

Digital

theorypoint.com

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

1. Producing the constt: (0, 1) [null, identity operation]
2. Unary operations : [Complement, Transfer]
Unary
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	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

$f_0 = 0 \Rightarrow$ Null-operation.

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

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

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

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

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

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

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

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

$$= x'y + xy'$$

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

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

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

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

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

$$f_9 = x \oplus y \quad \text{Ex-NOR}$$

$$= xy + x'y'$$

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

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

$$x < y \quad [\text{if } y \text{ then } x]$$

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

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

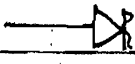
$$x > y \quad [\text{if } x \text{ then } y]$$

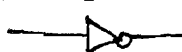
$$f_{14} = \overline{x \cdot y} \quad (\text{Nand})$$

$$x \uparrow y$$

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

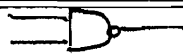
Logic-Gate Symbols :-

Buffer —  

NOT — 

AND — 

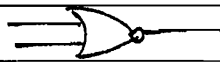
NAND



OR



NOR

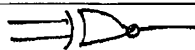


EX-OR



AUA

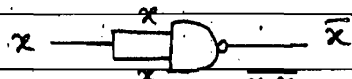
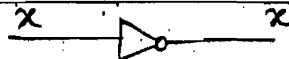
EX-NOR



NOTE:

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

⇒



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

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.

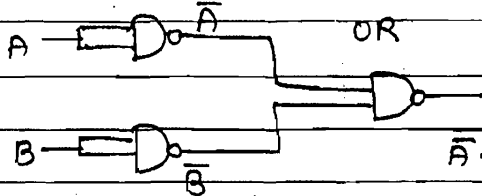
NOT



AND

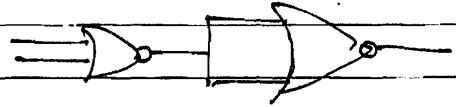
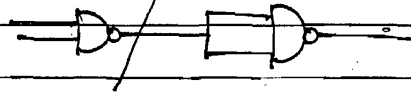
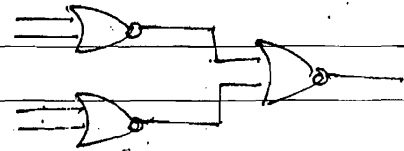
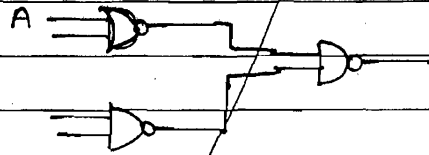
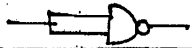


OR

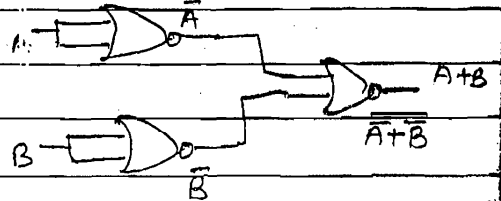
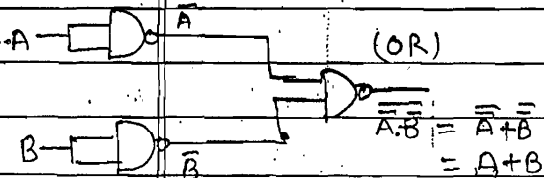
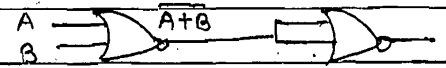
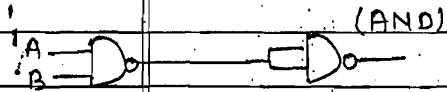
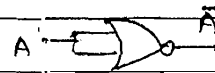
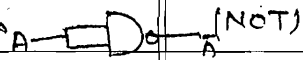


$$\bar{A} \cdot \bar{B} = A + B$$

NOR



Practise :

NAND - GatesNOR - Gates

Duality:Step: 1 Interchange the operators $[\cdot, +]$ Step: 2 Interchange the identities $[0, 1]$

AND-LAWS

OR

eg.. ① $x \cdot \bar{x} = 0$

$\bar{x} + x = 1$

AUF

ECS

eg: ② $x + 0 = x$

$x \cdot 1 = x$

eg.. ③ $f = (A \cdot B) + C$

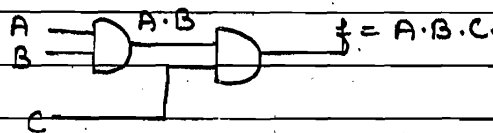
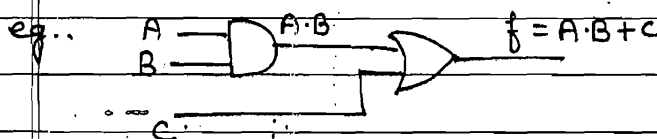
$f^D = ?$

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

$\Rightarrow \boxed{(f^D)^D = f}$

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 :-

AUF

ECS

Short-Cut

Duality

AND - OR

OR - AND

NAND - NAND

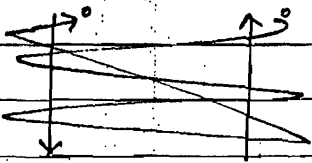
NOR - NOR

NOR - OR

NAND - AND

OR - NAND

AND - NOR



⇒ * 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 $x \rightarrow \neg x$

$x \rightarrow \neg x$

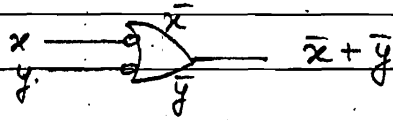
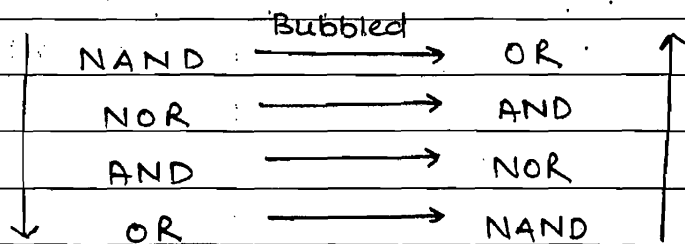
② $x \quad y \rightarrow \text{NOR} \quad \overline{x+y} = \bar{x} \cdot \bar{y}$

$x \quad y \rightarrow \text{Bubbled-OR} \quad \bar{x} + \bar{y}$

Bubbled-OR

Alternative - Logic - Gates :-

eg.. $x \quad y \Rightarrow \text{AND Gate} \quad \bar{x} \cdot \bar{y} = \overline{x+y}$

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

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

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

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

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

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

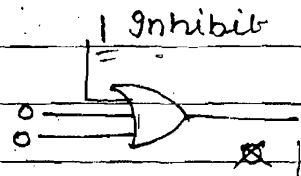
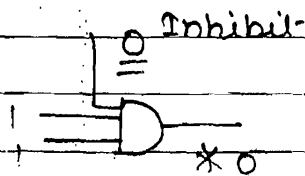
Ex-OR specialities :-

$$\begin{array}{lcl}
 x \oplus x & = & 0 \quad \uparrow \\
 x \oplus x' & = & 1 \\
 x \oplus 1 & = & x' \\
 x \oplus 0 & = & x \quad \downarrow
 \end{array}$$

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

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

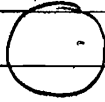
$$x \oplus y \oplus z \Rightarrow x \odot y \odot z$$

Inhibiting the logic-gates :-Short-Cut :-

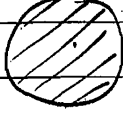
	Disable	Enable
NAND	L	H
NOR	H	L
AND	L	H
OR	H	L

Venn Diagram :-

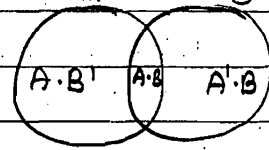
$A=0$



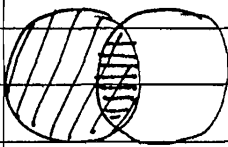
$A=1$



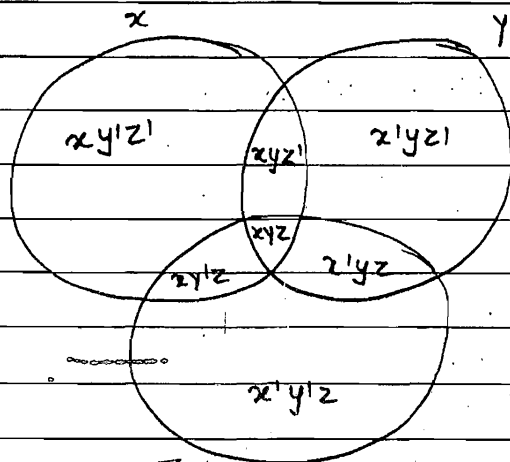
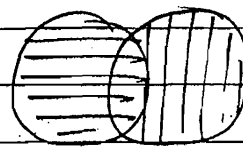
$$A+B = \bar{A}B + A\bar{B}$$



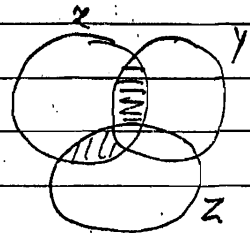
$$\Rightarrow x + xy = x$$



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



eg. ①



$$\Rightarrow f = xyz' + xy'z$$

LOGIC - MINIMIZATION :-

Boolean - Laws :-

NOT :

$$\text{if } x = 0$$

$$\bar{x} = 1$$

$$(x')' = x$$

AND :

$$x \cdot x = x$$

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

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

OR :

$$x + x = x$$

$$x + 0 = x$$

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

$$x + 1 = 1$$

Commutative - law :-

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

$$x + y = y + x$$

Associative - law :-

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

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

Distributive laws :-

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

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

De-Morgan's - law :-

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

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

Consensus - Theorem :-

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

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

must be there

Transposition - Theorem :-

operators and associations can be interchange.

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

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

Redundant lateral Rule (RLR) :-

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

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

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

Absorption-law :-

Absor'n-law :-

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

$$\rightarrow x + xy = x$$

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

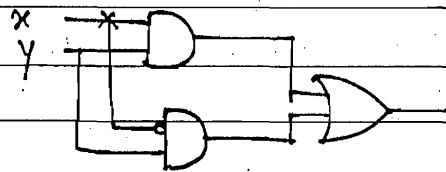
SOP :-

• - (Minterm)

• - (AND-OR)

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

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



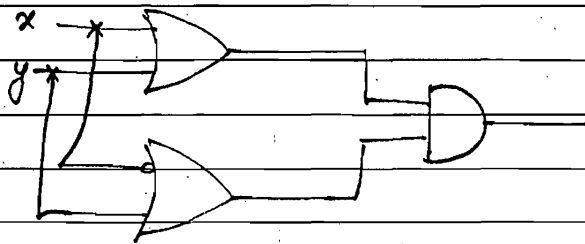
POS :-

(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

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

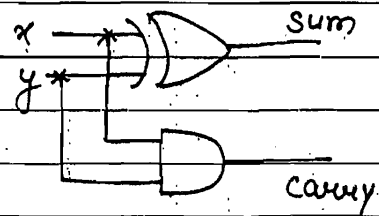


Half-Adder :-

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

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

(carry)
(x · y)



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

Complemental Boolean - Expression :-

Step: ① duality

Step: ② Complement the individual variable.

eg.. $f = A' \cdot B + C$
 $\bar{f} = ?$

Step: ① $(A' + B) \cdot C$

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

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

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

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

→ $\bar{f} = (x+\bar{y})(\bar{x}+\bar{y}) \dots$ complementⁿ of indiv.

⇒ 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 indiv.)

Home-work :

Workbook - Problems : P.No. 50

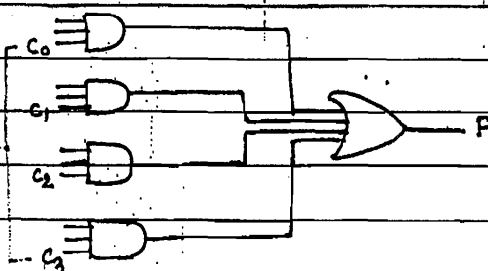
Q:12 Let $F(A,B) = A \oplus B$

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

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

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

Q:13



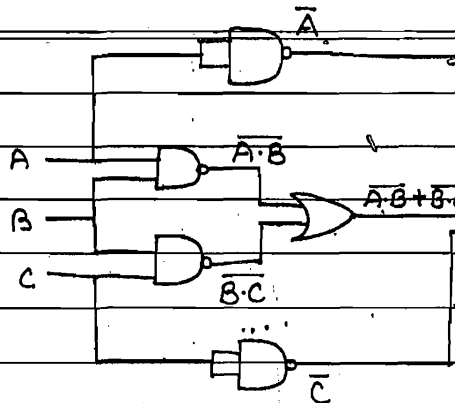
EXOR = $AB + \bar{A}\bar{B} = A + B$

⇒ $c_0, c_1, c_2 = 1$

⇒ $c_0, c_1, c_2 = 1$

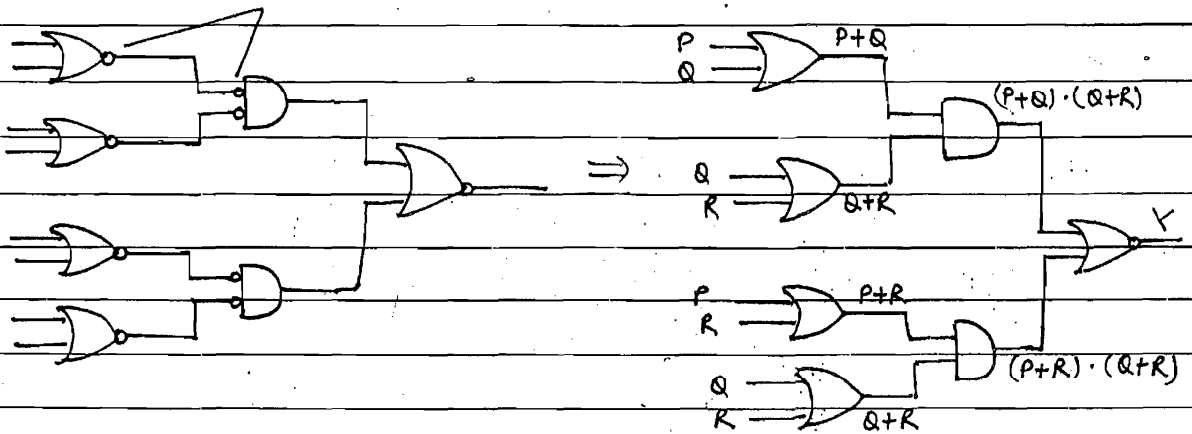
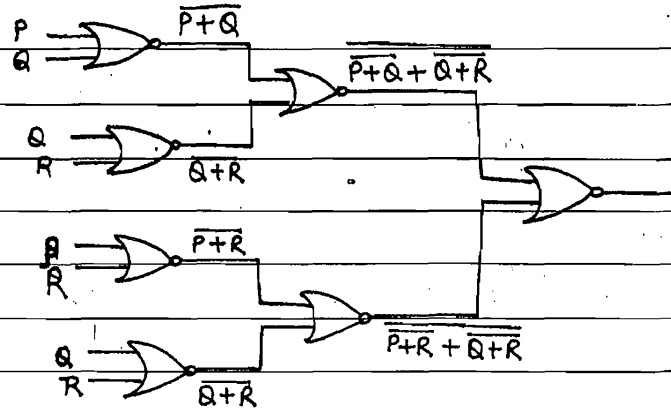
$(0, 1, 1, 1)$

Ques: 14



$$\begin{aligned} \bar{A} + \bar{A} \cdot \bar{B} + \bar{B} \cdot \bar{C} + \bar{C} &= \bar{A} + \bar{C} + \bar{A} \bar{B} \bar{C} \\ &= \bar{A} + \bar{C} + \bar{A} + \bar{B} + \bar{C} \\ Y &= \bar{A} + \bar{B} + \bar{C} \end{aligned}$$

Ques: 15



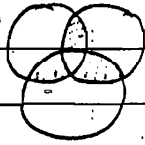
$$Y = (P+Q) \cdot (Q+R) + (P+R) \cdot (Q+R)$$

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

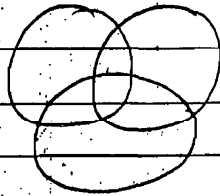
$$Y = (Q+R) \cdot (P+Q+P+R) = (Q+R) \cdot (P+Q+R)$$

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

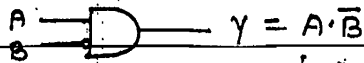
Ques: 16



Ques: 18



Ques: 20

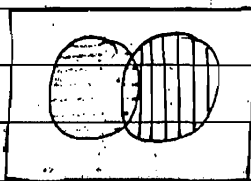


Ques: 21

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

Ques: 24

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



CONVERSION FROM SOP-POS and POS-SOP :-

$$\Sigma m \Rightarrow \text{SOP}$$

$$\Pi m \Rightarrow \text{POS}$$

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

$$\begin{array}{cc} 00 & 10 \\ \downarrow & \downarrow \\ \bar{x} \cdot \bar{y} & x \cdot \bar{y} \end{array}$$

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

$$\begin{array}{cc} 00 & 10 \\ \downarrow & \downarrow \\ x+y & \bar{x}+y \end{array}$$

$$\begin{array}{cc} 00 & 01 \\ 01 & 10 \\ 10 & 11 \end{array}$$

$$\textcircled{3} \Sigma m(0, 2) = \Pi M(1, 3)$$

$$\downarrow$$

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

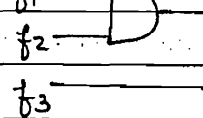
43-bit

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

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

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

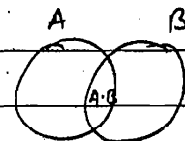
$$f_1 \cdot f_2 = \Sigma m(2, 3)$$



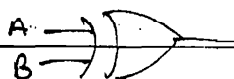
$$f = f_1 f_2 + f_3$$

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

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



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



$$f = ?$$

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

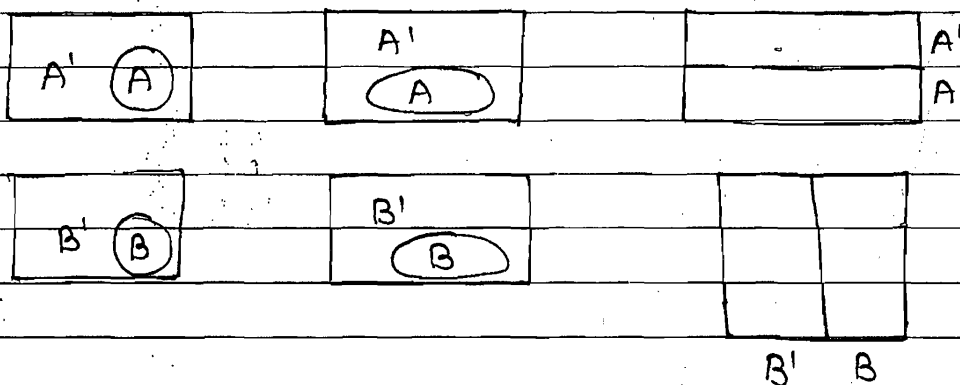
$$\bar{A} \cdot B = \Sigma m(3, 5, 6)$$

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

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

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

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



K-Map :-

	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 Miniterm of SOP - Exp.
2. Check the K-Map for 1's which are not adjacent to any other 1's. They are isolated miniterm. They are to be read as they are becoz they can't be combine.

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 :

① $f(A, B)$

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

② $f(A, B, C)$

$\downarrow$ MSB $\downarrow$ LSB

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	$\bar{A}\bar{B}\bar{C}_0$	$\bar{A}\bar{B}C_1$	$\bar{A}BC_2$	$\bar{A}B\bar{C}_3$
A	$A\bar{B}\bar{C}_4$	$A\bar{B}C_5$	ABC_6	$AB\bar{C}_7$

③ $f(A, B, C)$

$\downarrow$ MSB $\downarrow$ LSB

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	$\bar{C}\bar{A}\bar{B}_0$	$\bar{C}\bar{A}B_1$	$\bar{C}AB_2$	$\bar{C}A\bar{B}_3$
C	$C\bar{A}\bar{B}_4$	$C\bar{A}B_5$	CAB_6	$CA\bar{B}_7$

4. $f(A, B, C)$

$\downarrow$ MSB $\downarrow$ LSB

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

Ques:

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

$\downarrow \quad \downarrow \quad \downarrow \quad \searrow$
 000 001 010 110

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

1st method

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

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

2nd

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

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

eg.

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

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

eg.

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

$$f = C$$

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

**

1 ← Pair	for Pair : $(n-1)$; $n \rightarrow$ Variable no.
2 ← Quad	for Quad : $(n-2)$
3 ← Octet	for Octet : $(n-3)$

eg. In a 5-variable K-Map, octet gives: $5-3 = 2$ literals

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

eg.. ①

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

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

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

EPI EPI X[R.P.I.]

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

eg.. ②

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

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

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

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 implicant.

Essential Prime Imp. : EP1 is P1 only in which atleast a single 1s which is only 1-time grouped (Uniquely - grouped).

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

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

Selective Prime Imp. : It is the P.I. for which it was not at decided for the selection.

Note:

- o- Max-term is known as false - Minterm.
- o- 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 $B\bar{C}$

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

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

$\bar{A}\bar{B}$ (0, 1)

EP1 (P1)

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

P1 (P1)

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

P1 (P1)

AB (6, 7)

E.P.I. (P1)

S.P.I.

Imp. $\rightarrow$ 5

P.I. $\rightarrow$ 4

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		1	
A			1	1

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

Imp. $\rightarrow$ 4

Non-imp. $\rightarrow$ 1

P.I. $\rightarrow$ 2

E.P.I. $\rightarrow$ 2

eg. ③

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

$\bar{A}\bar{B}$

AB

EPI
(PI)

EPI
(PI)

DIAGONAL GROUPING :-

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

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

$$f = A \odot B$$

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

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

$$f = A \oplus B$$

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

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

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

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

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

$A \oplus B \Rightarrow$ 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 1	① 3	2 2
A	4 4	① 5	7 7	① 6

$$A[B \oplus C]$$

eg.. ②

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

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

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

POS forms in K-Map:-

* Step: ① operator change.

* Step: ② complement the given variable.

eg-

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

	B^c	B^c	B^c	B^c
A^c	0 0	1 1	0 3	2 2
A	4 4	5 5	0 7	6 6

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

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

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

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

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

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

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

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

SOP to SSOP [Canonical Form]:

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

Method: I

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

Method: II

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

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

Ques: $f = A + BD$

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

$$\begin{aligned} f &= \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} \\ &+ A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D} + A\bar{B}\bar{C}D \\ &+ A\bar{B}CD + A\bar{B}\bar{C}D + A\bar{B}C\bar{D} \\ &+ A\bar{B}CD + A\bar{B}\bar{C}D \end{aligned}$$

DON'T-CARE - Conditions : There may 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. $f = \sum m(1) + \phi(3)$

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

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

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

eg.

	$\bar{C}\bar{D}$	$\bar{C}D$	CD	$C\bar{D}$
$\bar{A}\bar{B}$				
$\bar{A}B$		1	1	
AB	x	x1	x1	x
$A\bar{B}$			x	x

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

with $\phi \Rightarrow BD$

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

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

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

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

Variable - Entered - Mapping :-

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

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

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

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

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 on 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 forms.

eg..

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0 ₀	A ₁	1 ₃	$\bar{A}$ ₂
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	$= yz$ [m_3 is fully covered]
x	0	0	X	0	

→ so need to be considered in final grouping]

Step: ②

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

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

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

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

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

↳ Redundant

eg.

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

method : step: (1)

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

$f = \bar{A}B$

step: (2)

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

$f = CB$

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

8/12/13 5-Variable K-MAP :-

Sunday

	$\bar{A}$	$\bar{B}\bar{C}$ will form Octet	A
$\bar{D}\bar{E}$	$\bar{D}\bar{E}$	DE	DE
$\bar{B}\bar{C}$	1 0	1 1	1 2
$\bar{B}C$	4	5	7 6
BC	13	(1) 13	15 14
$B\bar{C}$	8	(1) 9	10 10

BCE

{ will not form Octet as there are more than one literal changes }

$\bar{A}\bar{D}\bar{E}$ $\bar{A}\bar{D}E$

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.

* Mc. Clusky Method (Tabulation - Method):

Step: ① Take the given ckt 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.

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
		8	1	0	0	0
1	2	3	0	0	1	1
		9	1	0	0	1
	3	7	0	1	1	1
		11	1	0	1	1
	4	15	1	1	1	1

(2)

Group

Minterm

Variables

A B C D

0

0, 1

0 0 0 -

0, 8

- 0 0 0

1

1, 3

0 0 - 1

1, 9

- 0 0 1

8, 9

1 0 0 -

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 - } BD

0, 8, 1, 9

- 0 0 - }

1

1, 3, 9, 11

- 0 - 1 } BD

1, 9, 3, 11

- 0 - 1 }

2

3, 7, 11, 15

- - 1 1 } CD

3, 11, 7, 15

- - 1 1 }

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

0 1, 3, 7 8 9 11 15

 $\bar{B}\bar{C}$ (X) X X (X) X X $\bar{B}D$ (X) X X X X

CD X (X) X (X)

 $\bar{B}\bar{C}$ (X) X X (X) X

*** Note:

- - If any pair (Lower-Order-grouping) is not coming under ^[Higher-Order-grouping] Quad-grouping, then those numbers won't be carried the ^{4^m} second table to next table.

