] \Rightarrow P_1 = 1$$

2 0 1 0

even

3 0 1 1

$$P_2 [2, 3, 6, 7] \Rightarrow [P_2, \underline{101}] \Rightarrow P_2 = 0$$

4 1 0 0

$$P_4 [4, 5, 6, 7] \Rightarrow [P_4, \underline{101}] \Rightarrow P_4 = 0$$

5 1 0 1

6 1 1 0

7 1 1 1

Rx :

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

error

$$P_1(1, 3, 5, 7) \Rightarrow P_1(1011) \Rightarrow \gamma = 1$$

$$P_2(2, 3, 6, 7) \Rightarrow P_2(0001) \Rightarrow \beta = 1$$

$$P_4(4, 5, 6, 7) \Rightarrow P_4(0101) \Rightarrow \alpha = 0$$

$$\alpha \beta \gamma = 0, 1, 1 \quad (011) \Rightarrow 3$$

so we can also check for the place where error has been occurred.

### Code-Conversions $\Rightarrow$

#### ① Dec. to Any (successive division) :

10 <sup>2</sup>	10 <sup>1</sup>	10 <sup>0</sup>	10 <sup>-1</sup>	10 <sup>-2</sup>
↓				

Decimal Point

$$(72)_{10} \Rightarrow 72_{(10)}$$

$$2 \times 10^0 \rightarrow 2$$

$$7 \times 10^1 \rightarrow 7.0$$

$$72_{(10)}$$

#### \* Decimal to Binary :

$$2 | 12$$

2	6	$\rightarrow 0$
2	3	$\rightarrow 0$

$$\frac{12}{2} \rightarrow 1$$

$$12_{(10)} \Rightarrow 1100_{(2)}$$

Fractions : (successive multiplication) :-

$$0.75_{(10)}$$

$$\begin{array}{rcl} 2 \times 0.75_{(2)} & = & 1.50 \\ 2 \times 0.50 & = & \downarrow 1.00 \end{array} \quad 0.75_{(10)} = 0.11_{(2)}$$

Dec to Oct :-

$$286_{(10)}$$

$$\begin{array}{r} 8 | 286 \\ 8 | 35 \rightarrow 6 \\ 4 \rightarrow 3 \end{array}$$

$$286_{(10)} = 436_{(8)}$$

Dec to Hexa :-

$$428_{(10)}$$

$$\begin{array}{r} 16 | 428 \\ 16 | 26 \rightarrow 12 \\ 1 \rightarrow 10 \end{array}$$

$$428_{(10)} = 1AC_{(16)}$$

(Q) Any to Dec :-

a). Bin to Dec.

$$1101.011$$

$$\Rightarrow 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

$$\Rightarrow 13.375_{(10)}$$

b). Oct to Dec. :-

$$372_{(8)}$$

$$= 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0$$

$$= 250_{(10)}$$

c). Hexa to Dec. :-

$$356_{(16)}$$

$$\Rightarrow 3 \times 16^2 + 5 \times 16^1 + 6 \times 16^0$$

$$\Rightarrow 854_{(10)}$$

d) Bin  $\rightarrow$  Dec  $\Rightarrow$

Bin  $\rightarrow$  Oct  $\Rightarrow$

$11011.010$

$011/011 \cdot 010/100$

$\Rightarrow 33.24(8)$

e) Bin.  $\rightarrow$  Hexa :-

$11101.10$

$0001/1101.1000$

$\Rightarrow 1D.8(16)$

Any  $\rightarrow$  Bin :-

a). dec  $\rightarrow$  bin :

oct  $\rightarrow$  bin :

7S: 36<sub>(8)</sub>

7/5: 3/6<sub>(8)</sub>

$\Rightarrow 111/101 \cdot 011/110(2)$

b). Hexa  $\rightarrow$  Bin  $\Rightarrow$

A B C . D (16)

A | B | C . D

1 0 1 0 1 0 1 1 1 1 0 0 . 1 1 0 1 (2)

Oct  $\rightarrow$  Any  $\Rightarrow$

oct  $\rightarrow$  bin

oct  $\rightarrow$  dec

a). Oct  $\rightarrow$  Hexa :

$275 \cdot 3(8)$

bin

$0000/1011/1101 \cdot 0110$

$\Rightarrow BD.6(16)$

Any  $\rightarrow$  Oct :-dec  $\rightarrow$  Oct ✓bin  $\rightarrow$  Oct ✓AB<sup>9</sup>, 2C(16)Hexa  $\rightarrow$  Oct

bin

101010111001.001011000

5271, 130<sub>(8)</sub>

Hexa to Any :

Hexa to Dec. ✓

Hexa to Bin. ✓

Hexa to Oct. ✓

Any  $\rightarrow$  Hexa :

Dec. to Hexa ✓

Bin. to Hexa ✓

Oct. to Hexa, ✓

## # 2's Complement Arithmetic :

- Minuend is added with 2's complement of substrahend. If the res 'MSB' is 'Zero' then the result is (+)ve and the answer is True Binary.
- If MSB is '1' then result is negative then the ans. is 2's complement.

Note  $\Rightarrow$  Carry should be ignored. $x \Rightarrow$  minuend .. $-y \Rightarrow$  Substraend.

If MSB (Sign-bit) = 0  $\Rightarrow$  Result is +ve [ Ans. is true binary ] } Ignore  
 = 1  $\Rightarrow$  Result is -ve [ Ans. is 2's comp. ] } Carry

a) Case: (i)

$$+9 \rightarrow \begin{array}{|c|} \hline 0 \\ \hline \end{array} | 0 0 1$$

$$+4 \rightarrow \begin{array}{|c|} \hline 0 \\ \hline \end{array} | 0 0 0 0$$

$$\begin{array}{|c|} \hline 0 \\ \hline \end{array} | 1 1 0 1$$

↓  
Sign-Bit

sum = +13

Case : (ii)  $\Rightarrow$

$$\begin{array}{r}
 +9 \rightarrow 0100 \\
 -4 \rightarrow 1110 \\
 \hline
 100101
 \end{array}$$

↓      ↓  
Ignored      Sign Bit

$\therefore$  Sum = +5

case (iii)  $\Rightarrow$

$$\begin{array}{r}
 -9 \rightarrow 1011 \\
 +4 \rightarrow 00100 \\
 \hline
 11011
 \end{array}$$

↓  
Sign-Bit

$\therefore$  Sum = -5

Case : (iv)

$$\begin{array}{r}
 -9 \rightarrow 1011 \\
 +9 \rightarrow 0100 \\
 \hline
 10000
 \end{array}$$

↓      ↓  
Ignored      Sign  
Carry      Bit

$\therefore$  Result = 0

Case (iv)  $\Rightarrow$

$$\begin{array}{r}
 -9 \rightarrow 1011 \\
 -4 \rightarrow 11100 \\
 \hline
 110011
 \end{array}$$

↓      ↓  
Ignored      Sign  
Carry      Bit

$\therefore$  Sum = -13

\* 1's complement Arithmetic :-

Minuend should be added to 1's complement of substrahend with a carry generated it should become end-around carry. then check the MSB :-

if  $MSB \Rightarrow 0$  [Result is (+)ve] = Ans. is true Binary? end around  
 $\Rightarrow 1$  [Result is (-)ve] = 1's comp. be in Ans. / -nd carry

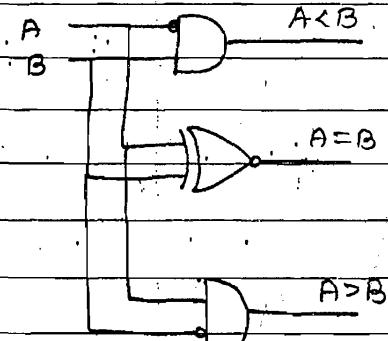
Note :-

Magnitude - Comparator  $\Rightarrow$

$$A \rightarrow A_3 A_2 A_1 A_0$$

$$B \rightarrow B_3 B_2 B_1 B_0$$

A	B	$A < B$	$A = B$	$A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



$$\begin{array}{cccc} 1 & x & x & x \\ 0 & x & x & x \\ \hline A > B \end{array}$$

$$\begin{array}{cccc} 0 & x & x & x \\ 1 & x & x & x \\ \hline A < B \end{array}$$

$$A_3 = B_3 \Rightarrow (A_2, B_2) \text{ go for it}$$

4-Bit Magnitude - Comparator  $\Rightarrow$

A < 1

$A_3B_3$	$A_2B_2$	$A_1B_1$	$A_0B_0$	$A < B$	$A = B$	$A > B$
----------	----------	----------	----------	---------	---------	---------

$A_3 > B_3$	x	x	x	0	0	1
$A_3 < B_3$	x	x	x	1	0	0
$A_3 = B_3$	$A_2 > B_2$	x	x	0	0	1
$A_3 = B_3$	$A_2 < B_2$	x	x	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	x	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	x	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0

$A < B$

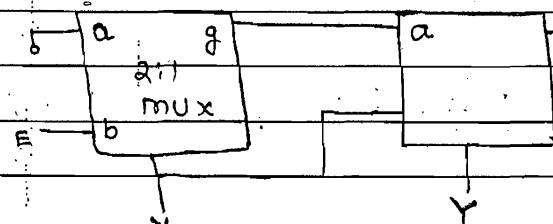
$$(A_3 \cdot B_3) + (A_3 \oplus B_3)(A_2 \cdot B_2) + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \cdot B_1) \\ + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \oplus B_1)(A_0 \cdot B_0)$$

$| A = B | (A_3 \odot B_3)(A_2 \odot B_2)(A_0 \odot B_0)$

$A > B$

$$(A_3 \cdot B_3) + (A_3 \oplus B_3)(A_2 \cdot B_2) + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \cdot B_1) \\ + (A_3 \odot B_3)(A_2 \odot B_2)(A_1 \cdot B_1)(A_0 \cdot B_0)$$

Ques: 10



given:  $g = ac + b\bar{c}$

i)  $g = C(0) + E\bar{C}$

$g = E\bar{x}$

ii)  $\bar{C}b + ca$

$$= \bar{Y}(x) + Y(g)$$

$$= \bar{Y}x + YE\bar{x}$$

### Hazard :-

Bcoz of diff propagation delays at different parts of ckt the O/P may not be the expected result known as Hazard.

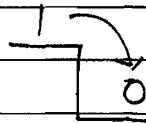
### Glitch :-

for temporary interval of time we get a wrong O/P, known as glitch.

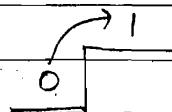
To avoid hazard in ckt ie to get hazard-free ckt, redundant groups should be consider.

### Hazard - Type :-

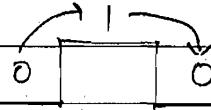
Static '1'



Static '0'



Dynamic



## LOGIC - FAMILY

1. Classification of digital IC's :-

a) Small scale integration (SSI) :-

no. of logic gates < 12

no. of components upto 99.

b) Medium scale integration (MSI) :-

no. of logic gates : 12 - 99.

no. of components : 100 - 999.

c) Large scale integration (LSI) :-

no. of logic gates : 100 - 999

no. of components : 1000 - 9999.

