—: DIGITAL ELECTRONICS :—

Boolean Algebra :—

→ G. Boole in 1854

→ He used switches → ON — '1'
→ OFF — '0'

Minimization :—

→ Boolean algebra is used when no. of variables are less and o/p is either '0' or '1'

→ k-Map is used when no. of variables are upto 5 and o/p can be 0, 1 or X (Don't care)

→ Quine Mccluskey is used for any no. of variable

Theorems :—

(I) NOT :—

$A \longrightarrow \bar{A}$ or A'

$\bar{\bar{A}} \longrightarrow A$

(II) AND Operation :—

$0 \cdot 0$	=	0
$0 \cdot 1$	=	0
$1 \cdot 0$	=	0
$1 \cdot 1$	=	1

**

$A \cdot A$	=	A
$A \cdot 0$	=	0
$A \cdot 1$	=	A
$A \cdot \bar{A}$	=	0

(III) OR Operation :—

$0 + 0$	=	0
$0 + 1$	=	1
$1 + 0$	=	1
$1 + 1$	=	1

**

$A + A$	=	A
$A + 0$	=	A
$A + 1$	=	1
$A + \bar{A}$	=	1

→ $A + A + A - - - - = A$

→ $1 + A + B + C - - - - + Z = 1$

Ques:- Minimize $AB + A\bar{B}$

Soln:- $A(B + \bar{B}) = A$

→ B (Reductant)

Ques:- $A\bar{B} + AB\bar{C} + ABC$ Min. no. of NAND

Soln:- $A\bar{B} + AB(C + \bar{C})$

$= A\bar{B} + AB = A(B + \bar{B}) = A$

→ Zero Nand gate

Advantages of Minimization:-

(i) No. of logic gates will decreases

(ii) Speed inc.

(iii) Power dissipation dec.

(iv) Cost dec.

(v) Complexity of ckt dec.

Ques:- Minimize $A\bar{B} + AB + \bar{A}B + \bar{A}\bar{B}$ → SOP

Soln:- $A(\bar{B} + B) + \bar{A}(B + \bar{B})$

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

Ques:- Minimize $(A+B)(A+C)$

Soln:- $A \cdot A + AC + AB + BC$

$= A + AC + AB + BC$

$= A(1 + C + B) + BC = A + BC$

Short-cut:-

$(A+B)(A+C) = A + BC$

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

Ques:- Minimize $(A+B+C)(A+\bar{B}+C)(A+B+\bar{C})$

Soln:- $(A+B+0)(A+\bar{B}+C)$

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

Ques:- Minimize $(A+B)(A+\bar{B})(\bar{A}+B)(\bar{A}+\bar{B})$

Soln:- $(A+0)(\bar{A}) = 0$

Distribution theorem:-

$A + BC = (A+B)(A+C)$

(1) (2) (3) $\quad (1+2)(1+3)$

Ques:- Minimize the following expressions

(I) $A + \bar{A}B$

(II) $A + \bar{A}\bar{B}$

(III) $\bar{A} + AB$

(IV) $\bar{A} + A\bar{B}$

Soln:- (I) $(A+\bar{A})(A+B) = A+B$

(II) $(A+\bar{A})(A+\bar{B}) = A+\bar{B}$

(III) $(\bar{A}+A)(\bar{A}+B) = \bar{A}+B$

(IV) $(\bar{A}+A)(\bar{A}+\bar{B}) = \bar{A}+\bar{B}$

Ques:- Minimize $ABC + \bar{A}BC + AB\bar{C}$

Soln:- $(A+\bar{A})BC + AB\bar{C}$

$= BC + AB\bar{C}$

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

$= AB + BC$

Ques:- Minimize $AB + \bar{A}B + \bar{A}\bar{B}$

Soln:- $(A+\bar{A})B + \bar{A}\bar{B} = B + \bar{A}\bar{B}$

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

Ques:- $A\bar{B} + BC + B\bar{C} \longrightarrow$ Minimize

Soln:- $A\bar{B} + B(C+\bar{C}) = B + A\bar{B}$

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

Ques:- Minimize $AB + \bar{A}C + BC$ & Also find reductant

Soln:- $AB + C(\bar{A} + B)$

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

$= (AB + C)(B(A+1) + \bar{A})$

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

$= AB + \bar{A}C + BC(A + \bar{A})$

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

$= AB(1 + C) + \bar{A}C(1 + B) = AB + \bar{A}C$

Short cut :-

Consensus theorem :-

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

(I) Three variables (II) Each variable twice

(III) Only one variable is complemented / uncomplement

$\therefore AB + AC + \bar{B}C = AB + \bar{B}C$

For POS form :-

Ques:- $(A+B)(\bar{A}+C)(B+C) \rightarrow$ Minimize

Soln:- $(A+B)(\bar{A}+C)$ / Ans

Ques:- $(A+B)(B+C)(A+\bar{C}) \rightarrow$ Minimize

Soln:- $(B+C)(A+\bar{C})$

Ques:- $(A+B)(\bar{B}+C).(A+C) \rightarrow$ Minimize

Soln:- $(A+B)(\bar{B}+C)$

Ques:- $\bar{A}\bar{B} + \bar{B}\bar{C} + A\bar{C} \rightarrow$ Minimize

Soln:- $\bar{A}\bar{B} + A\bar{C}$

Ques:- $\bar{A}\bar{B} + \bar{B}C + \bar{A}\bar{C} \rightarrow$ Minimize

Soln:- $\bar{B}C + \bar{A}\bar{C}$

Ques:- $(\bar{A}+\bar{B})(\bar{B}+\bar{C})(A+\bar{C}) \rightarrow$ Minimize

Soln:- $(\bar{A}+\bar{B})(A+\bar{C})$

Transposition theorem :-

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

Demorgan's theorem :-

$$\overline{ABC} = \bar{A} + \bar{B} + \bar{C}$$

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

Boolean Algebra :-

(1) Theorems → Minimization

(2) * SOP → Minimal
→ Canonical

(3) POS

(4) Truth Tables

(5) Dual (6) compliment (7) Venn diagram

(8) switching circuits (9) Statements

Minimize → $\underbrace{XY}_{A} + \underbrace{\overline{XY}}_{\bar{A}}\underbrace{WZ}_{B}$

Soln :- $(A+\bar{A})(A+B) \equiv XY+WZ$

SOP :-

→ SOP form is used when o/p logic expression is logic "1".

→ $\underline{ABC} + \underline{\bar{A}BC} + \underline{AB\bar{C}}$ → each term is called minterm

→ min term = 5 → 101 → $A\bar{B}C$

Ques :- For the given truth table minimised SOP is

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	1

Soln:- $Y = \bar{A}\bar{B} + A\bar{B} + AB$

$= \bar{B}(\bar{A}+A) + AB = \bar{B} + AB$

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

Standard SOP form :-

$Y_{(A,B)} = \Sigma m(0,2,3)$

Ques:- Minimize $f(A,B) = \Sigma m(0,1,3)$

Soln:- $\bar{A}\bar{B} + \bar{A}B + AB$

$= \bar{A}(\bar{B}+B) + AB = \bar{A} + AB$

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

Note:-

$A + \bar{A}B = A+B \rightarrow$ Minimal SOP (can't min. further)

$\downarrow$

$A \cdot 1 + \bar{A}B = A \cdot (B+\bar{B}) + \bar{A}B$

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

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

Canonical SOP (in variables)

In canonical form each min term contains all variables

$\Sigma m(1,2,3) \rightarrow$ Standard SOP (in numbers)

Ques:- No of minterms present in canonical expression of $A + \bar{B}C$

(a) 3 (b) 4 (c) 5 (d) 6

Soln:- $A(B+\bar{B})(C+\bar{C}) + \bar{B}C(A+\bar{A})$

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

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

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

$\rightarrow$ 5 minterms

$\Sigma m(1,4,5,6,7)$

→ canonical form / necessary_ used when implemen using MUX and decoders

→ Minimal form is used when implement using logic gates.

POS form :-

→ POS form is used when o/p is logic '0'

→ $(A+B+C)\ (A+\overline{B}+C)\ \underbrace{(A+\overline{B}+\overline{C})}_{\text{Max. term}}$.

→ 5 ⇒ Max. term

↓

1	0	1	→	$(\overline{A}+B+\overline{C})$

$\overline{A}+B+\overline{C}$

Ques:- For the given truth table minimized POS expression

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	0

Soln:- $Y = A\overline{B} + \overline{A}\overline{B} = \overline{B}(A+\overline{A}) = \overline{B}$

$Y(A,B) = \Pi M(1,3) = \overline{B}$

$Y(A,B) = \Sigma m(0,2)$

↓

$Y = \overline{A}\overline{B} + A\overline{B} = \overline{B}(\overline{A}+A) = \overline{B}$

$\Sigma m(0,2) = \Pi M(1,3)$
$m_0 + m_2 = M_1 \cdot M_3$

→ With n-variable max possible minterm or maxterm are 2^n

→ With 2 variable possible no. of logical expression is 16

$$2 \longrightarrow 16$$

→ With n-variable max possible logical expression are 2^{2^n}

0 → 0V
1 → +5V
→ +ve logic

0 → +5V
1 → 0V
→ -ve logic

logic 0 → −5V
logic 1 → 0V
→ +ve logic

Dual :-

→ In digital systems there are two types of logic (i) +ve logic (ii) -ve logic

→ 'Dual expression is used to convert +ve logic to -ve logic or -ve logic to +ve logic

+ve logic AND

A	B	Y
0 (0V)	0 (0V)	0 (0V)
0 (0V)	1 (+5V)	0 (0V)
1	0	0
1	1	1

+ve logic OR

A	B	Y
0 (0V)	0 (0V)	0 (0V)
0 (0V)	1 (+5V)	1 (+5V)
1	0	1
1	1	1

-ve logic OR :-

A	B	Y
1	1	1
1	0	0
0	1	0
0	0	0

→ +ve logic AND = -ve logic OR
-ve logic AND = +ve logic OR

→ AND ⟷ (Dual) OR

→ $AB \overset{\text{dual}}{\longleftrightarrow} A+B$

In dual :-

→ AND ⟷ OR

→ $\cdot \longleftrightarrow +$

→ $1 \longleftrightarrow 0$

→ Keep variable as it is

Ques :- Find dual of $A\bar{B}\bar{C} + \bar{A}BC + ABC$

Soln :- $(A+B+\bar{C})(\bar{A}+B+C)(A+B+C)$

→ For any logical expression if two times dual is used resulting same expression.

Ques :- simplify $(A+\bar{B}+C)(A+B+\bar{C})(A+B+C)$

Soln :- Taking dual

$(A\bar{B}C) + AB\bar{C} + ABC$

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

$= AC + AB$

↓ dual

$= (A+C)(A+B)$

Self - Dual :-

→ In self dual expression if one time dual is used results in same expression.

eg :- $AB + BC + AC$

↓ dual

$(A+B)(B+C)(A+C)$

$= (AC + B)(A+C) = AC + AC + AB + BC$

$= AB + BC + AC$

→ $\bar{A}\bar{B} + \bar{B}\bar{C} + \bar{A}\bar{C}$ → self dual

→ With n- variables max. possible self dual expression are $2^{2^{n-1}}$

$n \xrightarrow{\text{self dual}} 2^{2^{n-1}}$

→ $n = 1 \xrightarrow[\text{dual}]{\text{self}} 2^{2^0} = 2$

$A \xrightarrow{\text{dual}} A$

$A \cdot 1 \xrightarrow{\text{dual}} A + 0 = A$

$\bar{A} \xrightarrow{\text{dual}} \bar{A}$

Complement :-

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

$$\bar{Y} = ?$$

(i) AND ⟷ OR (Complement)

(ii) $\cdot$ ⟷ $+$

(iii) 1 ⟷ 0

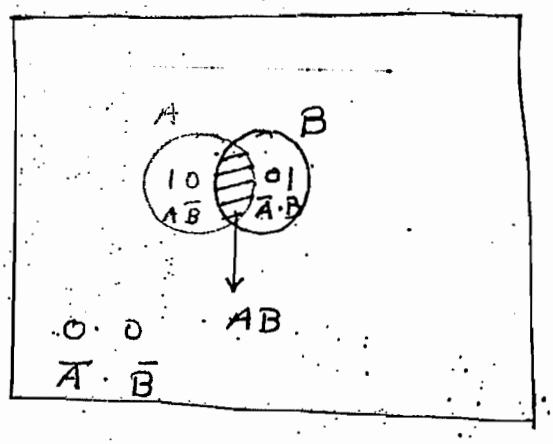
(iii) Complement each variable

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

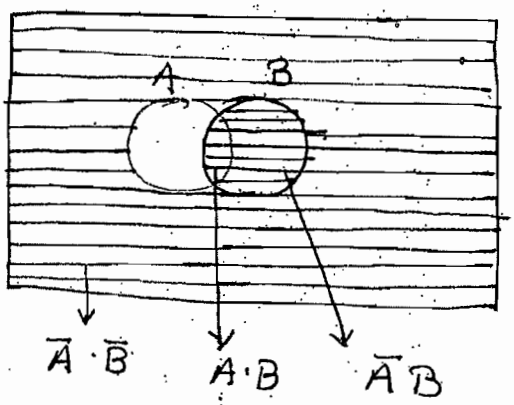
Venn - Diagram :-

$\cap$ → AND

$\cup$ → OR



Ques :- For the given venn-diagram minimize the SOP expression for ~~the~~ the area shaded in figure

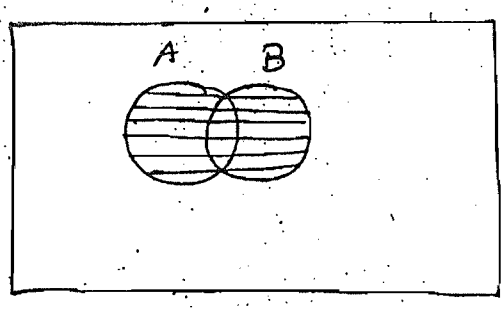


Soln :-

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

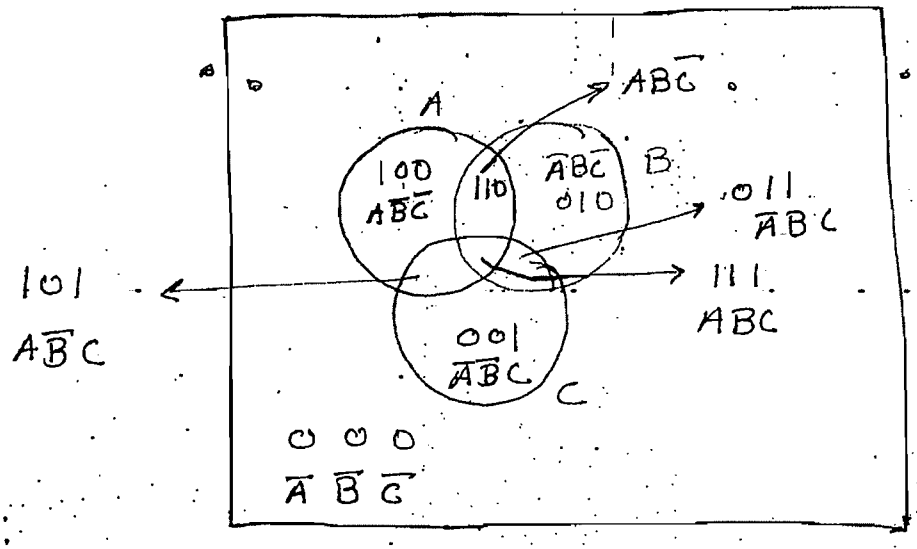
$$= \bar{A}\cdot\bar{B} + B \quad = \bar{A} + B$$

Ques:- For the given venn diagram minimize the SOP expression for the area shaded in figure

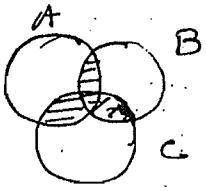


Soln:- $A\bar{B} + AB + \bar{A}B = A + \bar{A}B = A+B$

For 3-Variable System:-



Ques:- For the given venn-diagram minimize the SOP expression for shaded area



Soln:- $A\bar{B}\bar{C} + ABC + A\bar{B}C + \bar{A}BC$

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

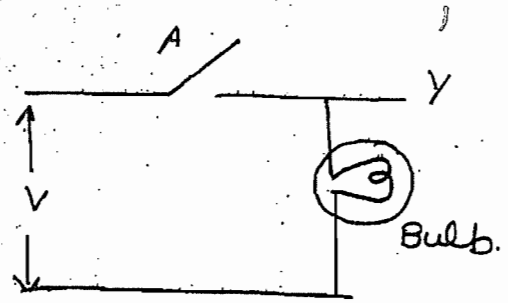
$= AB + AC + BC$

Switching Circuits :-

(a) Buffer :-

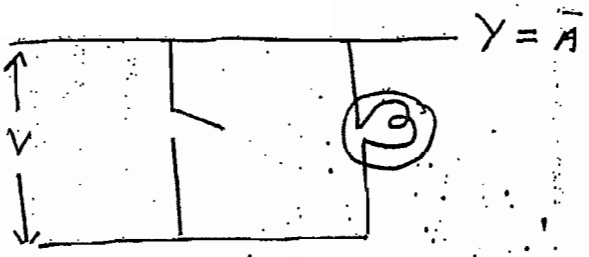
→ $Y = A$

→ Use to improve fan out

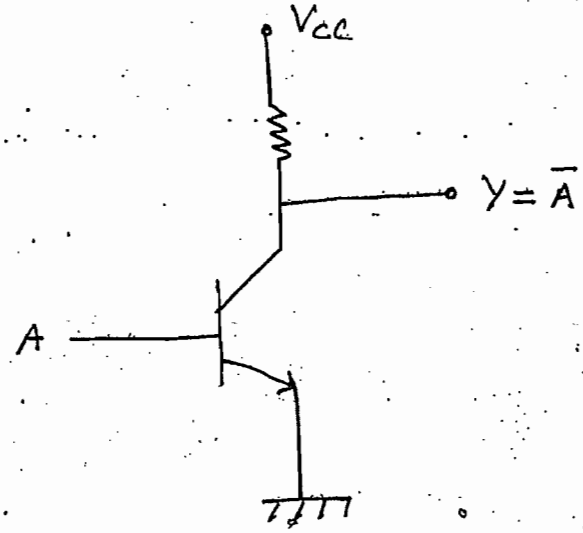


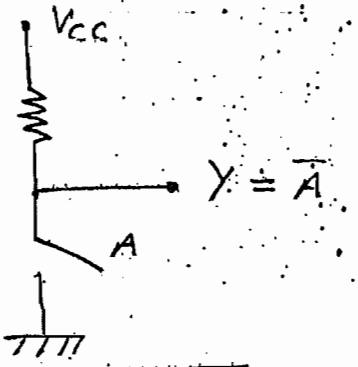
A	Y
0	0
1	1

(b) NOT :-



A	Y
0	1
1	0

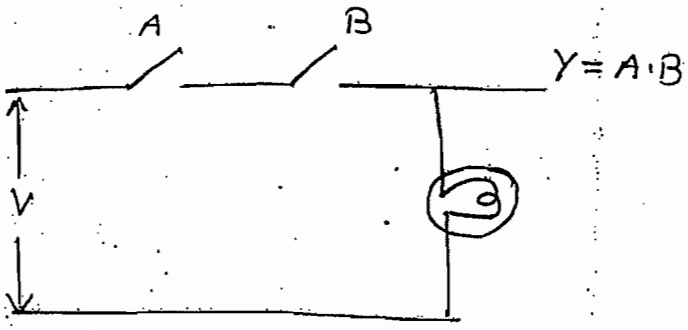




A	T_1	Y
0	OFF (cut-OFF)	1
1	ON (Sat)	0

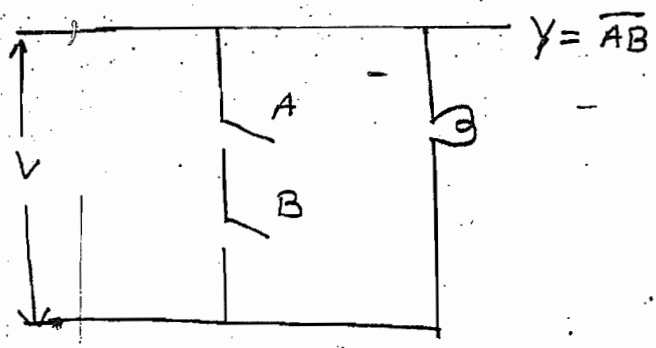
(c) AND :-

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



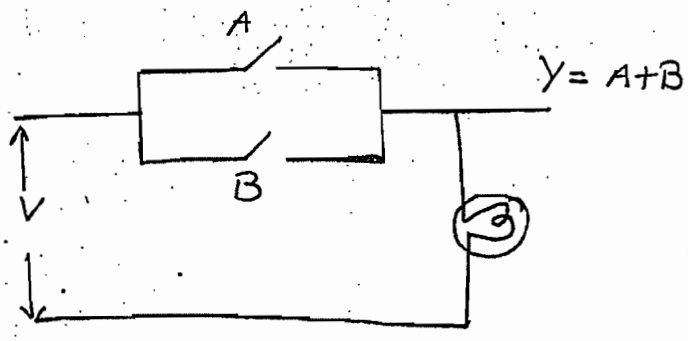
(d) NAND :-

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

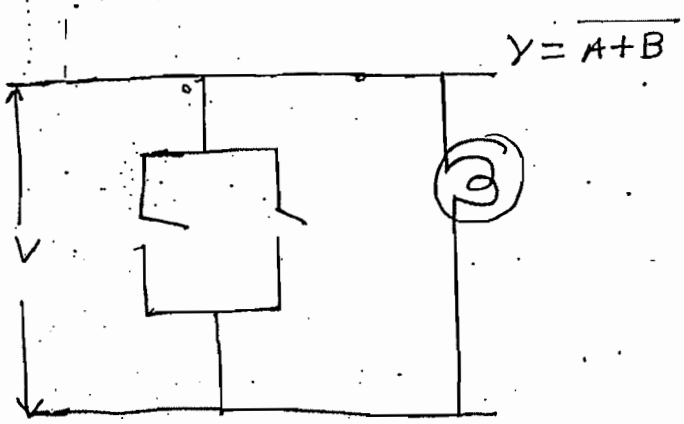


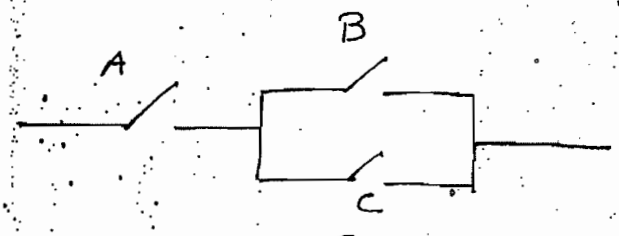
(e) OR :-

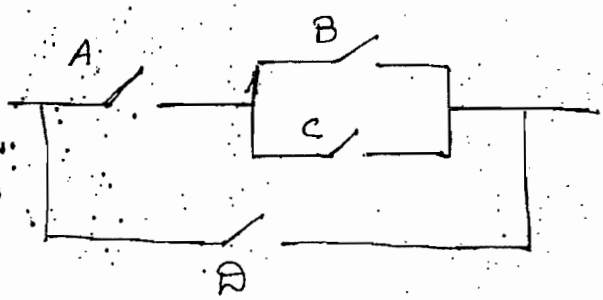
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

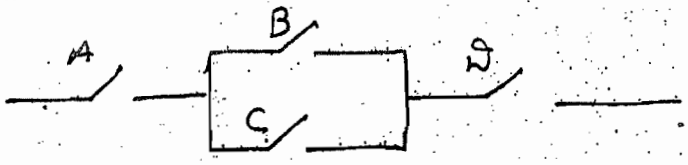


(f) NOR :-

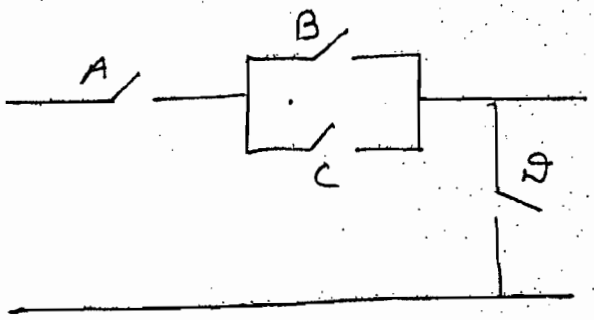




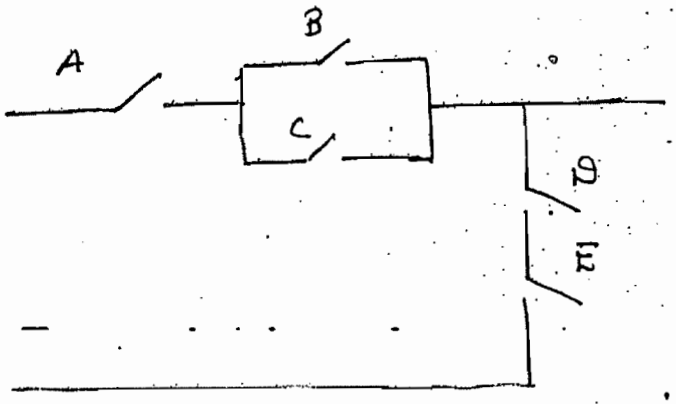




$Y = A \cdot (B+C) \cdot D$



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



$Y = A(B+C) \cdot \overline{DE}$

Statements :-

Ques :- A logic ckt have three i/p A, B, C & o/p Y Y is logic 1 for the following combination

(1) A and C are true (logic 1) →

(2) B and C are false

(3) A, B and C are true

(4) A, B and C are false

then minimize o/p Y

Soln :-

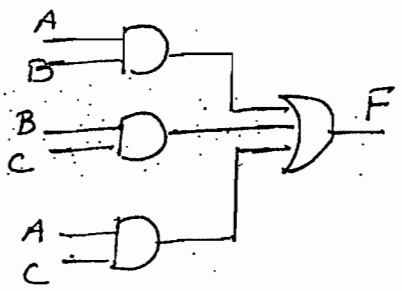
A	B	C	Y	F
0	0	0	1	0
0	0	1		0
0	1	0		0
0	1	1		1
1	0	0	1	0
1	6	1	1	1
1	1	0		1
1	1	1	1	1

$$Y = AC + \bar{B}\bar{C} + AB\bar{C} + \bar{A}\bar{B}\bar{C}$$
$$= AC(1+B) + \bar{B}\bar{C}(1+\bar{A})$$
$$= AC + \bar{B}\bar{C}$$

Ques:- A logic ckt has 3 i/ps & o/p F. o/p is logic 1 when majority no. of i/ps are logic 1 the minimise exp. for o/p F is

Soln:-
$$F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$
$$= (\bar{A}B + A\bar{B})C + AB$$
$$= \bar{A}BC + A\bar{B}C + AB$$
$$= BC + AC + AB$$

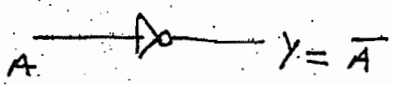
Alternatively

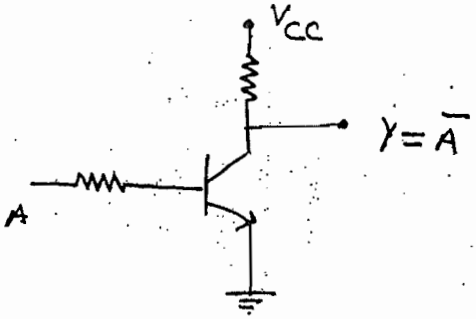


LOGIC GATES

- NOT, AND, OR } Basic gates
- NAND, NOR } Universal gate
- EXOR, EXNOR → Used in.
  - (I) Arithmatic circuits
  - (II) Comparators
  - (III) Parity generators / checker
  - (IV) Code converter

NOT Gate :-



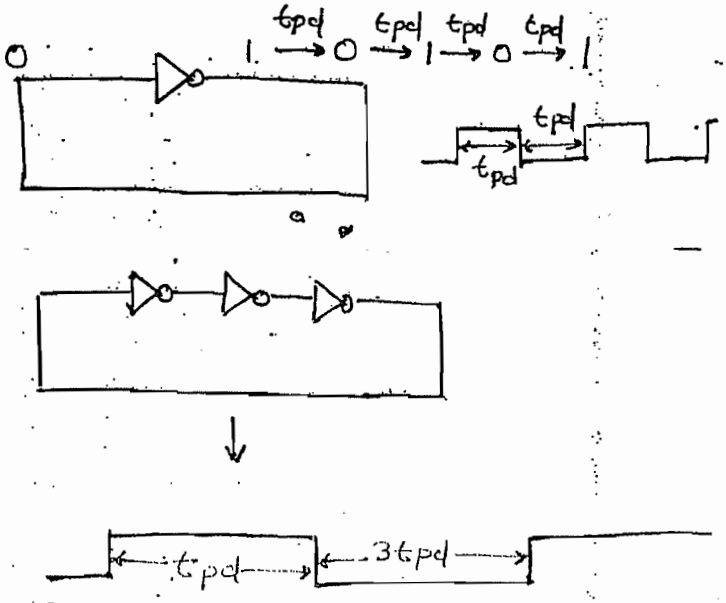


A	Y
0	1
1	0

→ Basic Memory ckts.

Bistable Multivibrator (Two stable states (0,1)
(even no. of not gate)

→ AStable Multivibrator (odd no. of not gate)
→ Free Running ckt.
→ Square Wave generator
→ Clock generator
→ Ring Oscillator



→ $\boxed{T = 2Nt_{pd}}$

where N = No. of gates in feedback

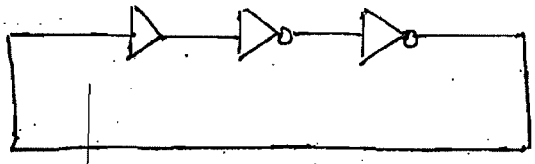
$$\boxed{f = \frac{1}{T}}$$

→ In Ring oscillator time period of generated square wave is

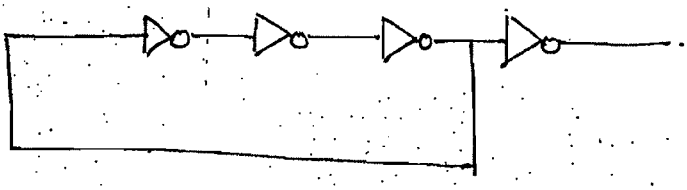
$$T = 2N t_{pd}$$

where N = no. of gates in feedback

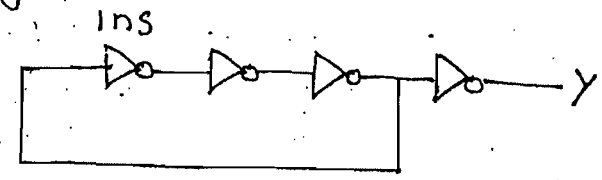
→ Biostable multivibrator



→ Astable multivibrator



Ques:- In the ckt shown in figure
Find T

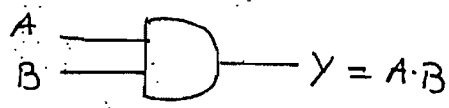


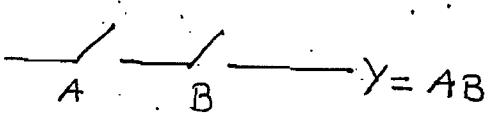
Soln:- (i) T = 6ns

→ The last not gate provide only delay and it doesn't change the time period.

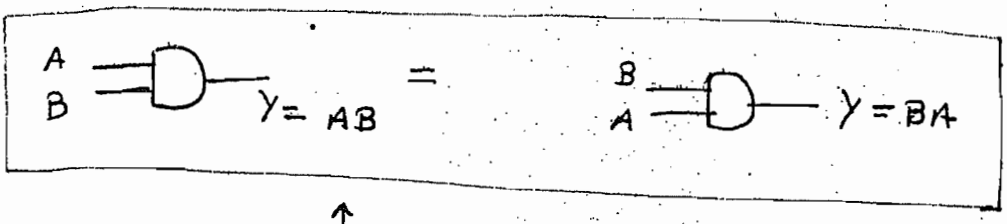
AND Gate:-

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

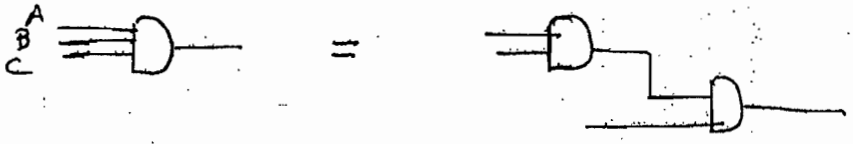




→ Series Switches

→ 

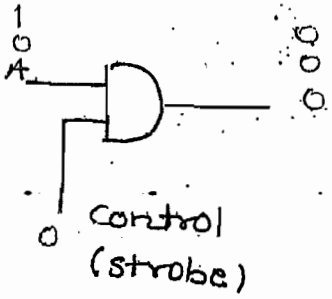
→ Follows commutative law

→ 

$ABC = (AB)C = (AC)B = (BC)A$ → Associative law

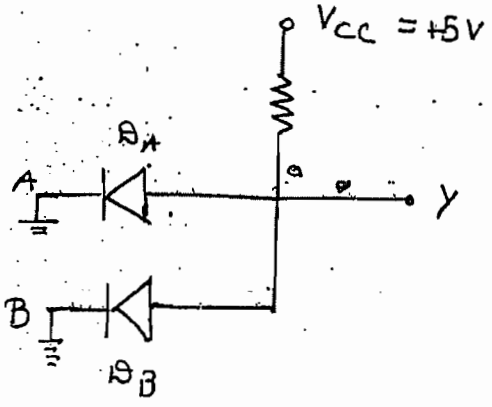
→ Follows both commutative and associative law

→
0 → Disable (o/p doesn't change)
1 → Enable (o/p change)

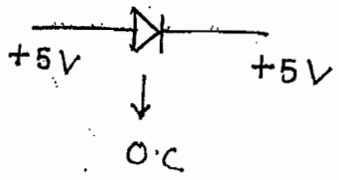


AND Gate Using Diode :-

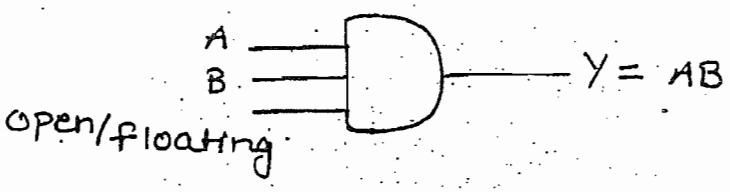
logic 0 → 0V
logic 1 → 5V



A	B	D_A	D_B	Y
0	0	ON	ON	0
0	1	ON	OFF	0
1	0	OFF	ON	0
1	1	OFF	OFF	1



→ Positive logic AND Gate

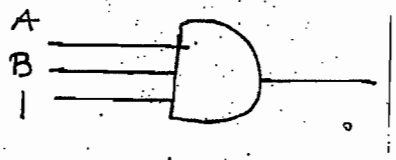


→ In TTL logic circuits, if any i/p is open or floating then it acts as logic 1

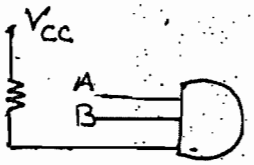
→ In ECL ~~got~~ logic circuits floating i/p acts as logic '0'

→ The unused i/p in AND gate

(i) can be connected to logic '1' or pull up

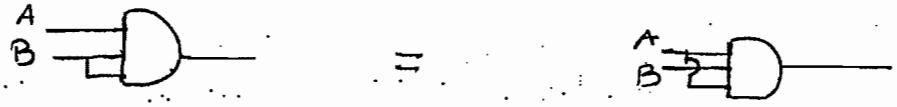


OR



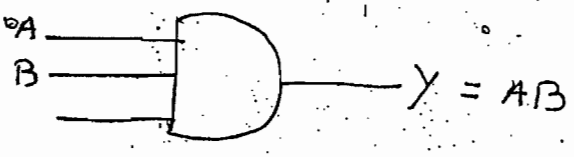
→ Passive pull up

(ii) can be connected to one of used i/p



→ fanout problem

(iii) If it is TTL gate then unused i/p can be open or float



→ Noise Margin

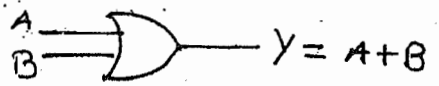
Among these best way of connecting unused i/p in AND gate is connecting to logic 1 or pull up

Note:—

(i) Symbol (ii) Logical exp. (iii) Truth Table
(iv) Switch (v) Comm/Associate (vi) Enable/Disable
(vii) Unused i/p (viii) Diode (ix) Transistor ~~(x) Diode~~
(x) Alternative Symbol (xi) Min. no. of NAND/NOR/MUX

OR Gate :-

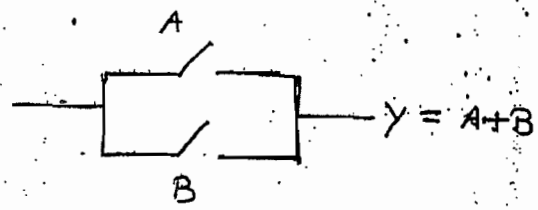
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



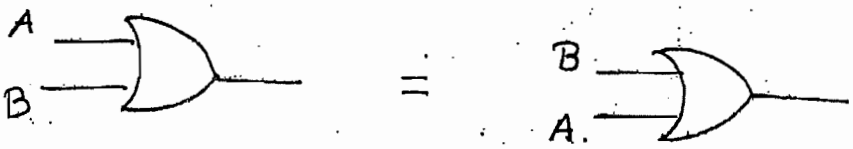
→ Enable → 0

→ Disable → 1

→ Parallel switches

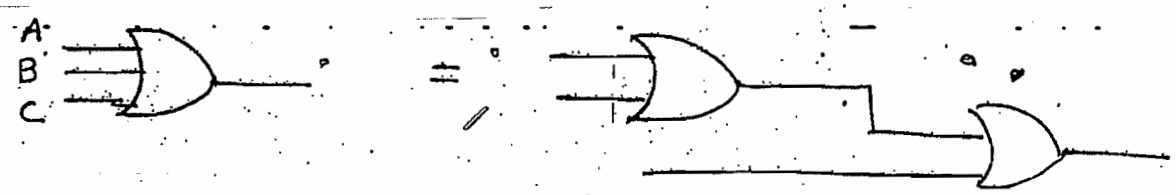


→ Follows commutative law



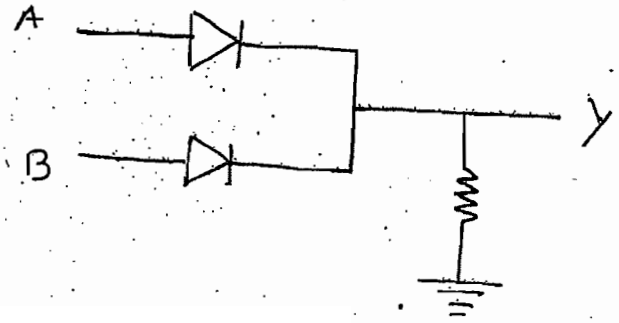
A+B = B+A

→ Follows associative law

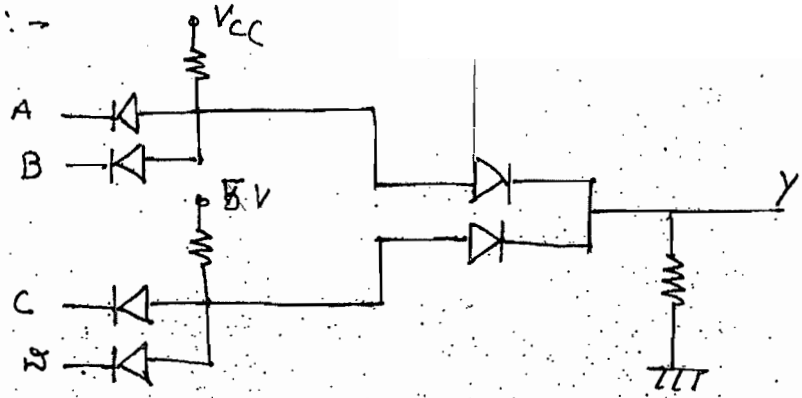


→ Using Diode Ckt :-

A	B	y
0	0	0
0	1	1
1	0	1
1	1	1



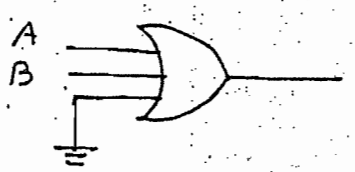
Ques :-



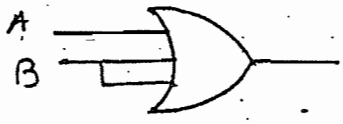
Ans :- $Y = AB + C\bar{D}$

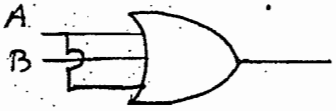
→ The unused i/p in OR gate

(i) can be connected to logic '0' or pull down



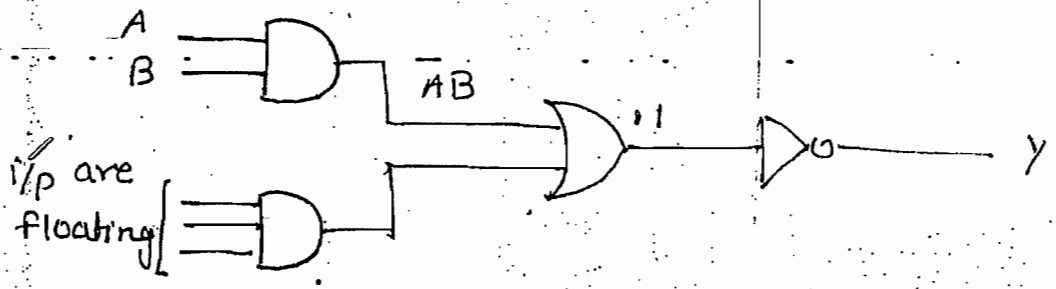
(ii) Can be connected to one of used i/p





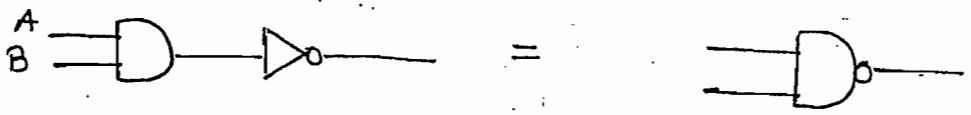
(iii) If it is ECL gate then unused i/p can be open or float

Ques:- The ckt shown in fig. is TTL AND-OR-Inverter (AOI) for given i/p, o/p = ?