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

- - Tabulation Method for DON'T CARE CONDⁿ:-

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

⇒ In final chart we should have:

	6	7	8	9
1				

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 $f = \sum M(0, 1, 4, 5, 9, 11, 13, 15, 16, 17, 25, 27, 28, 29, 31)$
 $\phi(20, 21, 22, 30)$

*** This Tabⁿ Method is playing an imp. role for the ckts 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.

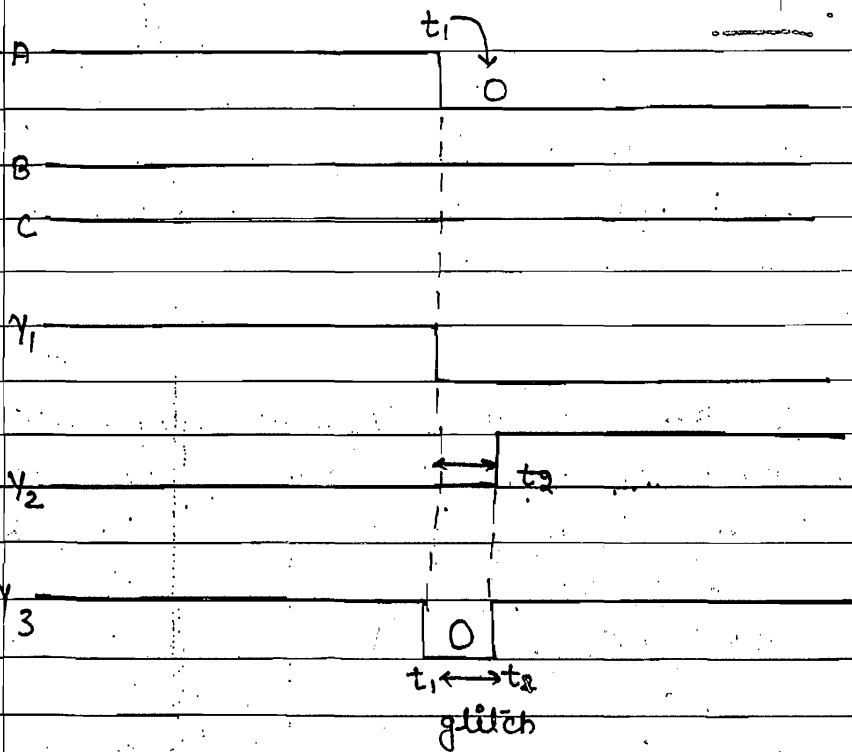
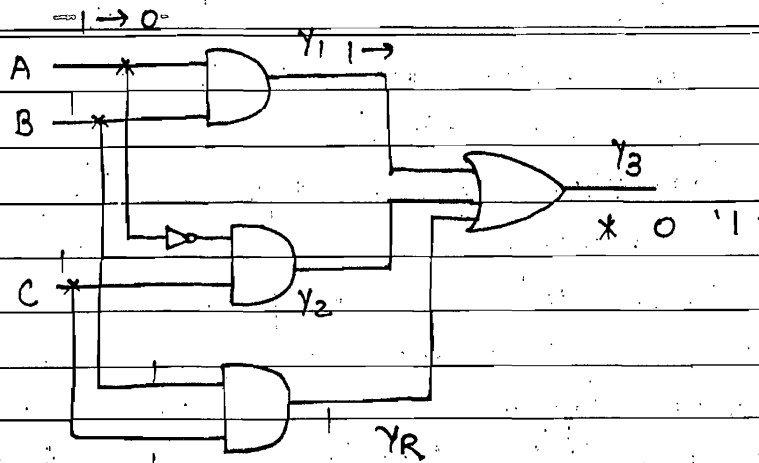
eg.. $f = \sum m(1, 3, 6, 7)$

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

$$f = \bar{A}C + AB \quad \text{--- } \times$$

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

Redundant group



Types:

- ① Static '1'
- ② Static '0'
- ③ Dynamic

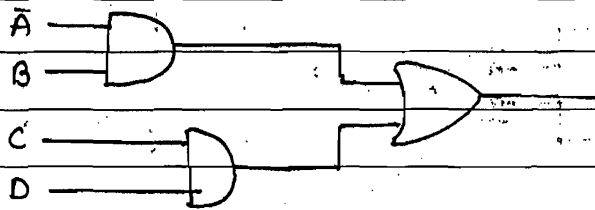
Implimentation of Boolean-functions by Using NAND, NOR - gates only :-

①

$$f = A'B + CD$$

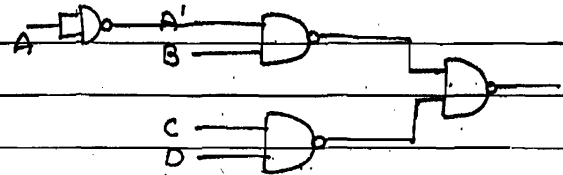
(NAND)

AND-OR



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

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

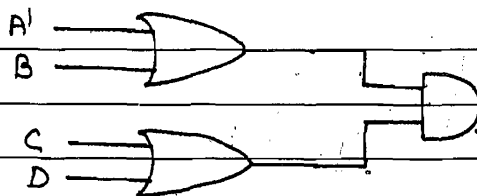


②

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

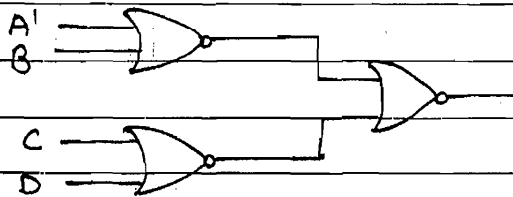
(NOR)

OR-AND



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

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

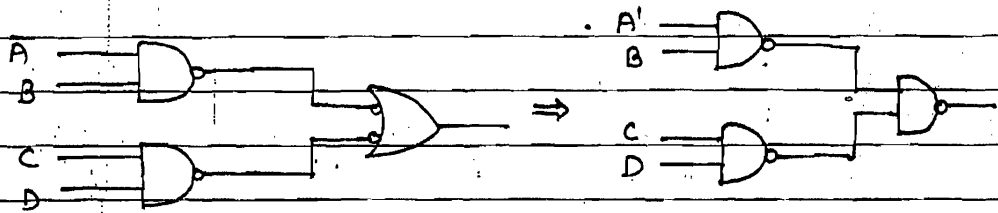


Method : ②

Step: ① AND (NAND)

Step: ② OR (Bubbled OR = NAND)

Step: ③ NOT (NAND)

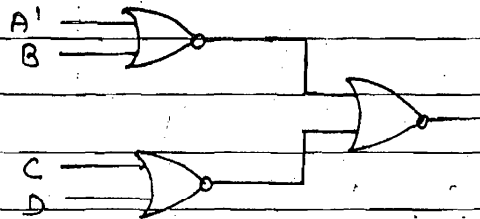


NOR :

Step : ① OR (NOR)

Step: ② AND (Bubbled AND = NOR)

Step: ③ NOT (NOR)



"Active - low":

SR-latch $\Rightarrow$

And-gate connect hai islea clk hmesha 'high'

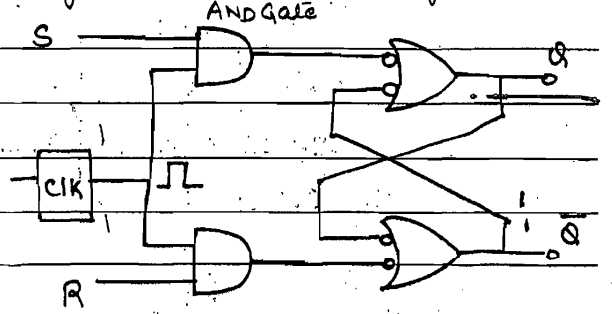
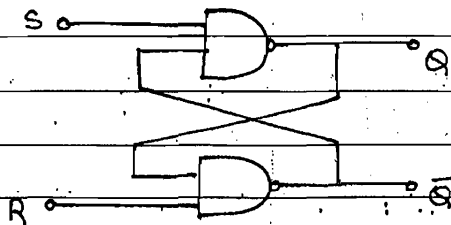
Gated-SR-latch $\Rightarrow$

hamega aur ise clk ko hmesha high ke rakht-

Clocked-SR-latch $\Rightarrow$

-enge q ki agar use ke low kam denge to karna

he sig high de, wo '0' he na jayga.

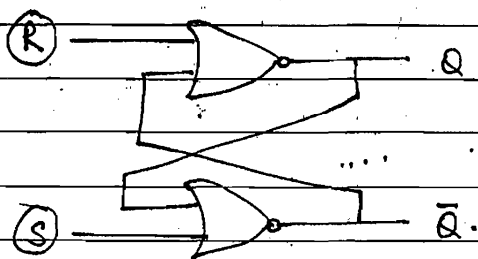


NAND $\Rightarrow$ Bubbled OR

Table for Active - low latch:-

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

Active - High latch :-



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

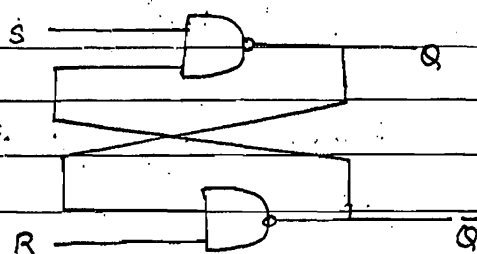
Sequential Circuits :-

Logic gates with feedback connection are known as sequential circuits.

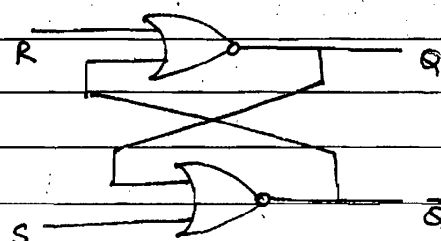
eg. Latches :-

Latches are two-types -

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



fig(a)

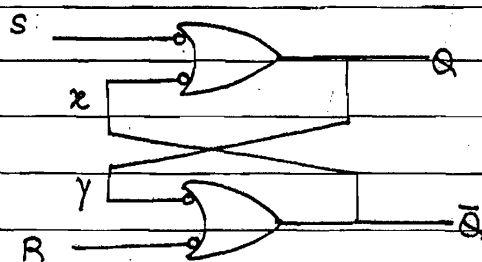


fig(b)

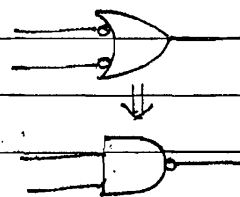
S → Set

R → Reset

SR-Latch operations (ACTIVE-LOW)



Bubbled OR = NAND



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 $S=1$, $R=0 \Rightarrow$

- $\rightarrow$ If initially $Q=0$, then the feedback S/P will be $x=1$ & $y=0$ and we get the next-state $Q^+ = 0$
- $\rightarrow$ If initially $Q=1$, Then the feedback S/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 S/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.

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

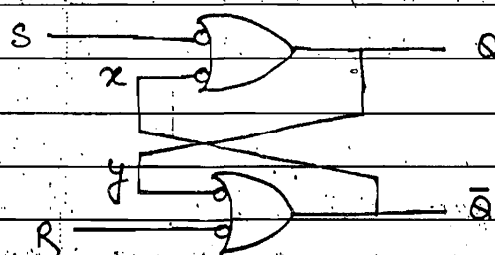
- °- Logic-gates with feedback connections behaves as sequential ckt's
e.g. Latches

°- 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 $S=1$, $R=1$

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

if initially ' $Q=1$ ', then the feedback I/P'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 $S=0, R=1$

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

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

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

- o- In the same above manner, the procedure should be conducted until we get the stable O/P. Feedback O/P'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.

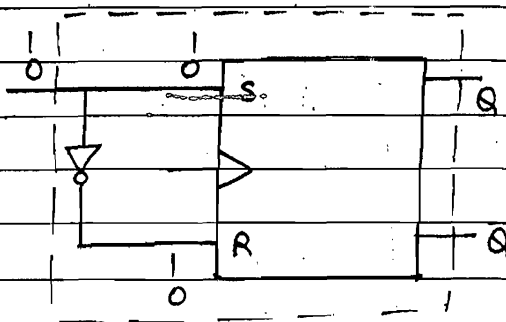
Case: 4 $S=0, R=0$

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.

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

** The Controlling can also be done by using clock-generator known as clocked S-R FF.

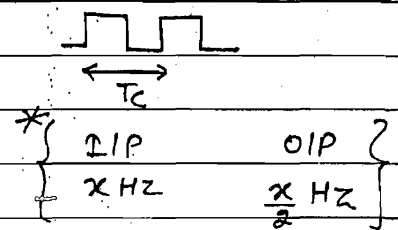
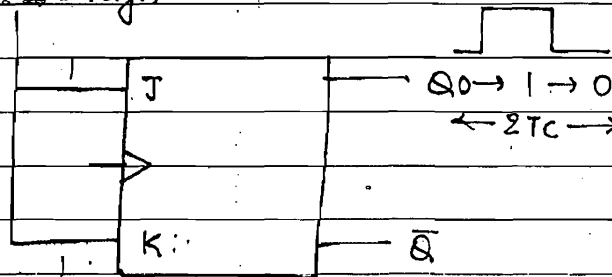
D-Flip-Flop :-



CLK	D	Q ⁺
	0	0
	1	1

2. T-Flip-Flop :

T = high



$$f = \frac{1}{T}$$

⇒ 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 ⁺
0	0	0	0 NC
0	0	1	0 Reset
0	1	0	1 set
0	1	1	1 $\bar{Q}$ Toggle
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

	J'K'	J'K	JK	JK'
$\bar{Q}$	0	1	1	1
Q	1	1	0	0

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

Characteristics of SR :

	S'R'	S'R	SR	SR'
$\bar{Q}$			X	1
Q	1		X	1

Q S R Q^+ 0 0 0 0 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

Chara. of D :- FF

	$\bar{D}$	D
$\bar{Q}$		1
Q		1

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

0 1 1

1 0 0

1 1 1

Char^r of T FF:-

	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 I/P's.

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

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

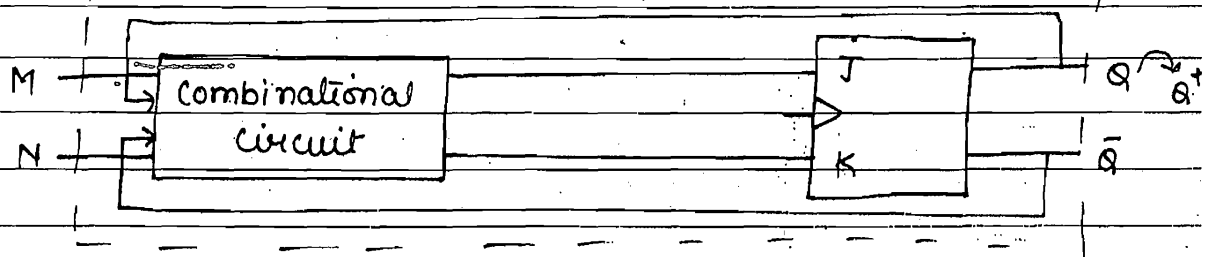
	Q	Q ⁺	J	K
0	0	0	0	X ↑
1	0	1	1	X ↑
2	1	0	X	1
3	1	1	X ↓	0

	Q	Q ⁺	S	R
0	0	0	0	X ↑
1	0	1	1	0
2	1	0	0	1
3	1	1	X ↓	0

	Q	Q ⁺	D
0	0	0	0
1	0	1	1
2	1	0	0
3	1	1	1

	Q	Q ⁺	T
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

Designing of F.F. ⇒ CONVERSION OF F.F. ⇒



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

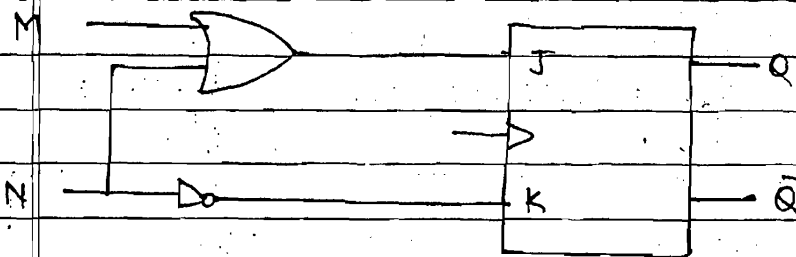
Q	M	N	Q ⁺	J	K
0	0	0	0	0	X
0	0	1	1	1	X
0	1	0	1	1	X
0	1	1	1	1	X
1	0	0	0	X	1
1	0	1	1	X	0
1	1	0	0	X	1
1	1	1	X	X	X

	M'N'	M'N	MN	MN'
Q	0	1	1	1
Q	X	X	X	X

$$J = M + N$$

	M'N'	M'N	MN	MN'
Q	X	X	X	X
Q	1		X	1

$$K = N'$$



Using S-R F.F. $\Rightarrow$

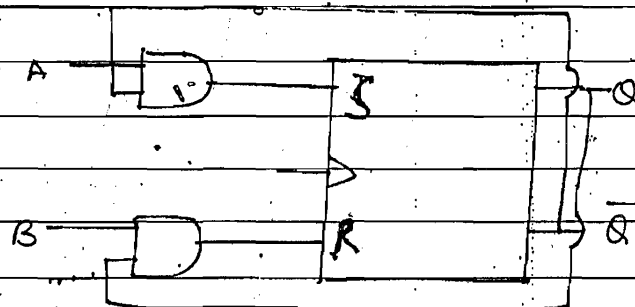
	A	B	Q ⁺	S	R	Q	Q ⁺	S	R
0	0	0	0	0	X	0	0	0	X
0	0	1	0	0	X	1	0	0	1
0	1	0	1	1	0	1	1	X	0
0	1	1	1	1	0				
1	0	0	1	X	0				
1	0	1	0	0	1				
1	1	0	1	X	0				
1	1	1	0	0	1				

	S'R'	S'R	SR	SR'
Q	0	X	1	1
Q	X	4	5	X

$$S = A Q' \quad / \quad S = J Q'$$

	S'R'	S'R	SR	SR'
Q	X	0	X	3
Q	4	1	5	6

$$R = B Q \quad / \quad R = K Q$$



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 JIP is known as Race-Around Problem.

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

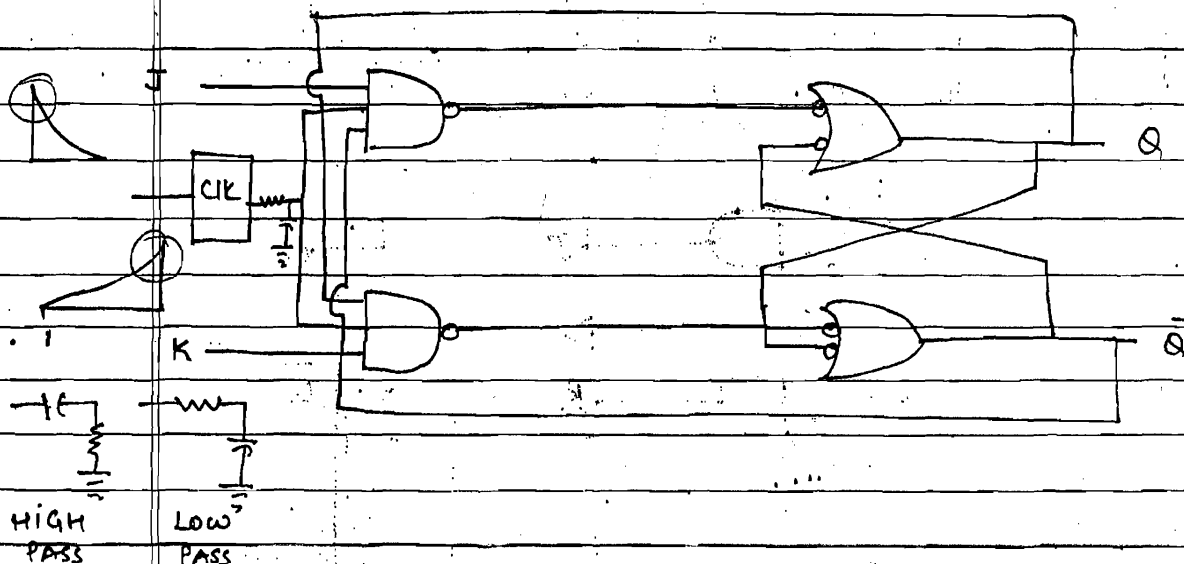
Methods to avoid Prob. $\Rightarrow$

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

Note: Edge-Trigg. is two types -

A. Rising Edge Trigg. [Positive Edge Trigg.]

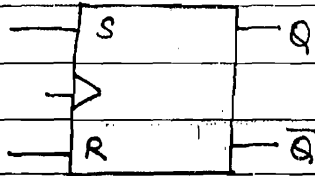
B. Falling Edge Trigg. [Negative Edge Trigg.]



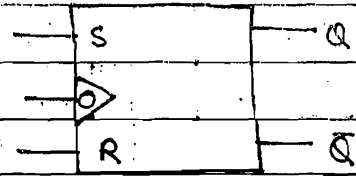
- * 3. By using Master slave Flip-Flop, Race-around problem can be avoided.

TIMING DIAGRAMS :

Negative Trigg.

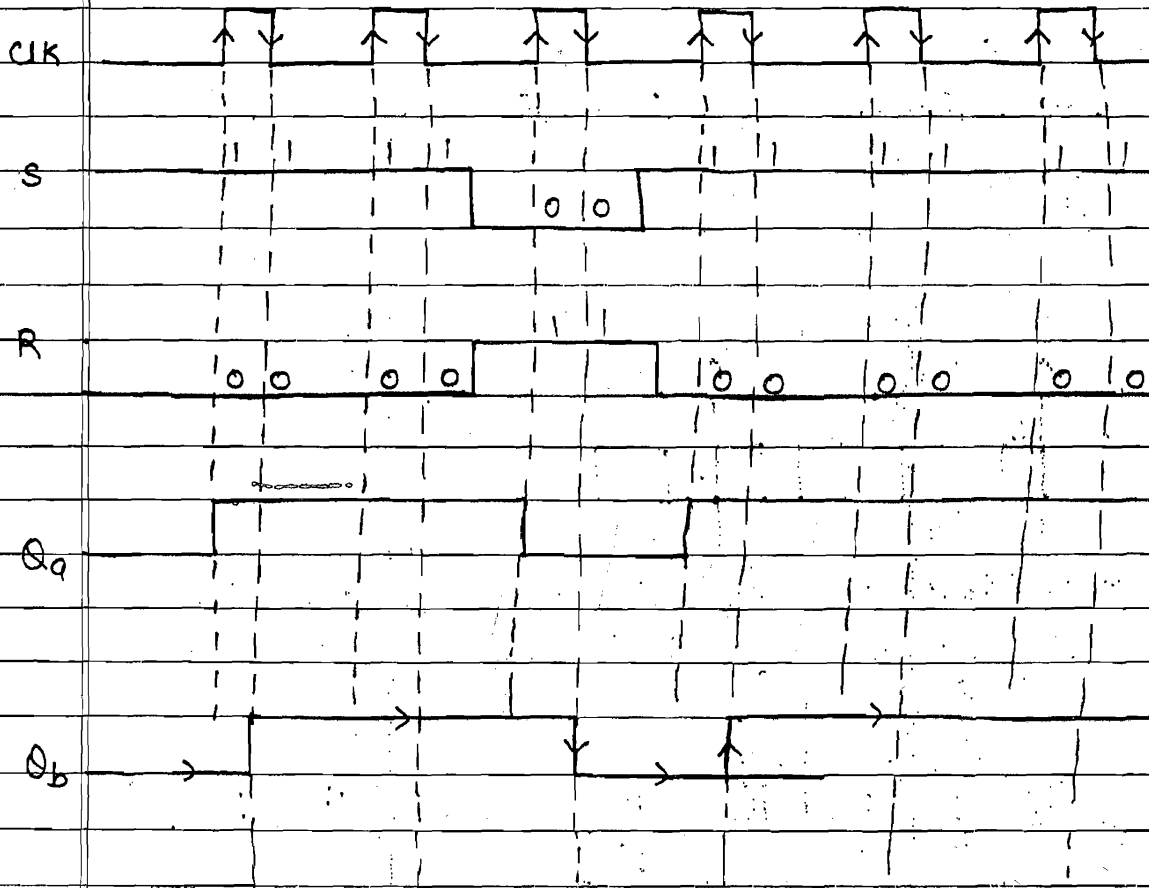


(a)

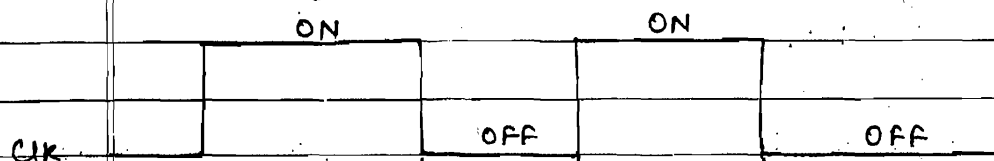


(b)

+ 've Triggered



Pulse-Trigg. $\Rightarrow$



clk

J

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

Ques: ①

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	1	D	$\bar{E}$	E
A		1	1	$\bar{D}$

Ques: ②

	$\bar{B}\bar{C}$	$\bar{B}C$	BC	$B\bar{C}$
$\bar{A}$	E	1	F	1
A		D		$\bar{F}$

$f = AC + ABD + BCE + \bar{A}BD + A\bar{B}\bar{C} + \bar{A}\bar{B}C$
 $+ \bar{A}BE + \bar{B}CD + B\bar{C}\bar{D} + \bar{B}\bar{C}C$
 $\bar{B}\bar{C} \quad \bar{B}C \quad BC \quad B\bar{C}$

classmate

53

Date

Page

Ans: 1.

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

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

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

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

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

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

$\Rightarrow \bar{E}BC$

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

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

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

$\Rightarrow \bar{D}AB$

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

Ans: 2

$\bar{A}$	E ₀	I ₁	F ₃	1 ₂
A	4	D ₅	7	$\bar{F}$ ₆

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

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

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

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

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

$\Rightarrow F\bar{A}C$

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

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

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

$\Rightarrow D\bar{B}C$

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

Sheet Problems

S R NC

Ques: 1 S Active-Low SR latch

0 1 0
1 0 0
0 0 X

CK :

S :

R

Q

Ques: 2

ACTIVE-H

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

S

R

Q

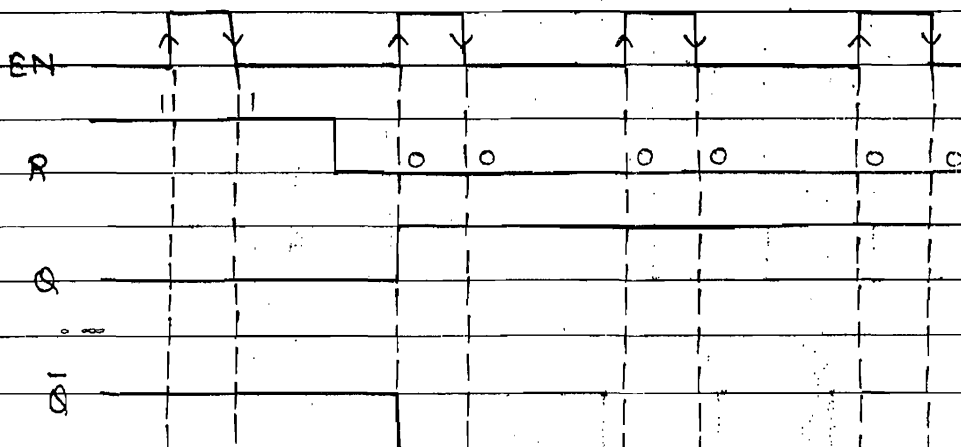
Ques: 3

S

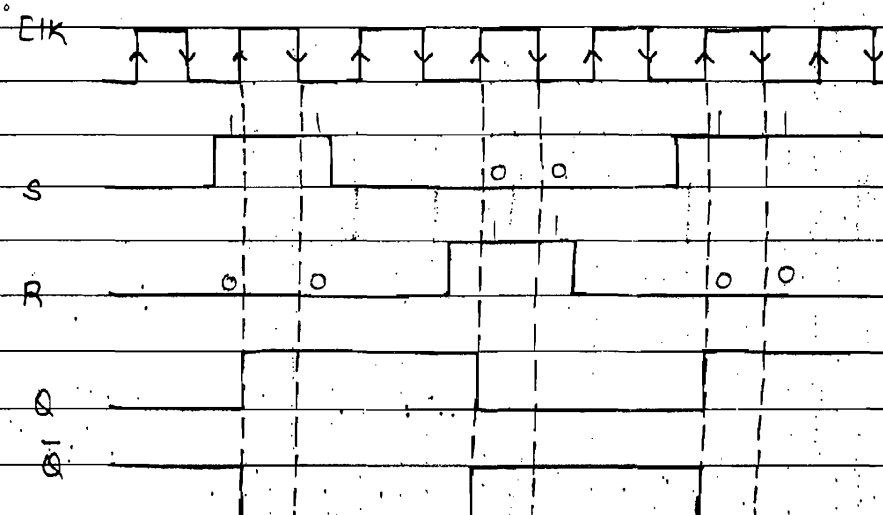
R

Q

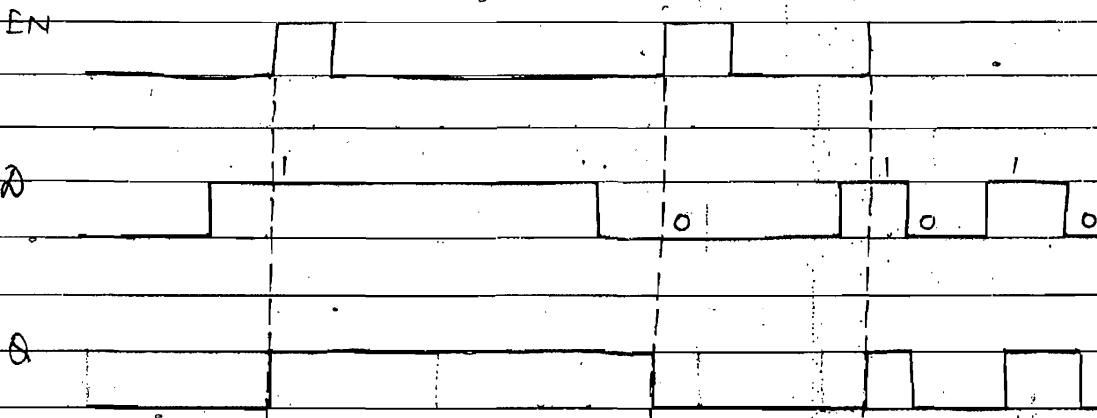
Que: 4



Que: 5



Que: 54



clk is active
as it is set as

Ques: 55

clk:

