

PHOTOSTAT AND BOOK CENTER ADDRESS

NEAR MADE EASY F-21A, LADO SARAI, NEW DELHI-30

(PHOTOSTAT, BOOK CENTER)

F-53 BER SARAI, NEW DELHI-110016 (PHOTOSTAT)

SHOP NO. 5/123-A, SADQAZAB MARKET, IGNOU ROAD,
SAKET, NEW DELHI-30 (PHOTOSTAT, BOOK CENTER)**PHONE NO.**

9311989030, 8130782342

9560163471

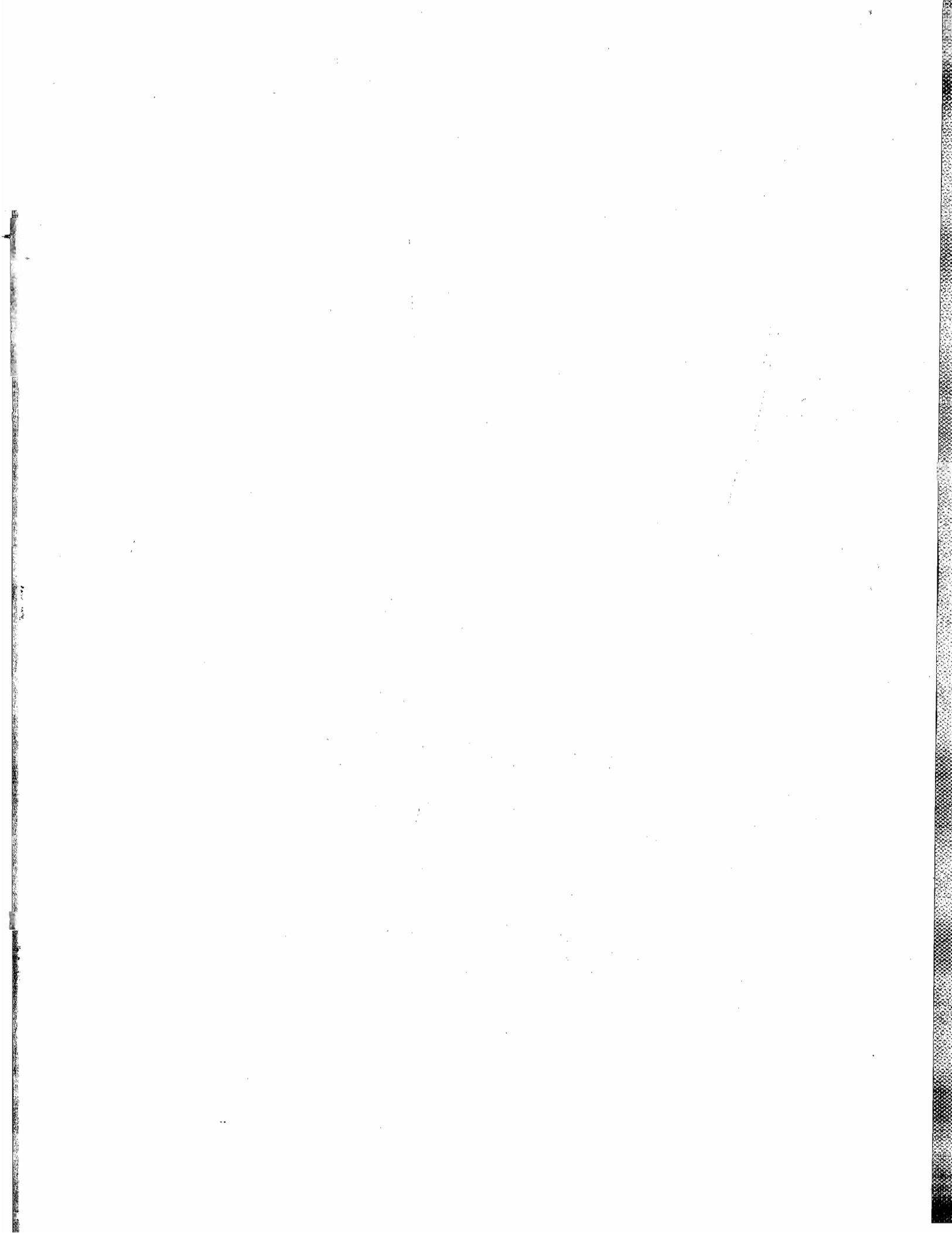
9654353111, 8595382884

HIND PHOTOSTAT AND HIND BOOK CENTER

NAME:-

SUBJECT:- DIGITAL ELECTRONICSINSTITUTE:- By - RAMESH SIR**PLEASE CONTACT FOR:-**

- PHOTOSTAT (LASER DIGITAL)
- PRINT OUT FROM COMPUTER
- SPIRAL BINDING, HARD BINDING
- TEST PAPER FOR PSU, GATE, IES
- ALL NOTES AVAILABLE
- ALL BOOK AVAILABLE



Digital

ANUPAM SHUKLA

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	1	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
x	y																

$f_0 = 0 \Rightarrow$ Null-operation.

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

MECS
2W

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

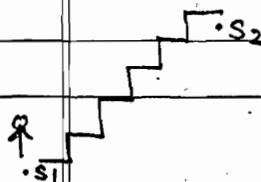
$y/x \Rightarrow$ [y but not x].

$f_5 \Rightarrow y$ [Transfer oper.] [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.



s_1	s_2	Bulb
0	0	0
0	1	1
1	0	1
1	1	0

$$f_7 = x + y \text{ OR-gate}$$

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

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

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

$$f_9 = x \odot y \text{ Ex-Nor}$$

$$= xy + x'y'$$

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

$$f_{11} = x + y' \text{ [Implication Op.]}$$

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

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

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

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

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

$$x \uparrow y$$

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

Logic-Gate Symbols :-

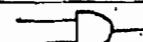
Buffer



NOT



AND



NAND



OR



NOR



EX-OR



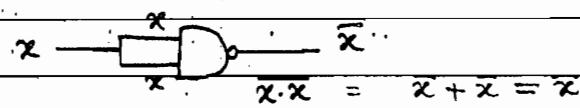
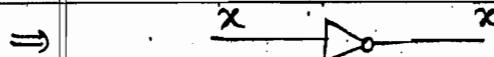
AUA

EX-NOR



NOTE:

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



NAND 4 NOR

NOT

1 1

AND

2 3

OR

3 2

EX-OR

4 5

EX-NOR

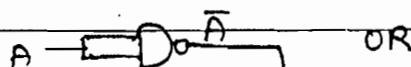
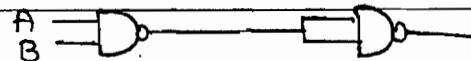
5 4

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

NOT



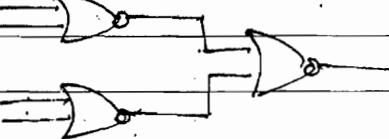
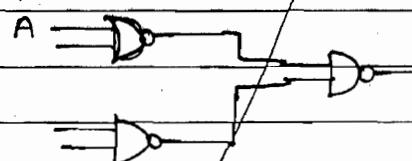
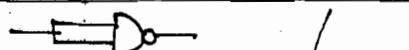
AND



OR

$$\begin{array}{c} \text{A} \\ \text{B} \end{array} \rightarrow \begin{array}{c} \bar{A} \\ \bar{B} \end{array} \quad \bar{A} \cdot \bar{B} = A + B$$

NOR



Practise:

NAND-Gates

$$\begin{array}{c} \text{A} \\ \text{B} \end{array} \rightarrow \begin{array}{c} \bar{A} \\ \bar{B} \end{array} \quad (\text{NOT})$$

(AND)

NOR-Gates

$$\begin{array}{c} \text{A} \\ \text{B} \end{array} \rightarrow \begin{array}{c} \bar{A} \\ \bar{B} \end{array}$$

(OR)

$$\begin{array}{c} \text{A} \\ \text{B} \end{array} \rightarrow \begin{array}{c} \bar{A} \\ \bar{B} \end{array} \quad \bar{A} \cdot \bar{B} = \bar{A} + \bar{B} = A + B$$

$$\begin{array}{c} \text{A} \\ \text{B} \end{array} \rightarrow \begin{array}{c} \bar{A} \\ \bar{B} \end{array} \quad \bar{A} + \bar{B}$$

$$\begin{array}{c} \text{A} \\ \text{B} \end{array} \rightarrow \begin{array}{c} \bar{A} \\ \bar{B} \end{array} \quad \bar{A} + \bar{B}$$

Duality:

Step: 1 Interchange the operators [. , +]

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

e AND-Laws

OR

eg.. ① $x \cdot \bar{x} = 0$

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

A.U.F

eg: ② $x + 0 = x$

$x \cdot 1 = x$

E.C.S

eg.. ③ $f = (A' \cdot B) + C$

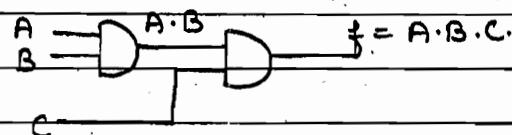
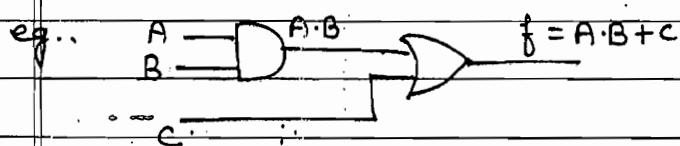
$f^D = ?$

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

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

Degenerative form :- when a 2-level logic gates system output is expressed with a single logic-gate then the 2-level logic gate sys. is known as degenerative form for single logic gate.

eg.. AND - AND/ is degenerative form for the 'AND' - gate.

Non-degenerative form :-

E.G.S

A.U.F

Short-cut -Duality

AND - OR

OR - AND

NAND - NOR

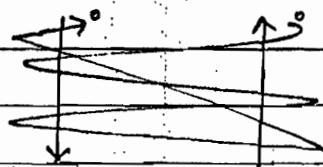
NOR - NAND

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

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

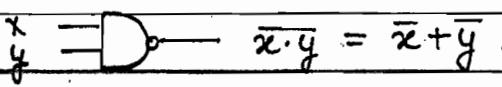
② $x \rightarrow \text{NOR} \rightarrow \bar{x} + \bar{y}$

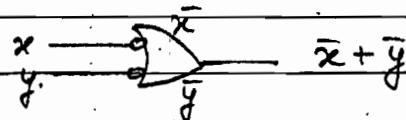
$$\bar{x} + \bar{y} = \bar{x} \cdot \bar{y}$$

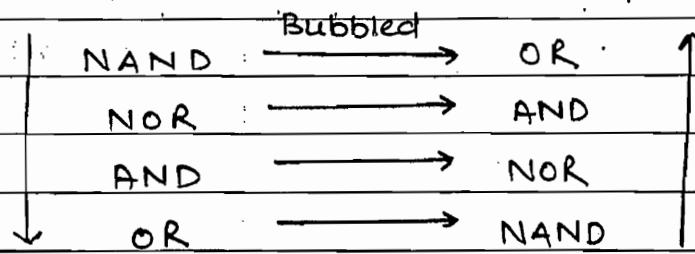
$$\begin{array}{ccc} x & \rightarrow & \text{OR} \\ y & \rightarrow & \bar{x} + \bar{y} \\ & & \bar{x} \\ & & \bar{y} \end{array}$$

Bubbled-OR

Alternative - logic - Gates :-

e.g..  $x \cdot y = \bar{z}$



Short-cut :-Positive , Negative - logics :-+ve - logicHigh \rightarrow 1Low \rightarrow 0-ve - logicHigh \rightarrow 0Low \rightarrow 1

e.g.. $-7V$ } \Rightarrow +ve - logic
 $-2V$

(+) '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

 \therefore -ve AND = +ve OR (Vice-versa) \therefore -ve NAND = +ve NOR (Vice-versa)

Ex-OR specialities :-

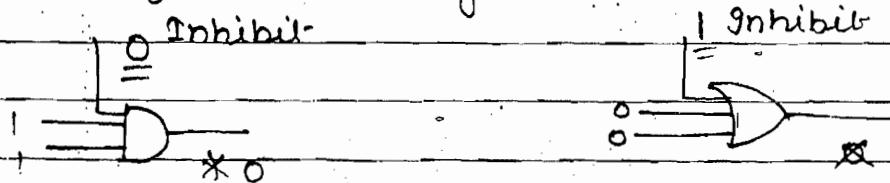
$x \oplus x$	$= 0$	\uparrow
$x \oplus x'$	$= 1$	
$x \oplus 1$	$= x'$	
$x \oplus 0$	$= x$	

$$\begin{array}{c} \overline{x \oplus y} \\ x' \oplus y \\ x \oplus y' \end{array} \quad \Rightarrow \quad x \odot y$$

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

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

Inhibiting the logic-gates :-



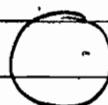
Short-Cut :-

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

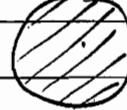
tg

Venn-Diagramm :-

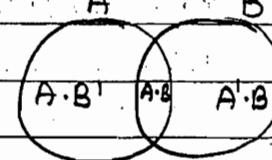
$$A = 0$$



$$A = 1$$

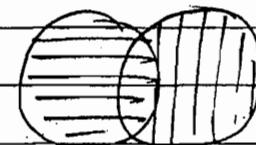
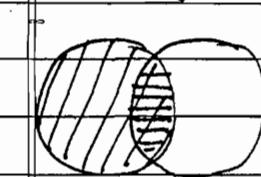


$$A + B = \bar{A} \cdot \bar{B}$$

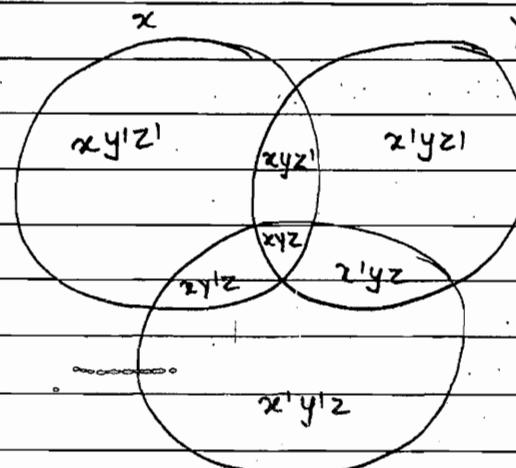


B

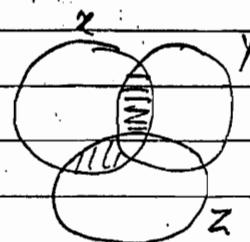
$$\Rightarrow x + xy = x$$



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



e.g. ①



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

LOGIC-MINIMIZATION :-

Boolean-draws :-

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 :-

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

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

Consensus - Theorem :-

A variable is associated with some variable and its complement is asso. 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.

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

... \downarrow
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 D lateral Rule (RLR) :-

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

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

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

Absorption - law :-Absor' n - law :-

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

$$\rightarrow x + xy = x$$

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

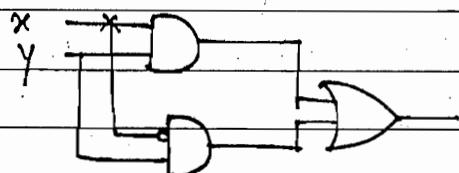
SOP :-

- (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

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

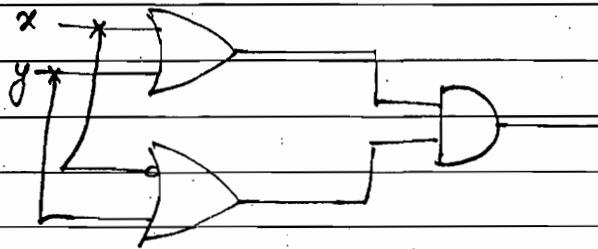
POS :-

- (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

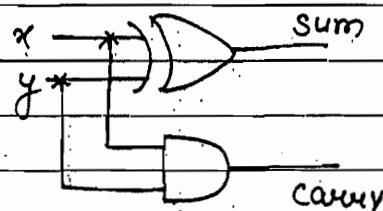
0 0 0 0

0 1 0 1

1 0 0 1

1 1 1 0

(carry)
(x.y)



$$\boxed{\text{Minterm} = \text{Maxterm}}$$

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

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

$\rightarrow f = (x+\bar{y})(\bar{x}+\bar{y}) \dots$ complementⁿ of 3ndiv.

⇒ This can also be written from the table with 'new-zeros'. 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:② ⇒ $f = (\bar{x} \cdot y) + (x \cdot y) \dots$ (complementⁿ of 3ndiv.)

Home-work :

Workbook - Problems : P.No. 50

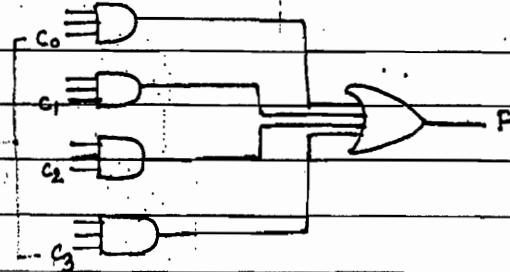
Q:12 Let $f(A, B) = A \oplus B$

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

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

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

Q:13



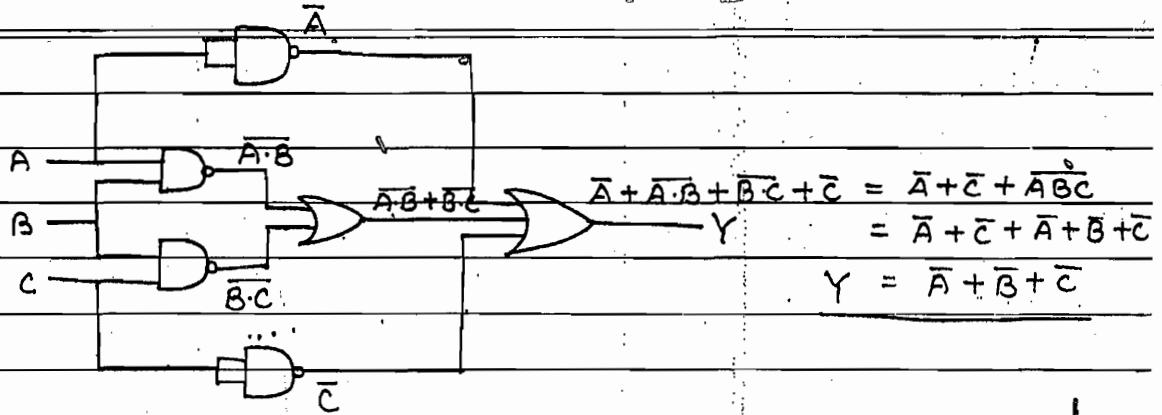
EX-OR is ALGEBRAICALLY

$$\Rightarrow C_1 \oplus C_2 = C_1 \cdot C_2$$

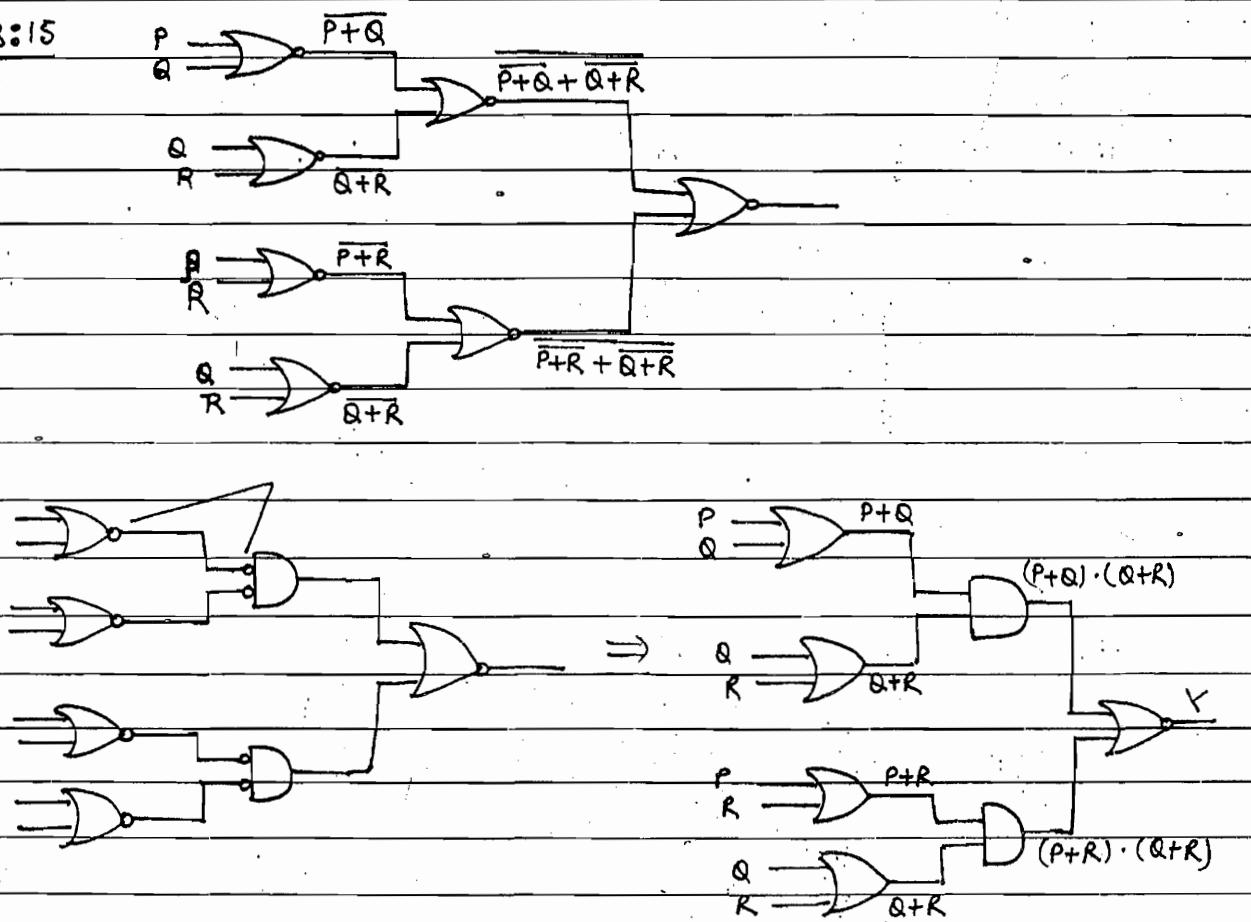
$$\Rightarrow C_1 \cdot C_2 = C_1 \oplus C_2$$

(C1, C2)

Ques: 14



Ques: 15

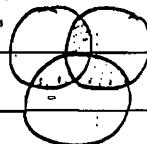


Y = (P + Q) · (P + R) + (P + R) · (Q + R)

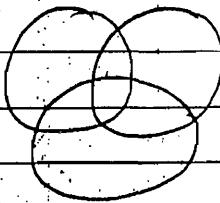
P + Q + R = P + Q + R

P + Q + R = P + Q + R

Ques: 16



Ques: 18



Ques: 20

A
B

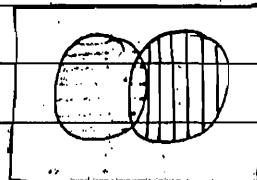
$$\gamma = A \cdot \bar{B}$$

Ques: 21

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

Ques: 24

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



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

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

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

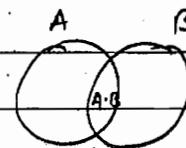
$$f_3 = \Sigma m(3, 4, 6, 7)$$

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

$$f_1 \cdot f_2 + f_3 = f = f_1 f_2 + f_3$$

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

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



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

$$A \rightarrow D \rightarrow f = ? \quad = A \oplus B = A \cdot B + A \bar{B}$$

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

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

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

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

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

A'	A	A'			A'
B'	B	B'			
					B' B

	B'	B
A'	$A' \cdot B'$	$A' \cdot B$
A	$A \cdot B'$	$A \cdot B$

Notes :

* Neighbouring cell should be 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 minterm. They are to be read as they are becoz they can't be combined.

3. Check for those 1s which are adjacent to neighbouring 1s.
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
$f \notin A, B$	\bar{A}	$\bar{A}\bar{B}$	$\bar{A}B$
	A	$A\bar{B}$	AB

		$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$f(A B C)$	\bar{A}	$\bar{A}\bar{B}\bar{C}_0$	$\bar{A}\bar{B}C_1$	$\bar{A}B\bar{C}_3$	$\bar{A}B\bar{C}_2$
	A	$A\bar{B}\bar{C}_4$	$A\bar{B}C_5$	ABC_7	$A\bar{B}\bar{C}_6$

		$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$f(A B C)$	\bar{C}	$\bar{C}\bar{A}\bar{B}_0$	$\bar{C}\bar{A}B_2$	$\bar{C}AB_3$	$\bar{C}A\bar{B}_4$
	C	$C\bar{A}\bar{B}_1$	$C\bar{A}B_3$	$CA\bar{B}_7$	$CA\bar{B}_5$

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

Ques.

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

↓	↓	↓	→
000	001	010	

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

1st method

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

\bar{A}	1	0	1	1	3	1	2
A	4	5	6	7	1	6	

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

2nd

$$\bar{A}\bar{B} \quad \bar{B}B \quad AB \quad A\bar{B}$$

\bar{C}	1	0	1	2	1	6	4
C	1	1	3	0	7	5	

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

eg..

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

$\bar{A}\bar{B}$	1	0	1	1	3	1	2
$\bar{B}B$	4	1	5	1	7	6	
AB	1	12	1	13	1	15	14
$A\bar{B}$	1	8	1	9	1	11	10

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

eg.

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

\bar{A}		1	1				
A		1	1	1	1	1	1

$$f = C$$

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

**

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

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

eg.. ①

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

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

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

EPI EPI X(R.P.I.)

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

eg.. ②

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

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

$\bar{A}\bar{B}$	0	1	1	3	2
$\bar{A}B$	4	1	1	7	6
AB	11	12	1	13	14
A \bar{B}	8	9	1	11	10

$$f = ABC + ACD + \bar{A}\bar{C}\bar{D}$$

+ BD

R.P.I.

* Implicant: The minterm 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.: E.P.I is P.I 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 our result known as R.P.I.

Selective Prime Imp.: It is the P.I. for which it was not 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	$B\bar{C}$	$f = \Sigma m(0, 1, 2, 6, 7)$
A	1	0	1	3	1
A	4	5	7	6	

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

$\bar{A}\bar{C}$ (0, 2)

$B\bar{C}$ (2, 6)

AB (6, 7)

P.I

P.I

E.P.I - (P.I)

(P.I)

(P.I)

(P.I)

S.P.I.

$\text{Imp.} \rightarrow S$ $P.I. \rightarrow 4$ $\text{Non-P.I.} \rightarrow 0$ $E.P.I. \rightarrow 2$

eg..(2)

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

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

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	EPI	EPI
A	4	5	(1) 7	6	(P.I.)	(P.I.)

DIAGONAL GROUPING :-

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

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

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

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

$f = A \odot B$

$f = A \oplus B$

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
\bar{A}	(1)		(1)	
A		(1)		(1)

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$	(1)		(1)	
$\bar{A}B$		(1)		(1)

$B[A \odot C] + B[A \oplus C]$

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

 $A \odot B \Rightarrow$ when on with same nature.

$+ B\bar{D}[A \odot C]$

 $A \oplus B \Rightarrow$ 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

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

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

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

POS forms in K-map :-

* Step: ① Operator change.

* Step: ② complement the given variable.

eg:-

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

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

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

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

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

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

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

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

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

SOP to SSOP [Canonical- form]:

Ques:

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

Method I

$$\begin{aligned} f &= \bar{A}B(C + \bar{C}) + \bar{A}B\bar{C} + AC(B + \bar{B}) \\ &= \bar{A}BC + \bar{A}B\bar{C} + \bar{A}B\bar{C} + ACB + AC\bar{B}. \end{aligned}$$

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

Method II

$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	BC			
\bar{A}	0	1	1	3	1	2
A	4	1	5	1	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	$+ A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + ABCD$
$\bar{A}B$	4	1	5	$+ ABC\bar{D} + A\bar{B}\bar{C}\bar{D} +$
$A\bar{B}$	12	13	15	$A\bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C}D +$
AB	18	19	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	0

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

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

eg.

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

without $\phi \Rightarrow \bar{A}BD$

with $\phi \Rightarrow BD$.

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

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

\bar{A}	0	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	1	2
A	4	1	x_5	x_7	6

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

Variable - Entered - Mapping :-

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

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$	
\bar{A}	0	D	D + D	D	2
A	4	5	D	7	6

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

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 eqn.