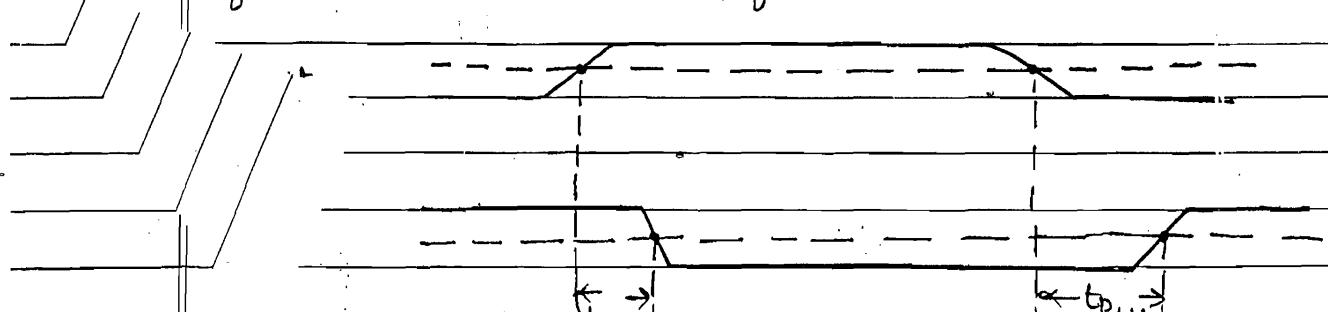
d) Very large scale integration (VLSI) :

no. of logic gate: above 1000.

no. of compone : above 10,000.

Characteristics parameters :

a. Propagation delay : It is the time delay b/w application of i/p and occurrence of o/p.



$$t_p = t_{PHL} + \frac{t_{PLH}}{2}$$

$t_{PLH} > t_{PHL}$ ; BJT (Reverse-recovery-time)

$t_{PLH} < t_{PHL}$ ; FET ('charge storage time')

Unit : nano-sec.

POWER-DISSIPATION :- It is amount of the power dissipated in an IC.

$$P.D = V_{CC} \cdot I_{CC}$$

$$I_{CC} = \frac{I_{CCH} + I_{CL}}{2}$$

$I_{CCH} \Rightarrow$  Current when all inputs are high.

$I_{CL} \Rightarrow$  " " " " " low.

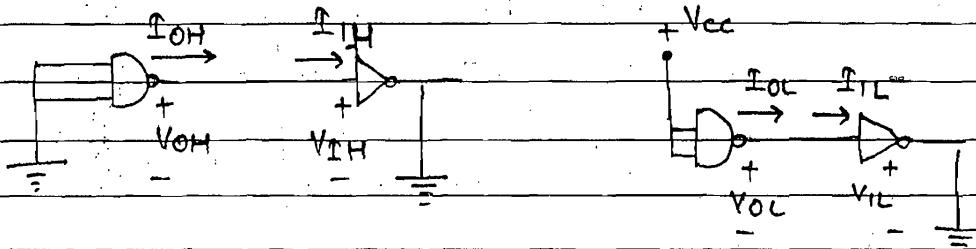
-- Unit  $\Rightarrow$  milli-watts.

- \* Figure of Merit :- It is the product of power dissipation and propagation delay.  
unit  $\Rightarrow$  pico-joules.

Note: For the best use of IC figure of merit should be as small as possible. It is also known as speed-power product.

- \* FAN-IN :- It is ability to handle the no. of inputs for a logic gate.

- \* FAN-OUT :- It is ability to drive the no. of loading gates by a logic gate O/P.

# Voltage and Current Parameters:**High****Low** $V_{IH}$  (min)  $\Rightarrow$  (High-level input voltage).

It is the min. high vlg. required for a logic '1' as o/p. any vlg below this can't be treated as high.

 $V_{IL}$  (max)  $\Rightarrow$  (Low-level input vlg).It is the max<sup>n</sup> vlg limit acceptable for logic '0' as o/p. $V_{OH}$  - High level o/p voltage. $V_{OL}$  - Low level o/p voltage.Current-Parameters: $I_{OH}$  : (High-level o/p current) $I_{IH}$  : (High-level g/p current) $I_{OL}$  : (Low-level o/p current) $I_{IL}$  : (Low-level g/p current)Formulas:

$$\text{Fan-out} = \frac{I_{OH}}{I_{IH}}$$

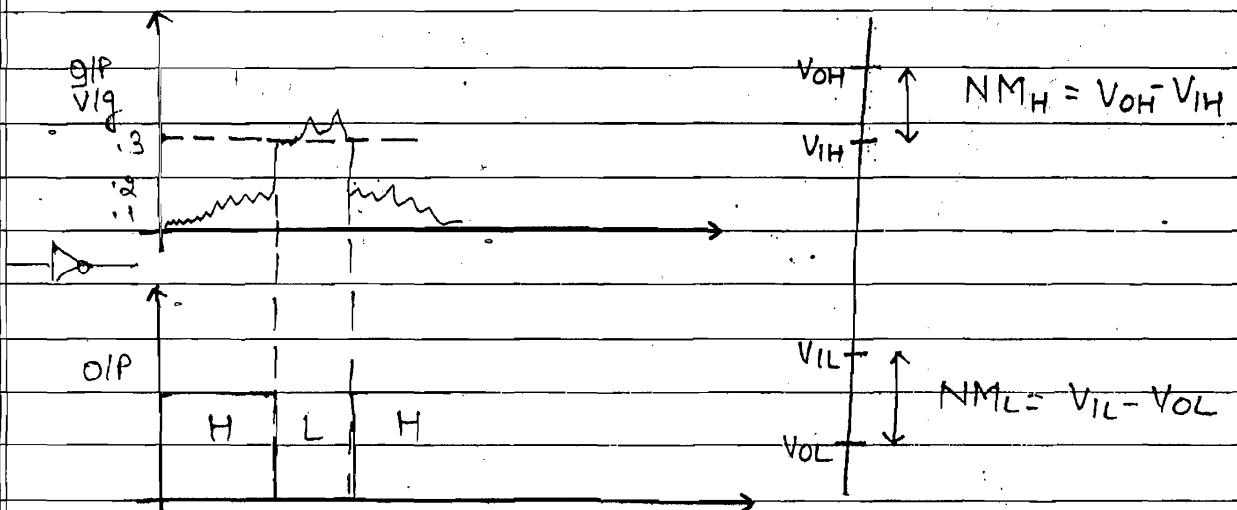
$$\text{Fan-out} = \frac{I_{OL}}{I_{IL}}$$

Note :

From above two values the least value should be consider for the overall fan-out.

Noise-immunity : It is ability to withstand the for v/Ig fluctuations

noise-margin : Measure of noise-immunity.



Note :

from above two values the least-values should be considered for the overall noise-margin.

\* Operating Temperatures :  $0^\circ C$  to  $+70^\circ C$

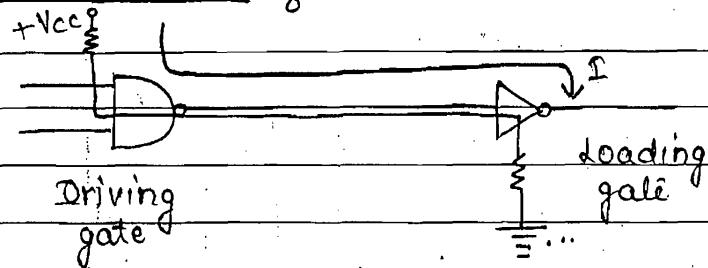
... Industrial Applications

... Consumer Applications

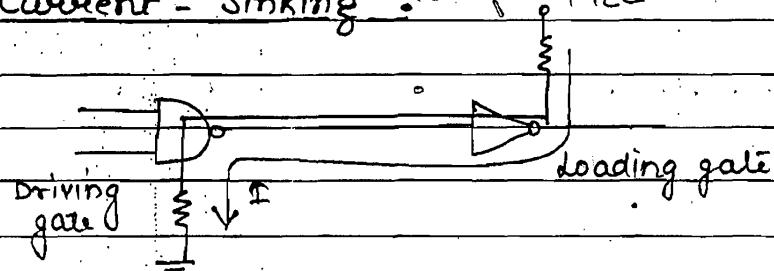
$-55^\circ C$  to  $+125^\circ C$

... for military applications

## \* Current - Sourcing :-

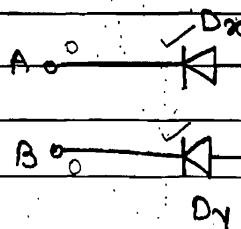


\* Current - sinking :-  $\downarrow$  +ve



## Devices as switching circuits (logic-gates) :-

And 'ealē ⇒



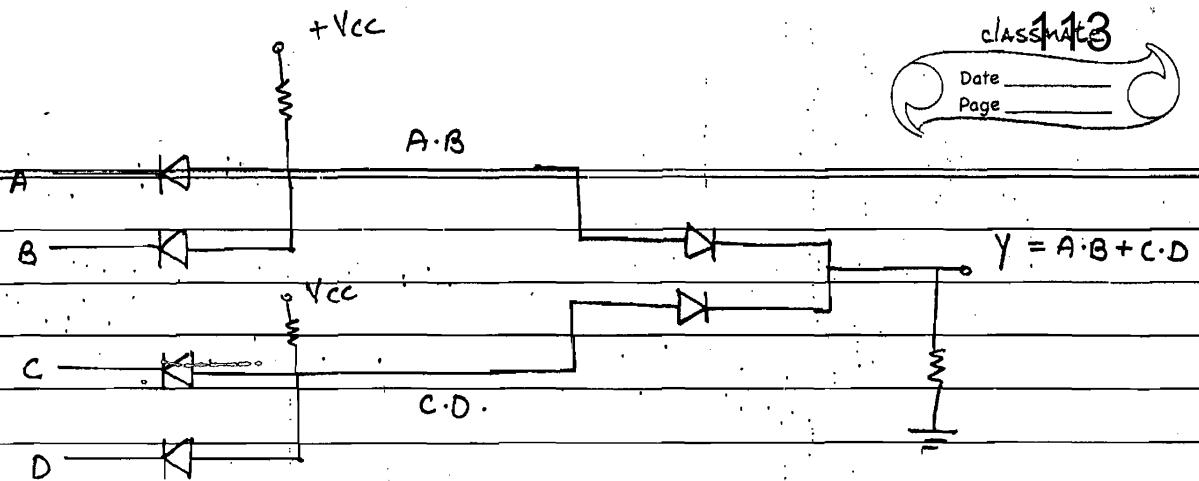
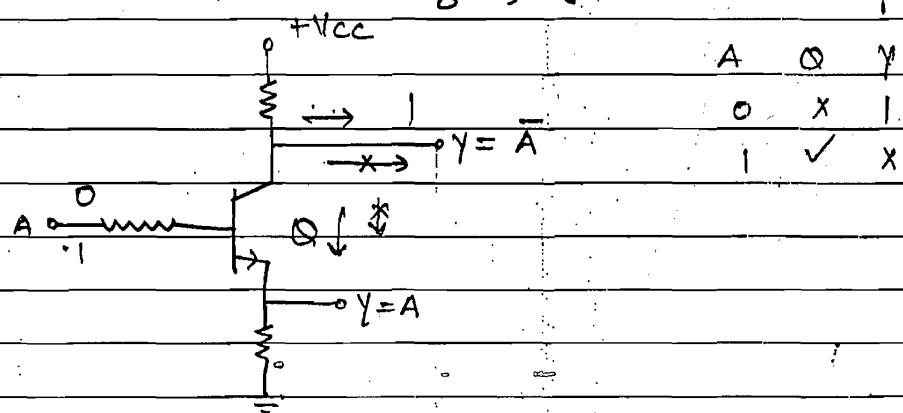
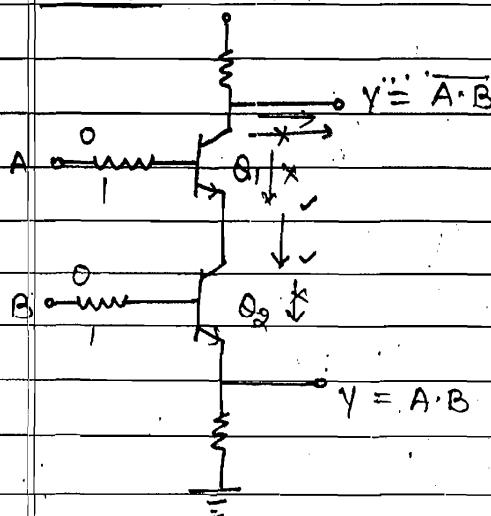
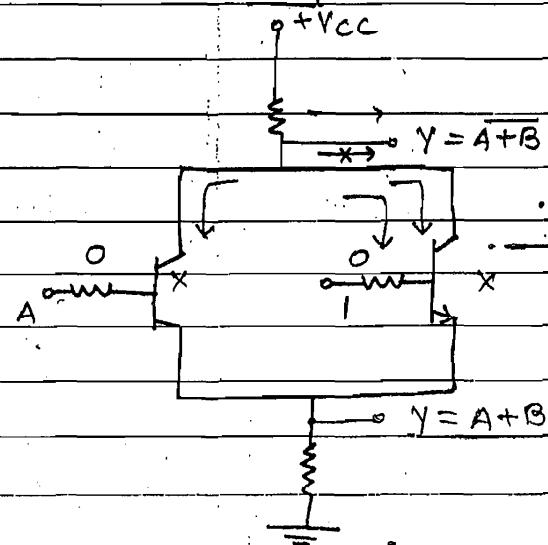
$$y = A \cdot B$$