S

R

+ve trigg $\leftarrow Q_a$ -ve trigg $\leftarrow Q_b$

Ques: 56

clk

+ve Trigg

Q

57

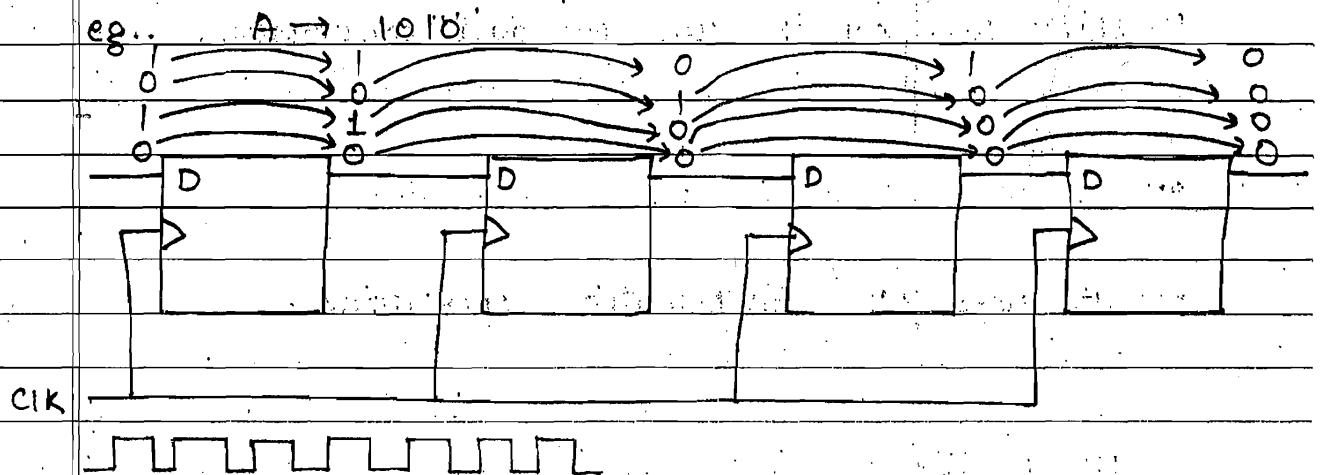
clk

D

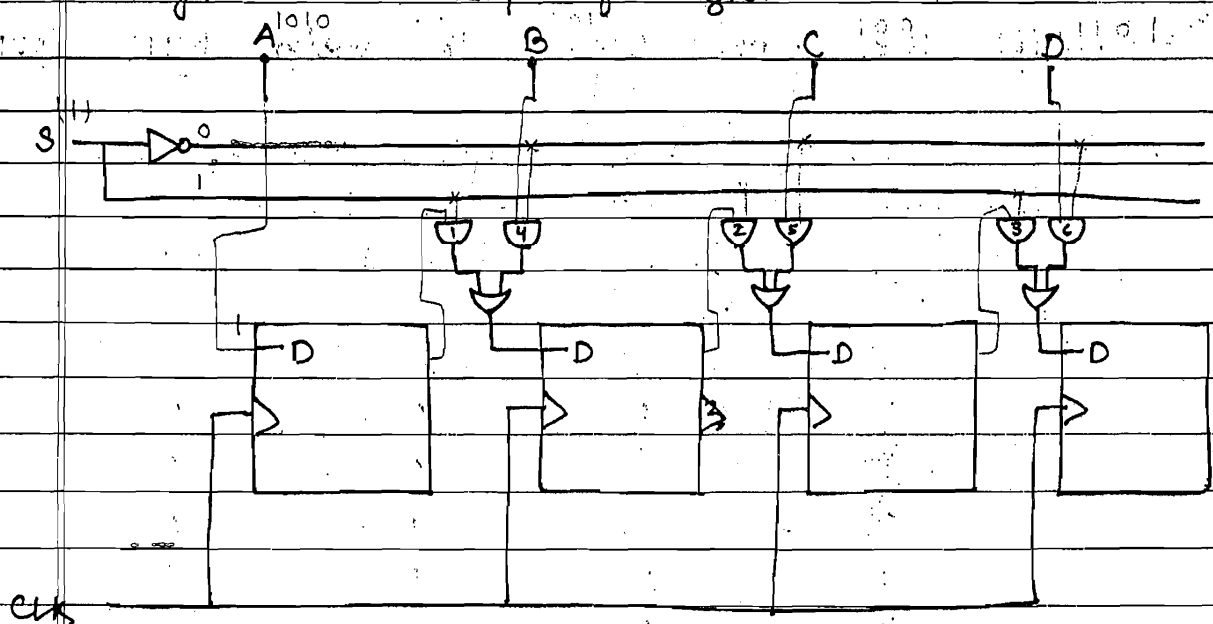
Q

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 0
- C). PISO $\Rightarrow$ 1 and $(n-1)$
- D). PIPO $\Rightarrow$ 1 and 0



Ques: Design 'PISO' binary shift register :



$S=0$

$S=1$

Parallel IO $\Rightarrow$
4, 5, 6 $\checkmark$

1, 2, 3 $\checkmark$ (operale)

* Application of Binary - shift - Register :

◦ - Temporary data storage

◦ - We can provide time delay : $\{ N T_c = \Delta t \}$

N : no. of f.f.

T_c : clock period

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

◦ - shift Res. can be use for data conversion. [SISO → ^{comm} sys.]
eg.. SISO converts serial form of data into Parallel form. [SIPO]

◦ - PISO → converts Parallel → Series.

◦ - It uses for Airthematic operations.

eg. Left shift Res. is multiplication by 2 network
Right shift Res. is division by 2 Network

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

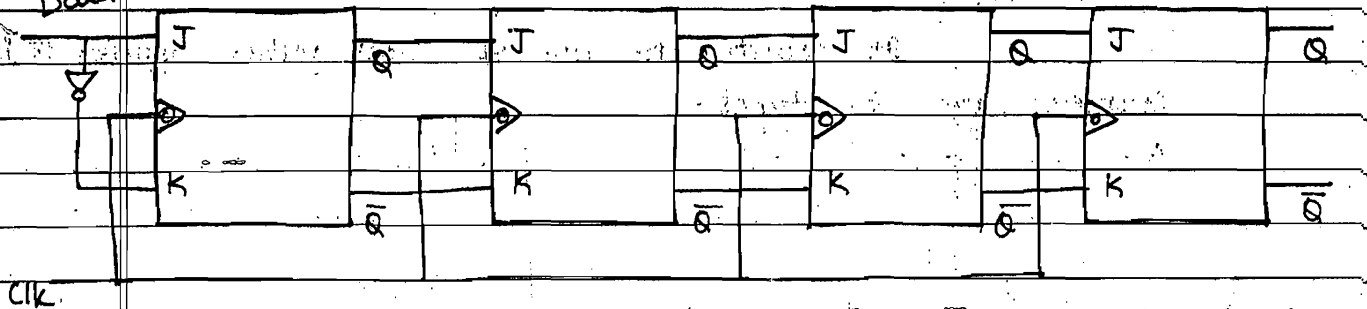
* ◦ - shift Regi. are used to design RING - COUNTER

* QUES

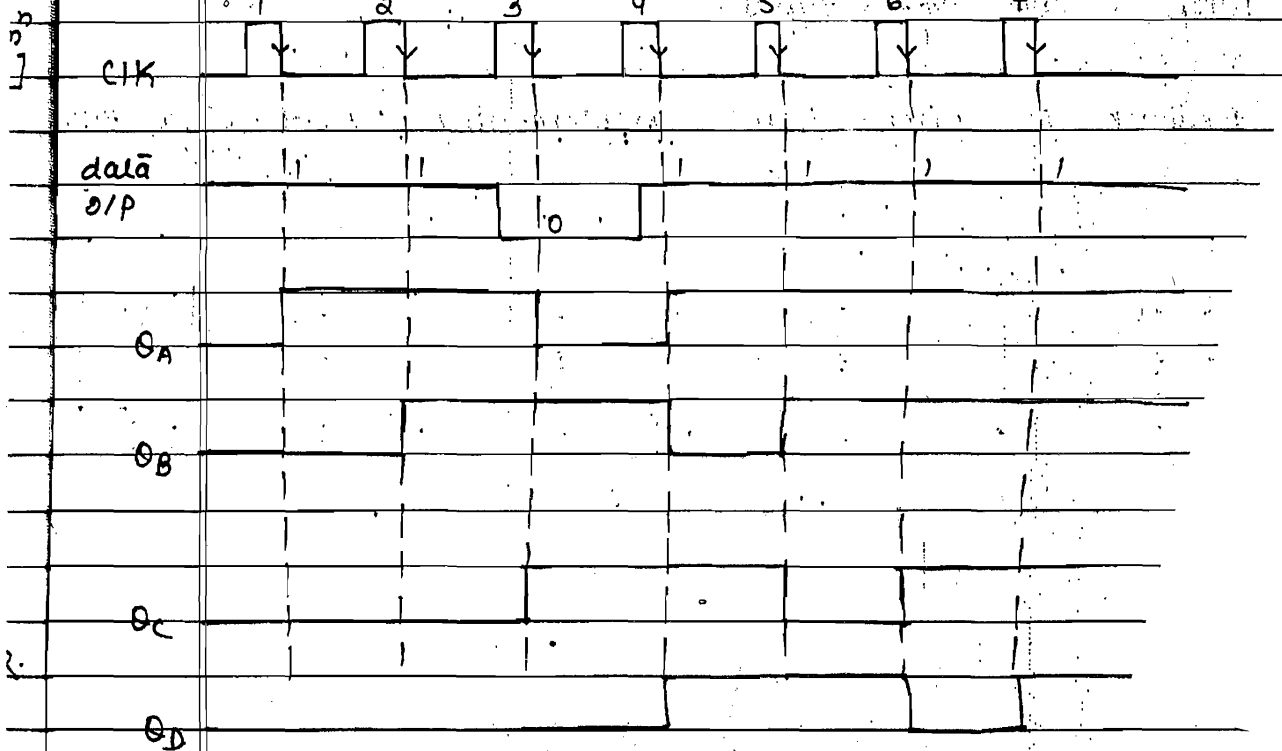
Ques: Design 4-Bit BINARY - SHIFT - REGI. By using JK . F.F. with -ve edge Trigg. and draw the wave form diagrams for all F.F. O/P's for given data I/P.

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 I/P

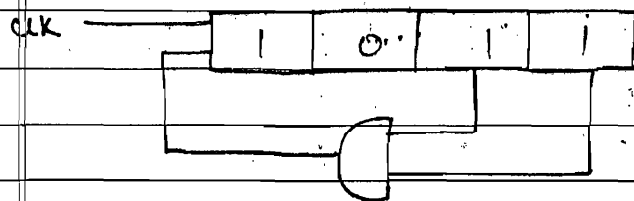


CLK

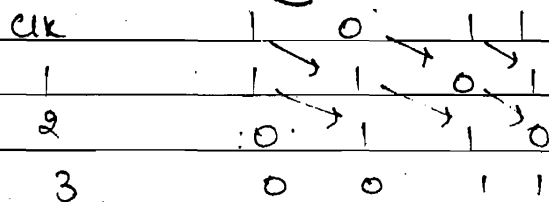


Bi-directional

Ques: Design a 4-Bit Binary Shift Register.



After 3-pulses?



COUNTERS :

It counts the no. of clock pulses applied to it.
counters are 2- types-

A). Synchronous Counter
(Parallel - counters)

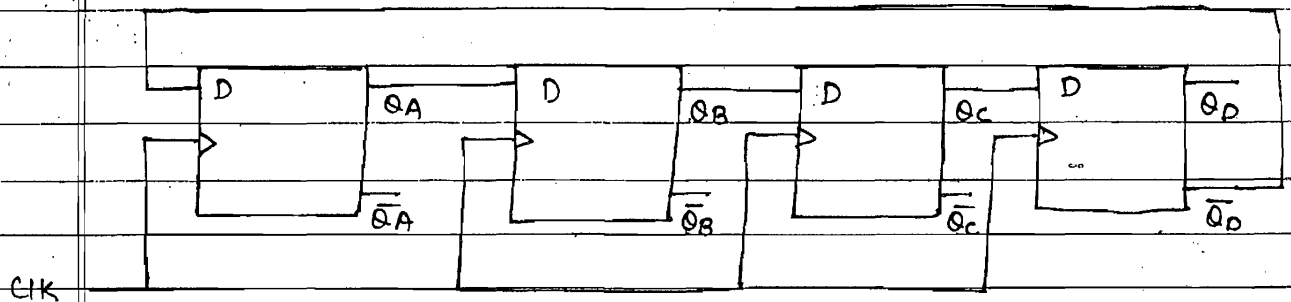
B). Asynchronous Counters
(Ripple - counters)

⇒ 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 complementry O/P of final F.F.

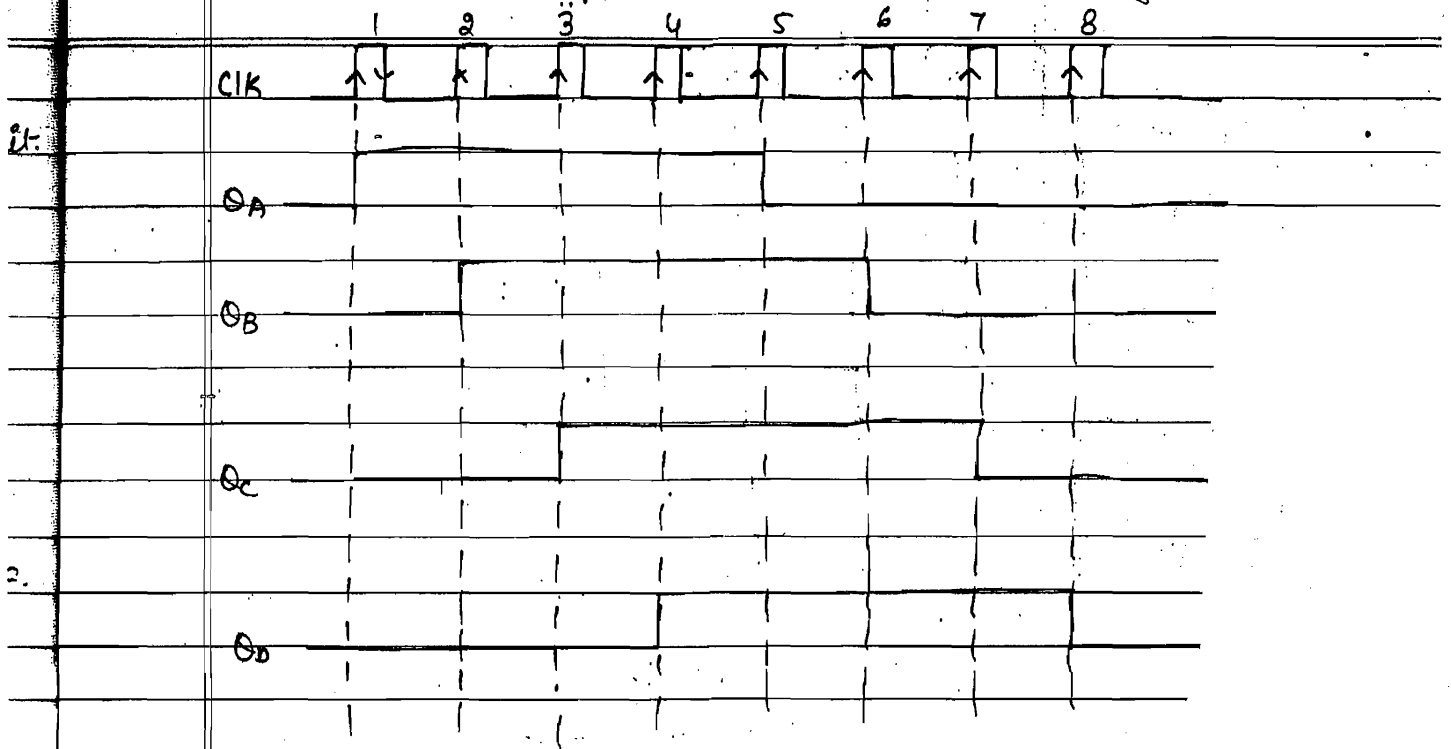
2. Ordinary Ring C.



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
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

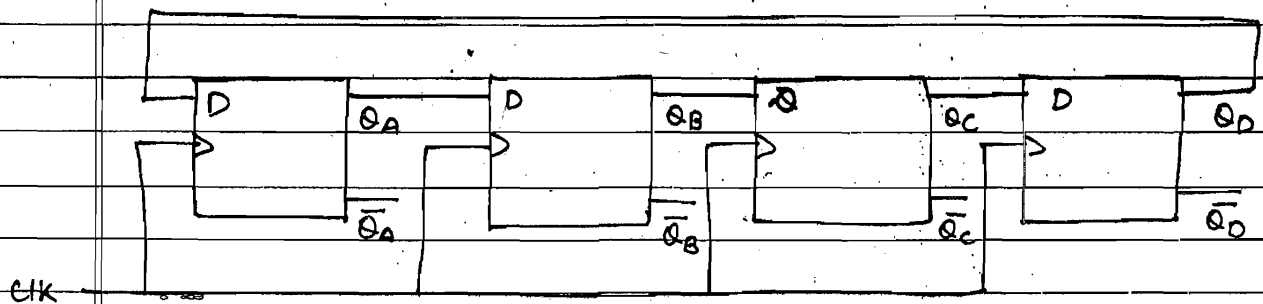
mod 8

X

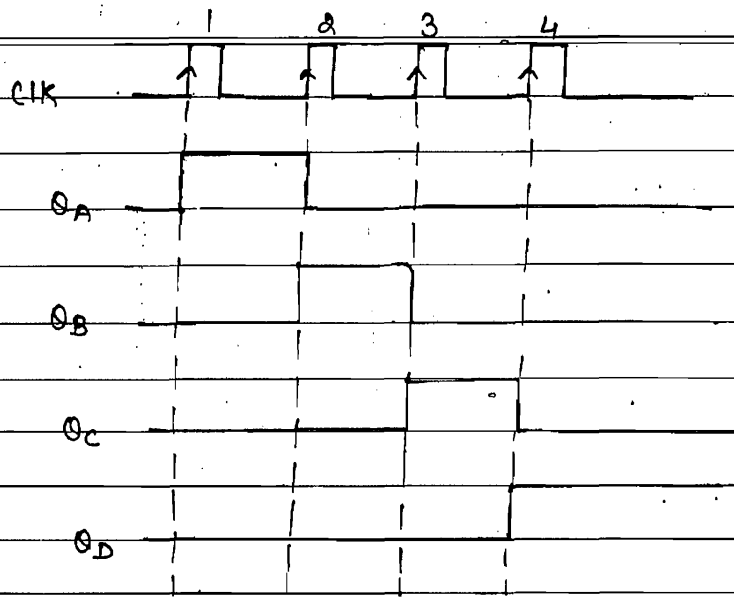


$n \rightarrow 2^n$ — Possible states.
clk pulses

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



CLK	QA	QB	QC	QD	
1	1	0	0	0	} mod 4
2	0	1	0	0	
3	0	0	1	0	
4	0	0	0	1	
5	1	0	0	0	X



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

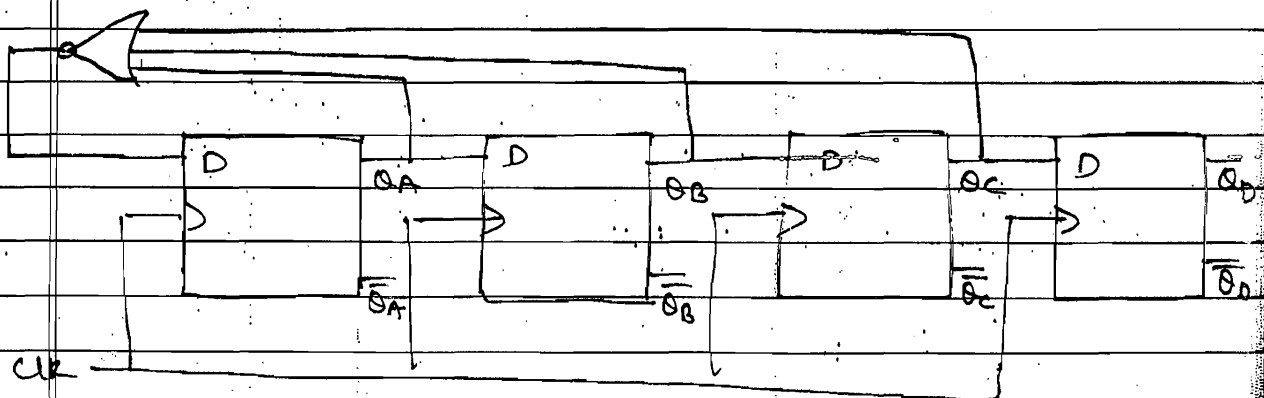
$\Rightarrow$ State identification (Decoding) is easy in RING-C.

* Unused states :

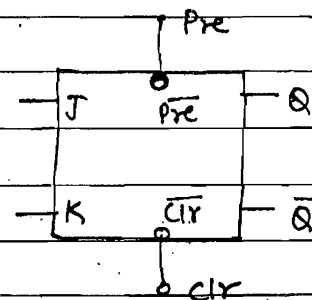
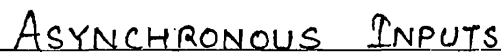
Johnson's C $\Rightarrow 2^n - 2n$

RING C $\Rightarrow 2^n - n$

* SELF-STARTED RING-C $\Rightarrow$

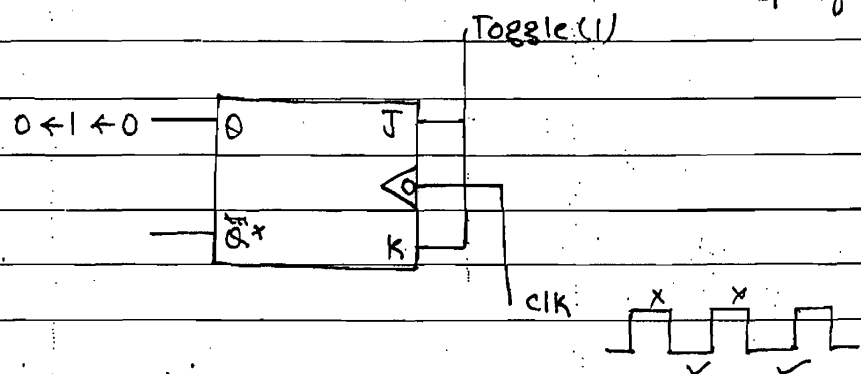


c. Switch Tail C.



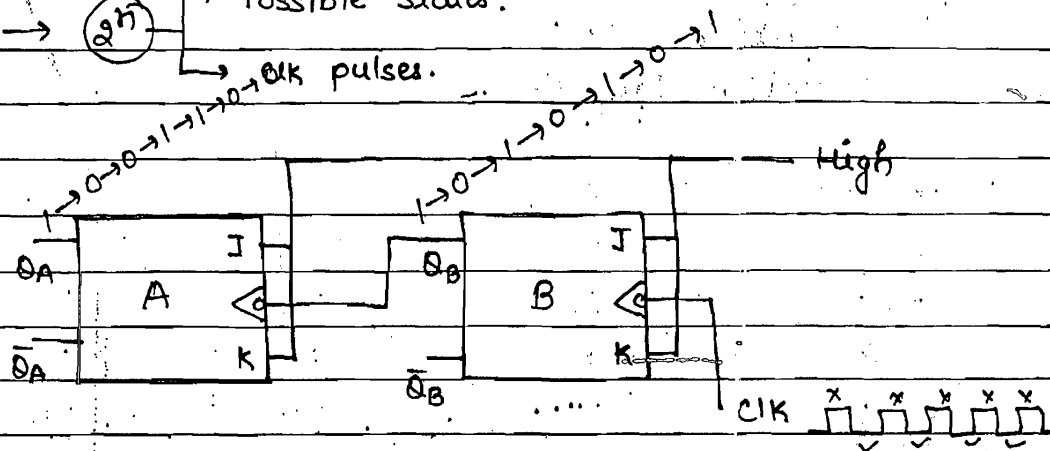
$\overline{Pre}$	$\overline{clr}$	Q^+
1	1	ff
1	0	0
0	1	1
0	0	x Don't Care unspecified oper.

-ive things :



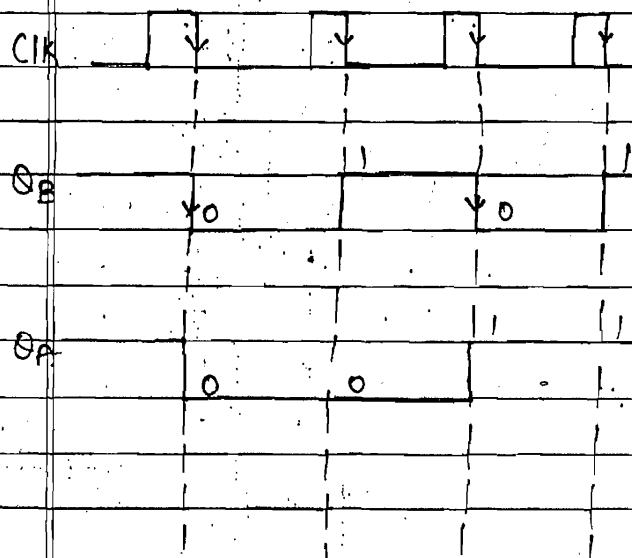
Asyn. C. $\Rightarrow$ F.F. are not triggered simultaneously.

$n \rightarrow$ Q_n Possible states.
 $\rightarrow$ Q_k pulses.