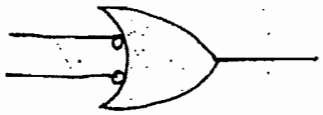


Ans:- $Y = 0$

NAND Gate:-

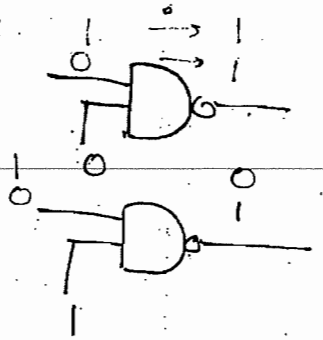


NAND → Bubbled OR =

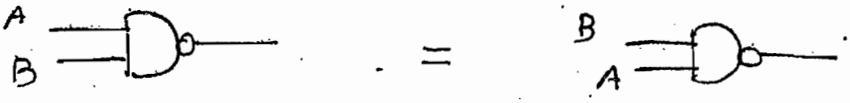


→ Enable → 1

AND	→ 0	→ disable
NAND	→ 1	→ enable

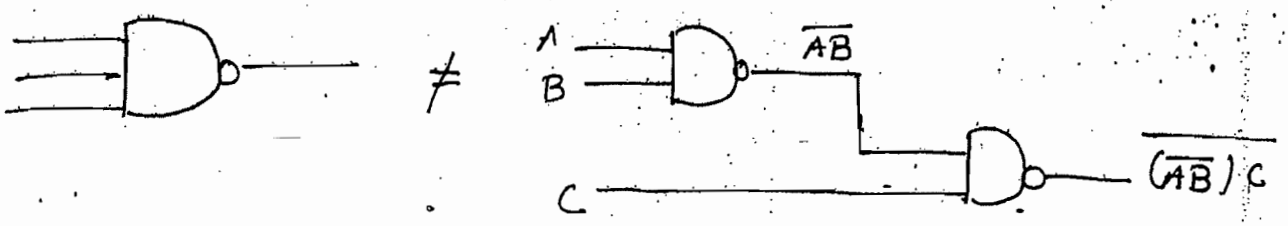


→ Follows commutative laws



$\overline{AB} = \overline{BA}$

→ Doesn't follows associative laws

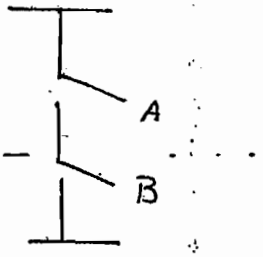


$= AB + \overline{C}$

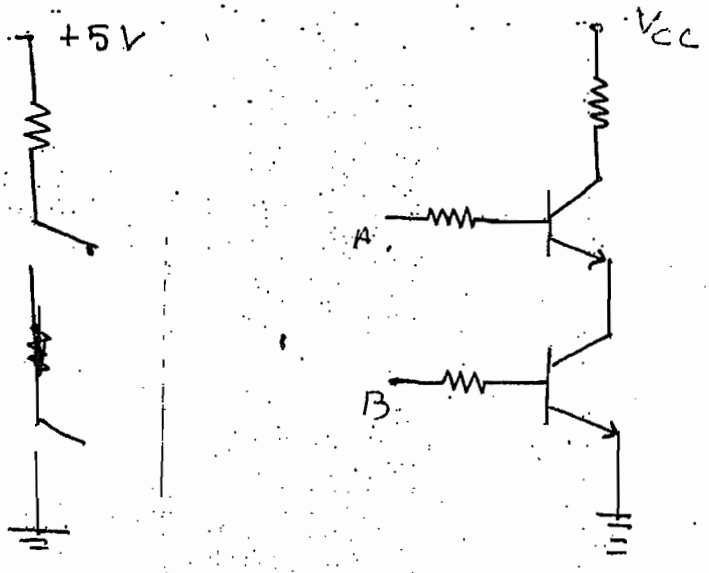
→ Truth Table :-

A	B	y
0	0	1
0	1	1
1	0	1
1	1	0

→ Series switches in parallel

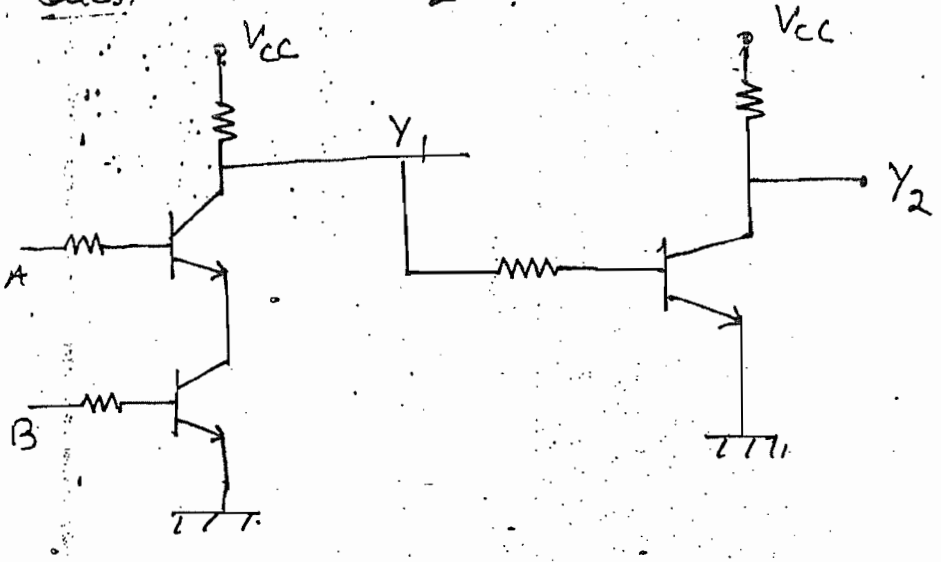


→ Using Diode Ckt :-



→ Unused i/p in NAND gate can be connected similar to unused i/p in AND gate

Ques:- $Y_1 = ?$



Ans:- $Y = AB$

NOR Gate :-

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

Alternative symbol

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

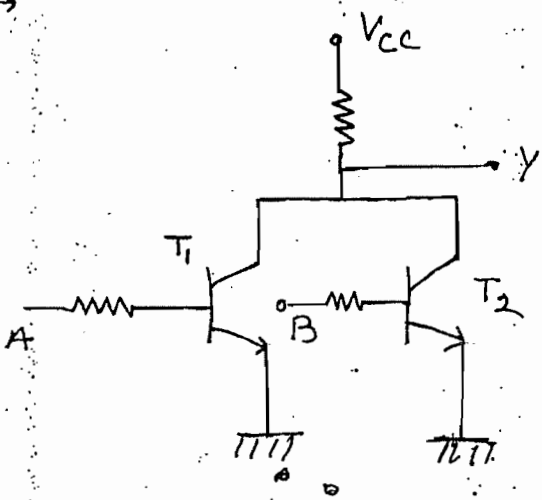
OR
NOR

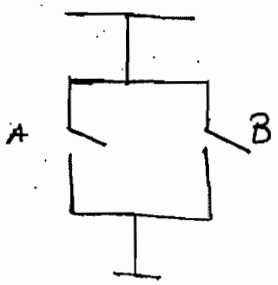
0 → Enable

1 → disable

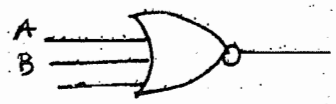
→ Follows commutative law but doesn't follow associative law

→



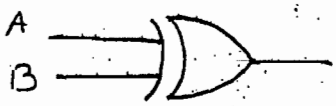


→ Except EXOR / EXNOR, all gates are present more than two i/p. (Multiple i/p)



→ In NOR gate unused i/p can be connected similar to OR gate

Exclusive OR → EXOR → XOR :-



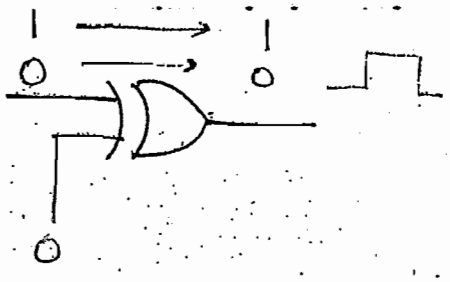
$Y = A \oplus B$ → $\bar{A}B + A\bar{B}$

→ $(A+B)(\bar{A}+\bar{B})$.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

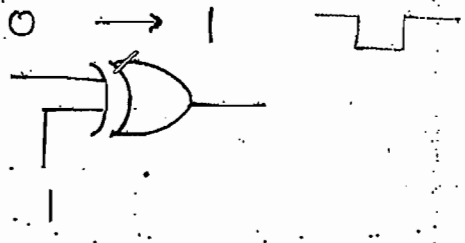
$A = B \Rightarrow 0$

$A \neq B \Rightarrow 1$



(Buffer)

A	→	$\bar{A}$
1	→	0
0	→	1



(Inverter)

→ $A \oplus A = 0$ ⇒ $A \oplus 0 = A$

→ $A \oplus \bar{A} = 1$ ⇒ $A \oplus 1 = \bar{A}$

→ Follows <u>commutative law</u>

$A \oplus B = B \oplus A$

→ Follows <u>associative law</u>

$A \oplus B \oplus C = (A \oplus B) \oplus C$

Ques:- Let $A \oplus B = C$ then $A \oplus C$, $B \oplus C$, $A \oplus B \oplus C$

Ans:- $A \oplus C = B$ * $B \oplus C = A$ $\underbrace{A \oplus B}_{C} \oplus C = 0$

→ $A \oplus A = 0$

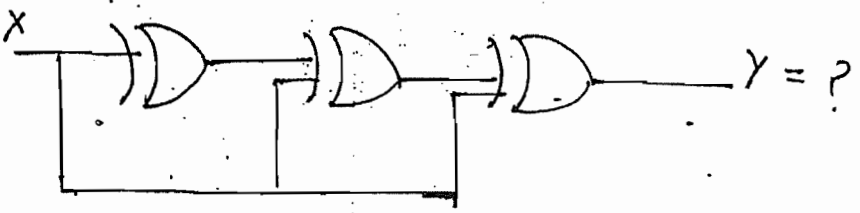
→ $A \oplus A \oplus A = A$

→ $A \oplus A \oplus A \oplus A = 0$

→ Odd no of A then result = A

→ Even no of A then result = 0

Ques:-



Ans:- $Y = 0$

Ques:- $A \oplus B \oplus AB = ?$

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

→ In EX-OR gate O/p is logic '1' when no. of one's in the I/p is odd. Hence it is known as odd no. of 1's detector ckt.

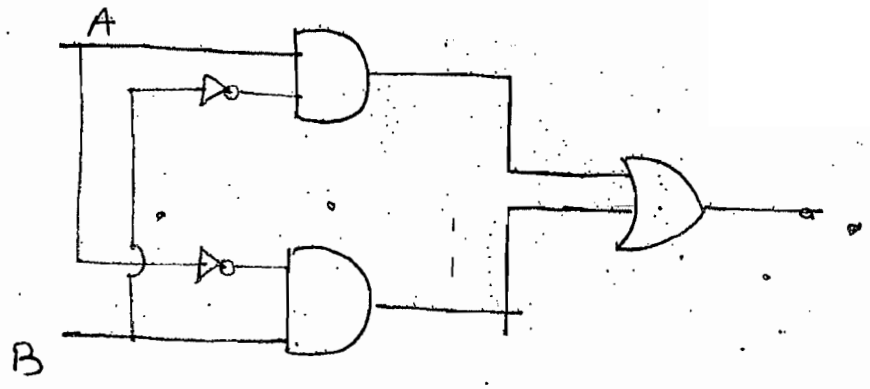
A	B	C	$Y = A \oplus B \oplus C$
0	0	0	0
→ 0	0	1	1
→ 0	1	0	1
0	1	1	0
→ 1	0	0	1
1	0	1	0
1	1	0	0
→ 1	1	1	1

$Y = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= \Sigma m(1,2,4,7)$

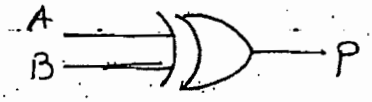
→ If n-variable XOR gate contains no. of min terms or max terms = 2^{n-1}

Internal Diagram :—



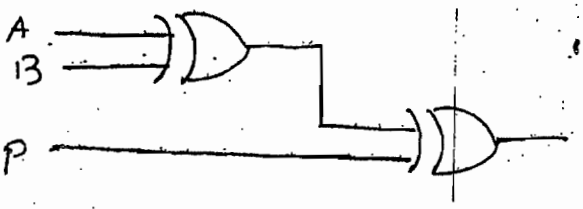
→ EXOR / ENOR gates are available only with two i/p

→ Even Parity Generator



A	B	P
0	0	0
0	1	1
1	0	1
1	1	0

→ Parity Checker (odd)

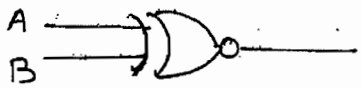


→ If all i/p are odd 1's

→ EXOR gate is known as even parity generator and odd parity checker ckt.

Parity → including i/p + Parity

EX-NOR :—



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

$Y = A \odot B$ → $\bar{A}\bar{B} + AB$

→ $(A+\bar{B})(\bar{A}+B)$

→ ∴ $A = B \Rightarrow$ → Coincidence logic ckt (i/p are same)

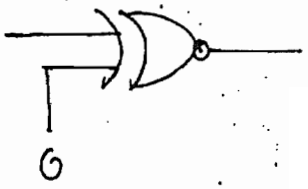
→ Equivalence logic ckt

$A \neq B \Rightarrow$ o/p = 0

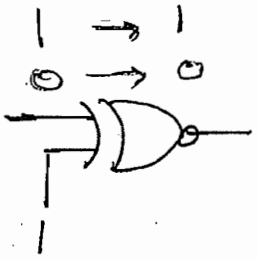
∴ A → $\bar{A}$

1 → 0

0 → 1



(Inverter)



(Buffer)

→ EX-NOR gate follows commutative law but not associative law.

$A \odot B = B \odot A$

$A \odot B \odot C \neq (A \odot B) \odot C$

→ $A \odot A = 1 \Rightarrow A \odot 1 = A$

→ $A \odot \bar{A} = 0 \Rightarrow A \odot 0 = \bar{A}$

Ques:- If $A \odot B = C$ then

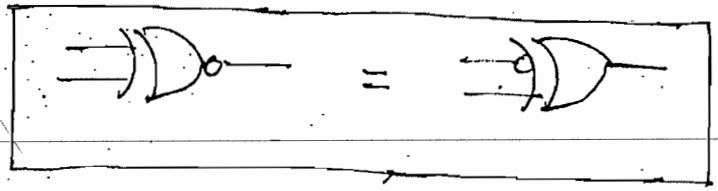
Soln:- $A \odot C = B$, $B \odot C = A$

$\boxed{A \odot B \odot C = 0}$

→ $A \oplus \bar{B} = A \odot B$

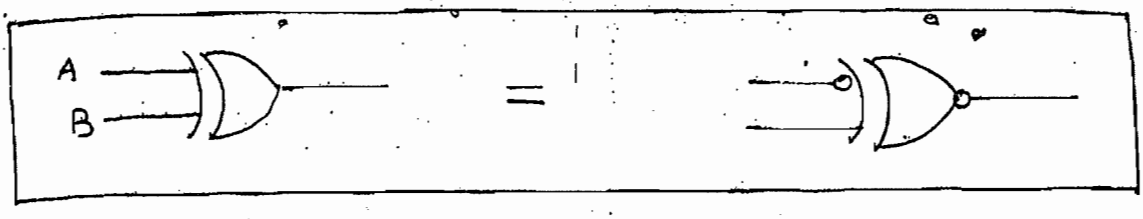
→ $\bar{A} \oplus \bar{B} = A \oplus B$

Alternate Symbol :-



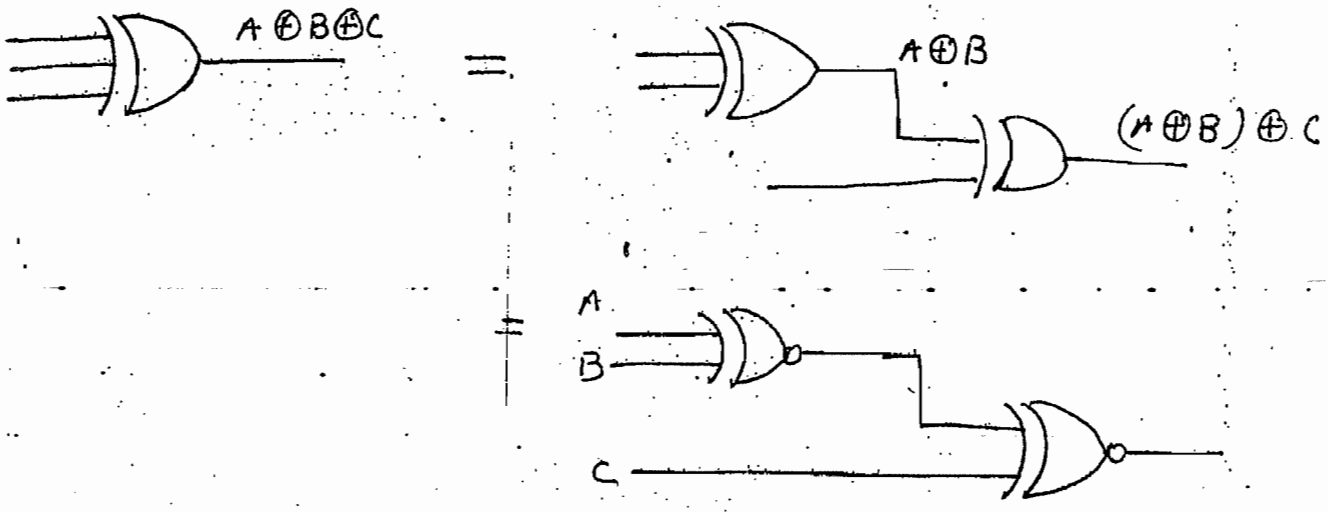
→ $A \odot \overline{B} = A \oplus B$

→ $\overline{A} \odot \overline{B} = A \odot B$

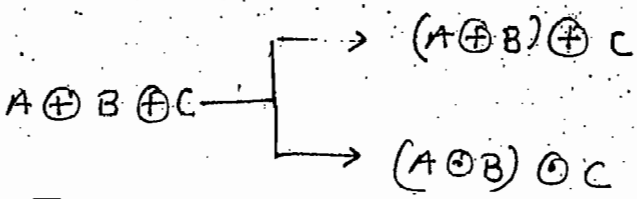


→ $A \odot B \odot C = \overline{A \oplus B \oplus C}$

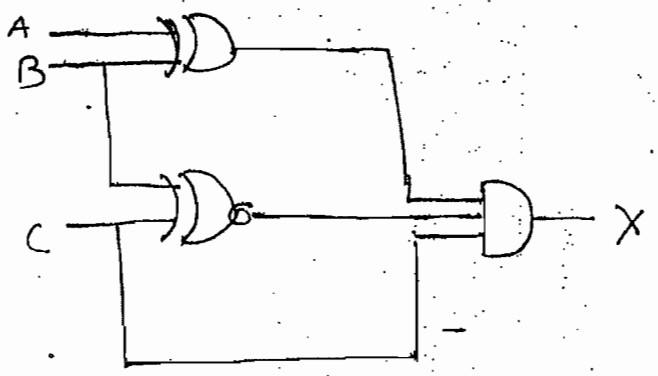
$= \overline{(A \oplus B) \oplus C} = \overline{(A \odot B) \odot C}$



→



Ques:- In the ckt shown in fig. to provide o/p X=Y i/p A, B, C respectively



(A) 101

~~(B)~~ 011

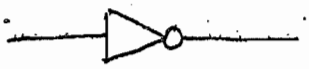
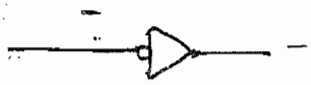
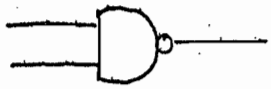
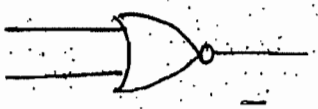
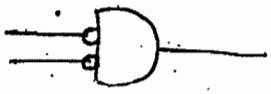
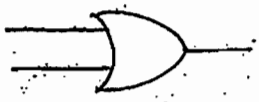
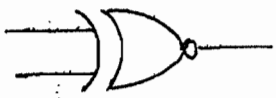
(C) 110

(D) 111

→ In XNOR gate o/p is logic '1' when no. of 1's in the i/p is even no. of 1's.

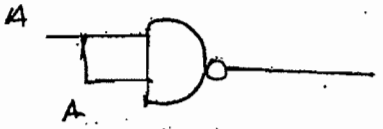
A	B	C	$Y = A \odot B \odot C$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

A	B	C	$Y = \overline{(A \odot B) \odot C}$
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
1	1	1	1

	Logic Gate	Alternative Symbol
1.		
2.		
3.		
4.		
5.		
6.		
7.		

NAND as Universal gate :-

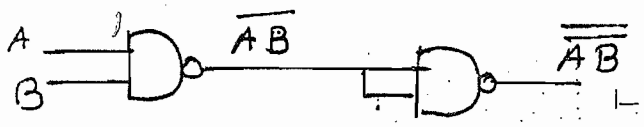
(1) NOT Gate :-



$\overline{A \cdot A} = \overline{A}$

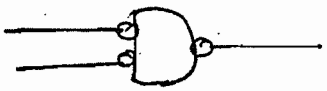
→ 1 NAND

(II) AND :—

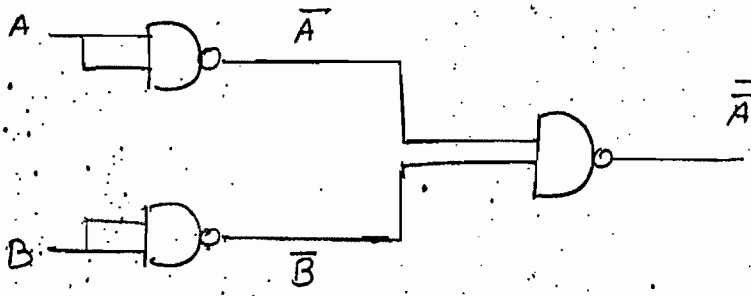


$= AB$ → 2 NAND

(III) OR :—



→ 3 NAND



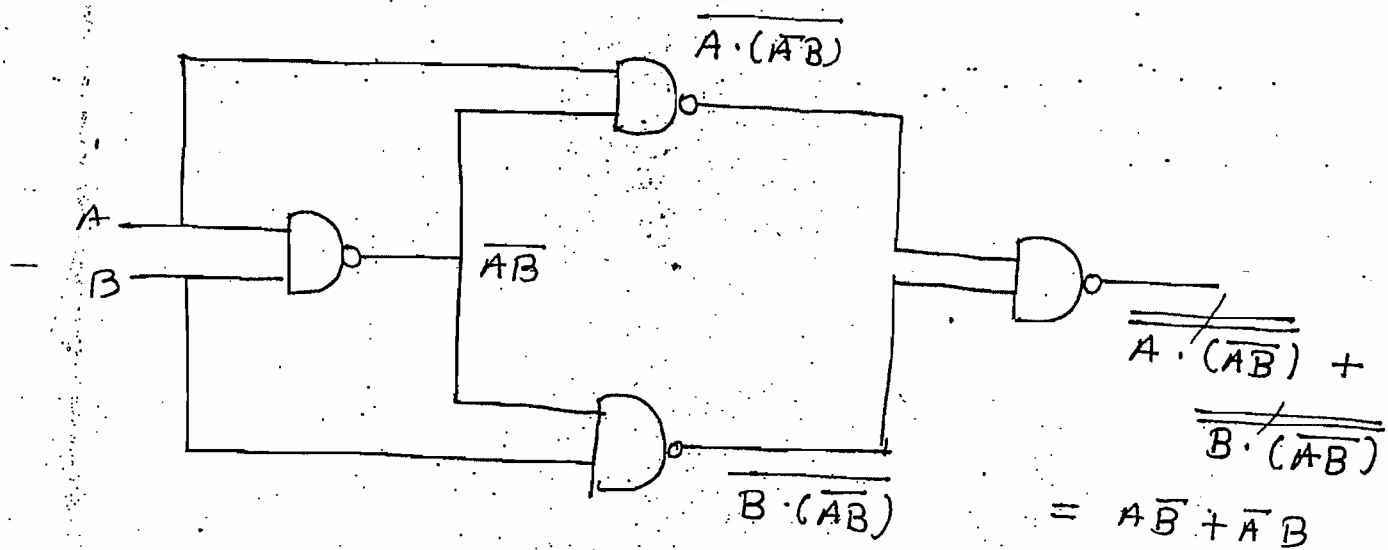
$\overline{\overline{A}\,\overline{B}} = \overline{\overline{A}} + \overline{\overline{B}}$

$= A + B$

(IV) NOR :—

→ 4 NAND

(V) EX-OR :—

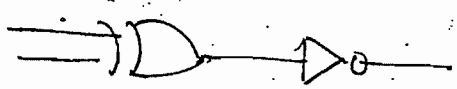


$= A\overline{B} + \overline{A}B$

$= A \oplus B$

→ 4 NAND

(VI) EX-NOR :—

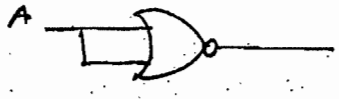


4 NAND + 1 NAND

= 5 NAND

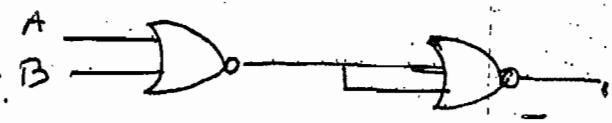
NOR as Universal :—

(i) NOT Gate :—



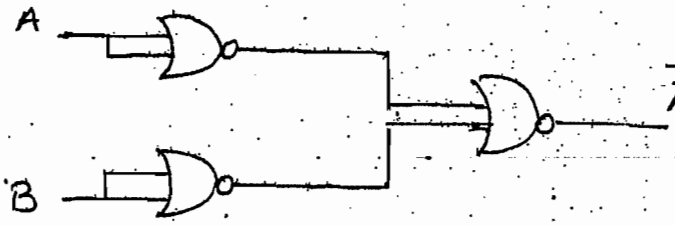
→ 1 NOR

(ii) OR Gate :—



→ 2 NOR

(iii) AND :—

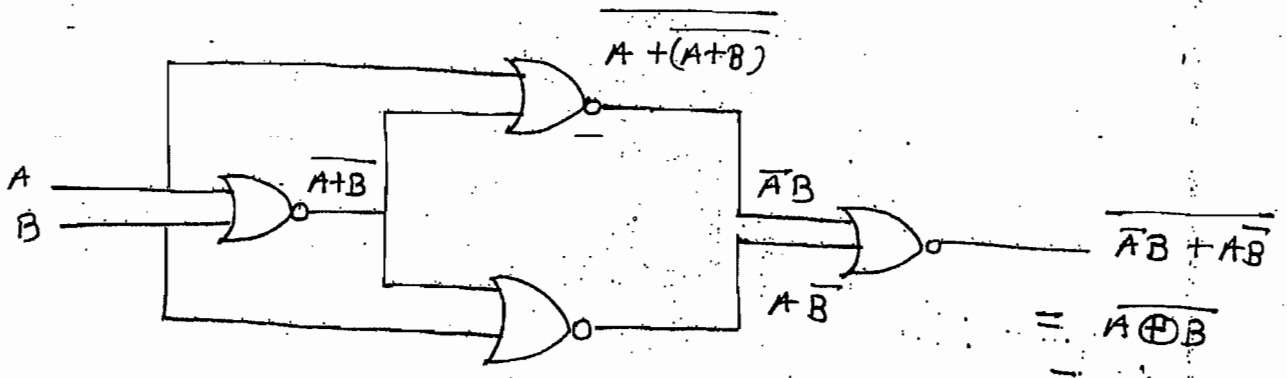


→ 3 NOR

(iv) NAND :—

→ 4 NOR

(v) EXNOR :—



→ 4 NOR

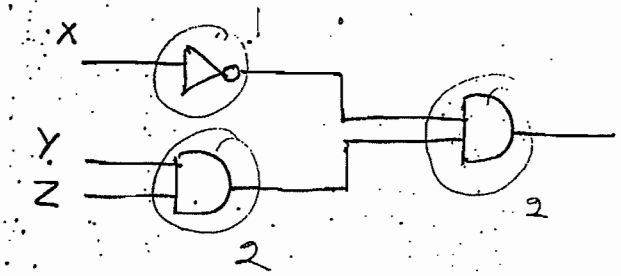
(vi) EX-OR :—

→ 5 NOR.

Logic Gates	No. of NAND	No. of NOR
NOT	1	1
AND	2	3
OR	3	2
EXOR	4	5
EXNOR	5	4
NOR	4	1
NAND	1	4

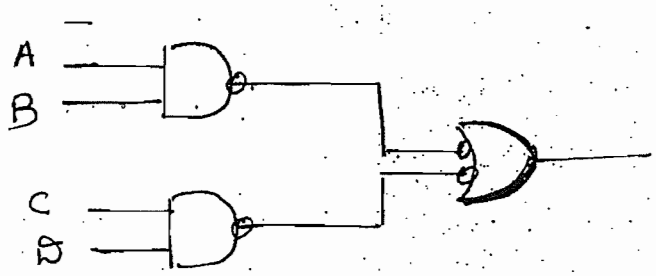
Ques:- To implement $\bar{X}YZ$. Min. no. of two i/p NAND gates used (A) 3 (B) 4 (C) 5 (D) 7

Soln:-



Ques:- To implement $AB + C\bar{D}$. Min. no. of NAND

Ans.



→ Non-degenerative

→ 3 NAND

Note:-

→ Two level AND-OR = Two level NAND-NAND

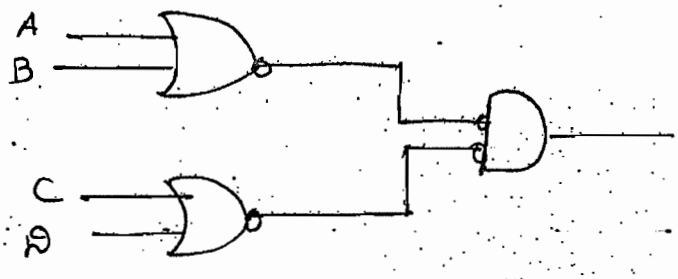
Note:-

→ NAND-NAND is preferred over AND-OR

→ SOP exp. can be implemented with min. no. of NAND gate

Degenerative :- → level changed but gate doesn't changed

Ques:- To implement $(A+B)(C+\bar{D})$. Min. no. of NOR Gate



→ 3NOR

Two level OR-AND = two level NOR-NOR

POS → Min. no. of NOR

Ques:- To implement $A+BC$. Min. no. of NAND & NOR

Ans:- NAND → 3 & NOR - 3

Ques:- To implement $(\bar{X}+\bar{Y})(W+Z)$. Min no. of two i/p NAND

(A) 3 (B) 4 (C) 5 (D) 6

Number System :—

(I) Conversion (II) Arithmetic operation

(III) Compliments.

Codes :—

→ Gray Code → Excess 3 code

→ BCD code → 2421 Code

Data representation :—

↓ +ve, -ve

(I) Sign magnitude (II) 1's compliment

(III) 2's compliment

→ Conversion :—

→ A number system with base r will have 'r' different digit and they are from 0 to (r-1)

Number System	Different digit
Binary (2)	0, 1
Octal (8)	0, 1, 2 - - - 7
decimal (10)	0, 1, 2 - - - 9
Hexadecimal (16)	0, 1, - - - 9, A, B, C, D, E, F
4	0, 1, 2, 3
6	0, 1, - - - 5
12	0, 1, - - - 9, A, B

→ Decimal to other

$(\quad)_{10}$ —— $(\quad)_r$

$(X_2 X_1 X_0 \cdot Y_1 Y_2)_{10}$ ——— $(\quad)_r$

r	$X_2 X_1 X_0$

$(Y_1 Y_2) \times 10$

→ To convert decimal to any base r, divide integer part r and multiply fraction part with base r

eg:- $(19.125)_{10}$ ——— $(\quad)_2$

2	19
2	9 — 1
2	4 — 1
2	2 — 0
2	1 — 0
	0 — 1

$(19)_{10} \longrightarrow (10011)_2$

$0.125 \times 2 = 0.25 \longrightarrow 0$

$0.25 \times 2 = 0.50 \longrightarrow 0$

$0.50 \times 2 = 1.0 \longrightarrow 1$

$(19.125)_{10}$ —— $(10011.001)_2$

→ $(27.625)_{10}$ ——— $(\quad)_8$

8	27
8	3 — 3
	0 — 3

$(27)_{10}$ — ↑$(33)_8$↓

$0.625 \times 8 = 5.000$

→ $(27.625)_{10}$ ——— $(\quad)_{16}$

16	27
16	1 — B
	0 — 1

$0.625 \times 16 = 10.000$

$= A$

$(1B.A)_{16}$

→ $(127.5)_{10}$ ——— $(\quad)_{16}$

16	127
16	7 – F
	0 – 7

$0.5 \times 16 = 8.00$

$(7F.8)_{16}$

→ $(27)_{10}$ ——— $(\quad)_4$

4	27
4	6 – 3
4	1 – 2
	0 – 1

$(1\ 2\ 3)_4$

(2) Others to Decimal :—

$(X_2 X_1 X_0 \cdot X_{-1} X_{-2})_r$ ——— $(\quad)_{10}$

$r^2\ r^1\ r^0\ r^{-1}\ r^{-2}$

$X_2 r^2 + X_1 r^1 + X_0 r^0 + X_{-1} r^{-1} + X_{-2} r^{-2} + \ldots$

→ $(10111.11)_2$ ——— $(\quad)_2$

$1 \times 16 + 0 + 1 \times 2^2 + 1 \times 2^1 + 1 + 1 \times 2^{-1} + 1 \times 2^{-2}$

$16 + 4 + 2 + 1 + \frac{1}{2} + \frac{1}{4}$

$(23.75)_2$

→ $(75.4)_8$ ——— $(61.5)_{10}$

→ $(BA9)_{16}$ ——— $(2989)_{10}$

$11 \times 256 + 10 \times 16 + 13 \times 1$

Ques:-

$$\begin{pmatrix} ECE \\ CAB \\ DAB \\ FACE \end{pmatrix}_{16} \longrightarrow (\quad)_{10}$$

(III) Octal to Binary :-

$8 \longrightarrow 2^3$

each octal digit $\longrightarrow$ 3 bit binary

0	$\longrightarrow$ 000
1	$\longrightarrow$ 001
2	
3	
4	
5	
6	
7	$\longrightarrow$ 111

$(275.36)_8 \longrightarrow (\quad)_2$

010 111 101 . 011 110

(IV) Binary to octal :-

→ Group of 3bit into one digit in octal

$(275.36)_8 \longrightarrow (\quad)_2$

010 111 101 . 011 110

$(133.54)_8$

(V). Hex to Binary :—

16 ⟶ 24

0 ⟶ 0000

⋮

F ⟶ 1111

Ques:- No. of 1's present in binary form of $(CAD)_{16}$.

Soln:-

C A D

↓ ↓ ↓

1100 1010 1101

(VI) Binary to Hex :—

→ Grp of 4 bits ⇒ one hex digit

→ 0001101011101000

$(1 \cdot A \cdot E8)_{16}$

→ $(1.1)_2$ —— $(1.8)_{16}$

0001.1000 —— $(1.8)_{16}$

(VII) Hex to Octal :— $(DAD)_{16}$ —— ( )$_8$

Binary

D A D

↓ ↓ ↓

1101 1100 1101 ⟶ 6655

Arithmatic Operation on number system :-

(I) Binary System (Add, sub, Mult)

(II) Octal system (Add, sub)

(III) Hex (Add, sub)

(IV) r (base) (Add, sub)

Binary Addition :-

$0+0 = 0$

$0+1 = 1$

$1+1 = 10$

$1+1+1 = 11$

$$\begin{array}{r} 111001 \\ 10110 \\ \hline 1100110 \end{array}$$

Binary subtraction :-

$$\begin{array}{r} 11011 \\ 10110 \\ \hline 00100 \end{array}$$

Binary Multiplication :-

$(1011)_2 \times (1101)_2$

$$\begin{array}{r} 1011 \\ 0000\times \\ 1011\times\times \\ 1011\times\times\times \\ \hline 10001111 \end{array}$$

Ques:- Multiply $(1111)_2 \times (1111)_2$

Soln:-

$$\begin{array}{r} 1111 \\ 1111\times \\ 1111\times\times \\ 1111\times\times\times \\ \hline 11100001 \end{array}$$

Octal System :-

0 1 2 ..... 7 | 10 11 12 13 ... 17, 21, 22 ...

↓ ↓ ↓

8 9 10 ....

$1+1 = 2$

$1+2 = 3$

$\vdots$

$1+6 = 7$

$1+7 = 10$

$7+1 = 10$

$7+2 = 11$

$7+3 = 12$

$7+7 = 16$

Add :-

$$\begin{array}{r} 2673 \\ 5462 \\ \hline 10355 \end{array}$$

8 | 13

8 | 1 - 5

0 - 1

8 | 11

8 | 1 -

Octal subtraction :-

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

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

Hexadecimal :-

0 - 9 , A, B, C, D, E, F , 10, 11, 12

Addition :-

1 + 1 = 2

1 + 2 = 3

:

1 + 8 = 9

1 + 9 = A

1 + A = B

1 + B = C

:

A + A = 14

F + F = 1E

A + A

16	20
16	1-4
	0-1

F + F

16	30
16	1-14-E
	0-1

Sum of two Hex No :-

5	6	8	9
7	5	4	3
C	B	C	C

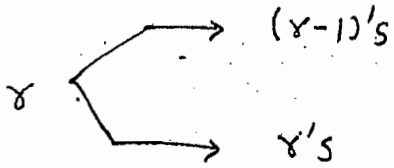
A	D	D	
D	A	D	
1	8	8	A

2	6	8	9
5	4	7	6
7	A	F	F

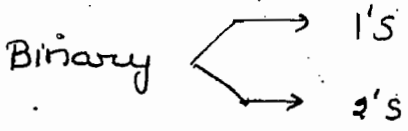
Subtraction :-

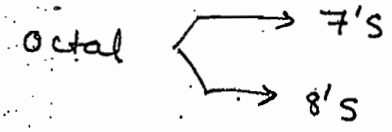
9	6	5	4
2	6	7	3
6	F	E	1

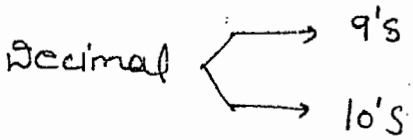
Compliments:-

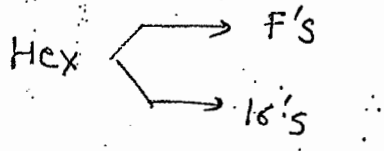


A number system with base r, having two types of compliments i.e. (r-1) compliment & r's compliment









1's Compliment:-

1 1 1 1 1 1 → Max. No.

→ 1 1 0 0 1 0

0 0 1 1 0 1

→ To determine (r-1)'s compliment, first write max. no in given base then subtract the no.

7's Compliment:-

→ Octal No = 2673

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

9's Compliment:-

→ 2689

```
9 9 9 9
2 6 8 9
-------
7 3 1 0
-------
```

F's compliment :-

→ 3579

F F F F
3 5 7 9
C A 8 6

r's compliment :-

→ To determine r's compliment, first write (r-1)'s compliment and add '1' to LSB (right most place)

2's compliment :- ← scanning upto first one and after that compliment

1 1 0 1 0 1 0

2's compliment 0 0 1 0 1 1 0

1's compliment 0 0 1 0 1 0 1

1

2's " 0 0 1 0 1 1 0

0 0 1 0 1 0 0

→ 1 1 0 1 0 1 1

0 0 1 0 1 0 0 → 1's

+ 1

1 0 1 0 1 0 1 → 2's

Ques :- Determine 8's compliment 2670

Soln :-

7 7 7 7
2 6 7 0
5 1 0 7 → 7's

8's 1

5 1 1 0 → 8's

Ques:- Determine 10's compliment of 2690

Soln:-

9's → 9 9 9 9

2 6 9 0

7 3 0 9

10's 1

7 3 1 0

Ques:- Determine 16's compliment of 2689

Soln:-

16's F F F F

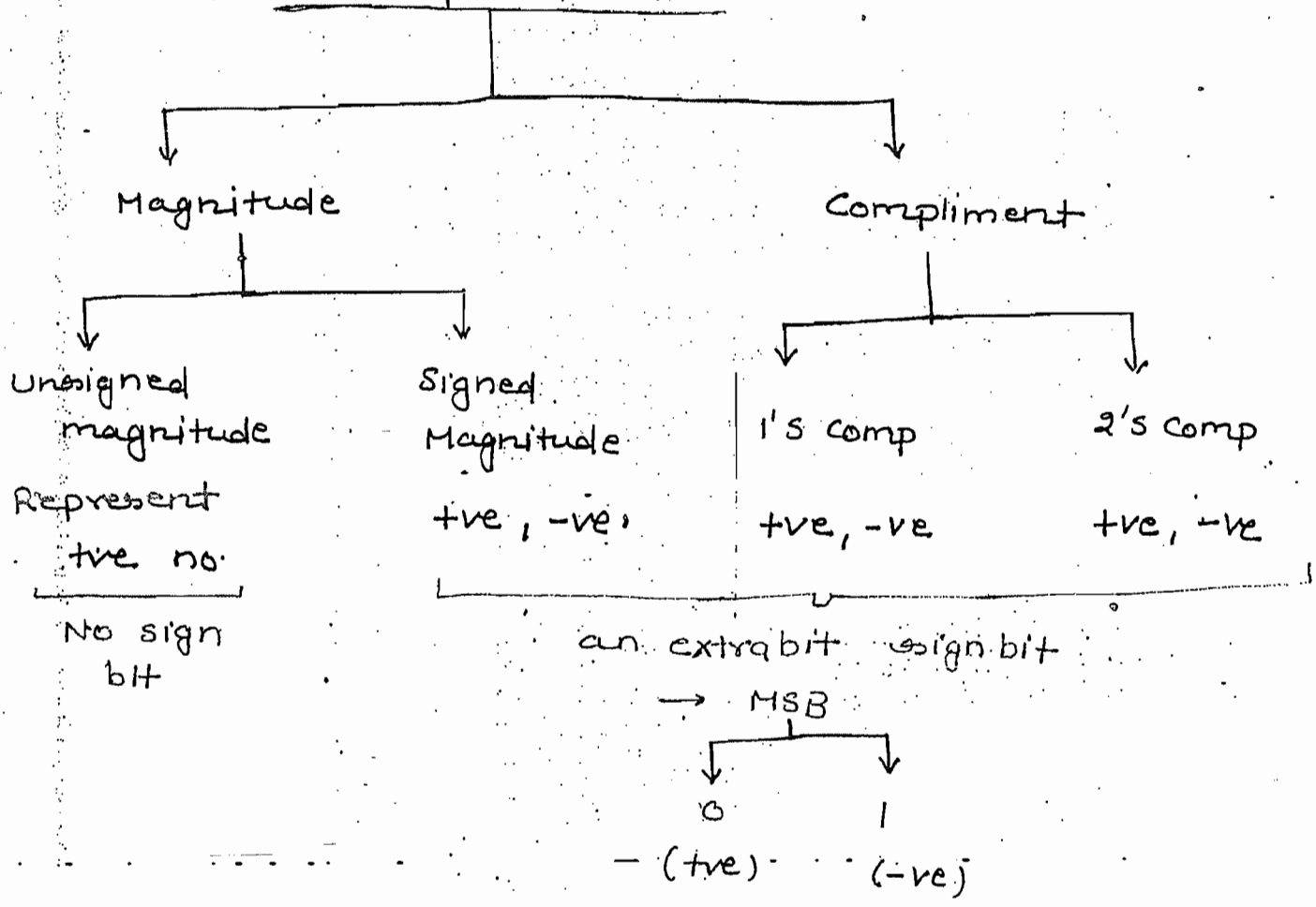
2 6 8 9

D 9 7 6

1

D 9 7 7

Data Representation



+5 in all system :-

+5	1 0 1	0 1 0 1	0 1 0 1	0 1 0 1

→ Compliments are used only for -ve No.

-5 (4bit)	X	1 1 0 1	first write +ve & then compliment 1 0 1 0	1 0 1 1

-5 (8 bit) → For signed magnitude, separate sign and mag. put zero b/w them

→ For 1's & 2's copy MSB

-5 (8 bits) $\times$ 10000101 11111010 11111011

Unsigned Magnitude :-

→ only for +ve numbers

→ 4 bits range of integers ⇒ 0 to 15

→ n bits " " " ⇒ 0 to $2^n - 1$

Signed Magnitude :-

→ For +ve, -ve numbers

→ +ve no. represented normal binary with MSB '0'

-ve " " " " '1'

+15 → 01111

-15 → 11111

Ques:- A no. is represented in signed mag. representa as 1011.11 . Its equivalent in decimal

Soln:- $-(3.75)$

Signed Magnitude For Decimal No :-

				Decimal
0	0	0	0	+0
0	0	0	1	+1
0	1	1	0	+6
0	1	1	1	+7
1	0	0	0	-0
1	0	0	1	-1
1	1	1	0	
1	1	1	0	

Range :-

→ For 4 bit No ⇒ −7 to +7

→ For n bit ⇒ $-(2^{n-1}-1)$ to $(2^{n-1}-1)$

Disadvantage :-

1. Two zeroes are present
2.

1's compliment :-

→ Both +ve & −ve no.

→ In this +ve no. is represented as normal binary with MSB → 0

→ To represent −ve no.

(i) Write +ve no.

(ii) 1's compliment it

eg:- −13

+13 → 0 1 1 0 1

1 0 0 1 0 → 1's compliment = −13

1's compliment	Decimal
0 0 0 0	+0
0 0 0 1	+1
⋮	⋮
0 1 1 0	+6
0 1 1 1	+7
1 0 0 0	−7
1 0 0 1	−6
⋮	⋮
1 1 1 0	−1
1 1 1 1	−0

→ Range of n-bit 1's compliment ⇒ $-(2^{n-1}-1)$ to $(2^{n-1}-$

Disadvantage:-

(i) Two zeros are present (+0

(ii) During addition if any carry is present, it is added to LSB to get correct result. (Same for signed mag.)

Ques:- Perform (5-4) using 1's compliment

Soln:-

+5 ⇒ 0 1 0 1

−4 ⇒ 1 0 1 1

carry ← (1) 0 0 0 0

Add in LSB 1

EAC 0 0 0 1

End around carry.

+4 → 0 1 0 0

−4 → 1 0 1 1

2's Compliment:-

→ For both +ve & −ve no.

→ +ve no. are represented as normal binary with MSB 0

→ To represent −ve no.

(i) Write +ve no.

(ii) 2's compliment it

+13 → 2's → 0 1 1 0 1

−17 → +17 ⇒ 0 1 0 0 0 1

−17 ⇒ 1 0 1 1 1 1

Ques:- A number is represented in 2's compliment fi 1011 then find equivalent decimal.

Soln:- 1 0 1 1

Take 2's compliment

−(0 1 0 1)

−5

Note:-

For -ve number = $N - 2^n$

eg:- 1 0 0 1

$9 - 16 = -7$

→ 1000 ⇒ $8 - 2^4 = -8$

2's compliment:-

2's compliment	Decimal
0 0 0 0	+0
0 0 0 1	+1
⋮	
0 1 1 0	+7
0 1 1 1	+7
1 0 0 0	-8
1 0 0 1	-7
⋮	
1 1 1 0	-2
1 1 1 1	-1

Note:-

Discared carry present in 2's compliment representation

During 1's and 2's complements representation to extent no of bit, copy MSB to required no. of bits

Range of 2's compliment:-

-2^{n-1} to $2^{n-1} - 1$

Ques:- Perform (5-4) using 2's compliment operation

Soln:-

+5 => 0 1 0 1

-4 => 1 1 0 0

1 0 0 0 1

(1) Discard

Ques:- Perform (5+4) using 2's compliment

Soln:-

+5 → 0 1 0 1

+4 → 0 1 0 0

1 0 0 1 = -7 → overflow

Ques:- Perform (-5-4) using 2's compliment operation

Soln:-

-5 => 1 0 1 1 → in 2's compliment

-4 => 1 1 0 0

1 0 1 1 1

(1) Discard

Overflow → May occur in signed arithmatic operation when two same sign no. are added and results exceeds with given no. of bits.