A	B	Dx	Dy	Y
0	0	✓	✓	0
0	1	✓	✗	0
1	0	✗	✓	0
1	1	✗	✗	1

OR-Gate  $\Rightarrow$

$$Y = A + \beta$$

A	B	Dx	Dy	Y
0	0	x	x	0
0	1	x	v	1
1	0	v	x	1
1	1	v	v	1

QES Ques:RTL (Resistor-Transistor-logic)NOT-gateNAND:NOR:

A	B	$Q_1$	$Q_2$	Y
0	0	X	X	1
0	1	X	✓	1
1	0	✓	X	1
1	1	✓	✓	0

A	B	$Q_1$	$Q_2$	Y
0	0	X	X	1
0	1	X	✓	0
1	0	✓	X	0
1	1	✓	✓	0

# RTL ( Resistor Transistor logic ) :-

NOT , NAND , NOR gates previously given .

\*\* Advantages :

\*\* Disadvantages :

# DCTL ( Direct- coupled - Transistor - logic ) :-

DCTL is same as that of RTL except input-resistors are removed .

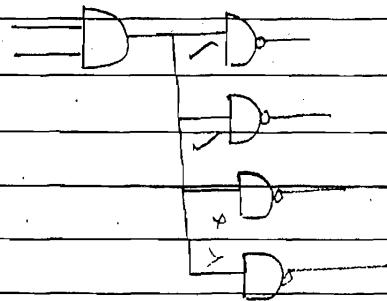
\*\* advantages .

\*\* disadvantages .

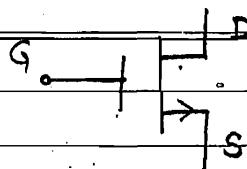
Note :

⇒ DCTL suffers from current hogging problem .

\* Current - Hogging :- Bcoz of different saturation levels of loading gate transistors , the current will be hogged in certain loads only and other loads will be in starvation of current known as current hogging .

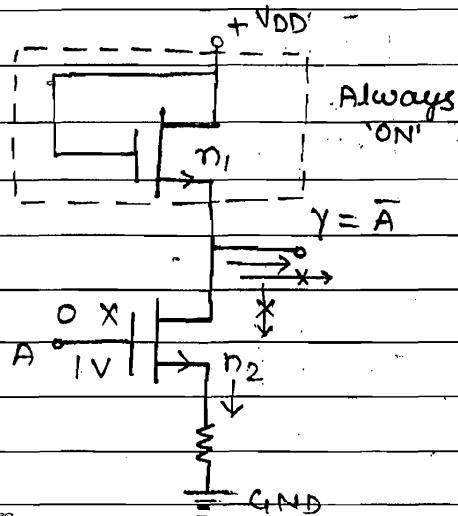


# NMOS :-

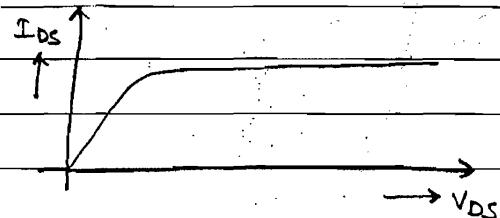
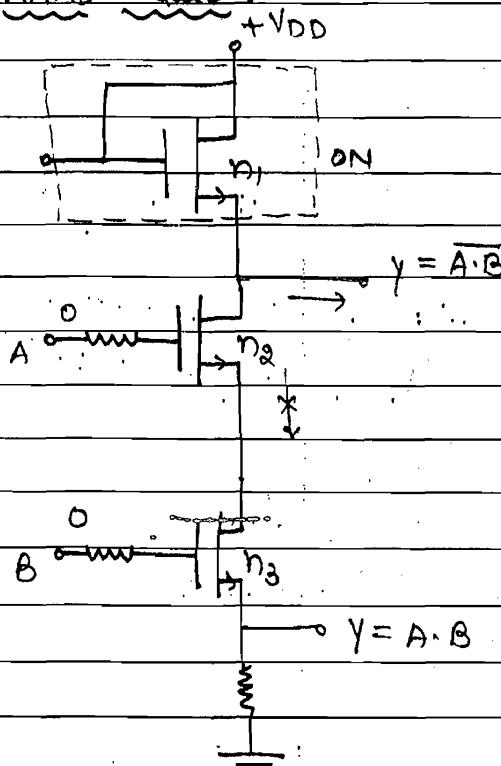


Gate  $\Rightarrow$  (+) ve  $\Rightarrow$  ON  
 (-) ve  $\Rightarrow$  OFF.

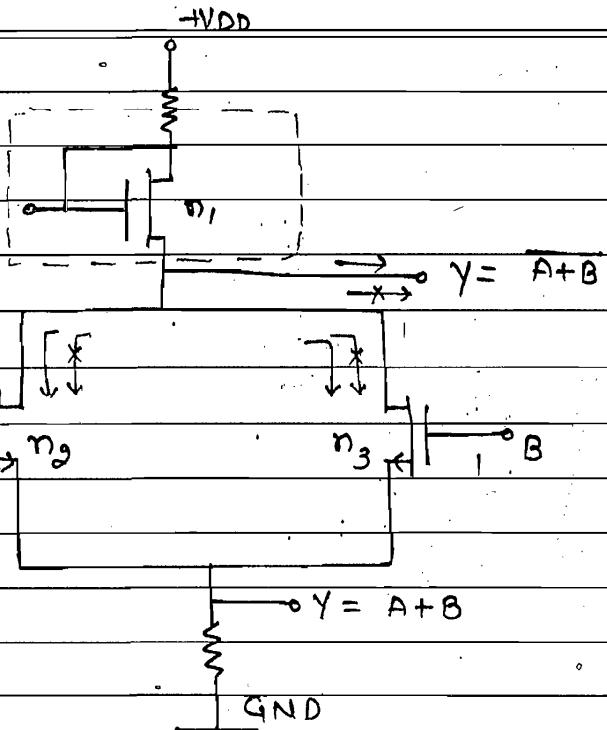
gate diffusion always (+) ve  
 always 'ON'



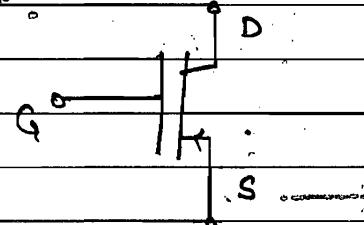
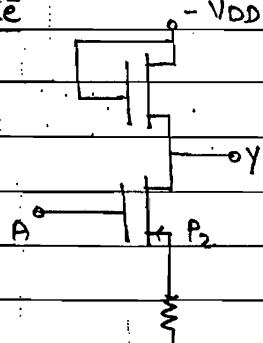
A	$n_1$	$n_2$	$y$
0	V	X	1
1	V	V	0

NAND - Gate :

A	B	$n_1$	$n_2$	$n_3$	$y$
0	0	V	X	X	1
0	1	V	X	V	1
1	0	V	V	X	1
1	1	V	V	V	0

NOR Gate:

A	B	$n_1$	$n_2$	$n_3$	$Y$
0	0	✓	x	x	1
0	1	✓	x	✓	0
1	0	✓	✓	x	0
1	1	✓	✓	✓	0

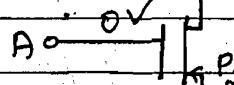
P-MOS Techn.:Gate  $\Rightarrow$  (+) ve  $\rightarrow$  OFF(-) ve  $\rightarrow$  ONi) NOT-gate

A	$P_1$	$P_2$	$Y$	
0	✓	✓	1	(GND)

A	$P_1$	$P_2$	$Y$	
1	✓	x	0	(-VDD)