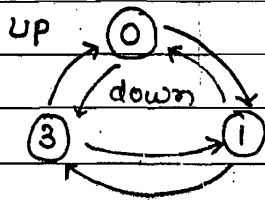


CLK	QA	QB
0	0	0
1	0	1
2	1	0
3	1	1
4	0	0

mod '4'



Synchronous up-down Count - design :-



up d'

0

1

3

y=0
up

Present state		Next-state					
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
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$\bar{Q}_1, \bar{Q}_0$	$\bar{Q}_1, Q_0$	Q_1, Q_0	$Q_1, \bar{Q}_0$
1 0	X 2	X 3	X 2
1 1	X 5	X 7	X 6

$$J_0 = 1$$

y=1
down

Present-state		Next-state					
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

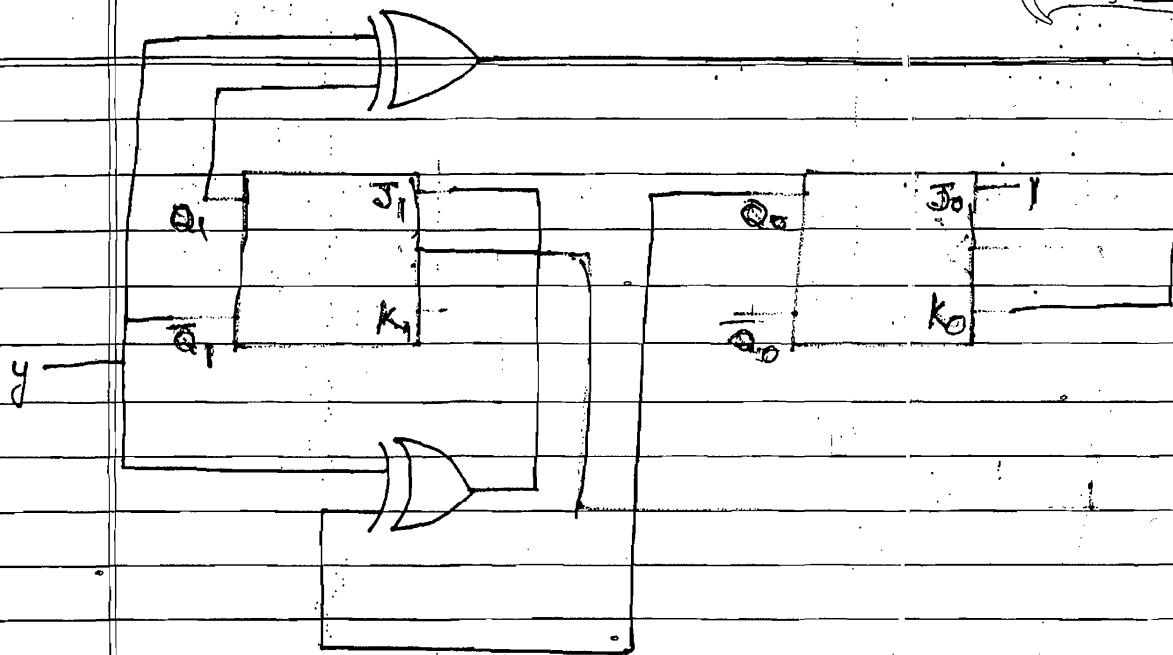
Q	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

$\bar{Q}_1, \bar{Q}_0$	$\bar{Q}_1, Q_0$	Q_1, Q_0	$Q_1, \bar{Q}_0$
X	0	1	X
X	1	0	X

$$y=1$$

$$K_0 = yQ_1 + y\bar{Q}_1$$

$$y \oplus Q_1$$



$$J_1 = y \oplus Q_0$$

$$K_1 = 1$$

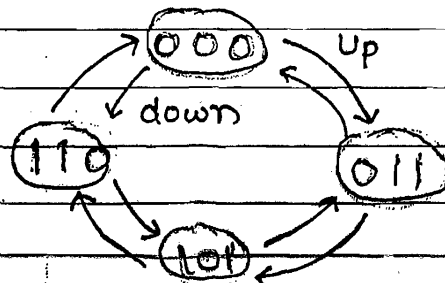
Ques: wanted Sequence By using T-flip-flop

0

3

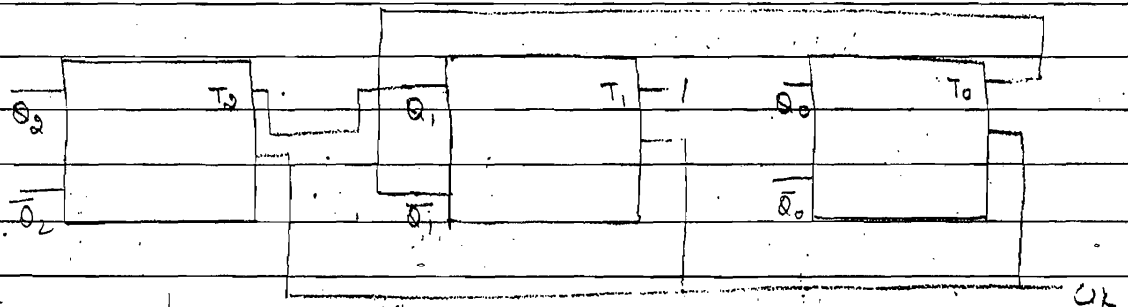
5

6



Present-State			Next-State			T_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

		$\bar{Q}_1, \bar{Q}_0$	$\bar{Q}_1, Q_0$	$Q_1, \bar{Q}_0$	Q_1, Q_0
$T_0 = \bar{Q}_1$	$T_0 = \bar{Q}_1, \bar{y}$	1	x	0	x
$T_1 = 1$	y	x	1	x	0
$T_2 = Q_1$	$T_2 = Q_1, \bar{y}$	0	x	1	x
	$T_2 = Q_1, y$	x	0	x	1



Procedure: corrected self-started counter: Bcoz of vig 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.

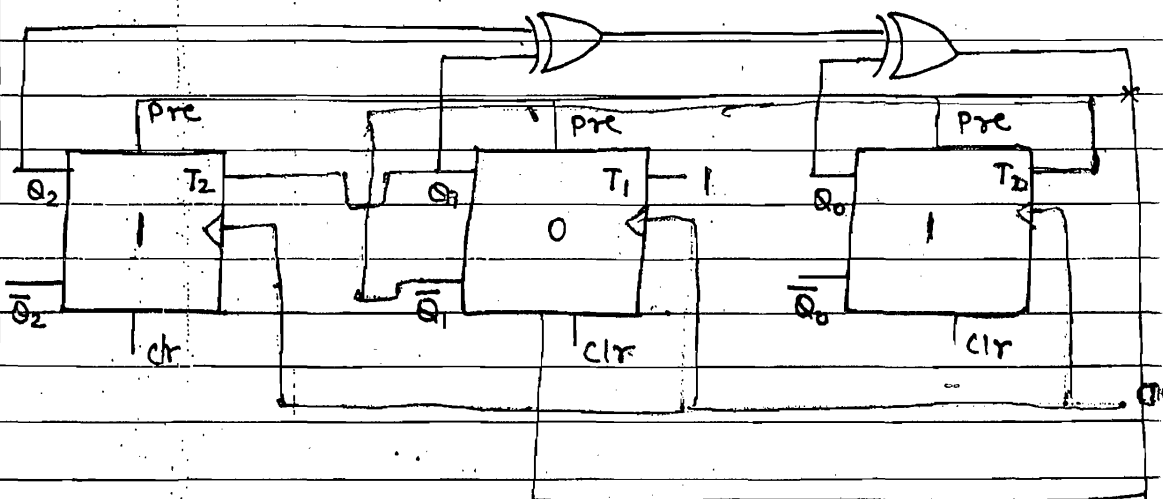
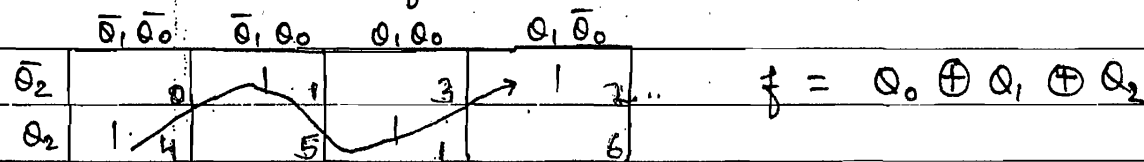
lock-out condition: if the counter is not coming back from unwanted seq. to wanted seq. it is known as lock-out condition.

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 achieved without any loss of clock-pulses.

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

$\rightarrow$ it is not self-corrected.



* 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.

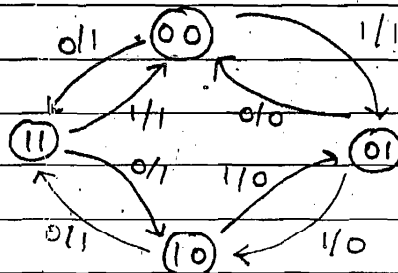
Step:3 Identify the equivalent states.

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.

⇒



x/y

$x \Rightarrow \text{I/P}$

$y \Rightarrow \text{O/P}$

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

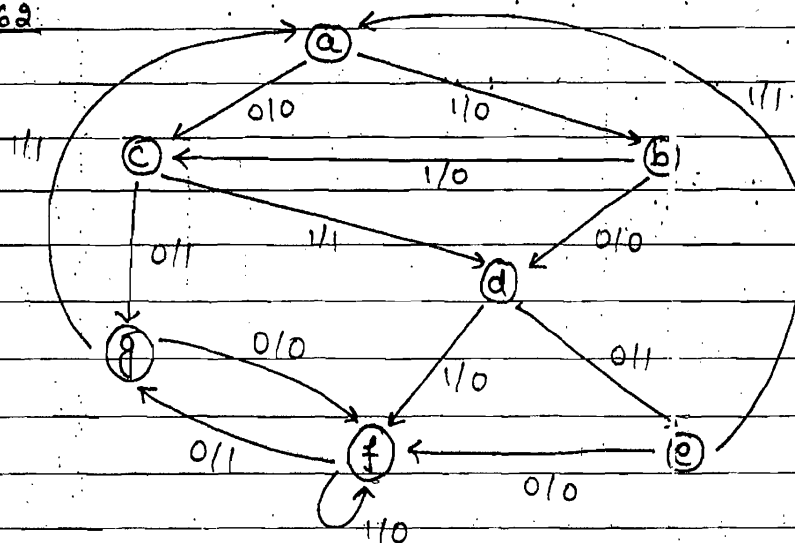
1 0 0 0

1

1

P.N. 62

Ques: 32



$d=f$

$e=g$

Present state

Next-state

Output

$x=0$

$x=1$

$x=0$

$x=1$

a

c

b

0

0

b

d

c

0

0

c

~~f~~e

~~d~~

1

1

d

e

~~d~~

1

0

e

~~f~~d

a

0

1

~~f~~

~~g~~e

f

1

0

~~g~~

f

a

0

1

$\Rightarrow$ 5-states

Ques: 33

$x=0$

$x=1$

$x=0$

$x=1$

a

c

b

d

c

g

d

e

e

f

f

g

g

f

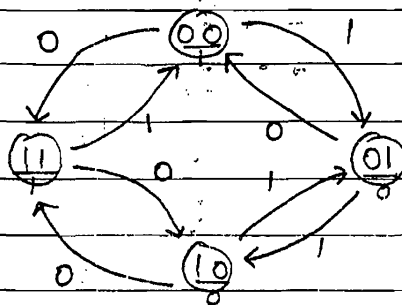
State - deduction methods are 2-types :

1. Mealy method:
2. Moore method.

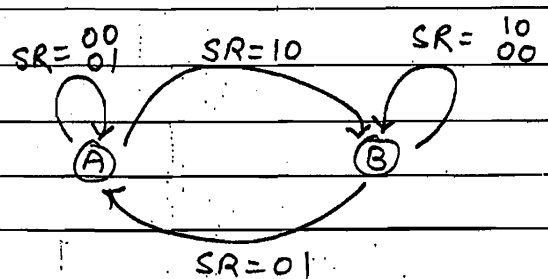
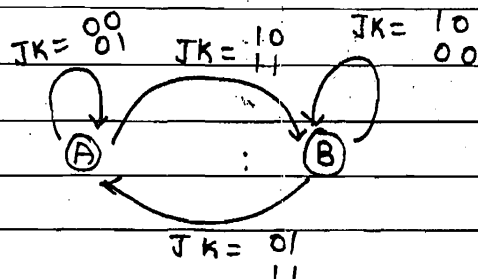
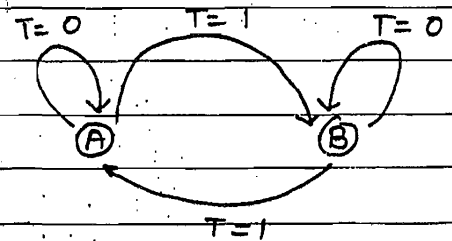
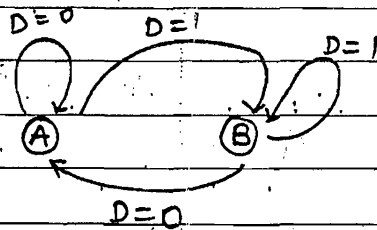
1. Mealy - method : In this method present O/P depends on present I/P and present - state.

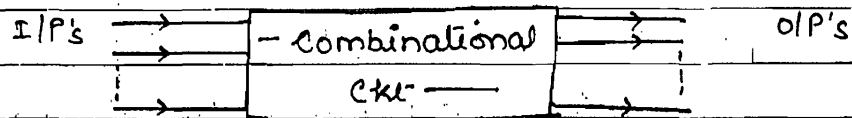
Note: above discussed examples are Mealy method only.

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



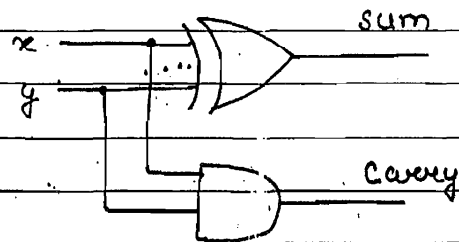
State diagram for J.K. $\Rightarrow$



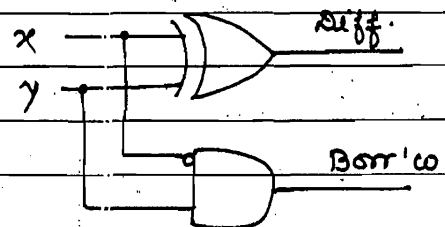
* Combinational Circuit :-

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

H.A.



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



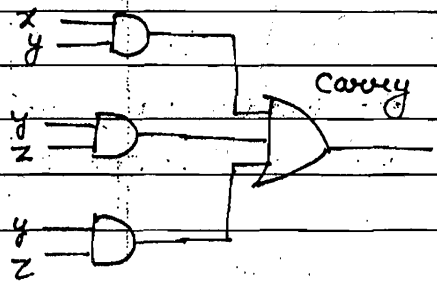
F.A.

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

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

$\bar{x}$	0	1	1	0
x	4	5	7	6

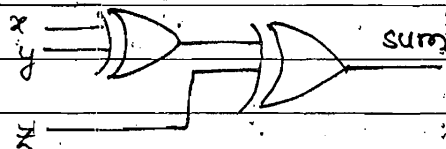
$$\text{Carry} = xy + yz + yz$$



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

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

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



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

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

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

	1	1	1
	1	1	1

$$\begin{aligned} \text{Borrow} &= \bar{x}y + \bar{y}z + \bar{x}z \\ &= \bar{x}y + yz + \bar{x}z \end{aligned}$$

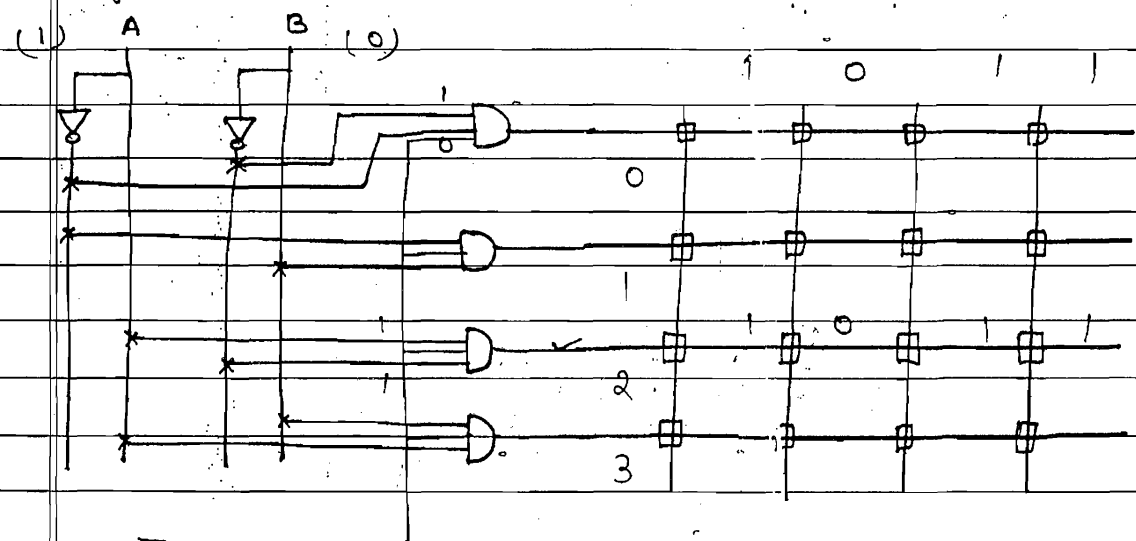
Decoders :

P.N. 59

Ques: 32

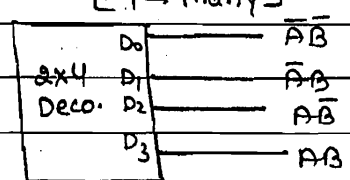
A	I_0	D_0
B	I_1	D_1
C	I_2	D_2
		D_3
		D_4
		D_5
		D_6
		D_7

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



Enable '1'
MUX
1 → many

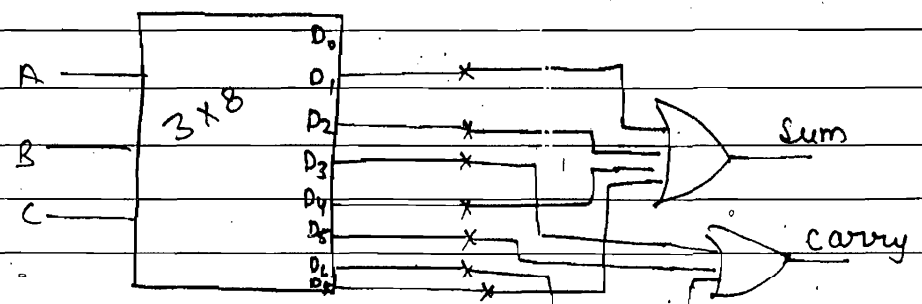
$$n \rightarrow 2^n$$



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

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

F.A.



H.A.

A B C Sum

0 0 0 0

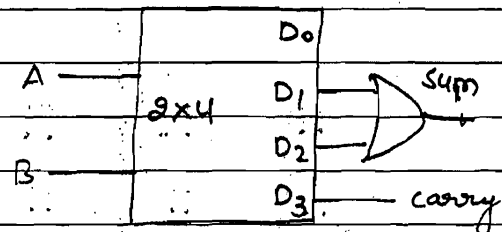
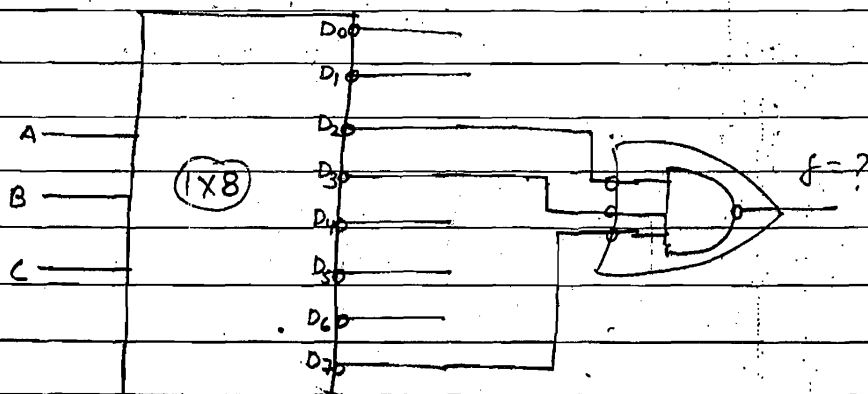
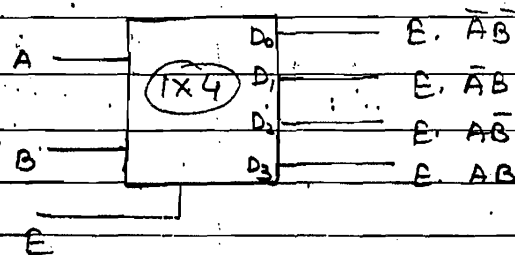
1 0 1 1

2 1 0 1

3 1 1 0

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

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

Demux :-

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

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

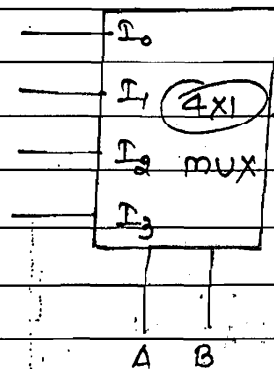
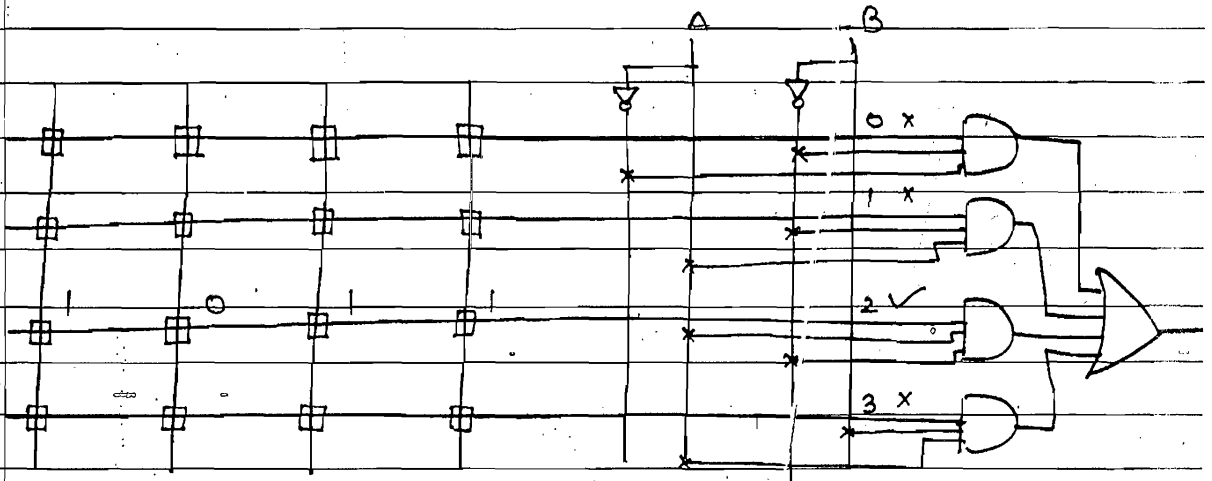
$$x = \bar{A}B + BC$$

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

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

Multiplexer : • Many to One Circuit .

- Data Selector .
- Parallel to Serial Converter .
- waveform Generator .

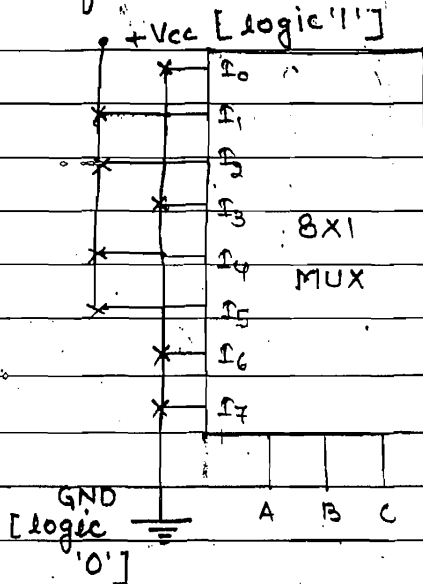


$$f = \bar{A}\bar{B}(I_0) + \bar{A}B(I_1) + A\bar{B}(I_2) + AB(I_3)$$

Designing the combinational ckt by MUXs :-

I:

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



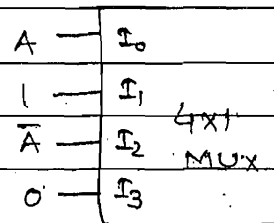
$$f = \sum m(1, 2, 4, 5)$$

II MSB:

$$f(A, B, C)$$

$\downarrow$ MSB $\downarrow$ LSB

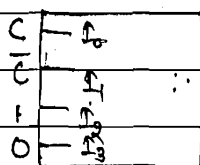
	A	B	C
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1



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

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

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



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

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

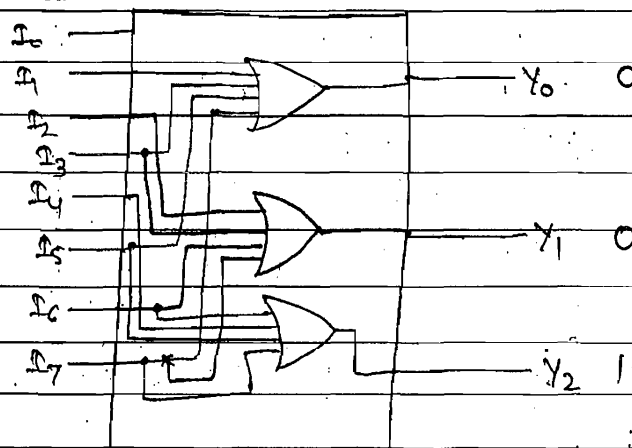
Encoder:

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

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

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

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



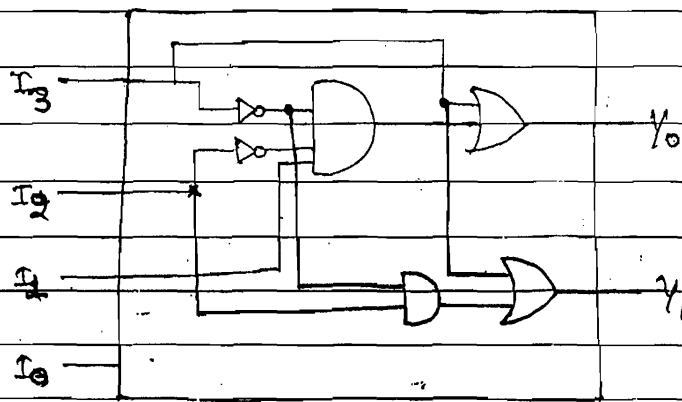
8x3 (Decimal to Binary Encoder.)