→ To detect overflow there are two methods

(i) By using sign bits

Let X and Y are sign bit of no. and Z is result sign bit then condition for overflow is

$\bar{X}\bar{Y}Z + XY\bar{Z}$

(ii) By using carry bit

Let Cin (carry into MSB) & Cout (carry from MSB)

Cin ⊕ Cout

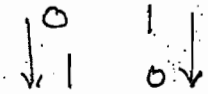
Cin, Cout → = 0 No overflow

Cin, Cout → = 1 (Overflow)

K- Map :-

→ deals with 0, 1, X

→ When two bits are change then we can't minimize



Two bits change

→ Gray code representation

→ It is a graphical method

→ In K-Map, Gray code representation is used to minimize logical expression

f (A, B) — MSB LSB

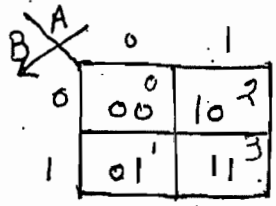
f (A, B, C) — MSB

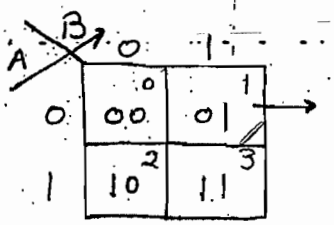
f (A, B, C, D)

Two Variable :-

f (A, B)

M L

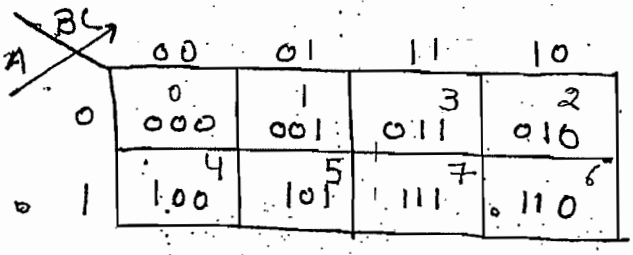




→ cell or square

Three Variable :-

f (A, B, C) =

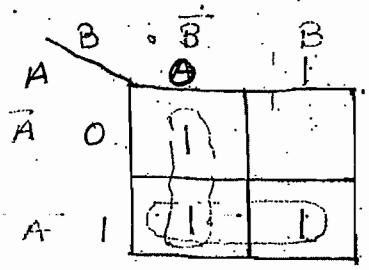


Four Variable :-

AB \ CD	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

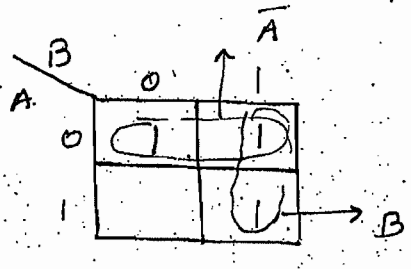
SOP :-

$$f(A,B) = \bar{A}\bar{B} + AB + A\bar{B}$$



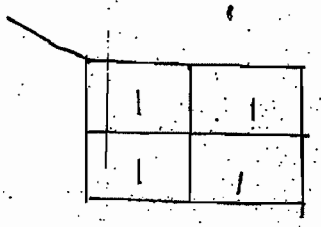
$$= A + \bar{B}$$

$$f(A,B) = \Sigma m(0,1,3)$$



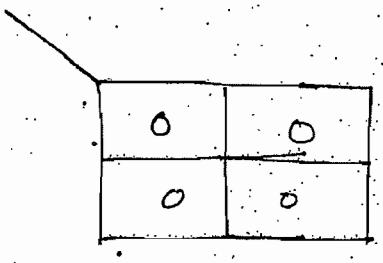
$$= \bar{A} + B$$

$$f(A,B) = \Sigma m(0,1,2,3)$$



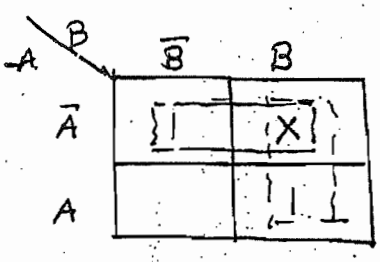
$$= 1$$

Note :-



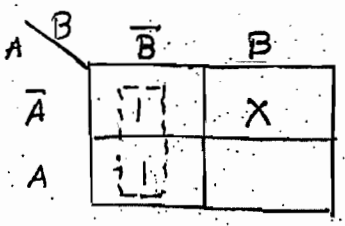
$$= 0$$

$f(A,B) = \Sigma m(0,3) + \Sigma d(1)$



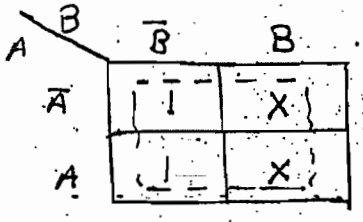
$= \bar{A} + B$

$f(A,B) = \Sigma m(0,2) + \Sigma d(1)$



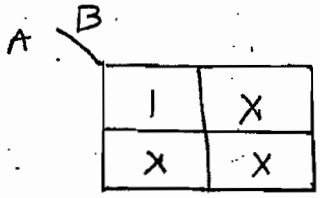
$= \bar{B}$

$f(A,B) = \Sigma m(0,2) + \Sigma d(1,3)$

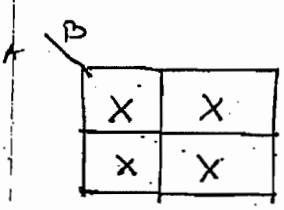


$= 1$

Note :-



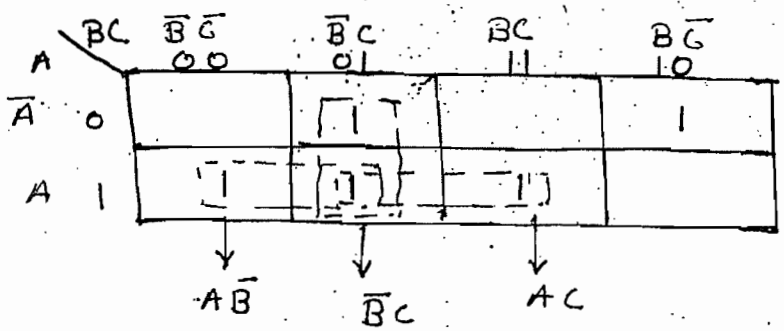
$= 1$



$= X$

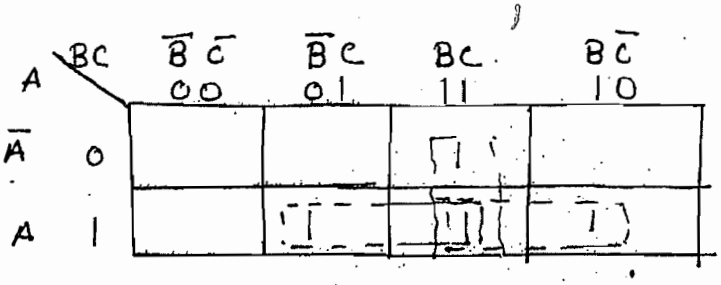
Three Variables :-

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



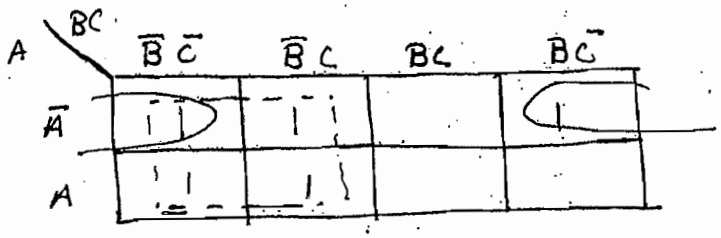
$= A\bar{B} + \bar{B}C + AC + \bar{A}B\bar{C}$

Ques:- $f(A,B,C) = \Sigma m\ (3,5,6,7)$



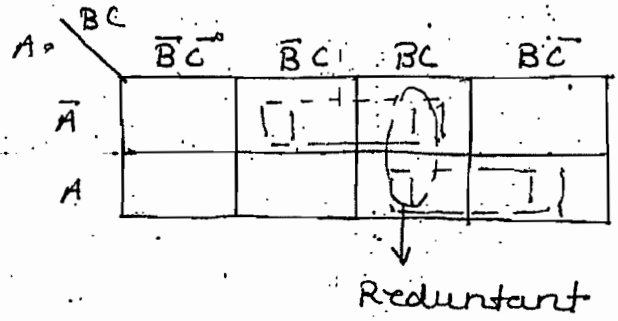
$= AC + A\bar{B} + BC$

Ques:- $f(A,B,C) = \Sigma m\ (0,1,2,4,5)$



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

Ques:- $f(A,B,C) = \Sigma m\ (1,3,6,7)$



$= \bar{A}C + AB$

Hazard ~~free~~

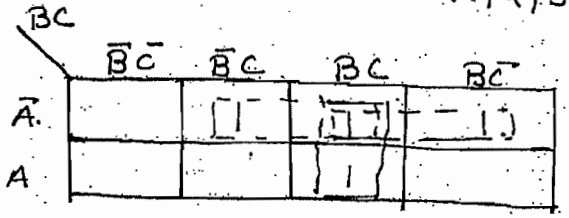
Note:-

→ In Hazard free term, redundant term is added

Minimization:-

(1) Octet ⇒ group of 8 cells ⇒ eliminates 3 variables
(2) Quad ⇒ group of 4 cells ⇒ " 2 "
(3) Pair ⇒ " " 2 " ⇒ " 1 "
(4) Single term ⇒ 1 cell ⇒ " 0 "
(5) Remove redundant term

Ques:- $f(A,B,C) = \Sigma m\ (1,2,3,7)$ Identify I, PI, EPI



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

I = 4

PI = 3 ($\bar{A}B$, $\bar{A}C$, BC)

EPI = 3

Implicant :-

→ Each minterm in the given expression is known as implicant

Prime Implicant :-

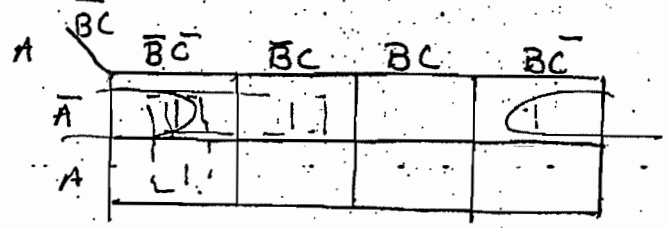
→ It is a product term obtained by combining max. possible cells

Essential Prime Implicant (EPI) :-

→ It is the prime implicant which is possible to combine only in one way & there is no alternate (It is the termed required compulsory in our Ans)

Ques :- $f(A,B,C) = \Sigma m(0,1,2,4)$ Minimize & identify I, PI & EPI

Soln :-

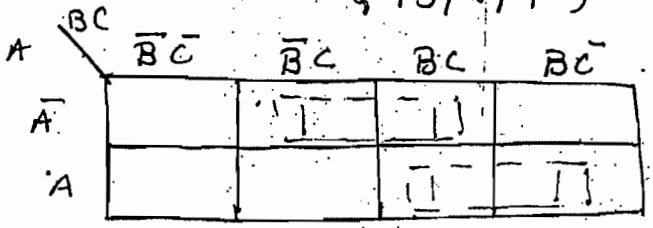


$= \bar{B}\bar{C} + \bar{A}\bar{B} + \bar{A}\bar{C}$

I = 4 , PI = 3 , EPI = 3

Ques :- $f(A,B,C) = \Sigma m(1,3,4,7)$

Soln :-

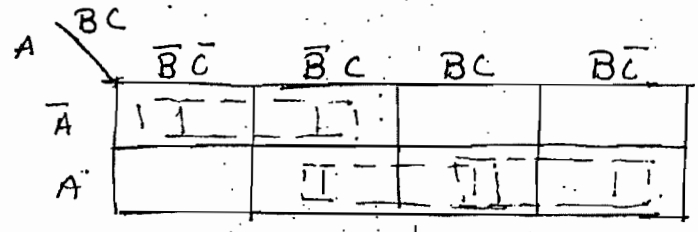


$= \bar{A}C + AB$

I = 4 , PI = 3 ($\bar{A}C$, AB, BC) , EPI = 2

Ques :- $f(A,B,C) = \Sigma m(0,1,5,6,7)$

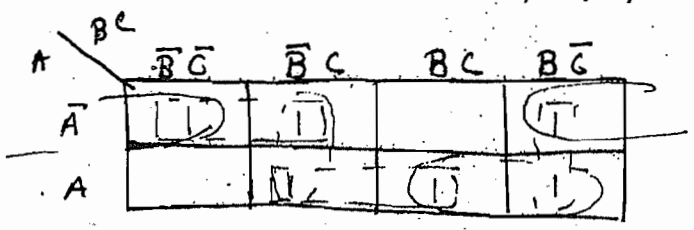
Soln :-



I = 5
PI = 4
EPI = 2

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

Ques:- $f(A,B,C) = \Sigma m(0,1,2,5,6,7)$



Another way

Soln:- $I = 6$, $PI = 6$, $EPI = 0$

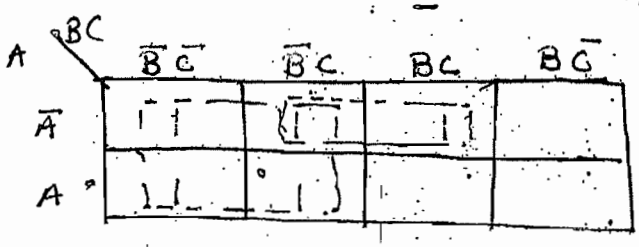
→ $\bar{A}\bar{B} + AC + B\bar{C}$

→ $\bar{B}C + AB + \bar{A}\bar{C}$

} Both are correct

Ques:- $f(A,B,C) = \Sigma m(0,1,3,4,5)$

Soln:-

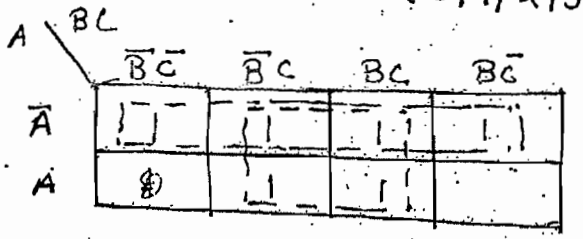


$= \bar{B} + \bar{A}C$

$I = 5$, $PI = 2$, $EPI = 2$

Ques:- $f(A,B,C) = \Sigma m(0,1,2,3,5,7)$

Soln:-

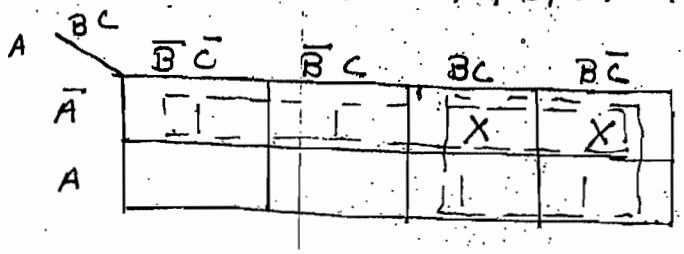


$= \bar{A} + C$

$I = 6$, $PI = 2$, $EPI = 2$

Ques:- $f(A,B,C) = \Sigma m(0,1,6,7) + \Sigma d(2,3)$

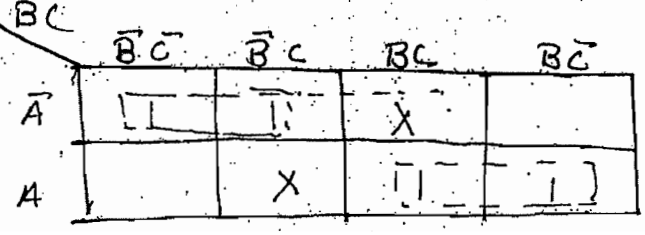
Soln:-



$= \bar{A} + B$

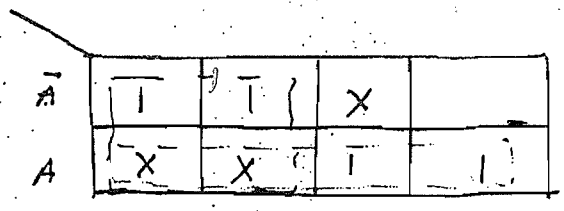
Ques:- $f(A,B,C) = \Sigma m(0,1,6,7) + \Sigma d(3,5)$

Soln:-



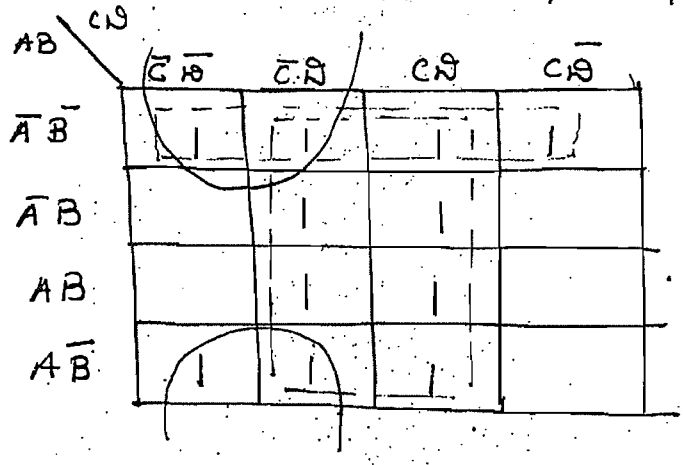
$= \bar{A}\bar{B} + A\bar{B}$

Note :-



Four Variables :-

$f(A, B, C, D) = \Sigma m(0, 1, 2, 3, 5, 7, 8, 9, 11, 13, 15)$

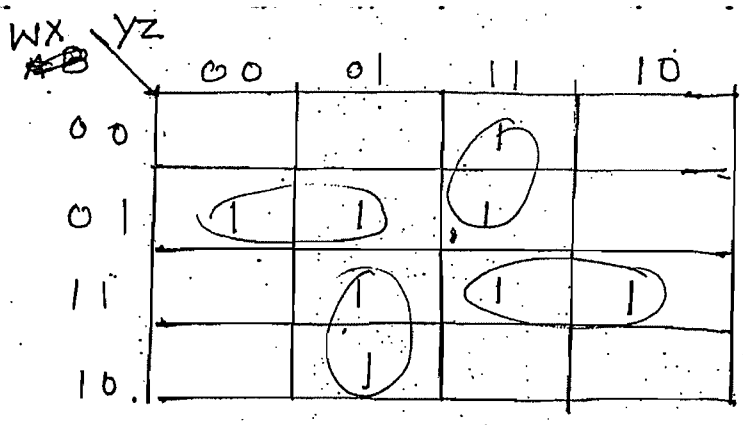


PI = 3

EPI = 3

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

Ques :-



PI = ?

EPI = ?

Soln :- PI = 5 , EPI = 4

Note :- $\underbrace{AB}_{2} + \underbrace{CD}_{2}$ → No. of variable = 2 = literal count.

→ fan in = No. of i/p

literal count = 2 + 2 = 4

POS :-

5 Variable :-

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

	A	B	C	D	E
0	0	0	0	0	0
1	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
⋮	⋮	⋮	⋮	⋮	⋮
31	1	1	1	1	1

$A = 0$

BC \ DE	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

$A = 1$

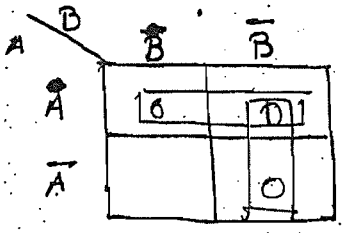
BC \ DE	00	01	11	10
00	16	17	19	18
01	20	21	23	22
11	28	29	31	30
10	24	25	27	26

→ If pair / quad is present in both then A & $\bar{A}$ gets cancelled then result is acc. to only Map

→ If pair / quad is present in only one Map then it is ~~multi~~ result is multiply either by A or $\bar{A}$ acc. to condition / presence

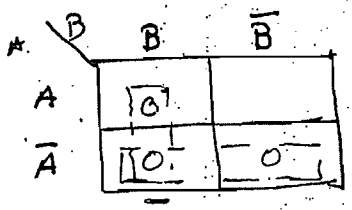
POS :-

$f(A,B) = \Pi M(0,1,3)$



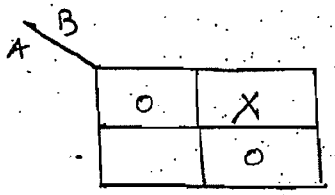
$= (A)(\bar{B})$

$f(A,B) = \Pi M(0,2,3)$



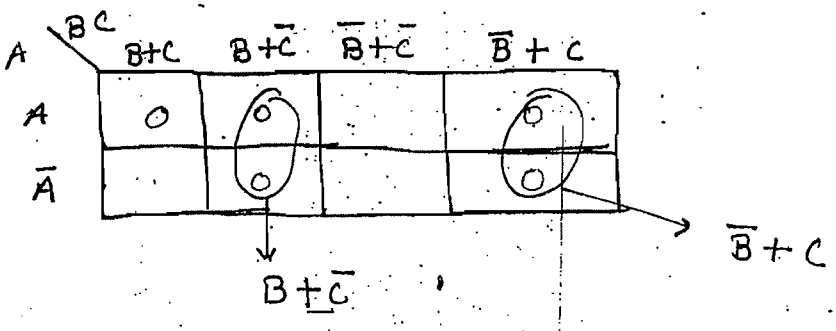
$= (\bar{A}) \cdot (B)$

$f(A,B) = \Pi M(0,2,3) + \Pi d(1)$



$= A \cdot \bar{B}$

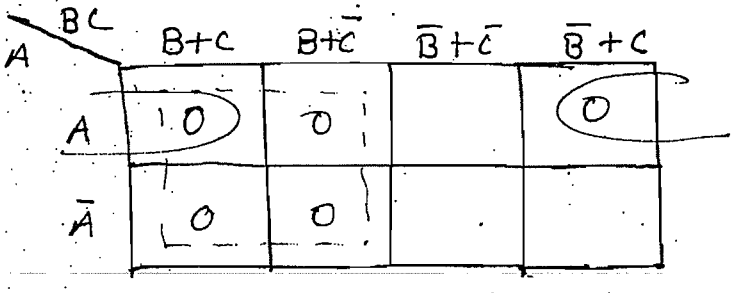
$f(A,B,C) = \Pi M(0,1,2,5,6)$



→ $(\bar{B}+C)(B+\bar{C}).(A+B)$

→ $(\bar{B}+C)(\bar{B}+C)(A+C)$

→ $B \cdot (A+C)$



Ques:- Minimize POS

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

Ans:- 1 → removed then Ans = $\bar{B} \cdot C$

Ques:- f(A, B, C, D) POS = $\Pi M(0,1,4,5,8,9,13,15)$

Soln.-

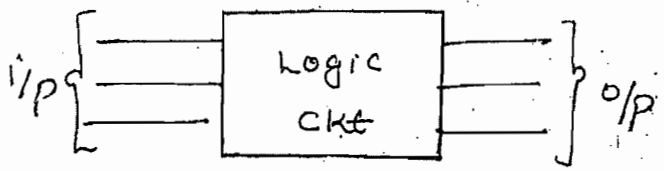
AB \ CD	C+D	C+$\bar{D}$	$\bar{C}$+$\bar{D}$	$\bar{C}$+D
A+B	1 0	0		
A+$\bar{B}$	1 0	0		
$\bar{A}$+$\bar{B}$		1 0	0	
$\bar{A}$+B	0	0		

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

literal count = 3 + 2 + 2 = 7

Digital Circuits

Combinational



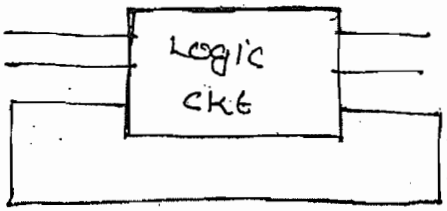
→ Present o/p depends on only present i/p's

→ No feedback

→ No Memory

→ eg:- HA, FA, MUX, DEMUX

Sequential Circuit



→ Present o/p depends

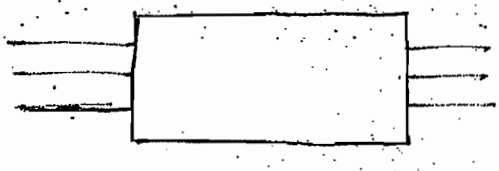
- Present i/p
- Previous/Past o/p's

→ feedback

→ Memory

→ eg:- FF, Registers, counters

Combinational Circuit:-



Procedure for Design:-

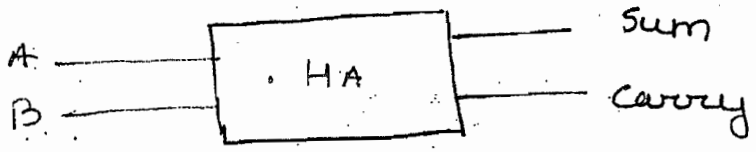
→ Identify no. of i/p's and o/p's
→ Construct truth table
→ Write logical expression in SOP or POS
→ Minimize logical expression
→ Implement logic ckt

Arithmatic Circuits :-

1) Half Adder :-

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

1 0 → Sum
↑ carry



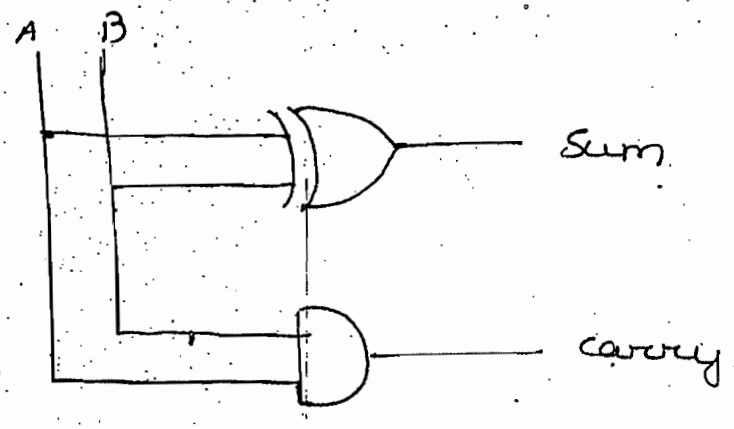
Truth Table :-

A	B	Sum	Carry
0	0	0	0
0	1	1 ✓	0
1	0	1 ✓	0
1	1	0	1 ✓

Logical Expression :-

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

Carry => AB



HA Questions :-

→ Logical expression for sum → $A \oplus B$

→ Carry → AB

→ Min. no of NAND → 5

→ " " " NOR → 5

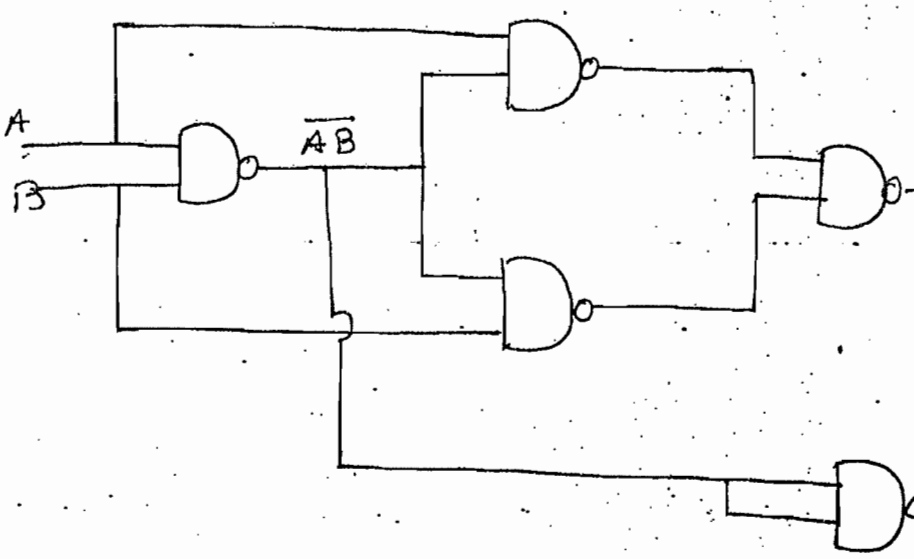
→ " " " MUX

→ " " " Decoder

HA using NAND Gate :-

Sum = 4 NAND

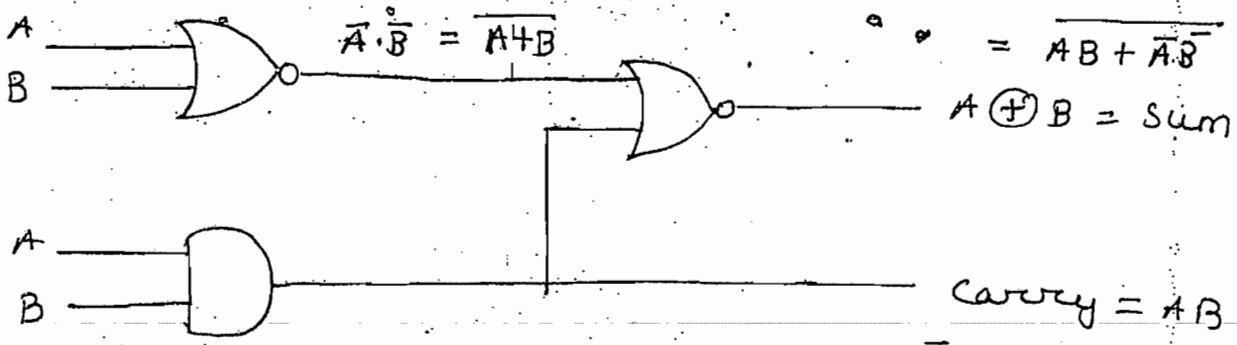
Carry = 2 NAND

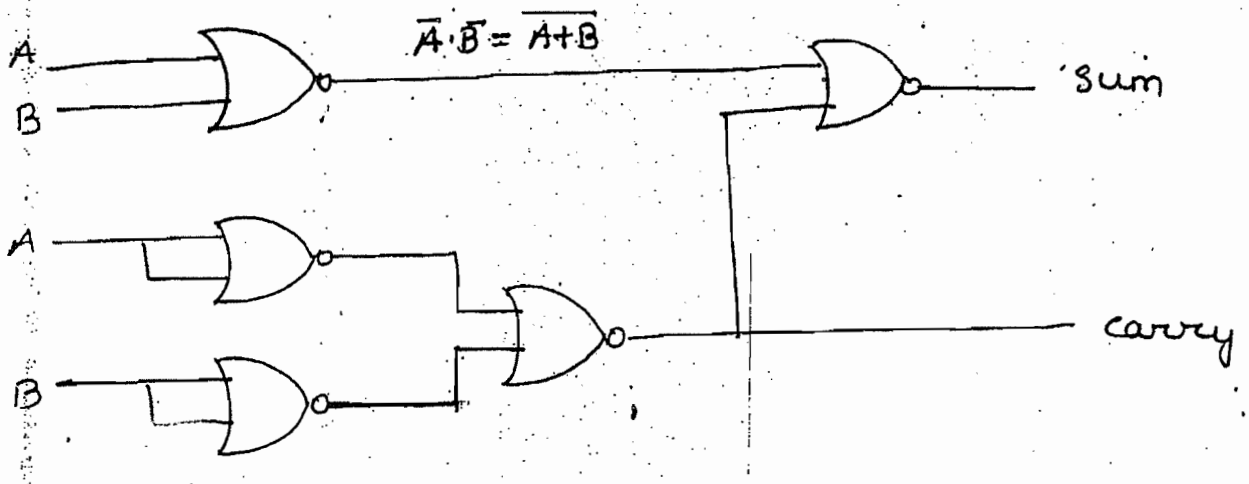


HA using NOR Gate :-

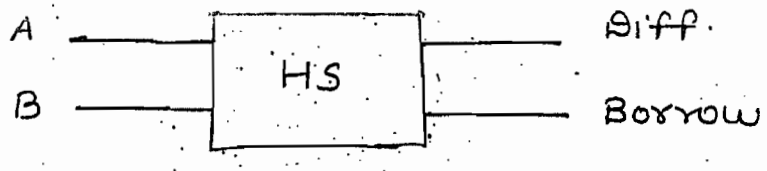
$A \oplus B = \overline{A \odot B} = \overline{AB + \overline{A}\,\overline{B}}$ ①

3





Half subtractor :-

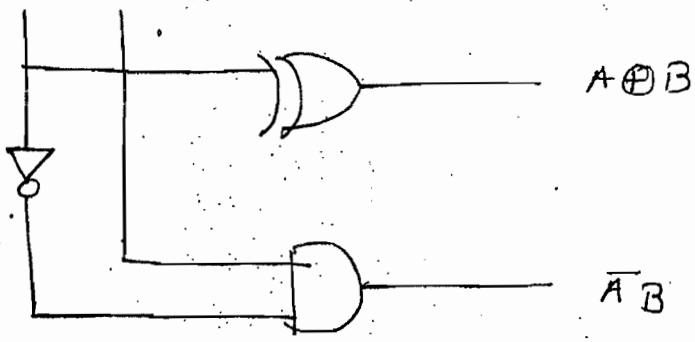


Truth Table :-

A	B	Diff.	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Logic Ckt :-

Diff. $\Rightarrow$ $A \oplus B$ Borrow $\Rightarrow$ $\overline{A}B$



Question of HS :-

→ Logical exp. for Diff. → $A \oplus B$

→ Logical exp. for Borrow → $\bar{A}B$

→ Min. no. of NAND → 5

→ Min. no. of NOR → 5

→

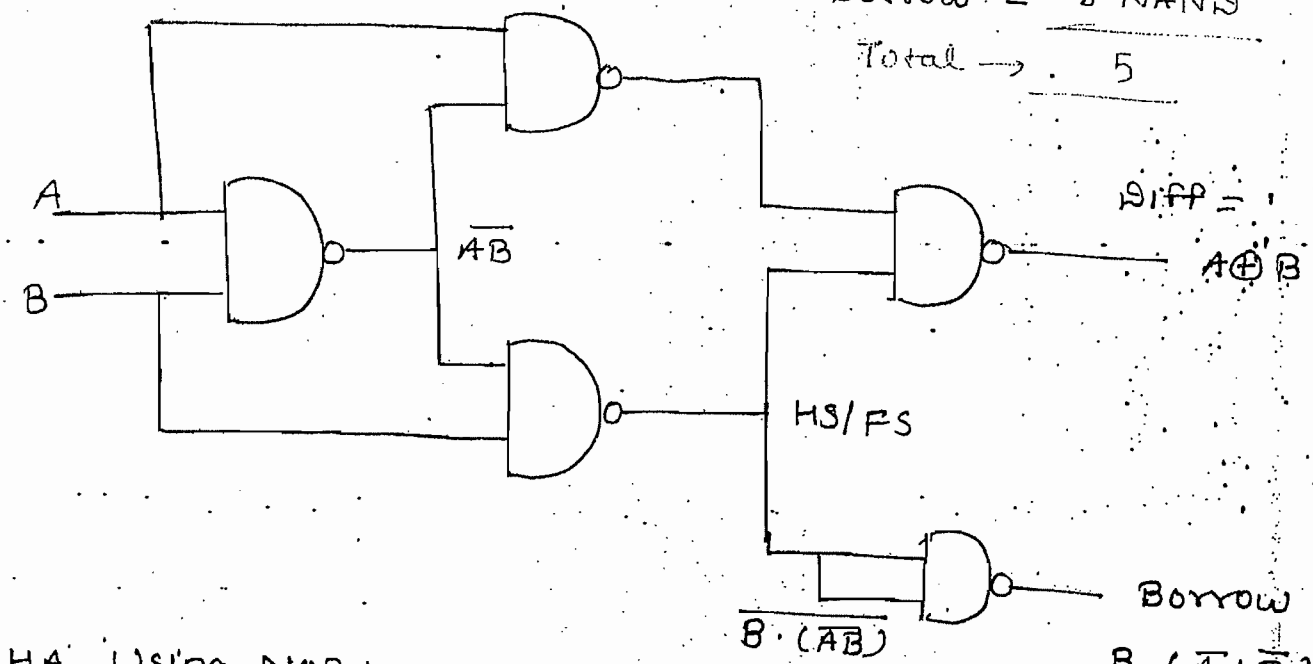
→

HA Using NAND :-

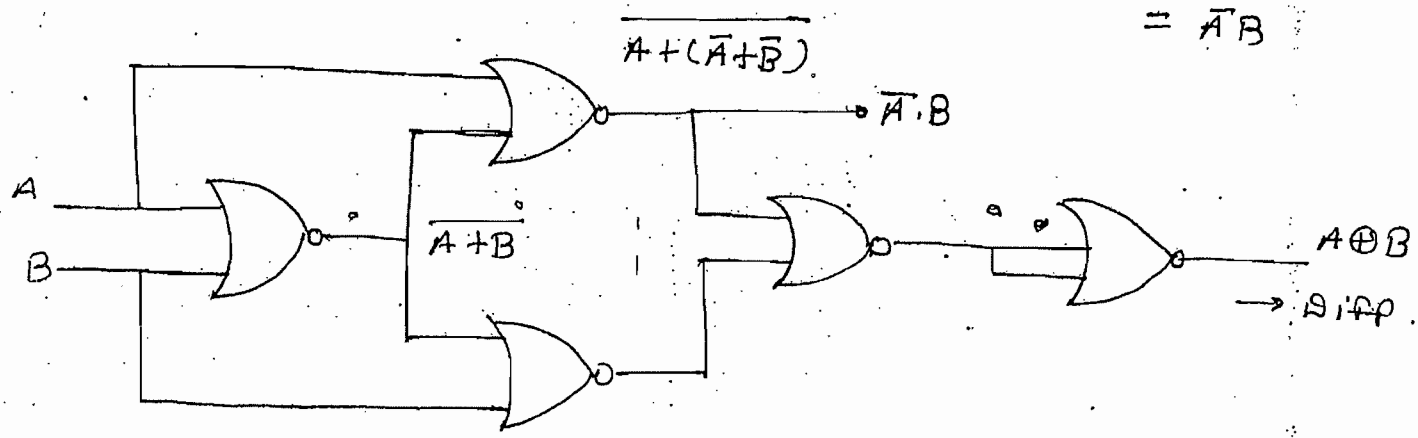
Diff = 4 NAND

Borrow = 3 NAND

Total → 5



HA Using NOR :-

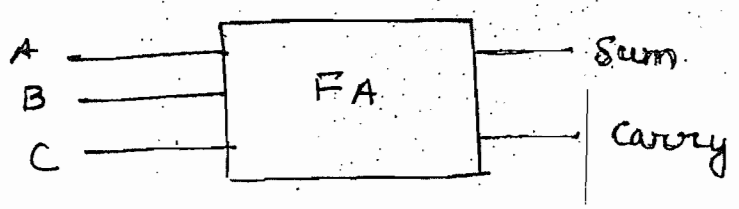


Diff ⇒ 5

Borrow ⇒ 2

5

Full Adder :— → Add 3 bits



A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Logical Expression :—

$$\text{Sum} = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

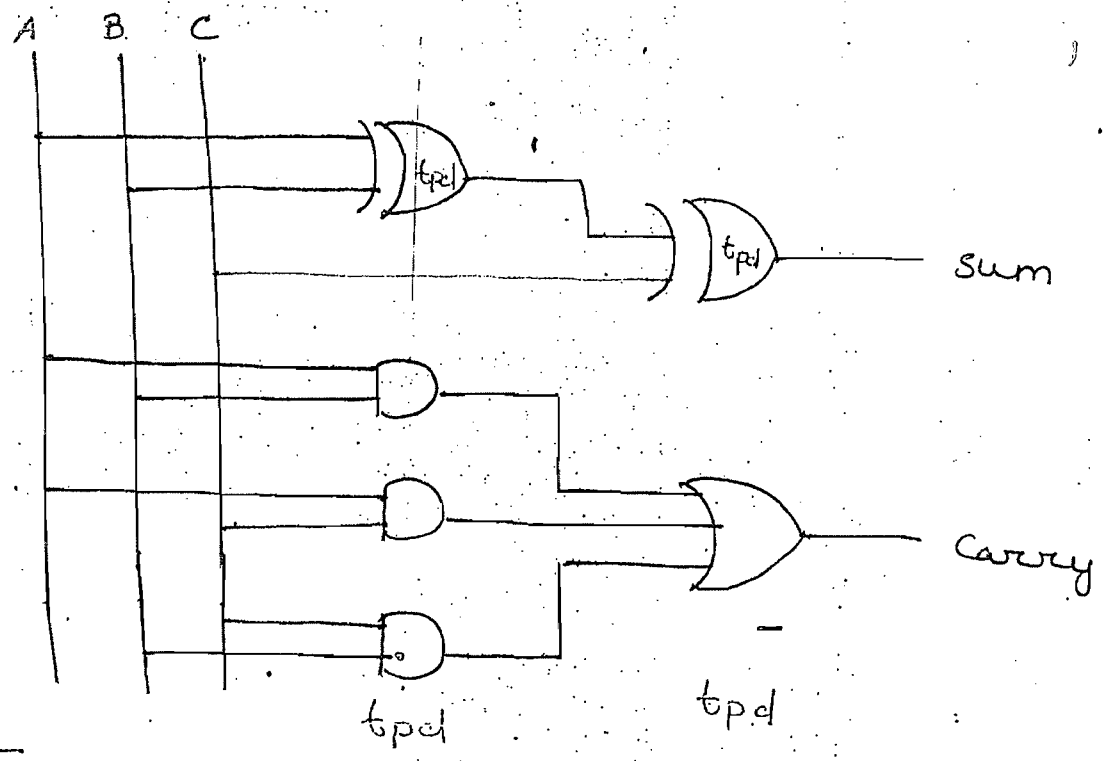
$$= A \oplus B \oplus C$$

$$= \Sigma m(1, 2, 4, 7)$$

$$\text{Carry} = \Sigma m(3, 5, 6, 7) \quad \text{—— (1)}$$

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

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



Note:-

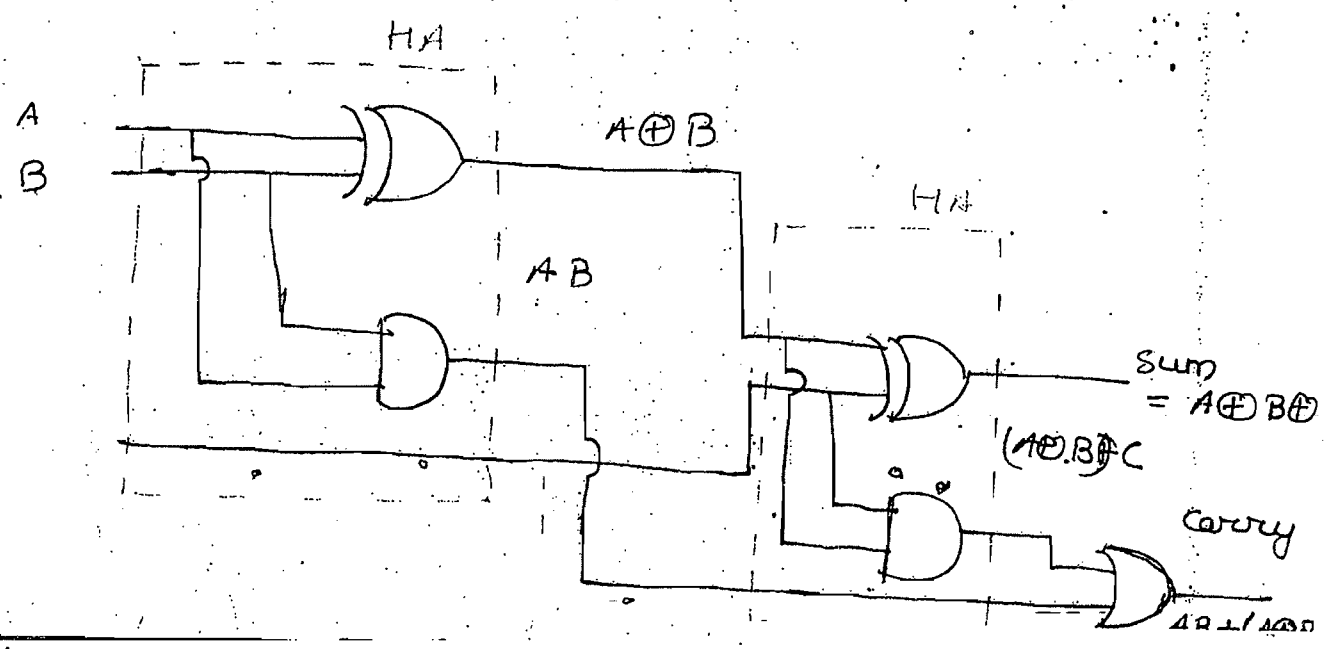
→ In FA if all logic gate have same propogation delay t_{pd} then to provide sum or carry o/p minimum two t_{pd} delay is required