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

eg:-

	$\bar{y}\bar{z}$	$\bar{y}z$	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

Step: (1)

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

→ so need to consider in final grouping]

Step: (2)

	$\bar{y}\bar{z}$	$\bar{y}z$	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

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

↳ Redundant

B B

eg.	A	I	
	A	:	C

B B

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

Step : ②

A	I	X		$f = CB$
A		O		

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

5-variable K-MAP :-

8/12/13

Sunday

	\bar{A}	$\bar{B}\bar{C}$ will form Octet				A		
	$\bar{D}\bar{E}$	$\bar{D}E$	$D\bar{E}$	DE	$\bar{D}\bar{E}$	$\bar{D}E$	$D\bar{E}$	
$\bar{B}\bar{C}$	1	1	1	1				
$\bar{B}C$	4	5	7	6				
$B\bar{C}$	13	13	15	14				
$B\bar{C}$	8	9	10	10				

BCE

{will not form Octet as there are
{more than one literal changes}} $\bar{A}DE$ $A\bar{D}\bar{E}$ Identification 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 CKR 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.

$$\text{eg.. } f = \sum m(0, 1, 3, 7, 8, 9, 11, 15)$$

①	Group	Minterm	Variable			
			A	B	C	D
	0	0	0	0	0	0
	1	1	0	0	0	1
	2	8	1	0	0	0
	3	3	0	0	1	1
	4	9	1	0	0	1
	5	7	0	1	1	1
	6	11	1	0	1	1
	7	15	1	1	1	1

(2) Group

Minterm

Variables

A B C D

0

0, 1

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

 $\begin{array}{ccccccccc} \bar{B} & \bar{C} & \times & \times & \times & \times & \times & \times \\ \swarrow & \text{EPI} & & & & & & \\ \bar{B} & D & X & X & & X & X & \end{array}$
 $\begin{array}{ccccccccc} C & D & & & & & & & \\ \swarrow & \text{EPI} & & & & & & & \\ & & X & \times & & & X & & \times \end{array}$

** Note :

- If any pair (lower-order-grouping) is not coming under Quad-grouping, then those numbers won't be carried the 4th 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: Tabⁿ Method for POS is exactly same as that for SOP form.

- Treat the Maxterms as if they are the Minterms and complete the process.
- 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 $f = \Sigma M(0, 1, 4, 5, 9, 11, 13, 15, 16, 17, 25, 27, 28, 29, 31) + \phi(20, 21, 22, 30)$

** This Tabⁿ 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.

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

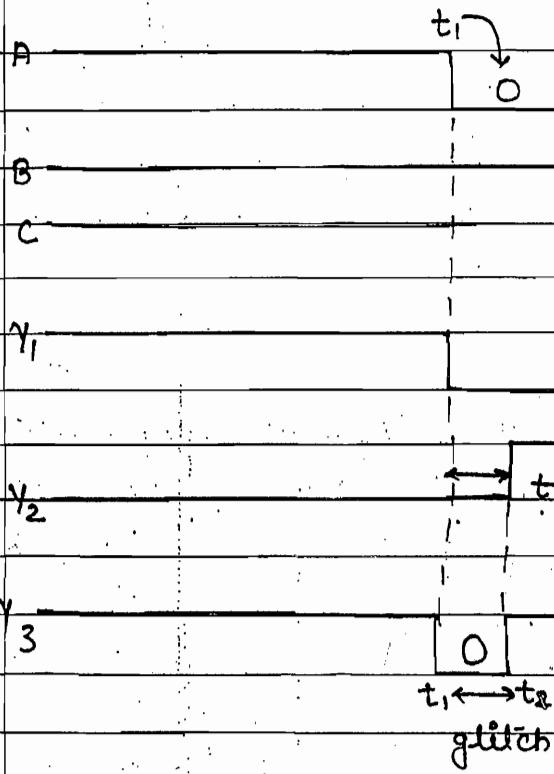
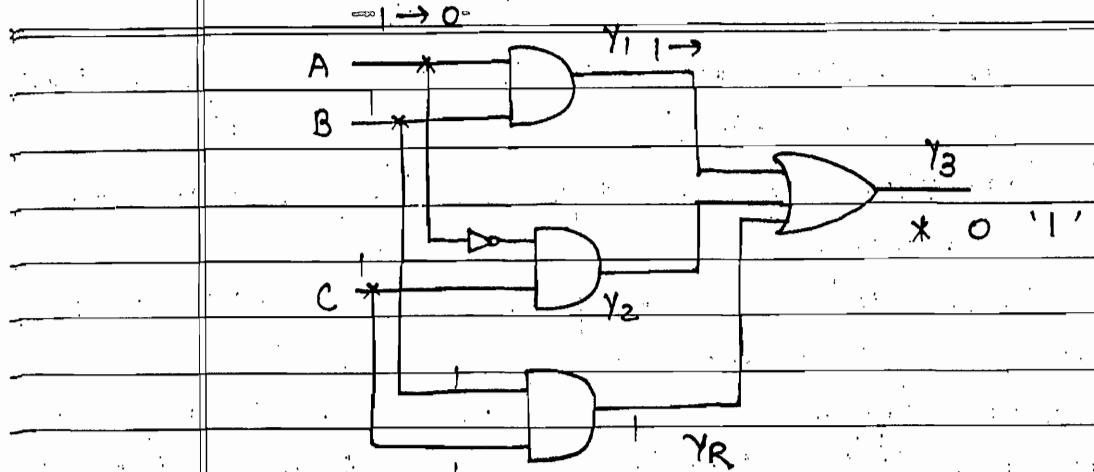
$\bar{B}\bar{C}$ $\bar{B}C$ $B\bar{C}$ BC

\bar{A}		1	1	
A			1	1

$$f = \bar{A}C + AB. \quad x$$

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

Redundant group



Types:

① static '1'

② static '0'

③ Dynamic

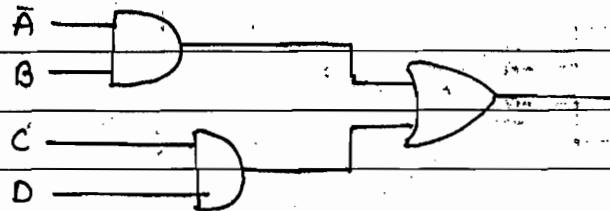
Implementation of Boolean-functions by Using NAND, NOR-gates only :-

(1)

$$f = A'B + CD$$

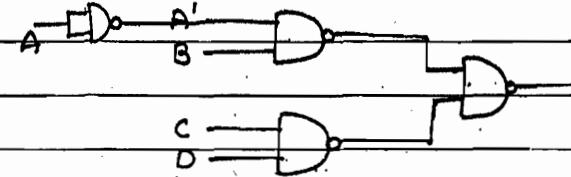
(NAND)

AND-OR



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

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

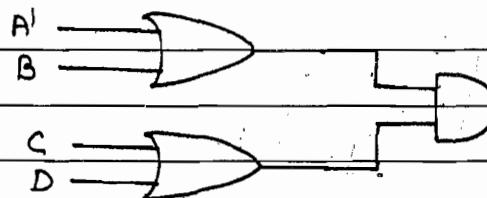


(2)

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

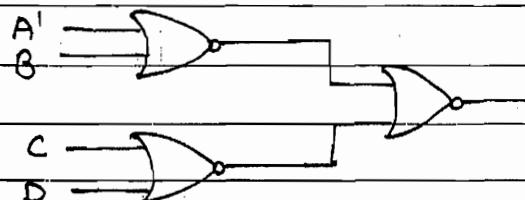
(NOR)

OR-AND



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

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

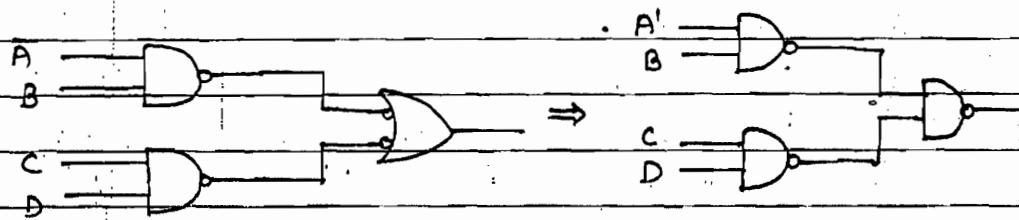


Method : ⑧

Step: ① AND (NAND)

Step: ② OR (Bubbled OR = NAND)

Step: ③ NOT (NAND)

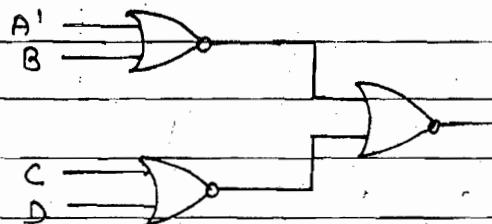


NOR :

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

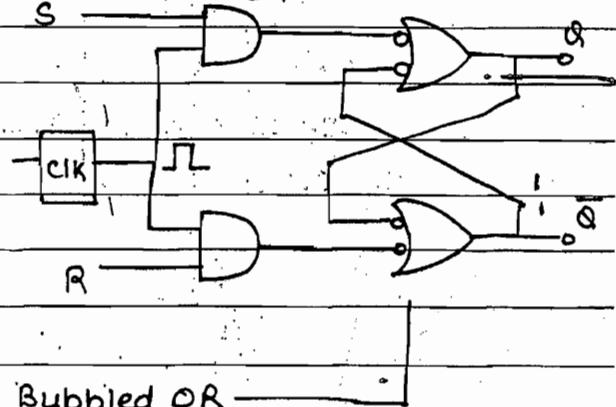
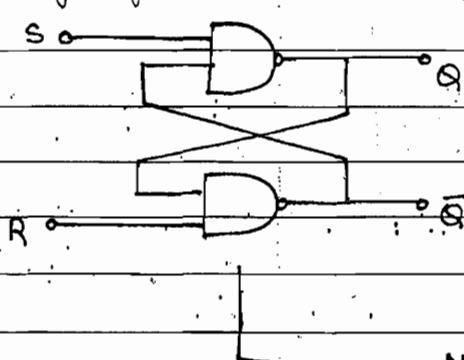
manega aur ise CLK ko hmesha high ke rakh-

clocked - SR - latch \Rightarrow

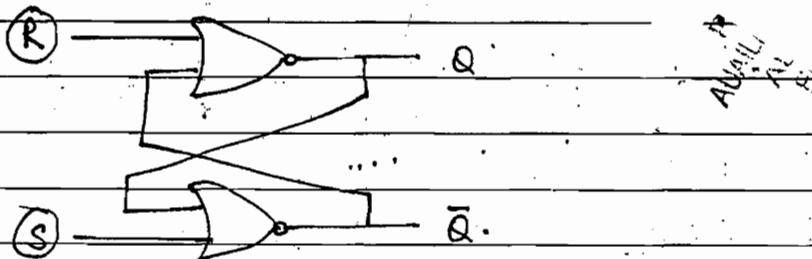
enge gki agar use har low kar denge to kitna

te sig high de, wo '0' hi ho jaega.

AND gate

NAND \Rightarrow Bubbled ORTable for Active - low latch :-

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

Active - High latch :-

CLK	S	R	Q^+
Low	0	0	NC
Low	1	0	1 Set
Low	0	1	0 Reset
Low	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)

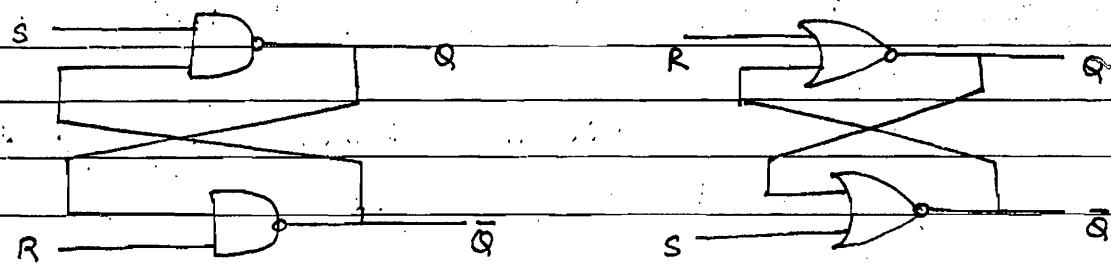


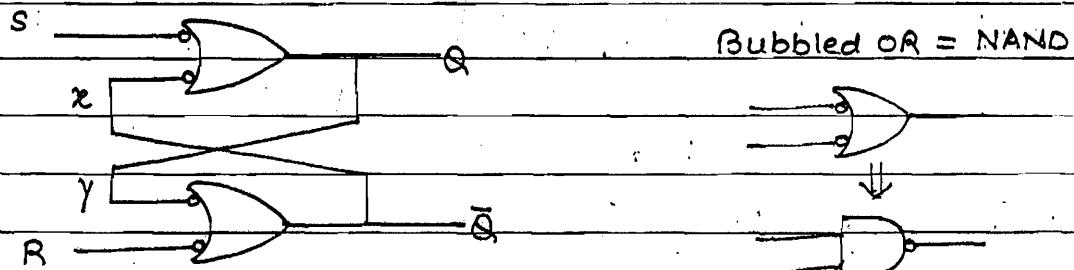
fig (a)

fig (b)

S → Set

R → 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 o/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 O/P will be $x=1 \& y=0$ and we get the next-state $Q^+ = 0$.

→ If initially $Q=1$, Then the feedback O/P 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 O/P must be operated until we get the stable O/Ps. 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 conducted for the stable O/Ps. 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

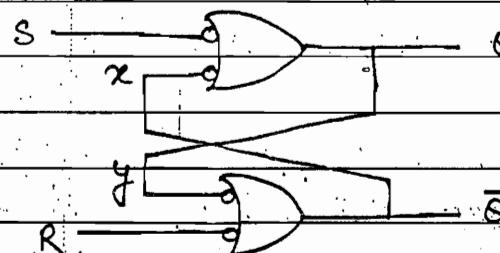
Sequential Circuit \Rightarrow In the sequential ckt's present O/P not only depends on the present S/G's it also depends on the past output.

- o Logic-gates with feedback connections behaves as sequential ckt's.

e.g. latches

- o Latches are 2-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 S/G's will be $x=0$ and $y=1$ then you'll get the next stage $Q^+ = 0$

if initially ' $Q=1$ ' , then the feedback S/G'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
 $S=D, R=1$, 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 galēd SR-latch.

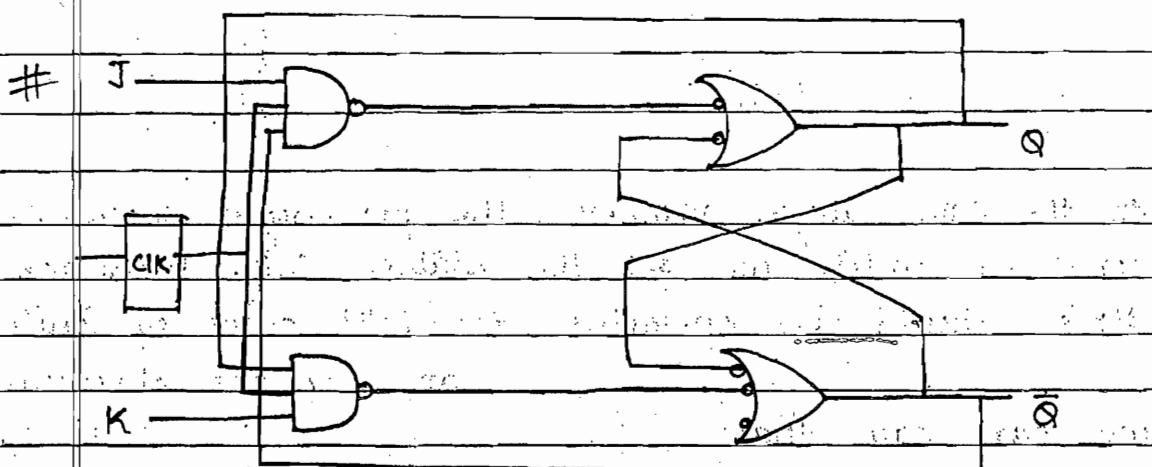
** The controlling can also be done by using clock-gener-alōr known as clocked SR FF.

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.

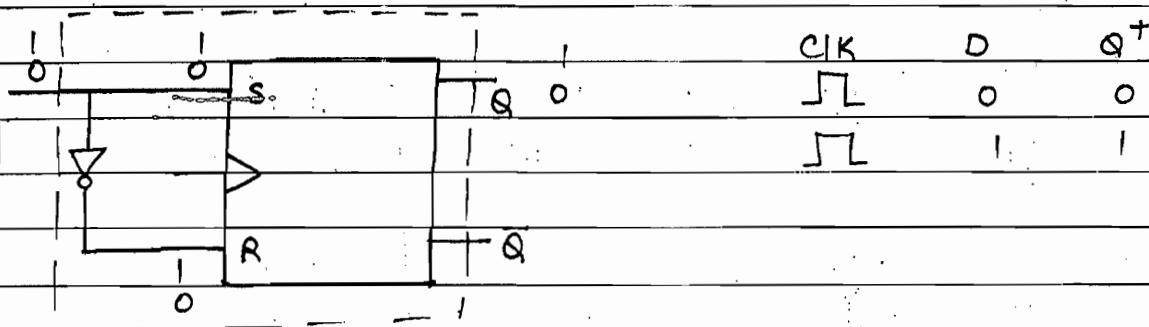
* External feedback should be operated only 1 time for single clock-pulse.



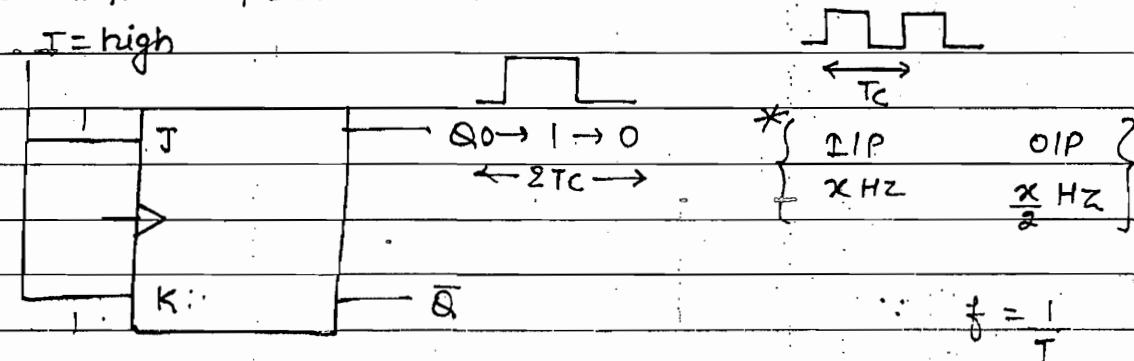
C.I.K	J	K	Q^+
0	0	0	NC
0	0	1	Set
1	0	1	Reset
1	1	1	\bar{Q} (Toggle)

* 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^+		$J'K'$	$J'K$	JK	JK'
0	0	0	0 NC		0	1	1	0
0	0	1	0 Reset-		0	0	1	1
0	1	0	1 Set-		0	1	0	1
0	1	1	1 \bar{Q} Toggle		1	0	1	0
1	0	0	1					
1	0	1	0					
1	1	0	1					
1	1	1	0					

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

$S'Q'$ $S'R'$ SR SR'

Characteristics of SR:

Q			X	1
Q	1		X	1

$Q \quad S \quad R \quad Q^+$

$$0 \quad 0 \quad 0 \quad 0 \quad NC \Rightarrow Q^+ = S + R'Q$$

0 0 1 0 Reset

0 1 0 1 set

0 1 1 X X

1 0 0 1

1 0 1 0

1 1 0 1

1 1 1 X

0 0

1 1

0 1

1 0

Q D Q⁺

$$0 \quad 0 \quad 0 \Rightarrow Q^+ = D$$

0 1 1

1 0 0

1 1 1

T' T

Q' 1

Q 1

Q T Q⁺

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.

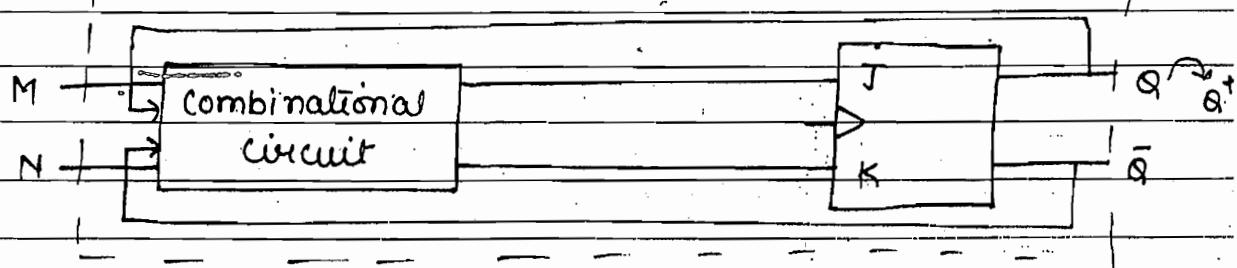
Transition or Excitation-Table: - It gives the info about all the possible way of G/Ps to get a particular O/P.

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

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

	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

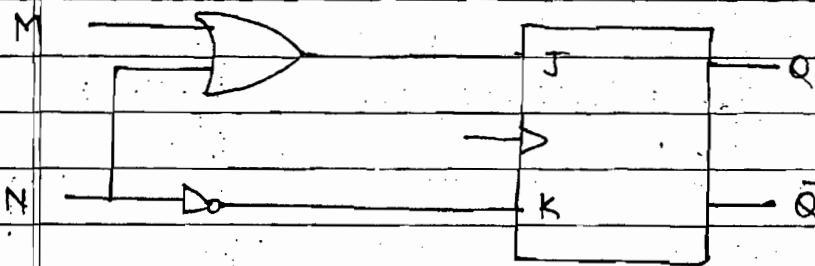


	Q	Q^+	J	K	
0	0	0	0	X	
0	1	1	1	X	
1	0	X	X	1	
1	1	X	0	0	

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'	
Q	0	1, 1, 3	1, 2		$J = M + N$
Q'	x ₄	x ₅	x ₇	x ₆	

	MN'	M'N	MN	MN'	
Q	x	x	x	x	$K = N'$
Q'	1		x	1	

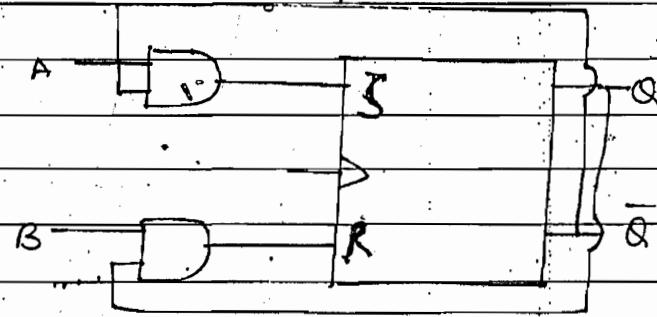


Using $\cdot \overline{SR} - F.F. \Rightarrow$ AUALL
A
F

Q	A_1	B	Q^+	S	R	O	O	O	x
0	0	0	0	0	0	x	0	1	10
0	0	1	0	0	0	x	1	0	01
0	1	0	1	1	1	0	1	1	x0
0	1	1	1	1	1	0			
1	0	0	1	1	x	0			
1	0	1	0	0	1				
1	1	0	1	x	0				
1	1	1	0	0	1				

$S'R'$	$S'R$	SR	SR'		
\bar{Q}	0	*	(13)	D_2	$S = A\bar{Q}'$
Q	x	4	5	*	$S = JQ'$

$S'R'$	$S'R$	SR	SR'		
\bar{Q}	x	0	x	3	$R = BQ$
Q	4	(15)	17	6	$R = KQ$



H.W.

Q.1 $S-R \Rightarrow JK$

$\Rightarrow T$

$\Rightarrow D$

Q.2 $J-K \Rightarrow SR$

$\Rightarrow T$

$\Rightarrow D$

Q.3 $D \Rightarrow S-R$

$\Rightarrow J-K$

$\Rightarrow T$

Q.4 $T \Rightarrow S-R$

$\Rightarrow J-K$

$\Rightarrow D$

Race-Around - Problem (Repetition)

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

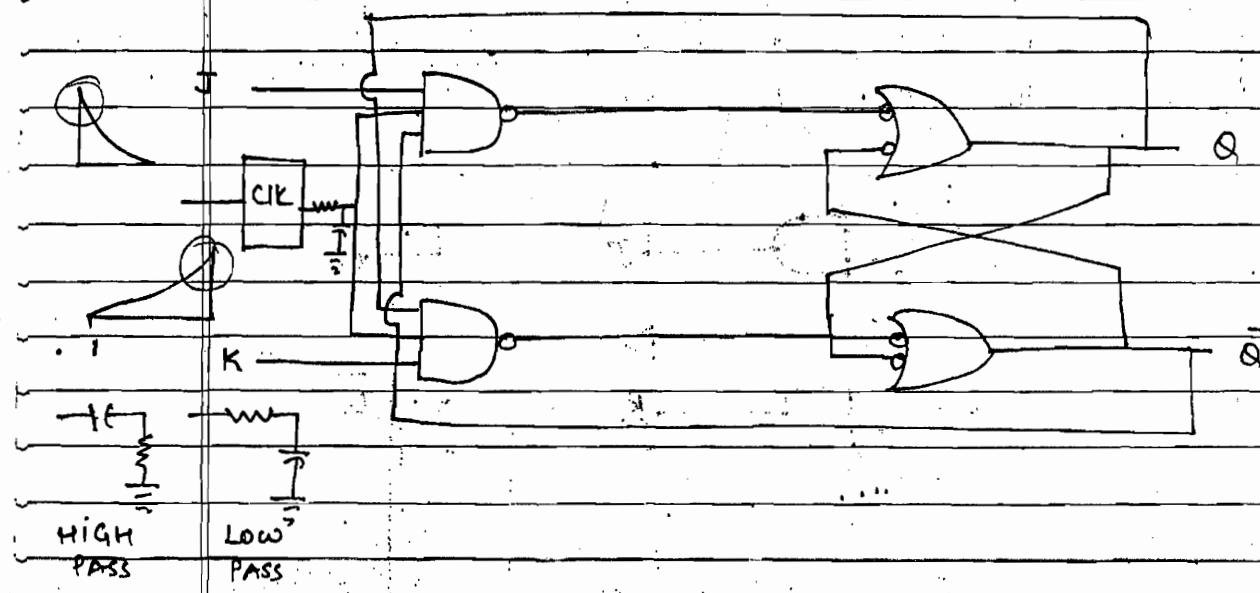
Reason: Q1P occurrence time Q1P CLK-pulse is in the ON-state only which causes one more time triggering - leads to repetition of Toggle known as RACE AROUND PROB.

Methods to avoid Prob. =

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

Note: Edge-Trigg. is two types -

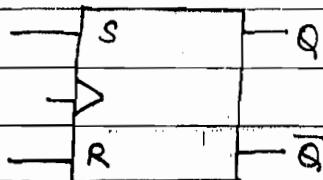
- A. Rising Edge Trigg. [Positive Edge Trigg.]
- B. Falling Edge Trigg. [Negative Edge Trigg.]



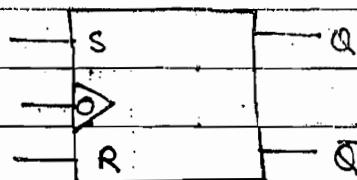
- * 3. By using Master slave flip-flop, Race-around problem can be avoided.

TIMING DIAGRAMS :

Negative Trigg.