NOR-gate  $\Rightarrow$ 

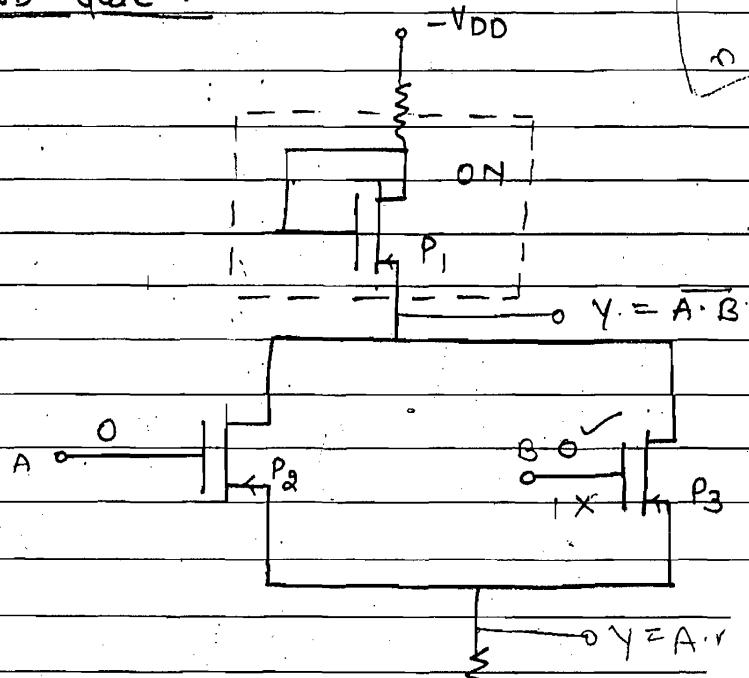
$$\circ Y = A + B$$



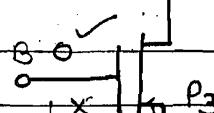
A	B	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	Y
0	0	✓	✓	✓	1 GND,
0	1	✓	✓	X	0 -VDD,
1	0	✓	X	✓	0 -VDD
1	1	✓	X	X	0 -VDD

$$\circ Y = A + B$$

Gate (-) →  
Source (-) →

NAND-gate  $\Rightarrow$ 

$$\circ Y = \overline{A \cdot B}$$

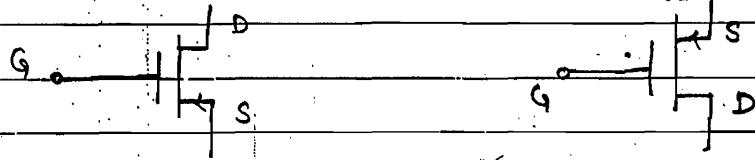


$$\circ Y = A \cdot r$$

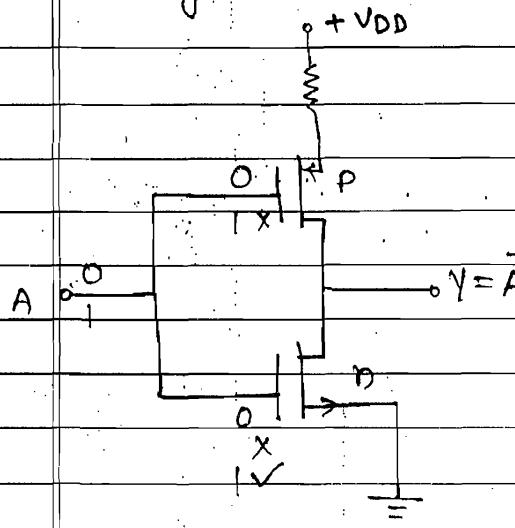
A	B	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	Y
0	0	✓	✓	✓	1 GND
0	1	✓	✓	X	1 GND
1	0	✓	X	✓	1 GND
1	1	✓	X	X	0 -VDD

$$\circ Y = A \cdot r$$

C-MOS Tech.  $\Rightarrow$

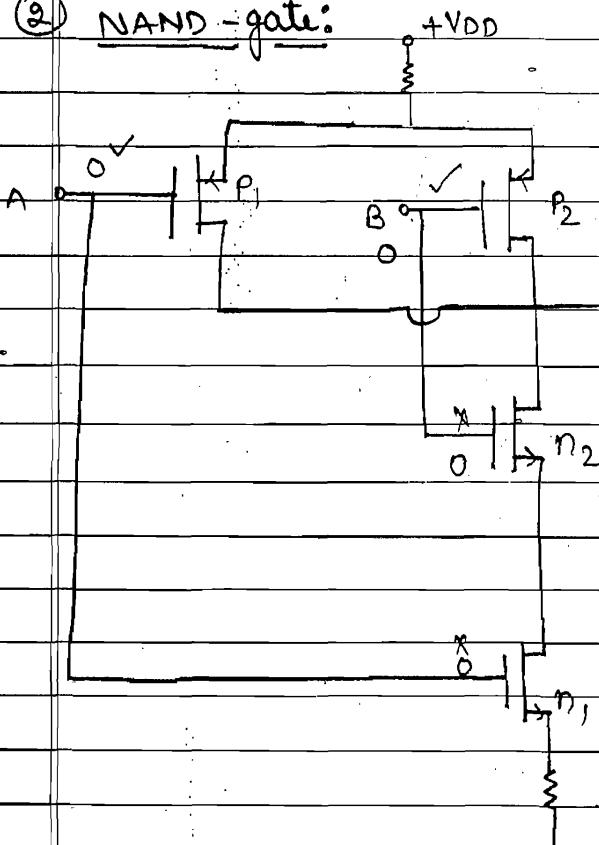


① NOT-gate



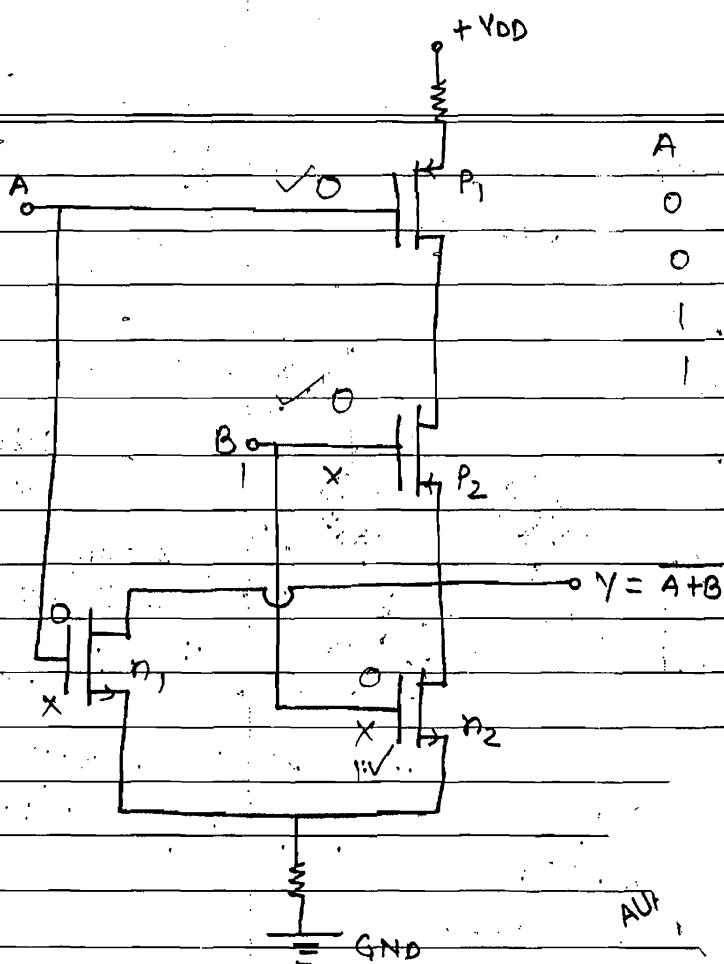
A	P	n	y
0	✓	x	+VDD
1	x	✓	0 GND

② NAND-gate:

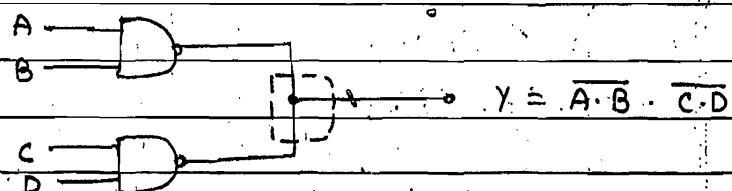
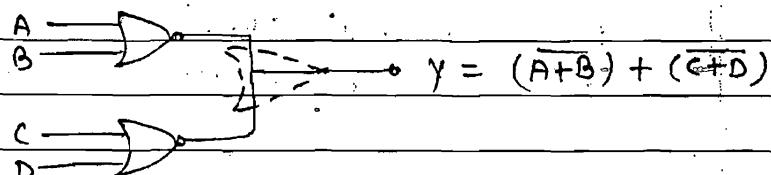


A	B	P <sub>1</sub>	P <sub>2</sub>	n <sub>1</sub>	n <sub>2</sub>	y
0	0	✓	✓	x	x	+VDD
0	1	✓	x	x	✓	+VDD
1	0	x	✓	✓	✓	+VDD
1	1	x	x	✓	✓	0 GND

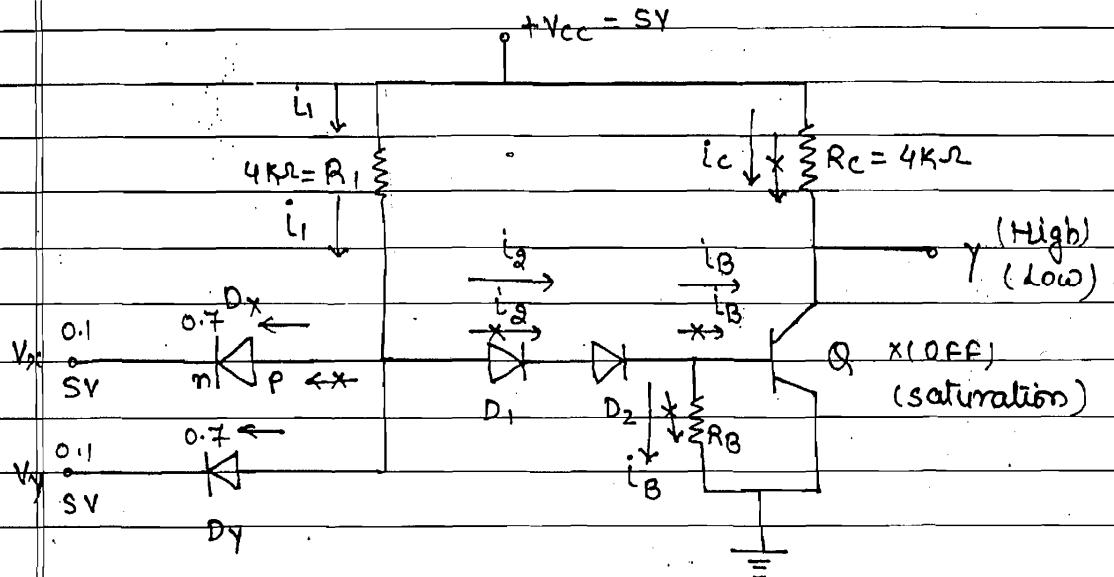
(3)

NOTE :

\* floating SIP in TTL is taken as: logic '1'

Wired logic :-a) wired AND logic -b) wired 'or' logic :-

## Diode

# Digital - Transistor - logic (DTL) :-

Given :  $V_V = 0.7V$

$$V_{BE(ON)} = 0.7V$$

$$V_{BE(sat)} = 0.8V$$

$$V_{CE(sat)} = 0.1V$$

$$\beta = 25$$

Solution  $\Rightarrow$

case(i)  $V_x = 0.1V \quad V_y = 0.1V$

$$i_1 = \frac{V_{cc} - V_1}{R_1} = \frac{5 - 0.8}{4k} = 1.05mA$$

$$i_B = i_R = 0$$

$\Rightarrow Q$  is off

$y = \text{High (5V)}$

case(ii)  $V_x = 0.1V \quad V_y = 5V$

Same as that of case(i)

$\Rightarrow Q$  is high off.

$\Rightarrow y = \text{high (5V)}$

case(iii)  $V_x = 5V$ ,  $V_y = 0.1V$

same as that of case(i):

$\Rightarrow Q$  is OFF.