Alternate carry expression:-

$$\text{Carry} = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

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

$$\boxed{\text{Carry} = AB + (A \oplus B)C} \quad \text{---(3.)}$$



FA = 2HA + 1OR

→ Sum → $A \oplus B \oplus C$

→ Sum → $\Sigma m\,(1,2,4,7)$

→ Carry → $\Sigma m\,(3,5,6,7)$

→ Carry → $AB + BC + AC$

→ Carry → $AB + (A \oplus B)C$

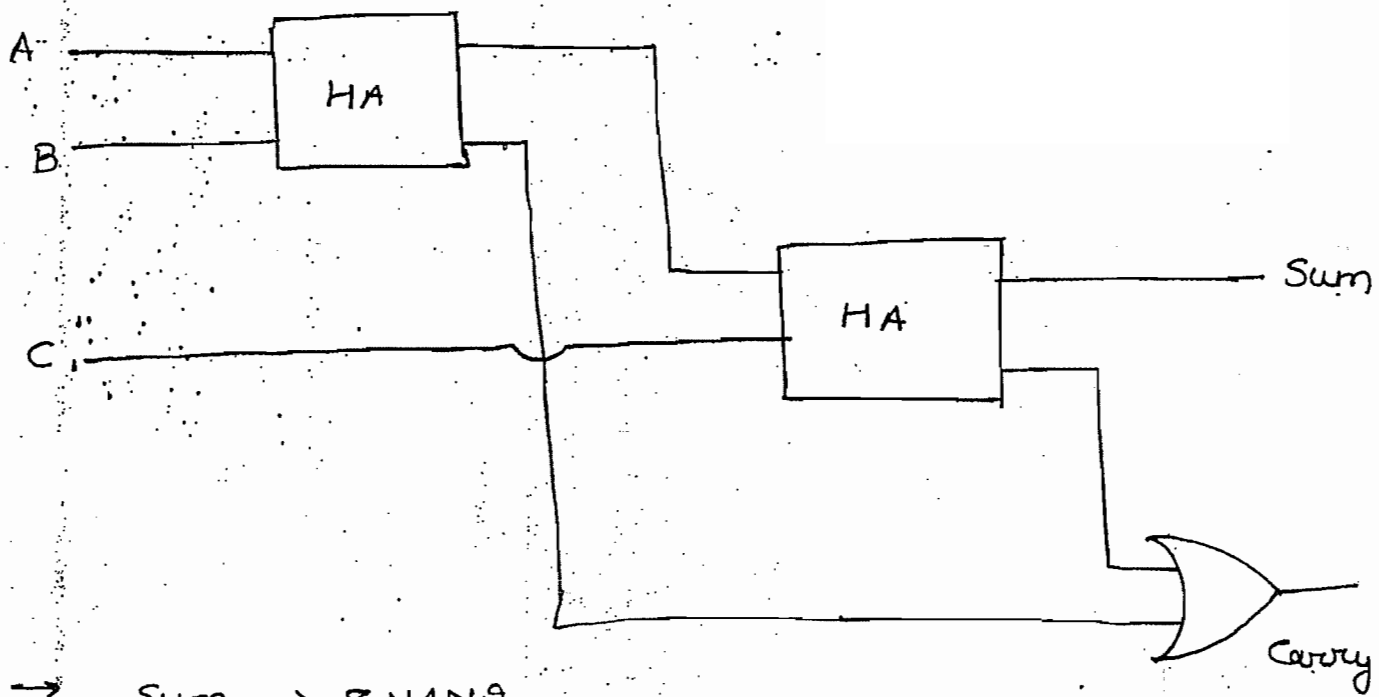
→ No. of HA and OR ⇒ 2HA + 1OR

→ Min. no. of NAND ⇒ 9

→ " " " NOR ⇒ 9

→ No. of MUX ⇒

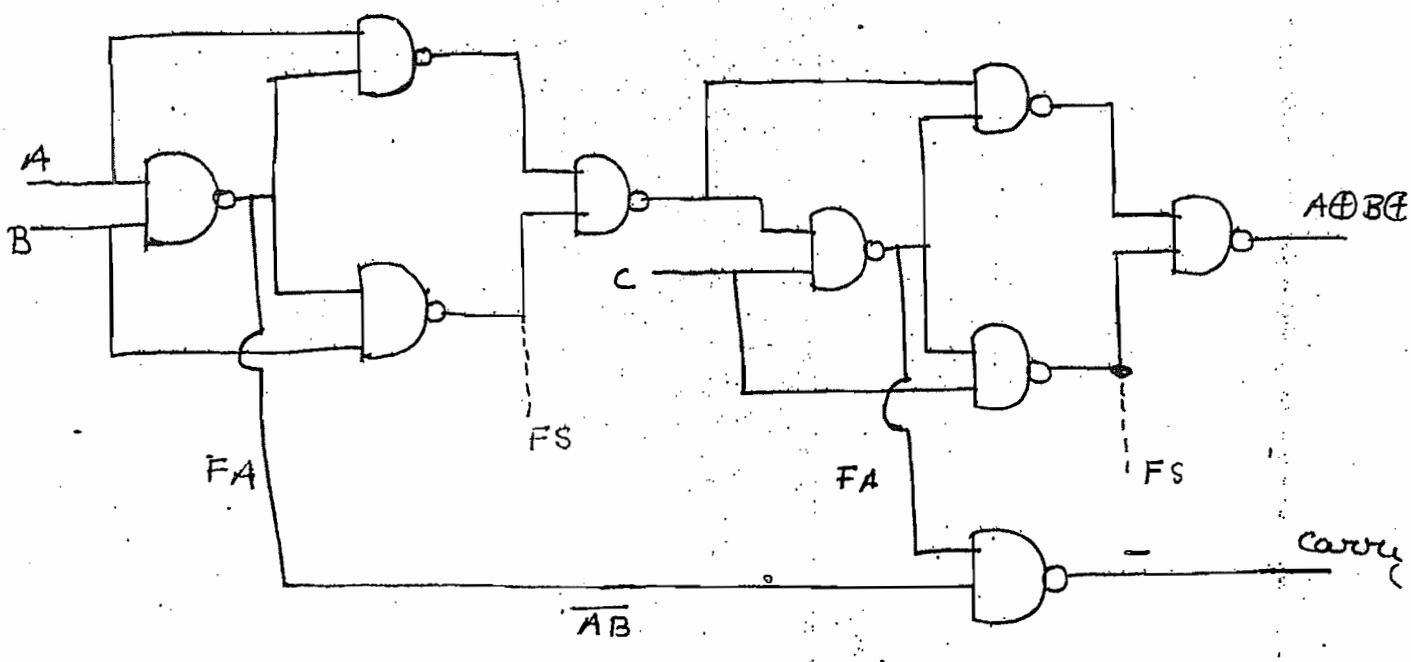
→ No. of Decoder ⇒



→ Sum ⇒ 8 NAND

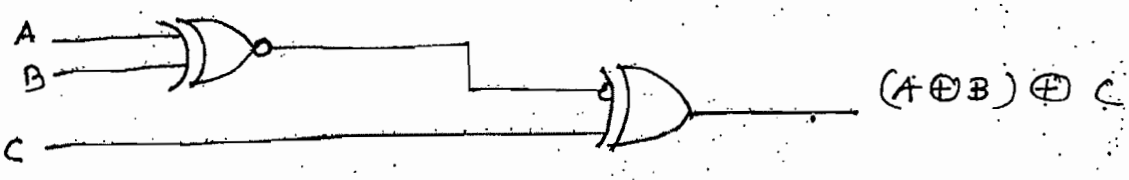
→ Carry ⇒ 6 NAND

FA using NANDS :-



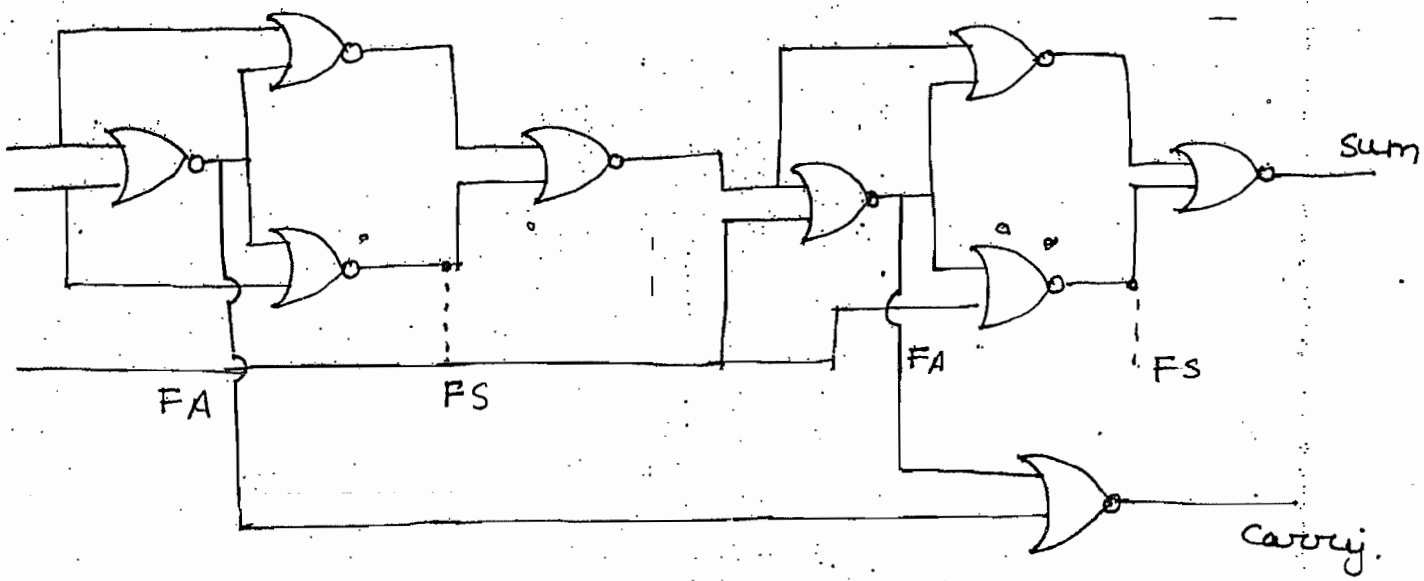
$$= \overline{\overline{AB}\ \overline{(A \oplus B)C}} = AB + (A \oplus B)C$$

FA Using NOR :-

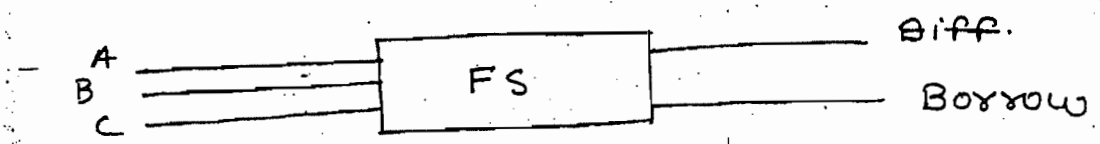


$A \oplus B \oplus C$ → $(A \oplus B) \oplus C$

$A \oplus B \oplus C$ → $(A \odot B) \odot C$



Full subtractor :-



Truth Table :-

A	B	C	Difference	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Difference $= \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

$= A \oplus B \oplus C = \Sigma m(1,2,4,7)$

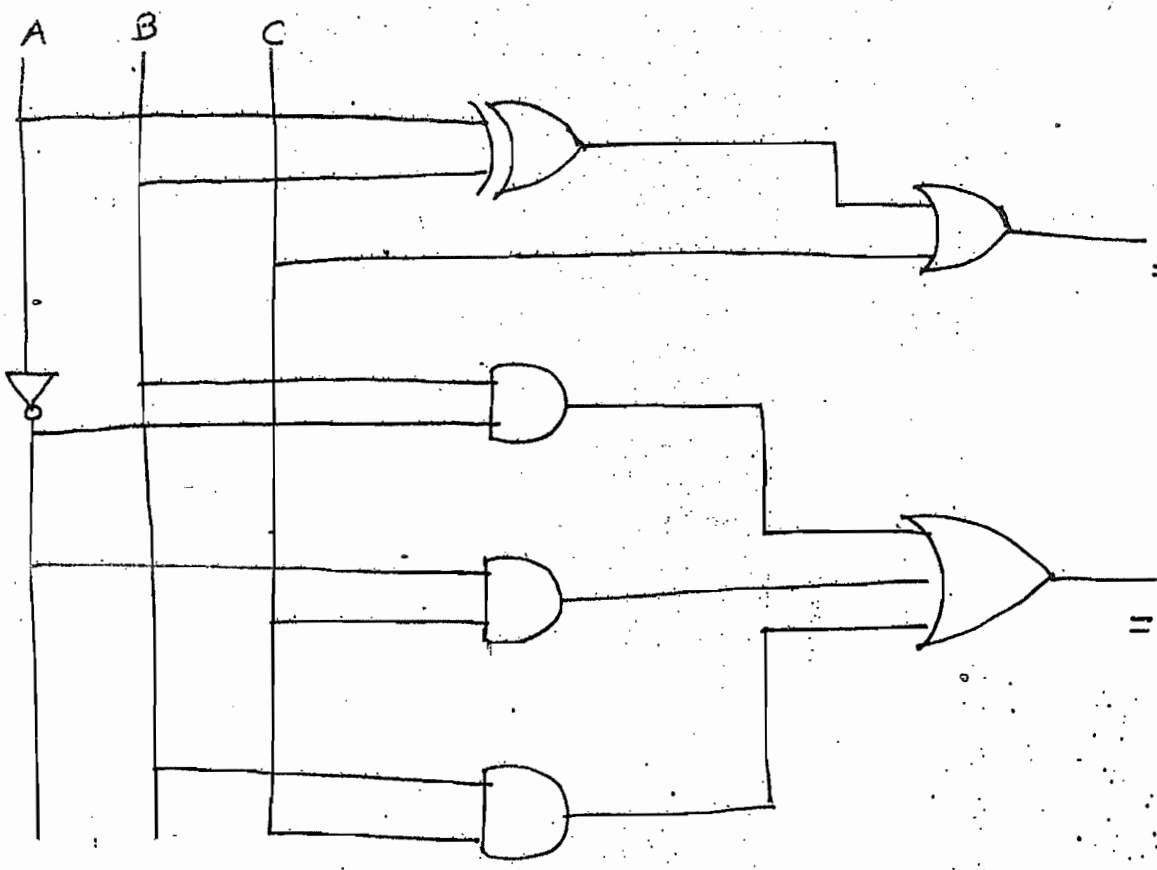
Carry $= \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + \bar{A}BC + ABC = \Sigma m(1,2,3,7)$

$= \bar{A}C + \bar{A}B + BC \quad \text{—(1)}$

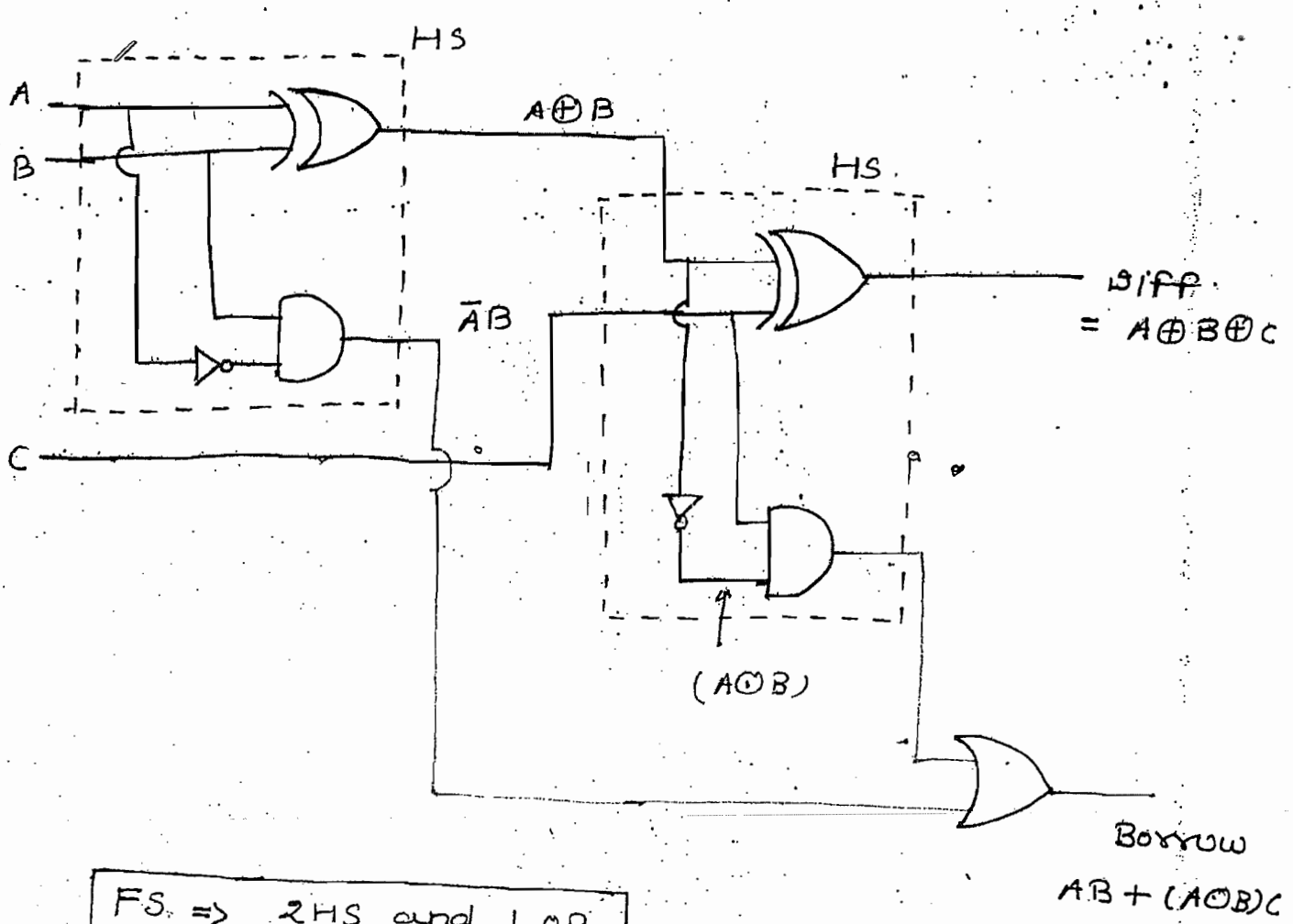
OR

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

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



OR



FS. => 2HS and 1 OR

Question :-

→ Logical expression for difference → $\Sigma m(1,2,4,7)$ / $A \oplus B \oplus C$

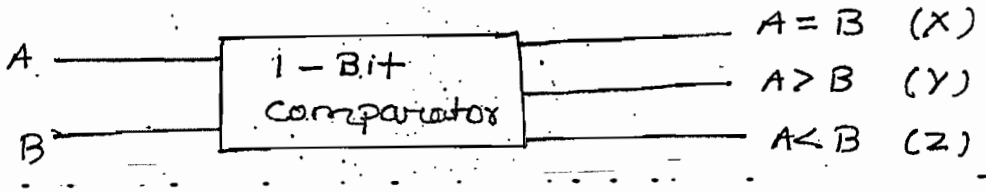
→ Logical expression for Borrow → $\Sigma m(1,2,3,7)$ / $\bar{A}B + BC + \bar{A}C$ / $\bar{A}B + (A \odot B)C$

→ Min. no. of NAND => 9

→ " " " NOR => 9

→ No. of HS & OR => 2HS + 1OR

Comparator :-



Truth Table :-

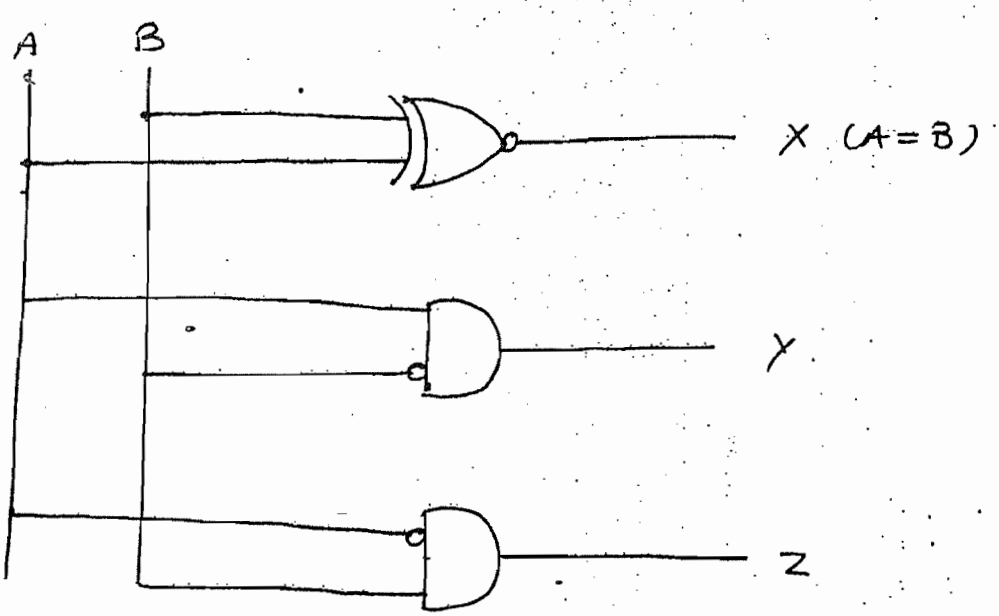
A	B	A = B (X)	A > B (Y)	A < B (Z)
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

Logical Expressions :-

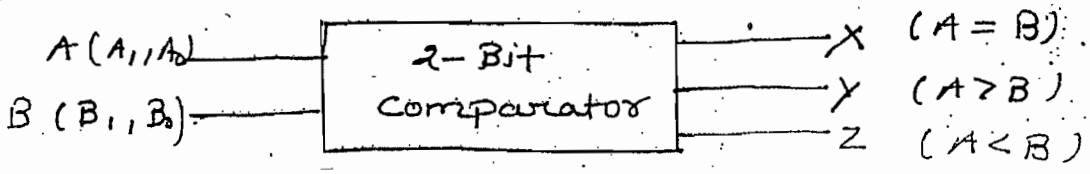
$X = \bar{A}\bar{B} + AB = A \odot B$

$Y = A\bar{B}$

$Z = \bar{A}B$



2-Bit Comparator :-



A		B		A=B	A>B	A<B
A_1	A_0	B_1	B_0	X	Y	Z
0	0	0	0	1	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0			
0	1	1	1			

→ $A = B = (A_1 = B_1) \cdot (A_0 = B_0)$

$= \overline{(A_1 \odot B_1)} \cdot (A_0 \odot B_0)$

→ $A > B = (A_1 > B_1)$ or $(A_1 = B_1) \cdot (A_0 > B_0)$

$= (A_1 \cdot \overline{B_1}) + (A_1 \odot B_1) \cdot A_0 \overline{B_0}$

→ $A < B = \overline{A_1} \cdot B_1 + (A_1 \odot B_1) \cdot \overline{A_0} \cdot B_0$

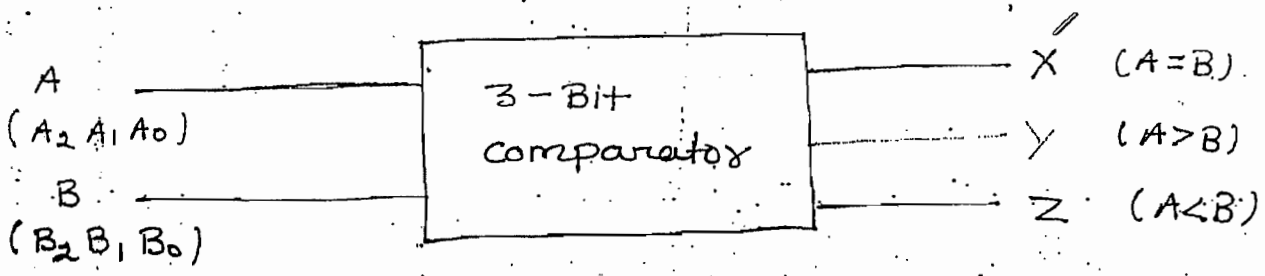
Note :-

→ In n-bit comparator total no. of combination

→ In n-bit comparator, no. of equal condition = 2^n

→ In n-bit comparator, no. of greater/lesser condition present = $\frac{2^{2n} - 2^n}{2}$

3-Bit comparator :-



→ $X = (A_2 \odot B_2) \cdot (A_1 \odot B_1) \cdot (A_0 \odot B_0)$

→ $Y = A_2 \overline{B_2} + (A_2 \odot B_2) A_1 \overline{B_1} + (A_2 \odot B_2)(A_1 \odot B_1) A_0 \overline{B_0}$

→ $Z = \overline{A_2} \cdot B_2 + (A_2 \odot B_2) \overline{A_1} B_1 + (A_2 \odot B_2)(A_1 \odot B_1) \cdot \overline{A_0} B_0$

Note :-

→ $X_i = A_i \oplus B_i$

$X_0 = A_0 \oplus B_0$
$X_1 = A_1 \oplus B_1$
$X_2 = A_2 \oplus B_2$

→ For $A = B$, $X = X_2 \cdot X_1 \cdot X_0$

→ For $A > B$, $Y = A_2 \overline{B_2} + X_2 \cdot A_1 \overline{B_1} + X_2 X_1 A_0 \overline{B_0}$

→ For $A < B$, $Z = \overline{A_2} B_2 + X_2 \cdot \overline{A_1} B_1 + X_2 X_1 \overline{A_0} B_0$

Note:-

A	B	Y
0	0	1
0	1	0
1	0	1
1	1	C

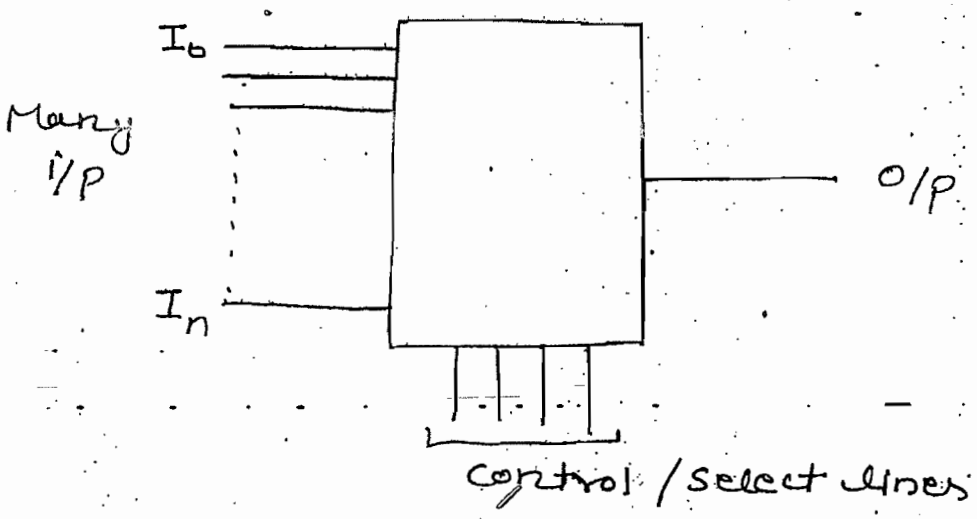
SOP Expression:-

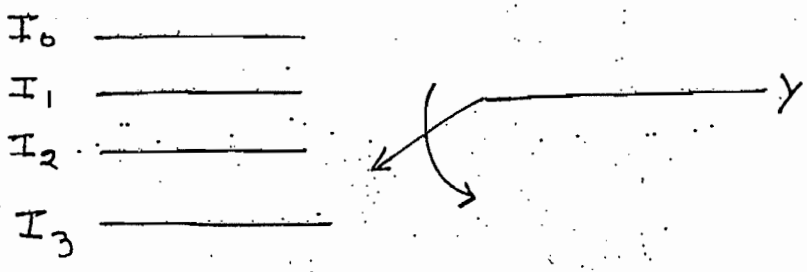
$Y = \bar{A}\bar{B}\cdot 1 + \bar{A}B\cdot 0 + A\bar{B}\cdot 1 + AB\cdot$

POS Expression:-

$Y = (A+B+1)(A+\bar{B}+C)(\bar{A}+B+1)$
$(\bar{A}+\bar{B}+C)$

Multiplexer:-





→ It is a combinational circuit which have many data I/P's and single O/P

→ Depending on select, one of the data I's transfers to the O/P

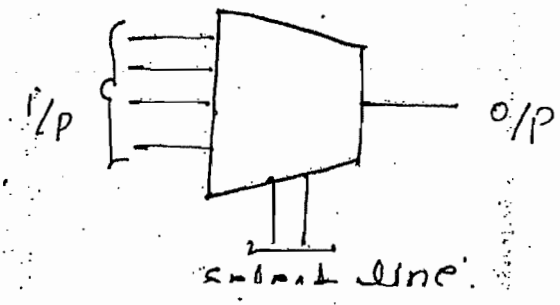
→ Hence MUX is known as Many to one circuit

→ Data selector

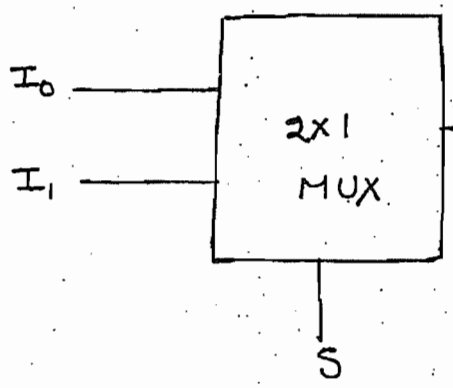
→ Universal logic circuit

→ Parallel to serial data converter

Symbol:-

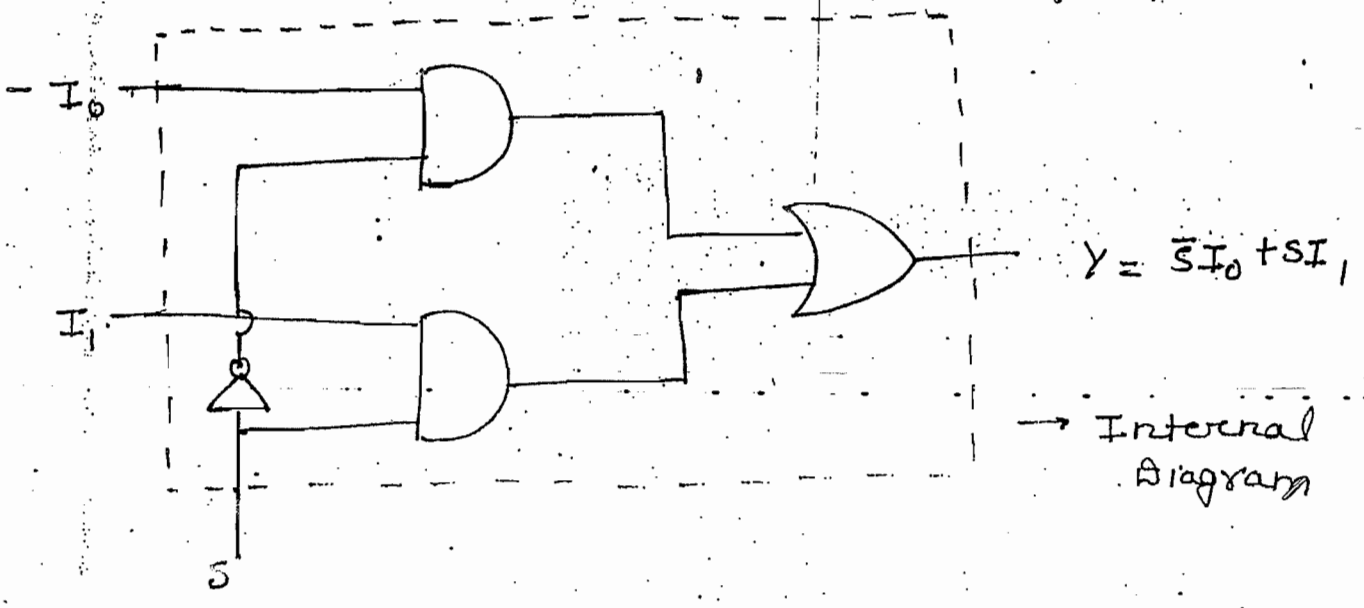


2 : 1 MUX :-



$Y = \bar{S} I_0 + S I_1$

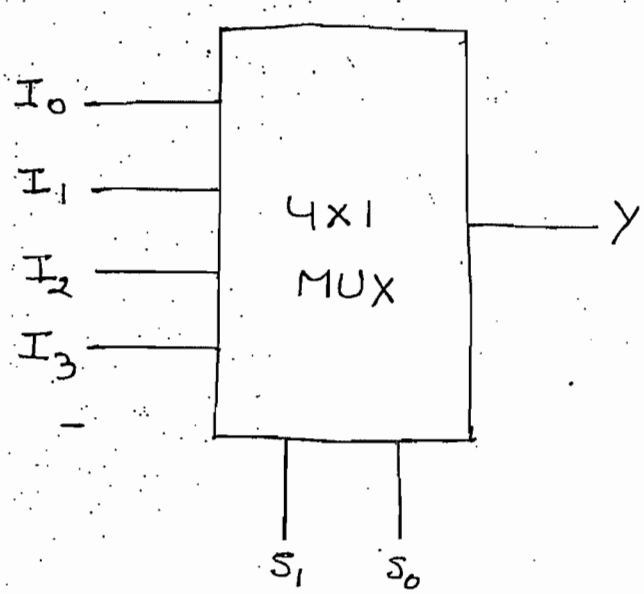
S	Y
0	I_0
1	I_1



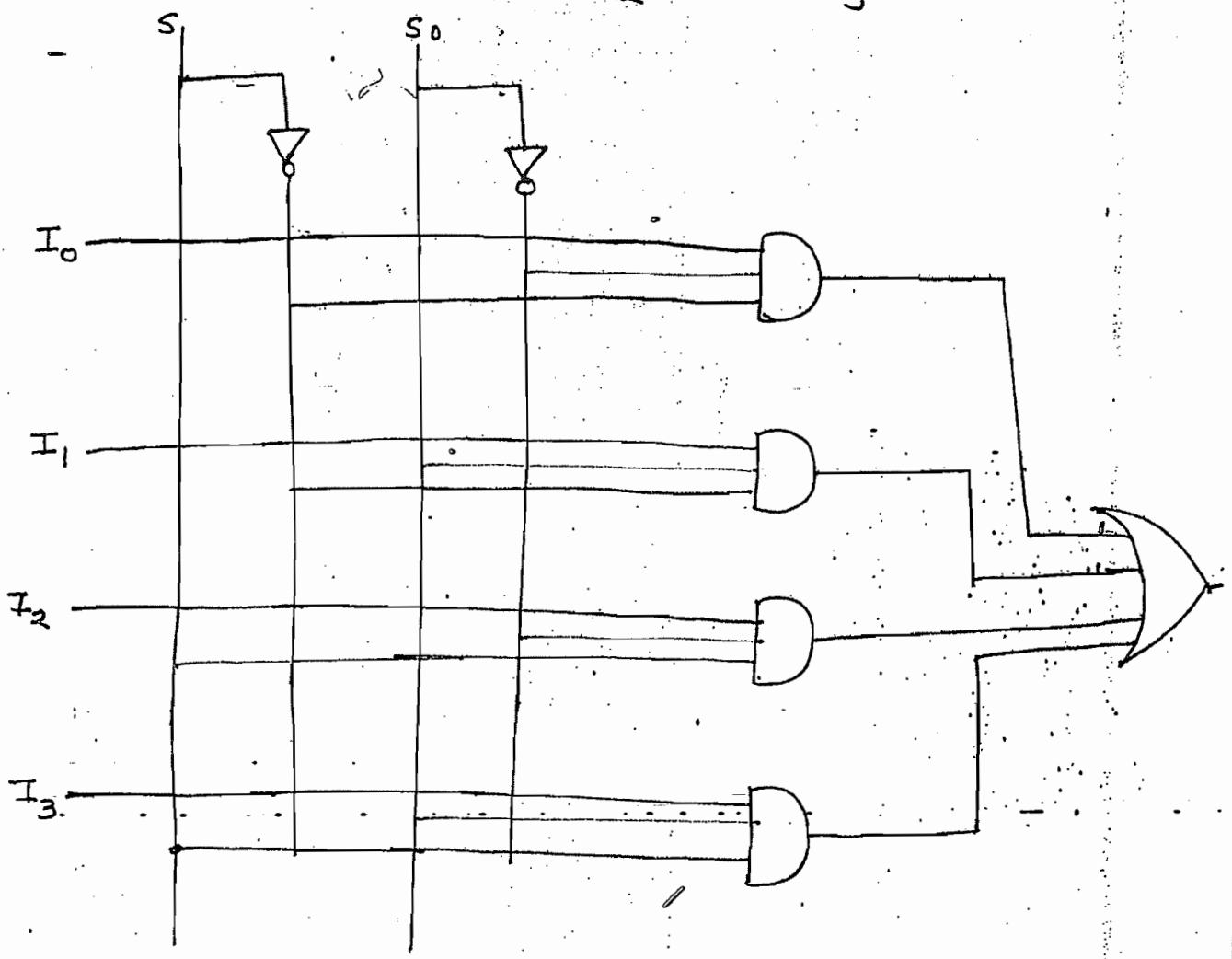
→ Two level AND - OR because data is not going through Not gate.

4 : 1 MUX :-

S_1	S_0	Y
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3



$$Y = \bar{S_1}\bar{S_0}I_0 + \bar{S_1}S_0I_1 + S_1\bar{S_0}I_2 + S_1S_0I_3$$



For 8:1 MUX :-

$$Y = \bar{S_2}\bar{S_1}\bar{S_0}I_0 + \bar{S_2}\bar{S_1}S_0I_1 + \cdots + S_2S_1S_0I_7$$

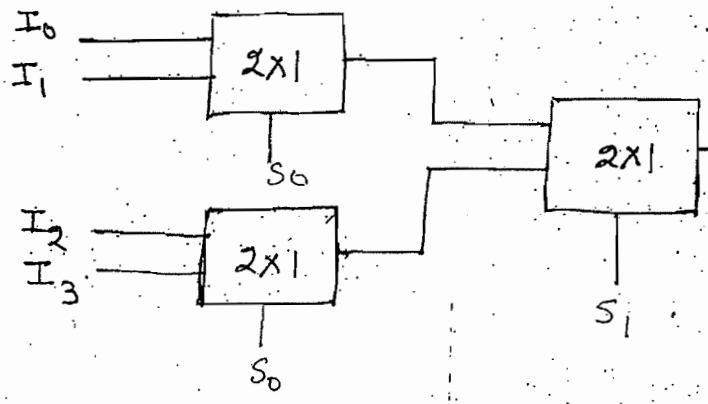
Types of questions in MUX :-

(1) Implementation of Higher order MUX using Lower order MUX :-

(a) 4×1 $\longrightarrow$ 2×1 MUX
S_1 S_0

operation :-

S_1	S_0	Y
1	0	I_2



$\frac{4}{2} = 2, \frac{2}{2} = 1$

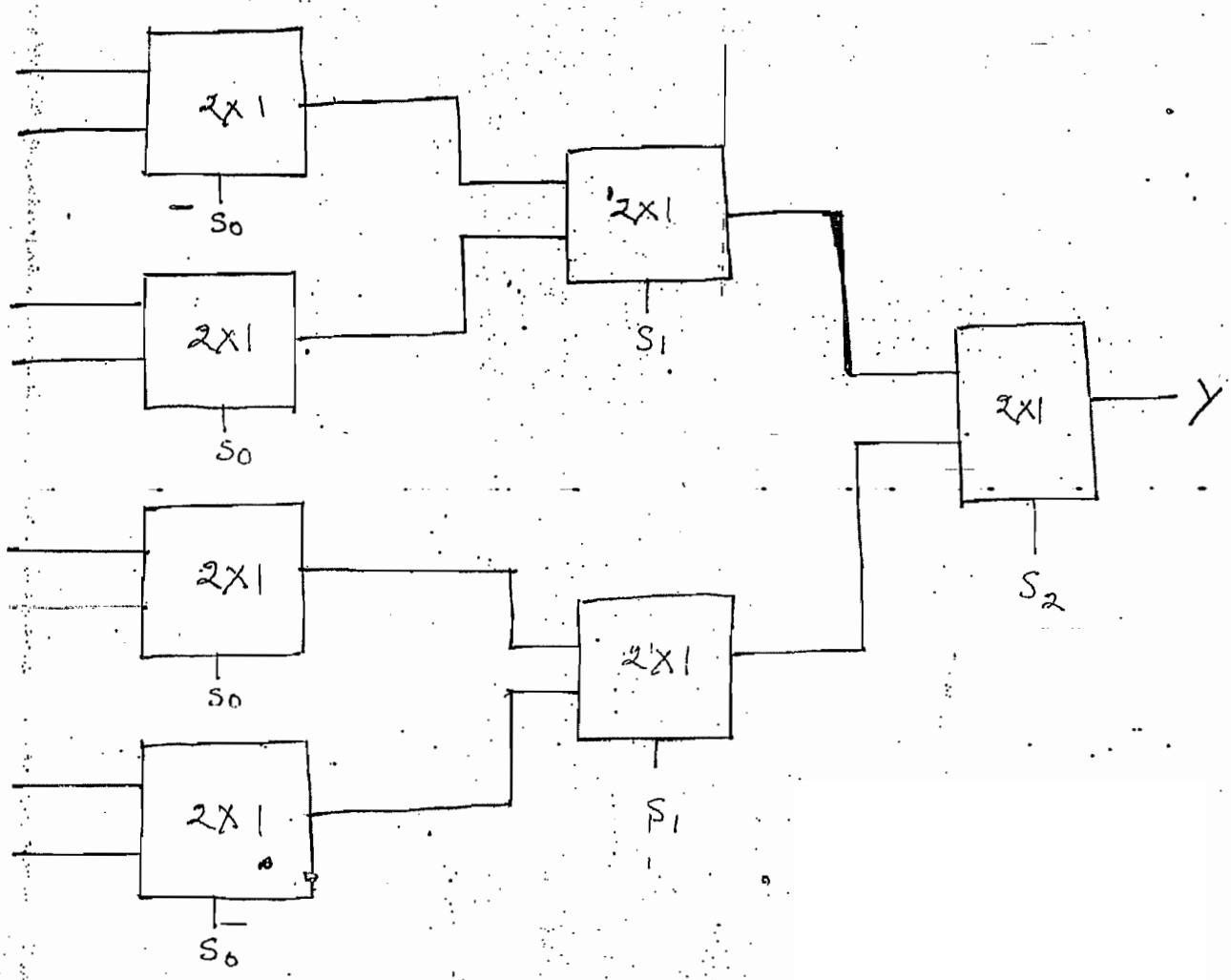
No. of MUX

(b) 8 x 1 MUX Using 2x1 MUX :-

8 x 1 ⟶ 2x1 MUX

$\frac{8}{2} = (4)$, $\frac{4}{2} = (2)$, $\frac{2}{2} = (1)$

No. of MUX required = 4 + 2 + 1 = 7



(c) 16x1 MUX using 2x1 MUX :-

16 x 1 ⟶ 2x1

$\frac{16}{2} = (8)$, $\frac{8}{2} = (4)$, $\frac{4}{2} = (2)$, $\frac{2}{2} = (1)$

No. of MUX req = 8 + 4 + 2 + 1 = 15

Note :-

No. of MUX req.