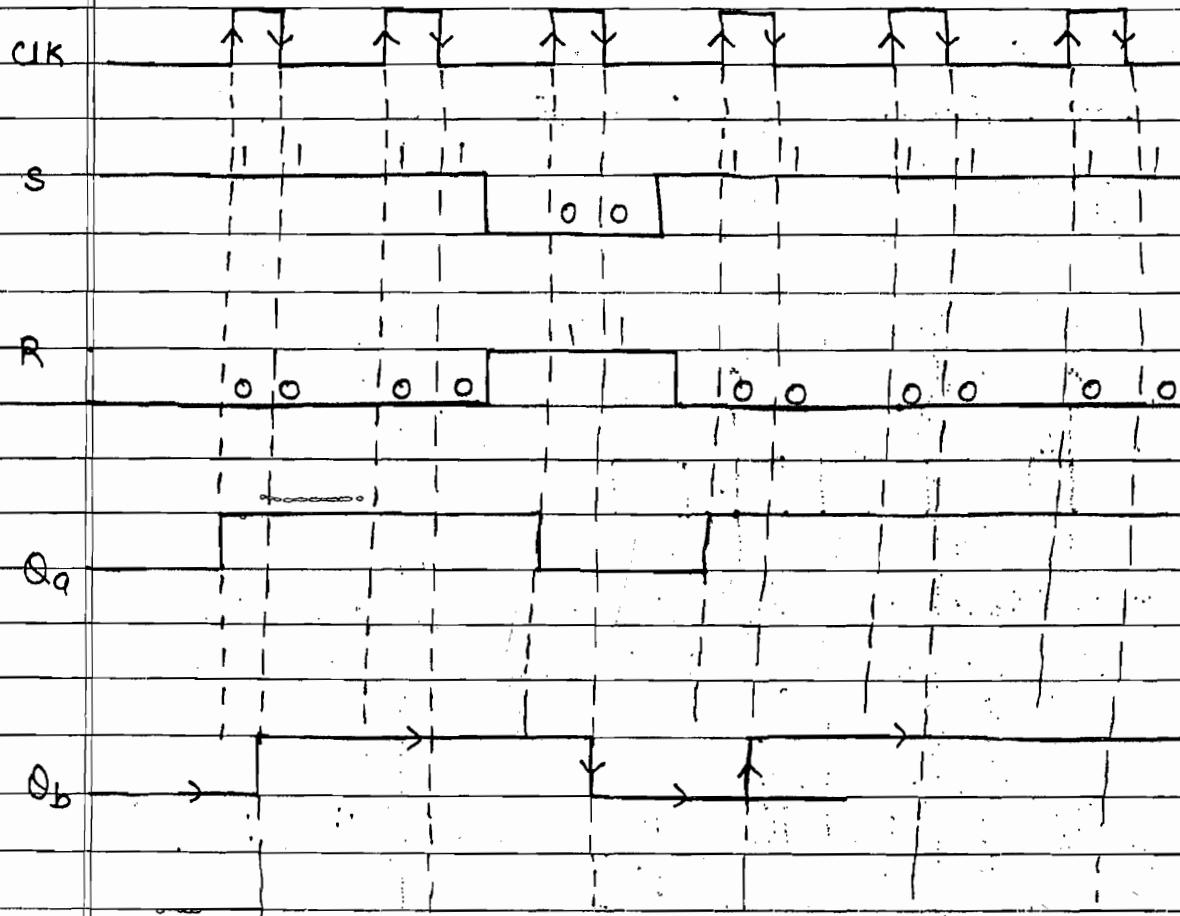


(a)

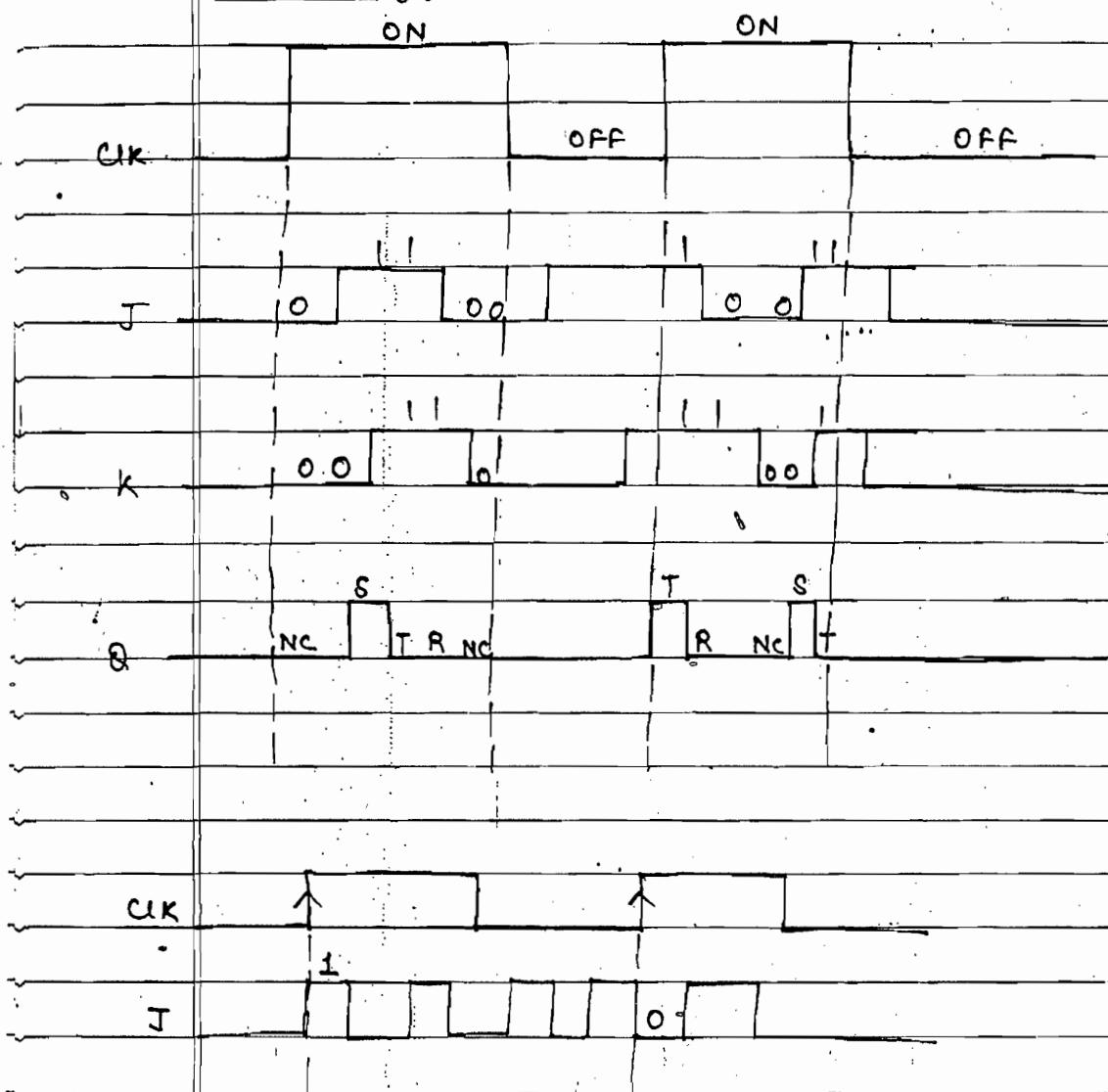


(b)

+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	\bar{A}	I	D	\bar{E}
	A	I	I	D

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

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

classmate
Date _____
Page _____

53

Ans: 1	\bar{A}	1	0	D	\bar{E}_3	E_2
	A	4	1	5	1	7

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

\bar{A}	1	0	0	3	0	2
A	4	1	5	1	7	0

\bar{A}	X	1	0	3	0	2
A	4	X	5	X	7	0

\bar{A}	X	0	0	1	3	0
A	4	X	5	X	7	0

\bar{A}	X	0	0	1	3	0
A	4	X	5	X	7	0

\bar{A}	X	0	0	1	3	0
A	4	X	5	X	7	1

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

Ans: 2	\bar{A}	E	1	F_3	1	0
	A	4	D	5	7	F

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

→ redundant

\bar{A}	1	X	D_3	X	0	1	0	3	1	0
A	4	0	5	7	0	6	A	4	0	5

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

$\Rightarrow F\bar{A}C$

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

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

→ redundant

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

Sheet Problems

S	R	NC
---	---	----

Ques: 1 S

Active-Low SR latch

0	1	0	0
0	0	X	

CK :

S :

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

R

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

Q

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

Ques: 2

S

ACTIVE-H

S	R	Q
0	0	NC
0	1	0
1	0	1
1	1	X

R

Q

Ques: 3

S

R

Q

Ques: 4

S

O O

W A C A S A S A

O O

EN

R

Q

I

Ques: 5

EIK

↓ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓

S

O O

R

O O

Q

I

Ques: 54

EN

W A C A S A S A

R

O O O O O

Q

EIK is active

as it is set as

Ques: 55

Clk: ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓ ↑ ↓

S

| 0 10 | | 0 10 |

R

0 0 | | 10 0 0 0 0

+ve trigger $\rightarrow Q_a$ -ve trigger $\leftarrow Q_b$

Ques: 56

Clk

+ve trigger

Q

10 0 10 | | 10 .. 10 10 10 10

57

Clk

D

Q

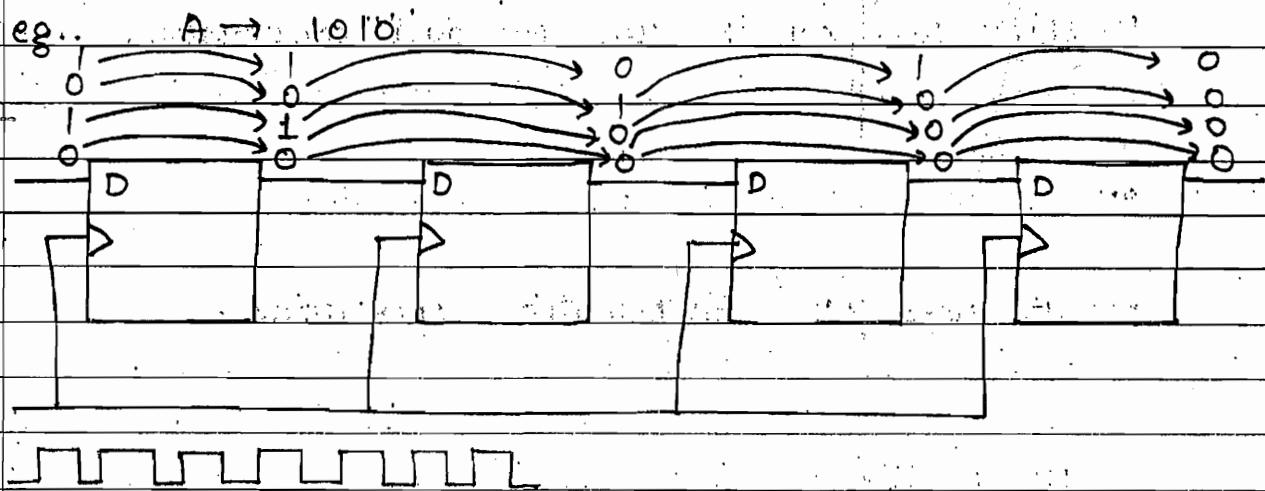
BINARY SHIFT REGISTER :- When F.F. are connected in cascaded manner, binary shift registers. These are 4 - types.

A). SISO \Rightarrow n and $(n-1)$

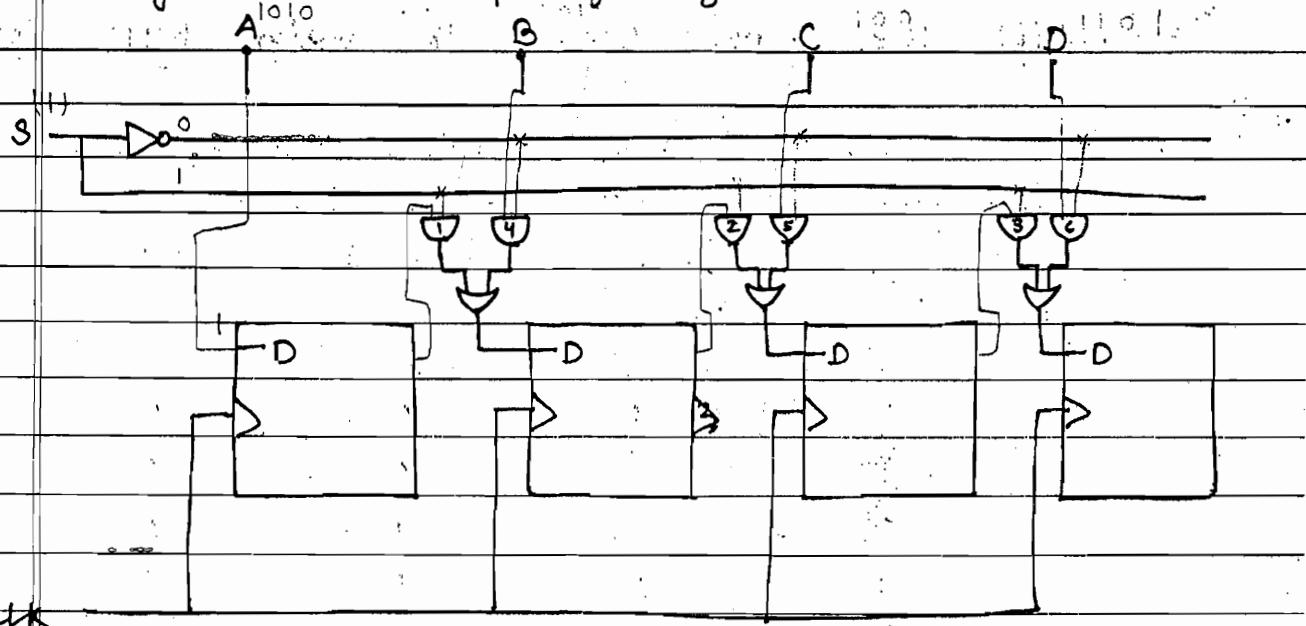
B). SIPO \Rightarrow n and o

C). PISO \Rightarrow l and $(n-1)$.

D). PIPO \Rightarrow l and o



Ques: Design 'PISO' binary shift register :



S=0

S=1

Parallel io \Rightarrow
4, 5, 6 v

1, 2, 3 v (operable)

* Application of Binary - shift - Register :

1. o- Temporary data storage

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

N : no. of f.f.

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

T_c : clock period

o- Shift Reg. can be used for data conversion. [SISO \rightarrow ^{comm} sys.]

eg.. SIPO converts serial form of data into parallel form.

o- PISO \rightarrow converts Parallel \rightarrow Series.

o- ^{gt} uses for Arithmetic operations.

eg. Left shift Reg. is multiplication by 2 network

Right- shift Reg. is division by 2 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

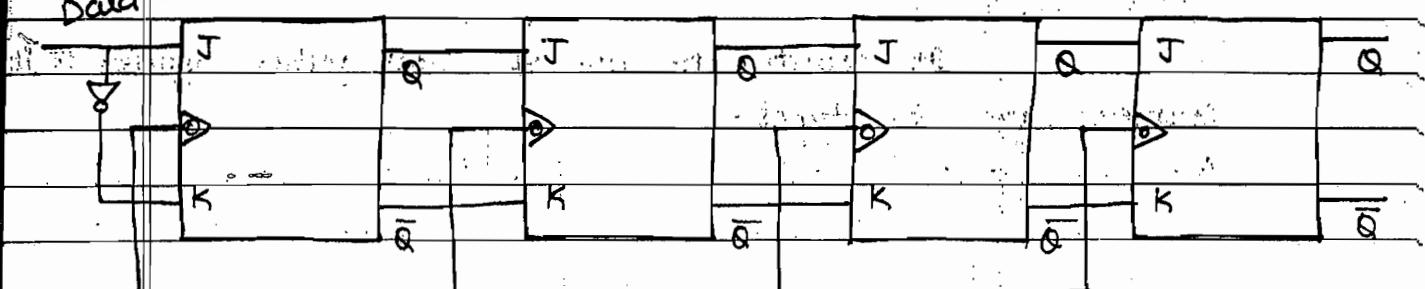
* Ques:

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

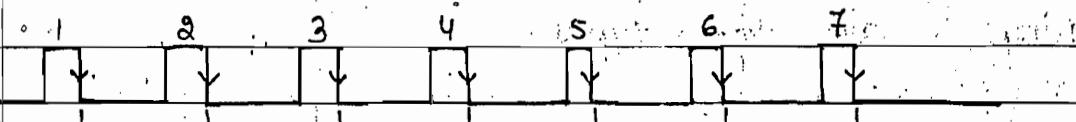
J K . F.F. with -'ve edge Trigg. and draw the
wave form diagrams for all F.F. op's for
given data 9/0.

CLK	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 9/10



CLK



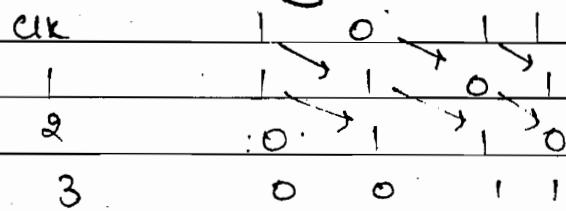
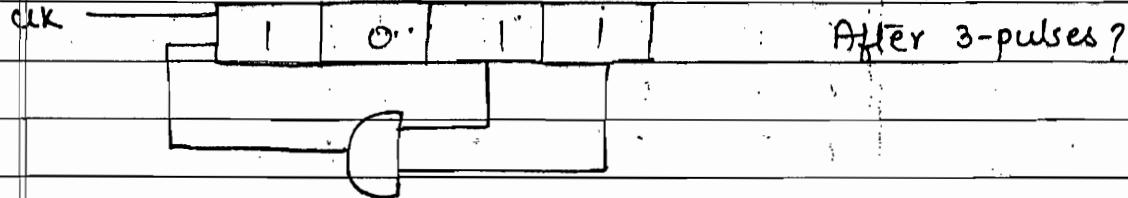
data 9/10

D/I/P

Q_AQ_BQ_CQ_D

Bi-directional

Ques: Design a 4-bit Binary shifter Register.



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)

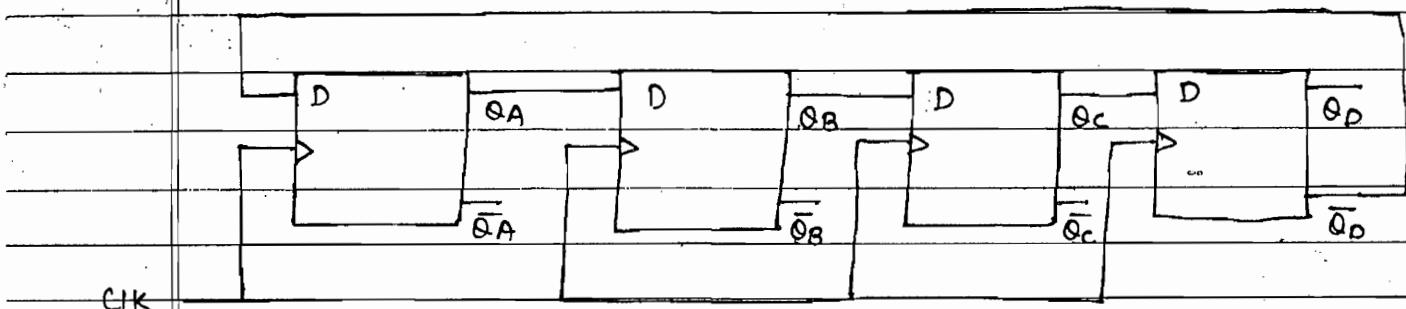
⇒ Ring counters are special category of synchronous C.

Ring C. are two - types -

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

feedback is given from complementary O/P 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

1 2 3 4 5 6 7 8

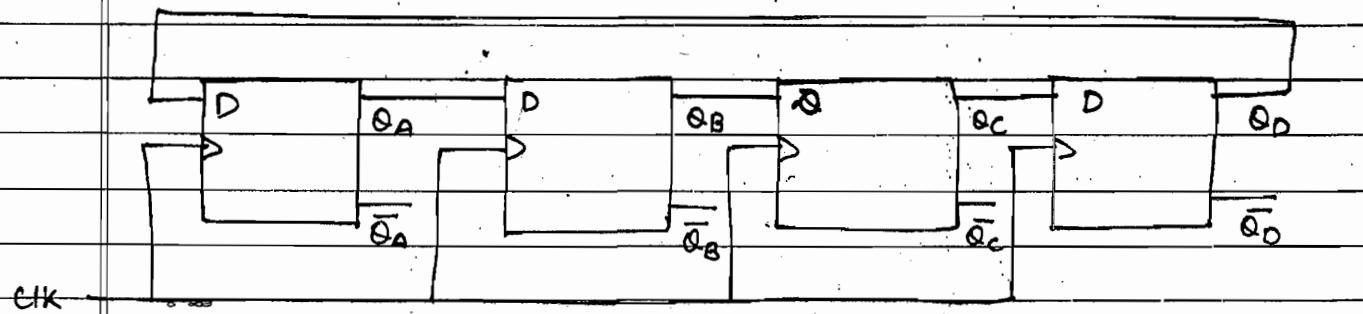
CK	↑	x	↑	↑	↑	↑	↑
----	---	---	---	---	---	---	---

it.

 Q_A Q_B Q_C Q_D

$n \rightarrow 2n$ → Possible states
 → CK pulses

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



CK	Q_A	Q_B	Q_C	Q_D
----	-------	-------	-------	-------

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

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

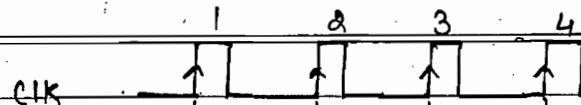
3	0	0	1	0
---	---	---	---	---

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

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

} mod 4

x

 Q_A Q_B Q_C Q_D

$n \rightarrow (n)$ → Possible - states.
 → CLK pulses.

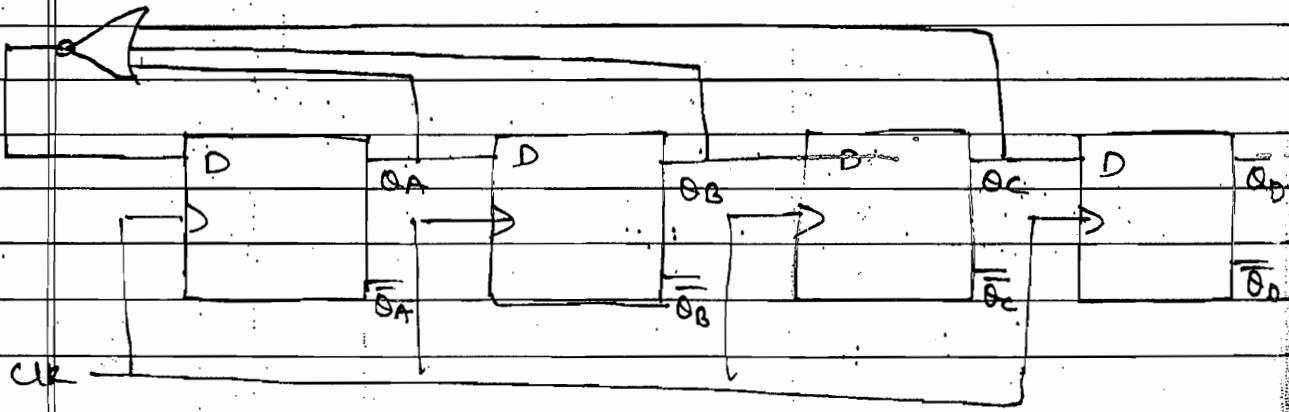
⇒ State identification (Decoding) is easy in RING-C.

* Unused - states :

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

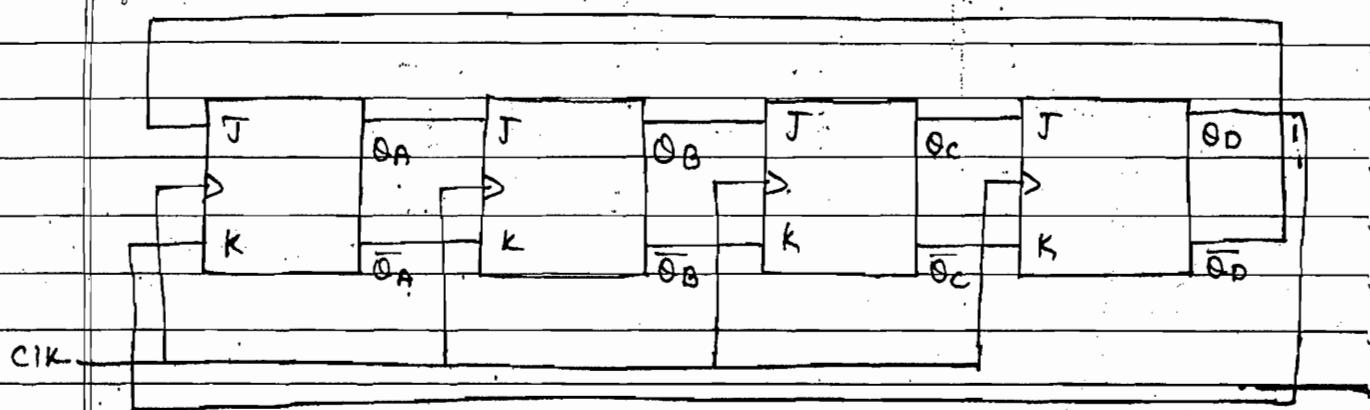
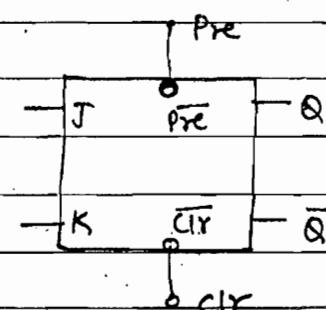
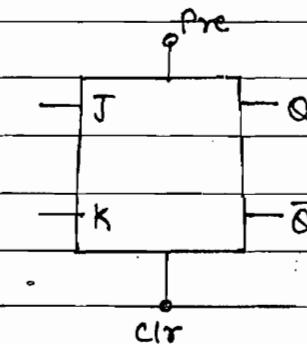
$$\text{RING C} \Rightarrow 2^n - 0^n$$

* SELF- STARTED RING - C. ⇒



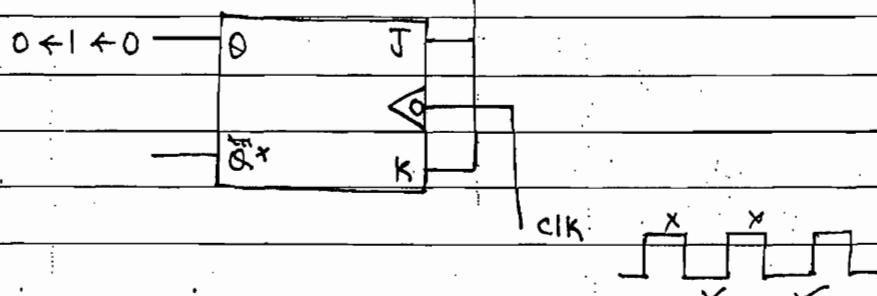
Johnson's RING C. \Rightarrow

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

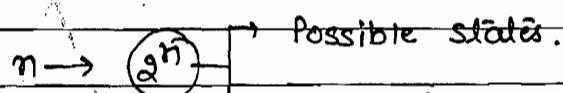
ASYNCHRONOUS INPUTS

Pre	Clr	Q^+
0	0	ff
0	1	0
1	0	1
1	1	X

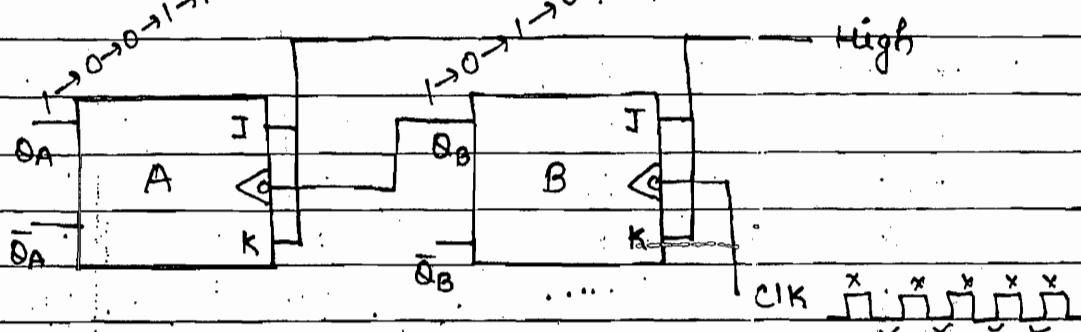
Pre	$\bar{C}lr$	Q^+
1	1	ff
1	0	0
0	1	1
0	0	X Don't Care Unspecified open

-'ve trigger:

Asyn. C → F.F. are not triggered simultaneously.



CLK pulses.