$\Rightarrow y$  is high.

case(iv)  $V_x = 5V$ ,  $V_y = 5V$

$$i_1 = \frac{V_{cc} - V_x}{R_1} = \frac{5 - 2.2}{4} = 0.7mA$$

$$i_2 = i_1 = 0.7mA$$

$$i_R = \frac{V_{BE(\text{sat})}}{R_B} = \frac{0.8}{10k} = 0.08mA$$

$$i_R = 0.08mA$$

$$i_B = i_2 - i_R$$

$$= 0.7 - 0.08$$

$$i_B = 0.62mA$$

$$i_C = \frac{V_{cc} - V_{CE(\text{sat})}}{R_C} = \frac{5 - 0.1}{4k} = 1.22mA$$

$$i_C = 1.22mA$$

Saturation condi.

$$\frac{i_C}{i_B} < \beta$$

$$\frac{1.22}{0.62} < 25$$

$$1.96 < 25$$

$\Rightarrow Q$  is saturation

$\Rightarrow y$  is low

$\Rightarrow$  "NAND-Gate"

$V_x$	$V_y$	$y$
0	0	1
0	1	1
1	0	1
1	1	0

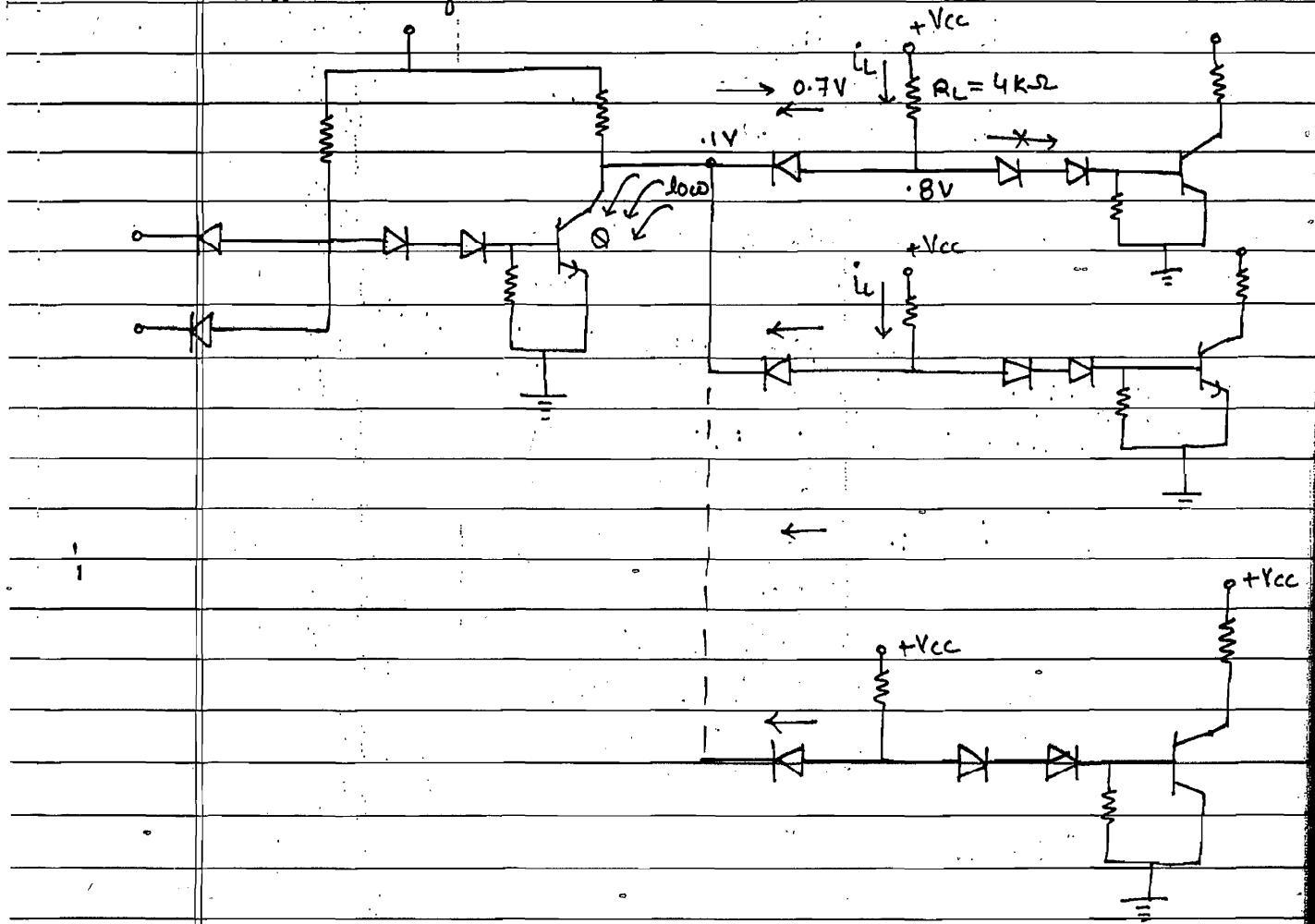
when on adding loads:

procedure to find FAN-OUT:

case(i): output of driving gate is low:

In this case the voltage across  $V_L$  for the loading gate is  $0.1 + 0.7 = 0.8V$  which cause a load current of  $i_L$  to pass through the collector of saturated transistor of the driving gate. By the no. of loads the collector current  $i_C$  now changes to  $i_C'$  i.e.  $i_C' = i_C + N \cdot i_L$  and new saturation condition will be  $\frac{i_C + N i_L}{i_B} \leq \beta \quad \{ \frac{i_C'}{\beta} \leq i_B \}$

where  $N \rightarrow$  fan-out.



saturation condn:  $i_C' = i_C + N i_L$

$$\Rightarrow i_C' \leq \beta i_B \quad \text{we know:}$$

$$i_B = 0.62 \text{ mA}$$

$$\Rightarrow i_C + N i_L \leq \beta i_B \quad i_C = V_{CC} - V_{CE(\text{sat})} / R_C$$

$$\Rightarrow \frac{1.22 + N (1.05)}{0.62} = 25 \quad i_C = 1.22 \text{ mA}$$

$$\Rightarrow N = 13 \quad \text{fan-out condition}$$

Case (ii) The OIP of the driving gate is "High"

In this case the JIP diode off loading gate is "reverse-biased" which causes a reverse saturation current passing through  $R_C$  of driving gate. By the no. of loads the current passing through  $R_C$  increases causing the voltage drop across  $R_C$  which OIP of driving gate alters from high v/g which leads to improper operation so there is a limit even in this case for fan-out. But from above two cases the least value shud be taken for overall fan-out.

Note:

"The OIP 'low' case of driving gate is preferable."

Procedure to find Noise-MARGIN:

case(i) : OIP of driving gate  $\rightarrow$  'Low'

$\rightarrow$  In this case the v/g drop at  $V_L \rightarrow 0.1 + 0.7 = 0.8 \text{ V}$  and other section of loading gate to become ON to require  $D_1 + D_2 + V_{BE} = 0.7 + 0.7 + 0.7 = 2.1 \text{ V}$  so that the difference of v/g :  $2.1 - 0.8 = 1.3 \text{ V}$  is the acceptable fluctuating v/g at the OIP of driving gate so noise margin is  $1.3 \text{ V}$ .

"In the same manner Noise-margin should be calculate for the OIP of driving gate is 'High'.

from above 2-cases the least-values is preferable for overall noise-margin.

power

# Procedure to find for dissipation:

$$P.D. = V_{cc} \cdot I_{cc}$$

$$P.D. = P.D(0) + P.D(1)$$

2

\*\*

Ques: what will happen in DTL ckt if one of the diodes  $D_1$  (or)  $D_2$  is removed?

→

fan-out increases, noise-margin reduces.

\*\*

Ques: what will happen if one more diode  $D_3$  is inserted in series with  $D_1$  &  $D_2$ ?

⇒

fan-out reduces and noise-margin increases.

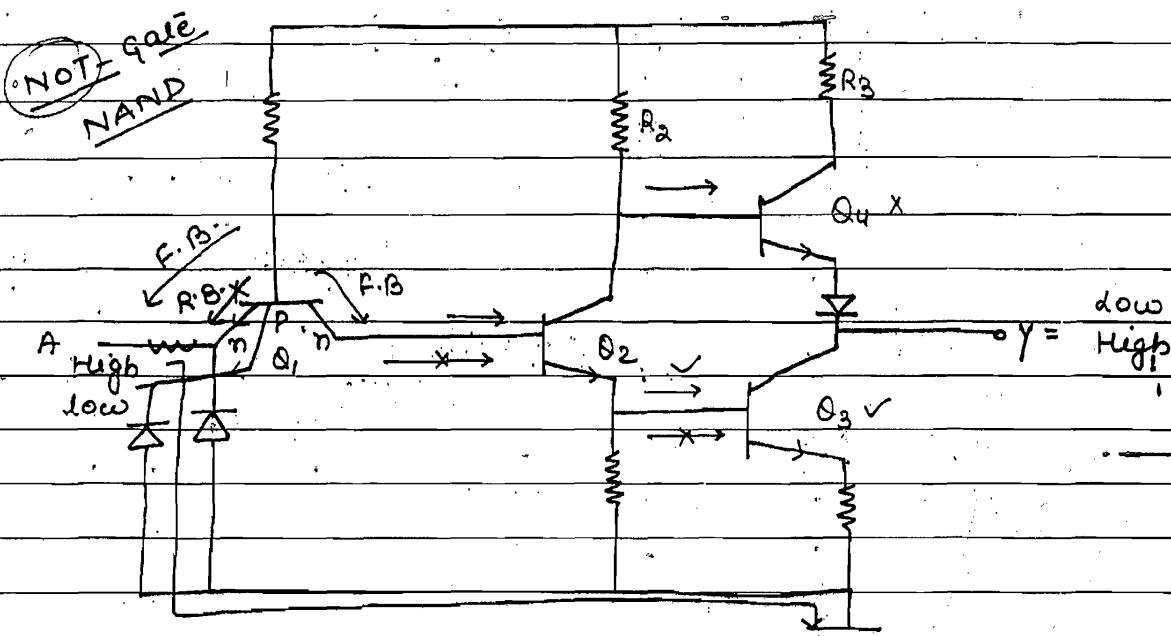
#

High-Threshold-logic (HTL) :-

;

→ if one of the diode in DTL is replace by zener diode, known as HTL. bcoz noise margin level will be increase.