Priority Encoder :

I_3	I_2	I_1	I_0	Y_1	Y_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

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

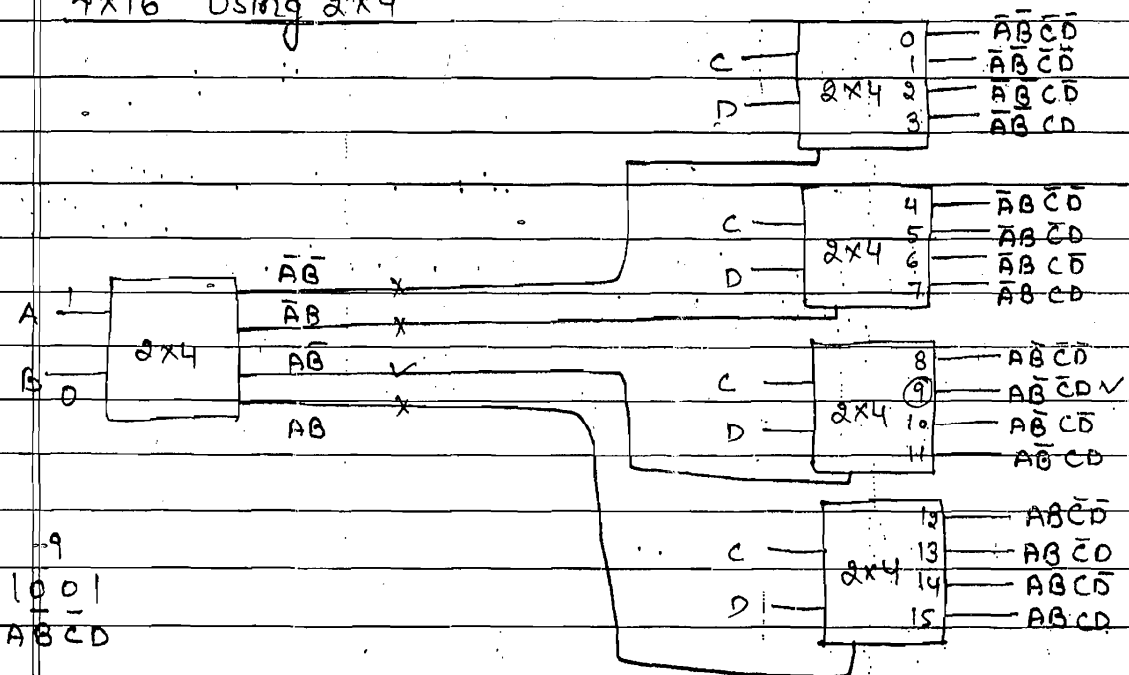
$$Y_1 = I_2 \cdot I_3' + I_3$$



4x2 MUX

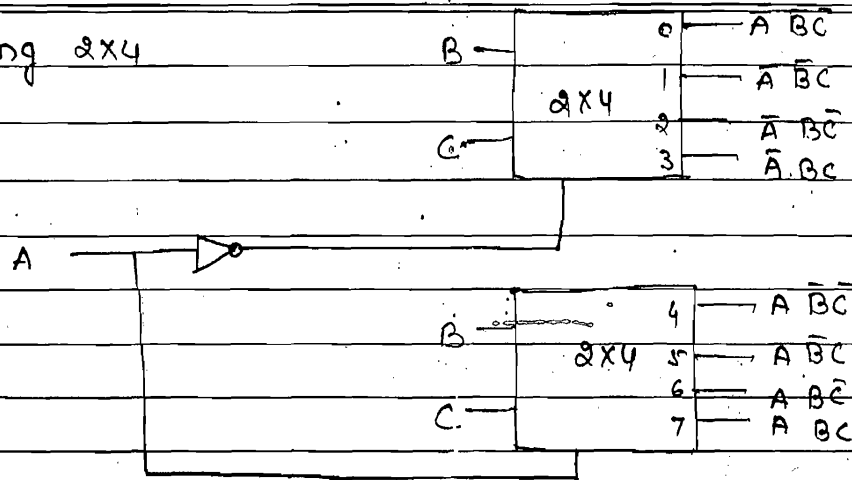
* Higher ordered circuits by using lower ordered circuits

4x16 using 2x4



② 3x8 using 2x4

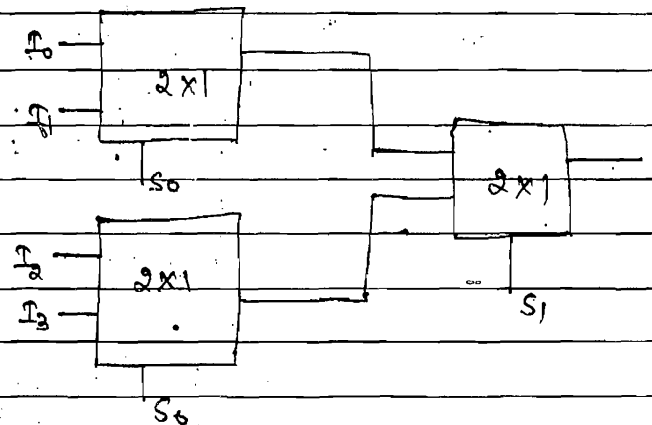
$$\frac{8}{4} = 2$$



③ # 4x1 MUX By using 2x1 + ...

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

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



Short-cut:

① 4x16 using 2x4

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

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

① → should fully divisible

⑤ → so we need ⑤, 2x4 Demux's

② 3x8 using 2x4

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

↓ not divisible

we have an odd no. so we'll have and NOT gate here and that will not complete divisi - like 160

Points:

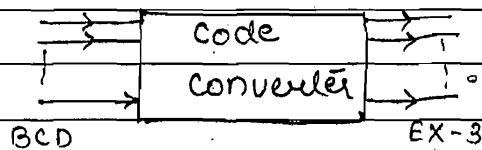
For eg: ①:

no. of decoders = ⑤

no. of enable decoders = ④.

NUMBER - SYSTEM:

Codes and Codes Converter:



BCD (8 4 2 1)

7/5/9/3/6

0111 / 0101 / 1001 / 0011 / 0110 BCD

8 4 2 1

0 1 1 1 ← 7

0 1 0 1 ← 5

1 0 0 1 ← 9

10 → 1010

→ invalid in BCD

11 → 1011

these are Binary - Codes.

12 → 1100

13 → 1101

14 → 1110

15 → 1111

→ 1/0

0001 0000 ← BCD

→ 1/1

0001 0001 ← BCD

→ 1/4

0001 0100 ← BCD

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

2 4 2 1
5 → 0 1 0 1 X

5 → 1 0 1 1 V

⇒ add '3' in BCD will get Ex-3

0 → 0000

+ 11 ← '3'

0011 ← Ex-3 of '0'

1 → 0001

+ 11 ← '3'

0100 ← Ex-3 of '1'

Ex-3

0 → 0011

1 → 0100

1

1

1

1

1

8 → 1011

9 → 1100

complement → 0100

→ 0011

self-complement



BCD to Ex-3 conversion :-

	BCD I/P				EX-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	0	1	1	$\bar{A}\bar{B}$	0	1	3	2		
1	0	0	0	1	0	1	0	0	$\bar{A}B$	4	1	5	7	6	
2	0	0	1	0	0	1	0	1	AB	X ₁₂	X ₁₃	X ₁₅	X ₁₄		
3	0	0	1	1	0	1	1	0	$A\bar{B}$	1	8	1	9	X ₁₁	X ₁₀
4	0	1	0	0	0	1	1	1							
5	0	1	0	1	1	0	0	0							
6	0	1	1	0	1	0	0	1							
7	0	1	1	1	1	0	1	0							
8	1	0	0	0	1	0	1	1							
9	1	0	0	1	1	1	0	0							
	1	0	1	0											

$W = A + BC + BD$

$$W = A + BC + BD$$

I/P's

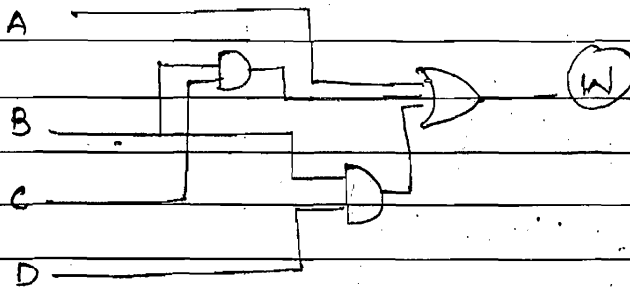
BCD

O/P's

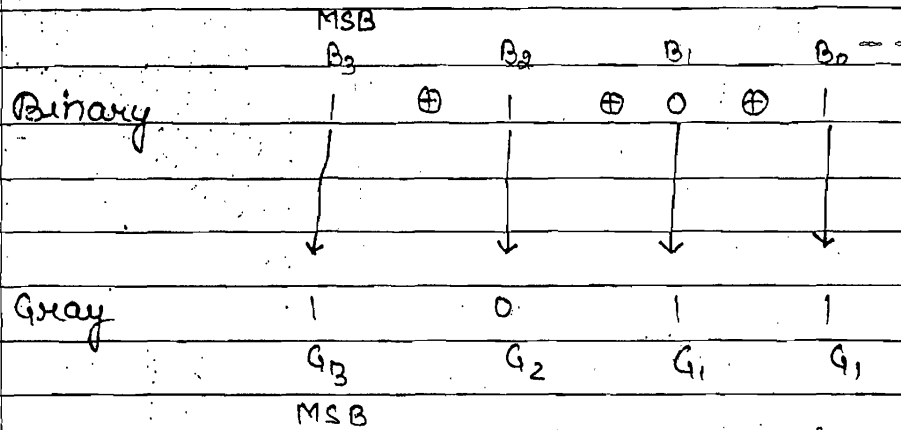
EX-3

HW: Make for

X, Y, Z as well.



Binary to Gray Conversion

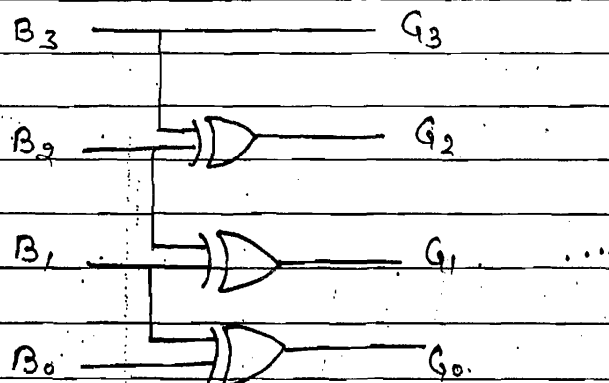


$$G_3 = B_3$$

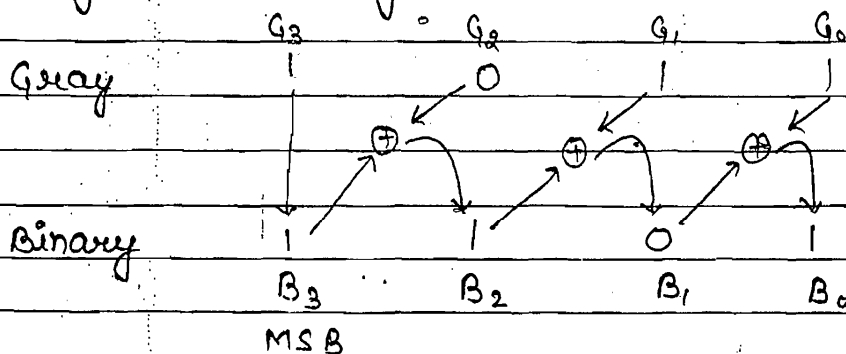
$$G_2 = B_3 \oplus B_2$$

$$G_1 = B_2 \oplus B_1$$

$$G_0 = B_1 \oplus B_0$$



* Gray to Binary Conversion

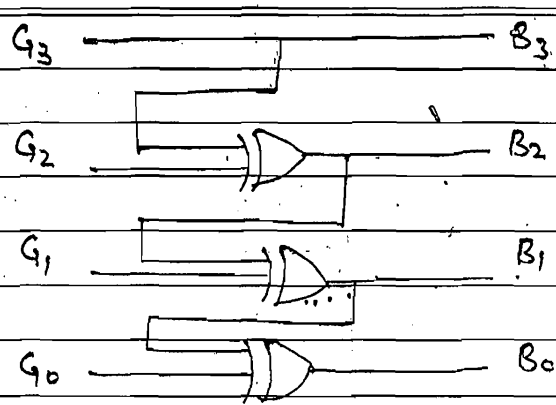


$$B_3 = G_3$$

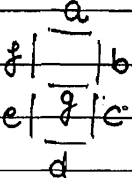
$$B_2 = B_2 \oplus G_2$$

$$B_1 = B_2 \oplus G_1$$

$$B_0 = B_1 \oplus G_0$$

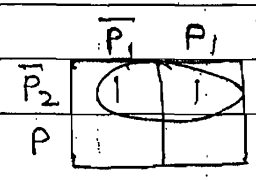


SEVEN SEGMENT :-

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

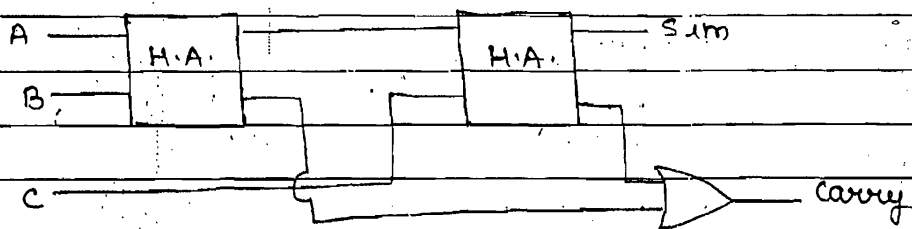
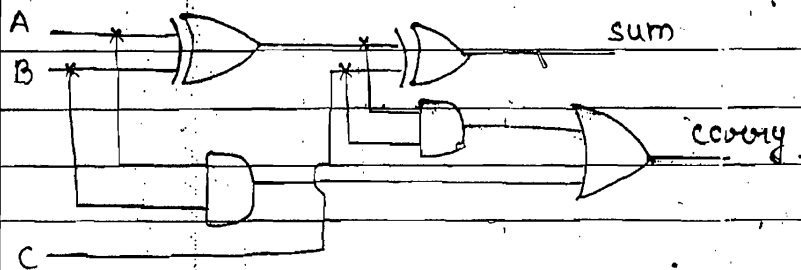
H.W. ⇒ 34, 35 P.N. 60

P ₂	P ₁	a	b	c	d	e	f	g	
0	0	1	1	1	1	1	1	0	'0'
0	1	1	1	0	1	1	0	1	'2'
1	0	1	0	1	1	0	1	1	'5'
1	1	1	0	0	1	1	1	1	'E'

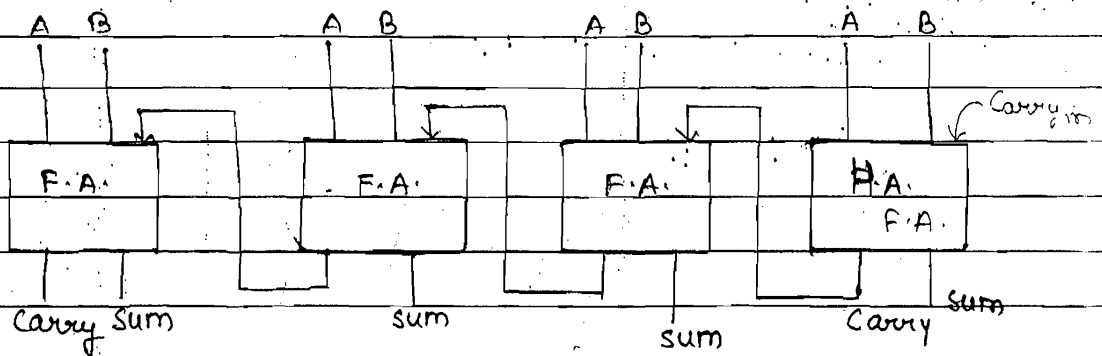


$b = \overline{P_2}$

Full-Adder Using Half-adder:



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



Note:

• - $N(F.A.)$ when there is a carry in is given.

• - $(N-1) F.A. + 1 H.A.$

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

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

$$\therefore N=4$$

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

$$(8 - 1)$$

$$7 = H.A.$$

$$(N-1)$$

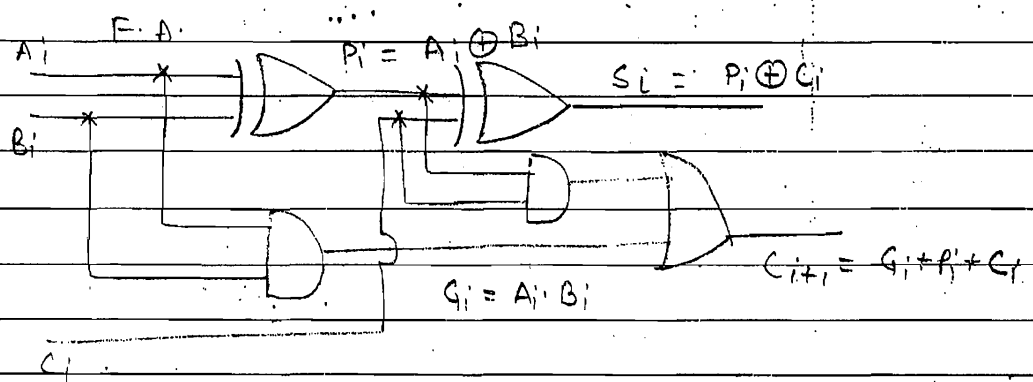
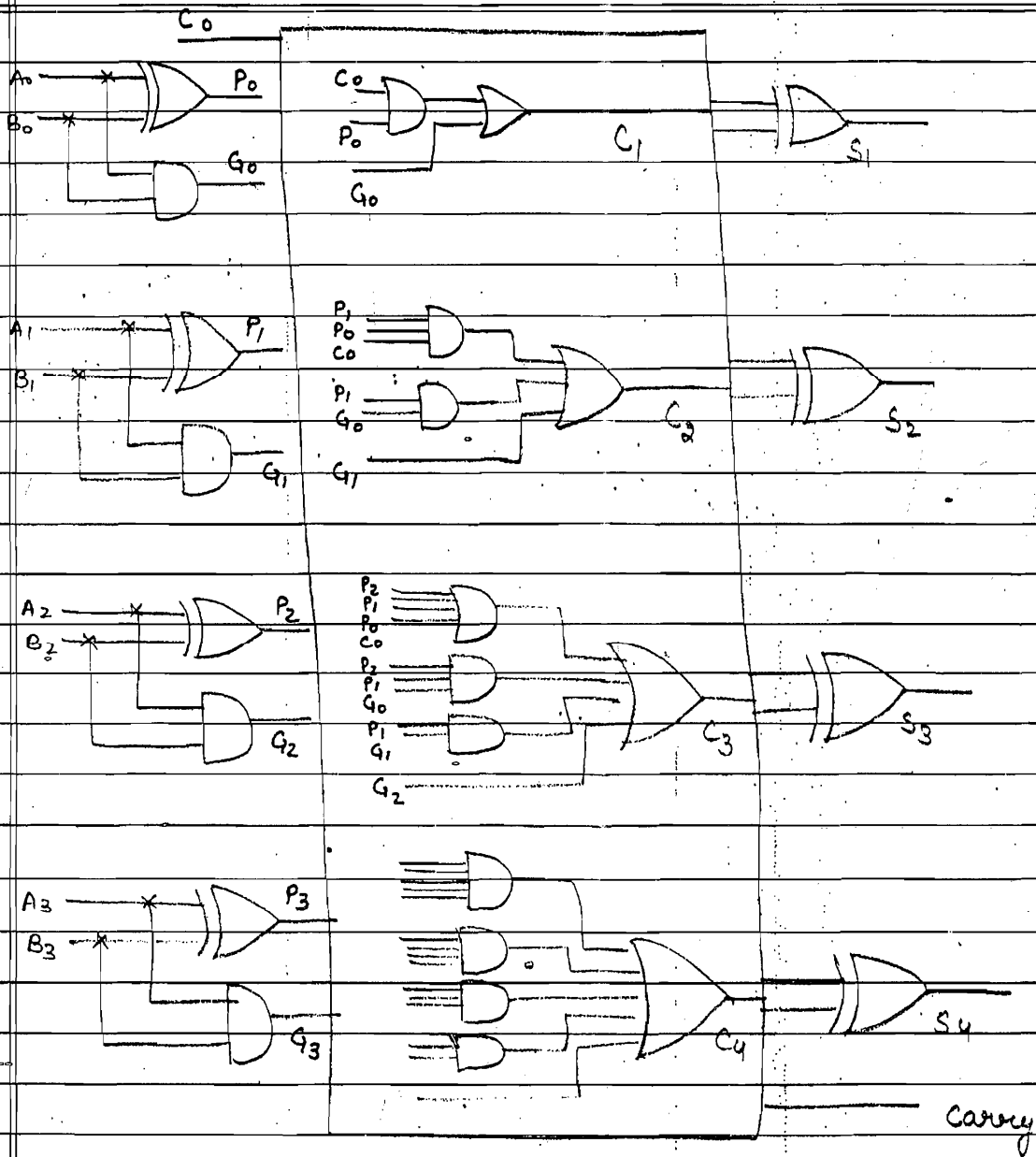
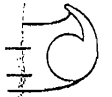
$$4 - 1$$

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

A'

AL
FC

de'smat



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

$$C_{i+1} = C_{1+1} = \underline{C_2} = G_1 + P_1 C_1 = G_1 + P_1 [G_0 + P_0 C_0] \\ = \underline{G_1 + P_1 G_0 + P_1 P_0 C_0}$$

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

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

Short-cut:

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

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

Complement - Methods:

	dec	bin	oct	Hexa
n's $\Rightarrow$	10's	2's	8's	16's
(n-1)'s $\Rightarrow$	9's	1's	7's	15's

n $\rightarrow$ radix
(or)
base

9's Complement:

$$\begin{array}{r} 9 \\ 7_{(10)} \\ \hline 2 \end{array}$$

$$\begin{array}{r} 9 \quad 9 \quad 9 \\ 9's \leftarrow 7 \cdot 3 \\ \hline 8 \quad 2 \cdot 6 \end{array}$$

1's Complement $\Rightarrow$

1's $\leftarrow 0$	1's $\leftarrow 1$
$\downarrow$	$\downarrow$
1	0

2's Complement $\Rightarrow$

2's $\leftarrow 0$	2's $\leftarrow 1$
$\downarrow$	$\downarrow$
1	0
$\downarrow$	$\downarrow$
0	1

10's Complement $\Rightarrow$

10's $\leftarrow (7)_{10}$	10's $\leftarrow 17.3$
$\downarrow$	$\downarrow$
2	82.6
$\downarrow$	$\downarrow$
3	82.7

2's

1 0 0 0	$\uparrow$
0 1 1 0	$\downarrow$

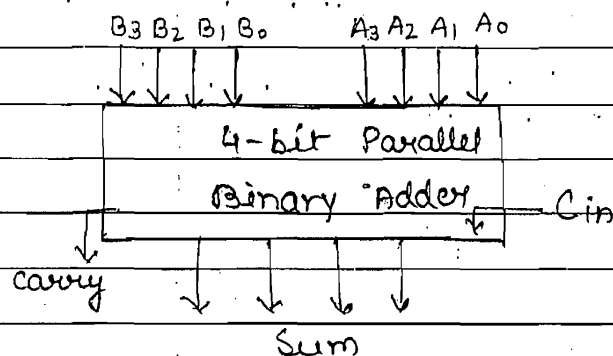
2's

1 0 1 0	$\downarrow$
0 1 0 1	$\downarrow$

2's

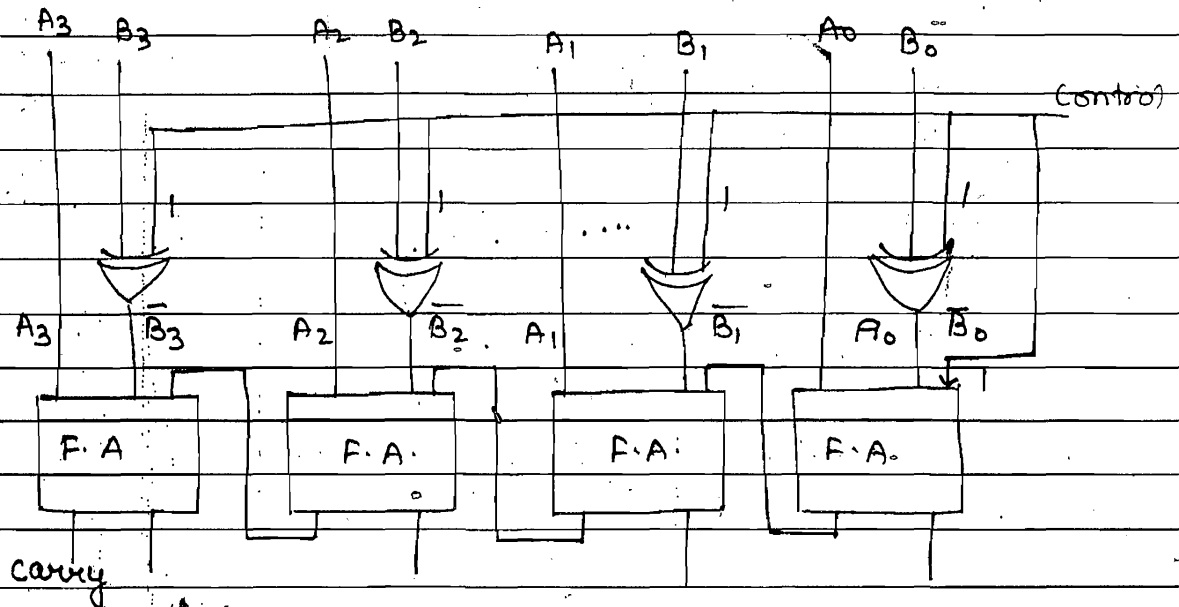
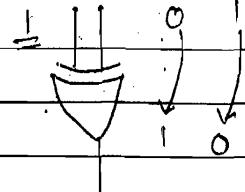
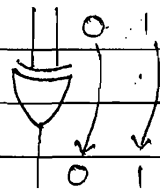
1 0 0 0	$\downarrow$
1 0 0 0	$\downarrow$

2's complement Adder / Subtractor :



$$A \rightarrow A_3 A_2 A_1 A_0$$

$$B \rightarrow B_3 B_2 B_1 B_0$$



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

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

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

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

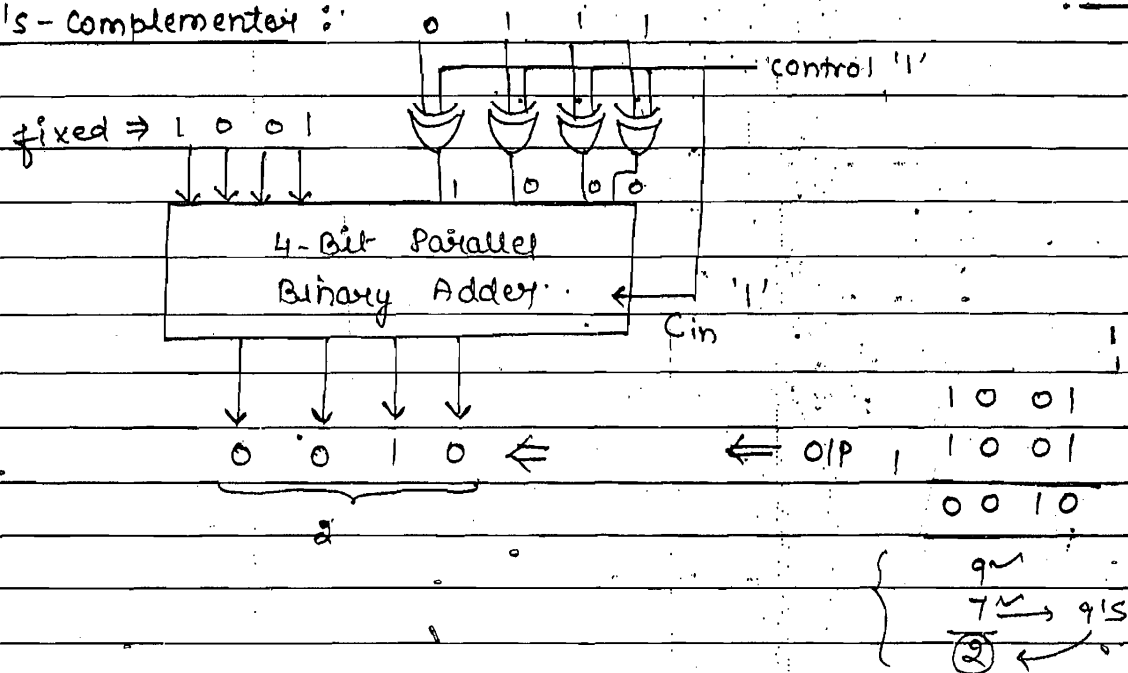
$$\Rightarrow A_0 - B_0$$

9's complement: 3 live within and 3 exist without

class 93

Date
Page

* 9's - Complementer :



* 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) $\Rightarrow$

5 $\rightarrow 0101$
4 $\rightarrow 0100$
9 $\rightarrow 1001$ valid

Case (ii) $\Rightarrow$

5 $\rightarrow 0101$
7 $\rightarrow 0111$
1100 invalid

0110 $\leftarrow +6$

0001/0010

12

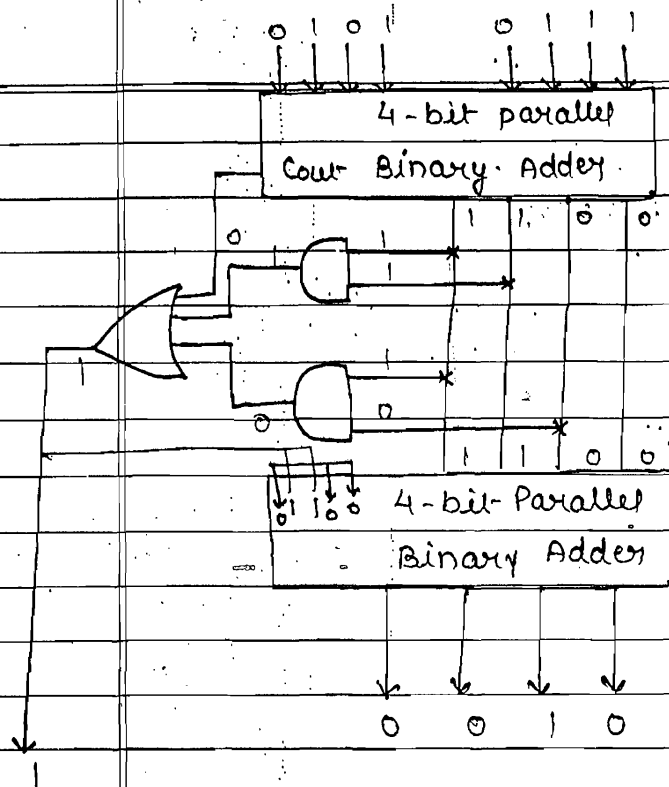
BCD invalid six no's

Case (iii) $\Rightarrow$

8 $\rightarrow 1000$
9 $\rightarrow 1001$
carry 10001 $\leftarrow$ invalid

0110

0001/0111 $\leftarrow 17$



B) BCD - Subtractor :

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

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

Case (i) $6 \rightarrow 0110$
 $-4 \rightarrow 0101$ [9's comp. of 4]
 $\hline 1011$ invalid
 $0110 \rightarrow 6$
 $\hline 0001$
 (+) endaround carry
 $\hline 0010$
 $\hline 0010$
 (+) 2

Case (ii) $4 \rightarrow 0100$
 $-6 \rightarrow 0011$ [9's comp. 6]
 $\hline 0111$ ← valid
 ↓
 No-carry

1. Valid
 2. No-carry } Result (-ve)
 for final answer Take's 9's comp.
 $0111 \rightarrow 7 = 9$

No. Representation :

Case (i) (+)ve number

+51

32	16	8	4	2	1
1	0	0	0	1	1

$\boxed{0} \ 110011$
 $\downarrow$ sign bit $\leftarrow$ magnitude

Case (ii) -ve number

-51

$\boxed{1} \ 110011 \Rightarrow$ sign-magnitude
 $\downarrow$ sign bit $\leftarrow$ magnitude

$\boxed{1} \ 001100 \Rightarrow$ 1's comp.
 $\downarrow$ sign bit $\leftarrow$ 1's comp.