CLK	Q _A	Q _B
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0

mod '4'

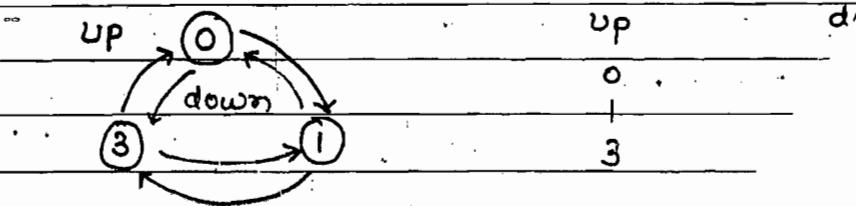
CLK ↓ ↓ ↓ ↓

Q_B ↑ ↑ ↑ ↑

Q_A 0 0 1 1

Date: _____

Synchronous up-down count - design :-



$y=0$	Present State		Next-State					
up	Q_1	Q_0	Q_1^+	Q_0^+	J_0	K_0	J_1	K_1
	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}, \bar{Q}_0	\bar{Q}, Q_0	Q, Q_0	Q, \bar{Q}_0
0	0	0	x	\bar{y}	1	x	x_3
0	1	1	x	y	1	x_5	x_7
1	0	x	1				
1	1	x	0				

$J_0 = 1$

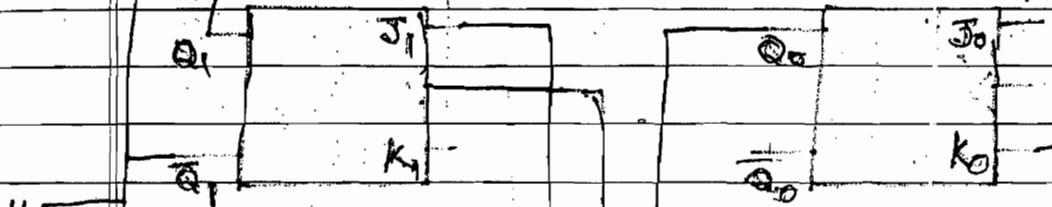
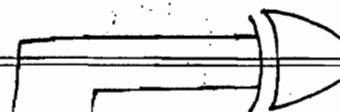
$y=1$	Present-state		Next-state					
down	Q_1	Q_0	Q_1^+	Q_0^+	J_0	K_0	J_1	K_1
	1	1	0	1	x	0	x	1
	0	1	0	0	x	-1	0	x
	0	0	1	1	1	x	1	x

Q	Q^+	J	K	\bar{Q}, \bar{Q}_0	\bar{Q}, Q_0	Q, Q_0	Q, \bar{Q}_0
0	0	0	x	\bar{y}	x	0	x
0	1	1	x	y	(x)	1	0
1	0	x	1				
1	1	x	0				

$y=1$

$K_0 = \bar{y}Q_1 + y\bar{Q}_0$

$y \oplus Q_1$



y



CLK

$$J_1 = y \oplus Q_0$$

$$K_1 = 1$$

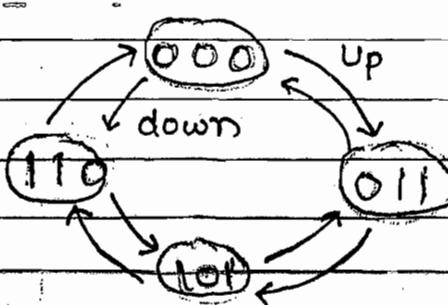
Ques: wanted Sequence By using T-flip-flop

0

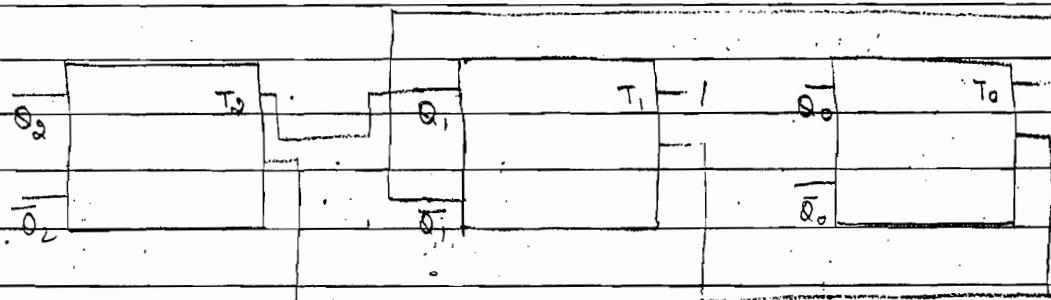
3

5

6



Present-State			Next-State			T ₀	T ₁	T ₂
Q ₂	Q ₁	Q ₀	Q ₂ ⁺	Q ₁ ⁺	Q ₀ ⁺			
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
			\bar{Q}_2, Q_0			\bar{Q}_1, Q_0		
$T_0 = \bar{Q}_1$			$T_0 = \bar{Q}_1, \bar{Y}$			\bar{Q}_1, Q_0		
$T_1 = 1$			Y			\bar{Y}		
$T_2 = Q_1$			$T_1 = 1$			\bar{Q}_1, \bar{Q}_0		
			\bar{Q}_1, \bar{Q}_0			\bar{Q}_1, \bar{Q}_0		
			$T_2 = Q_1, Y$			\bar{Y}		
			X			Y		
			Q			X		
			\bar{Q}			\bar{X}		



Q2

Procedure: corrected

Self-corrected counter: 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.

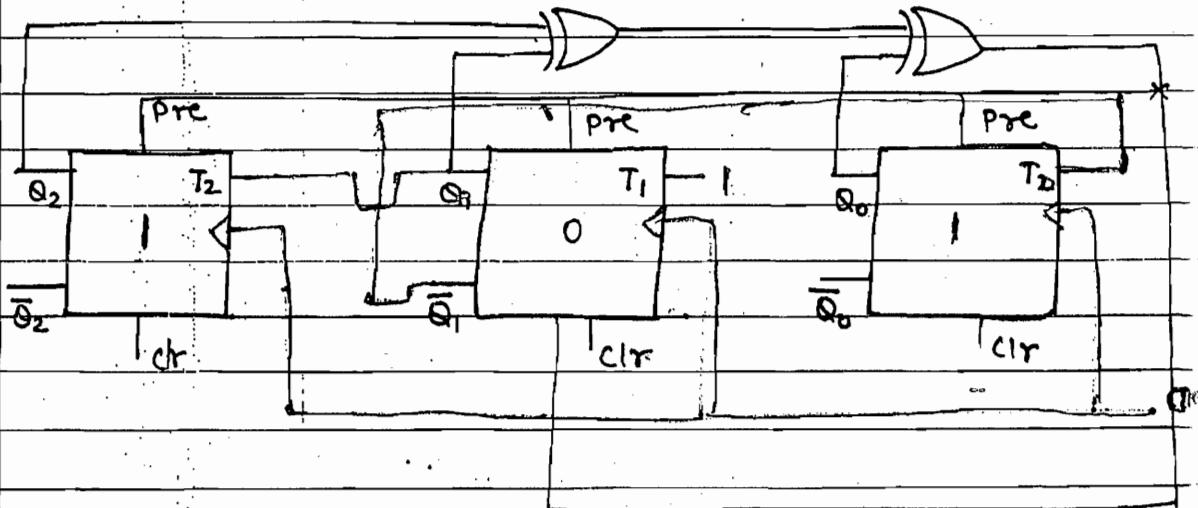
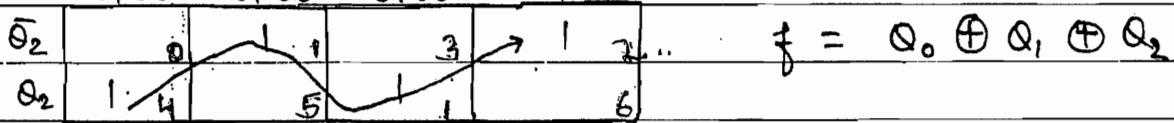
method to avoid lock-out condition: Get a circuit expression by unwanted state with the help of k-map

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

CLK	Q_1	T_2	T_1	\bar{Q}_1	T_0	Q_2	Q_3	Q_1	Q_0	Q_0	$Q_0 \rightarrow 4$
	0	1	1	1	1	1	1	0	1	1	$1 \rightarrow 7$
	1	1	0	0	0	0	0	0	1	1	$1 \rightarrow 1$
	0	1	1	1	0	0	0	1	0	0	$0 \rightarrow 2$
	1	1	1	0	0	1	0	0	0	0	$0 \rightarrow 4$

→ It is not self-connected.

$$\bar{Q}_2, \bar{Q}_0 \quad \bar{Q}_1, \bar{Q}_0 \quad Q_1, Q_0 \quad Q_1, \bar{Q}_0$$



* 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.

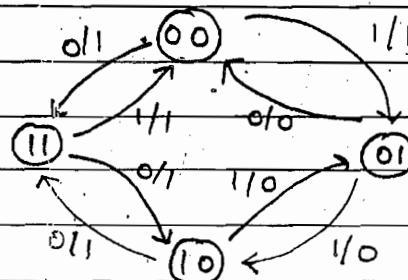
Step : 3 identify the equivalent states.

equivalent states : Two states are said to be equi. 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.

\Rightarrow



x/y

$x \Rightarrow I/P$

$y \Rightarrow O/P$.

Present state

Next state

Output

$x=0$ $x=1$

$x=0$ $x=1$

0 0

1 1 0 1

1 1

0 1

0 0 1 0

0 0

1 0

1 1 0 1

1 0

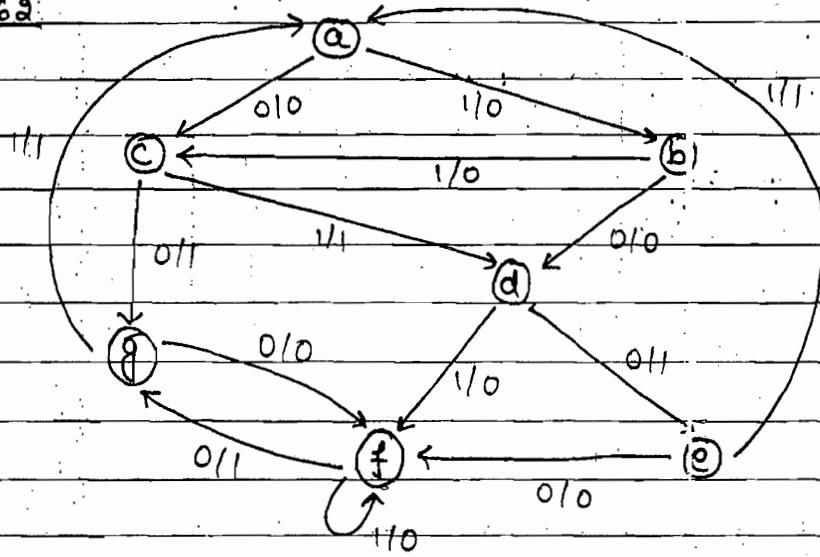
1 1

1 0 0 0

1 1

P.N. 69

Ques: 4

Present
state

Next-state

Output

	$x=0$	$x=1$	$x=0$	$x=1$
a	c	b	0	0
b	d	c	0	0
c	g e	g d	1	1
d	e	f d	1	0
$d=f$	f d	<u>a</u>	0	1
$e=g$	f	g e	1	0
$\dots \rightarrow g$	f	<u>a</u>	0	1

 \Rightarrow 5-states

Ques: 33

 $x=0$ $x=1$ $x=0$ $x=1$

a	c
b	d
c	<u>g</u>
d	e
e	<u>f</u>
f	<u>g</u>
g	<u>f</u>

State - deduction methods are 2-types :

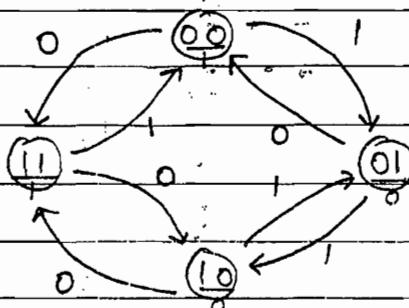
1. Meley method:

2. Moore method.

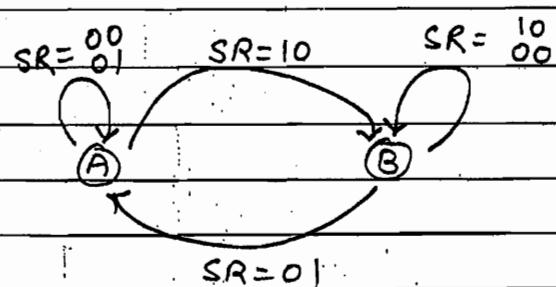
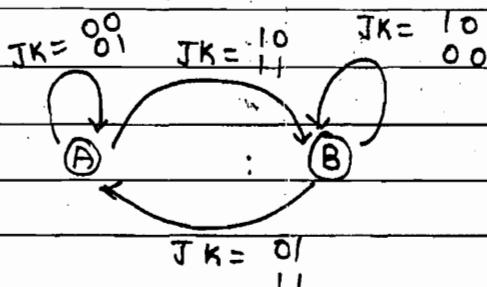
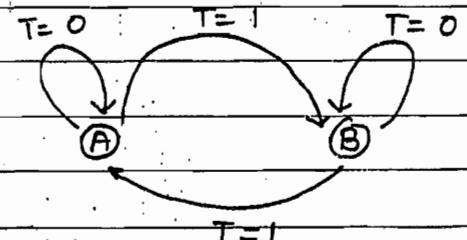
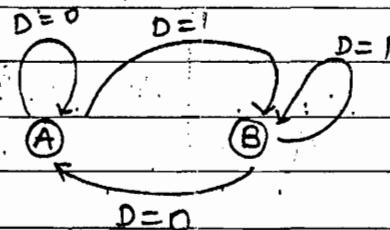
1. Meley - method : In this method present O/P depends on present S/I/P and present - state.

Note: above discussed examples are Meley method only.

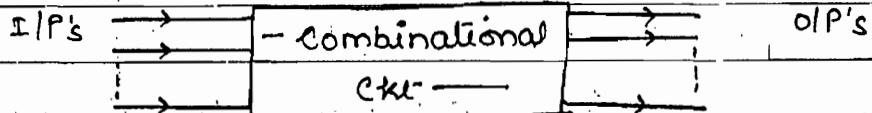
2. Moore - method : In this method the present O/P depends only on present state.



state diagram for f.f. \Rightarrow

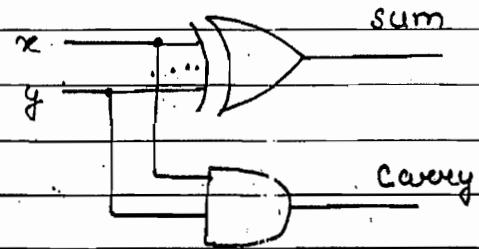


* Combinational Circuit :-



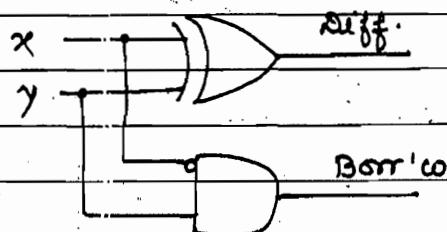
x	y	c	sum	sum = $\bar{x}y + x\bar{y}$
0	0	0	0	= $x \oplus y$
0	1	0	1	
1	0	0	1	$c = xy$
1	1	1	0	

H.A.



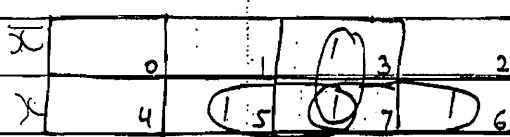
H.S.	x	y	B.	Diff.	Diff. = $\bar{x}y + x\bar{y}$
	0	0	0	0	= $x \oplus y$
	0	1	1	1	
	1	0	0	1	$B = \bar{x}y$
	1	1	0	0	

F.A.

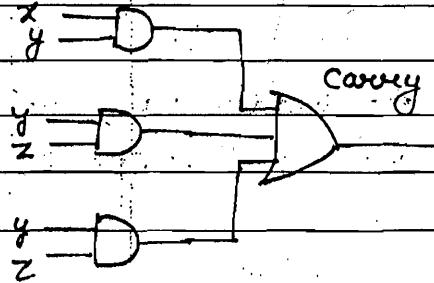


x	y	z	C	sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

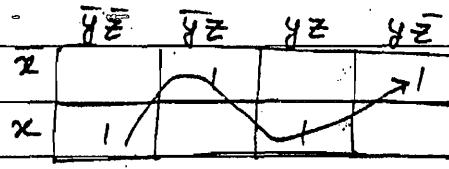
$$\text{carry} = \bar{x}yz + x\bar{y}z + xy\bar{z} + xyz.$$



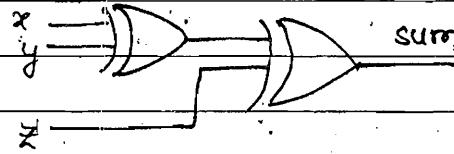
$$\text{carry} = xy + yz + yz$$



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



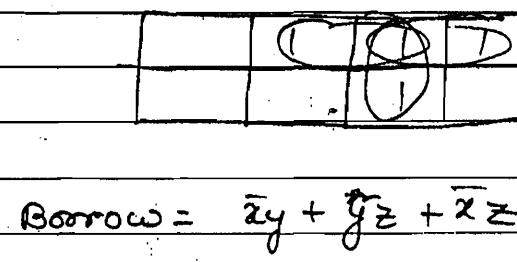
$$\text{sum} = x \oplus y \oplus z$$



x	y	z	B	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

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

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



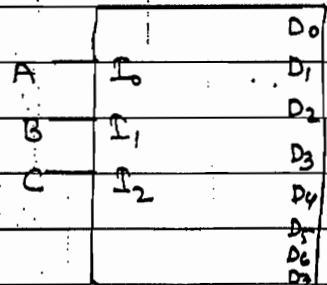
$$\text{Borrow} = \bar{x}y + \bar{y}z + \bar{x}z$$

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

Decoders :

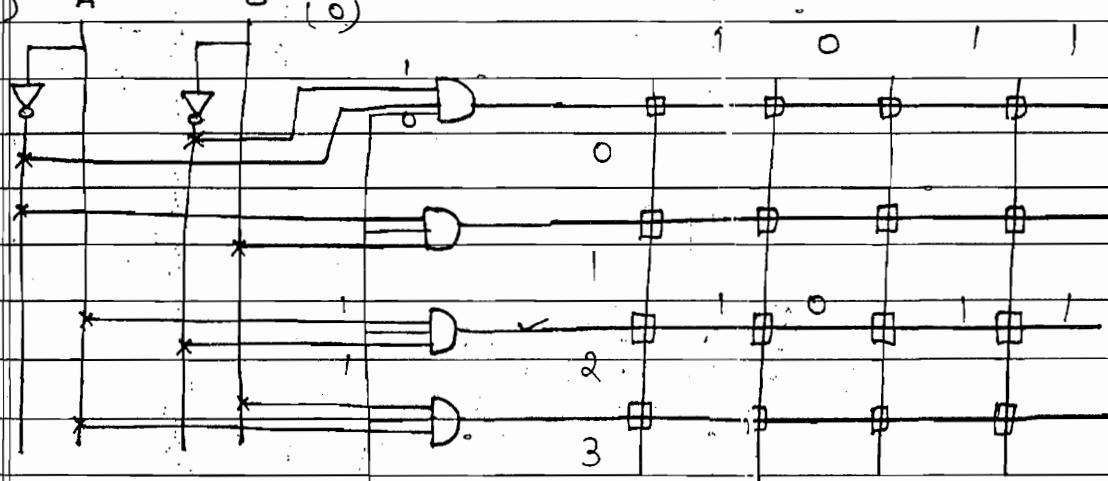
P.N. 59

Ques: 32



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

(1) A B (0)



Enable '1'
[0@ mux]
1 → many

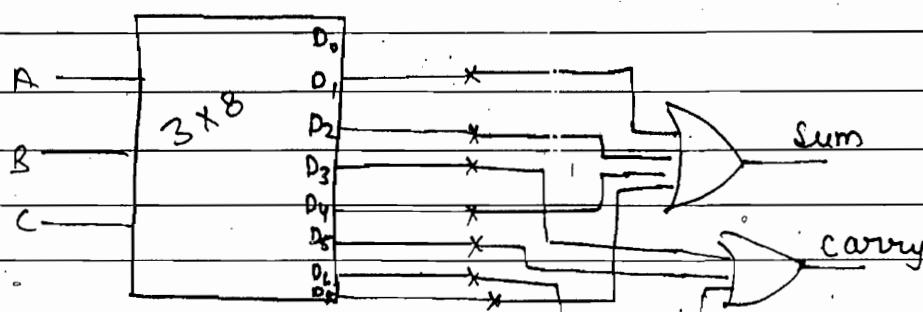
$$n \rightarrow 2^n$$

Do	$\bar{A}\bar{B}$
D ₁	$\bar{A}B$
Deco. D ₂	$A\bar{B}$
D ₃	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

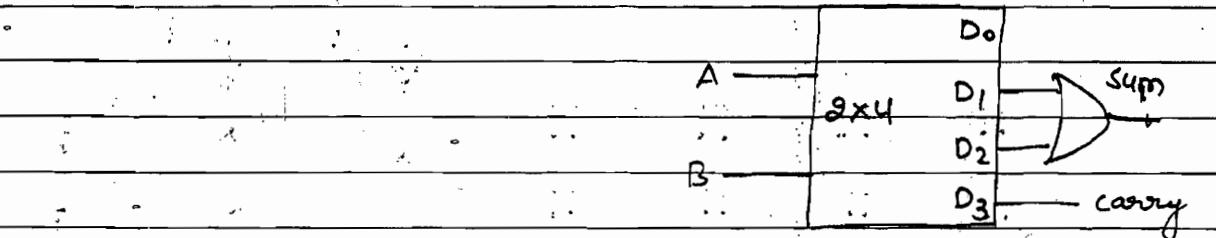
Sum = $\Sigma m(1,2)$

1 0 1 0 1

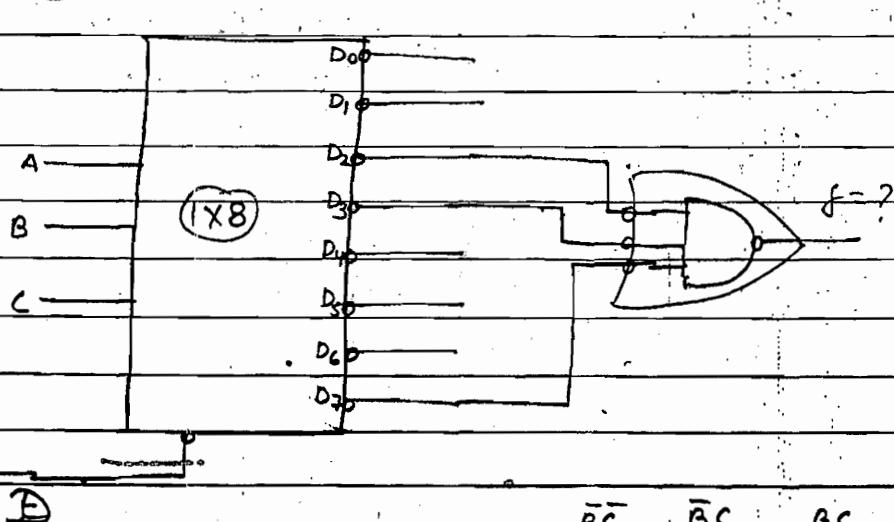
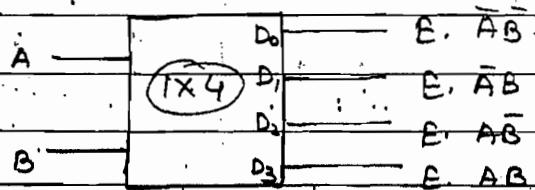
C = $\Sigma m(3)$

2 1 0 0 1

3 1 1 1 0



Demux :-



	$\bar{B}\bar{C}$	$\bar{B}C$	$B\bar{C}$	BC
\bar{A}	0	1	1	0
A	4	5	1	6
x = $\Sigma m(2,3,7)$				

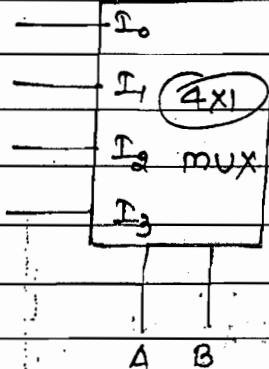
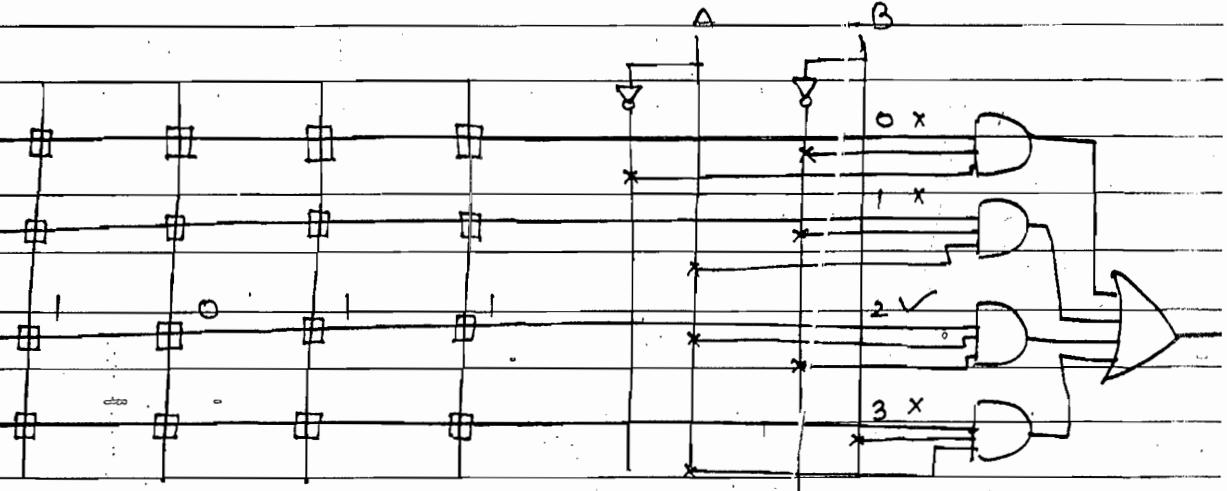
$$x = \bar{A}B + BC$$

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

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

Multiplexer :: Many to one circuit.

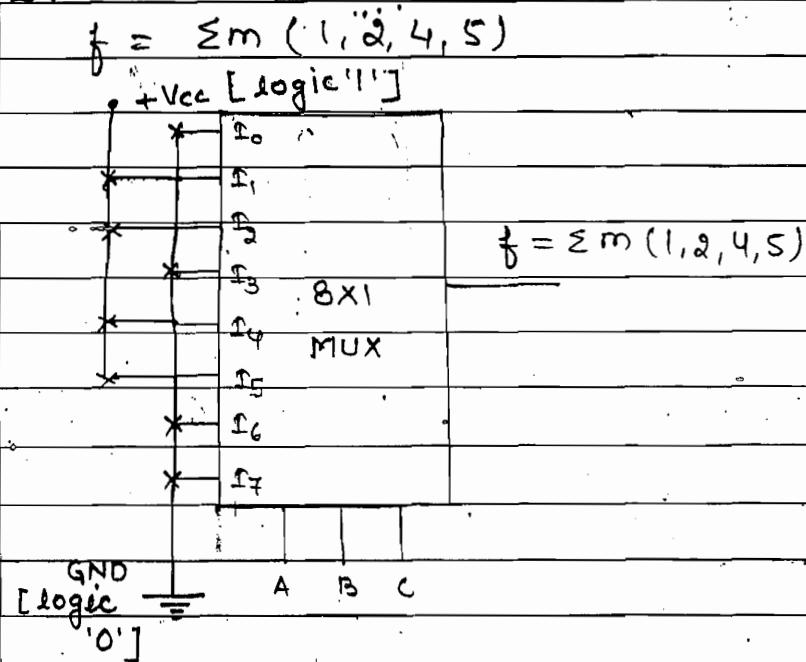
- Data selector.
- Parallel to serial converter.
- waveform generator.