⇒ 64 x 1 — 63 → 2 x 1

⇒ 256 x 1 — 255 → 2 x 1

⇒ 2^n x 1 — $2^n - 1$ → 2 x

n = select i/p

2^n = i/p's

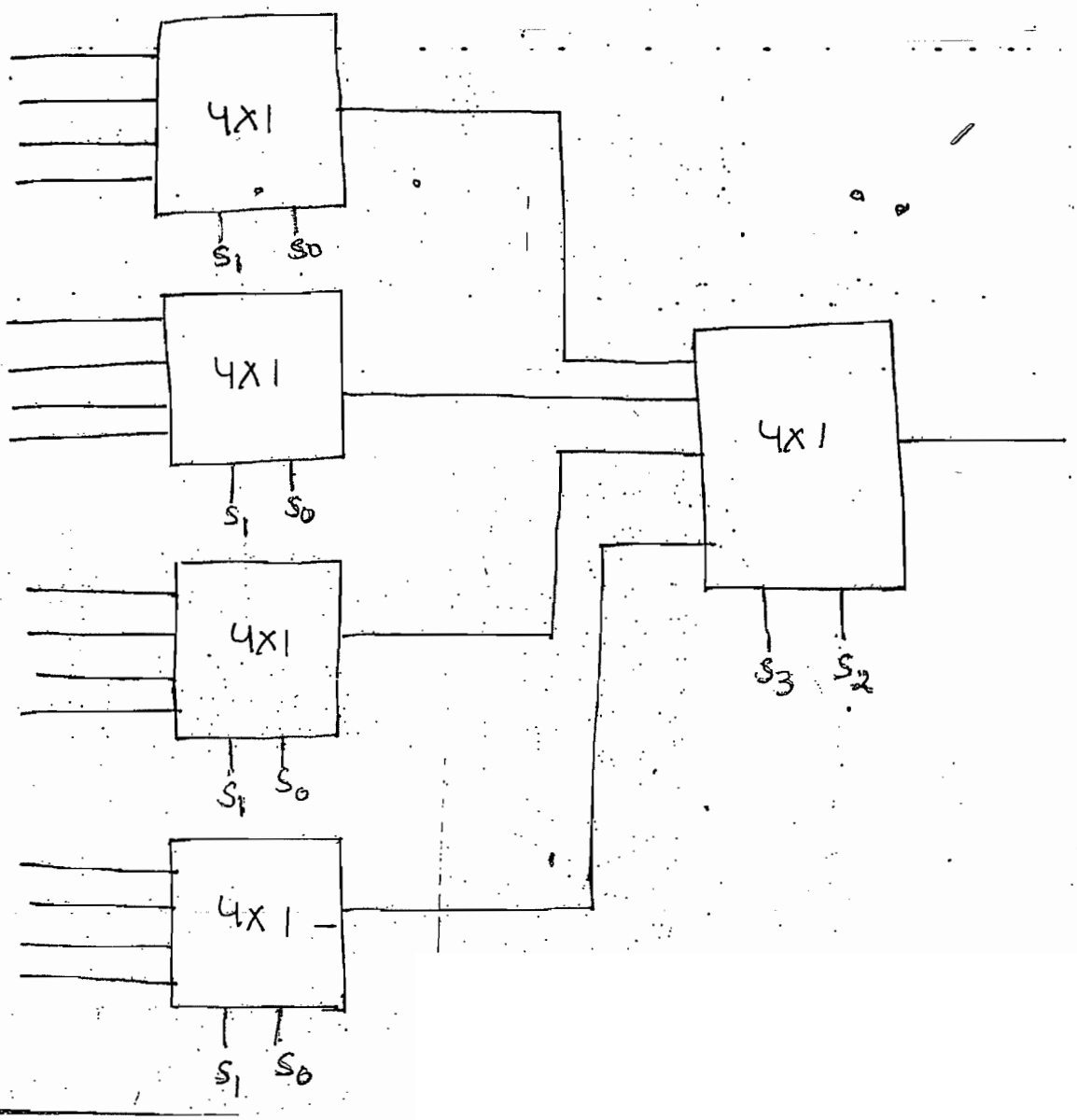
(d) 16 x 1 MUX using 4 x 1 MUX :-

16 x 1 ⟶ 4 x 1

S_3 S_2 S_1 S_0

$\frac{16}{4} = 4$, $\frac{4}{4} = 1$

No. of MUX Req = 4 + 1 = 5



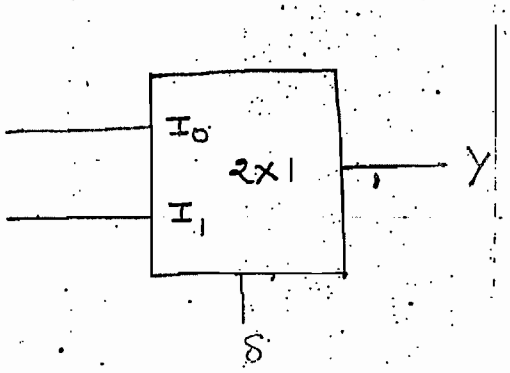
Note:- No. of MUX Required

⇒ 64 X 1 —21→ 4 X 1

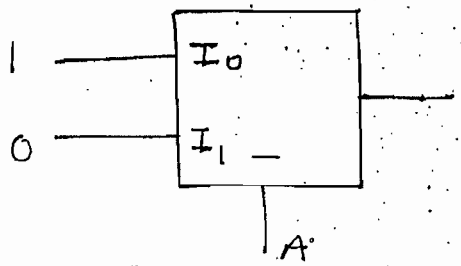
⇒ 64 X 1 —9→ 8 X 1.

⇒ 256 X 1 —17→ 16 X 1

(2) MUX as Universal Logic circuit :-



(a) NOT Gate :-



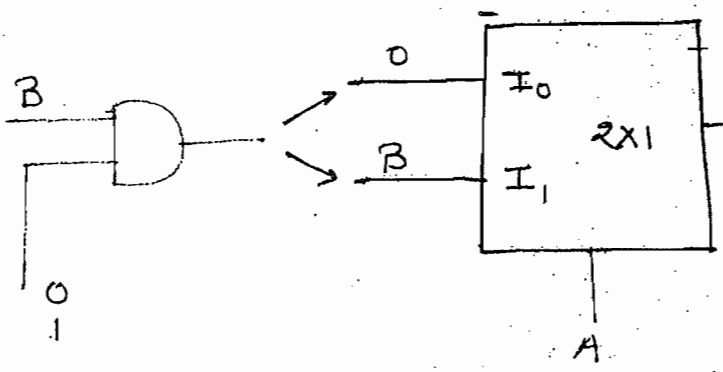
$Y = \bar{A} I_0 + A I_1$

$= \bar{A} \cdot 1 + A \cdot 0$

$\Rightarrow Y = \bar{A}$

A	Y
0	1
1	0

(b) AND Gate :-



$Y = \bar{S} I_0 + S I_1$

$= \bar{A} \cdot 0 + A B$

$= A B$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$\Rightarrow$

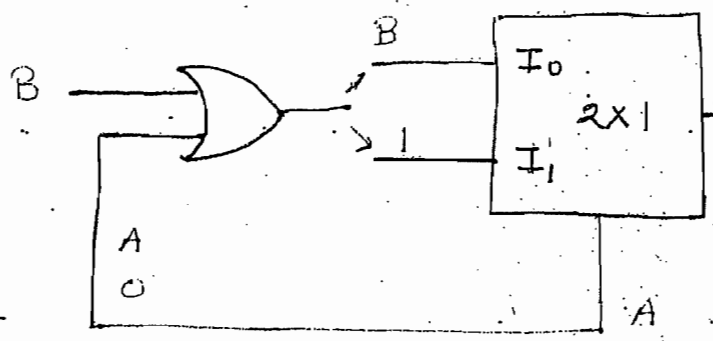
A	Y
0	0
1	B

$\rightarrow I_0$

$\rightarrow I_1$

$Y = \bar{A} \cdot 0 + A \cdot B = AB$

(c) OR Gate :-

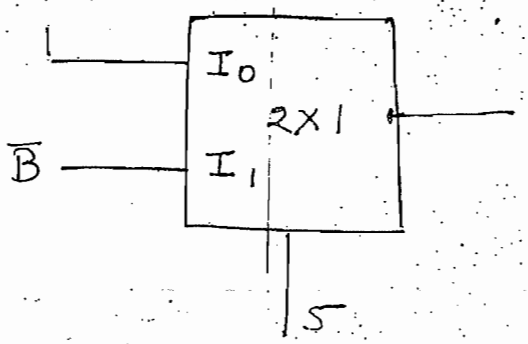


$Y = \bar{S} I_0 + S \cdot I_1$

$= \bar{A} B + A \cdot 1$

$= A + B$

(d) NAND Gate :-



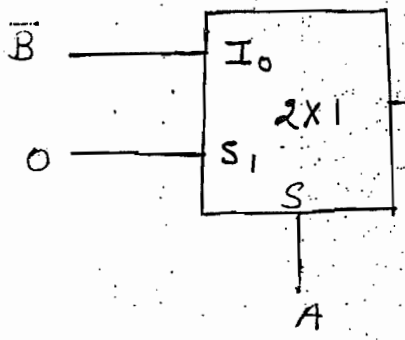
$Y = \bar{S} I_0 + S I_1$

$= \bar{A} \cdot 1 + A \bar{B}$

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

No. of 2x1 MUX implement NAND is 2

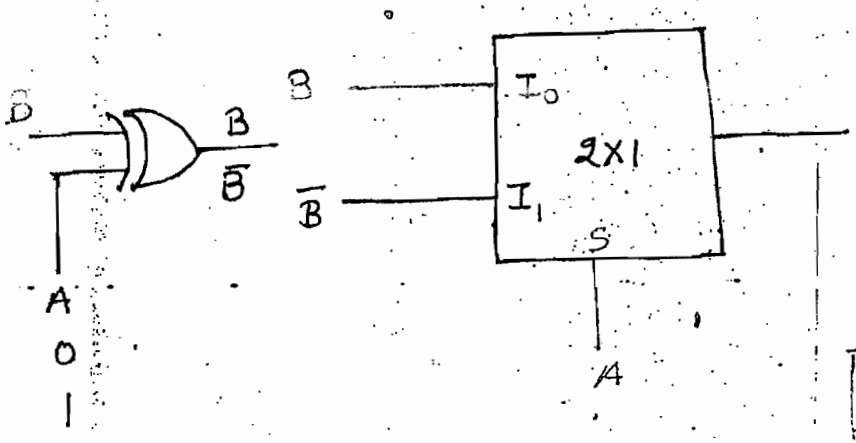
(e) NOR Gate :-



$$Y = \bar{S} I_0 + S I_1$$
$$= \bar{A}\bar{B} + A \cdot 0$$
$$= \bar{A} \cdot \bar{B} = \overline{A+B}$$

No. of ~~MUX~~ 2X1 MUX to implement NOR is 2.

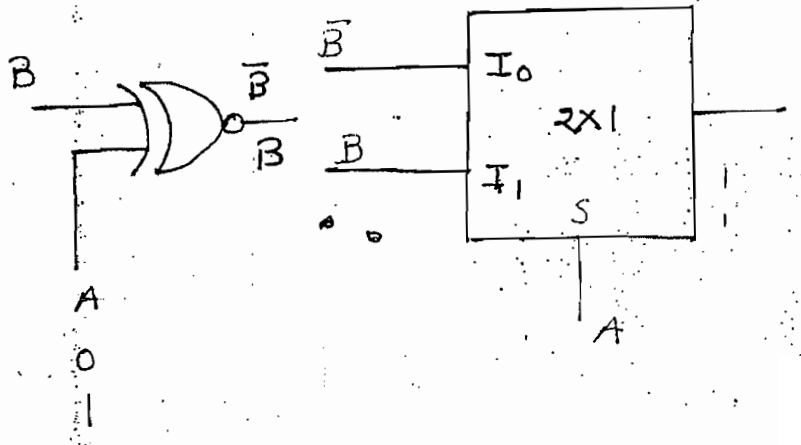
(f) EXOR Gate :-



$$Y = \bar{S} I_0 + S I_1$$
$$= \bar{A}B + A\bar{B}$$

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

(g) EX-NOR Gate :-



$$Y = \bar{S} I_0 + S I_1$$
$$= \bar{A}\bar{B} + AB$$

Ques:- To implement 2 i/p exclusive OR Gate & AND gate min. no. of 2X1 MUX required

Ans:- 2, 1

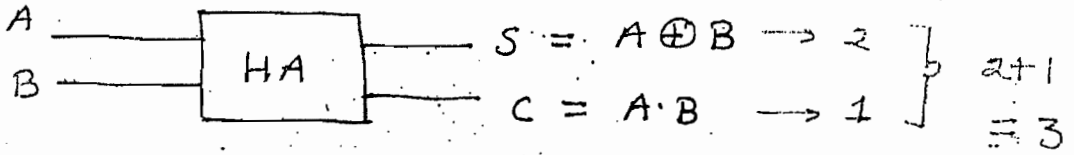
Note:-

NOT
AND
OR
→ Required (1) → 2X1 MUX

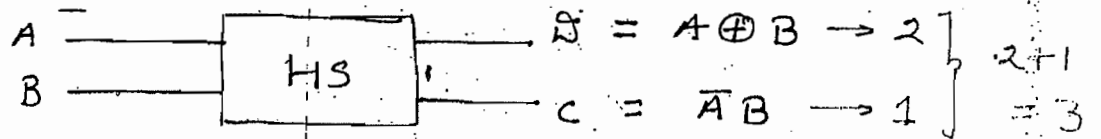
NAND
NOR
EXOR
EXNOR
→ Required (2) → 2X1 MUX

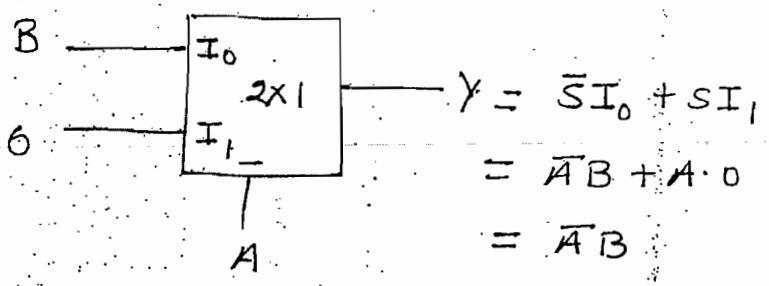
→ To implement half adder, No. of 2X1 MUX

(I) HA → 3 → 2X1 MUX



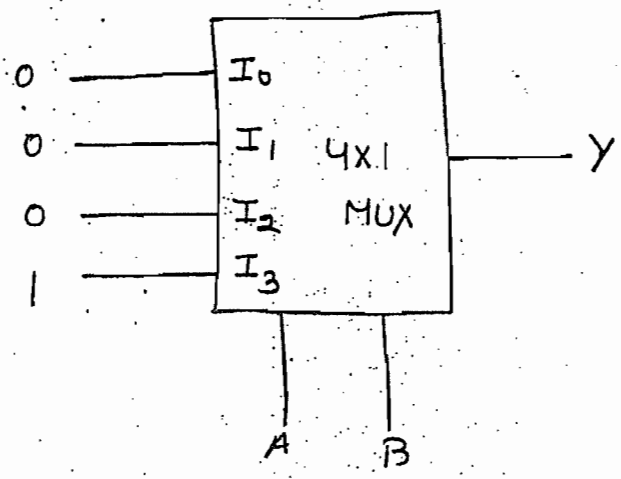
(II) HS → 3 → 2X1 MUX



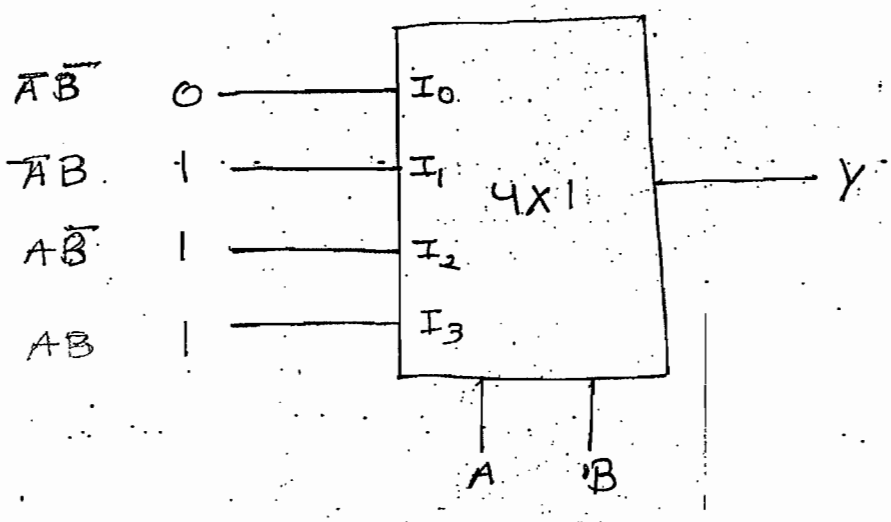


4X1 MUX as Universal circuit :-

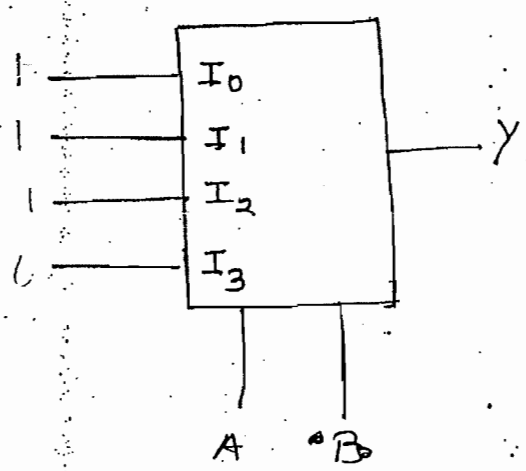
(i) AND :-



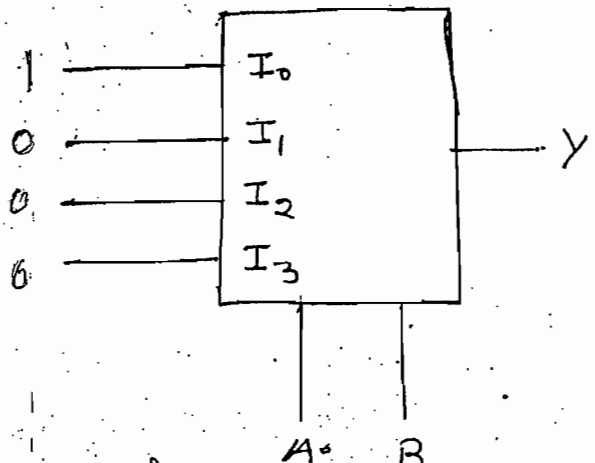
(ii) OR :-



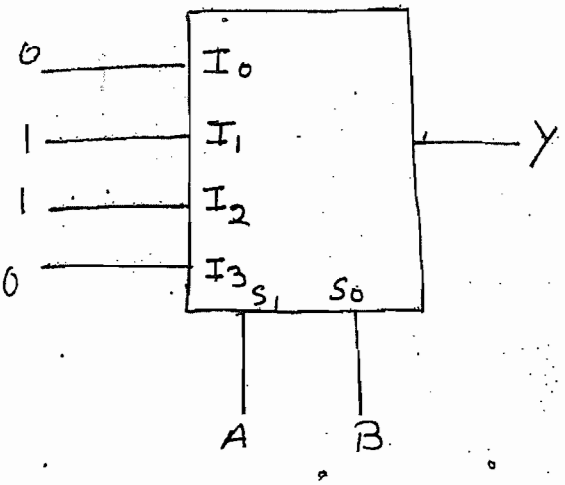
(iii) NAND :-



(iv) NOR :-



(v) EXOR :-



NOTE :-

⇒ HA $\xrightarrow{2}$ 4x1

⇒ HS $\xrightarrow{2}$ 4x1

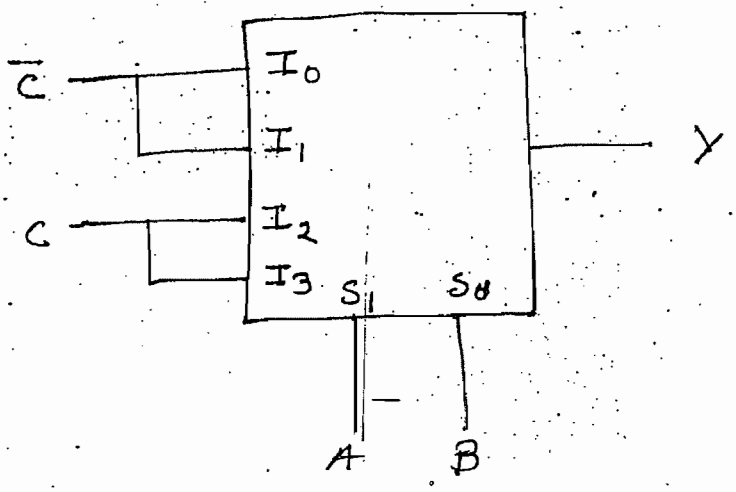
HA, HS } 5 → NAND / NOR

HA, HS } 3 → (2x1) MUX

HA, HS } 2 → (4x1) MUX

3. Determine minimised logical expression for given MUX circuit :-

Ques :-



(a) $A \oplus B$

(b) $A \odot B$

(c) $A \oplus C$

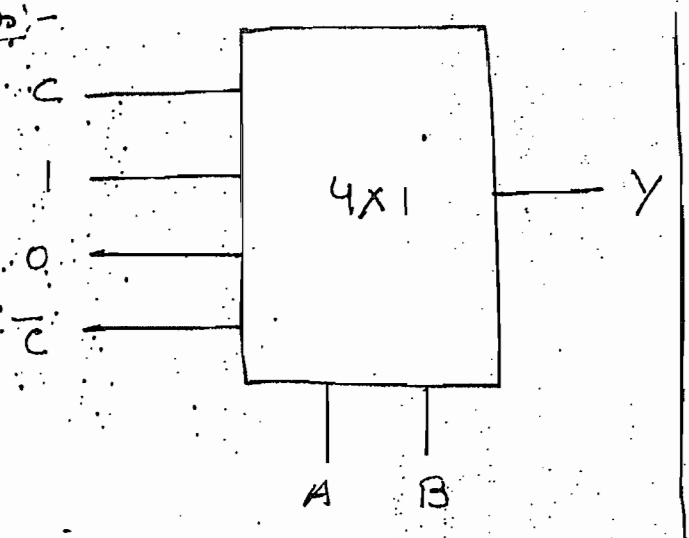
(d) $A \odot C$

Soln :- $Y = \bar{A}B\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$

$= \bar{A}\bar{C} + AC$

$= A \odot C$

Ques:-



Soln:- $Y = \bar{A}\bar{B}C + \bar{A}B + 0 + AB\bar{C}$

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

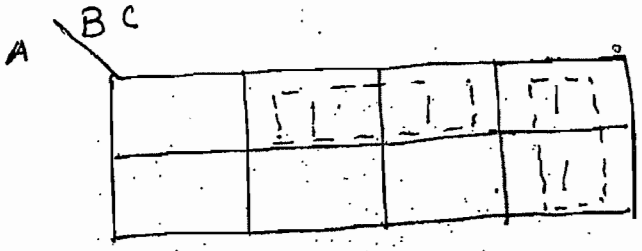
$= \bar{A}\bar{B}C + B(\bar{A}+\bar{C})$

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

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

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

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

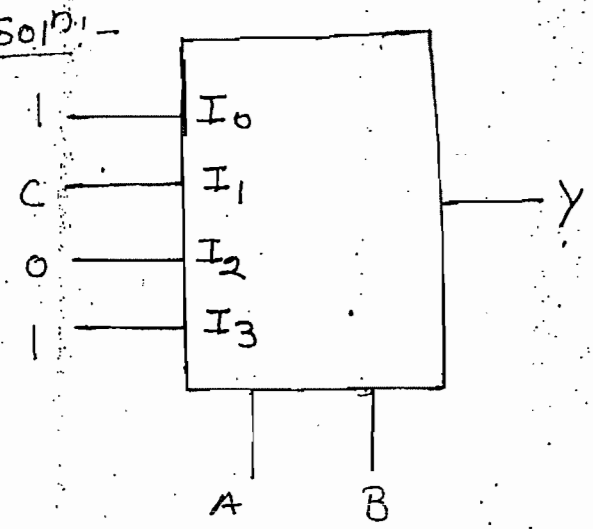


$= \bar{A}C + B\bar{C}$

(4.) Implementation of given logical expression using MUX:-

Ques:- Implement $f(A,B,C) = \Sigma m(0,1,3,6,7)$ Using 4X1 MUX with AB as select line

Soln:-



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

Implementation Table :-

	I_0	I_1	I_2	I_3
$\bar{C}$	(0)	2	4	(6)
C	(1)	(3)	5	(7)
	↓	↓	↓	↓
	1	C	0	1

Ques:- Implement $f(A,B,C) = \Sigma m(1,2,4,5,7)$ using 4:1 MUX with (i) AB as select line
(ii) AC as select line
(iii) BC as select line

Note:-

⇒ Using one 4X1 MUX → any 2 variables
→ some of 3 variable

⇒ Using one 4X1 MUX & 1 NOT gate → Any 2 Variables
→ Any 3 Variable

⇒ Using one 8X1 MUX → Any 3 variable
→ some of 4 variables

⇒ n ⟶ $2^n \times 1$

⇒ n variable ⟶ $2^{n-1} \times 1$ one NO

Questions:-

(i) Sum of FA

(ii) FA using 4X1 MUX

(iii) FA using 8X1 MUX

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

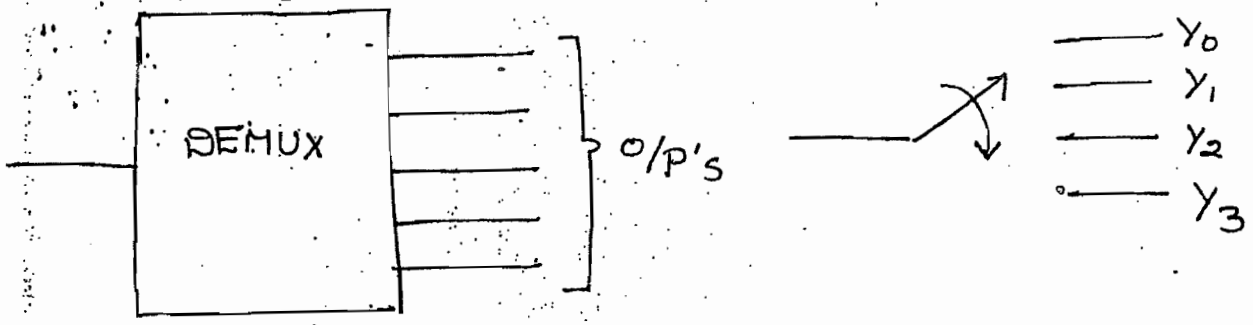
(v) $f(A,B,C) = \pi M(0,1,2,5)$

DEMUX :-

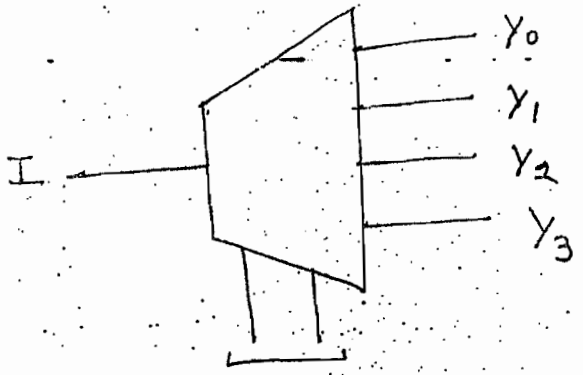
→ It is a combinational circuit having many o/p and one i/p depending on the select control

→ I/p is transferred to one of the o/p line. Hence, it is known as 1 to many circuit. It is also called data distributor

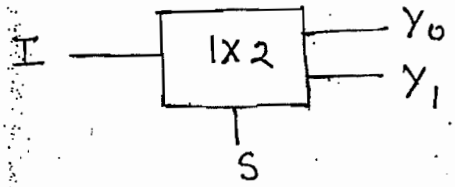
→ It is serial to parallel convertor



Symbol :-



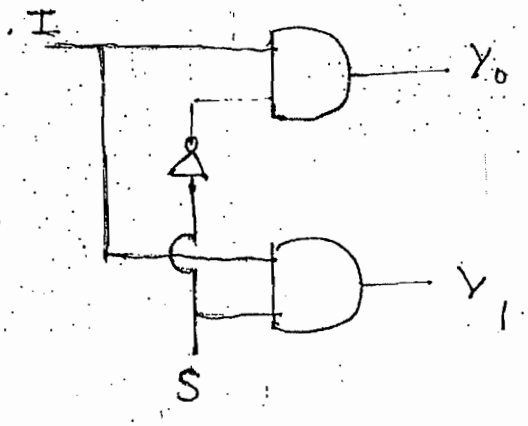
1 : 2 DEMUX :-



S	Y_1	Y_0
0	0	I
1	I	0

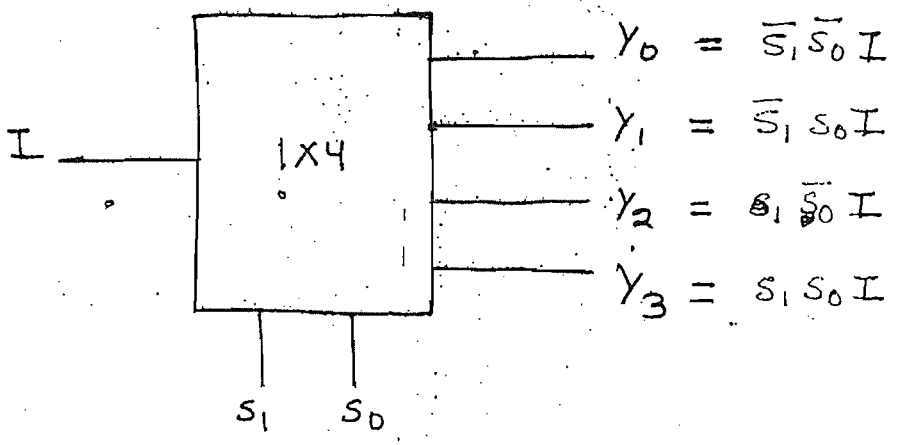
$Y_0 = \bar{S} I$

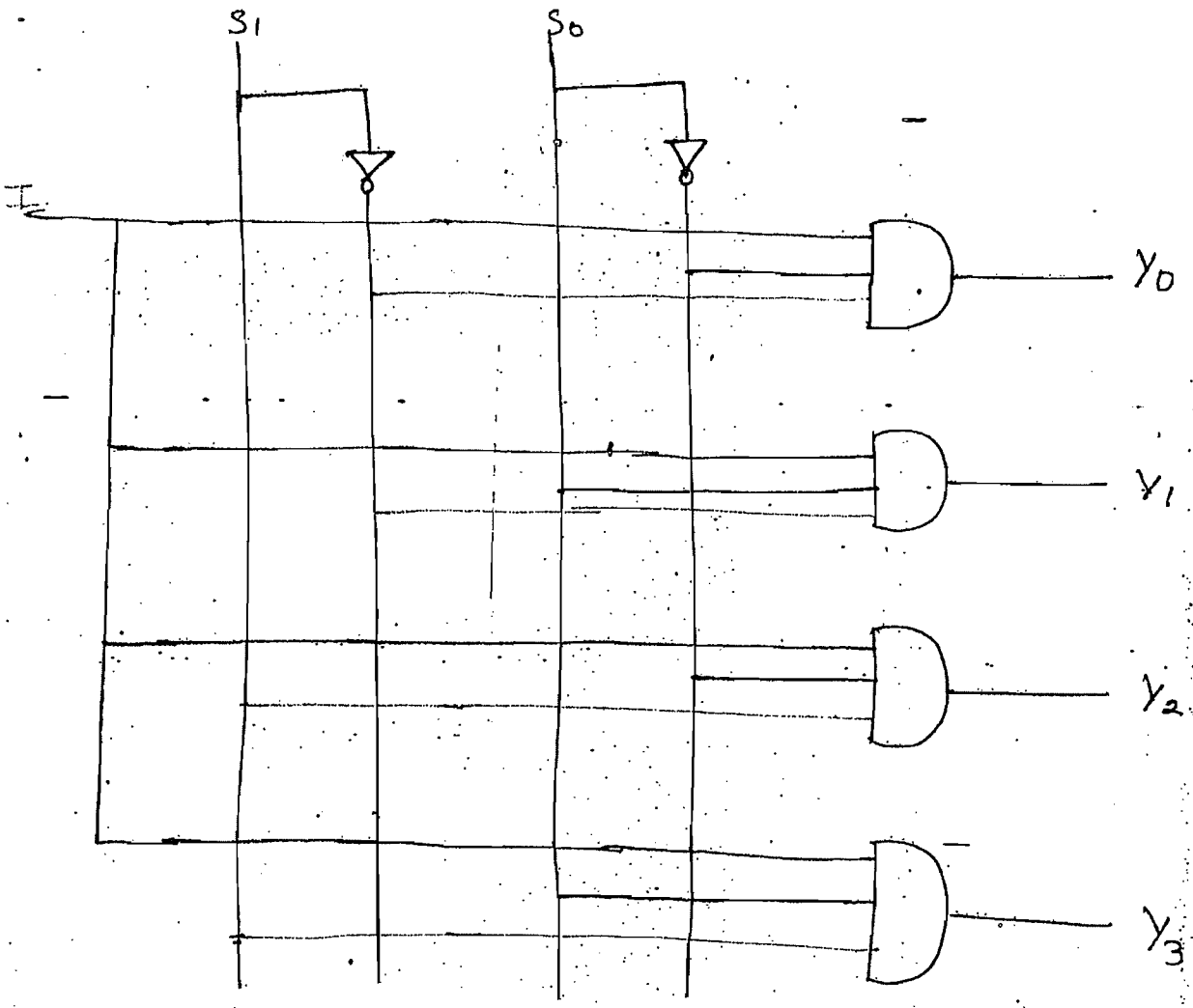
$Y_1 = S I$



→ DEMUX contain AND gate N/w only

1 X 4 DEMUX :-





Higher Order MUX DEMUX using Lower Order :

1 X 4 DEMUX $\xrightarrow{1+2=3}$ 1 X 2

4 X 1 DEMUX $\xleftarrow[(2+1}{3}$ 2 X 1

1 X 8 DEMUX $\xrightarrow{7}$ 1 X 2

8 X 1 $\xleftarrow[4+2+1]{7}$ 2 X 1

$\Rightarrow$ 1×16 — $\frac{5}{1+4}$ — 1×4

$\Rightarrow$ 1×64 — $\frac{21}{1+4+16}$ — 1×4

$\Rightarrow$ 1×64 — $\frac{9}{1+8}$ — 1×8

$\Rightarrow$ 1×256 — $\frac{17}{1+16}$ — 1×16

Decoder :-

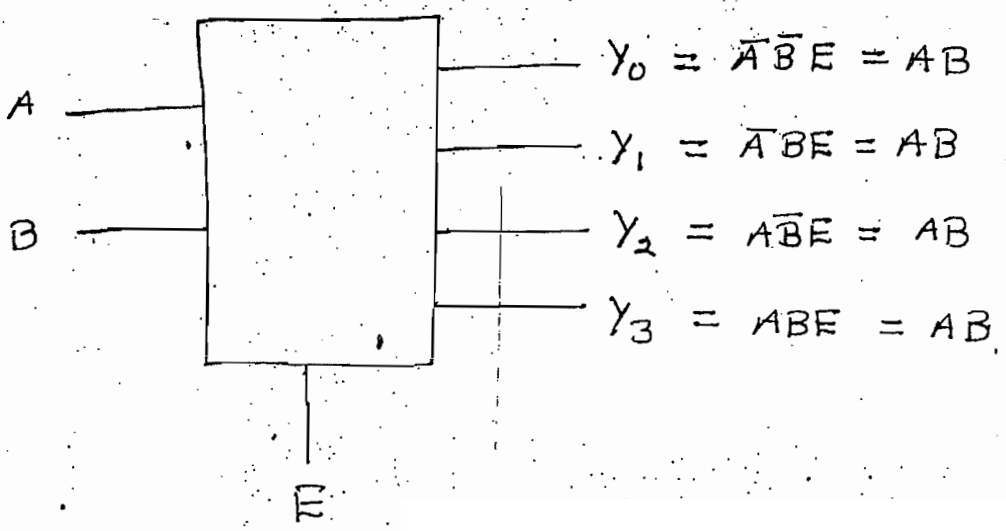
→ Decoder is a combinational circuit which has many data i/p's & many o/p's.

→ Decoder is used to convert binary to other codes such as → Binary to octal (3x8)

→ BCD to decimal

→ Binary to hexadecimal

→ Decoder can be used to decode addl and will generated chip select signal. [$\overline{CS}$, CS — Active low, Active high]



E	A	B	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$Y_0 = \bar{A}\bar{B}E$

$Y_1 = \bar{A}BE$

$Y_2 = A\bar{B}E$

$Y_3 = ABE$

→ Decoder and Demux internal circuit remain same

⇒ 2X4 DEMUX → 2X4 Decoder → 1X4 DEMUX

⇒ 3X8 —— 1X8 DEMUX

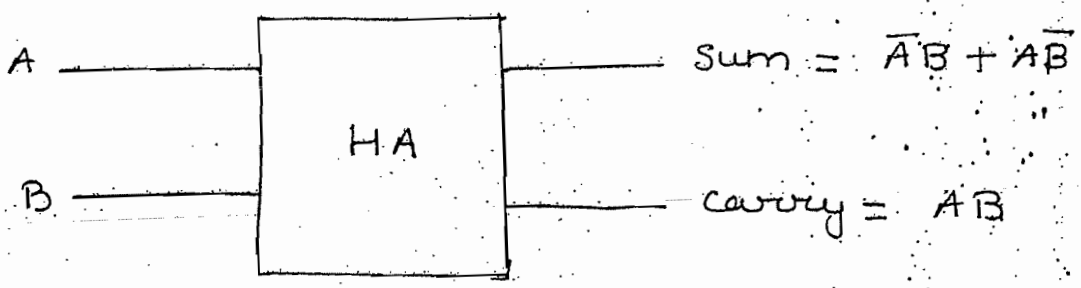
⇒ 4X16 —— 1X16 "

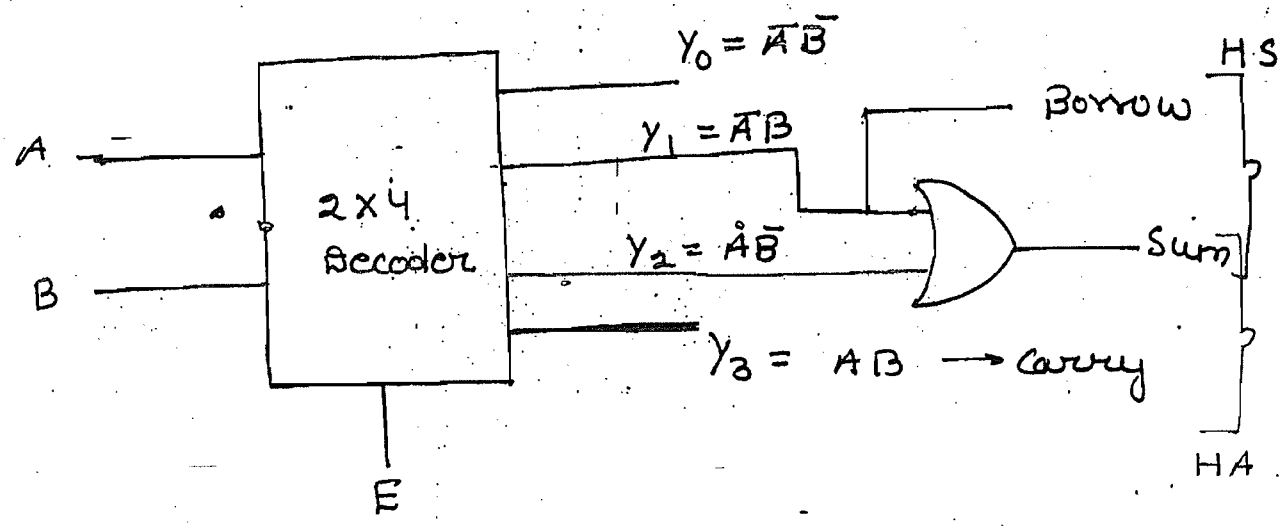
⇒ (8) X 256 —— 1X256 "

↓

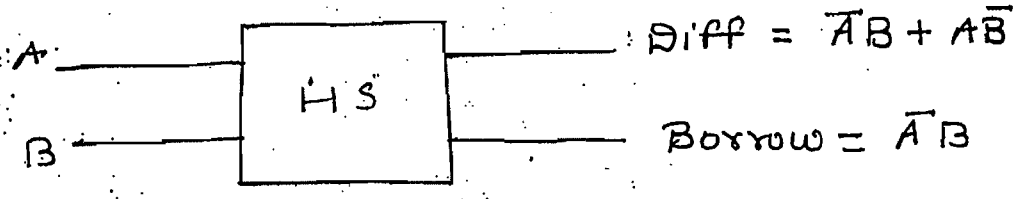
select line → 8

Implement HA using 2X4 decoder circuit :-





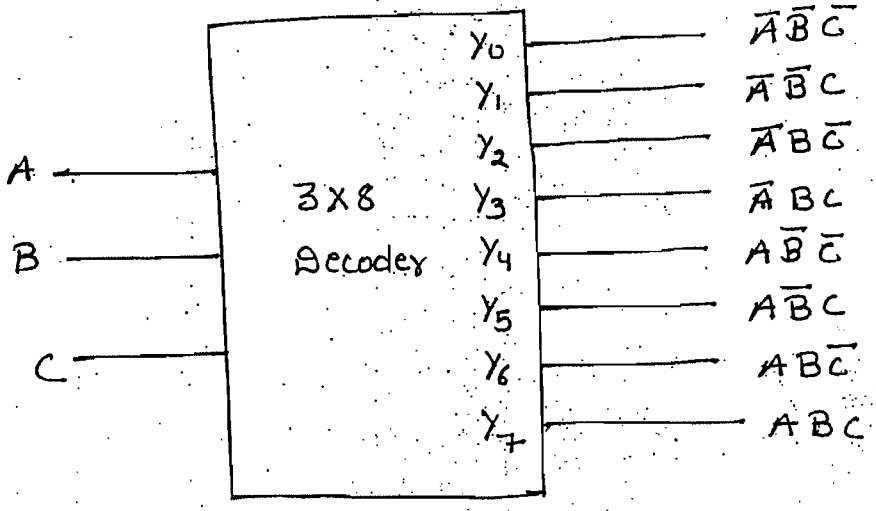
Implement HS :—



→ To implement HA one 2X4 and 1 or gate is required

→ To implement Both Subtract / Adder only 1 circuit with 2X4 decoder and 1 OR gate.

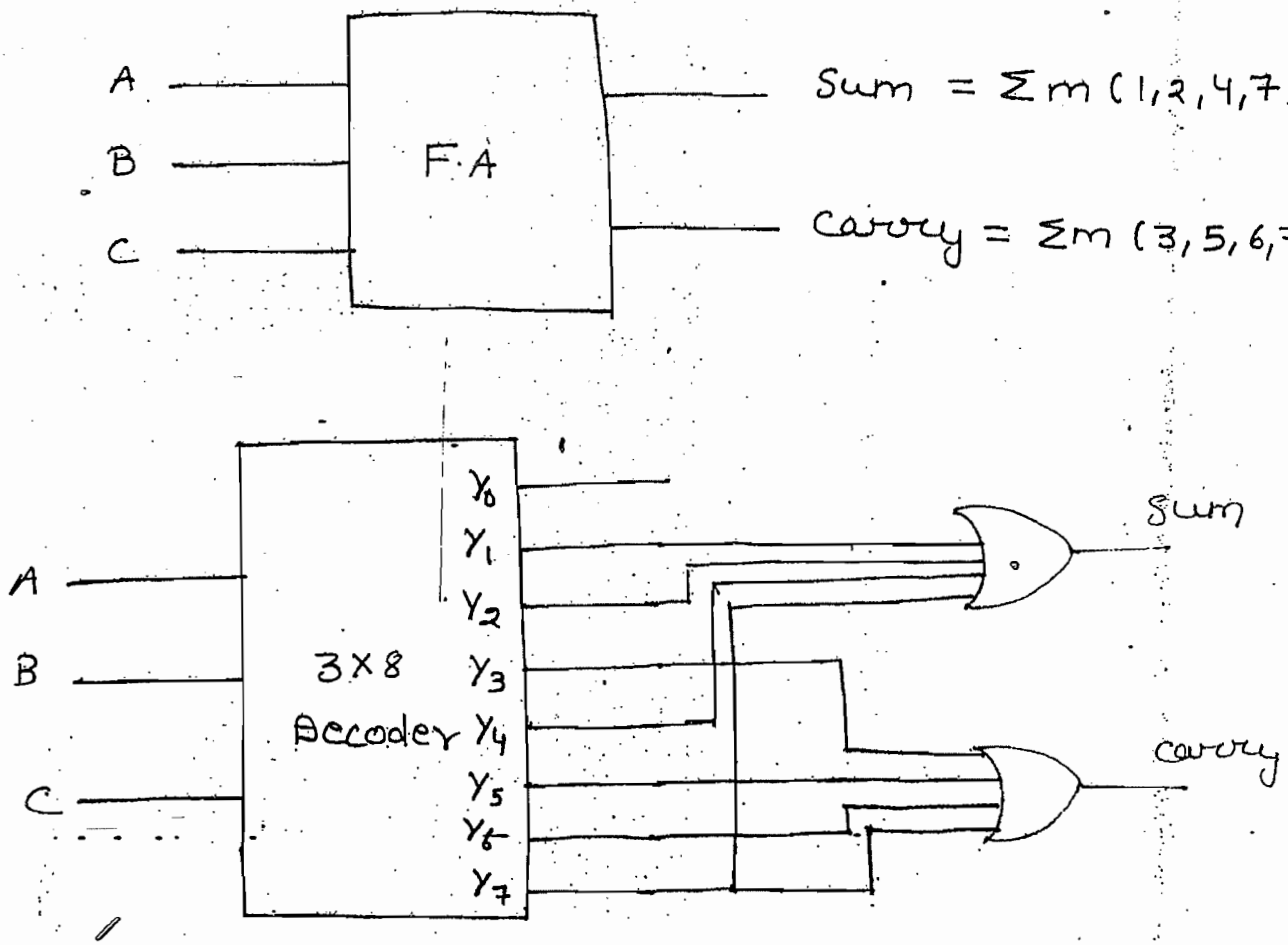
3X8 Decoder :—



Binary to Octal :—

A	B	C	Y
0	0	0	0
0	0	1	1
⋮			
1	1	1	7

Implement FA circuit using 3X8 Decoder:-



Questions:-

Implement

(I) FS using 3X8 decoder

(II) $f(A,B,C) = \Sigma m(1,2,4,5,7)$

(III) $f(A,B,C) = \bar{A} + BC$

Ques:- 4X16 decoder any code conversion.