001100

+1

$\boxed{1} \ 001101 \Rightarrow$ 2's complement
 $\downarrow$ sign bit $\leftarrow$ 2's comp

Given

sign

1's

2's

0101

+5

+5

+5

1101

-13

(-2)

(-3)

$\boxed{1} \ 1101$ sign.
 $\downarrow$ 13

11101

 $\downarrow$ 0010-2 $\rightarrow$ 1's comp.

11101

 $\downarrow$ 0010

+1

0011

(-3)

Range ———→ Sign
 → 1's comp.

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

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

Sign	1's comp.
000 +0	000 +0
001 +1	001 +1
010 +2	010 +2
011 +3	011 +3
100 -0	100
101 -1	101
110 -2	110
111 -3	111

fig (i)

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

Sign	2's comp. for $n = 3$
000 +0	000 +0 range $\Rightarrow -4 \text{ to } +3$
001 +1	001 +1
010 +2	010 +2
011 +3	011 +3
100 -0	100 -
101 -1	101
110 -2	110
111 -3	111

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

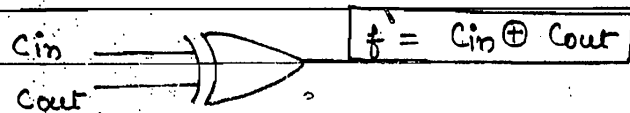
$$- (2^2 - 1) \text{ to } (2^2 - 1)$$

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

— if MSB 1

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

Overflow - Condition :



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

eg... $5 \rightarrow x \ 0 \ 1 \ 0 \ 1$
 $4 \rightarrow y \ 0 \ 1 \ 0 \ 0$
 $z \ 1 \ 0 \ 0 \ 1$

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

→ additional bit to overcome the overflow.

$5 \rightarrow \boxed{0} \ 0 \ 1 \ 0 \ 1$
 $4 \rightarrow 0 \ 0 \ 1 \ 0 \ 0$
 $0 \ 1 \ 0 \ 0 \ 1$

Tx:

Rx:

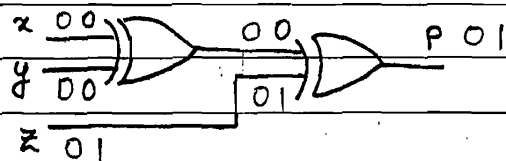
error

Even-Parity

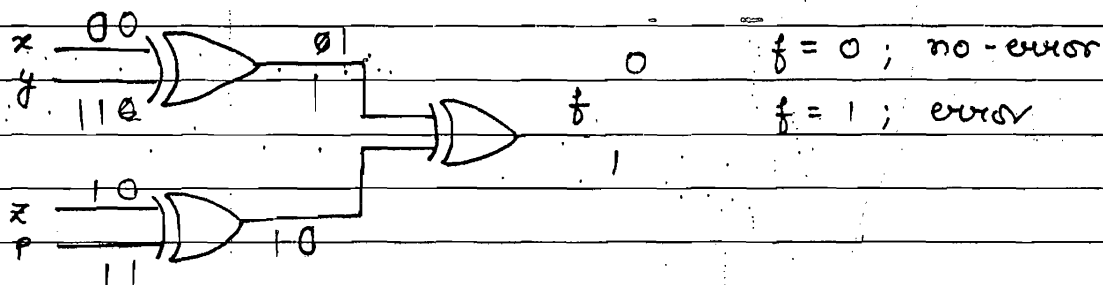
eg.. 0 1 0 1

0 1 1 1

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

AU

* 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-Parity7-bit12-bit15-bit16-bit

	P ₄	P ₂	P ₁
	2 ³	2 ¹	2 ⁰
	x	y	z

P₁
P₂
P₄P₁
P₂
P₄
P₈P₁
P₂
P₄
P₈P₁
P₂
P₄
P₈
P₁₆

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

$$P_1 (1, 3, 5, 7) \Rightarrow [P_1, \overset{\text{odd}}{111}] \Rightarrow P_1 = 1$$

$$P_2 (2, 3, 6, 7) \Rightarrow [P_2, \overset{\text{even}}{101}] \Rightarrow P_2 = 0$$

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

error

Rx:

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

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

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

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

$$\alpha \beta \gamma \Rightarrow 0, 1, 1 \quad (011) \Rightarrow \textcircled{3}$$

$\Rightarrow$ 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

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

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

$$7 \times 10^1 \rightarrow 70$$

$$72_{(10)}$$

* Decimal to Binary:

2	12	
2	6	$\rightarrow 0$
2	3	$\rightarrow 0$
	1	$\rightarrow 1$

$$12_{(10)}$$

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

Fractional : (Successive multiplication) :-

$$0.75_{(10)}$$

$$2 \times 0.75_{(2)} = 1.50$$

$$0.75_{(10)} = 0.11_{(2)}$$

$$2 \times 0.50 = 1.00$$

Dec to Oct :-

$$286_{(10)}$$

8	286	
8	35	→ 6
	4	→ 3

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

Dec to Hexa :-

$$428_{(10)}$$

16	428	
16	26	→ 1A
	1	→ 1

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

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 to Dec. $\Rightarrow$

Bin to Oct $\Rightarrow$

11011.0101

011/011.010/100

$\Rightarrow 33.24_{(8)}$

e) Bin. to Hexa :-

11101.10

0001/1101.1000

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

Any to Bin :-

a) dec. to bin :

oct to Bin :

75.36₍₈₎

7/5.3/6₍₈₎

$\Rightarrow 111/101.011/110_{(2)}$

b) Hexa to Bin $\Rightarrow$

A B C . D₍₁₆₎

A/B/C.D

10101011100.1101₍₂₎

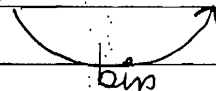
Oct to Any $\Rightarrow$

oct to Bin

oct to Dec

a) Oct to Hexa :

275.3₍₈₎

 Bin

0000/1011/1101.0110

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

Any to Oct :-

dec to Oct ✓

bin to Oct ✓

Hexa to Oct ✓

bin

AB9.2C(16)

101010111001.001011000

5271.130(8)

Hexa to Any:

Hexa to Dec. ✓

Hexa to Bin. ✓

Hexa to Oct. ✓

Any to Hexa:

Dec. to Hexa ✓

Bin. to Hexa ✓

Oct. to Hexa. ✓

2's Complement Arithmetic :

- Minuend is added with 2's complement of subtrahend.
if the 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 ⇒ Carry should be ignored.

x ⇒ minuend

- y ⇒ Subtrahend.

if MSB (Sign-bit) = 0 ⇒ Result is +ve [Ans. is true Binary] } Ignore
= 1 ⇒ Result is -ve [Ans. is 2's comp.] } Carry

a) case: (i)

+ 9 →

+ 4 →

01001
01000
01101

Sign - Bit

Sum = +13

Case: (ii) $\Rightarrow$

$$\begin{array}{r}
 +9 \rightarrow 01001 \\
 -4 \rightarrow 11100 \\
 \hline
 100101
 \end{array}$$

↓
Ignored
Carry

↓
Sign Bit

Sum = +5

Case (iii)

$$\begin{array}{r}
 -9 \rightarrow 10111 \\
 +4 \rightarrow 00100 \\
 \hline
 11011
 \end{array}$$

↓
Sign-Bit

Sum = -5

Case: (iv)

$$\begin{array}{r}
 -9 \rightarrow 10111 \\
 +9 \rightarrow 01001 \\
 \hline
 10000
 \end{array}$$

↓
Ignored
Carry

↓
Sign
Bit

Result = 0

Case (iv) $\Rightarrow$

$$\begin{array}{r}
 -9 \rightarrow 10111 \\
 -4 \rightarrow 11100 \\
 \hline
 110011
 \end{array}$$

↓
Ignored
Carry

↓
Sign
Bit

Sum = -13

* 1's complement Arithmetic :-

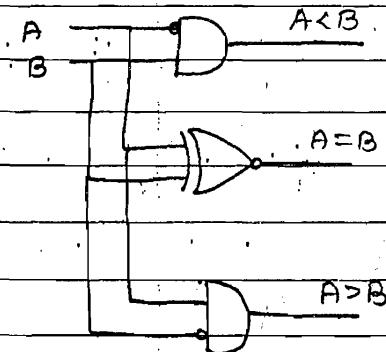
Minuend should be added to 1's complement of subtrahend with a carry generated it should become end-around carry. then check the MSB :

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

Note $\Rightarrow$

Magnitude - Comparator $\Rightarrow$

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



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 $\Rightarrow$

$A < B$

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

 $A < B$

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

 $A = B$

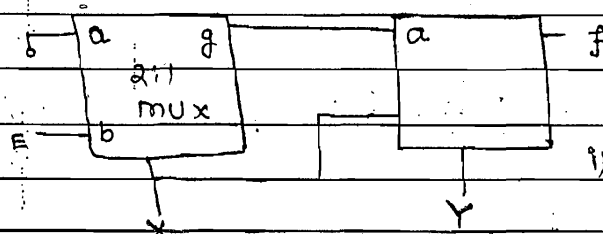
$$(A_3 \odot B_3) (A_2 \odot B_2) (A_0 \odot B_0)$$

 $A > B$

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

P.N. 56

Ques 10



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

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

$$g = E\bar{x}$$

$$ii) \quad \bar{c}b + ca$$

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

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

Hazard :-

Beoz of diff. propagation delays at different parts of ckt- the o/p may not be the expected result known as Hazard.

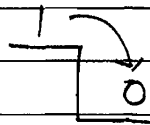
Glitch :-

For temporary interval of time we get a wrong o/p, known as glitch.

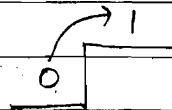
To avoid hazard in ckt ie to get hazard-free ckt, redundant groups should be consider.

Hazard - Type :

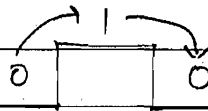
Static '1'



Static '0'



Dynamic



LOGIC - FAMILY

1. Classification of digital IC's :

a. Small scale Integration (SSI) :-

no. of logic gates < 12

no. of components upto 99.

b) Medium scale Integration (MSI) :-

no. of logic gates : 12-99.

no. of components : 100-999.

c) Large scale Integration (LSI) :-

no. of logic gates : 100-999

no. of components : 1000-9999.

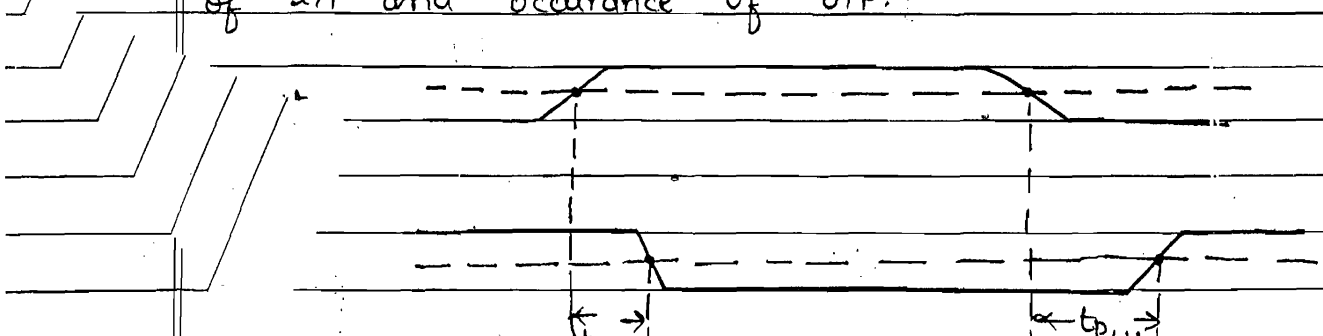
d) Very large scale Integration (VLSI) :

no. of logic gate: above 1000.

no. of compone : above 10,000.

Characteristics parameters :

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



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

$t_{PLH} > t_{PHL}$; 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 I/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.

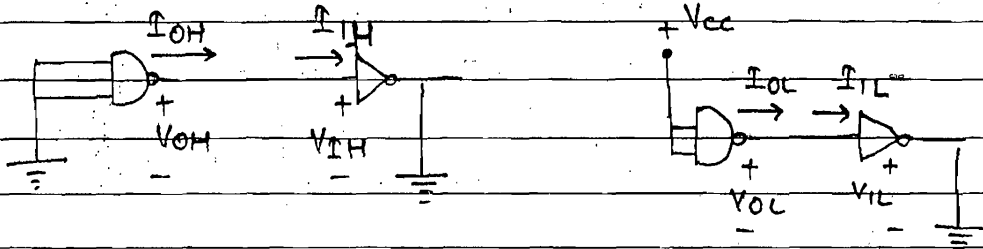
* FAN-IN :- It is ability to handle the no. of inputs for a logic gate.

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

Voltage and Current Parameters :

High

Low

 $V_{IH}(\min) \Rightarrow$ (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 :

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

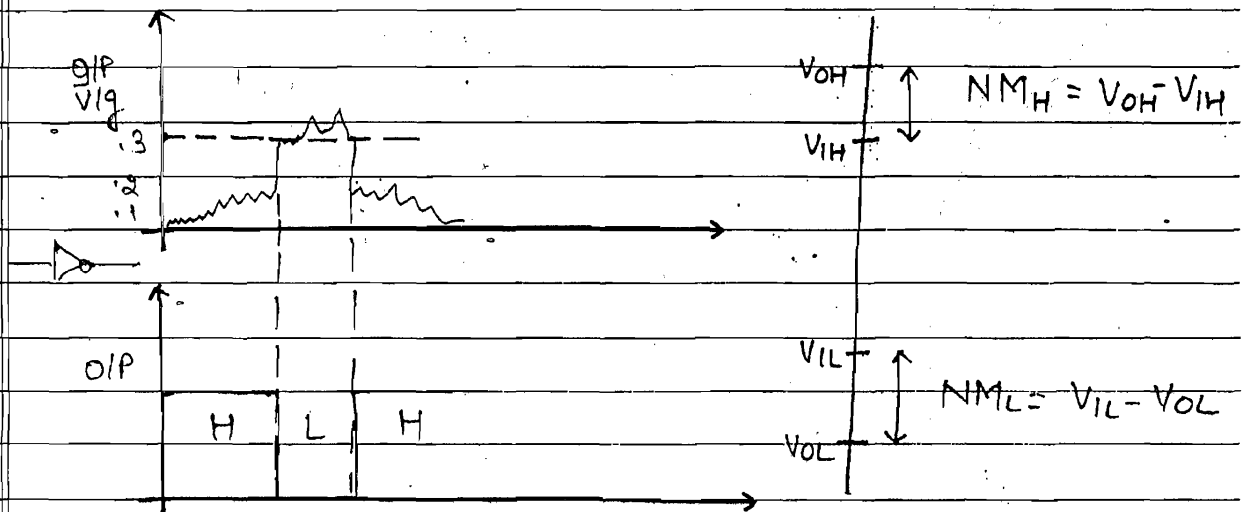
$$\text{Fan-out} = \frac{I_{OL}}{I_{IL}}$$

Note:

From above two values the least value should be considered 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

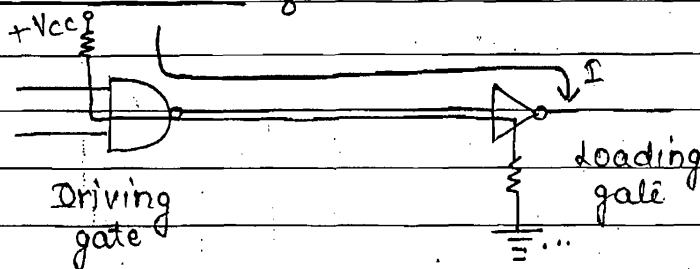
... Industrial Applications

... Consumer Applications.

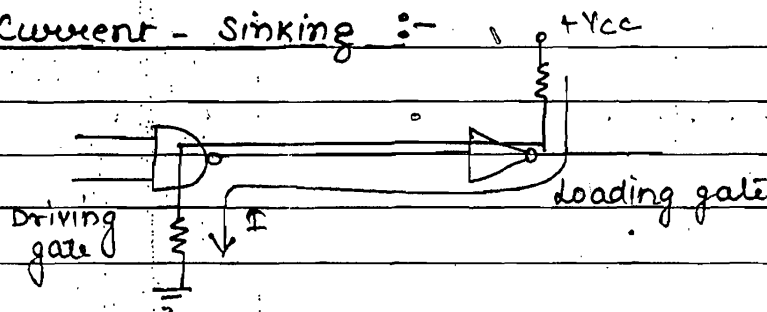
-55°C to +125°C

... for military applications.

* Current - Sourcing :-

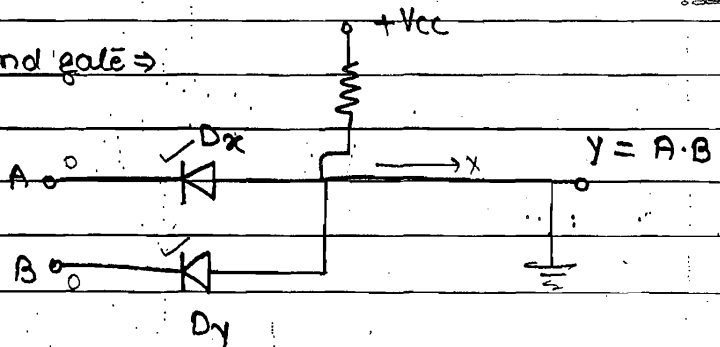


* Current - Sinking :-



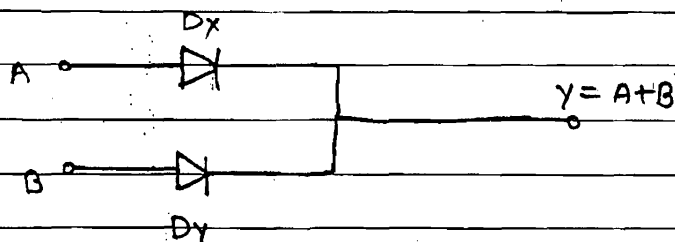
Devices as switching circuits (logic-gates) :-

And gate $\Rightarrow$



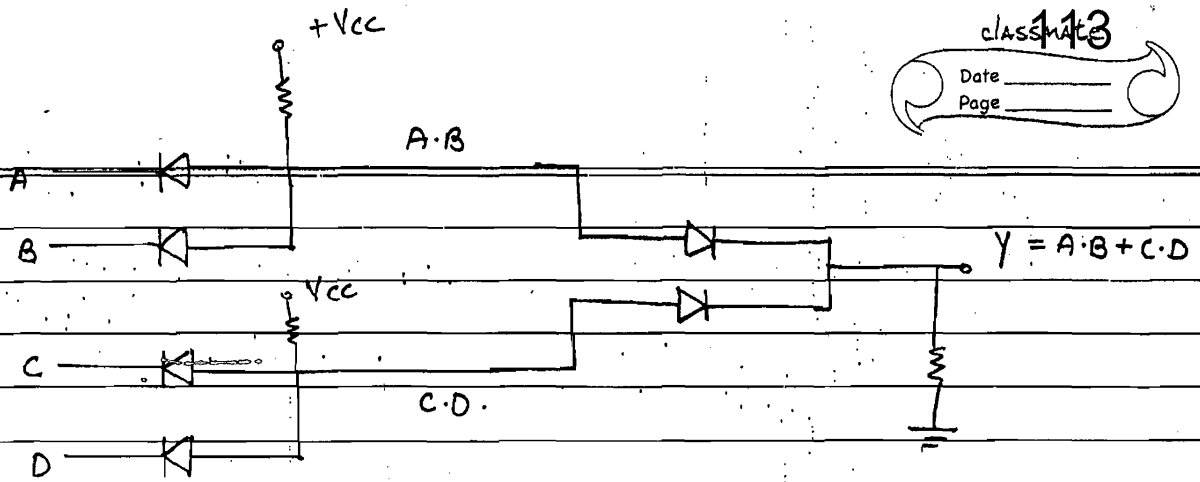
A	B	D_x	D_y	Y
0	0	✓	✓	0
0	1	✓	x	0
1	0	x	✓	0
1	1	x	x	1

OR - Gate $\Rightarrow$



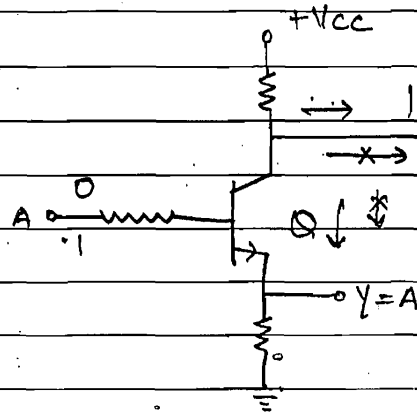
A	B	D_x	D_y	Y
0	0	x	x	0
0	1	x	✓	1
1	0	✓	x	1
1	1	✓	✓	1

Ques:



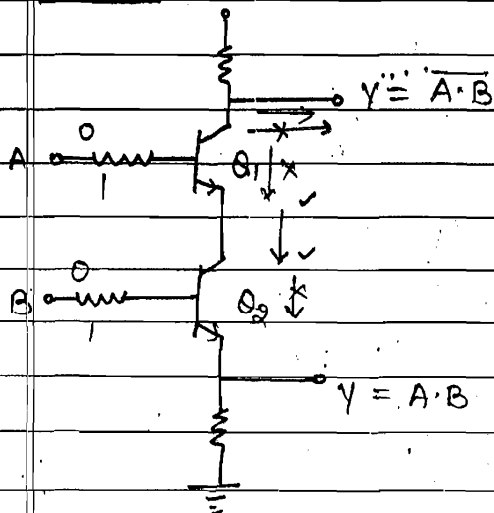
RTL (Resistor - Transistor - logic) :-

NOT - gate



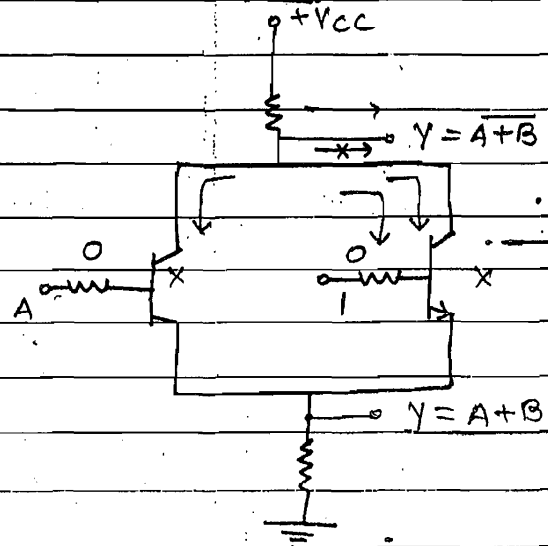
A	Q	Y
0	x	1
1	✓	x

NAND:



A	B	Q ₁	Q ₂	Y
0	0	x	x	1
0	1	x	✓	1
1	0	✓	x	1
1	1	✓	✓	0

NOR:



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.

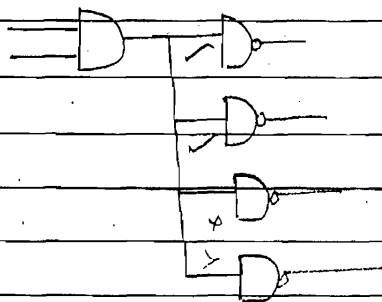
** advantages.

** disadvantages.

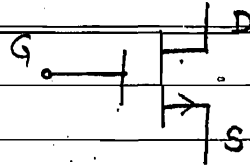
Note :

→ 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.

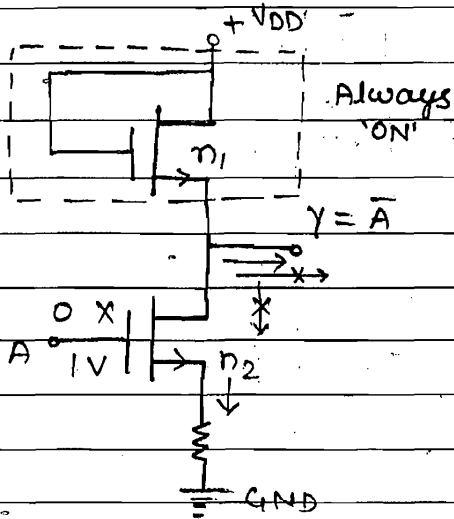


NMOS :-

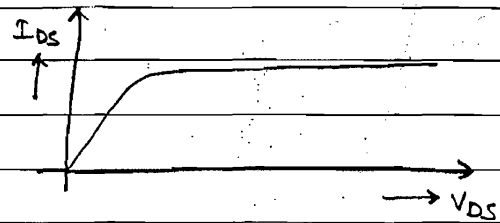


Gate $\Rightarrow$ (+)ve $\Rightarrow$ ON
(-)ve $\Rightarrow$ OFF.