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

Designing the combinational ckts by MUXs :-)

I:



II MSB:

	f (A B C)			A	f0	
				1	f1	$f = \sum m(1, 2, 4, 5)$
				\bar{A}	f2	4x1 MUX
0	0	0	0	0	f3	
1	0	0	1		B	C
2	0	1	0		f0 f1 f2 f3	
3	0	1	1	\bar{A}	0 1 2 3	
4	1	0	0	A	4 5 6 7	
5	1	0	1		A 1 \bar{A} 0	
6	1	1	0			
7	1	1	1			

 3. LSB: $f (ABC)$

		f0	f1	f2	f3
	\bar{C}	0	2	4	6
	C	1	3	5	7
	\bar{f}				
	0	2	4	6	

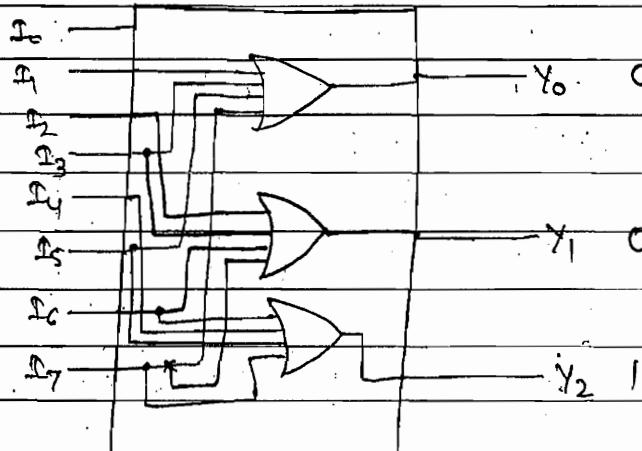
Encoder:

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	0	1
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

$$y_0 = I_1 + I_3 + I_5 + I_7$$

$$y_1 = I_2 + I_3 + I_6 + I_7$$

$$y_2 = I_4 + I_5 + I_6 + I_7$$



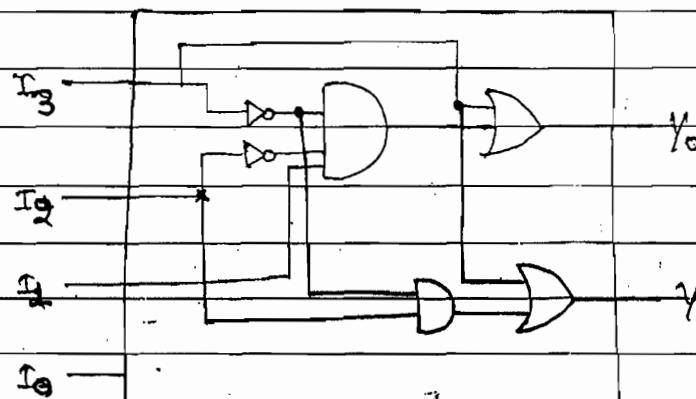
8x3 (Dctal to Binary Encoder).

Priority Encoder:

f_3	f_2	f_1	f_0	y_1	y_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$y_0 = f_1 \cdot f_2' f_3' + f_3.$$

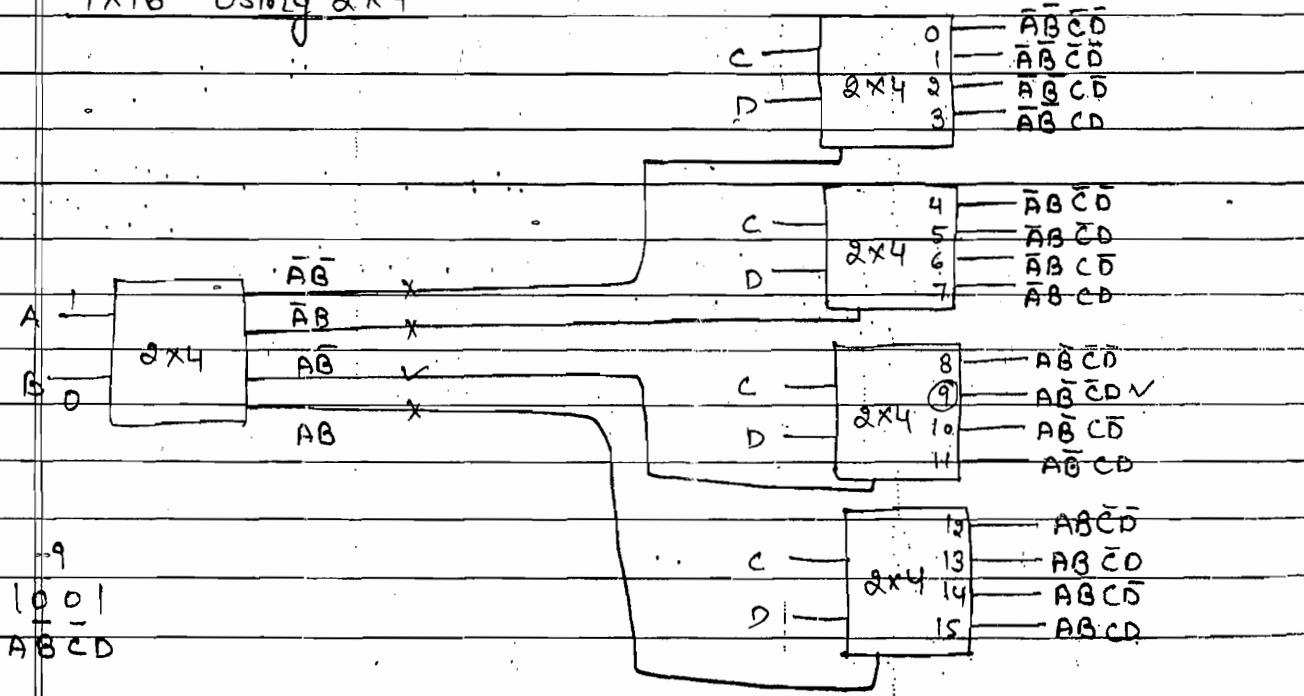
$$y_1 = f_2 \cdot f_3' + f_3.$$



4x2 MUX

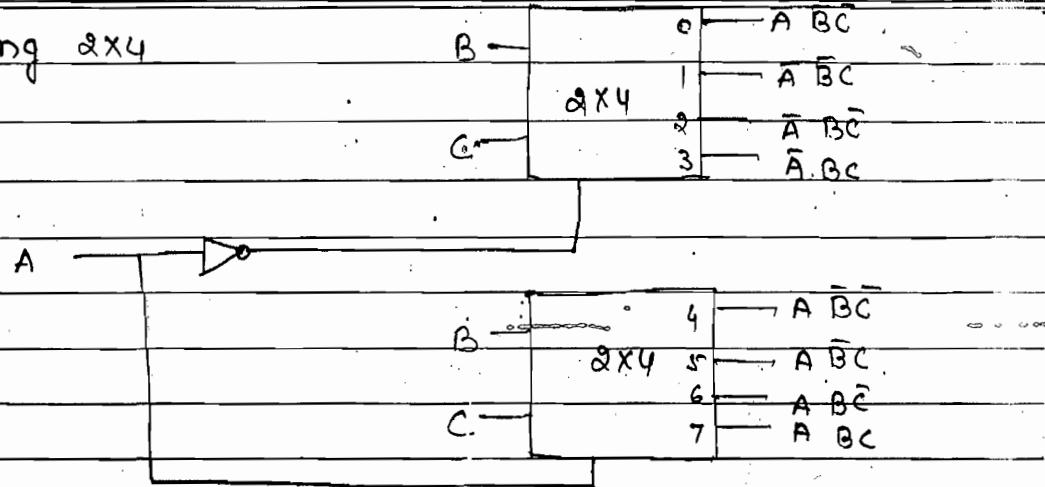
* Higher ordered circuits by using lower ordered circuits

4x16 using 2x4

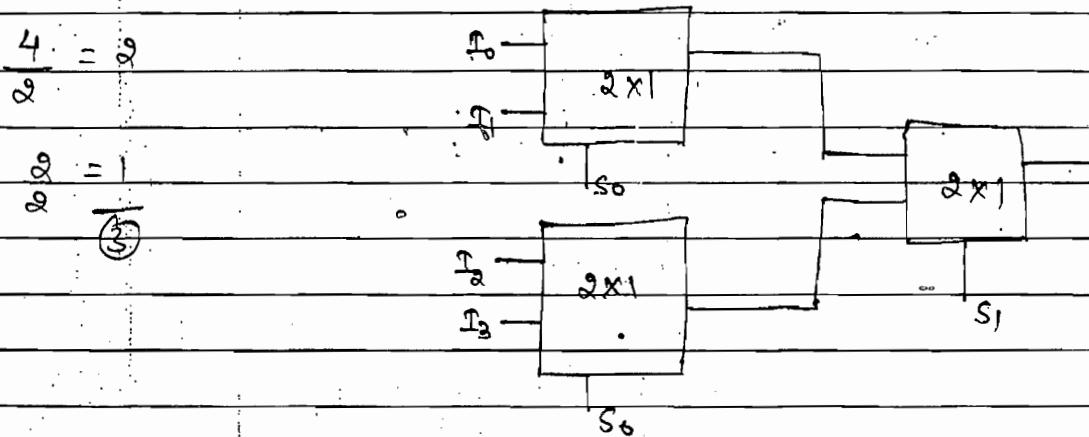


② 3×8 using 2×4

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



③ # 4×1 MUX By using 2×1 ...



short-cut:

① 4×16 using 2×4

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

should fully divisible

$$\frac{4}{4} = 1$$

(5) → so we need (5), 2x4 Demux's

② 3×8 using 2×4 *to take an odd no. so we'll have AND NOT gate here and that will not complete divisi - bte 60*

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

↓ not divisible

points:

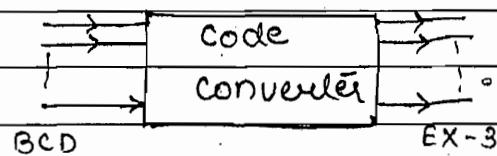
for eg: ① :

no. of decoders = ⑤

no. of enable decoders = ④

NUMBER - SYSTEM :

Codes and Codes Converter :



EX-3

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

0001 0000 ← BCD

→ 111

0001 0001 ← BCD

→ 114

0001 0100 ← BCD

2 4 2 1

0 0 0 0 0

2 4 2 1

1 0 0 0 1

5 → 0 1 0 1 X

2 0 0 1 0

5 → 1 0 1 1 V

3 0 0 1 1

4 0 1 0 0

5 1 0 0 1

6 1 0 0 0

7 1 0 1 0

8 1 1 1 0

9 1 1 1 1

⇒ 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

8 → 1 0 1 1 complement 0 1 0 0

9 → 1 1 0 0 → 0 0 1 1

BCD to Ex-3 conversion :-BCD
9/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	1	0	1	0	0	$\bar{A}B$	4	1	5	1	6	
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	X_{11}	X_{10}
4	0	1	0	0	0	1	1							
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

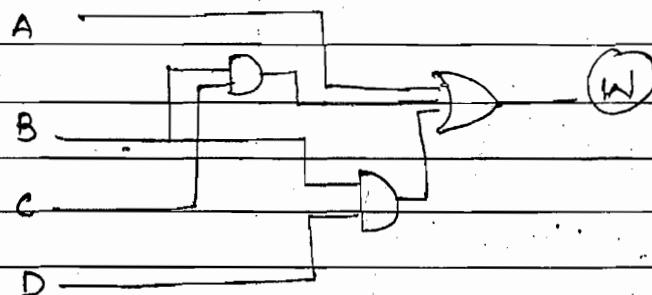
BCD

O/P's

EX-3

HW: make for

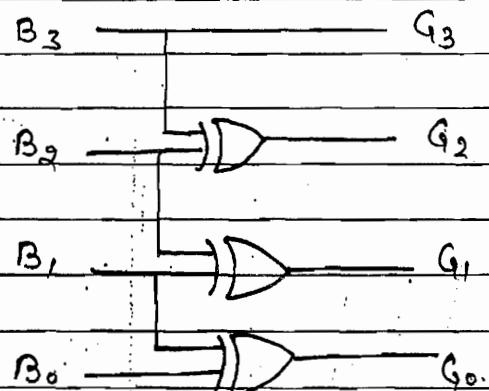
X, Y, Z as well.



Binary to Gray Conversion

MSB	B_3	B_2	B_1	B_0	...	
Binary		\oplus	\oplus	0	\oplus	
Gray	1	0	1	1		
	G_3	G_2	G_1	G_0		
MSB						

$G_3 = B_3$ $G_1 = B_2 \oplus B_1$
 $G_2 = B_3 \oplus B_2$ $G_0 = B_1 \oplus B_0$

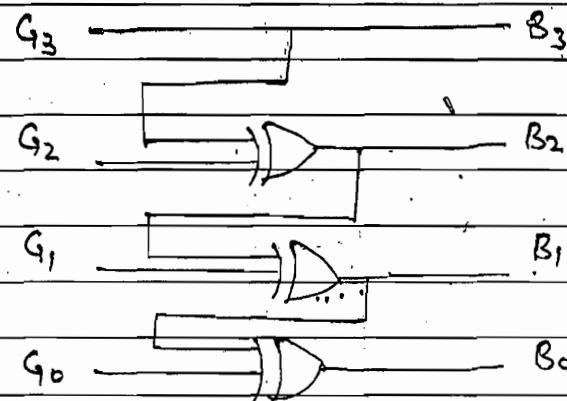


* Gray to Binary Convrs

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

$$B_3 = G_3 \quad B_2 = B_2 \oplus G_2 \quad B_1 = B_2 \oplus G_1$$

$$B_0 = B_1 \oplus G_0$$



SEVEN SEGMENT :-

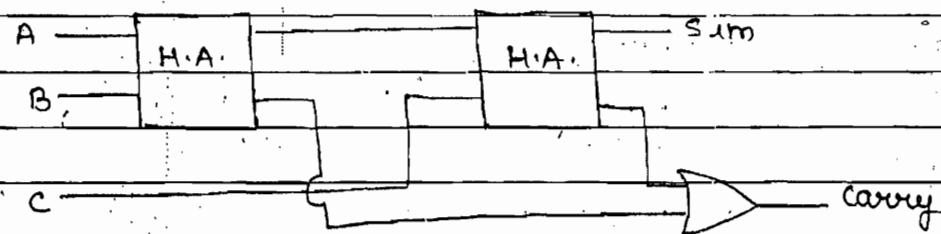
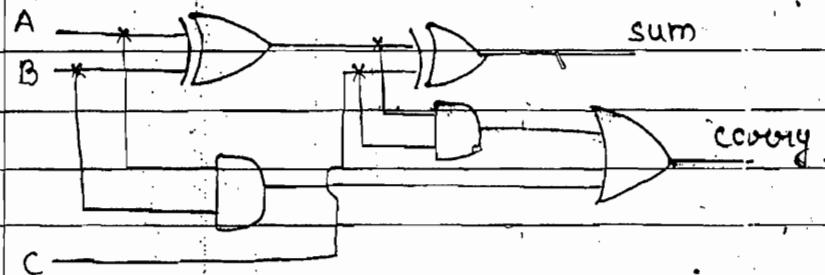
	I/P B.C.D.				O/P Seven segment						
a f b e g c d	A	B	C	D	a	b	c	d	e	f	g
1	0	0	0	0	1	1	1	1	1	1	0
2	0	0	1	0	1	1	0	1	1	1	0
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	0	1	0	0	1	1
5	0	1	0	1	1	0	1	0	1	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	0	0	1	1

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

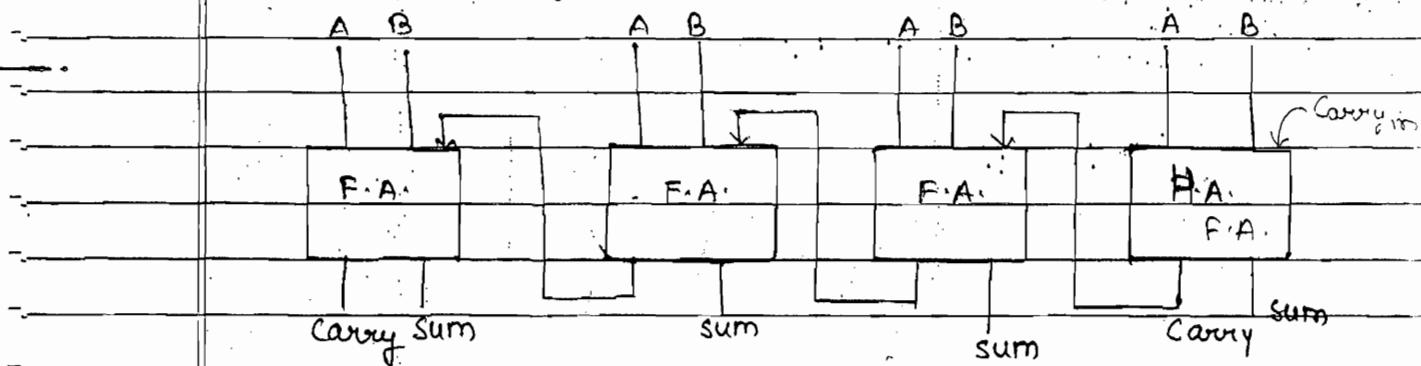
P ₂	P ₁	a f b e g c d	c	d	e	f	g
0	0	1	1	1	1	1	0
0	1	1	1	0	1	0	1
1	0	1	0	1	1	0	1
1	1	1	0	0	1	1	1

$$\begin{array}{c}
 \overline{P_1} \quad P_1 \\
 \overline{P_2} \quad \boxed{1 \quad 1} \\
 P \quad \boxed{1 \quad 0}
 \end{array}
 \quad b = \overline{P_2}$$

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 - $(N-1) F.A + 1 H.A.$

$$(N-1) [2H.A. + 1-OR gate] + 1H.A.$$

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

$$\therefore N=4$$

$$(N-1)$$

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

$$4-1$$

$$(8-1)$$

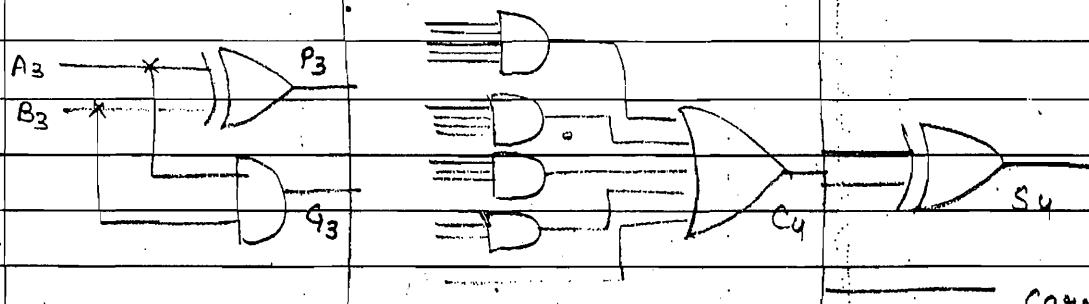
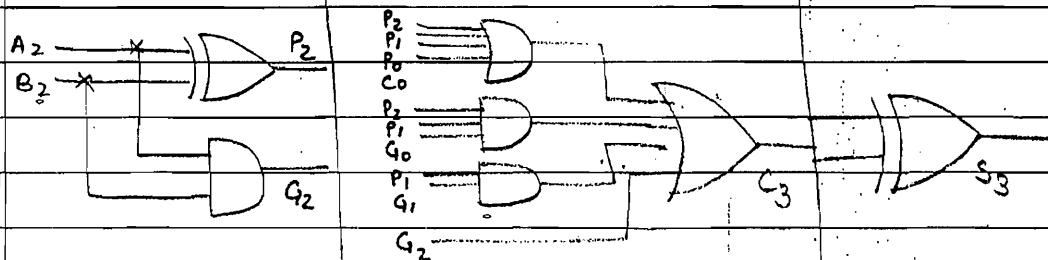
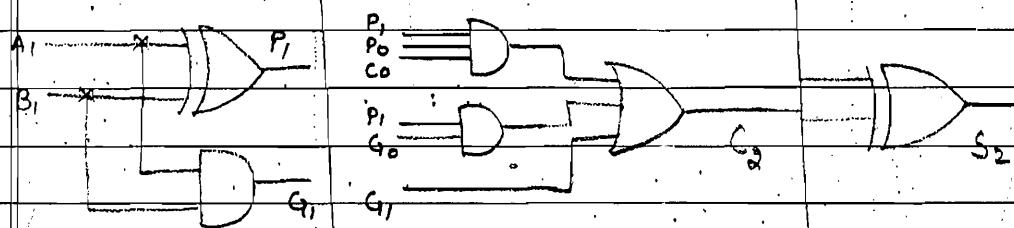
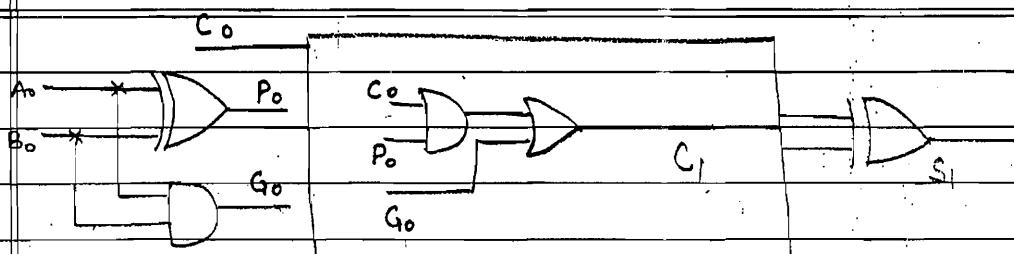
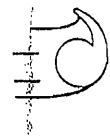
$$= 3 \text{ OR gates}$$

$$T = H.A.$$

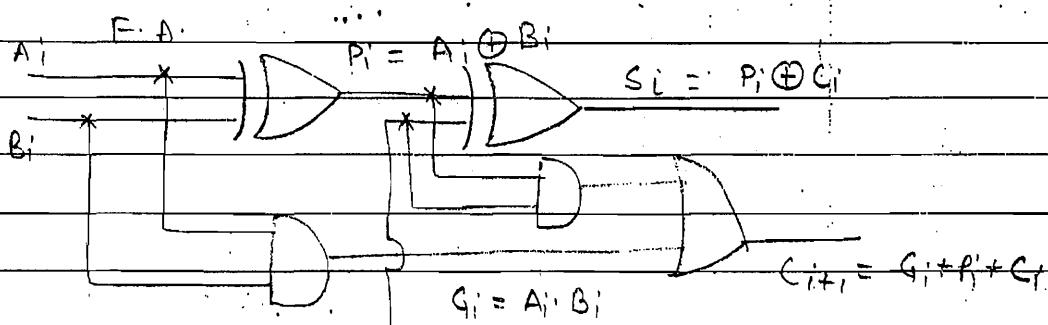
A'

AL
FC

de'smat™



Carry



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

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

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

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

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	10's	2's	8's	16's
(4-1)1's	9's	1's	7's	15's

$9 \rightarrow$ radix

(or)
base

9's Complement:

$$\frac{9}{7_{(10)}} \xleftarrow{\quad 2 \quad} \xrightarrow{\quad 2 \quad}$$

$$\begin{array}{r} 9 \quad 9 \quad 9 \\ 9'5 \leftarrow | \quad 7 \cdot 3 \\ \downarrow \quad \quad \quad \downarrow \\ 8 \quad 2 \cdot 6 \end{array}$$

$$\begin{array}{r} 1 \\ \text{1's complement} \Rightarrow 1's \leftarrow 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ \text{1's} \leftarrow 1 \\ \hline 0 \end{array}$$

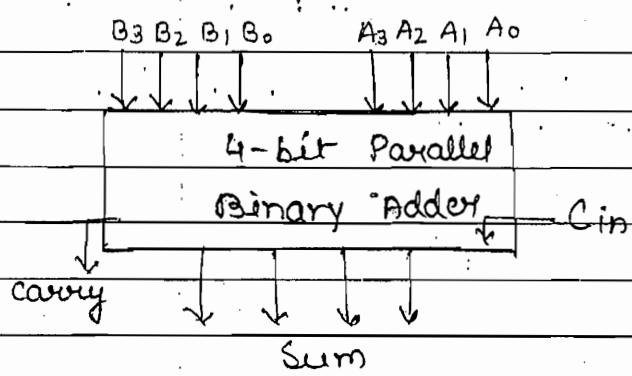
$$\begin{array}{r} 1 \\ \text{2's complement} \Rightarrow 2's \leftarrow 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ 2's \leftarrow 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 9 \\ \text{10's complement} \Rightarrow 10's \leftarrow (7)_{10} \\ \hline 2 \\ + 1 \\ \hline 3 \end{array} \quad \begin{array}{r} 9 \quad 9 \quad 9 \\ 10's \leftarrow 17 \cdot 3 \\ \hline 82 \cdot 6 \\ + 1 \\ \hline 82 \cdot 7 \end{array}$$

$$\begin{array}{r} 10100 \\ \uparrow \\ 2's \\ \hline 01100 \end{array} \quad \begin{array}{r} 1010 \\ \downarrow \\ 2's \\ \hline 01011 \end{array}$$

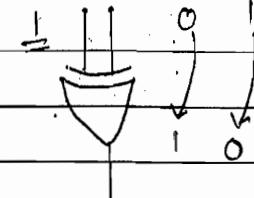
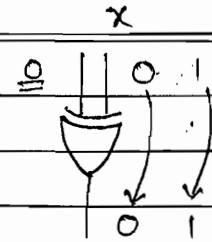
$$\begin{array}{r} 1000 \\ \downarrow \\ 2's \\ \hline 1000 \end{array}$$

2's complement Adder / Subtractor :



$$A \rightarrow A_3 A_2 A_1 A_0$$

$$B \rightarrow B_3 B_2 B_1 B_0$$



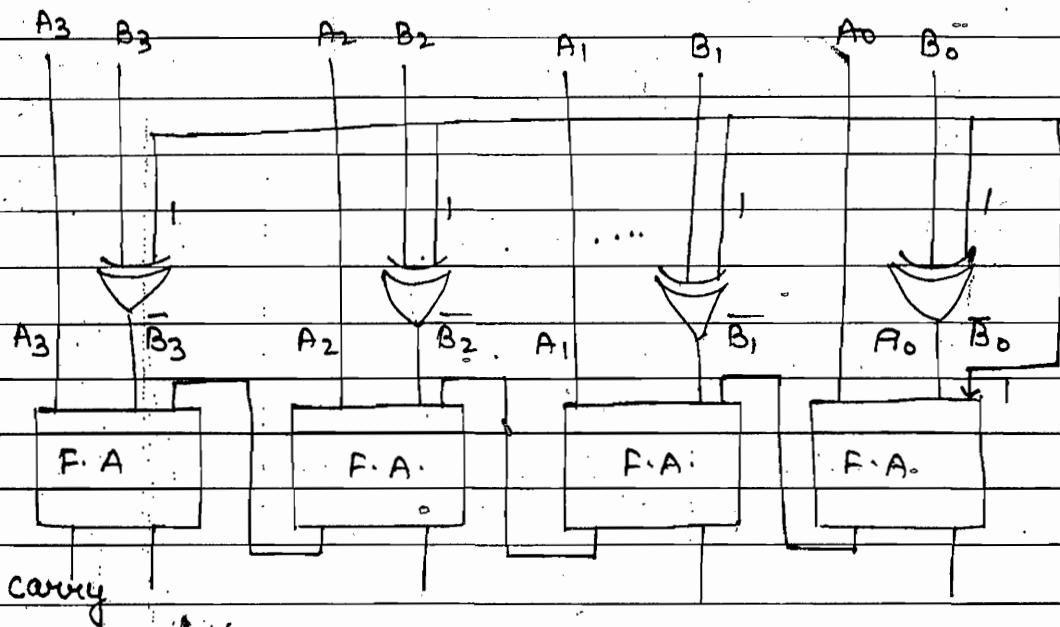
$$A_3 \quad B_3$$

$$A_2 \quad B_2$$

$$A_1 \quad B_1$$

$$A_0 \quad B_0$$

(control)