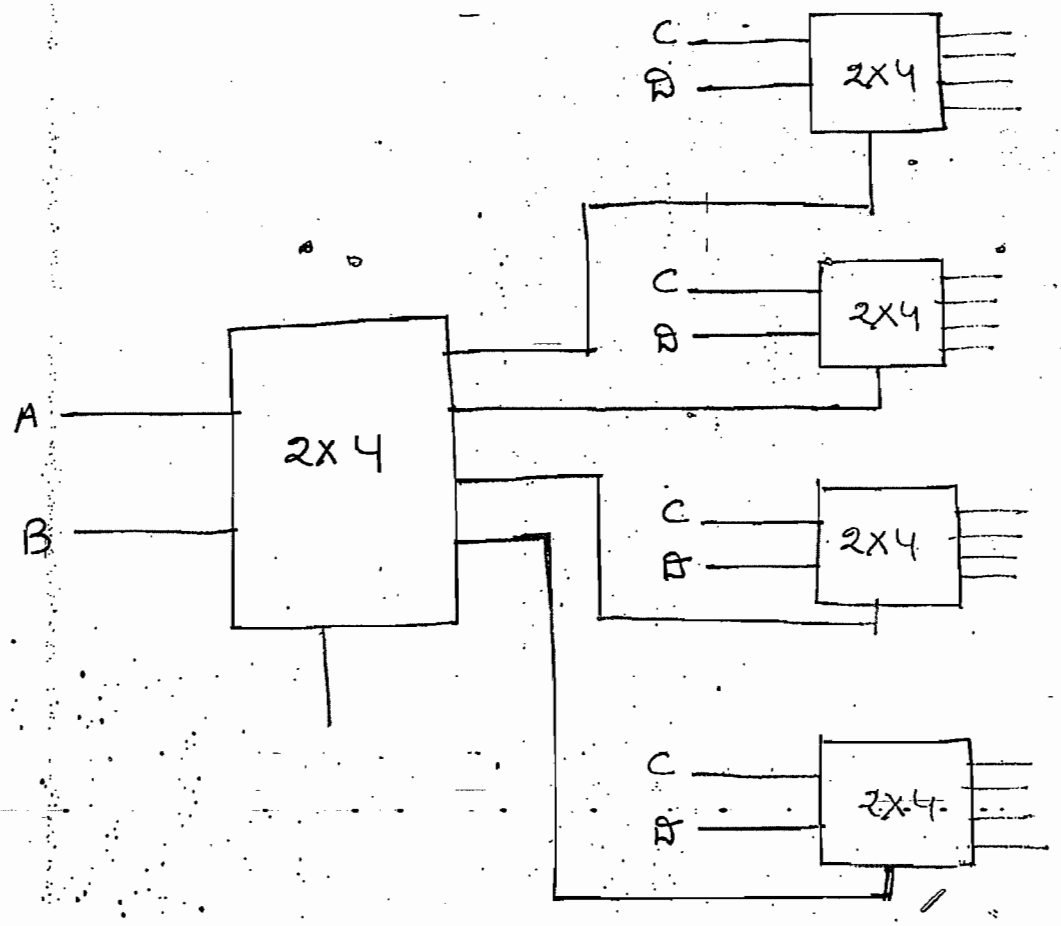
Implementation of Higher order decoders using lower order decorders:-

4 X 16 ↓ $\xrightarrow{5(1+4)}$ 2 X 4 ↓

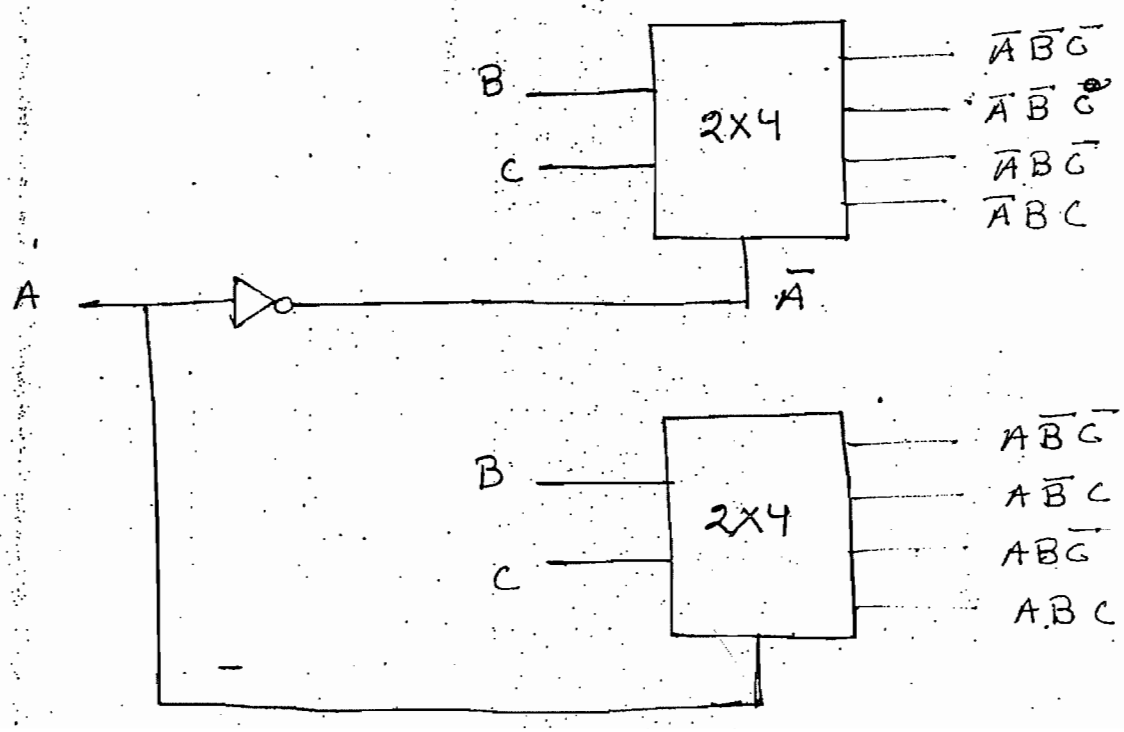
1 X 16 $\xrightarrow{5(1+4)}$ 1 X 4

16 X 1 $\xrightarrow[(4+1)]{5}$ 4 X 1

→ 4X16 using 2X4 :-



=> 3X8 Using 2X4 :-



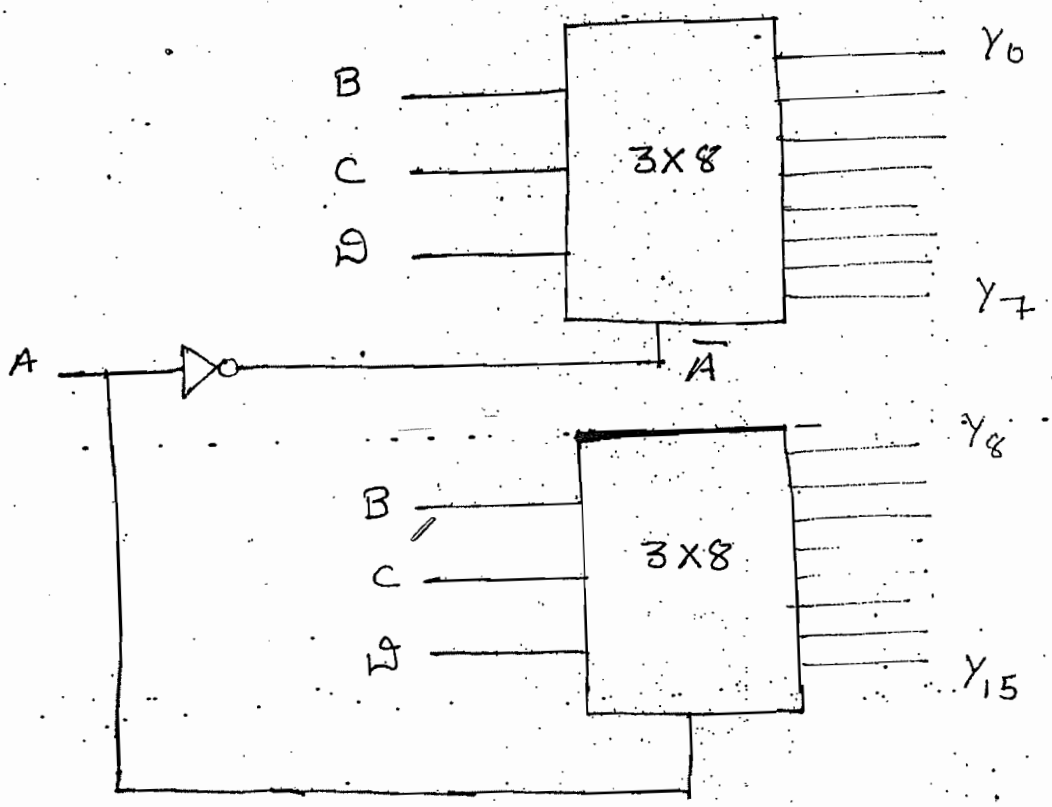
⇒ 16 X 64 $\xrightarrow{21\ (1+4+16)}$ 2 X 4 ↓

1 X 64 $\xrightarrow{21}$ 1 X 4

⇒ 8 X 256 $\xrightarrow{17\ (1+16)}$ 4X16

→ Decoder / DEMUX → IC same

4 X 16 Using 3X8 :-



ENCODER :-

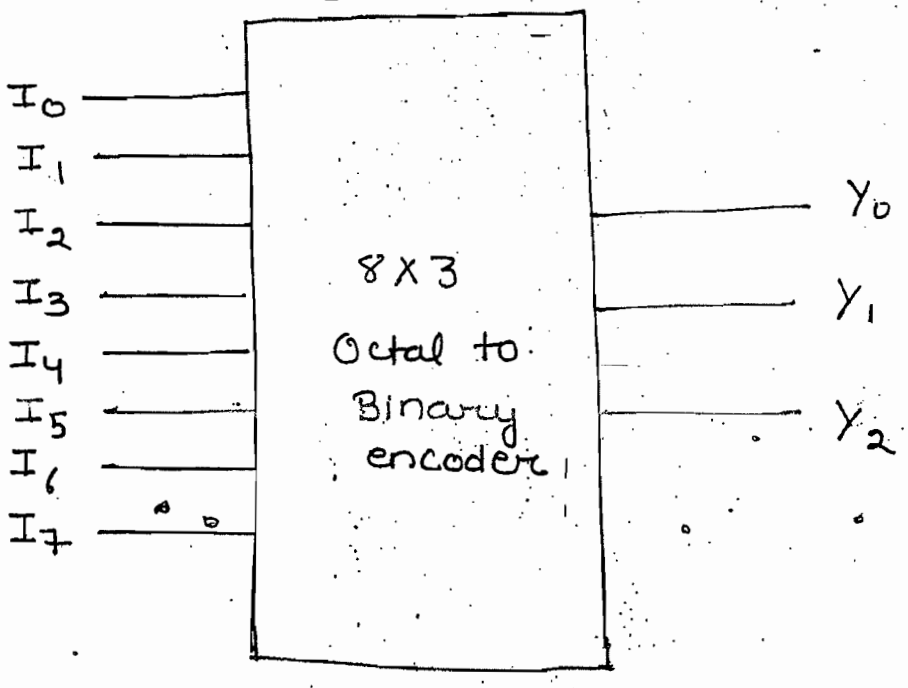
→ It is a combinational circuit which have man i/p & many o/p

→ It is used to convert other codes to binary such as → octal to binary (8X3)

→ Decimal to BCD (10X4)

→ Hex to Binary (16X4)

Octal to Binary Encoder :- / 8 X 3 Encoder :-



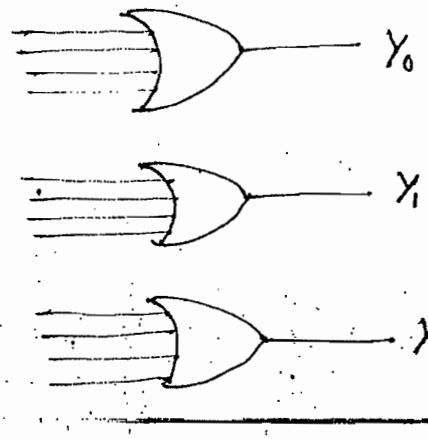
→ In encoder one of the i/p line is high and corresponding binary is available at the o/p

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	0	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

$Y_0 = I_1 + I_3 + I_5 + I_7$

$Y_1 = I_2 + I_3 + I_6 + I_7$

$Y_2 = I_4 + I_5 + I_6 + I_7$

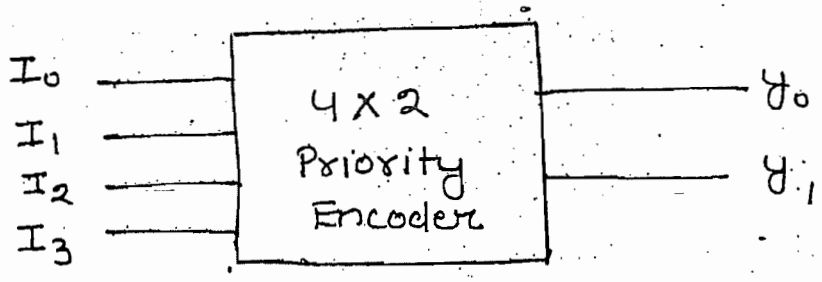


→ Encoder basic circuit is OR-gate

Encoder

Priority Encoder :-

→ In priority encoder, any no of i/p can be logic 1 but binary is available corresponding to highest priority i/p line at o/p

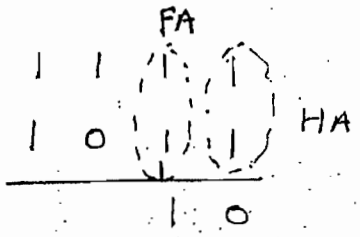


I_3	I_2	I_1	I_0	y_1	y_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

$$y_1 = I_3 + \overline{I_3} I_2 = I_3 + I_2$$

$$y_0 = I_3 + \overline{I_3}\,\overline{I_2} I_1 = I_3 + \overline{I_2} I_1$$

Adder :-



In order to add gp of bits following adders are used.

(i) Serial adder

ii) Parallel adder

iii) Look ahead adder

(i) Serial Adder :-

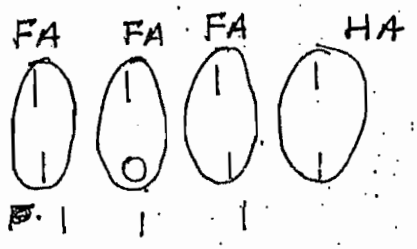
→ In serial adder only 1 full adder is used to add any no of bits

→ In this shift registers are used to provide data to full adder

→ In serial adder to provide n-bit addition it requir min. n clock pulse

→ serial adder is slowest adder

(ii) Parallel Adder :-



→ In parallel adder each bit require separate adder circuits to provide n-bit addition in parallel adder, it requires (n-1) FA and 1 HA

OR

n FA

(2n-1) HA and (n-1) OR Gates

4-bit Parallel adder circuit / Ripple carry Adder :-

A_3 A_2 A_1 A_0

B_3 B_2 B_1 B_0 C_0

C_3 C_2 C_1

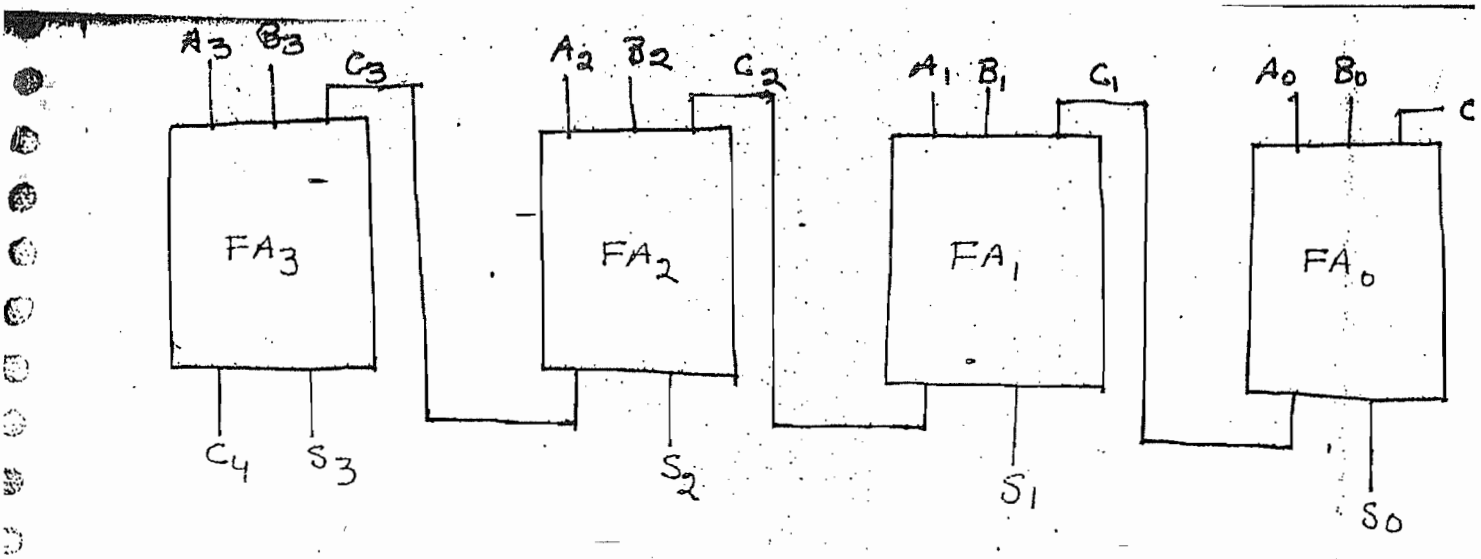
S_3 S_2 S_1 S_0

1 1 0 1

1 0 1 1 0

1 1 1 1

1 1 0 0 0



$C_4\ S_3\ S_2\ S_1\ S_0$

→ In parallel adder each full adder will provid 2tpd delay to provide sum or carry o/p.

→ In n-bit parallel adder to provide result it require 2Ntpd delay

→ In this carry propagation delay is present from i/p to o/p

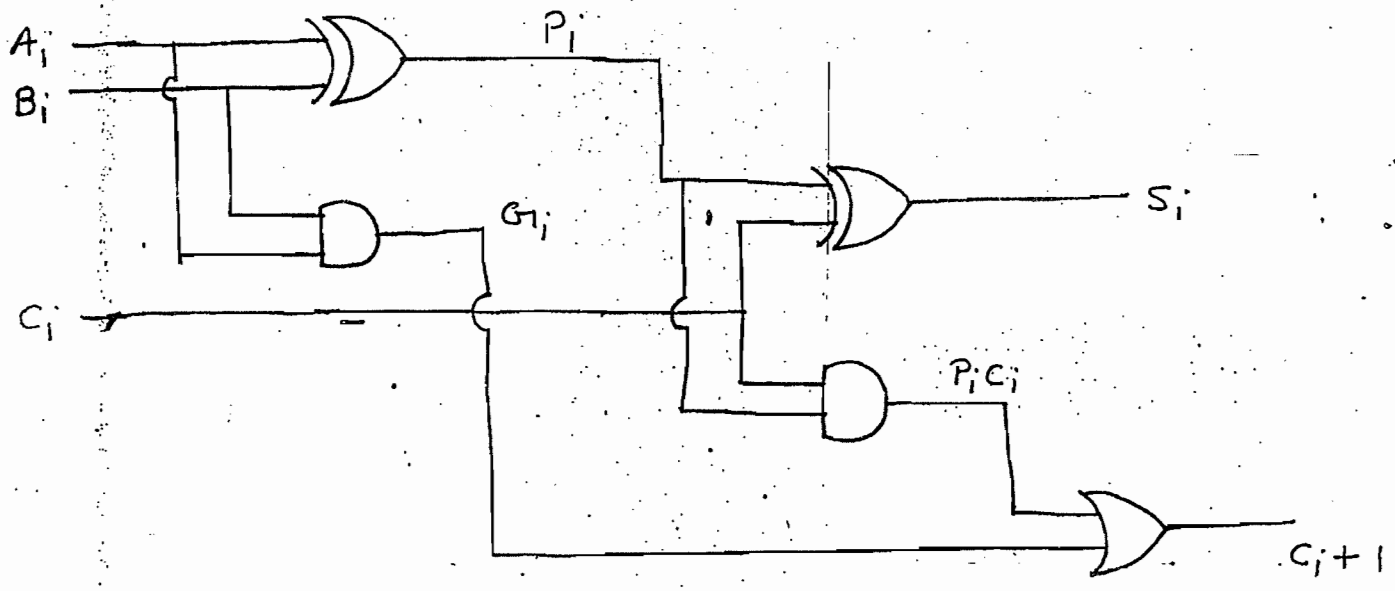
→ As no. of bits are increase, carry propagation delay's increase. Hence it is known as ripple carry adder circuit

→ Its disadvantage is as no. of bits are increase, speed of operation decrease

Look Ahead Carry Circuit:-

→ To remove the disadvantage of parallel adder, look ahead carry circuit is designed.

4-Bit Look ahead carry adder circuit:-



G_i = Carry generator term P_i → Carry Propagation term

$$P_i = A_i \oplus B_i \qquad G_i = A_i \cdot B_i \qquad S_i = P_i \oplus C_i$$

$$C_{i+1} = P_i C_i + G_i$$

$P_0 = A_0 \oplus B_0$	$G_0 = A_0 \cdot B_0$	$S_0 = P_0 \oplus C_0$
$P_1 = A_1 \oplus B_1$	$G_1 = A_1 \cdot B_1$	$S_1 = P_1 \oplus C_1$
$P_2 = A_2 \oplus B_2$	$G_2 = A_2 \cdot B_2$	$S_2 = P_2 \oplus C_2$
$P_3 = A_3 \oplus B_3$	$G_3 = A_3 \cdot B_3$	$S_3 = P_3 \oplus C_3$

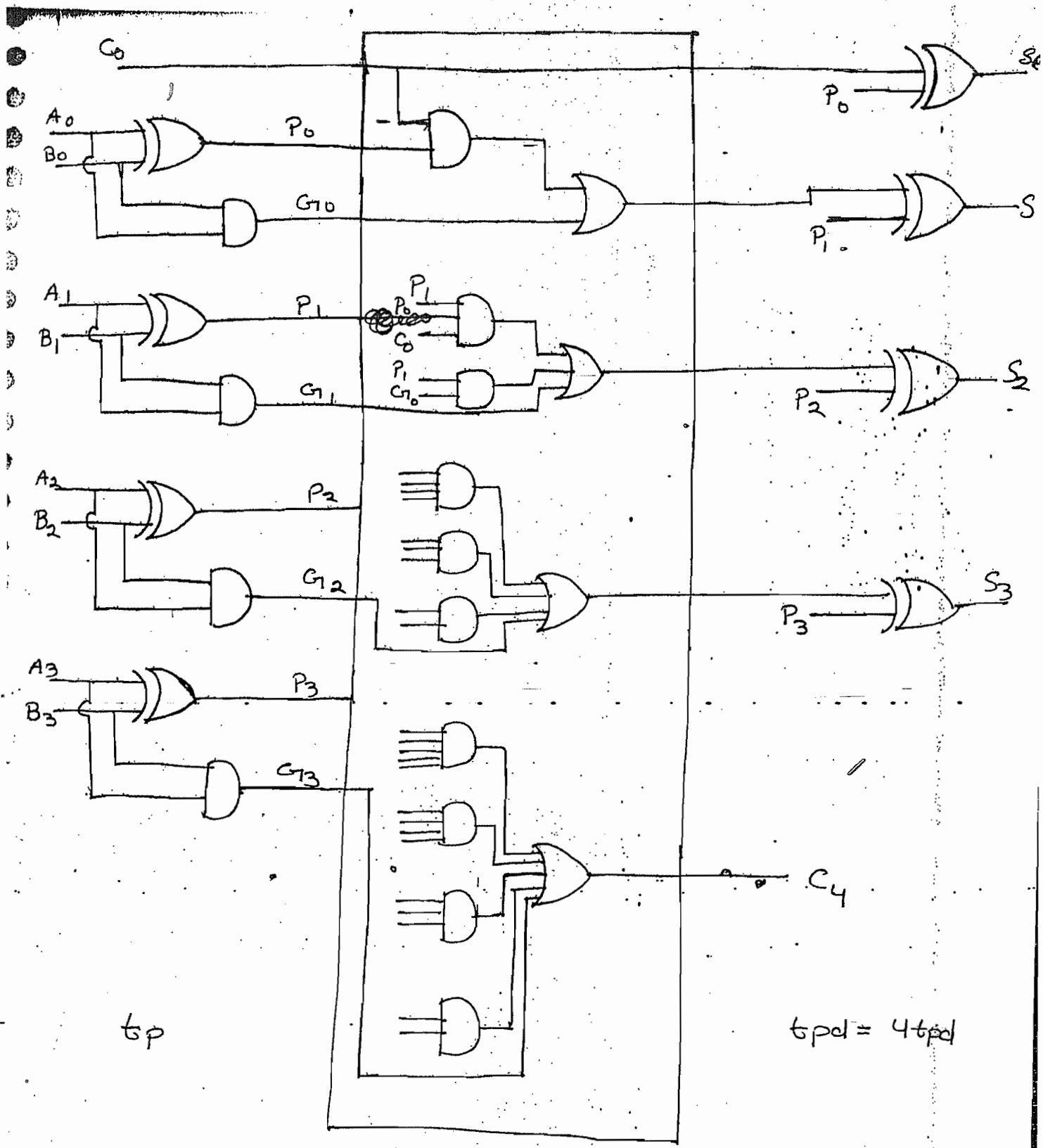
C_0 → input carry

$C_1 = P_0 C_0 + G_0$

$C_2 = P_1 C_1 + G_1 = P_1 (P_0 C_0 + G_0) + G_1 = P_1 P_0 C_0 + P_1 G_0 + G_1$

$C_3 = P_2 C_2 + G_2 = P_2 P_1 P_0 C_0 + P_2 P_1 G_0 + P_2 G_1 + G_2$

$C_4 = P_3 C_3 + G_3 = P_3 P_2 P_1 P_0 C_0 + P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3$



t_p

$t_{pd} = 4t_{pd}$

2-Logic levels

→ In look ahead carry adder, carry generated by 2-level AND-OR N/w

→ In look ahead carry N/w, no. of AND-gate used

$$= \frac{n(n+1)}{2}$$

→ No. of OR gates $= n$

→ If all logic gates have same tpd (AND, OR, EXOR), then to provide carry, delay = 3tpd
& to provide sum, delay = 4tpd

→ Propagation is independent of No. of bits and depends on no. of level.

→ LAC is fastest adder

→ In this O/P is generated directly from I/P bits

Sequential circuits

Flip Flops :-

→ Basic memory element

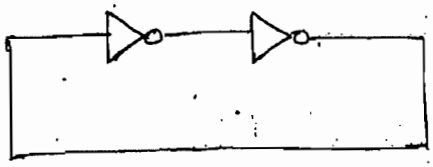
→ store 1 bit

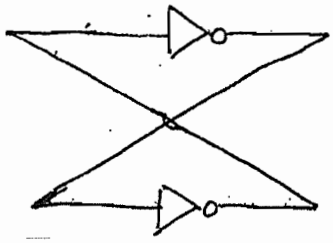
→ Having two stable state. Hence it is known as bistable multivibrator

→ Having two outputs which are complementary to each other

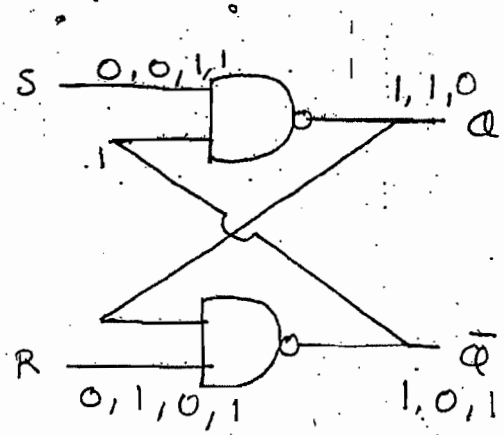
SR Latch :-

(I) Using NOT Gate :-





(II) Using NAND Gate :-



$\bar{S}$	R	Q
0	0	Invalid
0	1	1
1	0	0
1	1	Previous State

→ Previous state ($Q = 0$, $\bar{Q} = 1$ OR $Q = 1$, $\bar{Q} = 0$)

Invalid $Q = 1$, $\bar{Q} = 1$ → NAND

Note :-

→ In SR latch if both gates are disable then O/P is in invalid state

→ If both gates are enable then O/P remains in previous state, if previous state is valid state.

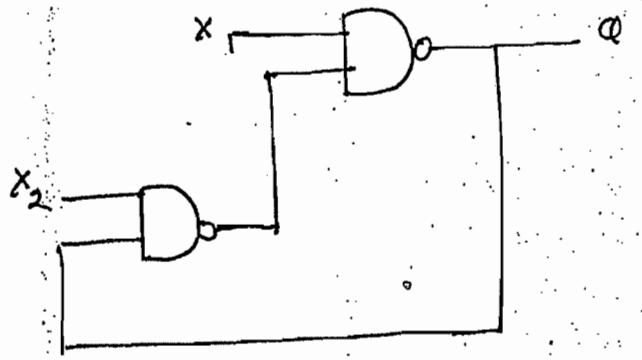
→ If both gates are enable, o/p of SR latch is in

$Q = 0, \bar{Q} = 1$

OR

$Q = 1, \bar{Q} = 0$

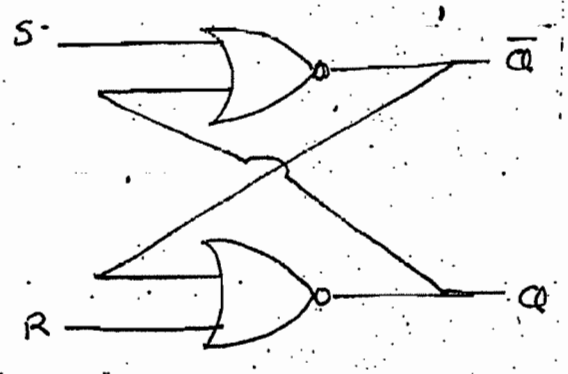
Q:- The circuit shown in figure is NAND latch if i/p $X_1 = 0$, $X_2 = 1$ applied then o/p Q will be



(a) logic 0
(b) logic "1"
(c) Invalid
(d) Previous state

Ans - b.

(III) Using NOR :-



S	R	Q
0	0	Prev. State
0	1	0
1	0	1
1	1	Invalid

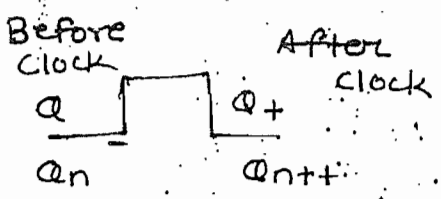
→ Invalid ($Q = 0, \bar{Q} = 0$) → For NOR

Application of SR Latch :-

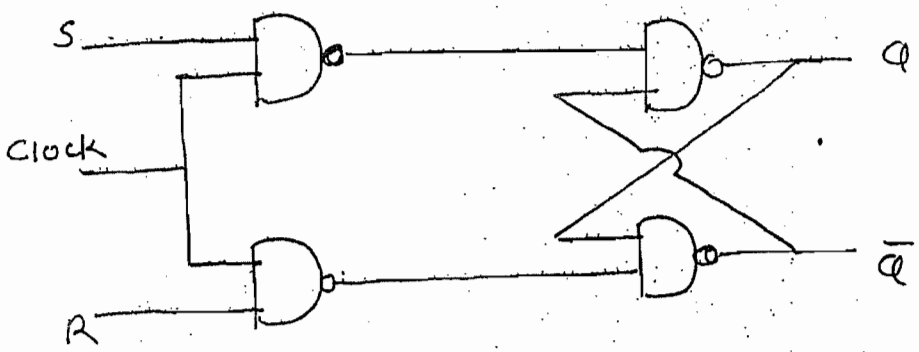
→ SR latch is mainly used as memory element in digital circuit.

→ To eliminate bounce in switches or keyboard SR latch is used

SR FF :-



(I) Using NAND :-

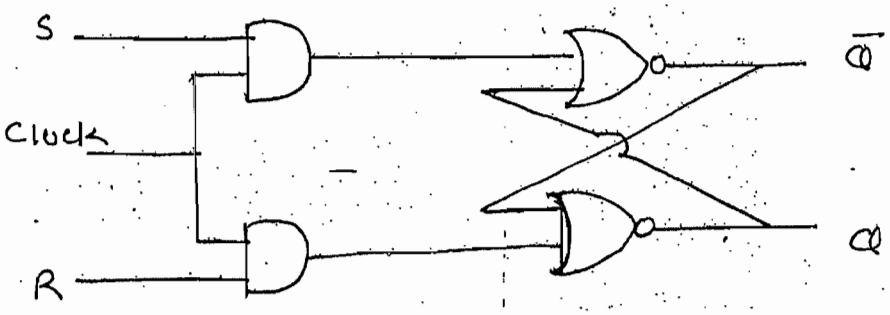


Clock	S	R	Q_{n+1}	
0	X	X	Q_n	→ Memory
1	0	0	Q_n	→ HOLD
1	0	1	0	→ Reset
1	1	0	1	→ set
1	1	1	Invalid	→ Unused state

Note :-

→ The purpose of two NAND gate (first) is to inverse the i/p

(II) Using NOR Latch :-



Clock	S	R	Q_{n+1}	
0	X	X	Q_n	→ Memory
1	0	0	Q_n	→ HOLD
1	0	1	0	→ Reset
1	1	0	1	→ Set
1	1	1	Invalid	→ Unused state

Characteristic Table :-

S	R	Q_n	Q_{n+1}	
0	0	0	0	} HOLD (Q_n)
0	0	1	1	
0	1	0	0	} Reset
0	1	1	0	
1	0	0	1	} Set
1	0	1	1	
1	1	0	X	
1	1	1	X	

Characteristic Equation :- (For O/P finding)

$$Q_{n+1} = \Sigma m(1,4,5) + \Sigma d(6,7)$$

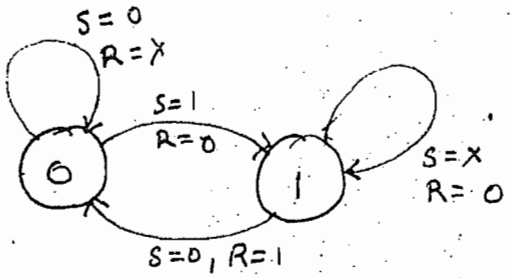
S \ RQ_n	$\bar{R}\bar{Q}_n$	$\bar{R}Q_n$	RQ_n	$R\bar{Q}_n$
$\bar{S}$		1		
S	1	1	X	X

$$\boxed{Q_{n+1} = S + \bar{R}Q_n} \rightarrow S \cdot R \neq 1$$

Excitation Table :- (For I/P finding)

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

State Diagram :-



Disadvantage :-

→ Invalid state is present when S and R i/p is logic 1 To avoid this JK ff is used.

JK FF :-

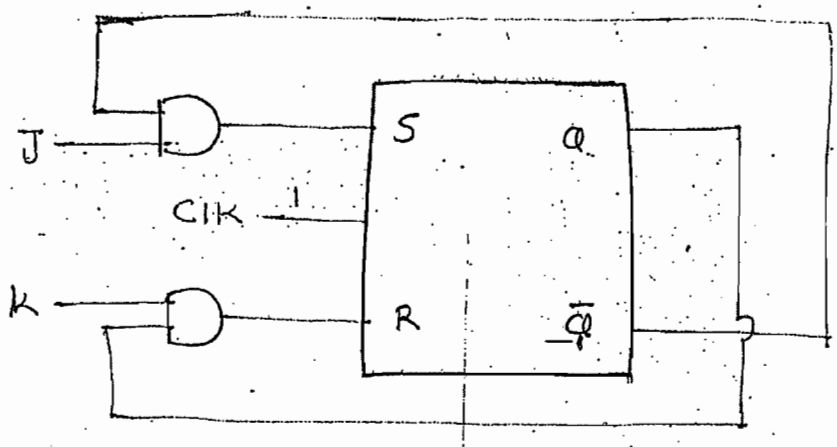
J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$ → Toggle

SR to JK :-

**

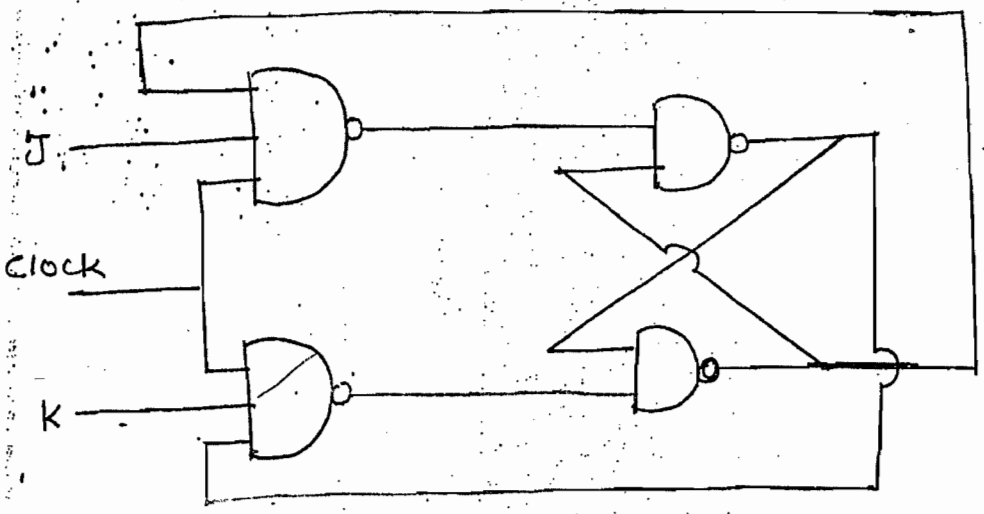
$S = J\overline{Q}$

$R = KQ$

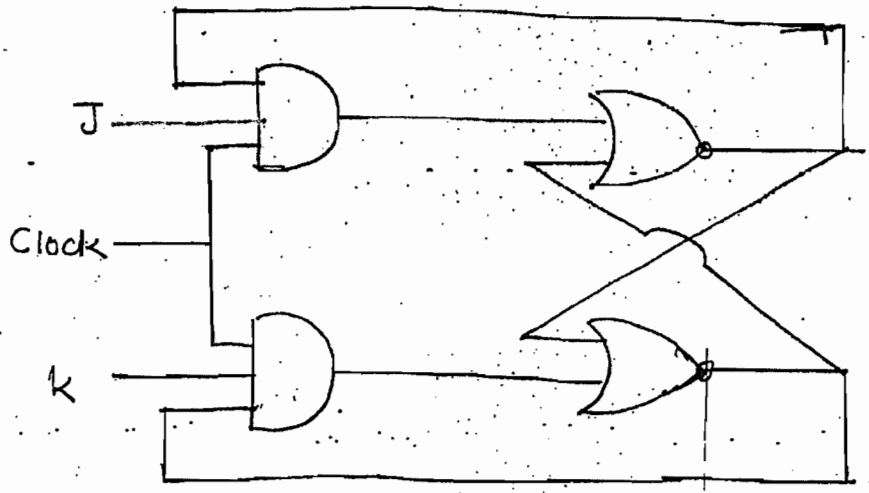


Clock	J	K	Q_{n+1}
0	X	X	Q_n
1	0	0	Q_n
1	0	1	0
1	1	0	1
1	1	1	$\overline{Q_n}$

JK FF Using NAND Gate :-



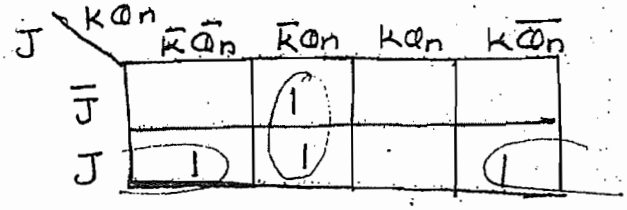
JK FF Using NOR Gate :-



Characteristic Table :-

J	K	Q_n	Q_{n+1}	
0	0	0	0	} H
0	0	1	1	
0	1	0	0	} R
0	1	1	0	
1	0	0	1	} S
1	0	1	1	
1	1	0	1	} Toggle.
1	1	1	0	

Characteristic Equation :-

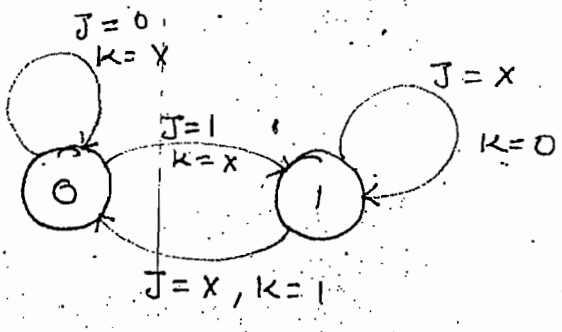


$$Q_{n+1} = J\bar{Q}_n + \bar{K}Q_n$$

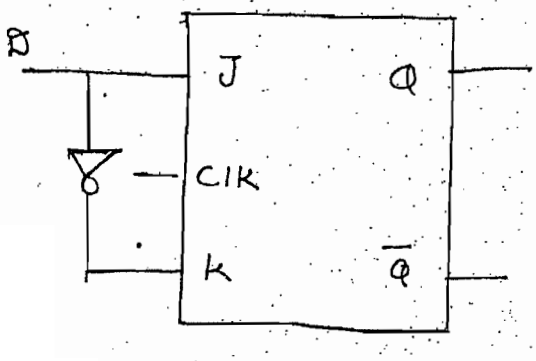
Excitation Table :-

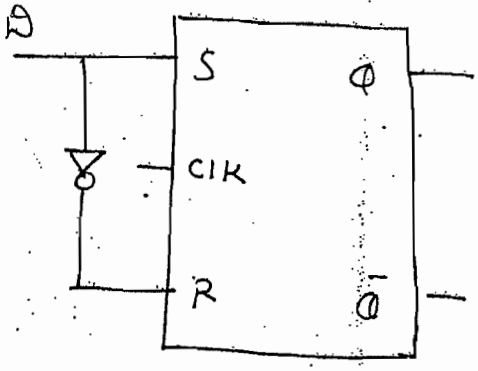
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

State Diagram :-



D FF :-





Clock	D	Q_{n+1}
0	X	Q_n
1	0	0
1	1	1

D	Q_{n+1}
0	0
1	1

Characteristic Table :-

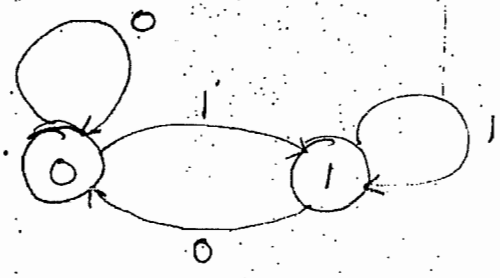
D	Q_n	Q_{n+1}
0	0	0
0	1	0
1	0	1
1	1	1

$Q_{n+1} = D$ → Delay FF (or) Data
→ Transparent Latch

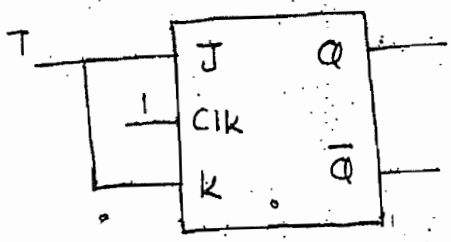
Excitation Table :-

Q_n	Q_{n+1}	D
0	0	0
0	1	1
1	0	0
1	1	1

State Diagram :-



T FF :-



$J = K = T$

Clock	T	Q_{n+1}	
0	X	Q_n	→ H
1	0	Q_n	→ H
1	1	$\bar{Q}_n$	→ T

T	Q_{n+1}
0	Q_n
1	$\bar{Q}_n$

Characteristic Table :-

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0

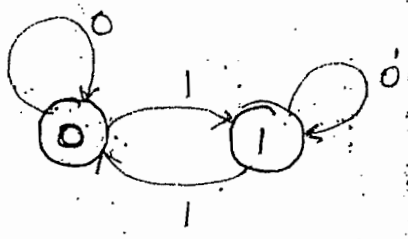
Characteristic Equation :-

$$Q_{n+1} = \bar{T} Q_n + T \bar{Q}_n$$

Excitation Table :-

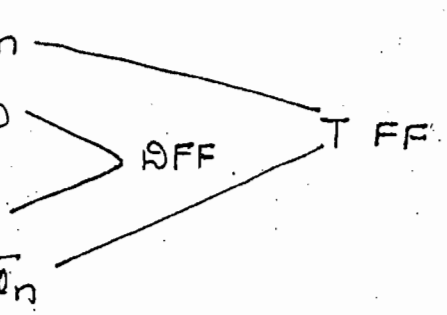
Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