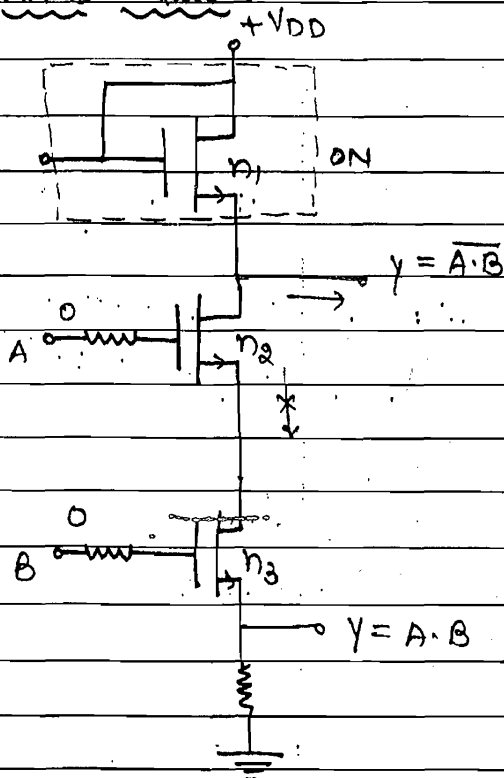
gate diffusion always (+)ve
always 'ON'



A	n_1	n_2	γ	
0	✓	x	1	1
1	✓	✓	0	1

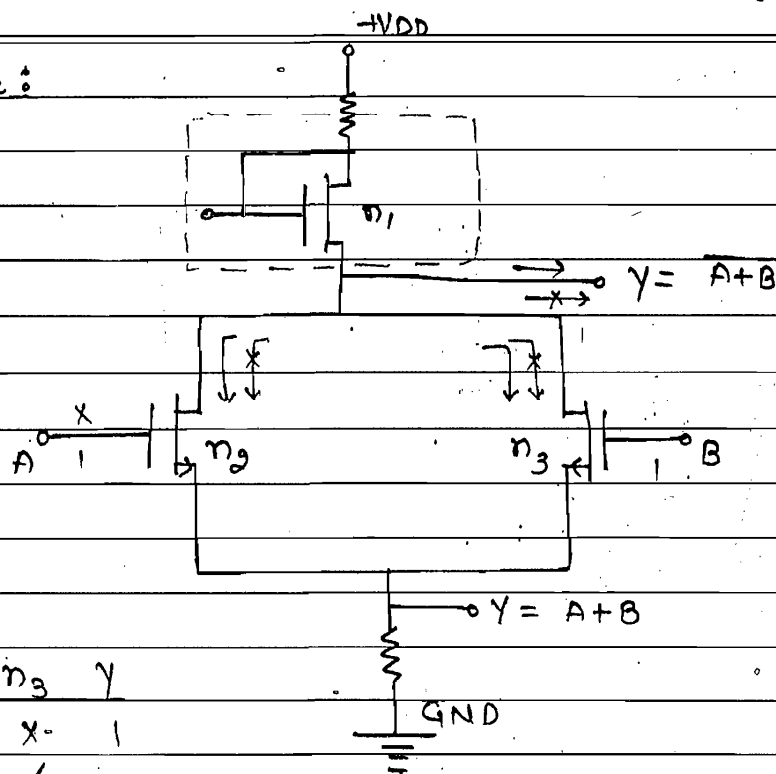


NAND - Gate :

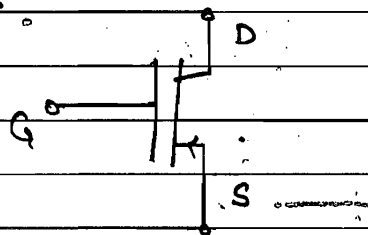


A	B	n_1	n_2	n_3	γ
0	0	✓	x	x	1
0	1	✓	x	✓	1
1	0	✓	✓	x	1
1	1	✓	✓	✓	0

NOR Gate:

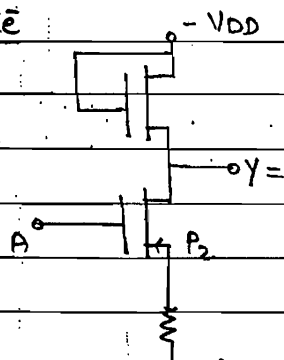


P-MOS Techn.:



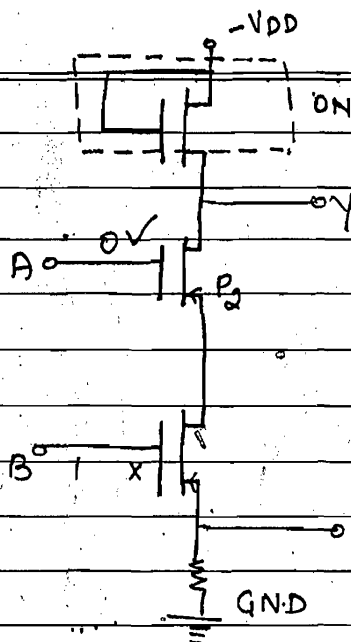
Gate $\Rightarrow$ (+) ve $\rightarrow$ OFF
(-) ve $\rightarrow$ ON

i) NOT-gate



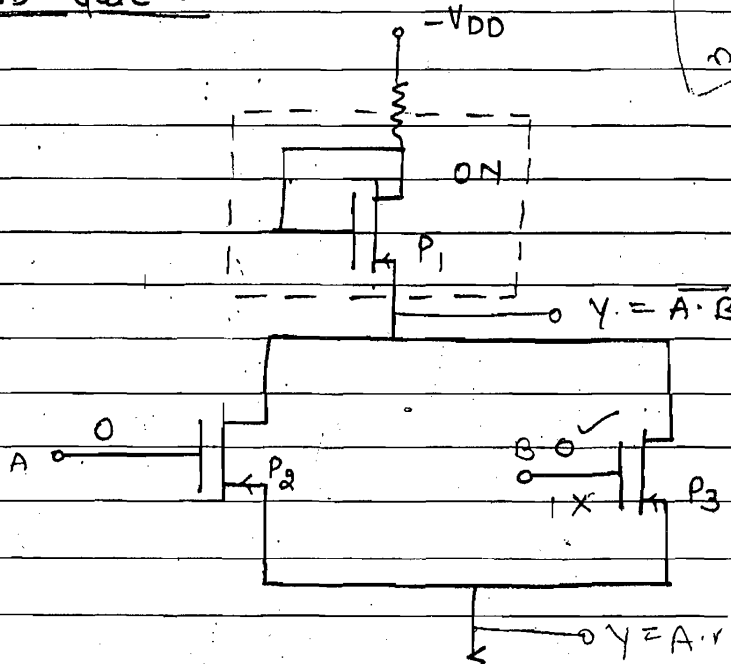
A	P ₁	P ₂	Y
0	✓	✓	1 (GND)
1	✓	✗	0 (-VDD)

NOR-gate $\Rightarrow$

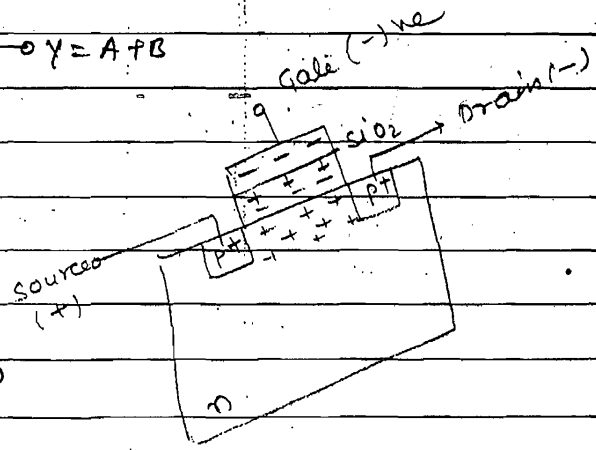


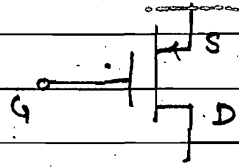
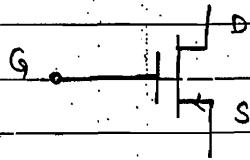
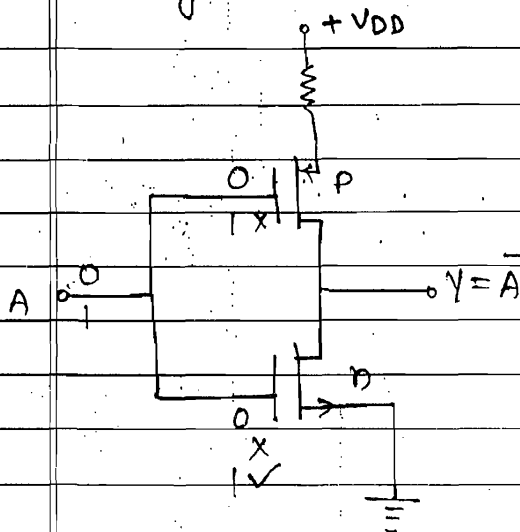
A	B	P ₁	P ₂	P ₃	Y
0	0	✓	✓	✓	1 GND
0	1	✓	✓	X	0 -VDD
1	0	✓	X	✓	0 -VDD
1	1	✓	X	X	0 -VDD

NAND-Gate $\Rightarrow$

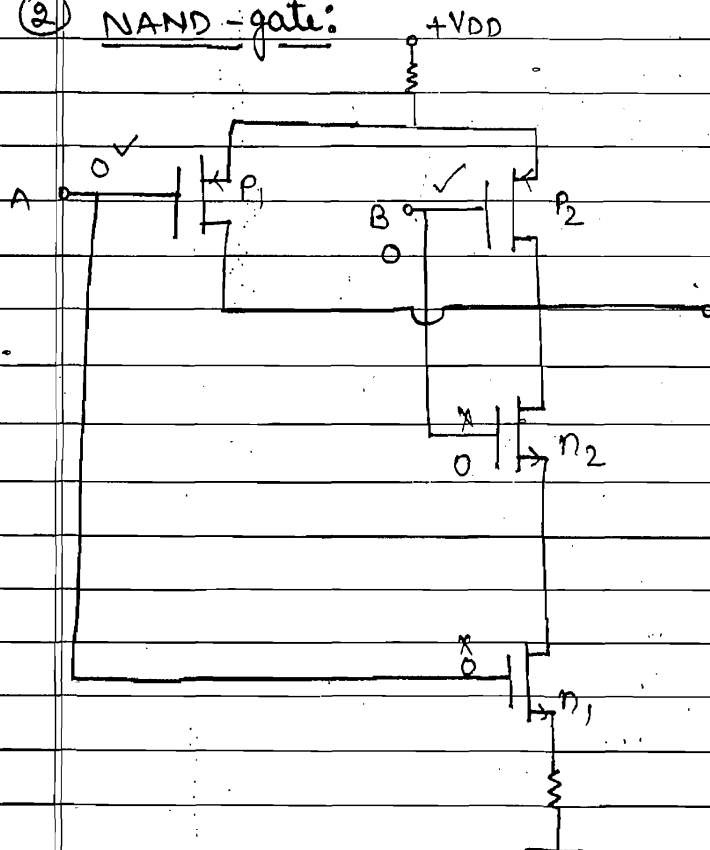


A	B	P ₁	P ₂	P ₃	Y
0	0	✓	✓	✓	1 GND
0	1	✓	✓	X	1 GND
1	0	✓	X	✓	1 GND
1	1	✓	X	X	0 -VDD



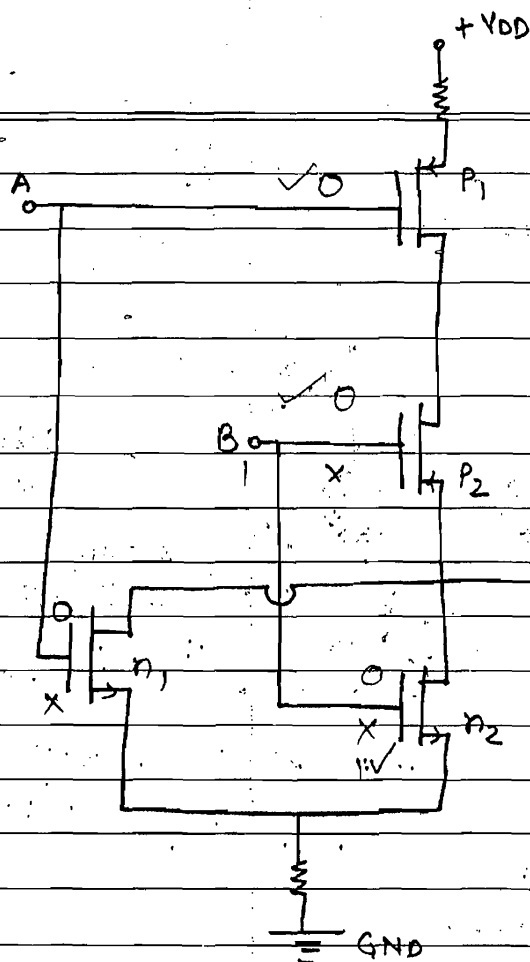
C-MOS Tech. $\Rightarrow$ ① NOT-gate

A	P	n	Y	
0	✓	x	1	+VDD
1	x	✓	0	GND

② NAND-gate:

A	B	P ₁	P ₂	n ₁	n ₂	Y
0	0	✓	✓	x	x	1 +VDD
0	1	✓	x	x	✓	1 +VDD
1	0	x	✓	✓	x	1 +VDD
1	1	x	x	✓	✓	0 GND

③



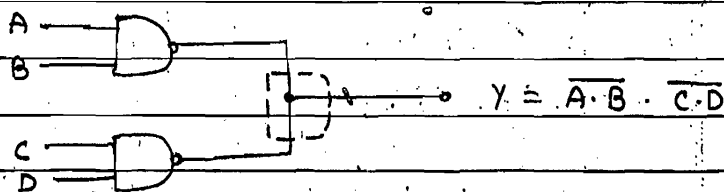
A	B	P ₁	P ₂	n ₁	n ₂	Y
0	0	✓	✓	x	x	1 +V _{DD}
0	1	✓	x	x	✓	0 GND
1	0	x	✓	✓	x	0 GND
1	1	x	x	✓	✓	0 GND

NOTE :

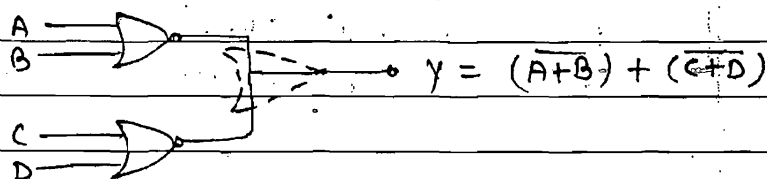
* Floating I/P in TTL is taken as: logic '1'

Wired Logic :-

a) wired AND logic -

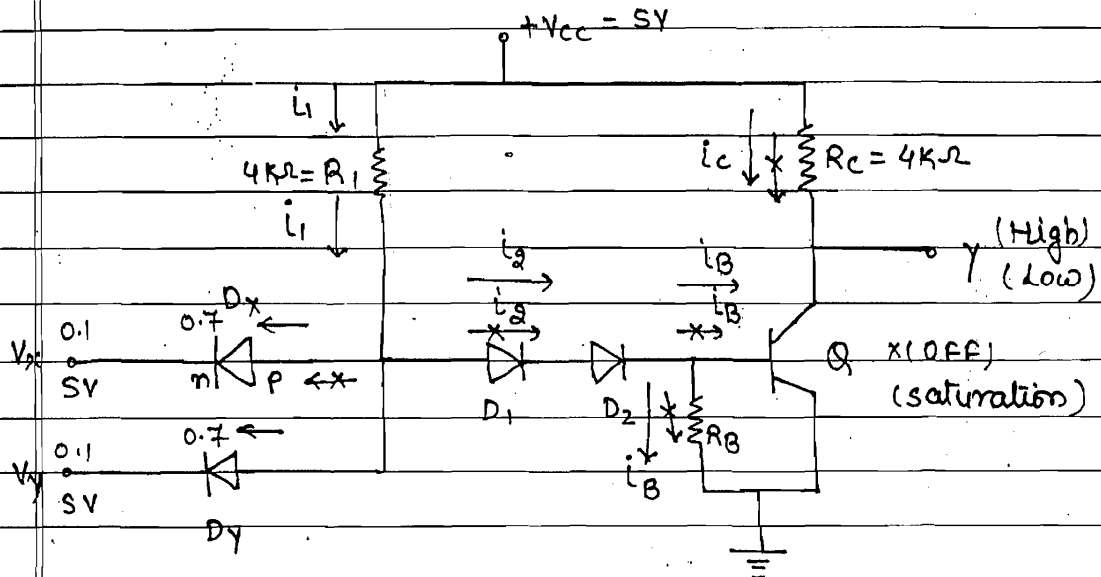


b) wired 'or' logic :-



Diode

Digital - Transistor - Logic (DTL):-



Given :

$$V_Y = 0.7V$$

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

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

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

$$\beta = 25$$

Solution $\Rightarrow$

case (i) $V_X = 0.1V$ $V_Y = 0.1V$

$$I_1 = \frac{V_{CC} - V_1}{R_1} = \frac{5 - 0.8}{4K} = 1.05mA$$

$$I_Q = I_B = I_R = 0$$

 $\Rightarrow Q$ is OFF $Y = \text{High (5V)}$

case (ii) $V_X = 0.1V$, $V_Y = 5V$

same as that of case (i)

 $\Rightarrow Q$ is High OFF. $\Rightarrow Y = \text{high : (5V)}$

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

same as that of case (i):

$\Rightarrow Q$ is OFF.

$\Rightarrow y$ is high.

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

$$i_1 = \frac{V_{CC} - V_{CE(sat)}}{R_1} = \frac{5 - 0.2}{4} = 0.7 \text{ mA}$$

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

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

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

$$i_B = i_2 - i_R$$

$$= 0.7 - 0.08$$

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

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

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

Saturation condi.

$$i_C \leq \beta$$

$$i_B$$

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

$$1.96 \leq 25$$

$$1.96 \leq 25$$

$\Rightarrow Q$ 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:

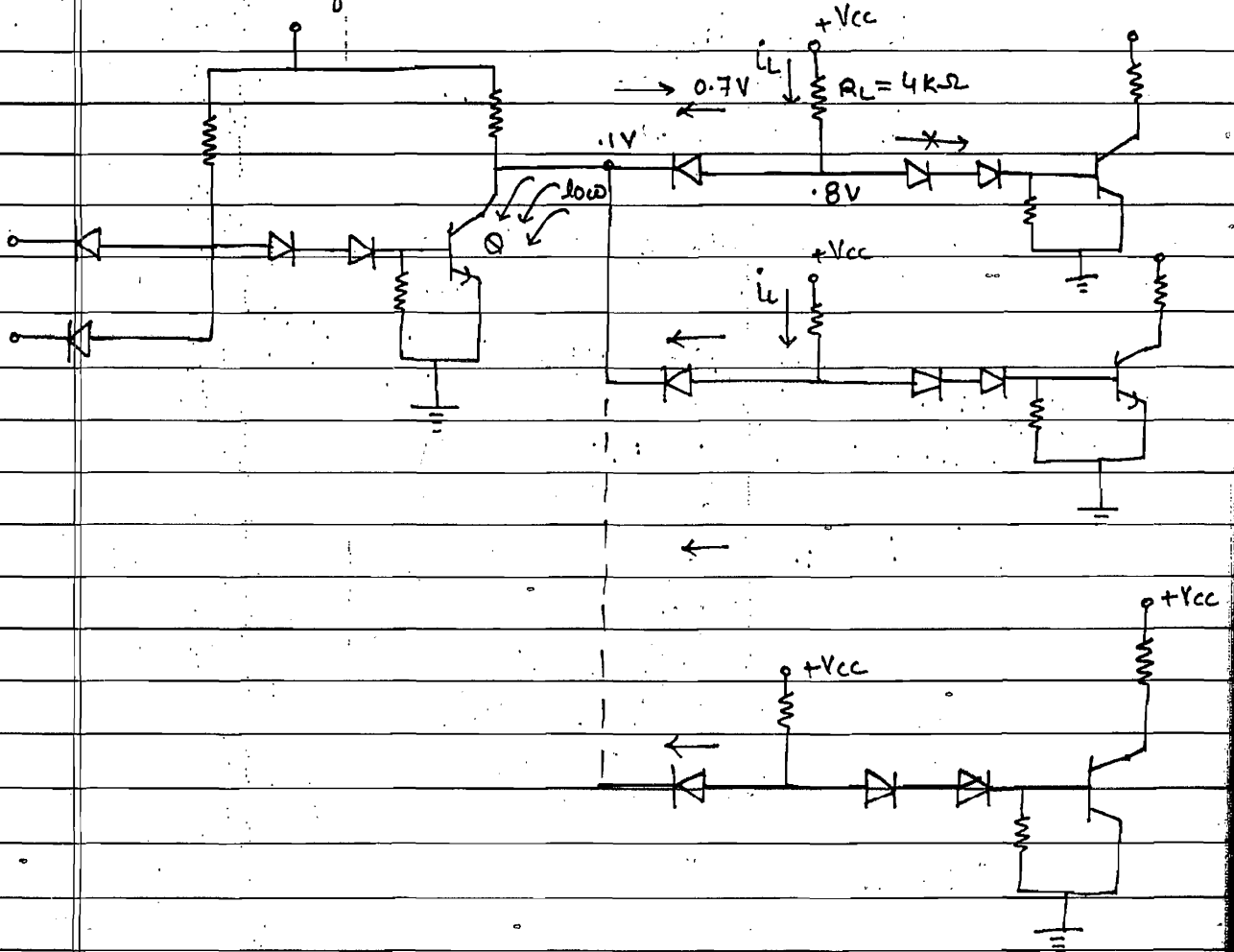
case(i): output of driving gate is low:

In this case the V_{LG} across V_L for the loading gate is $0.1 + 0.7 = 0.8V$ which cause a load-current of

I_L to pass through the collector of saturated transistor of the driving gate. By the no. of loads the collector-current I_C now changes to I_C' i.e. $I_C' = I_C + N \cdot I_L$ and new saturation

condition will be $\frac{I_C + N I_L}{I_B} \leq \beta \left\{ \frac{I_C'}{I_B} \leq \beta \right\}$

where $N \rightarrow$ fan-out.



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

$$\Rightarrow \frac{i_c'}{i_B} \leq \beta$$

we know:

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

$$\Rightarrow \frac{i_c + N i_L}{i_B} \leq \beta$$

$$i_c = \frac{V_{CC} - V_{CE(sat)}}{R_C}$$

$$\Rightarrow \frac{1.22 + N (1.05)}{0.62} = 25$$

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

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

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

In this case the i/p Diode of loading gate is -

'reverse-biased' which causes a reverse saturation

current passing through 'Rc' of driving-gate. By

the no. of loads the current passing through 'Rc' increases causing the voltage drop across 'Rc'.

becoz of which

OIP of driving gate alters from High vlg 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 vlg 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 differenceof vlg: $2.1 - 0.8 = 1.3 \text{ V}$ is the acceptable fluctuating

vlg 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.

Procedure to find ^{power} for dissipation:

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

$$P.D. = \frac{P.D(0) + P.D(1)}{2}$$

★★

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

⇒ Fan-out increases, noise-margin reduces.

★★

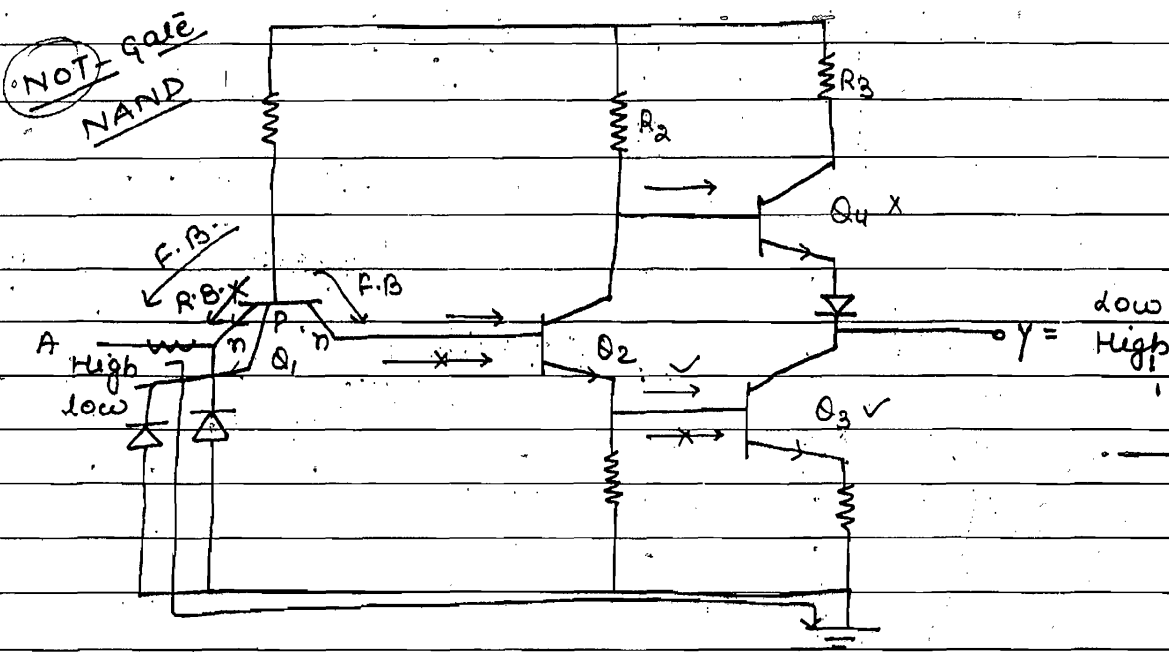
Ques: what will happen if one more diode D_3 is inserted in series with D_1 & D_2 ?

⇒ Fan-out reduces and noise-margin increases.

High-Threshold-logic (HTL) :-

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

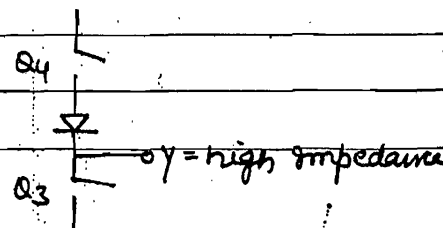
Transistor-Transistor-logic (TTL):-



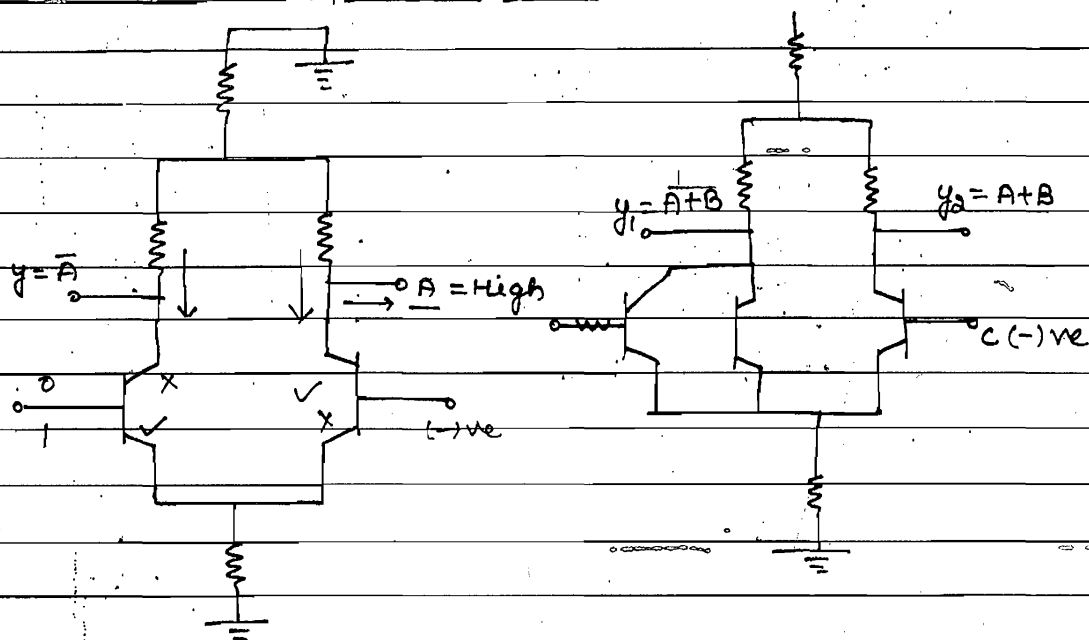
- ⇒ Diode D_1 is used to keep the input-Transistor safe from negative voltage fluctuations.
- ⇒ Diode D_2 is used for proper switching output vlg.