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

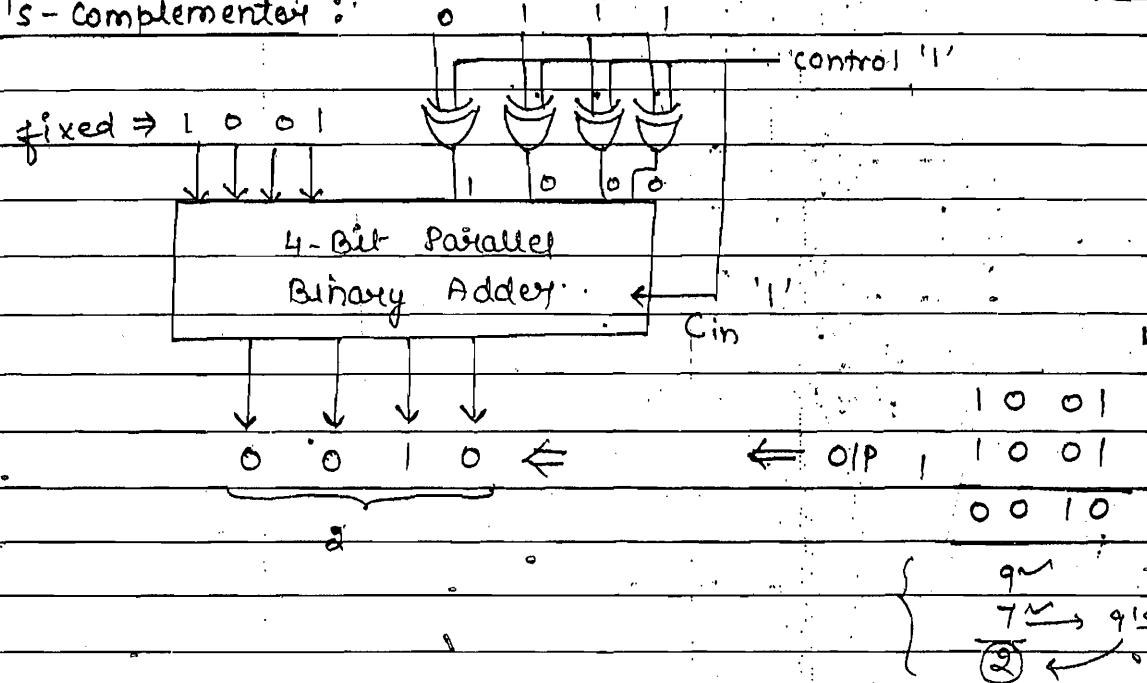
$$\Rightarrow 1 \Rightarrow A - B$$

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

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

$$\Rightarrow A_0 - B_0$$

* 9's-Complement:



* 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

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

12

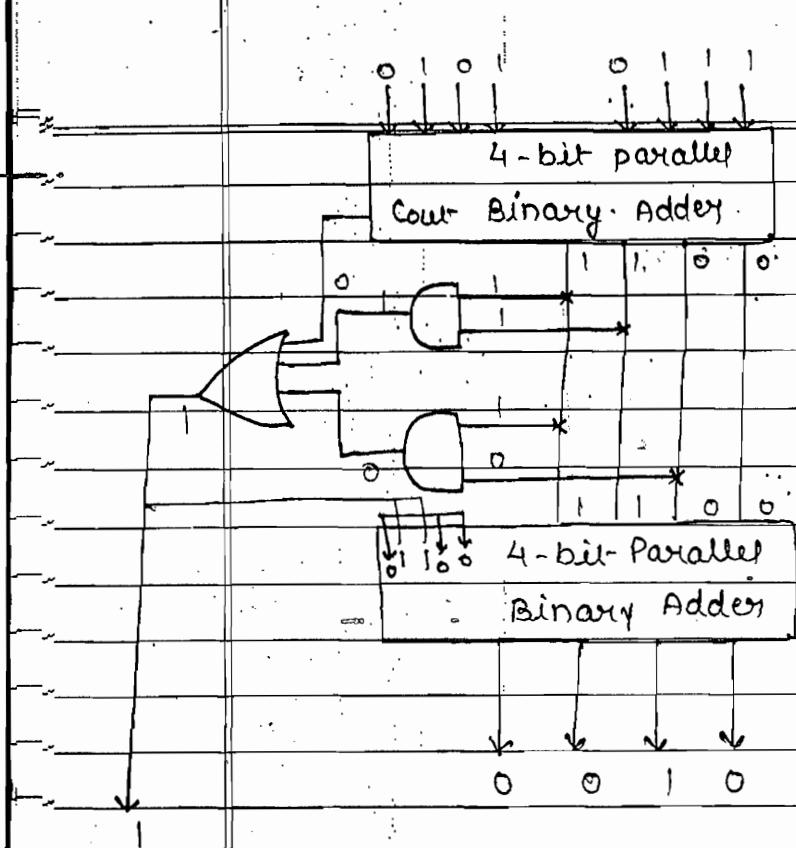
Case(iii) \Rightarrow

$$8 \rightarrow 1000$$

$$9 \rightarrow 1001$$

$\xrightarrow{\text{carry}} 10001 \leftarrow \text{invalid}$

$$\begin{array}{r} 0110 \\ \hline 0001/0111 \leftarrow +'7' \end{array}$$



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 end-around 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.

$$\text{case(i)} \quad 6 \rightarrow 0110$$

$$-4 \rightarrow 0101 \quad [9\text{'s comp. of } 4]$$

1011 invalid

$$0110 \rightarrow 6$$

$$\begin{array}{r} 000 \\ \text{end-around} \\ \text{carry} \end{array} \quad \begin{array}{r} +1 \\ \hline 0010 \end{array}$$

(+2)

$$\text{case(ii). } 4 \rightarrow 0100$$

$$-6 \rightarrow 0011 \quad [9\text{'s comp. } 6]$$

$\overline{0111}$ ← valid

↓
No-carry

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

for final answer Take 9's comp.

$$0111 \rightarrow 7 = \frac{9}{2}$$

g's comp.

0 1 0 1
4-bit parallel

Cout Binary Adder

1 0 1 1

0

1

0

1

0 1 0 1
4-Bit Parallel

Binary adder

0 0 1 0

0 0 1 0

0 0 0 0

0 0 1 0

4-bit

adder

Cin

Cout

0 0 1 0

+ 2

No. Representation :

Case(i) (+)ve number

+51

32	16	8	4	2
1	1	0	0	1

$\boxed{0} \ 1\ 1\ 0\ 0\ 1\ 1$
 ↓ sign bit ← magnitude

Case(ii) -ve number

-51

$\boxed{1} \ 1\ 1\ 0\ 0\ 1\ 1 \Rightarrow$ sign-magnitude
 ↓ sign bit ← magnitude

$\boxed{1} \ 0\ 0\ 1\ 1\ 0\ 0 \Rightarrow$ 1's comp.
 ↓ sign Bit ← 1's comp.

0 0 . 1 1 . 0 0

+ 1

$\boxed{1} \ 0\ 0 \ 1\ 1\ 0\ 1 \Rightarrow$ 2's complement
 ↓ sign Bit ← 2's comp

Given	sign	1's	2's
0 1 0 1	+5	+5	+5
1 1 0 1	-13	(-2)	(-3)

$\begin{array}{r} 1\ 1\ 1\ 0 \\ - 1\ 0 \\ \hline \end{array}$ sign.

$\begin{array}{r} 1\ 1\ 1\ 0 \\ \downarrow 0\ 0\ 1\ 0 \\ - 2 \rightarrow 1's \text{ comp.} \end{array}$

$\begin{array}{r} 1\ 1\ 1\ 0 \\ \downarrow 0\ 0\ 1\ 0 \\ + 1 \\ \hline 0\ 0\ 1\ 1 \end{array}$

(-3)

Range → sign

I's comp.

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

Ex:- $n = 3$ for $n = 3$, range $\Rightarrow -3 \text{ to } +3$

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

fig(i)

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

sign	2's comp. for $n = 3$
0 0 0 + 0	0 0 0 + 0 Range $\Rightarrow -4 \text{ to } +3$
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

$$[2^{(3-1)} - 1] \text{ to } +[2^{3-1} - 1]$$

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

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

.. if MSB 1

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

Overflow - Condition :



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

if ; $f = 0$, No-overflow.
 $f = 1$, Overflow.

eg...

$$\begin{array}{r} 5 \rightarrow x \ 0 \ 1 \ 0 \ 1 \\ 4 \rightarrow y \ 0 \ 1 \ 0 \ 0 \\ \hline z \ 1 \ 0 \ 0 \ 1 \end{array}$$

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

→ additional Bit to overcome the overflow.

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

Tx:

Even-parity

Rx:

↓ error

eg.. 0101

0111

x y z p

0 0 0 0

0 0 1 1

0 1 0 1

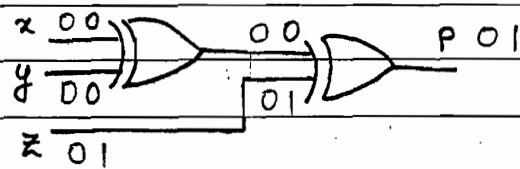
0 1 1 0

1 0 0 1

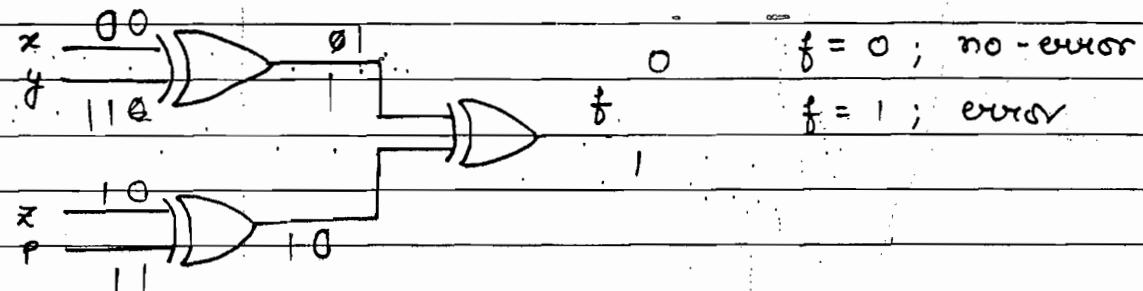
1 0 1 0

1 1 0 0

* Parity Generator



* Parity checker:



* Hamming Code : (7-bit)

given - data : 1101

Tx :	P ₁	P ₂	D ₃	P ₄	D ₅	D ₆	D ₇
	1	0	1	0	1	0	1

even-Parity

P ₄	P ₂	P ₁
2 ³	2 ¹	2 ⁰
x	y	z

7-bit

P ₁
P ₂
P ₄
P ₈

12-bit

P ₁
P ₂
P ₄
P ₈

15-bit

P ₁
P ₂
P ₄
P ₈

16-bit

P ₁
P ₂
P ₄
P ₈
P ₁₆

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

odd

P₁ (1, 3, 5, 7) \Rightarrow [P₁, 111] \Rightarrow P₁ = 1

even

P₂ (2, 3, 6, 7) \Rightarrow [P₂, 101] \Rightarrow P₂ = 0

P₄ (4, 5, 6, 7) \Rightarrow [P₄, 101] \Rightarrow P₄ = 0

Rx :

1	0	0	0	1	0	1
P_2	P_2	$\boxed{P_3}$	P_4	P_5	P_6	P_7

error

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

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

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

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

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

Code-Conversions \Rightarrow

① Dec. to Any (successive division) :

10^2	10^1	10^0	10^{-1}	10^{-2}
--------	--------	--------	-----------	-----------

↓
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	$\rightarrow 0$

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

Fractions : (Successive multiplication) :-

$$0.75_{(10)}$$

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

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 :-

$$428_{(10)}$$

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

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

(2) 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

11011.0101

011/011.010/100

$\Rightarrow 33.24_{(8)}$

e) Bin \rightarrow Hexa :-

11101.10

0001/1101.1000

$\Rightarrow 1D.8_{(16)}$

Any \rightarrow Bin :-

a) dec \rightarrow bin \Rightarrow

oct \rightarrow bin :

75.36₍₈₎

7/5.3/6₍₈₎

$\Rightarrow 111/101.011/110_{(8)}$

b) Hexa \rightarrow Bin \Rightarrow

A. B. C. . D₍₁₆₎

A/B/C/D

101010111100.1101₍₈₎

Oct \rightarrow Any \Rightarrow

oct \rightarrow bin

oct \rightarrow dec

a) Oct \rightarrow Hexa :

275.3₍₈₎

bin

000010111101.0110

$\Rightarrow BD.6_{(16)}$

Any \rightarrow Oct :-dec \rightarrow Oct ✓bin \rightarrow Oct ✓Hexa \rightarrow Oct

AB9, 2C(16)

101010111001.001011000

5271.130₍₈₎

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 bits '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:(ii)

$$+ 9 \rightarrow \begin{array}{r} 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{array}$$

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

$$\begin{array}{r} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{array}$$

↓
Sign-Bit

Sum = +13

Case: (ii) \Rightarrow $+9 \rightarrow 01001$
 $-4 \rightarrow 11100$

$$\begin{array}{r} 01001 \\ 11100 \\ \hline 100101 \end{array}$$

↓ ignored carry ↓ sign Bit

Sum = +5

case (iii) $-9 \rightarrow 10111$
 $+4 \rightarrow 00100$

$$\begin{array}{r} 10111 \\ 00100 \\ \hline 11011 \end{array}$$

↓ Sign-Bit Sum = -5

Case: (iv) $-9 \rightarrow 10111$
 $+9 \rightarrow 01001$

$$\begin{array}{r} 10111 \\ 01001 \\ \hline 100000 \end{array}$$

↓ ignored carry ↓ sign Bit Result = 0

Case (iv) \Rightarrow $-9 \rightarrow 10111$
 $-4 \rightarrow 11100$

$$\begin{array}{r} 10111 \\ 11100 \\ \hline 110011 \end{array}$$

↓ ignored carry ↓ sign Bit Sum = -13

* 1's complement Arithmetic :-

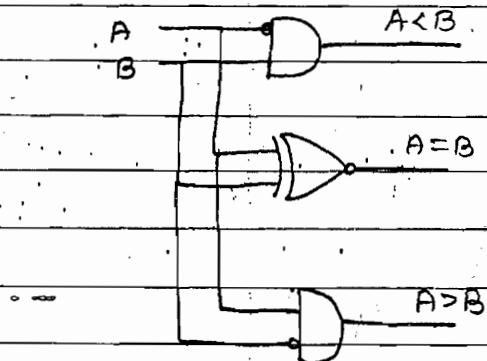
Minuend should be added to 1's complement of substrahend with a carry generated it should become end-around carry. then check the MSB :

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

Note \Rightarrow

Magnitude - Comparator \Rightarrow

A	B	$A < B$	$A = B$	$A > B$	$A \rightarrow A_3 A_2 A_1 A_0$	$B \rightarrow B_3 B_2 B_1 B_0$
0	0	0	1	0		
0	1	1	0	0		
1	0	0	0	1		
1	1	0	1	0		



1 x x x
0 x x x
A > B

0 x x x
1 x x x
A < B

$A_3 = B_3 \Rightarrow (A_2, B_2)$
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

 $A < B$

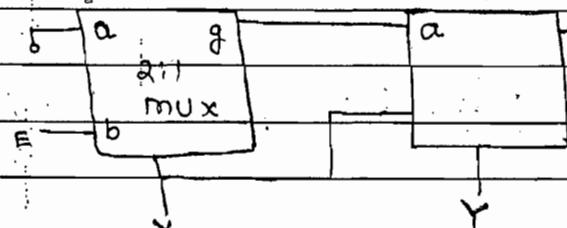
$$A_3' \cdot B_3 + (A_3 \odot B_3) (A_2' \cdot B_2) + (A_3 \odot B_3) (A_2 \odot B_2) (A_1' \cdot B_1) \\ + (A_3 \odot B_3) (A_2 \odot B_2) (A_1 \odot B_1) (A_0' \cdot B_0)$$

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

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

P.N. 56

Ques: 10



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

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

$g = E\bar{x}$

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.

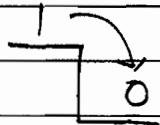
Glitch :-

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

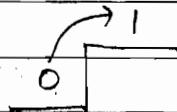
To avoid hazard in ckt ie to get hazard-free ckt, redundant groups should be consider.

Hazard - Type :

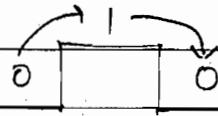
Static '1'



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.

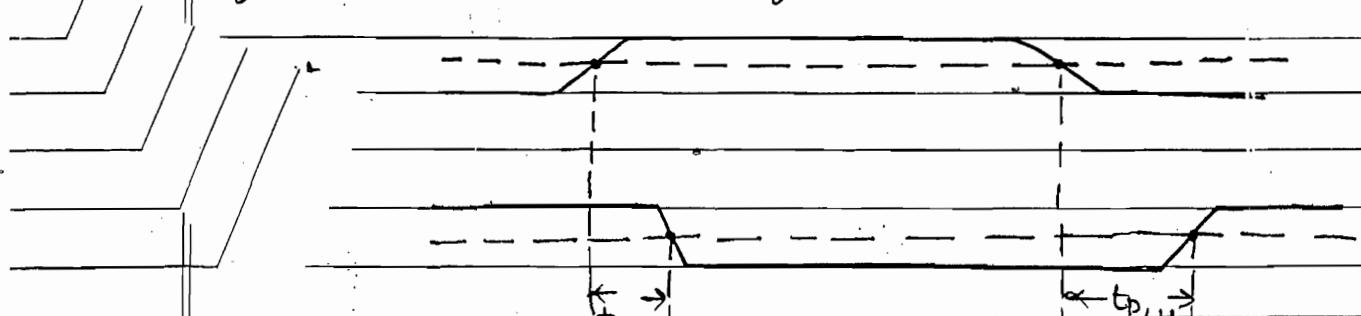
d) Very large scale integration (VLSI) :

no. of logic gates : above 1000,

no. of components : 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.

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

$$I_{cc} = \frac{I_{cch} + I_{ccl}}{2}$$

$I_{cch} \Rightarrow$ Current when all O/P's are high.

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

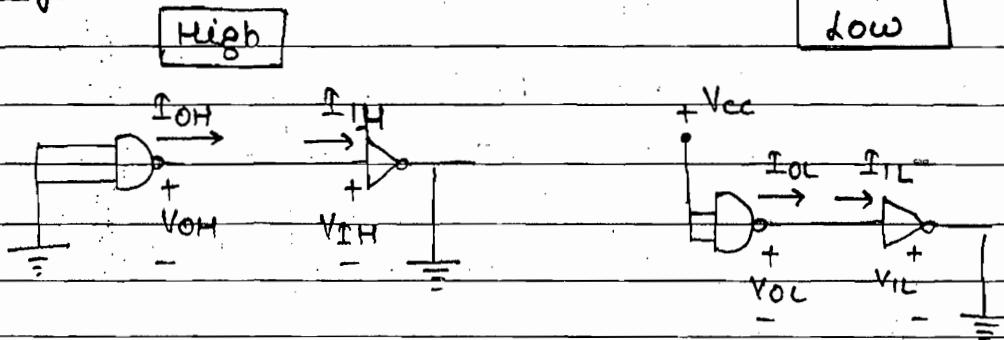
Unit \Rightarrow milli-watts.

* 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: V_{IH} (min) \Rightarrow (High-level input voltage).

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

 V_{IL} (max) \Rightarrow (Low-level input vlg).

It is the maxm vlg limit acceptable for logic '0' as i/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 i/p current) I_{OL} : (Low-level o/p current) I_{IL} : (Low-level i/p current)Formulas:

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

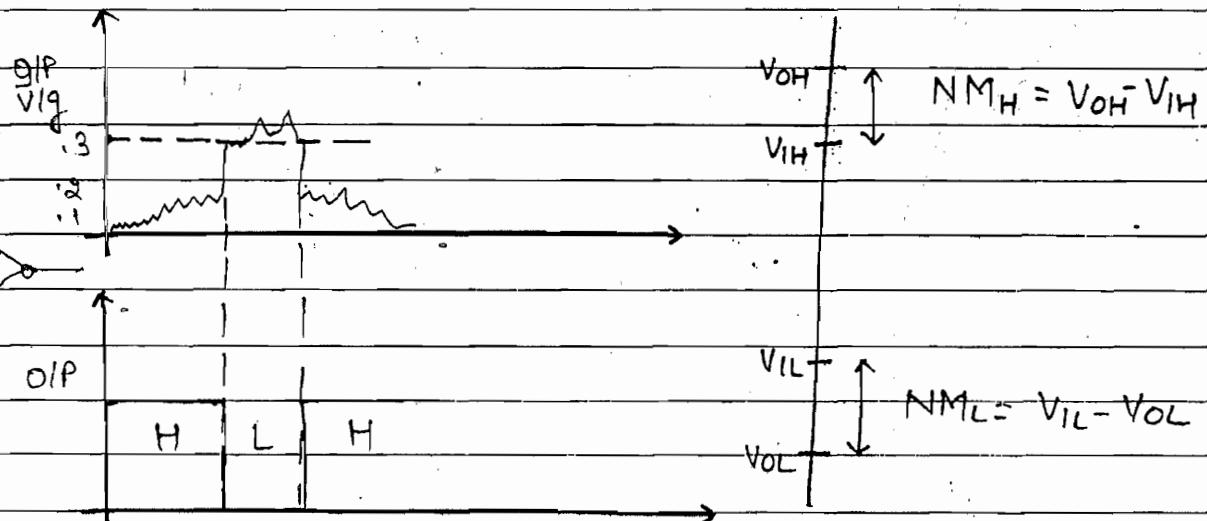
$$\text{Fan-out} = \frac{I_{OL}}{I_{IL}}$$

Note :

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

Noise-immunity : It is ability to withstand the per vlg fluctuations

noise-margin : Measure of noise-immunity.

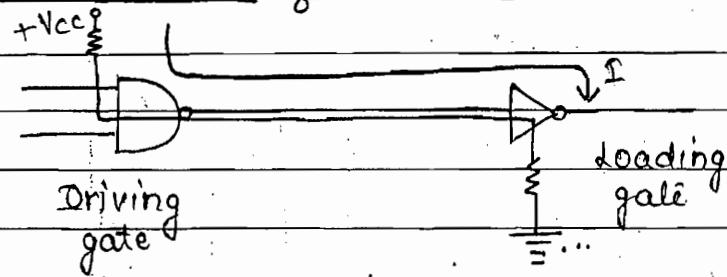


Note :

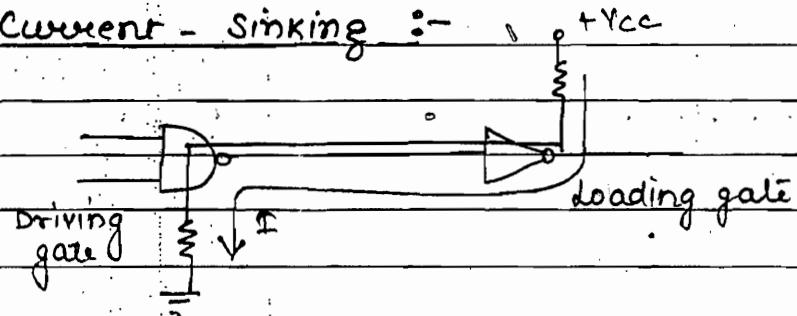
from above two values the least-values should be considered for the overall noise-margin.

- * Operating-Temperatures :
 - 0°C to +70°C
 - ... Industrial Applications
 - ... consumer Applications
 - 55°C to +125°C
 - ... for military applications

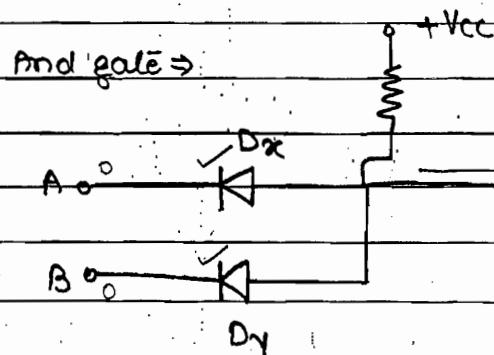
* Current-Sourcing :-



* Current-Sinking :-



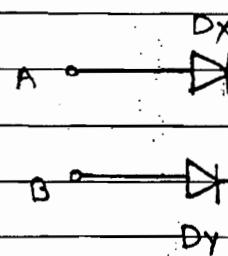
Devices as switching Circuits (logic-gates) :-



A	B	Dx	Dy	y
0	0	✓	✓	0
0	1	✓	x	0
1	0	x	✓	0
1	1	x	x	1

$$y = A \cdot B$$

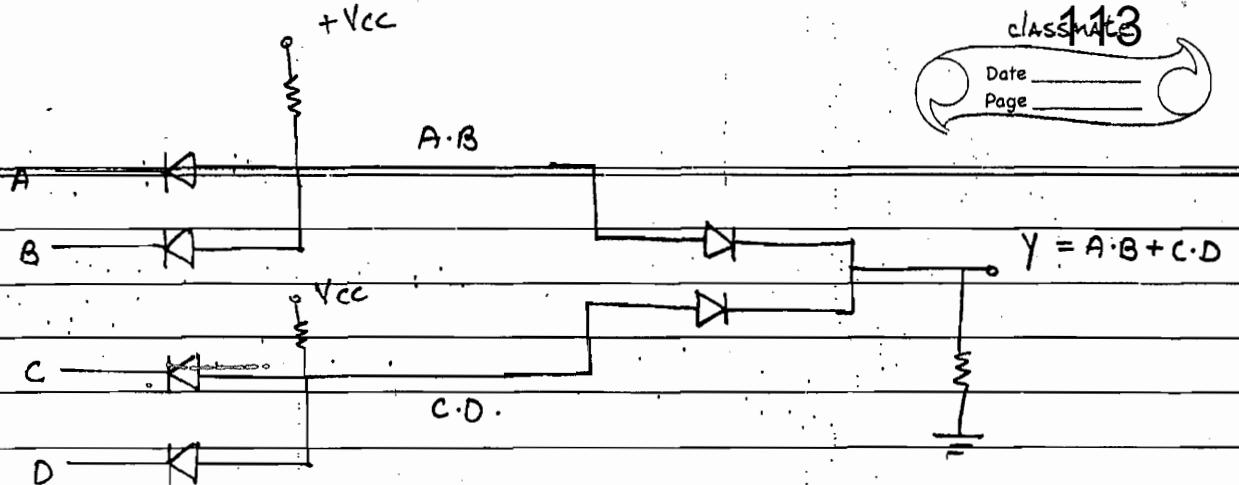
OR-Gate \Rightarrow



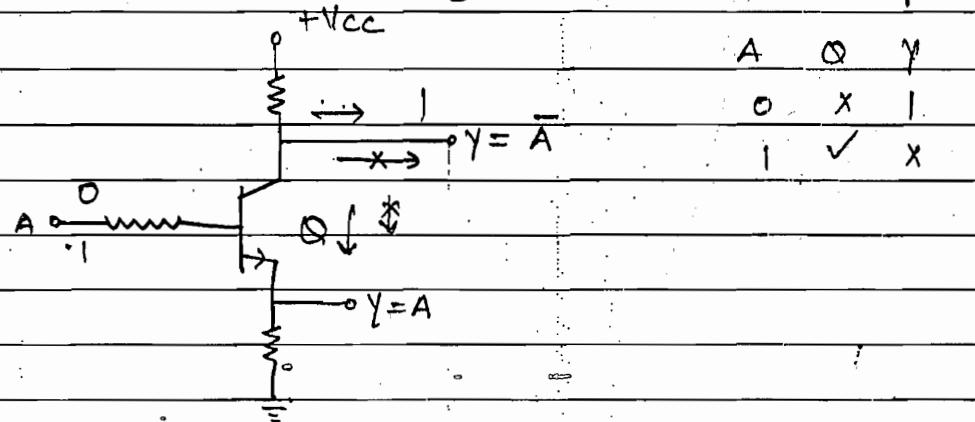
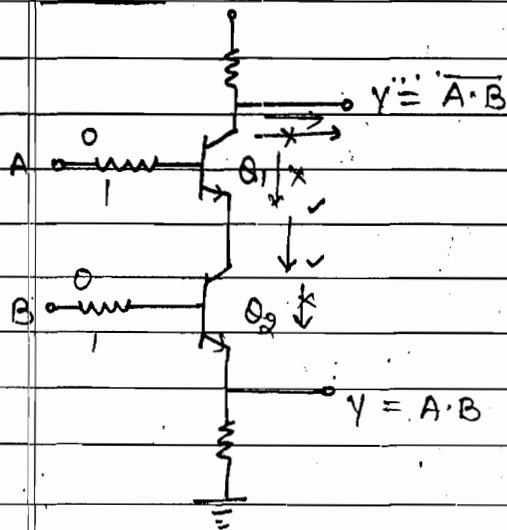
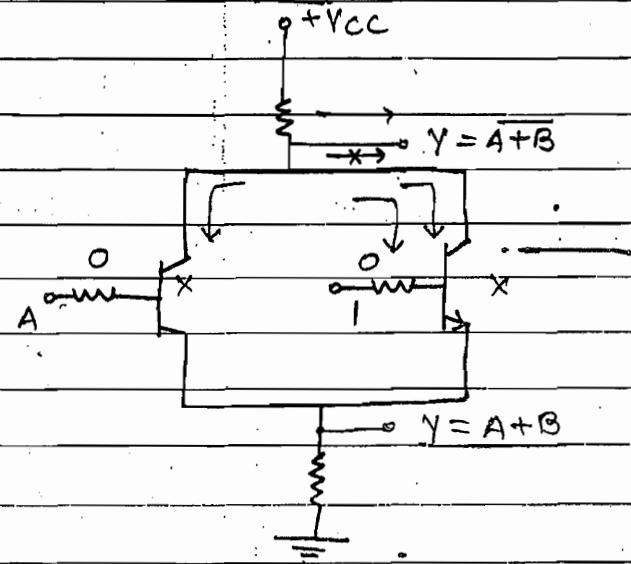
A	B	Dx	Dy	y
0	0	x	x	0
0	1	x	✓	1
1	0	✓	x	1
1	1	✓	✓	1

$$y = A + B$$

EBS Notes



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 .