State Diagram :-



Note:-

	J	K	Q_{n+1}
SR FF	0	0	Q_n
	0	1	0
	1	0	1
	1	1	$\bar{Q}_n$



$Q_{n+1} = S + \bar{R} Q_n$

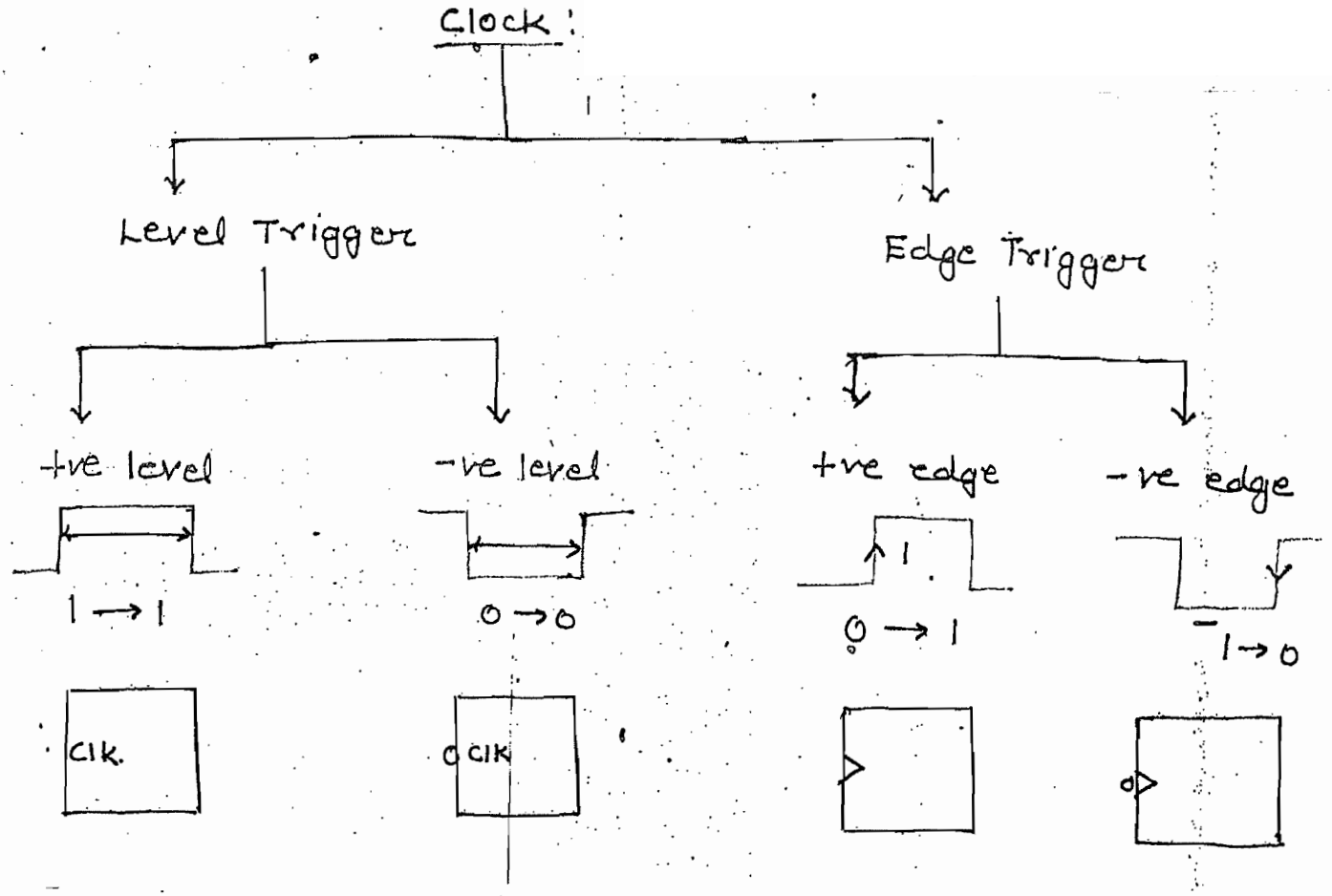
$\therefore Q_{n+1} = J\bar{Q}_n + \bar{K} Q_n$

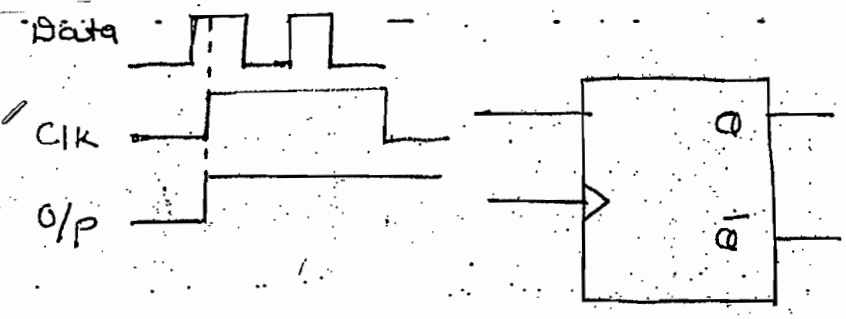
$Q_{n+1} = D$

$Q_{n+1} = T \oplus Q_n$

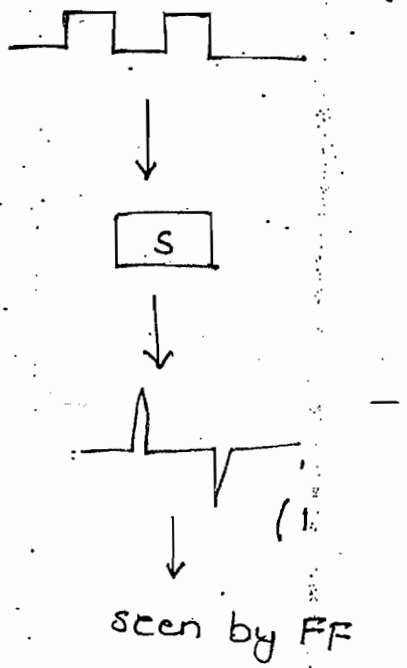
Q_n	Q_{n+1}	S	R	J	K	D	T
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0

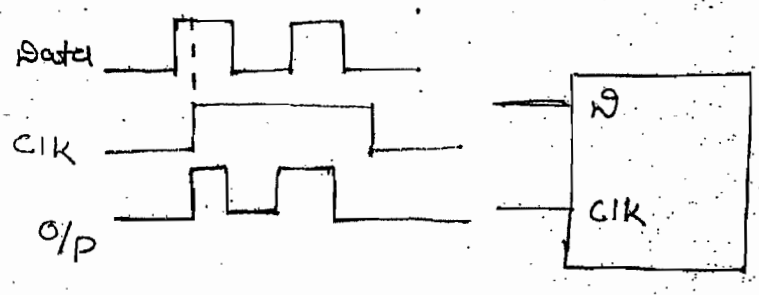
Clock :

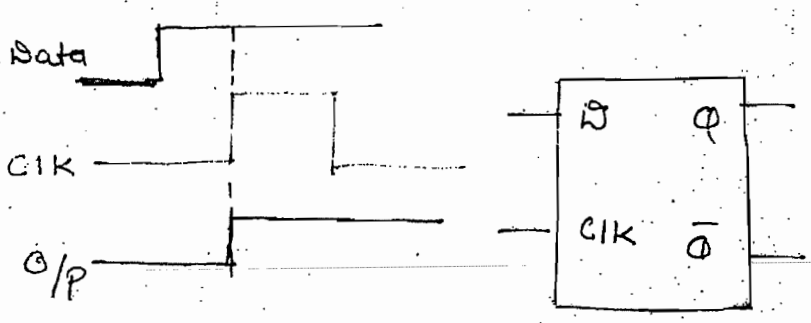




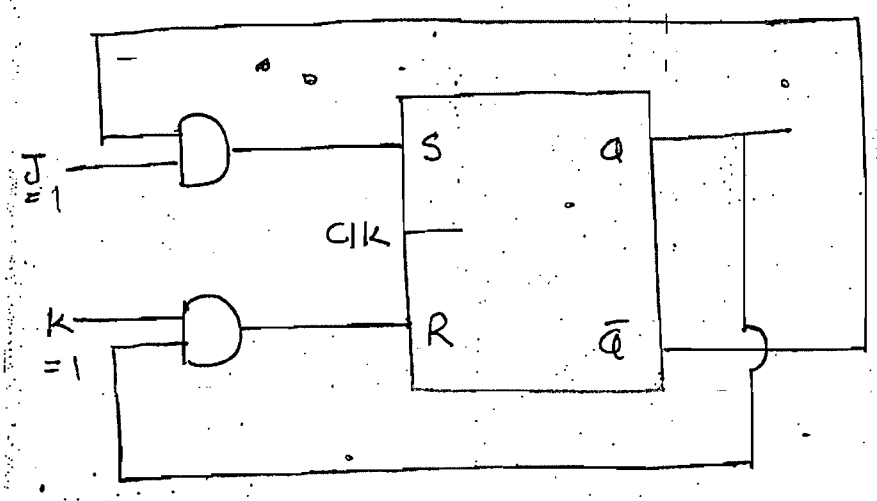
Level Triggering :-

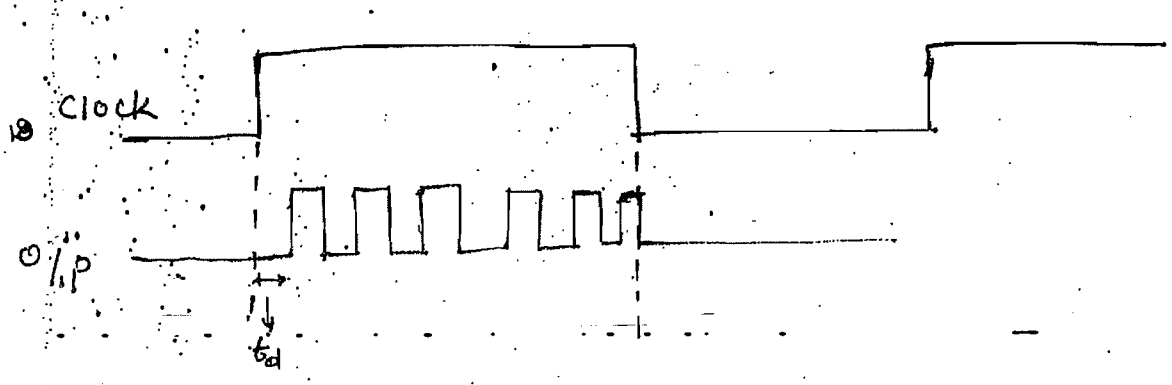






Race Around Condition :-





→ Disadvantage of JK FF is race-around

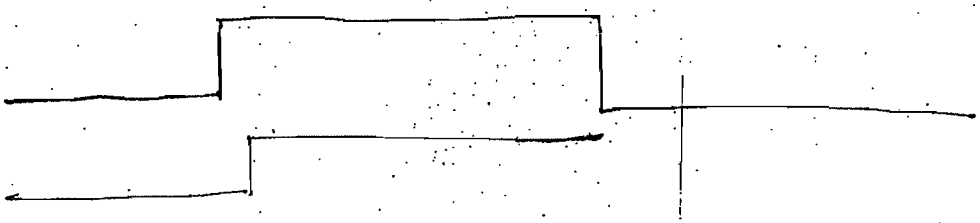
→ Race around occurs when

(i) $J = K = 1$ (ii) t_{pd} FF < t_{pw} (Pulse width delay)
(FF delay)

→ During race around O/P will changes many times in single clock

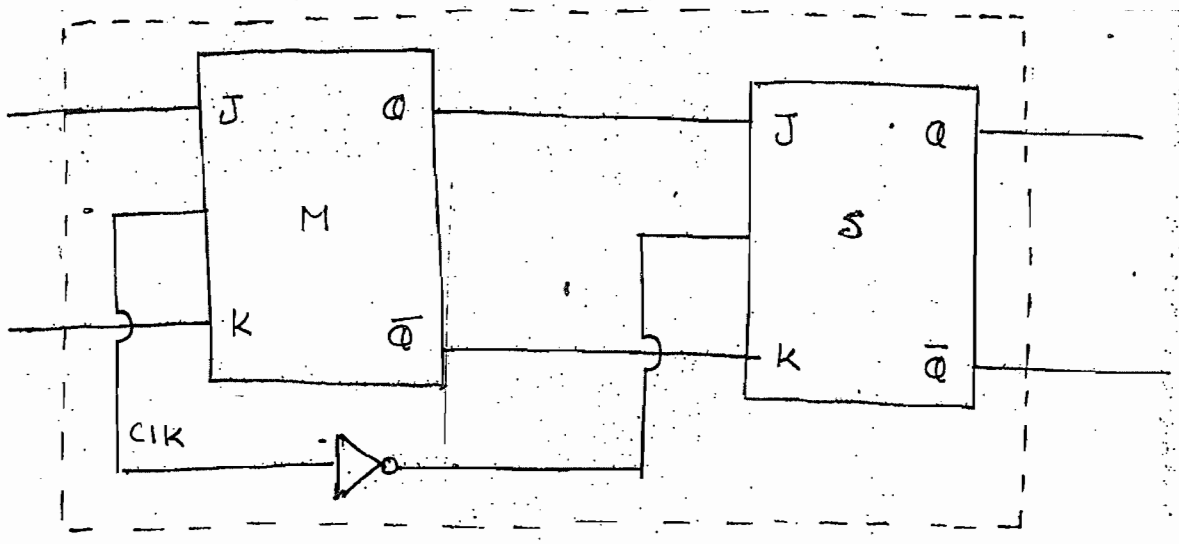
Condition to avoid Race Around :-

$$t_{pw} < t_{pd}\ FF < T_{CLK}$$



→ To avoid race around condition Master-Slave FF is used

Master Slave FF :-



→ In MS FF shown in figure, Master is applied with i/p clock and slave is applied with inverted clock. Due to this Master and slave will not change at a time.

→ O/p of MS FF change when slave O/p is changing

→ As for operation master is level triggered and slave is edge triggered (if M is +ve level , slave → -ve edge
M -ve , slave → +ve edge.

→ At the o/p of master slave there is no race around condition.

Conversion from one FF to another FF :-

→ Convert JK FF to D FF

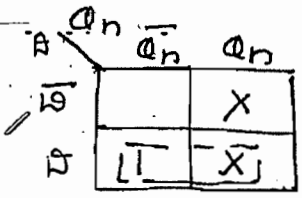
Procedure :-

(I) Construct required FF characteristic table
(II) Fill available flip flop excitation table
(III) Write excitation equation
(IV) Minimize logical exp.
(V) Implement circuit

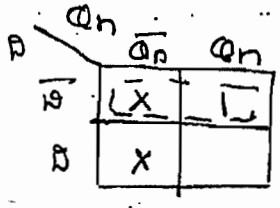
$\bar{R}$	Q_n	Q_{n+1}	J	K
0	0	0	0	X
0	1	0	X	1
1	0	1	1	X
1	1	1	X	0

$J = \Sigma m(2) + \Sigma d(1,3)$

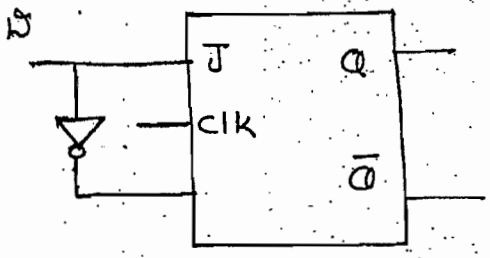
$K = \Sigma m(1) + \Sigma d(0,2)$



$J = D$



$K = \bar{D}$



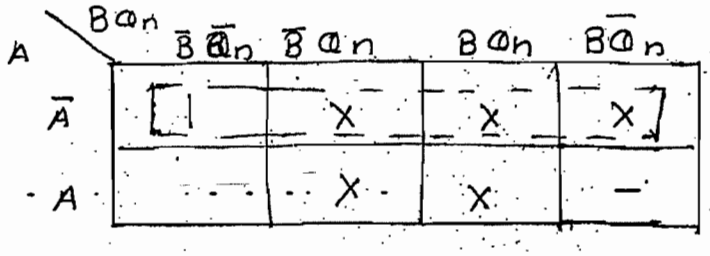
WB (Q→35)

→ JK → AB (Conv.)

A	B	Q_{n+1}
0	0	$\bar{Q}_n$
0	1	1
1	0	Q_n
1	1	0

A	B	Q_n	Q_{n+1}	J	K
0	0	0	1	1	X
0	0	1	0	X	1
0	1	0	1	1	X
0	1	1	1	X	0
1	0	0	0	0	X
1	0	1	1	X	0
1	1	0	0	0	X
1	1	1	0	X	1

$J = \Sigma m(0,2) + \Sigma d(1,3,5,7)$



$J = \bar{A}$

→ **JK to D :-**

$J = D$

$K = \bar{D}$

→ **JK to SR :-**

$J \Leftarrow S$

$K \Leftarrow R$

→ **SR to D :-**

$S = D, \quad R = \bar{D}$

→ **D to SR :-**

$Q_{n+1} = D$

$Q_{n+1} = S + \bar{R} Q_n$

$\boxed{D = S + \bar{R}Q}$

→ **JK to T :-**

$J = K = T$

→ **SR to JK :-**

$S = J\bar{Q}$

$R = KQ$

→ **SR to T :-**

$S = T\bar{Q}, \quad R = TQ$

→ **D to JK :-**

$D = J\bar{Q} + \bar{K}Q$

→ **D to T :-**

$D = T \oplus Q$

→ T to JK :-

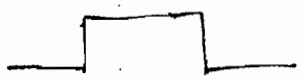
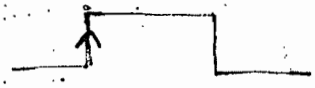
$T = J\bar{Q} + \bar{K}Q$

→ T to SR :-

$T = S\bar{Q} + RQ$

→ T to D :-

$T = D \oplus Q$

Latch	FF
→ level trigger 	edge trigger 
→ Asynchronous	Synchronous

Set up time :- (t_{SU})

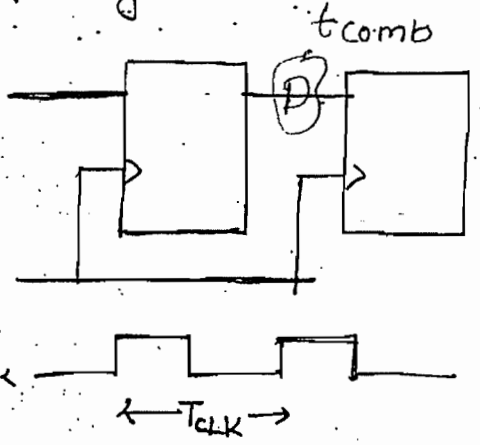
→ Min time required to keep i/p at proper level before applying clock

Hold time :- (t_H)

→ Min time to keep same data after applying clock to store data properly.

$t_{SU} > t_H$

→ $T_{CLK} \geq t_{pd}\,FF + t_{SU}$



→ $T_{CLK} \geq t_{pd}\,FF + t_{SU} + t_{comb}$

Registers:-

Registers:-

→ Registers are used to store group of bits

→ Registers are cascading of FF

→ For n-bits storage, n registers are required

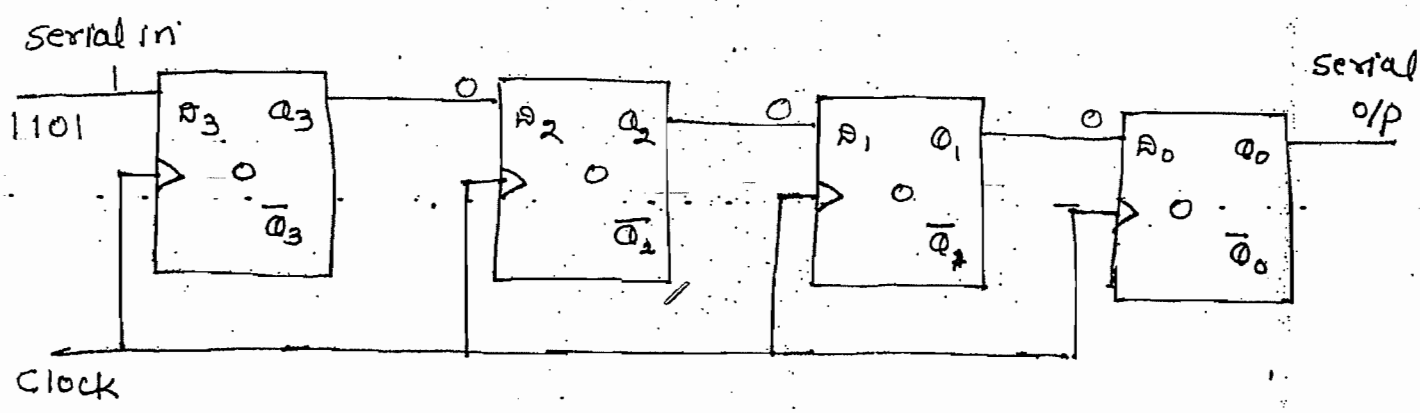
Depending on i/p and o/p, registers are of four types

(I) SISO (II) SIPO (III) PISO (IV) PIPO

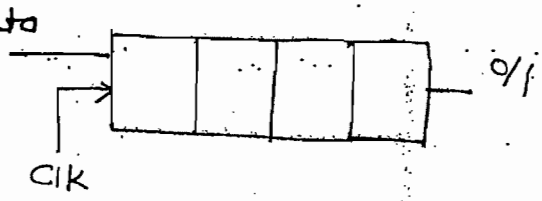
→ In registers, D FF is used

→ In shift registers - JK FF
- D FF

SISO :-



Clock	Data	$Q_3 Q_2 Q_1 Q_0$
0	1101	0 0 0 0
1		1 0 0 0
2		0 1 0 0
3		1 0 1 0
4		1 1 0 1



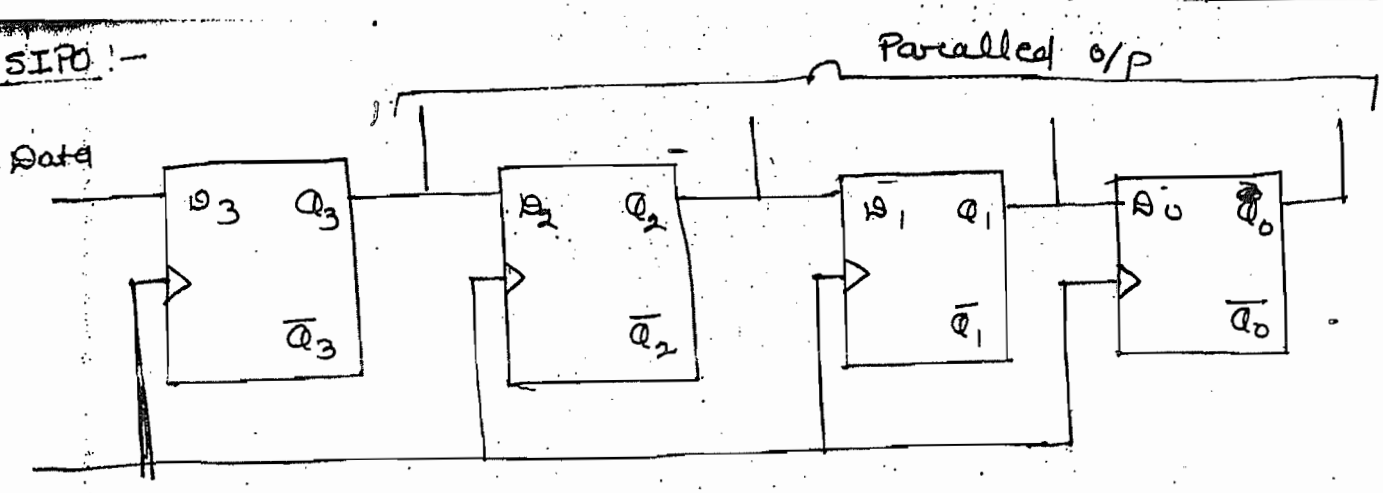
→ In n-bit SISO register to store n-bit data serial, minimum n-clock pulse are used.

→ SISO register is used to provide delay to the i/p data

→ In n-bit SISO register, it provides $n \times T_{CLK}$ delay to the i/p data.

→ In n-bit SISO register to provide serial out min (n-1) clock pulses are used.

SIPO :-



→ In SIPO register, to provide serial i/p, min. n clock pulses are required

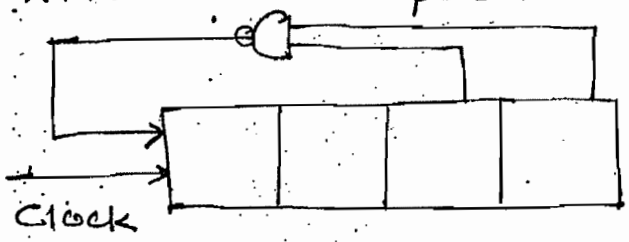
→ To provide parallel out data, no clock pulse is required

Serial → temporial code

Parallel → spacial code

→ Convert temporial code into spacial code

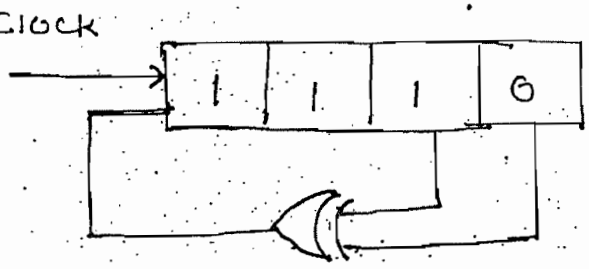
Ques :- The circuit shown in figure is SIPO register loaded with 1010. After 3-clock pulse data present in



Ans :-

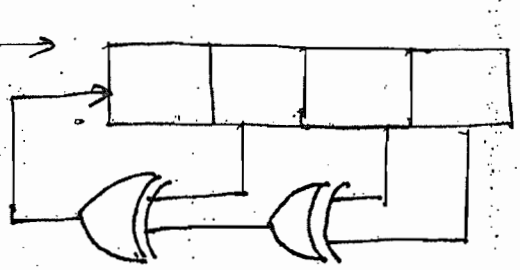
Clock				
0	1	0	1	0
1	1	1	0	1
2	1	1	1	0
3	1	1	1	1

Ques :- After 3-clock pulse data present in register is



Ans :- 0 0 1 1

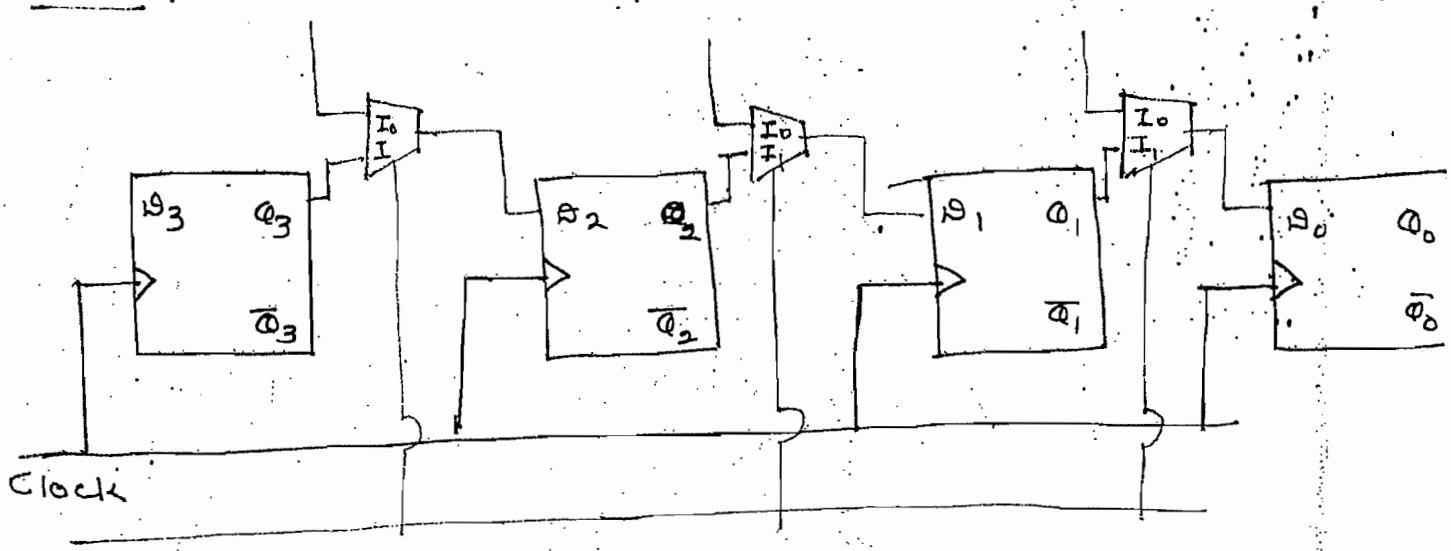
Ques:- 4-bit SIPO loaded with 1011 : If clock pulse are applied continously then after how many clock pulses - data will become 1011 again



Soln:-

Clock	Data
0	1011
1	0101
2	0010
3	1001
4	1100
5	1110
6	0111
7	1011

PISO :-



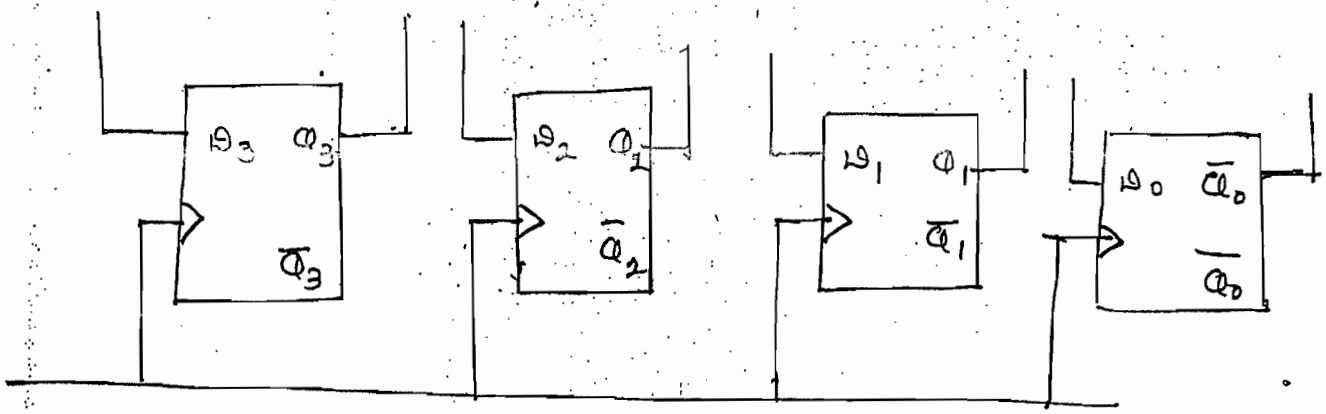
→ Control = 0 , parallel in

→ control 1 → serial o/p

→ PI (0° clock) → Serial o/p = n-1

→ convert spacial code into temporial code.

PIPO :-



Note:

	i/p	o/p
SISO	n	n-1
SIPO	n	0
PISO	1	n-1
PIPO	1	0

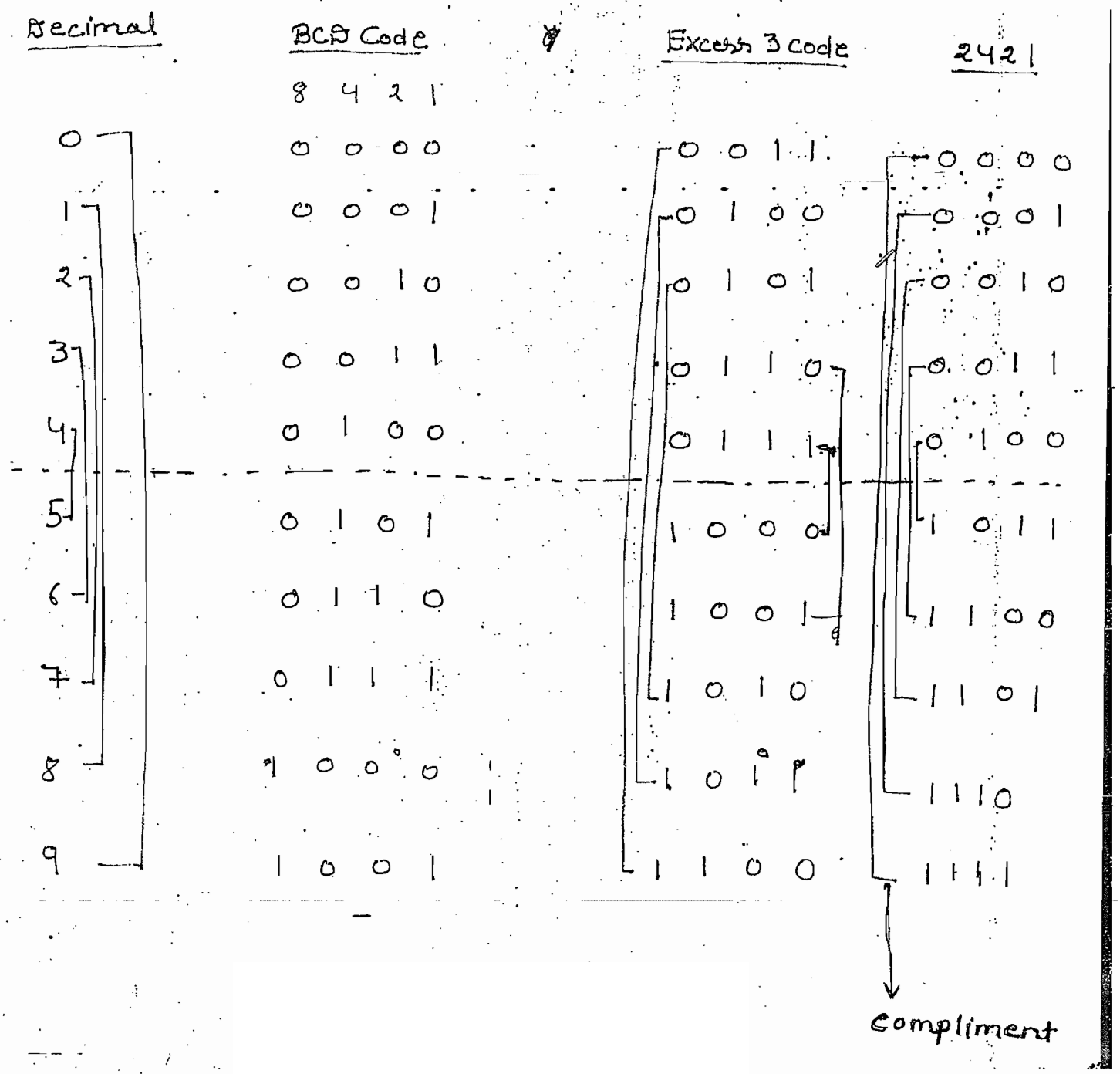
→ PIPO is fastest

BCD Code :—

→ 4 bit code
→ Each decimal digit is represented with 4 bit binary form
→ Weighted Code
→ 8421 code
→ In 4-bit BCD Code there are 6 invalid BCD code
i.e. 1010, 1011, 1100, 1101, 1110, 1111

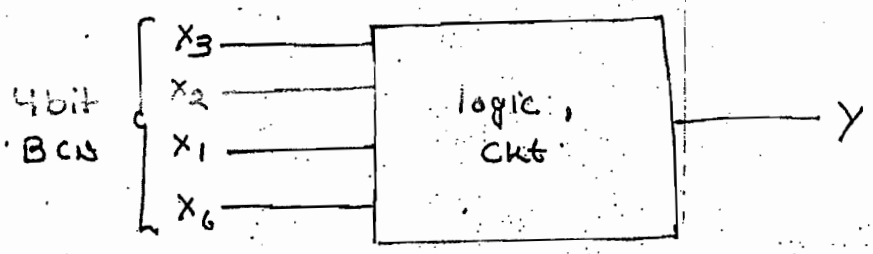
Note:—

→ In logic circuit design, consider invalid BCD code as dont care
→ During addition if invalid BCD code is present then add 0110 to get correct result



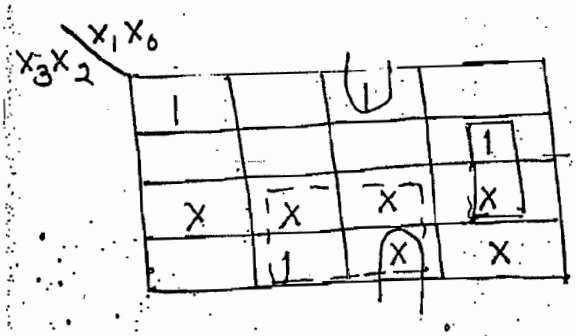
Ques:- The circuit shown in figure is a 4-bit logic circuit which has i/p as BCD code and o/p Y.

Y is logic "1" when i/p BCD is divisible by 3 then minimised expression for o/p Y



Soln:-

Decimal	BCD Code	Y
0	0 0 0 0	1
1	0 0 0 1	0
2	0 0 1 0	0
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	0
6	0 1 1 0	1
7	0 1 1 1	0
8	1 0 0 0	0
9	1 0 0 1	1



$= X_3X_0 + \bar{X}_2X_1X_0 + X_2X_1\bar{X}_0 + \bar{X}_3\bar{X}_2\bar{X}_1\bar{X}_0$

Note:-

For 8-bit BCD code

Valid 00 to 99 = 100

invalid 256 - 100 = 156.

Excess 3 code :-

→ BCD code + 0011
→ Excess 3 BCD code
→ Unweighted code
→ self complimentary code

Note :-

9 = 2+4+2+1 ⟶ 2 4 2 1
9 = 3+3+2+1 ⟶ 3 3 2 1 Self compliment & Weighted.
9 = 5+2+1+1 ⟶ 5 2 1 1 (If sum of digit comes 9)

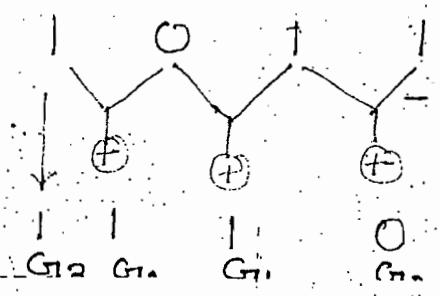
→ ASCII ⟶ 7 bit code
→ EBCDIC ⟶ 8 bit (data transmission) (Punching code)
→ Hollerath → 12 bit

Gray code :-

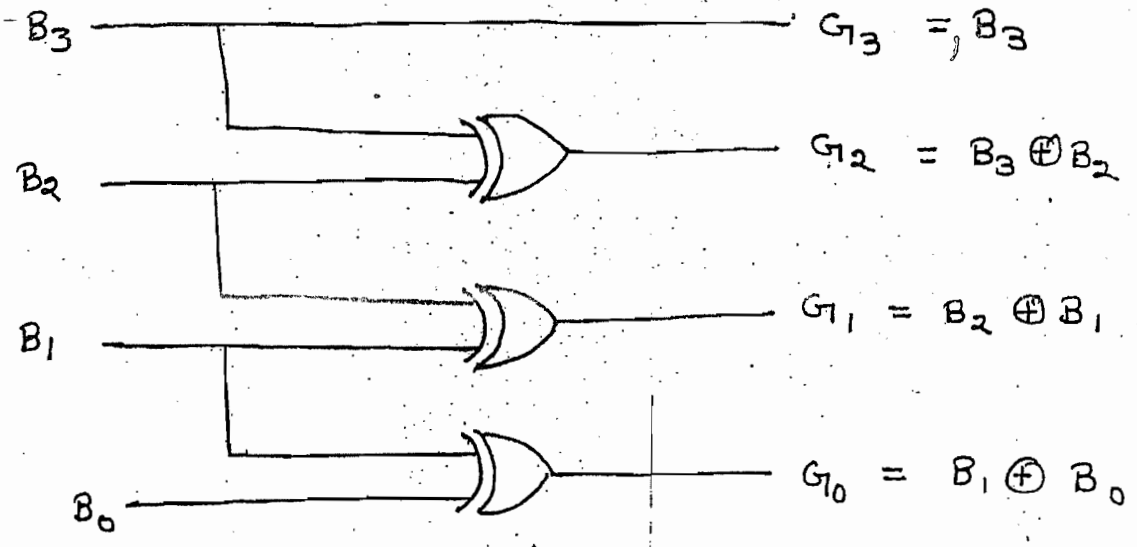
→ In this, successive no. will differ by only 1 bit
→ Also known as Unit distance code (UDC)
→ Reflective code or cyclic code

0	0 0
1	0 1
2	1 1
3	1 0

→ Minimum error code
→ Unweighted code



⟶ Binary
(Binary to Gray)



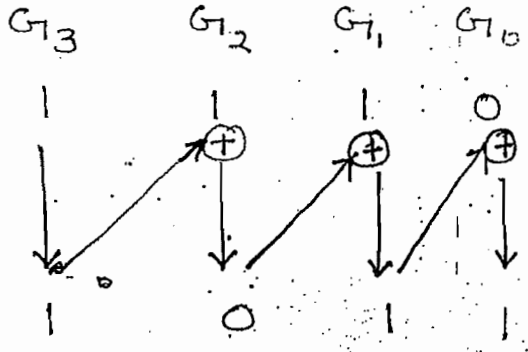
Gray to Binary :-

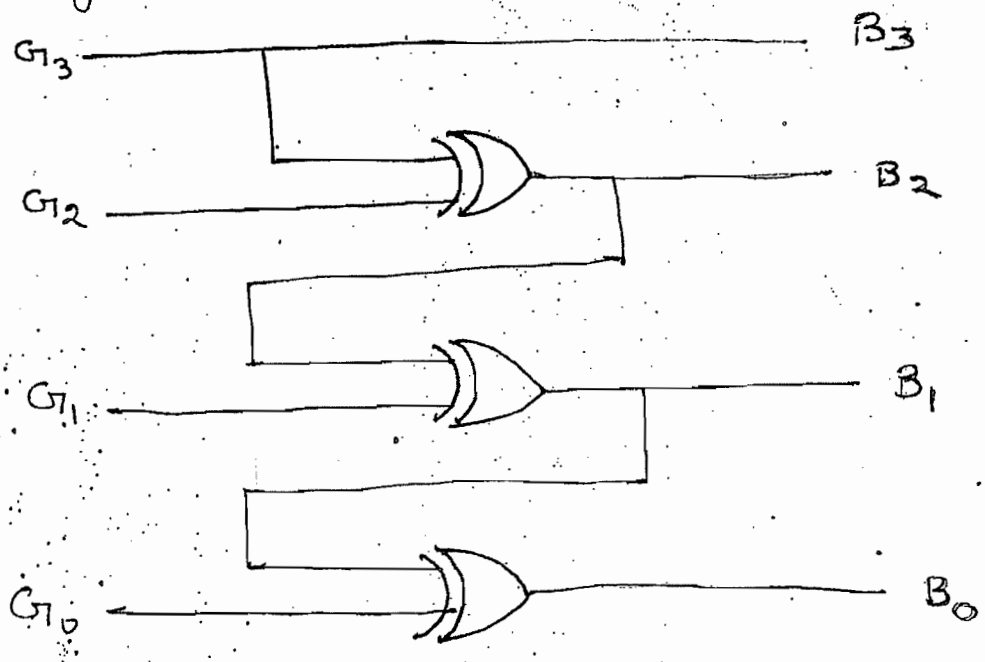
B_3 ——— B_3

$B_3 \oplus B_2$ ——— $B_2 = B_3$

$B_2 \oplus B_1$ ——— $B_1 = B_2 \oplus G_1$

$G_0 = B_1 \oplus B_0$ ——— $B_0 = B_1 \oplus G_0$





COUNTERS

Counters:-

→ Counters can be used

(I) To count no. of clock pulses

(II) Frequency dividers

(III) Timers

(IV) To generate waveforms

(V) To provide frequency measurements

(VI) Pulse Width measurements

→ In counters with n FF's max. possible states are 2^n

n → no. of FF

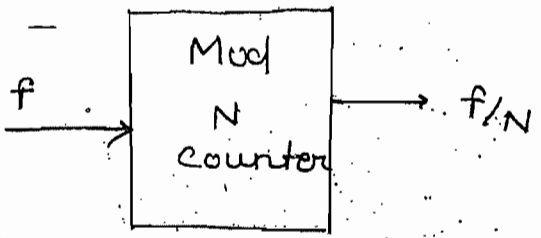
N → No. of states

$$N \leq 2^n$$

$$n \geq \log_2 N$$

→ No. of states used in counter is called as modulo count Mod no.

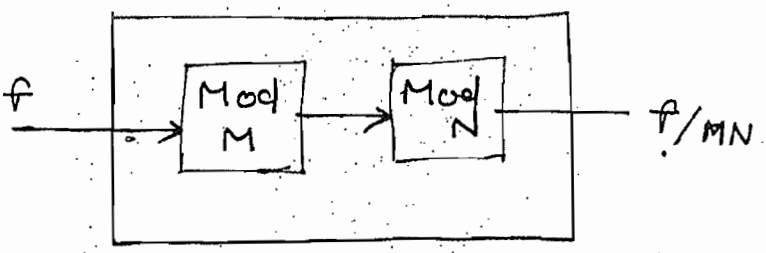
Mod 10 → Decade counter



Ques:- A decade counter is applied with clock frequency of 10MHz then o/p of frequency is.