* 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 ^{so} it can give 3-states - logic '0', logic '1' and third x/rp state is high-impedance state.



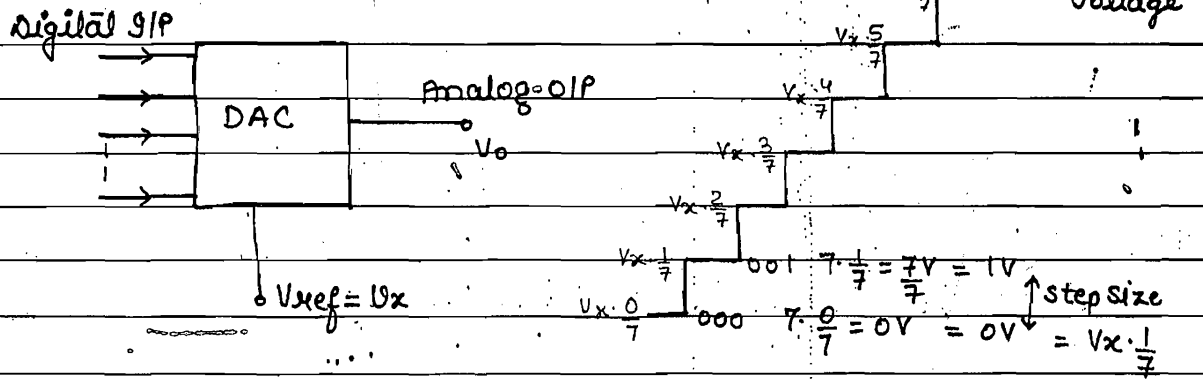
ECL (Emitter - Coupled - logic) :-



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 :-



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

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

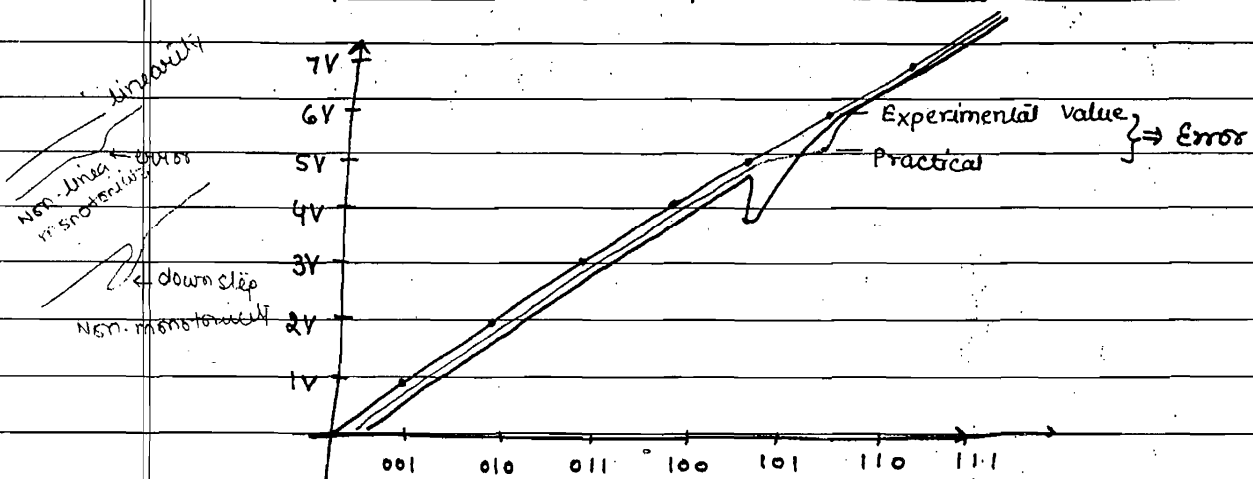
Decimal equivalent of binary data

$$= \frac{V_{f.s.}}{\text{No. of steps}} = \frac{7V}{8} = 1 \text{ LSB value}$$

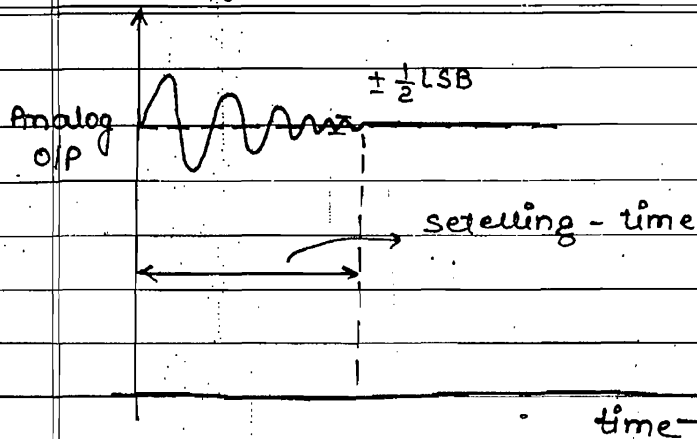
a) % Resolution :-

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

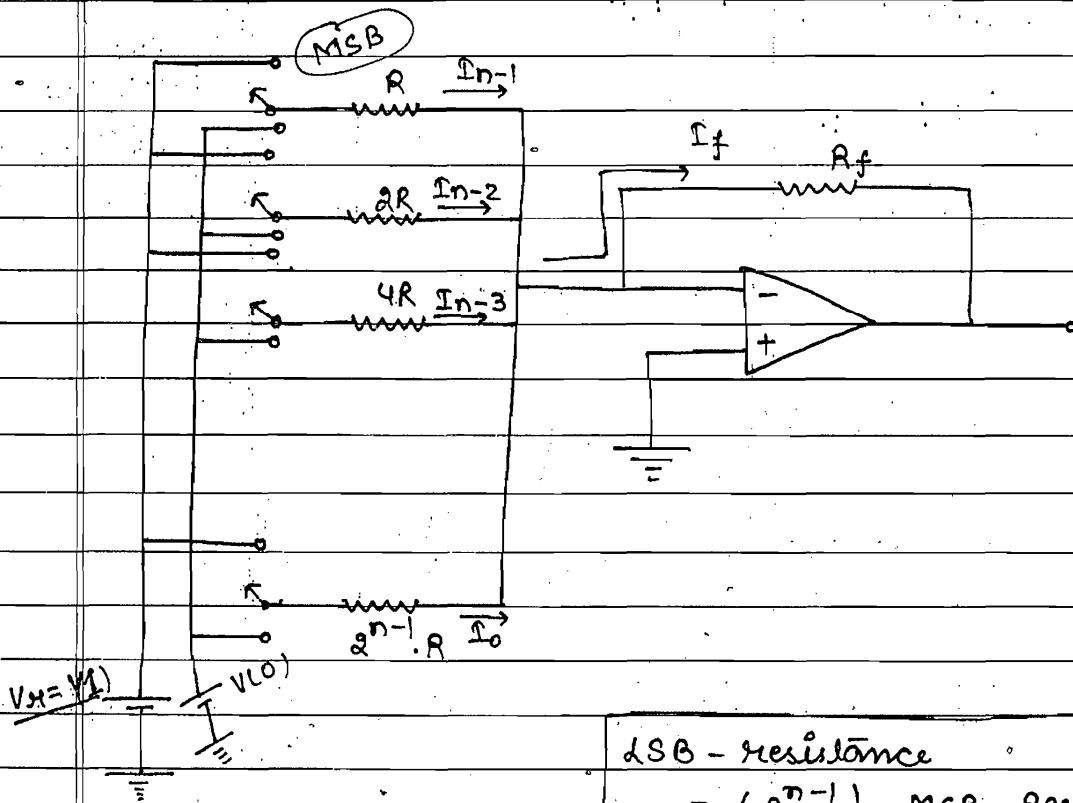
b) Linearity, Non-linearity, monotonicity, non-monotonicity :-



c). Settling - time :



Weighted Resistor Type DAC :-



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

$$V_o = -I_f \cdot R_f$$

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

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

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

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

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

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

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

decimal equi. binary data

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

P.N. 76

Ques: 6

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

Ques: 2

$$V_M = 20V$$

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

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

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

$$V_o = K [11111111]$$

$$20 = K [255]$$

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

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

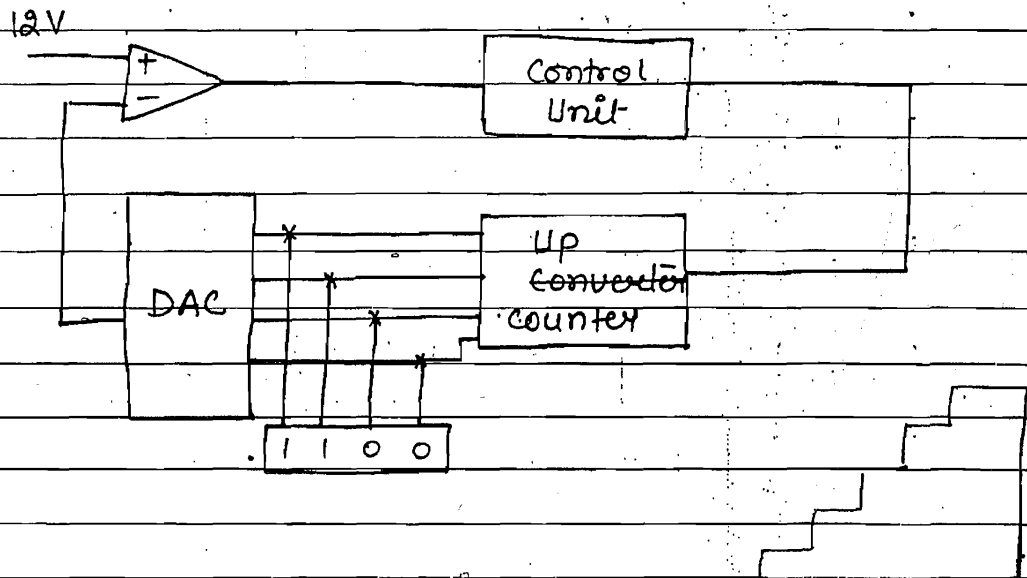
$$V_o = 17.33V$$

8.

ADC :-

Counter-Type - ADC :-

Ramp-Type - ADC :-

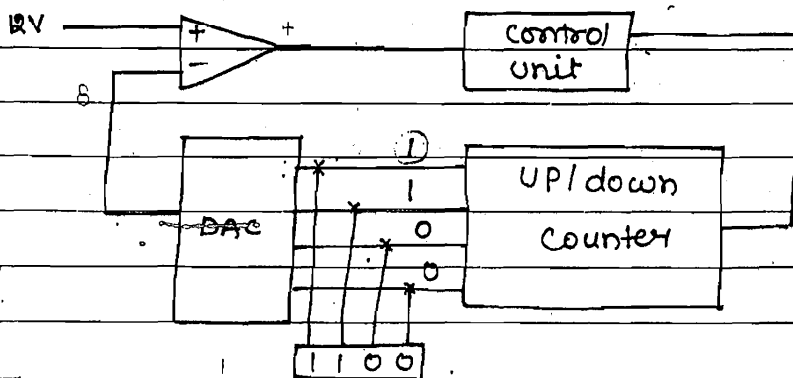


$$t_{c(max)} = (2^n - 1) \times \text{clock-cycles}$$

← Conversion time →

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

2). Successive Approximate ADC :-



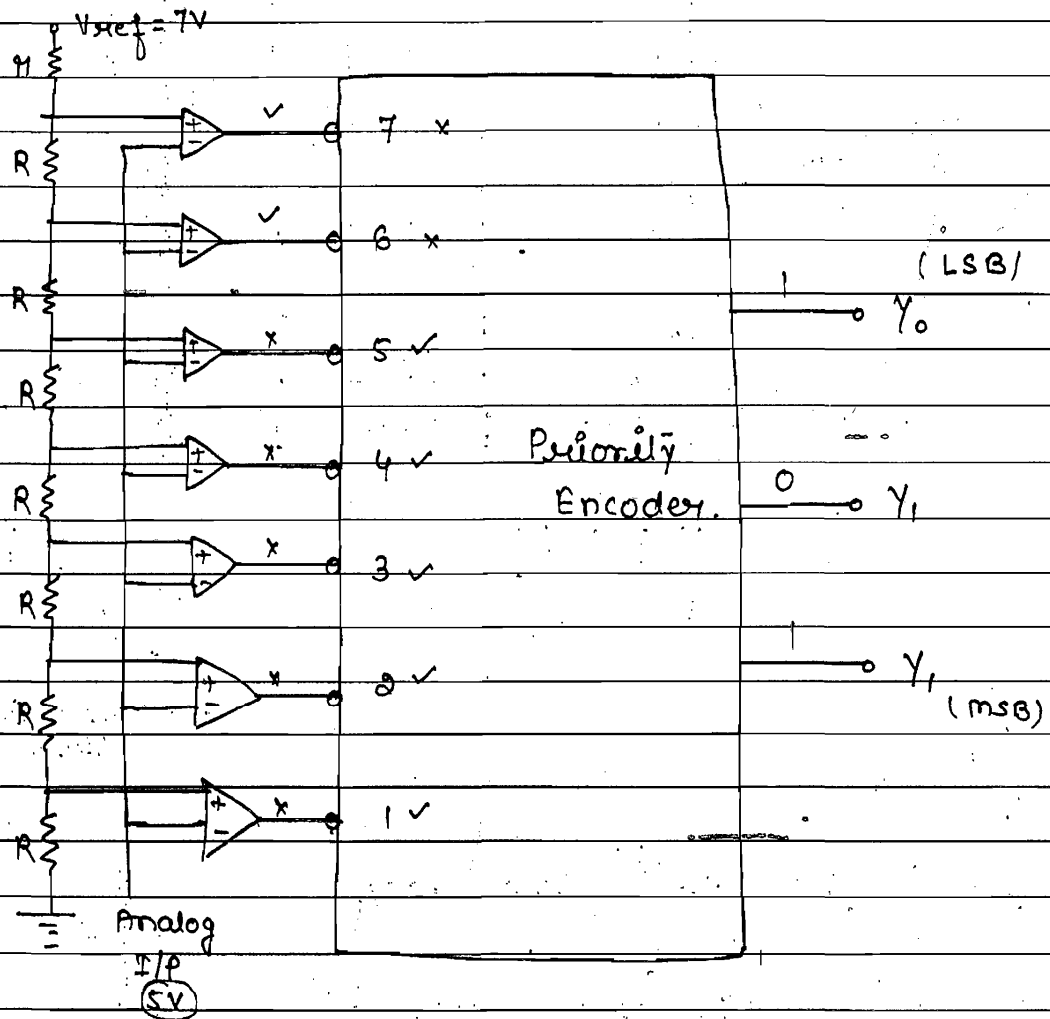
*** operation :

(P.N. 100
Theory book)

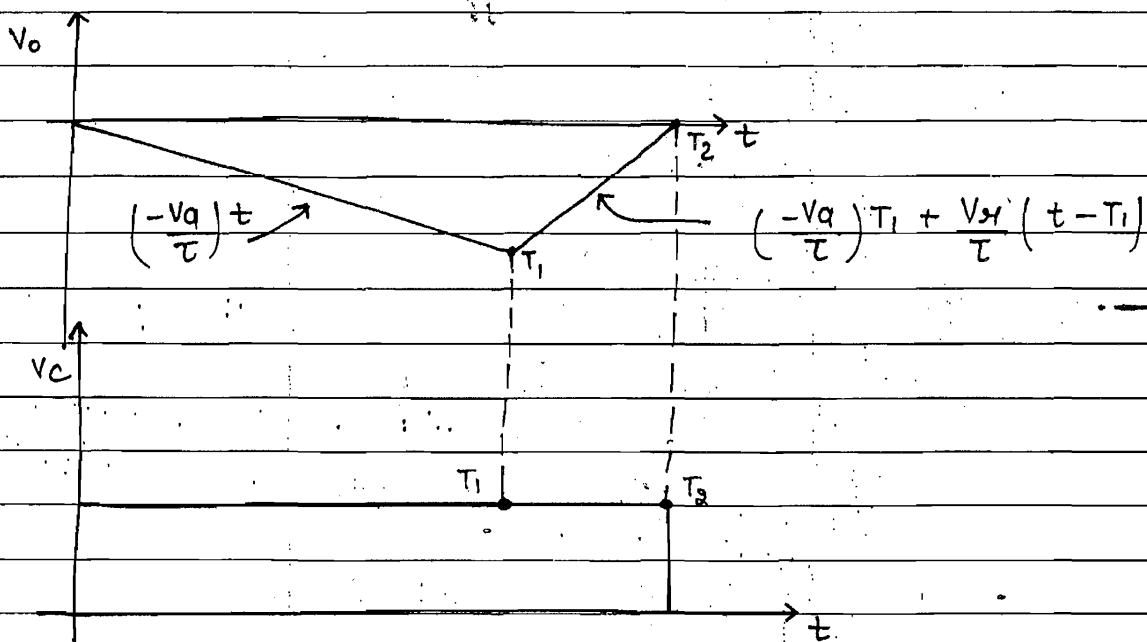
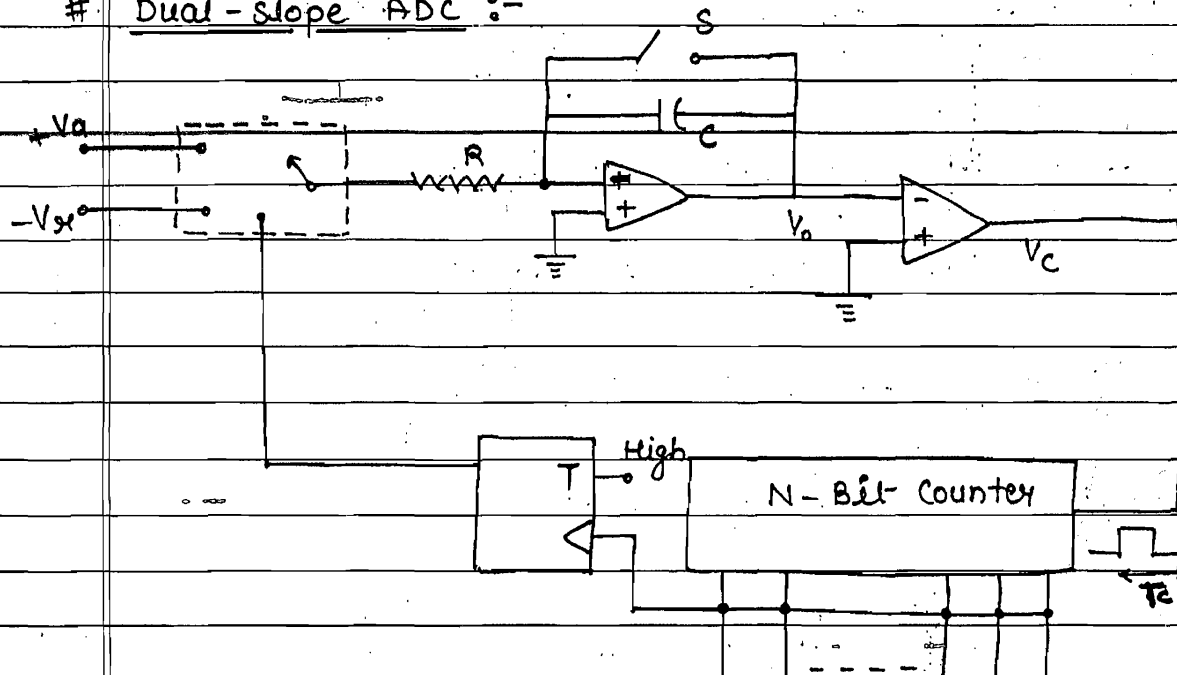
$$t_{c(max)} = N \times \text{clock-pulses (cycles)}$$

↳ no. of bits

- * Flash-Type ADC :-
- * Simultaneously Type ADC :-
- * Parallel Type ADC :-



Dual-slope ADC :-



$$V_0 = -\frac{1}{\tau} \int_0^t V_a dt$$

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

at $t = T_1$

$$V_0 = -\frac{V_a}{\tau} (T_1)$$

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

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

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

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

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

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

$$V_o = \frac{V_r}{2^N} \cdot n$$

$$\text{if } V_{in} = 2^N$$

$$V_o = n$$

$$t_{c(max)} = T_2$$

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

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

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

$$T_1 = 2^N T_c$$

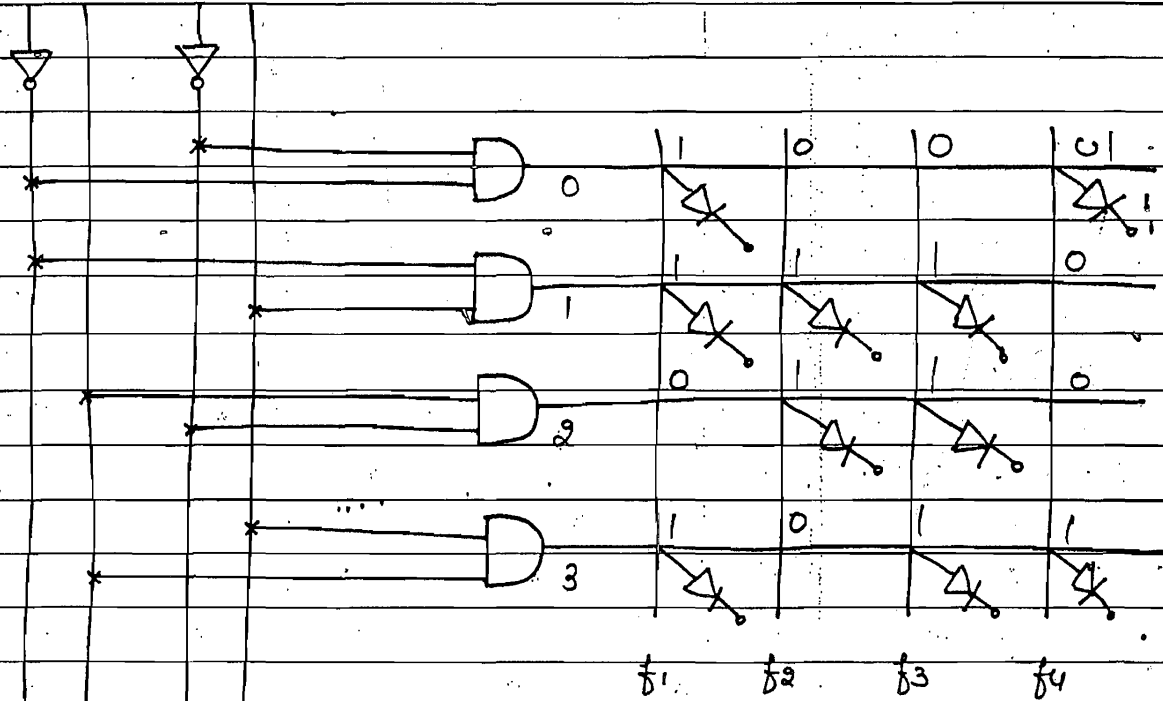
$$T_2 - T_1 = n \cdot T_c$$

$N \Rightarrow$ no. of bits

$n \Rightarrow$ no. of counts

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

MEMORY - DESIGNING :-



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

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

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

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

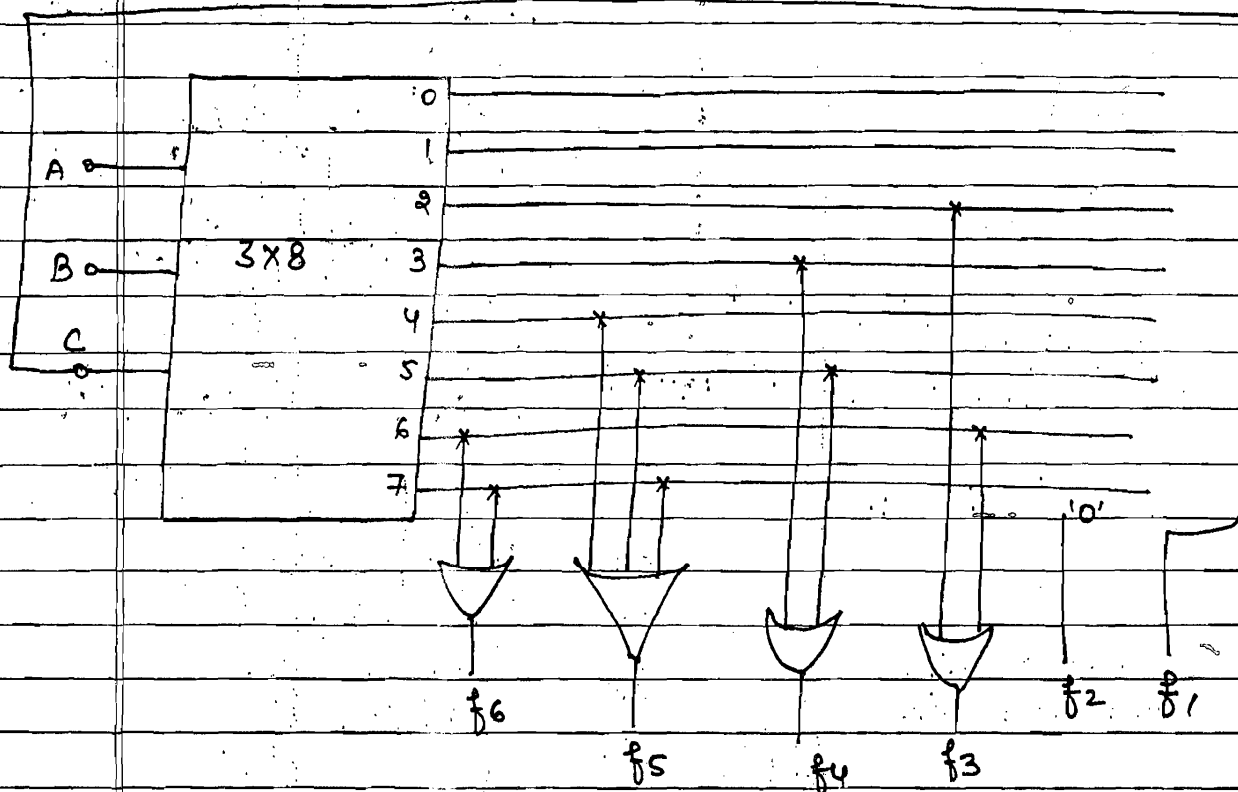
$n \rightarrow$ select-lines

$m \rightarrow$ data-lines

16
4x4
2²x4
2ⁿxm

16x4
Add. lines
data lines

	A	B	C	f_6	f_5	f_4	f_3	f_2	f_1	138
0	0	0	0	0	0	0	0	0	0	classmate
1	0	0	1	0	0	0	0	0	0	Date _____
2	0	1	0	0	0	0	1	0	0	Page _____
3	0	1	1	0	0	1	0	0	1	
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	0	0	0	1	49



Rom-size

$$2^n \times m$$

$$2^3 \times 6$$

$$8 \times 6$$

$$= 48$$

Reduced-Rom

$$2^n \times m$$

$$2^3 \times 4$$

$$8 \times 4$$

$$= 32$$

Master-slave JK flip-flop :-

o- It consists of 2 sections where the 1-section is triggered by the clock-generator and 2nd-section is triggered by the same clock. (G) but through an inverter.

o- So both the sections can't be triggered simultaneously. Final o/p occurrence time, the o/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.