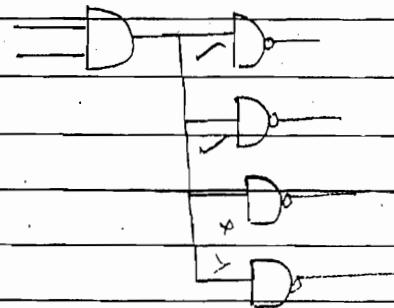
** advantages .

** disadvantages .

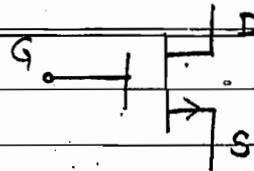
Note :

⇒ 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 .

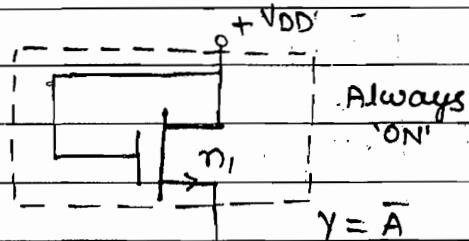


NMOS :-

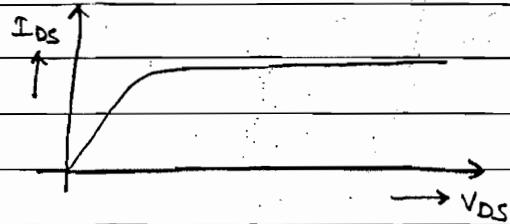
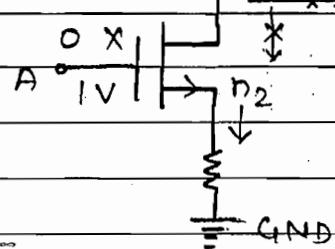
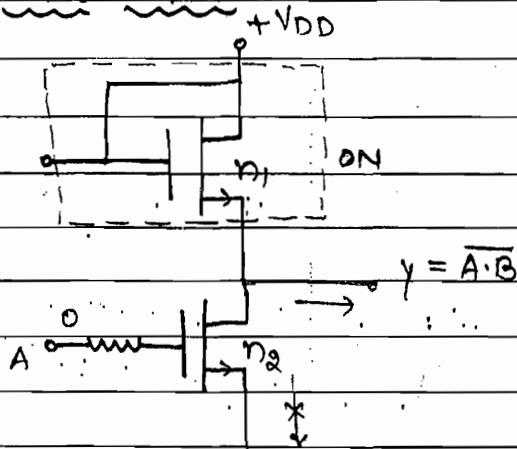


Gate \Rightarrow (+) ve \Rightarrow ON.
 (-) ve \Rightarrow OFF.

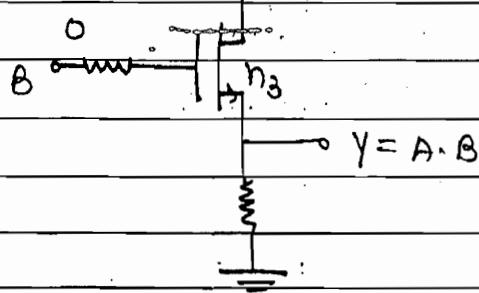
gate diffusion always (+) ve
 always 'ON'

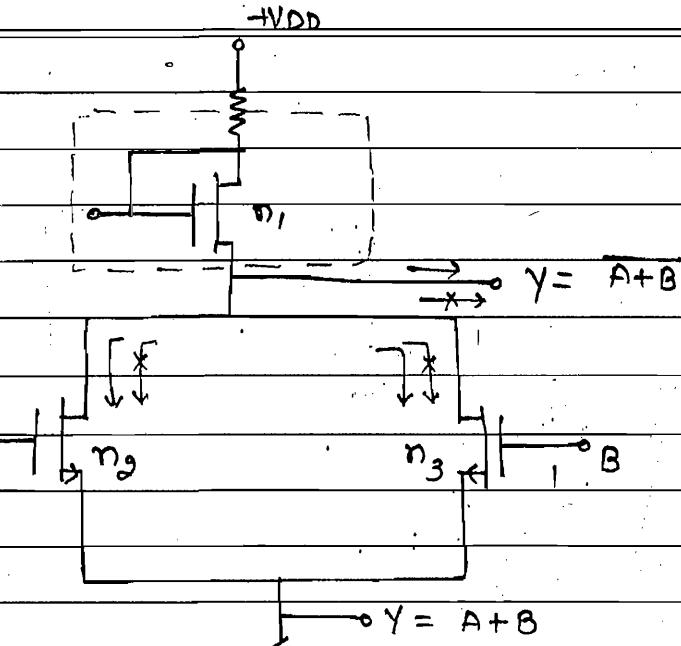


A	n_1	n_2	y
0	\checkmark	X	1
1	X	\checkmark	0

NAND - Gate :

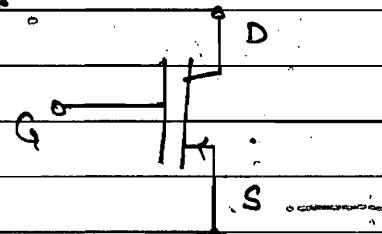
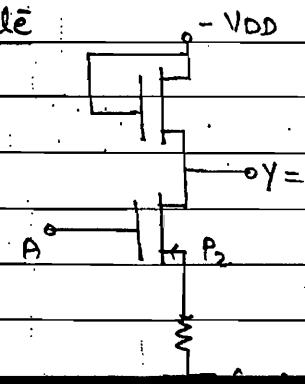
A	B	n_1	n_2	n_3	y
0	0	\checkmark	X	X	1
0	1	\checkmark	X	\checkmark	1
1	0	\checkmark	\checkmark	X	1
1	1	\checkmark	\checkmark	\checkmark	0



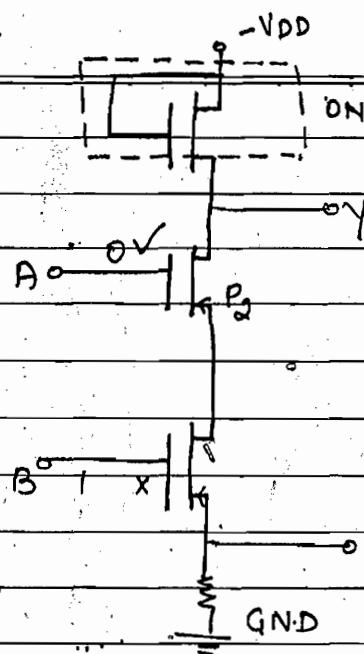
NOR Gate:

A	B	n_1	n_2	n_3	y
0	0	✓	X	1	0
0	1	✓	X	✓	0
1	0	✓	✓	X	0
1	1	✓	✓	✓	0

GND

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

A	P_1	P_2	y
0	✓	✓	1 (GND)
1	✓	X	0 (-VDD)

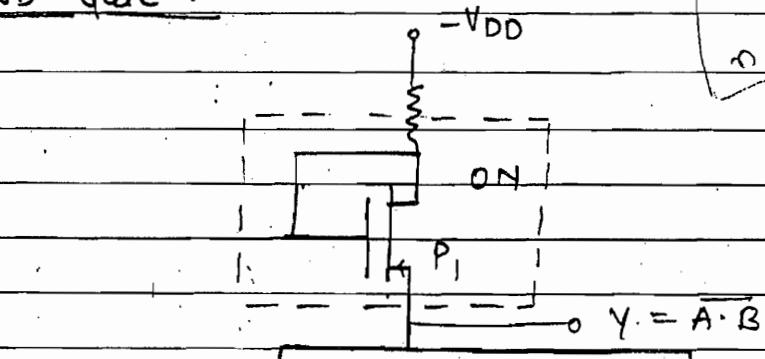
NOR-gate \Rightarrow 

A	B	P ₁	P ₂	P ₃	Y
0	0	✓	✓	✓	1 GND,
0	1	✓	✓	X	0 -VDD.
1	0	✓	X	✓	0 -VDD
1	1	✓	X	X	0 -VDD

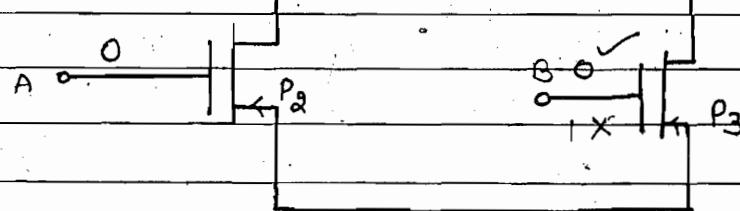
$$\circ Y = A + B$$

gate (\rightarrow)source (\rightarrow) drain (\rightarrow)source
(\rightarrow)

n

NAND-Gate \Rightarrow 

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



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

A B P₁ P₂ P₃ Y

0 0 ✓ ✓ ✓ 1 GND

0 1 ✓ ✓ X 1 GND

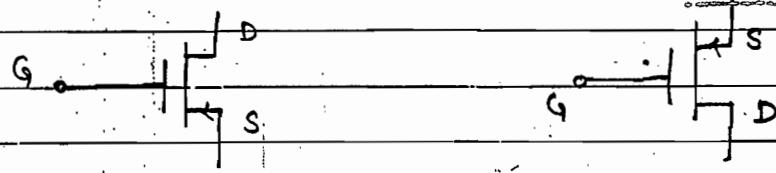
1 0 ✓ X ✓ 1 GND

1 1 ✓ X X 0 -VDD

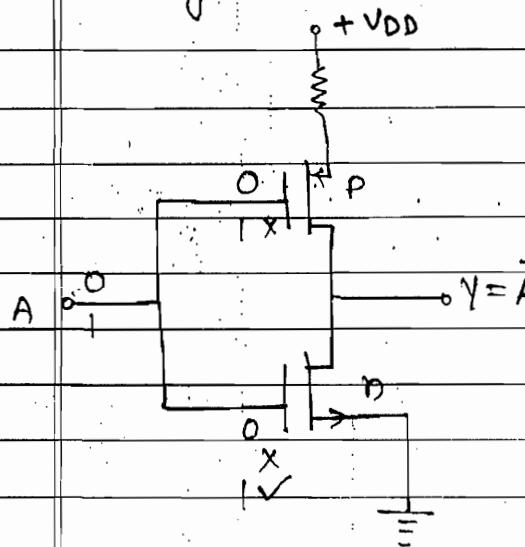
GND

GND

C-MOS Tech. \Rightarrow



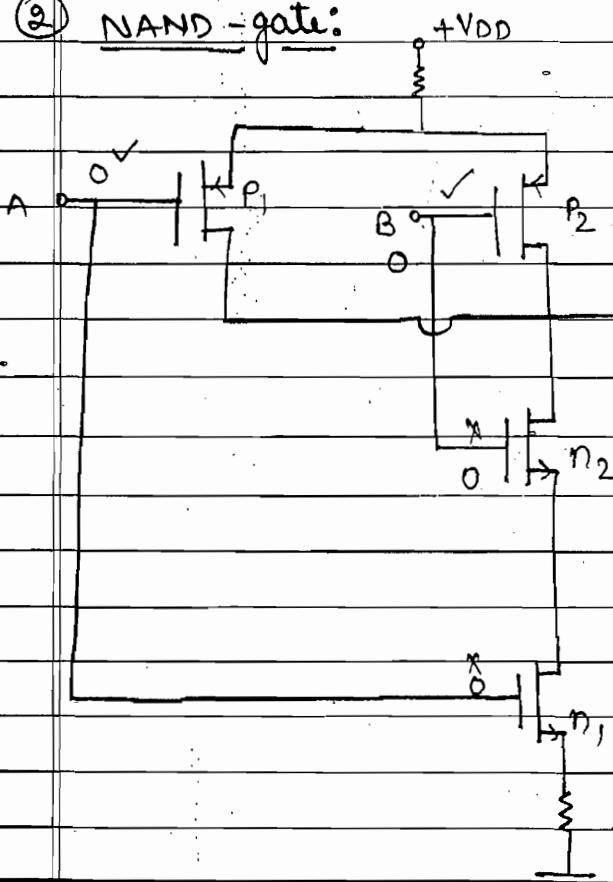
① NOT-gate



A	P	n	γ
0	✓	x	1
1	x	✓	0

+VDD GND

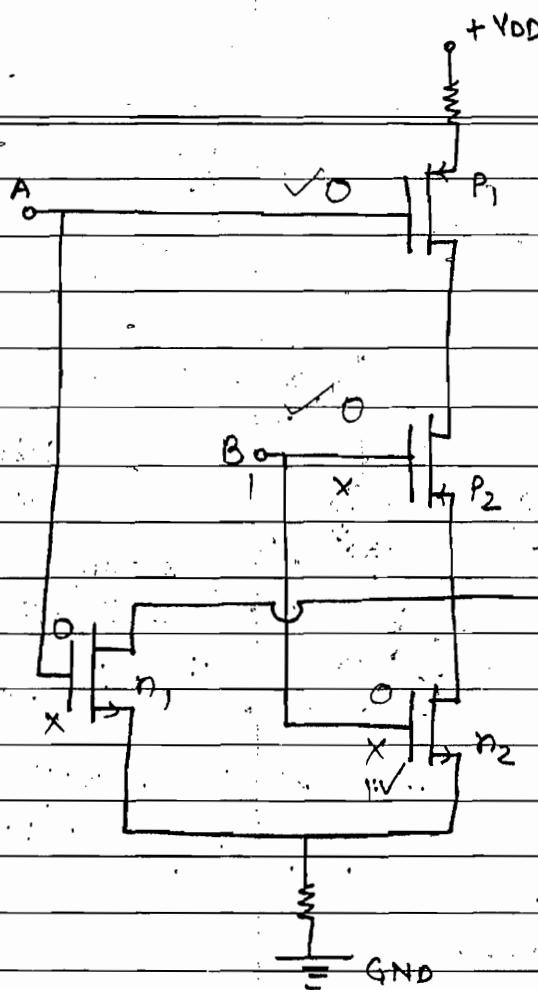
② NAND-gate:



A	B	P ₁	P ₂	n ₁	n ₂	γ
0	0	✓	✓	x	x	1
0	1	✓	x	x	✓	1
1	0	x	✓	✓	x	1
1	1	x	x	✓	✓	0

+VDD GND

(3)

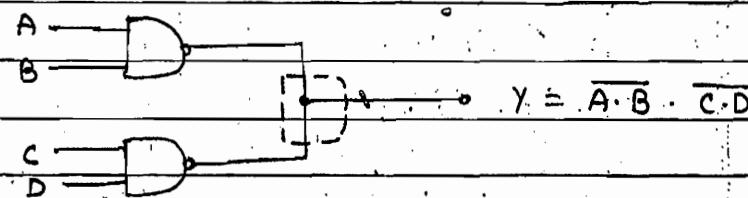
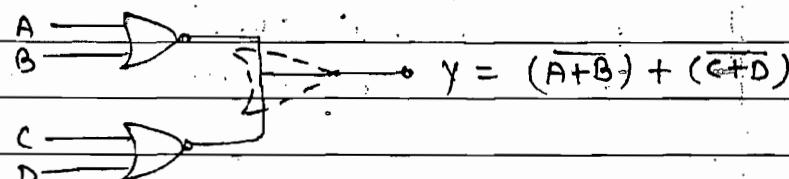


A	B	P ₁	P ₂	n ₁	n ₂	y
0	0	V	V	X	X	1 + V _{DD}
0	1	V	X	X	V	0 GND
1	0	X	V	V	X	0 GND
1	1	X	X	V	V	0 GND

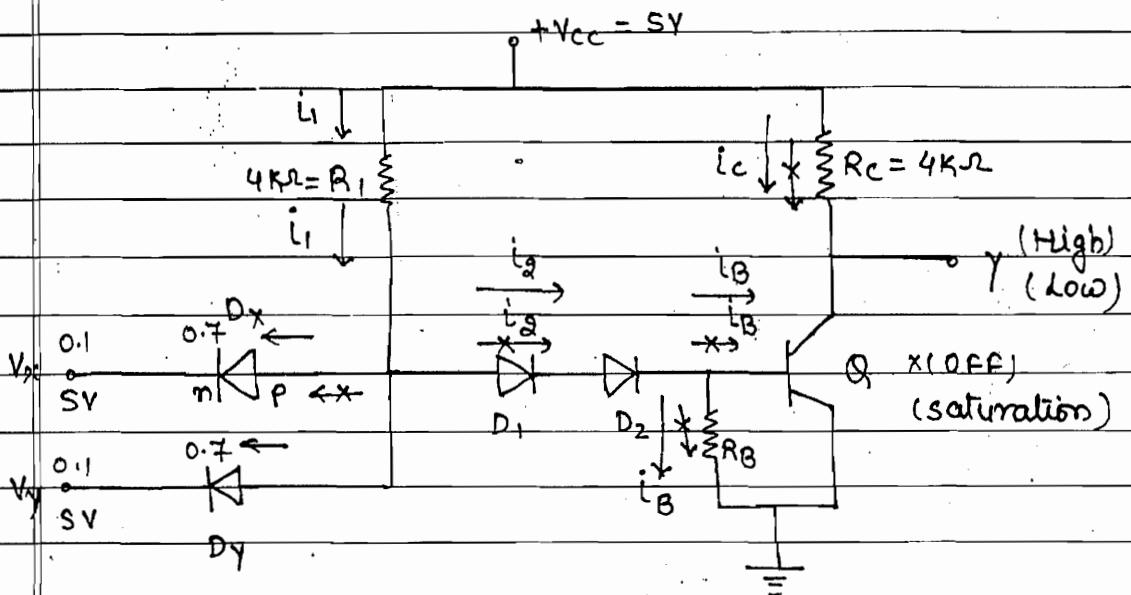
$$y = A + B$$

NOTE :-

* floating input in TTL is taken as: logic '1'

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

Diode

Digital - Transistor - logic (DTL) :-Given : $V_x = 0.1 \text{ V}$

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

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

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

$$\beta = 25$$

Solution \Rightarrow

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

$$i_i = \frac{V_{cc} - V_i}{R_1} = \frac{5 - 0.8}{4k} = 1.05 \text{ mA}$$

$$i_g = i_B = i_R = 0.$$

 $\Rightarrow Q$ is off $y = \text{High (5V)}$

$$\text{case(ii)} \quad V_x = 0.1 \text{ V}, \quad V_y = 5 \text{ V}$$

Same as that of case(i)

 $\Rightarrow Q$ is very off. $y = \text{high (5V)}$

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) $V_x = 5V$, $V_y = 5V$

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

$$i_2 = i_1 = 0.7 \text{ mA}$$

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

$$i_R = 0.08 \text{ mA}$$

$$i_B = i_2 - i_R$$

$$= 0.7 - 0.08$$

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

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

$$i_C = 1.22 \text{ mA}$$

Saturation condi.

$$i_C < \beta$$

$$i_B$$

$$1.22 < 25$$

$$0.62$$

$$1.96 < 25$$

$\Rightarrow Q$ 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

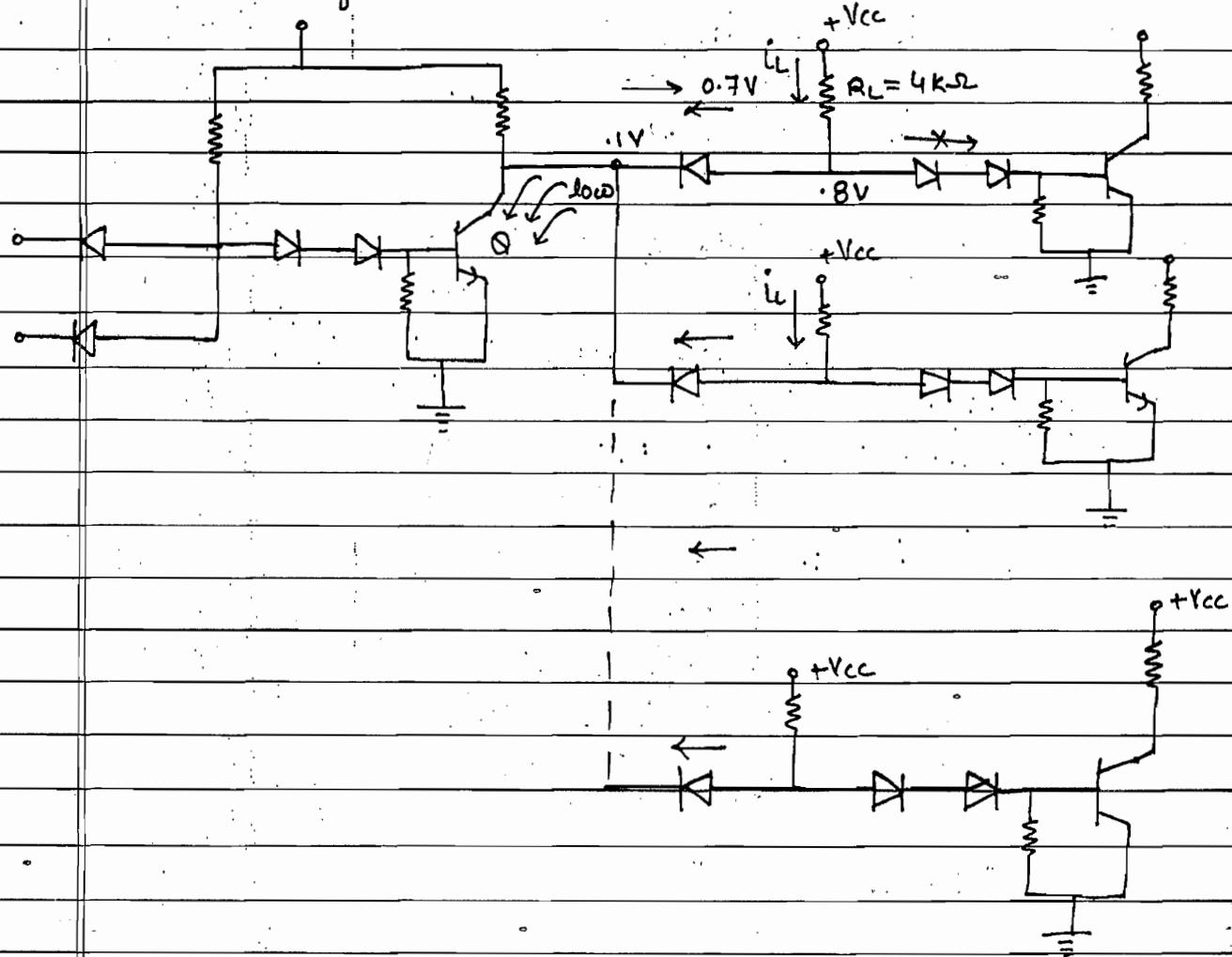
when on adding loads:

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' ie $i_C' = i_C + N \cdot i_L$ and new saturation condition will be $\frac{i_C'}{i_B} \leq \beta \quad \{ \frac{i_C}{i_B} \leq \beta \}$

where $N \rightarrow$ fan-out.



saturation cond'n: $i_C' = i_C + N.i_L$

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

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

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

$$\Rightarrow \frac{1.22 + N(105)}{0.62} = 25 \quad i_C = 1.22 \text{ mA}$$

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

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

In this case the input 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' becoz of which OIP of driving gate alters from high v_{IG} 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_{IG} 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_{IG}: $2.1 - 0.8 = 1.3 \text{ V}$ is the acceptable fluctuating v_{IG} at the OIP of driving-gate so noise margin is 1.3 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 cir if one of the diodes D_1 (or) D_2 is removed?

⇒ fan-out increases , noise - margin reduces .

**

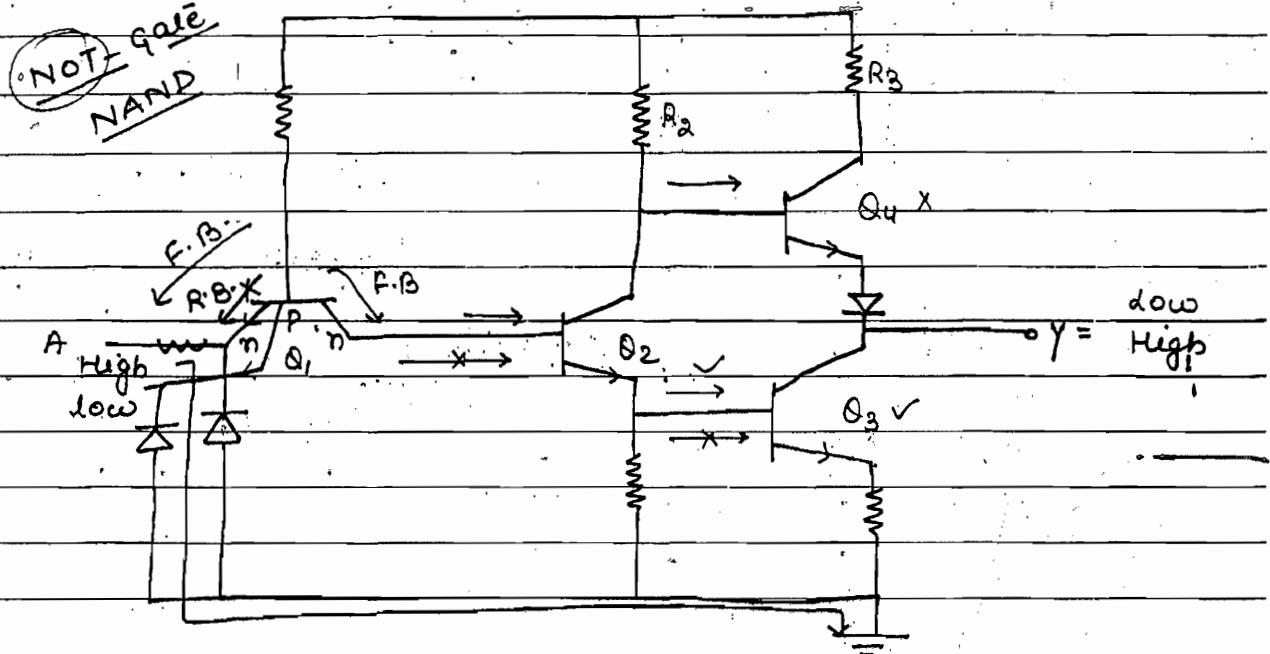
Ques: what will happen if one more diode D_3 is inserted in series with D_1 & D_2 ?

⇒ fan-out reduces and noise - margin increases .

#

High-Threshold-logic (HTL) :-

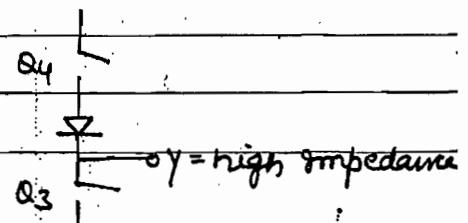
→ if one of the diode in DTL is replace by zener diode , known as HTL. bcoz noise marginal level will be increase .