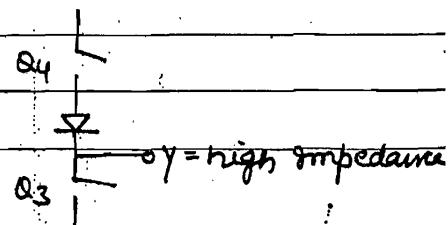
## Transistor-Transistor-Logic (TTL) :-



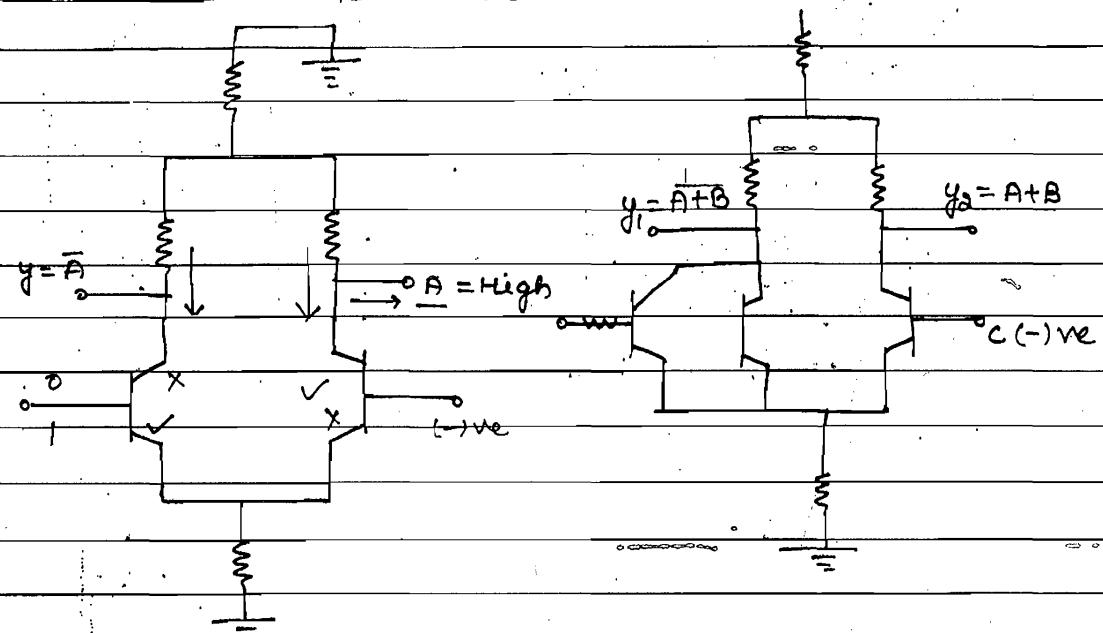
- ⇒ Diode D<sub>1</sub> is used to keep the input Transistor safe from negative voltage fluctuations.
- ⇒ Diode D<sub>2</sub> is used for proper switching output vlg.

\* TTL Tech. is of three - types :

- Totem-pole TTL :- Q<sub>3</sub> and Q<sub>4</sub> in the given ckt will be 'ON' at a different time so above discuss TTL ckt is Totem-pole TTL.
- Open-collector TTL : If the O/P section for the collector of Q<sub>3</sub> is opened then the TTL is known as open collector TTL. For proper switching operation the resistor is used known as Pull-up resistor. Pull-up resi. also reduces fluctuations of power-supply.
- 3) Tri-state TTL : This has another enable switch it can give 3-states - logic '0'; logic '1' and third imp state is high-impedance state.



## ECL (Emitter-Coupled-logic) :-

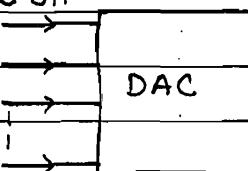


Becoz of two ground-connections and '1' negative supply power line fluctuations can be eliminated in ECL.

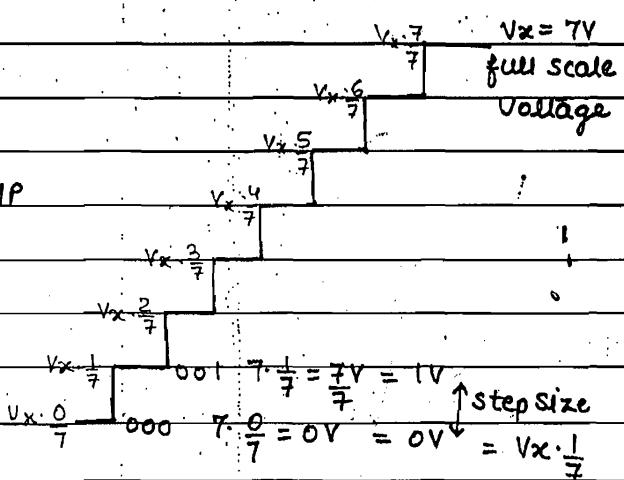
## # Digital to Analog Converter (DAC) :-

Parameters of DAC :-

Digital Input



Analog Output



$$\{ \text{No. of steps} = 2^n - 1 \}$$

No. of steps

$$= V_{fs} = 1 \text{ LSB}$$

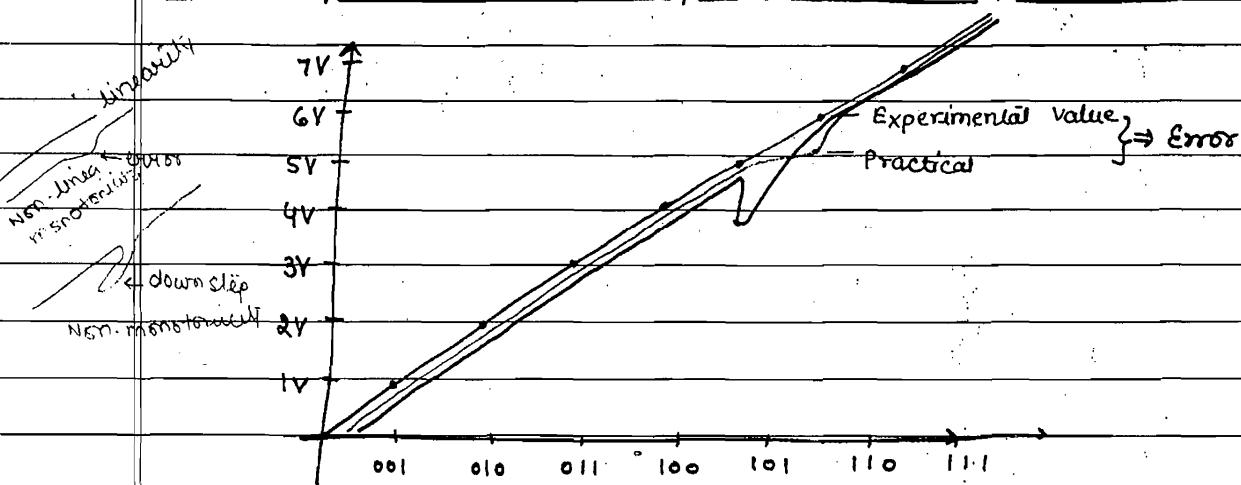
$$\Rightarrow V_o = K [2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \dots + 2^1 b_1 + 2^0 b_0] \quad 2^{n-1} \text{ value}$$

Decimal equivalent of binary data

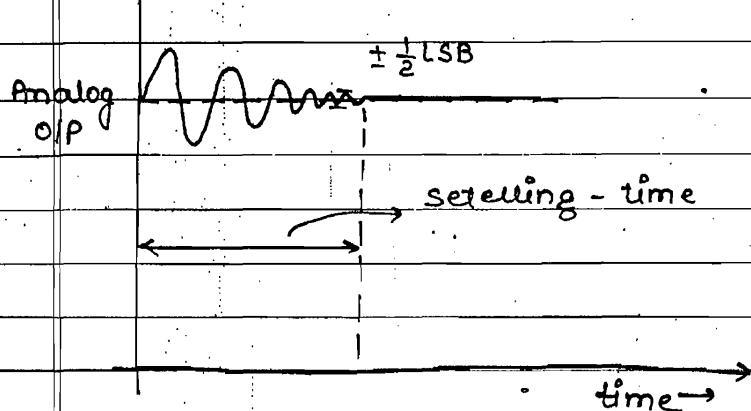
a) % Resolution :-

$$\% \text{ Resl'n} = \frac{1}{2^n - 1} \times 100\%$$

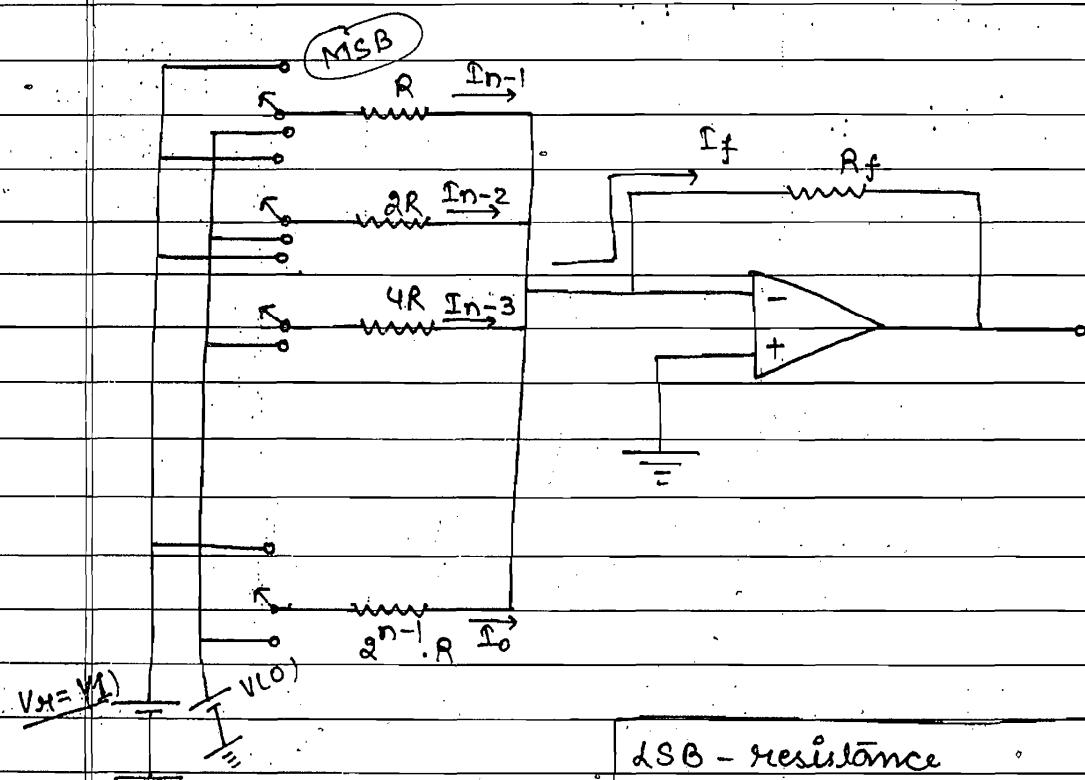
b) Linearity, Non-linearity, monotonicity, non-monotonicity:



c). Settling - time :