Transistor-Transistor-logic (TTL) :-

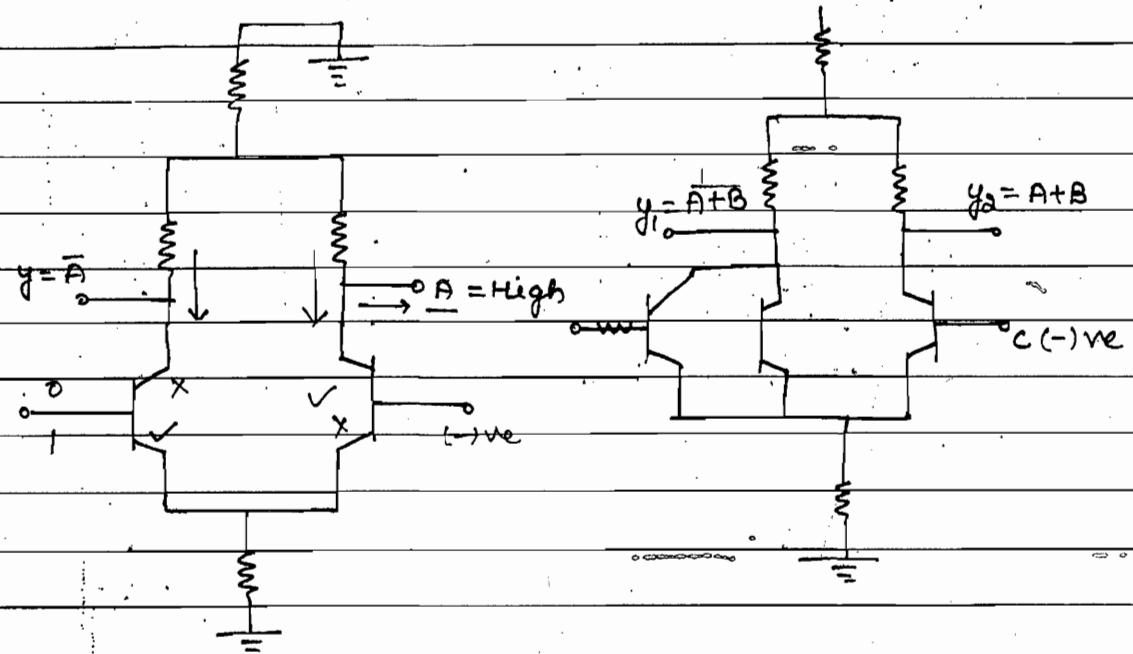
- ⇒ Diode D_1 is used to keep the input Transistor safe from negative voltage fluctuations.
- ⇒ Diode D_2 is used for proper switching output vlg.

* TTL Tech. is of three - types :

- o- Totem - pole TTL :- Q_3 and Q_4 in the given ckt will be 'ON' at a different time so above discuss TTL ckt is Totem - pole TTL.
- o- Open - collector TTL : If the O/P section for the collector of Q_3 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 state state is high - impedance state.

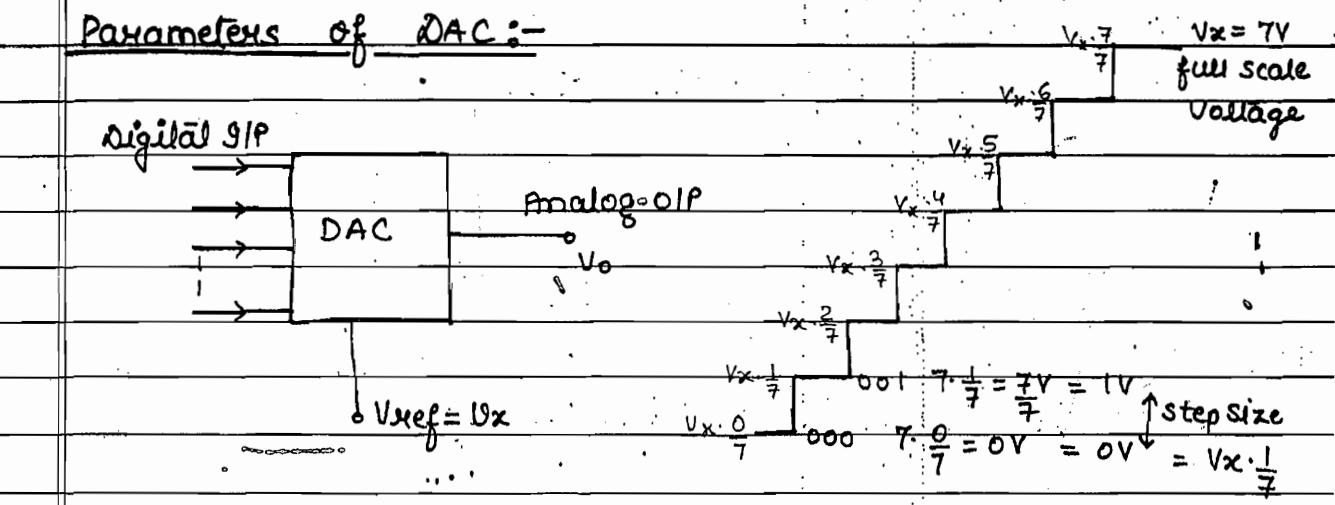


ECL (Emitter-Coupled-logic) :-



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 :-

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

= V_{f.s.}

No. of steps

= V_{FS} = 1 LSB

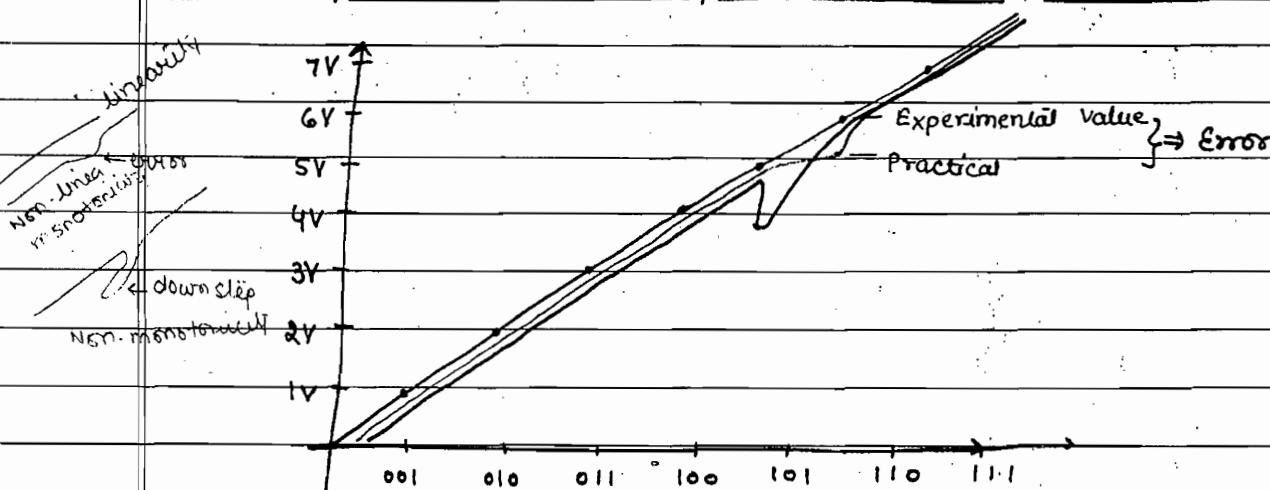
$$\Rightarrow U_o = K [2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \dots + 2^1 b_1 + 2^0 b_0]$$

2^{n-1} Value

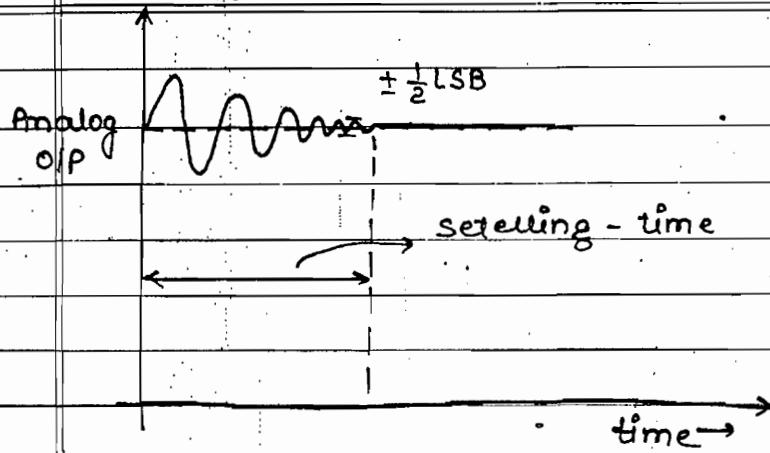
Decimal equivalent of binary data

a) % Resolution :-

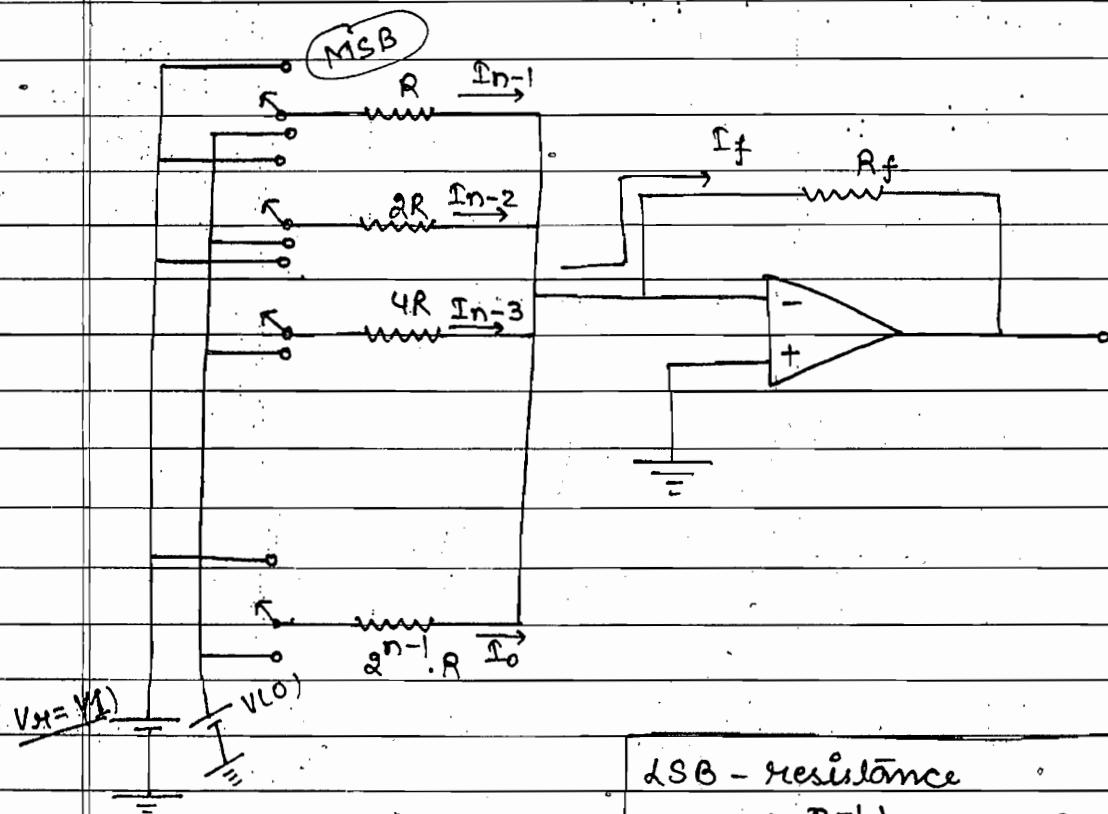
$$\% \text{ Res'l'n} = \frac{1}{2^n - 1} \times 100\%$$

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

c). Settling-time :



Weighted Resistor Type DAC :-



LSB - resistance

$$= (2^{n-1}) \text{ MSB - Resistance}$$

$$V_o = - I_f \cdot R_f$$

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

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

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

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

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

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

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

decimal equi. binary data

where ; $K = \frac{-V_R \cdot R_f}{2^{n-1} R}$

P.N.76Ques: 6

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

Ques: ②

$$V_{SI} = 20V$$

$$11011011 \leftarrow 8 \text{ bit}$$

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

$$V_o = K [2^{19}]$$

$$V_o = K [11 \ 11 \ 11 \ 11]$$

$$20 = K [255]$$

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

$$\rightarrow V_o = \frac{20}{255} \times 2^{19}$$

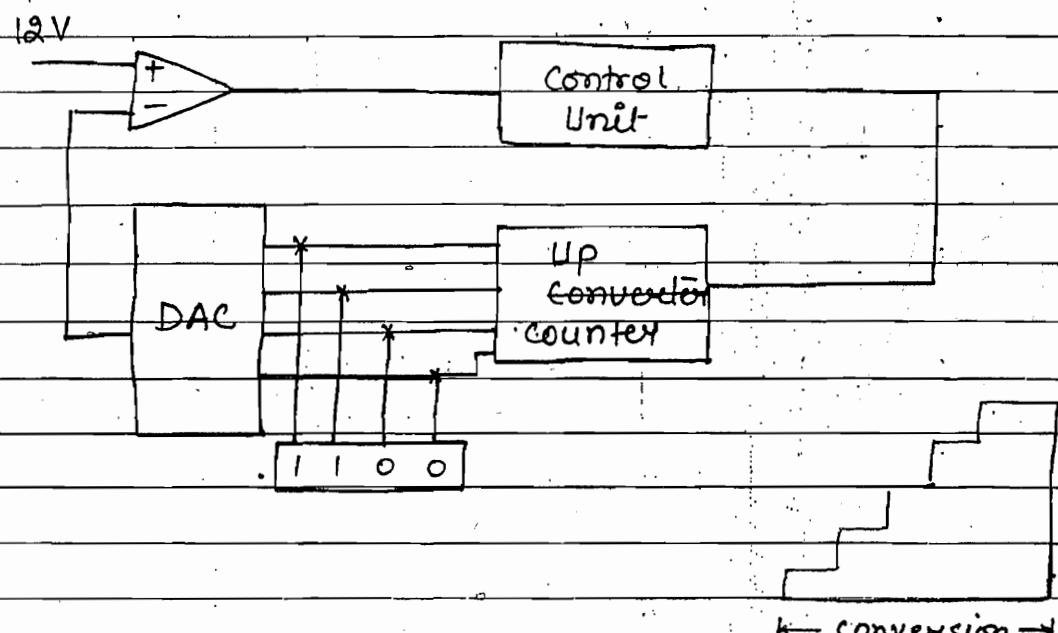
$$V_o = 17.29V$$

8.

ADC :-

Counter-Type-ADC :-

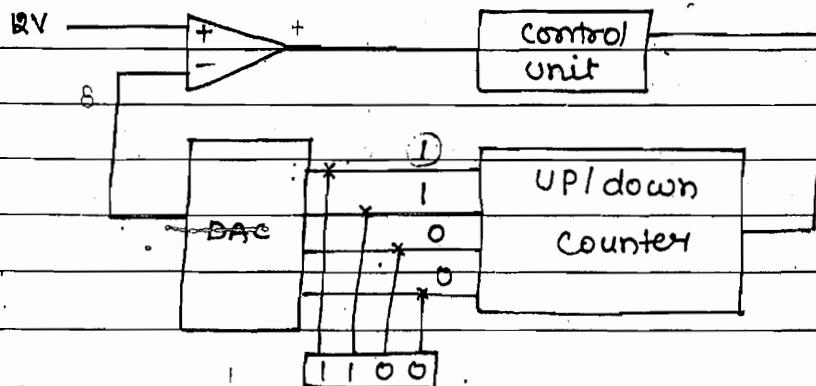
Ramp-Type-ADC :- A



$$t_c(\max) = (2^n - 1) \times \text{clock-cycles}$$

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

2). Successive Approximate ADC :-



operation:

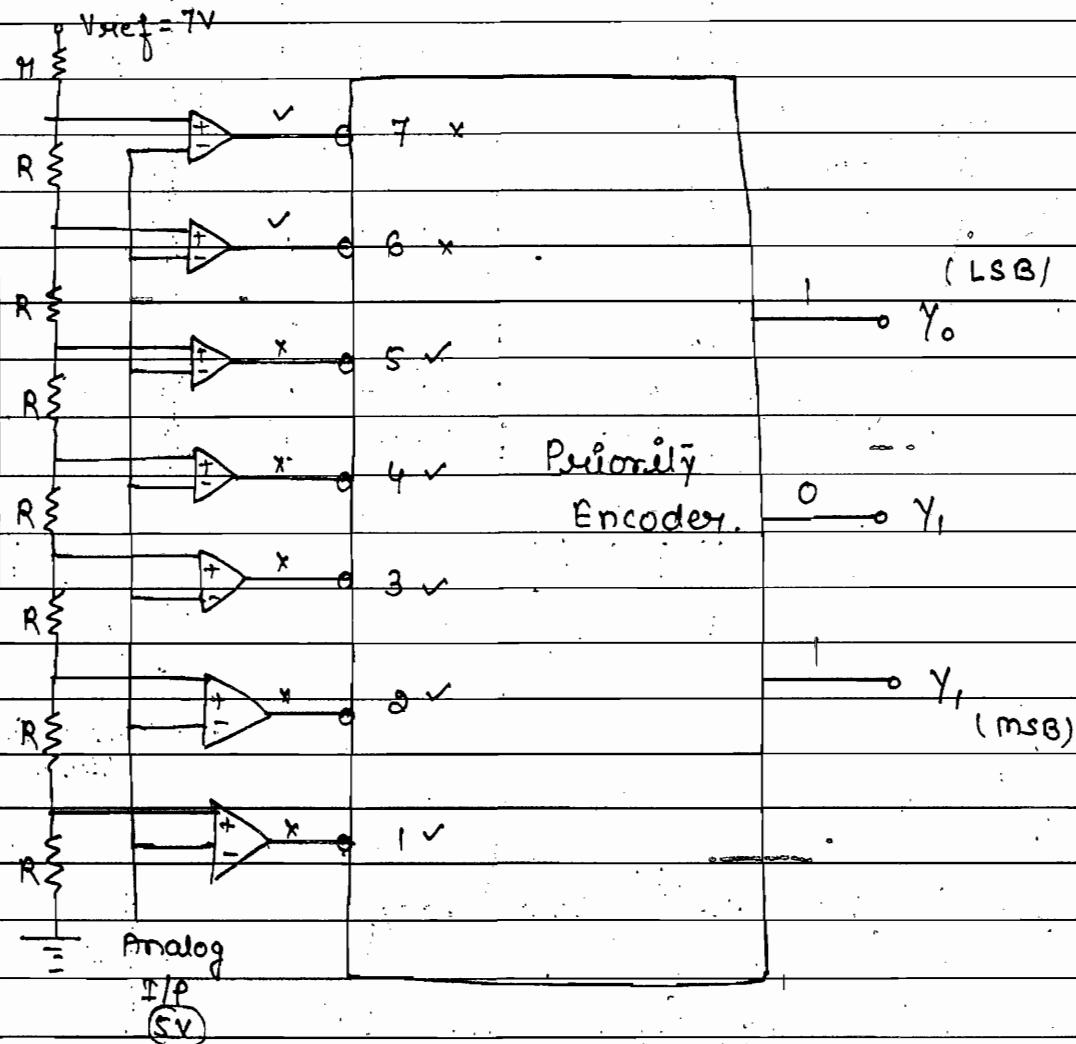
(P.N. 100
Theory book)

$$t_c(\max) = N \times \text{clock-pulses (cycles)} \\ \downarrow \text{no. of bits}$$

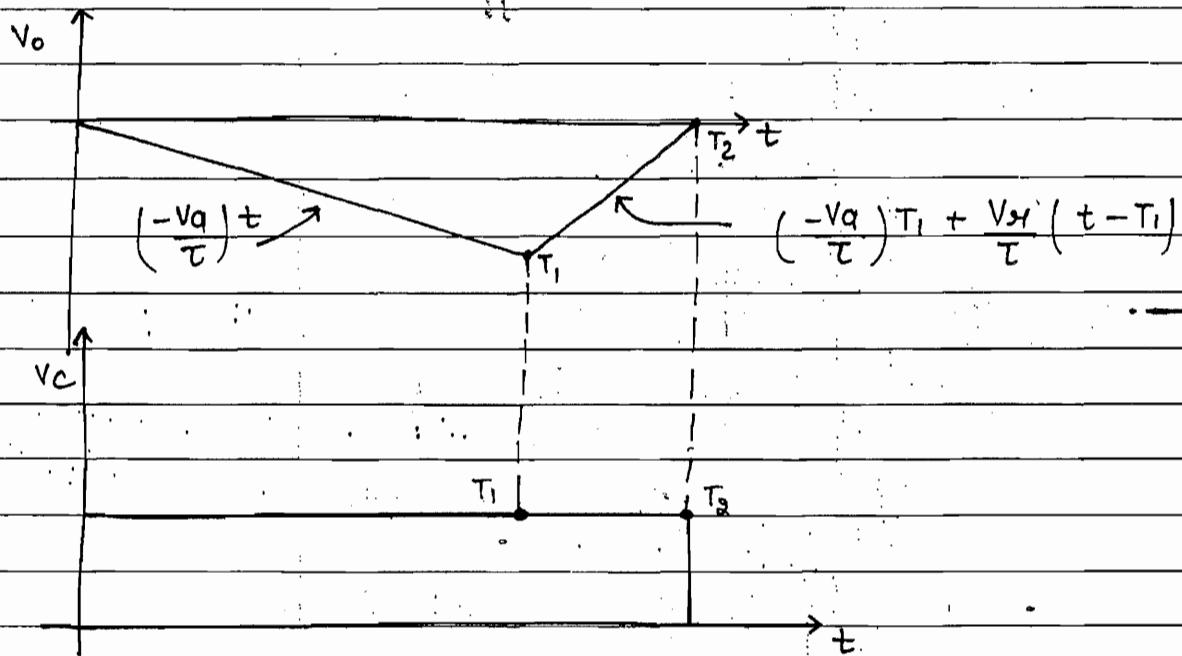
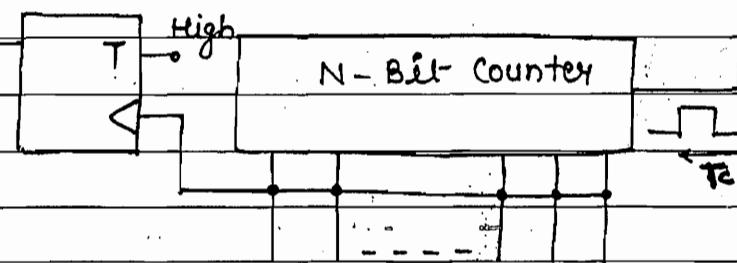
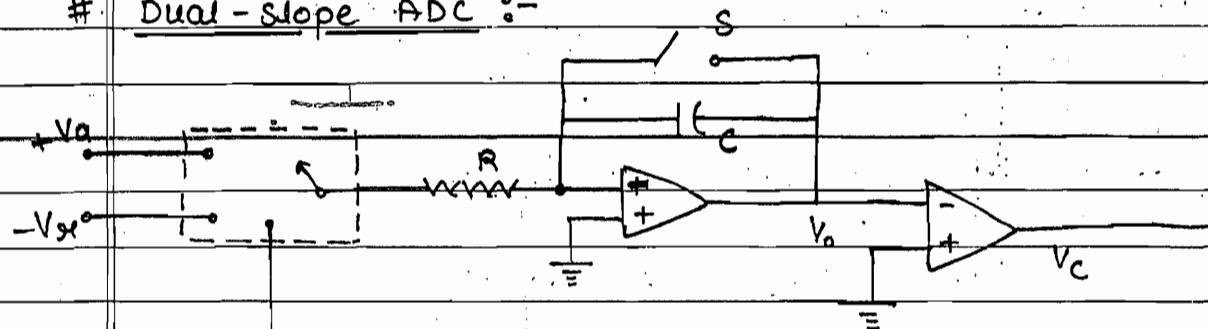
* Flash-Type ADC :-

* 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 $t = T_1$

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

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

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

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

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

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

$$V_a [2^N] = V_r \cdot n$$

$$V_a = V_r / 2^N \cdot n$$

$$\text{if } V_a = 2^N$$

$$V_a = n$$

$$t_{cc(\max)} = T_2$$

$$t_{cc(\max)} = T_1 + n T_c$$

$$= 2^N T_c + n T_c$$

$$t_{cc(\max)} = (2^N + n) T_c$$

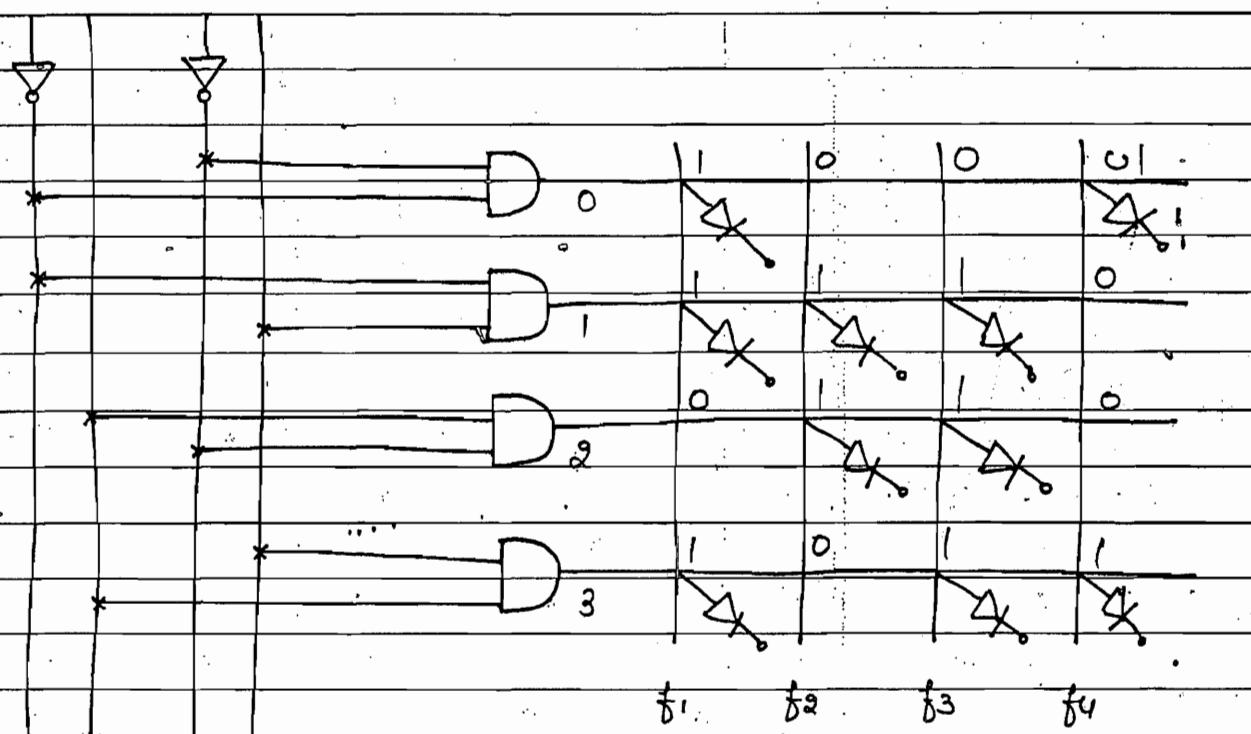
$$T_1 = 2^N T_c$$

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

$N \Rightarrow$ no. of bits

$n \Rightarrow$ no. of counts

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

MEMORY - DESIGNING :-

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

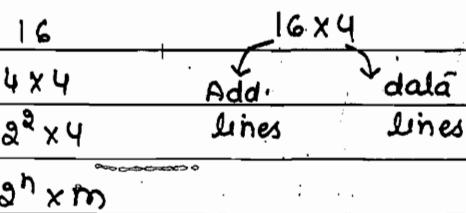
$$f_2 = \sum m(1, 2)$$

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

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

$n \rightarrow$ select-lines

$m \rightarrow$ data-lines



A B C f₆ f₅ f₄ f₃ f₂ f₁ 138 CLASSMATE

0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0

Date _____

Page _____

2	0	1	0	0	0	0	1	0	0	4
3	0	1	1	0	0	1	0	0	1	9
4	1	0	0	0	0	0	0	0	0	16
5	1	0	1	0	1	1	0	0	1	25
6	1	1	0	1	0	0	0	0	0	36
7	1	1	1	1	1	1	0	0	0	49

A

3x8

B

C

f₆

f₅

f₄

f₃

f₂

f₁

Rom-size

2ⁿ x m

2³ x 6

8 x 6

= 48

Reduced-Rom

2ⁿ x m

2³ x 4

8 x 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.