# Weighted Resistor Type DAC :-



LSB - resistance

$$= (\alpha^{n-1}) \text{ MSB - Resistance}$$

$$V_o = -I_f \cdot R_f$$

$$= -R_f [I_0 + I_1 + I_2]$$

$$= -R_f \left[ \frac{V_H}{2^2 \cdot R} + \frac{V_H}{2^1 \cdot R} + \frac{V_H}{2^0 \cdot R} \right]$$

$$= -\frac{R_f \cdot V_H}{R} \left[ \frac{1}{2^2} + \frac{1}{2^1} + \frac{1}{2^0} \right]$$

$$= -\frac{R_f \cdot V_H}{R} \left[ \frac{1}{2^{n-1}} + \frac{1}{2^{n-2}} + \dots + \frac{1}{2^1} + \frac{1}{2^0} \right]$$

$$\therefore = -\frac{R_f \cdot V_H}{2^{n-1}} \left[ 1 + 2^1 + \dots + 2^{n-1} \right]$$

$$= -\frac{V_H \cdot R_f}{2^{n-1} \cdot R} \left[ 2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \dots + 2^1 b_1 + 2^0 b_0 \right]$$

$$V_o = K \left[ 2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \dots + 2^1 b_1 + 2^0 b_0 \right]$$

decimal equi. binary data

$$\text{where ; } K = \left| \frac{-V_H \cdot R_f}{2^{n-1} R} \right|$$

P.N.76

Ques 6

$$\frac{1}{2^4} \left( 1 + \frac{7}{15} \right) (10)$$

Ques ②

$$V_{\text{in}} = 20 \text{ V}$$

11011011 ← 8 bit

$$V_o = K [\text{binary data}]$$

$$V_o = K [219]$$

$$V_o = K [11111111]$$

$$20 = K [255]$$

$$K = \frac{20}{255}$$

$$\rightarrow V_o = \frac{20}{255} \times 219$$

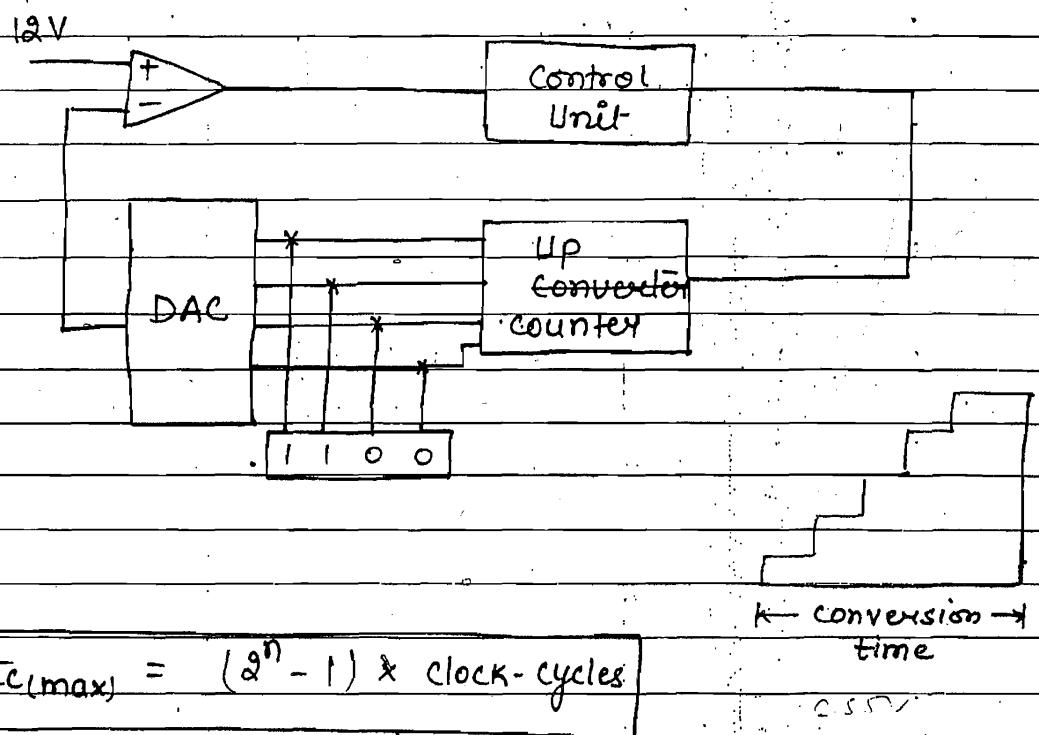
$$V_o = 17.2 \text{ V}$$

8.

ADC :-

Counter-Type-ADC :-

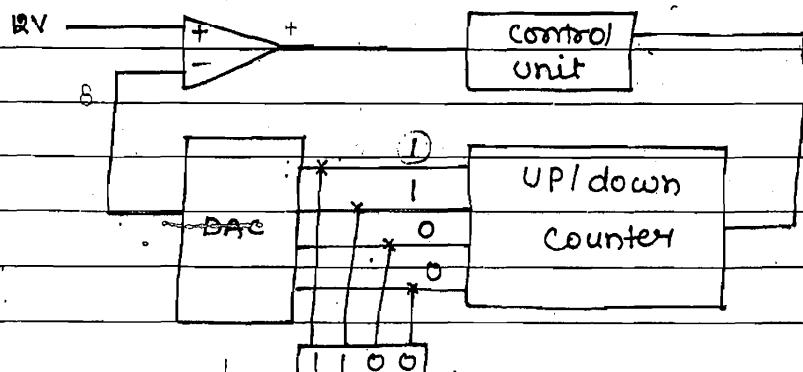
Ramp-Type-ADC :-



\*\*\*

Operation :- P.N. 99 (theory book)

2). Successive Approximate ADC :-



\*\*\*

operation:

(P.N. 100  
Theory book)

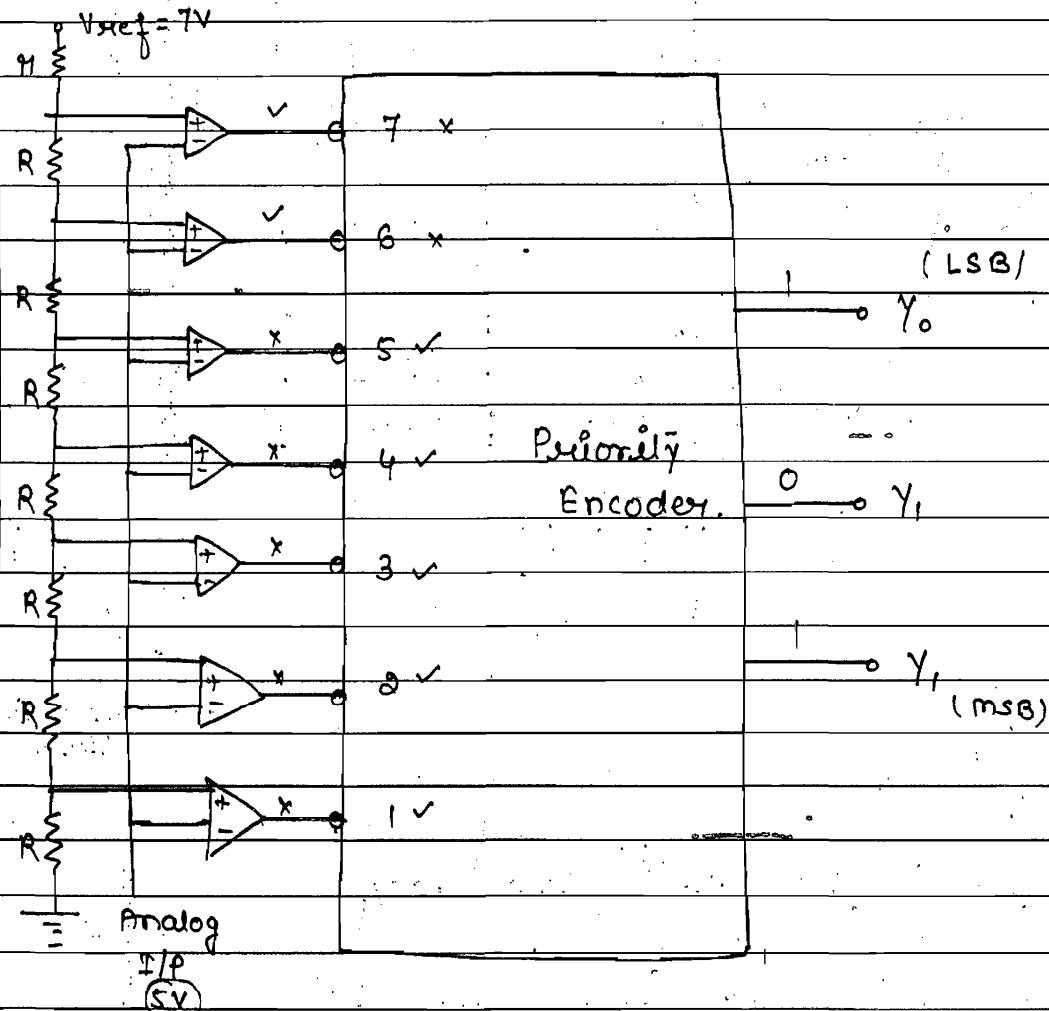
$$t_c(\max) = N \times \text{clock-pulses (cycles)}$$

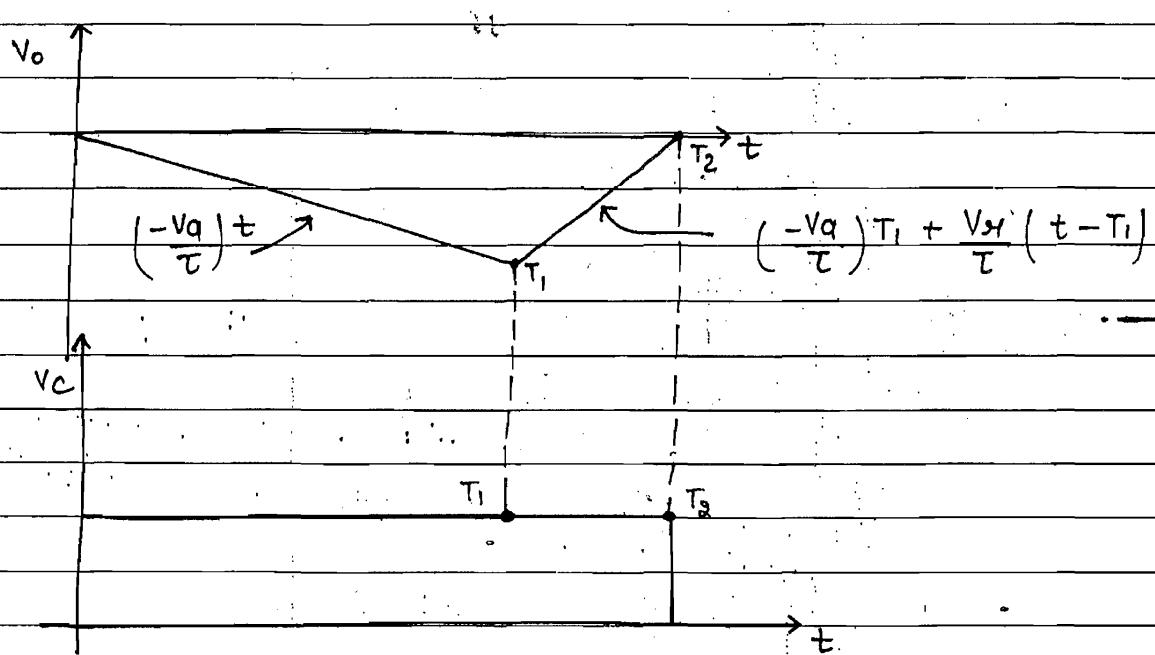
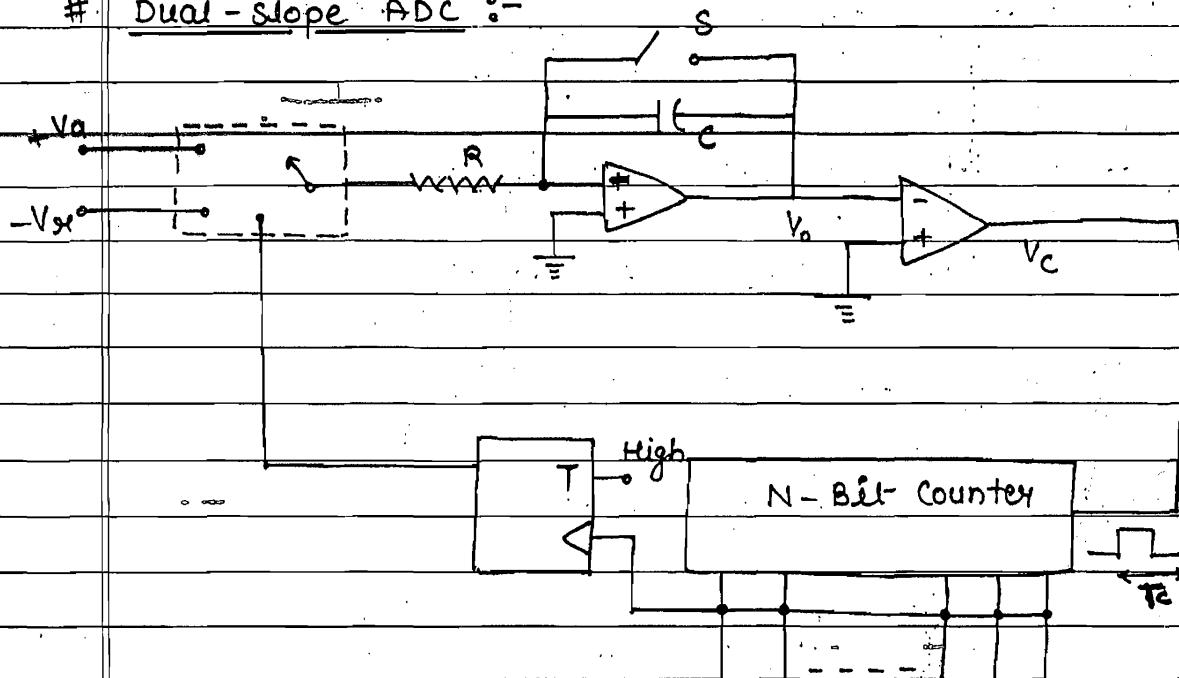
↳ no. of bits

\* Flash-Type ADC :-

\* Simultaneously Type ADC :-

\* Parallel Type ADC :-



# Dual-Slope ADC :-

$$V_o = -\frac{1}{\tau} \int_0^t V_a dt$$

$$= -\frac{1}{\tau} V_a (t)$$

at  $t = T_1$

$V_o = -\frac{V_a}{\tau} (T_1)$
---------------------------------

$$V_o = -\frac{V_a}{T} (T_1) + \left\{ -\frac{1}{T} \int_{T_1}^t (-V_r) dt \right\}$$

$$V_o = -\frac{V_a}{T} T_1 + \frac{V_r}{T} (t - T_1)$$

$$\text{at } t = T_2 \Rightarrow V_o = 0$$

$$0 = -\frac{V_a}{T} T_1 + \frac{V_r}{T} [T_2 - T_1]$$

$$\frac{V_a}{T} T_1 = \frac{V_r}{T} [n \cdot T_c]$$

$$V_a [\alpha^N] = V_r \cdot n$$

$$V_a = \frac{V_r}{2\pi \cdot n}$$

$$\text{If } V_a = \alpha^N$$

$$V_a = n$$

$$t_c(\max) = T_2$$

$$t_c(\max) = T_1 + n T_c$$

$$= \alpha^N T_c + n T_c$$

$$t_c(\max) = (\alpha^N + n) T_c$$

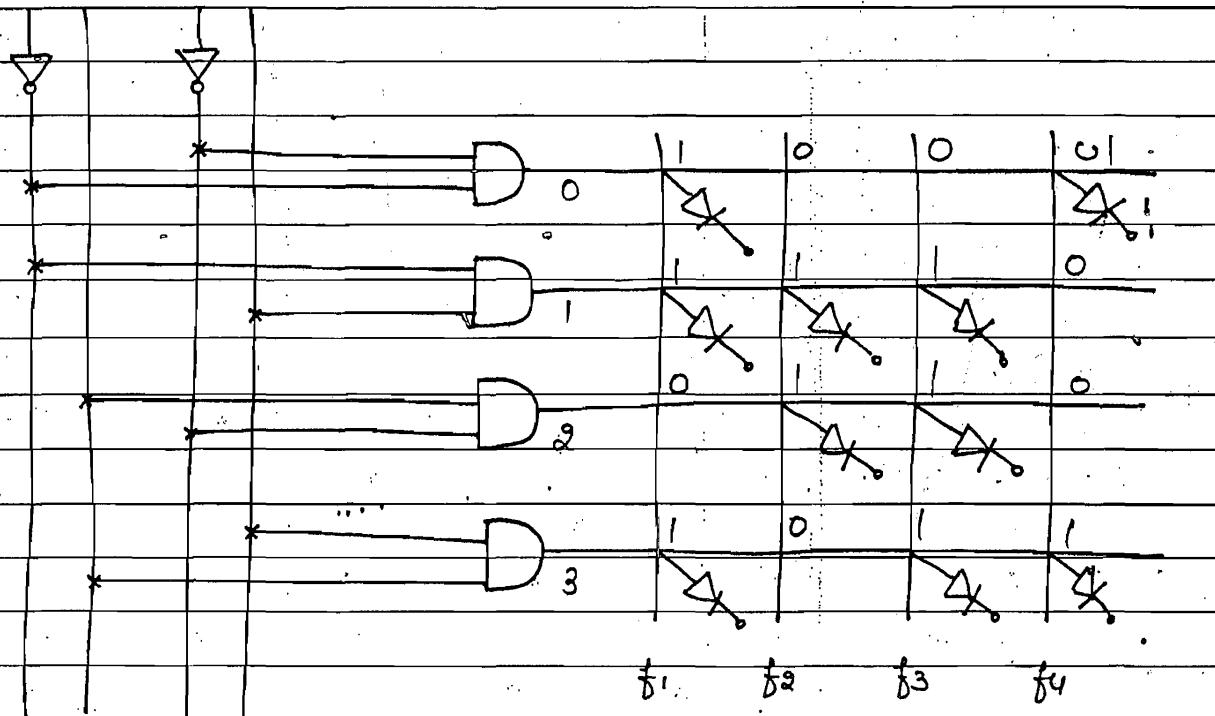
$$T_1 = \alpha^N T_c$$

$$T_2 - T_1 = n \cdot T_c$$

$n \Rightarrow$  no. of bits

$n \Rightarrow$  no. of counts

NOTE: Dual-slope ADC is very slow but it is very much accurate.

# MEMORY - DESIGNING :-

$$f_1 = \sum m(0, 1, 3)$$

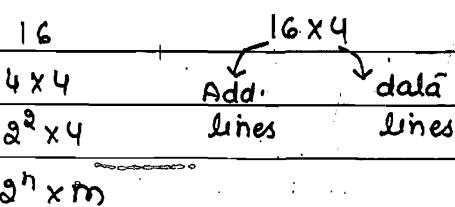
$$f_2 = \sum m(1, 2)$$

$$f_3 = \sum m(1, 2, 3)$$

$$f_4 = \sum m(0, 3)$$

$n \rightarrow$  select-lines

$m \rightarrow$  data-lines



A B C f<sub>6</sub> f<sub>5</sub> f<sub>4</sub> f<sub>3</sub> f<sub>2</sub> f<sub>1</sub> 138  
classmate

Date \_\_\_\_\_  
Page \_\_\_\_\_

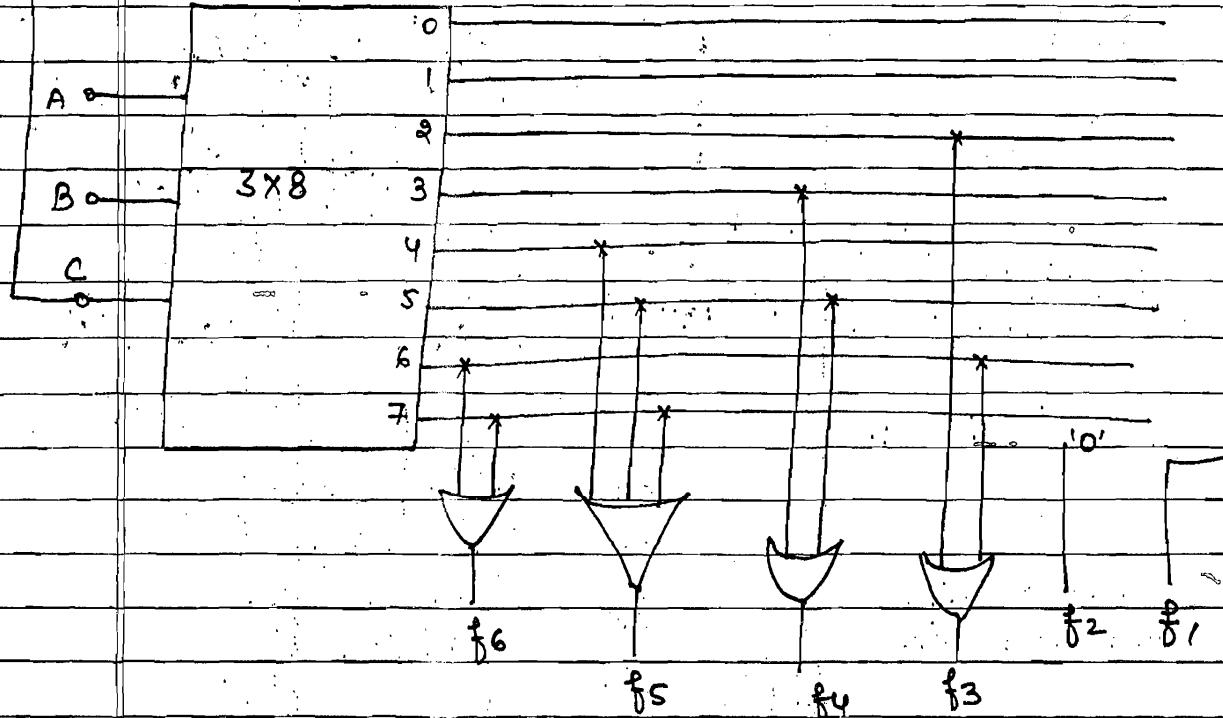
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
2	0	1	0	0	0	0	1	0	0
3	0	1	1	0	0	1	0	0	1
4	1	0	0	0	0	0	0	0	0
5	1	0	1	0	1	1	0	0	1
6	1	1	0	1	0	0	0	0	0
7	1	1	1	1	1	1	0	0	0

A

B

C

3x8



ROM-size

$$2^n \times m$$

$$2^3 \times 6$$

$$8 \times 6$$

$$= 48$$

Reduced - ROM

$$2^n \times m$$

$$2^3 \times 4$$

$$8 \times 4$$

$$= 32$$

## Master - Slave $JK$ flip-flop :-

- o It consists of 2 sections where the 1-section is triggered by the clock-generator and 2nd-section is triggered by the same clock. (G) but through an inverter.
- o So both the sections can't be triggered simultaneously. Final Q/P occurrence time, the Q/P clock pulse will be in the 'OFF' state so there's no repetition of toggle and race-around problem can be avoided.
- o Feedbacks are taken from second-section only. Theory-book diagram.

140