Ans:- 1MHz

→ If mod M and mod N counters are cascaded then it will acts as Mod MN counters.



Ques:- Mod 3, Mod 4 and Mod 5 counters are cascaded then no. of states are

Soln:- No. of states = 3x4x5 = 60 states

→ Depending clock pulse applied counter are of two types

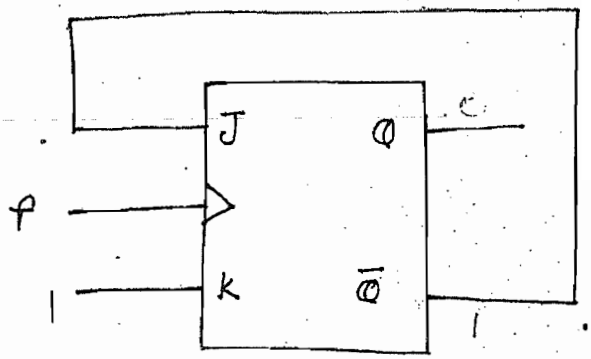
(I) Synchronous counter

(II) Asynchronous counters

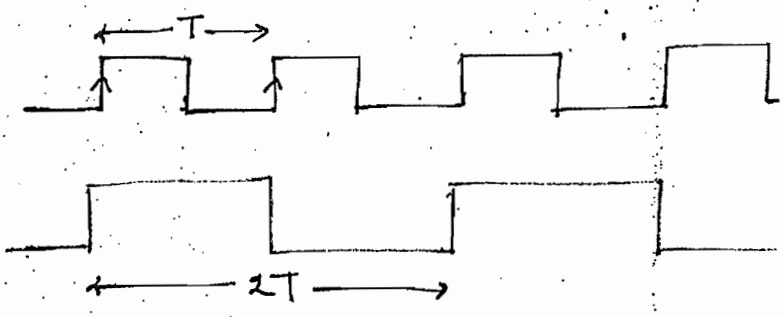
Clock

	Synchronous Counter	Asynchronous counter
1.	All FF's are applied with same clock	Different FF's are applied with applied different clock.
2.	Faster	Slower
3.	Any count sequence can be design	Fixed count sequence is possible i.e. Up & Down
4.	No decoding errors	Decoding errors will be present
5.	eg:- Ring counter, Johnson counter	eg:- Ripple counter

Ques:-



Clock	Q
0	0
1	1
2	0
3	1
4	0
5	1



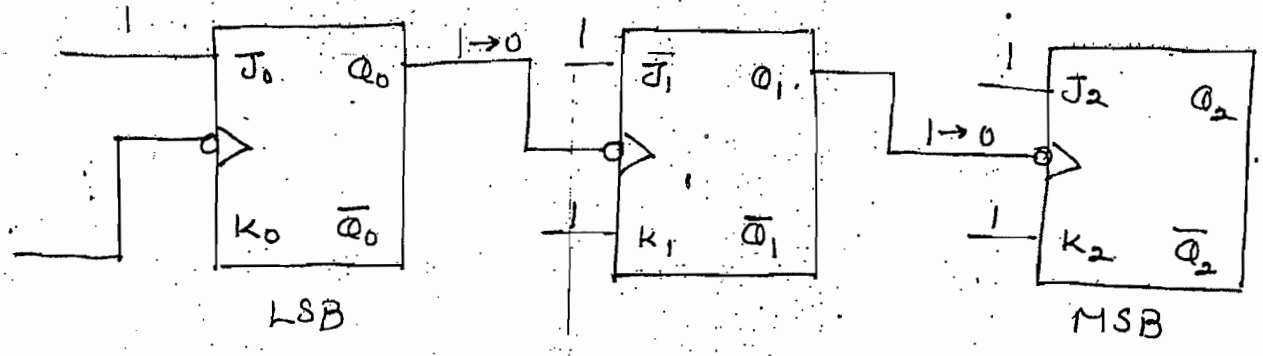
Ripple Counter :-

→ Asynchronous counter

→ Different flip flops are applied with different clock pulse

→ Toggle Mode

3-Bit Ripple Counter :-



→ In ripple counter shown in figure, Q_0 toggles for every clock pulse -ve edge

→ Q_1 toggles, Q_0 changes from 1 → 0

→ Q_2 toggles, Q_1 changes from 1 → 0

→ Q_0 toggles for every clock pulse -ve edge

Truth Table :-

Clock	Q_2	Q_1	Q_0
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
8	0	0	0

→ Mod 8 ripple up counter (above).

Note :-

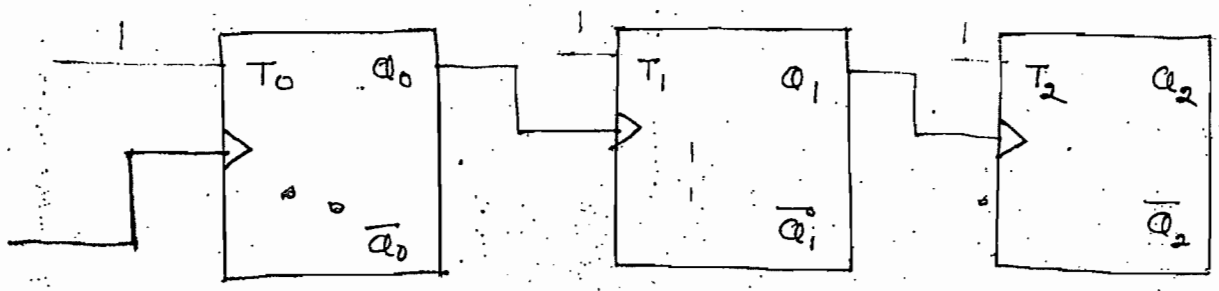
(I) -ve edge trigger - Q as clock → Upcounter

(II) +ve edge trigger - $\overline{Q}$ as clock → Upcounter

(III) +ve edge trigger - Q as clock → downcounter

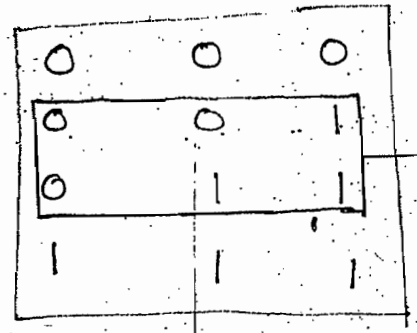
(IV) -ve edge trigger - $\overline{Q}$ as clock → downcounter

3 Bit Ripple Down counter (Mod 8 ripple down counter) :-



Truth Table :-

Clock	Q_2	Q_1	Q_0
0	0	0	0
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0



→ unwanted state
transition state
(due to propagation dela
↓
decoding error

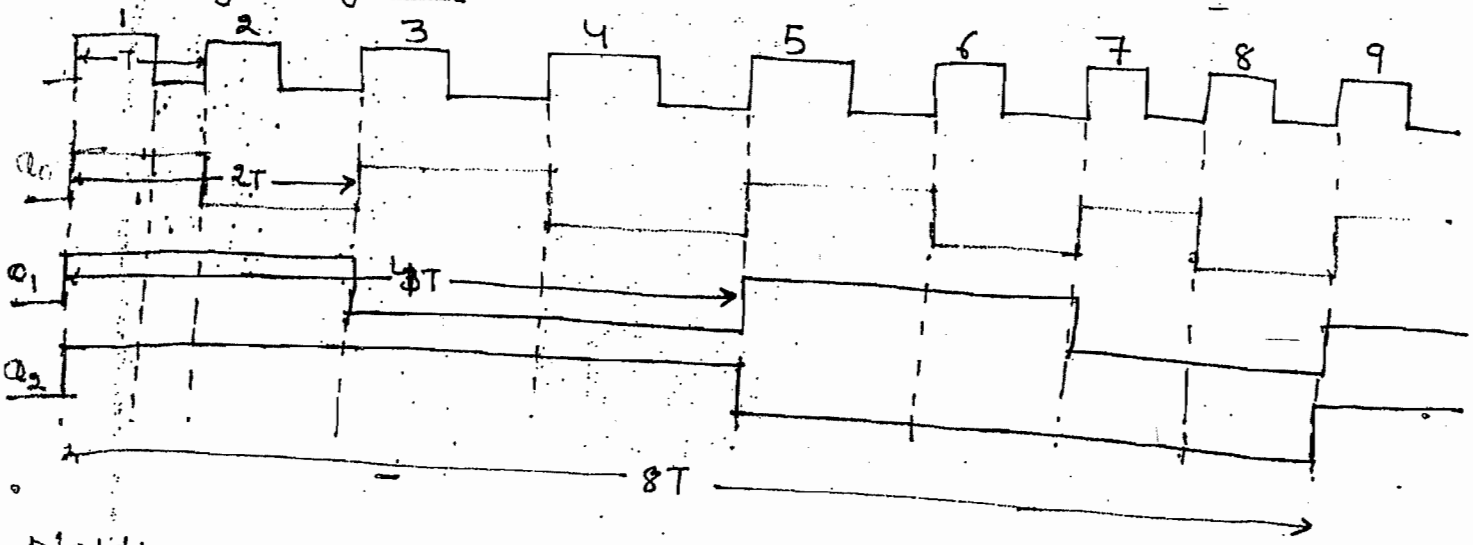
In n-bit ripple counter propagation delay of each ff is $t_{pd\,FF}$ then $T_{CLK} \geq n\, t_{pd\,FF}$

$$f_{CLK} \leq \frac{1}{n t_{pd\,FF}}$$

$$f_{max} = \frac{1}{n t_{pd\,FF}}$$

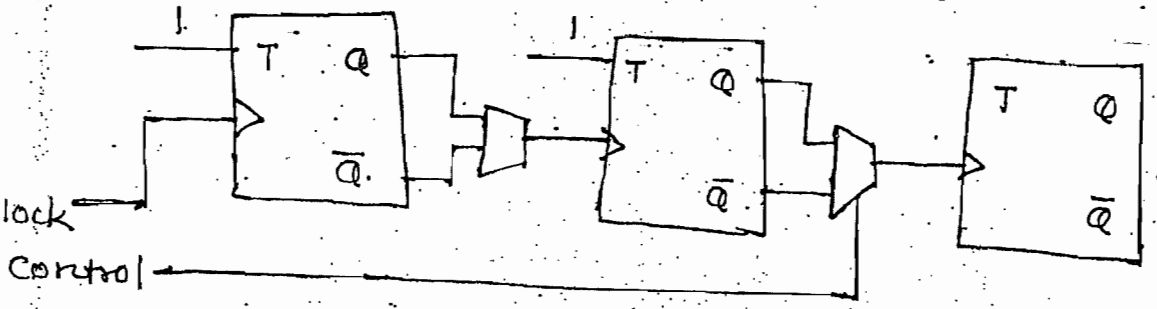
→ Decoding error is present in ripple counter. It can be avoided by using strobe/Control

Timing Diagram :-

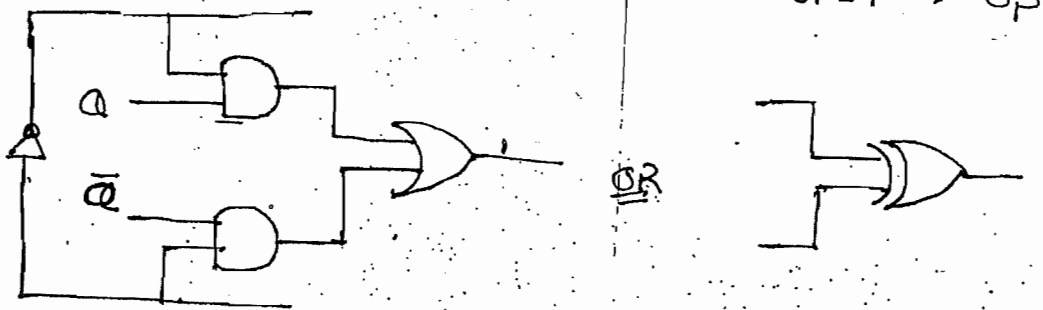


Note:-

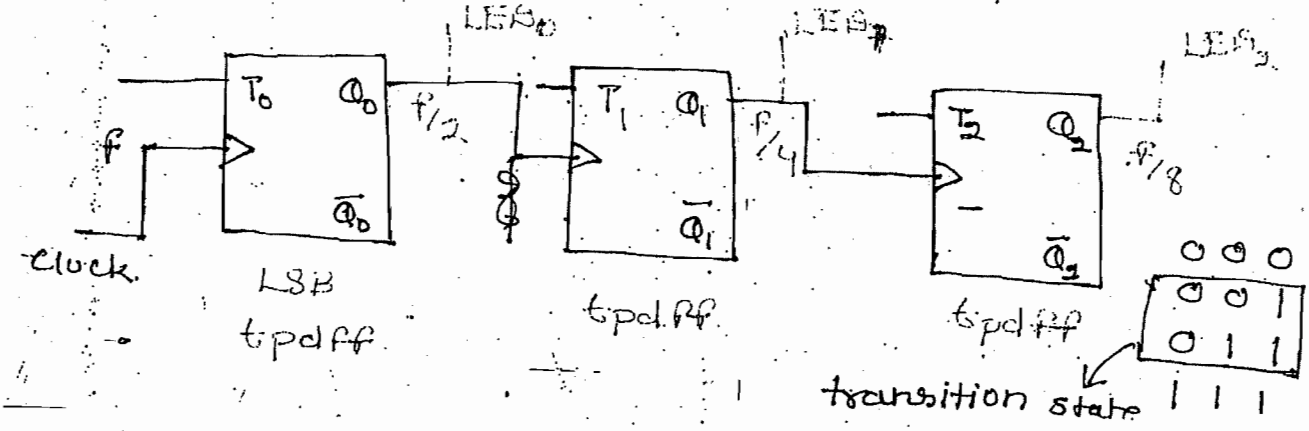
3 Bit Ripple up-down counter using +ve edge triggering

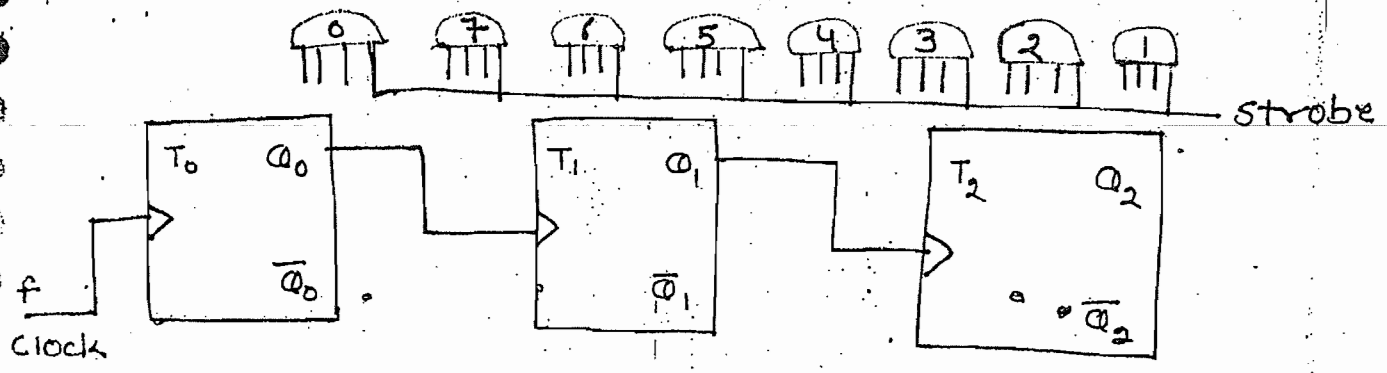


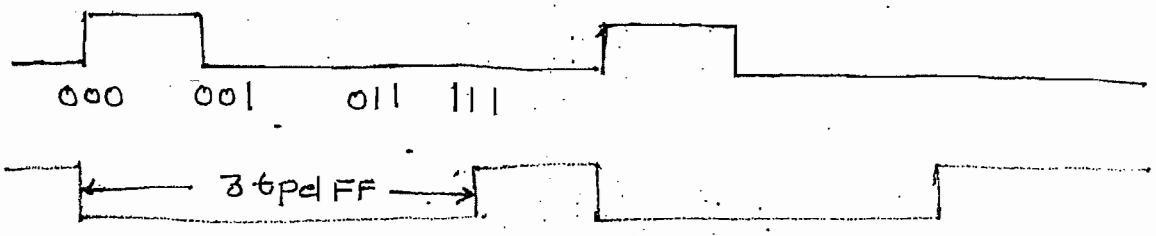
Control' = 0 → Down counter Control = 1 → Upcounter



Decoding in 3-Bit Ripple down counter:-





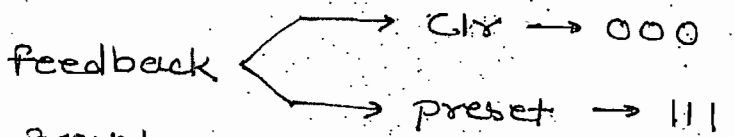


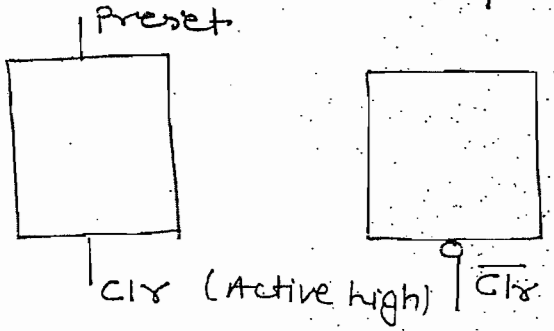
$$T_{CLK} \geq n\, t_{pd\,FF} + T_S$$

$$f_{CLK} \leq \frac{1}{n\, t_{pd\,FF} + T_S}$$

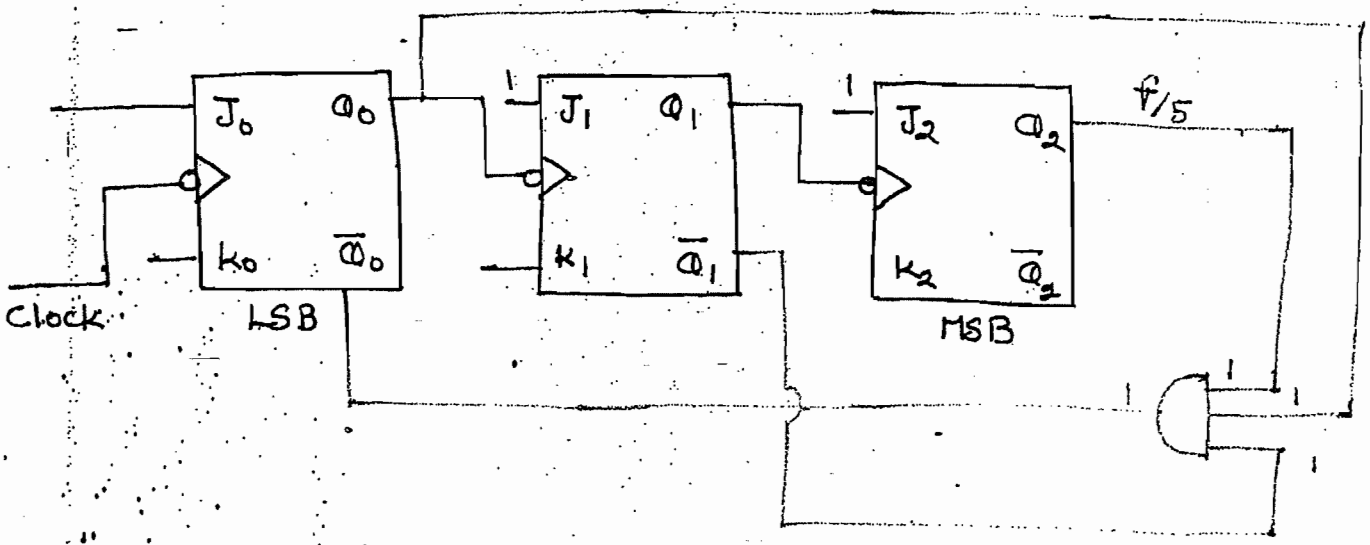
$$f_{max} = \frac{1}{n\, t_{pd\,FF} + T_S}$$

→ In counter to reduce no of states or to eliminate some of the states feedback is applied in the counter through clear or Reset the i/p





Mod 5 Ripple Counter :-

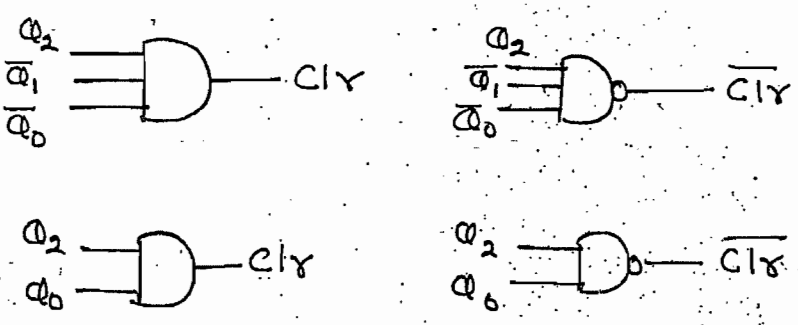


Clock	Q_2	Q_1	Q_0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	0	0	0

101

→ If +ve edge trigger then it is MOD 3 ~~by~~ Ripple down counter

→ In Ripple counter Max. possible states = 2^n



→ ~~Imp~~ Ripple Counter :-

(i) Trigger → +ve edge, -ve edge

(ii) Clock → Q, Q̄

(iii) Clock → Q, Q̄

(iii) Counter → up, down

(iv) feedback → Clr (000), Pr (111)

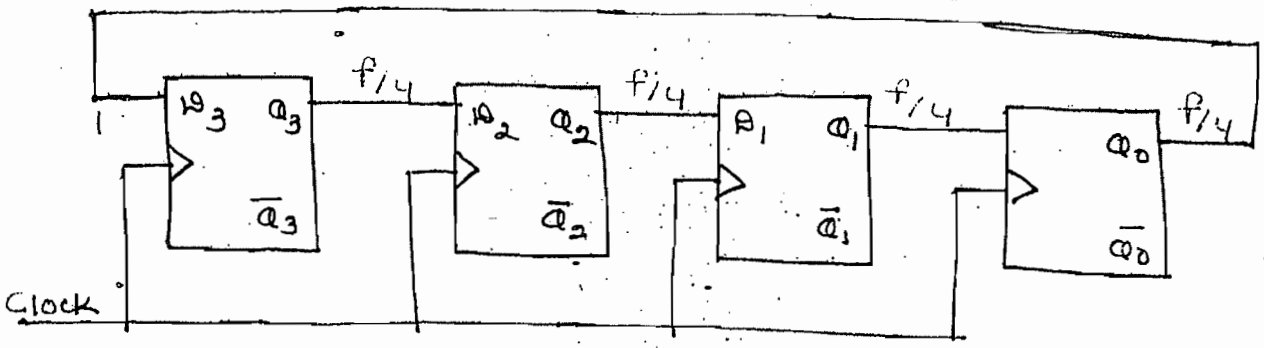
(v) Terminating logic/count → Clr – 000, Preset – 111

Synchronous counter :-

Ring counter :-

→ Synchronous counter

→ Shift register with feedback



→ In ring counter only one FF o/p. is logic '1' and it will rotates among all FF's with clock

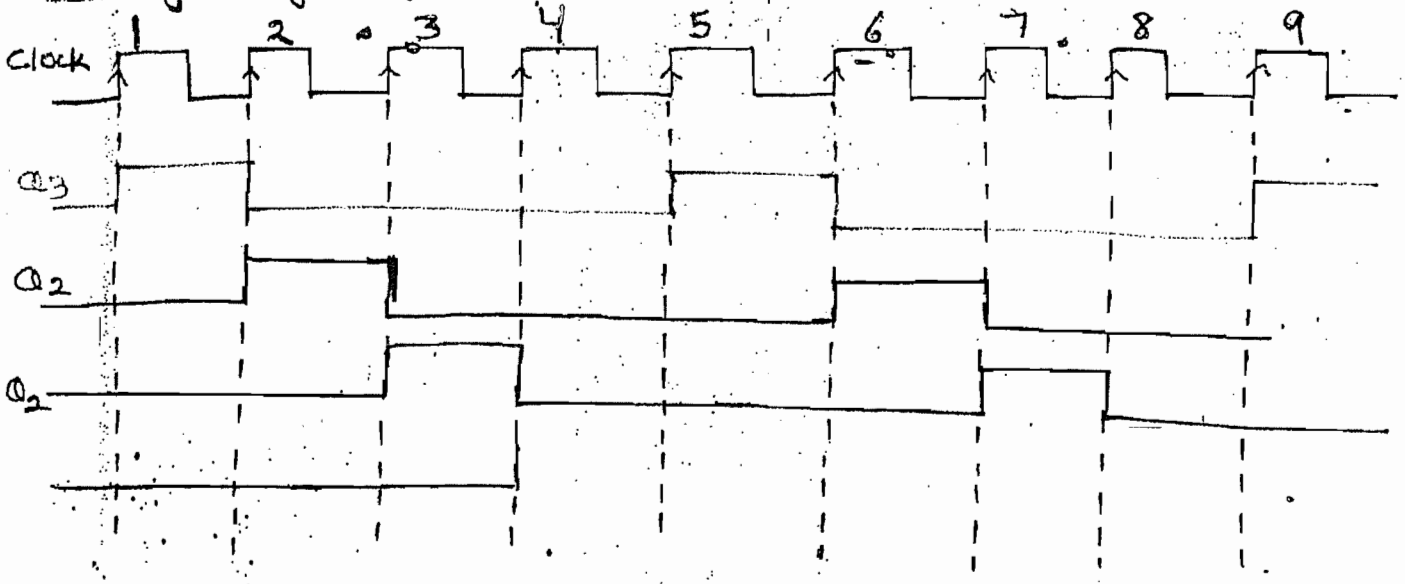
Clock	Q_3	Q_2	Q_1	Q_0
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0
6	0	1	0	0
7	0	0	1	0
8	0	0	0	1
9	1	0	0	0

→ In ring counter

No. of states = n

No. of unused states = $2^n - n$

Timing Diagram :-



→ In n-bit ring counter output frequency at any ff is $\frac{f}{n}$

→ In ring counter time period of clock

$$T_{CLK} \geq t_{pd\,FF} + t_{SU}$$

→ $$f_{CLK} \leq \frac{1}{t_{pd\,FF} + t_{SU}}$$

→ Ring counter is fastest among all

→ Ring counter will generate phase shift of $\frac{360}{n}$

Application :-

→ Stepper Motor Control

→ SAR

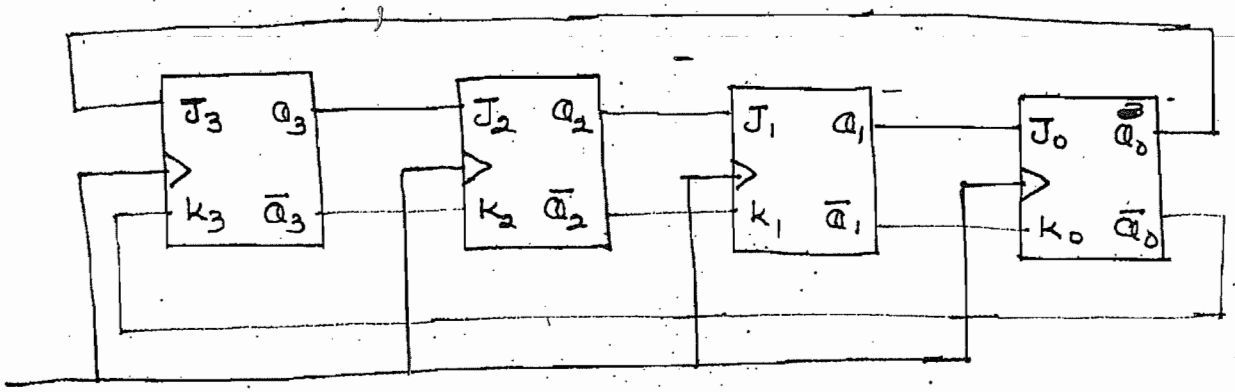
→ Waveform generators

Decoding :-

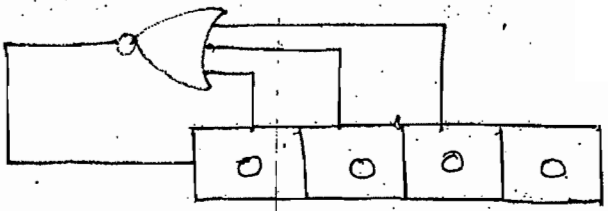
→ Main advantage of Ring counter is decoding logic is very simple and no hardware is used

Q_3	Q_2	Q_1	Q_0	Decoding
1	0	0	0	Q_3
0	1	0	0	Q_2

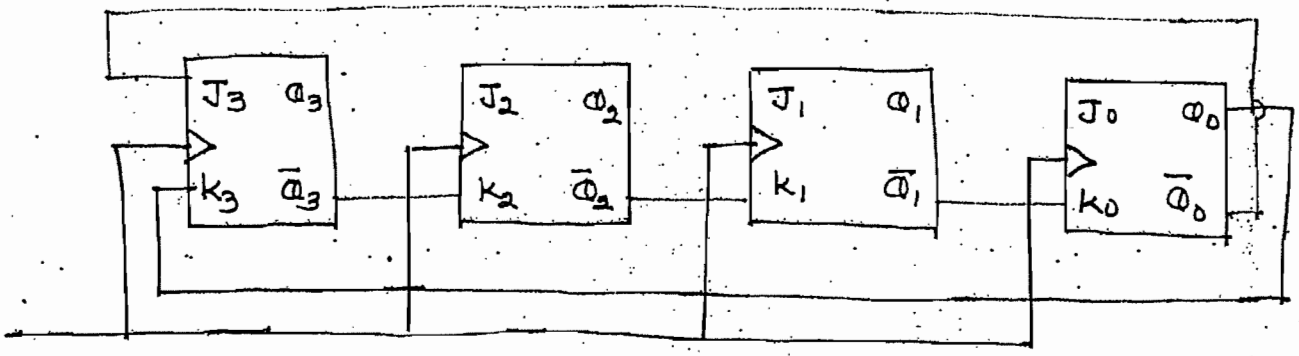
Ring Counter Using JK FF :-

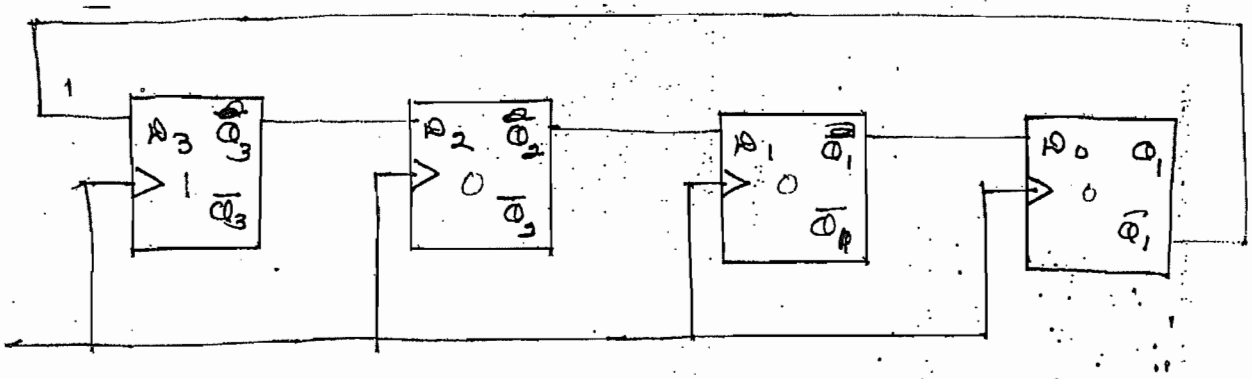


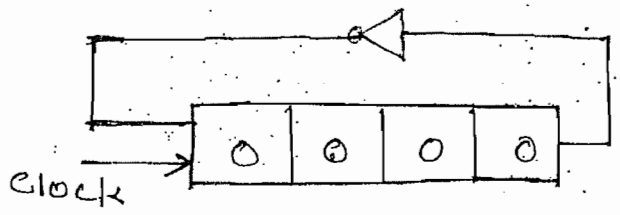
Self starting Ring Counter :-



Johnson Counter :-







Clock	Q_3	Q_2	Q_1	Q_0
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0
9	1	0	0	0
10	1	1	0	0

→ No of states used — $2n$

→ No of unused states — $(2^n - 2n)$

→ If Johnson counter is operated in used state then o/p frequency at any flip flop is $\frac{f}{2n}$

→ **Disadvantage:-**

→ Lockout may occur when counter is operated in unused state

$$T_{CLK} \geq t_{pdFF} + t_{SU}$$

Decoding:-

Clock	Q_3	Q_2	Q_1	Q_0	Decoding
0	0	0	0	0	$\overline{Q_3}\,\overline{Q_0}$
1	1	0	0	0	$\overline{Q_3}\,\overline{Q_2}$

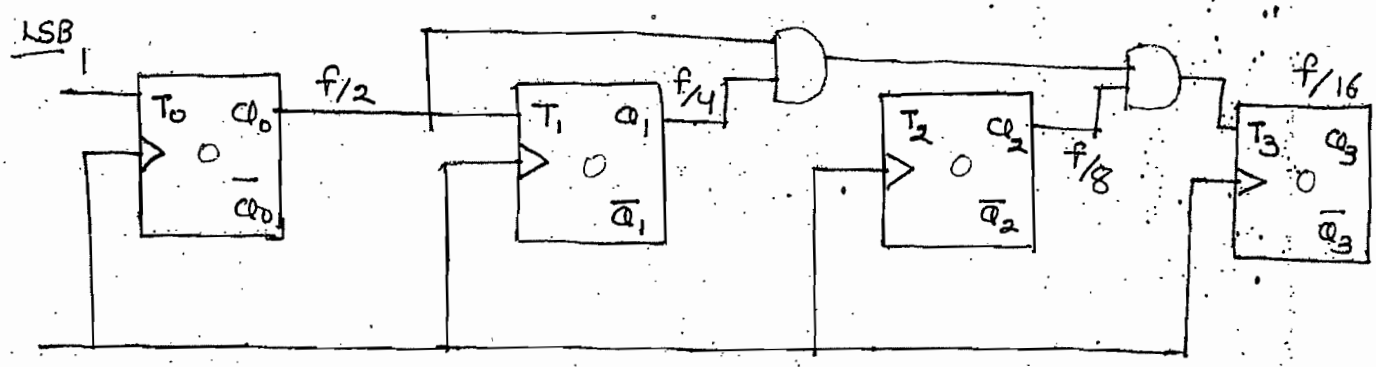
Clock	Q_3	Q_2	Q_1	Q_0	Decoding
0	0	0	0	0	$\bar{Q}_3 \bar{Q}_0$
1	1	0	0	0	$Q_3 \bar{Q}_2$
2	1	1	0	0	$Q_2 \bar{Q}_1$
3	1	1	1	0	$Q_1 \bar{Q}_0$
4	1	1	1	1	$Q_3 \bar{Q}_0$
5	0	1	1	1	$\bar{Q}_3 Q_2$
6	0	0	1	1	$\bar{Q}_2 Q_1$
7	0	0	0	1	$\bar{Q}_1 Q_0$

→ To decode Johnson counter one 2 i/p AND gate or NOR gate is used for each state

→ Johnson counter is also known as

(I) Twisted Ring counter
(II) switch tail counter
(III) Walking counter
(IV) Creeping counter
(V) Mobien Counter

Synchronous Series Carry counter :-



→ The circuit shown in figure is synchronous series carry upcounter.

→ If $\bar{Q}$ is connected then it is down counter

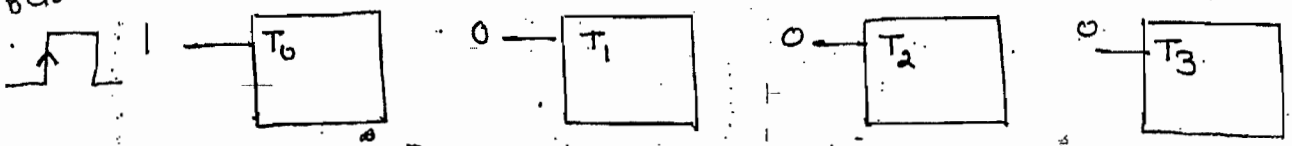
→ Q_0 toggles for every clock pulse +ve edge of clock

→ Q_1 toggles when $Q_0 = 1$ and +ve edge of clock

→ Q_2 toggles when $Q_1 \& Q_0 = 1$ and +ve edge of clo

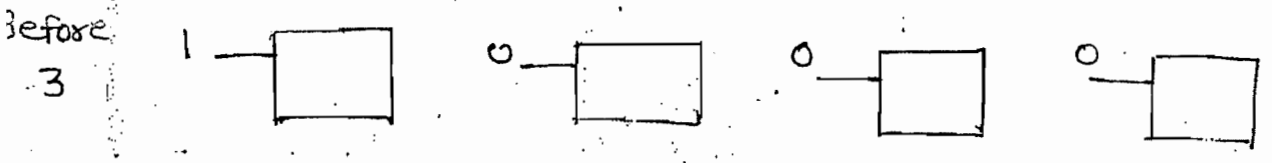
→ Q_3 toggles when $Q_2 Q_1 Q_1 = 1$ and −ve edge of clo

Before

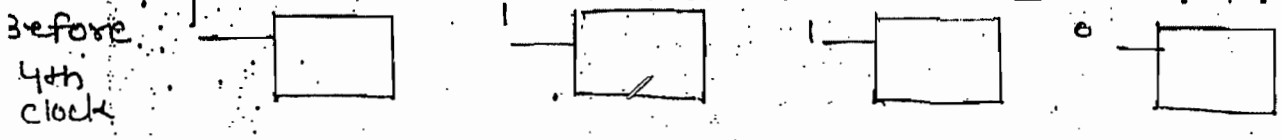


efore
2.

Before
3



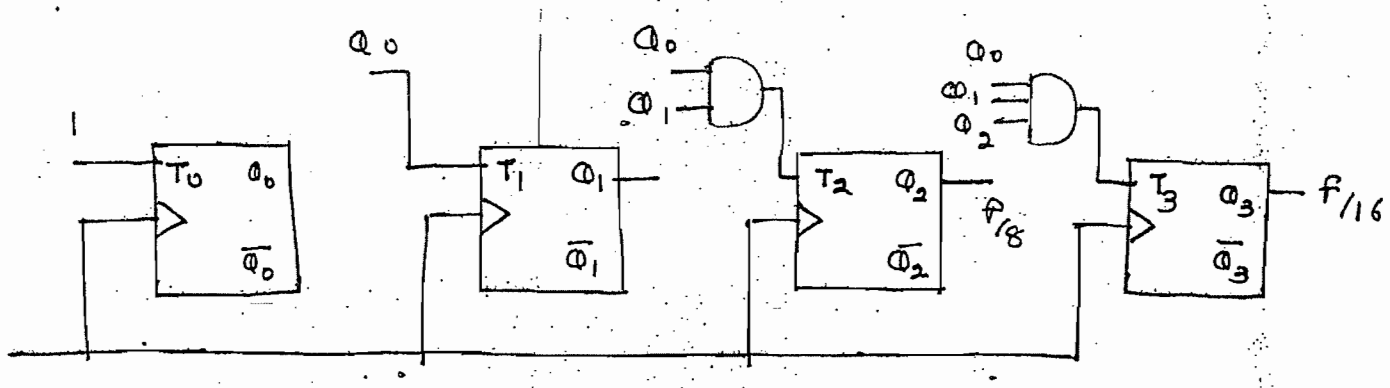
Before
4th
clock



Clock	Q_3	Q_2	Q_1	Q_0
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
⋮	⋮	⋮	⋮	⋮
15	1	1	1	1
16	0	0	0	0

Time period, $T_{CLK} \geq t_{pd FF} + (n-2)\, t_{pd AND} + t_{SU}$

Synchronous Parallel carry counter :-



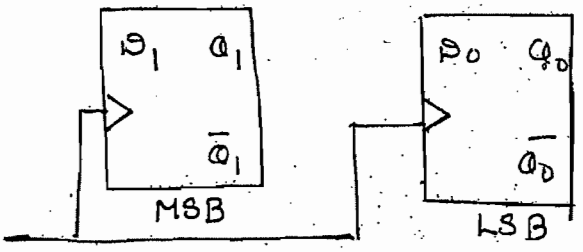
$$T_{CLK} > t_{pd FF} + t_{pd AND} + t_{SU}$$

Synchronous Counter Design :-

Design a synchronous counter with count sequence

$$00 \rightarrow 10 \rightarrow 01 \rightarrow 11 \rightarrow 00$$

→ Identify no. of states and no. of FF
→ Construct state Table
→ Write Excitation equation
→ Minimize
→ Implement



State Table :-

Present State		Next State		Excitation	
Q_1	Q_0	Q_{1+}	Q_{0+}	D_1	Q_0
0	0	1	0	1	0
1	0	0	1	0	1
0	1	1	1	1	1
1	1	0	0	0	0

$$D_1 = \bar{Q}_1\bar{Q}_0 + \bar{Q}_1 Q_0 = \bar{Q}_1$$

$$D_0 = \bar{Q}_1 Q_0 + Q_1\bar{Q}_0 = Q_1 \oplus Q_0$$

Clock	Q_1	Q_0
0	0	0
1	1	0

Note:-

→ In counter design, consider unused states as dont care.

→ To design lock out avoiding counter, consider unused state corresponding next state, one of the used state

State Machines

- Mealy
  - present o/p
    - present state
    - Present i/p's
- Moore
  - present o/p depends on only present state

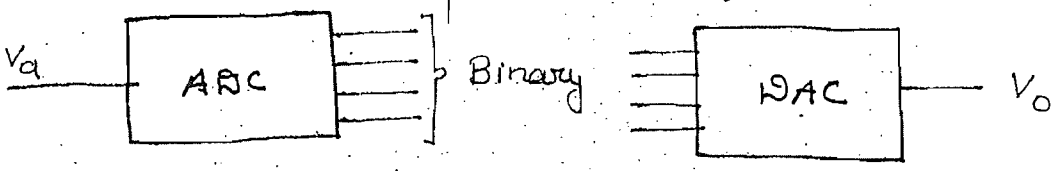
→ According to Moore's law, transistor density will be double in every 18 months.

Ques:- In a 4-bit ripple counter propagation delay of each flip flop is 10nsec then maximum clock frequency that can be applied to the counter is?

Soln:- $f_{max} = \frac{1}{n t_{pdd}} = \frac{1}{4 \times 10 \times 10^{-9}} = \frac{10^8}{4} = 25\ MHz$

, Ans

ADC / DAC



(I) Counter type
(II) SAR type
(III) Flash type
(IV) Dual slope

(I) Weighted Resistor
(II) R-2R ladder

DAC :-

(I) Resolution (II). Analog o/p Voltage (III) V_{FS}
(IV) % Resolution (V) Error / Accuracy

(I) Resolution :-

→ change in analog voltage corresponding 1 LSB bit increment at the i/p

$$\text{Resolution} = \frac{V_r}{2^n - 1}$$

where n = no of bits.

V_r = reference voltage corresponding to logic '1' i/p

(II) Analog o/p Voltage :-

$$V_a = \text{Resolution} \times \text{Decimal}$$

Ques:- In a 4-bit DAC ref voltage is 10V. If binary data 1001 is applied then analog o/p voltage is

Soln:-

$$V_a = \frac{10}{2^4 - 1} \times 9$$

$$= \frac{10^2}{15_3} \times 9^3 = 6V$$

(III) V_{FS} :- (Full scale o/p voltage)

$$V_{FS} = \text{Max. analog o/p voltage}$$
$$= R \times \text{max D}$$
$$= \frac{V_r}{2^n - 1} \times (2^n - 1)$$

$$\boxed{V_{FS} = V_r}$$

(IV) % Resolution :-

$$\% \text{ Resolution} = \frac{\text{Resolution}}{V_{FS}} \times 100$$
$$= \frac{\frac{V_r}{2^n - 1}}{V_r} \times 100$$

$$\boxed{\% \text{ Resolution} = \frac{1}{2^n - 1} \times 100}$$

(V) Error / Accuracy :-

→ In ADC / DAC max. error acceptable is 1LSB bit

$$\boxed{\begin{aligned} \text{error} &\leq 1\text{LSB} \\ \text{error} &\leq \text{Resolution} \end{aligned}}$$

Note :-

→ Resolution of R-2-R ladder N/W is

$$\boxed{R = \frac{V_r}{2^n}}$$

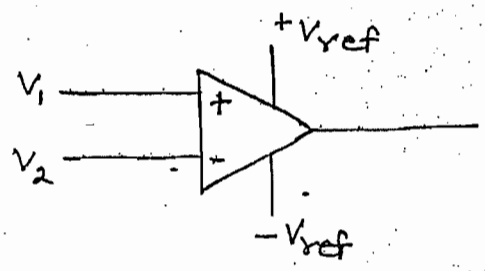
→

$$\boxed{\text{Resolution of ADC} = \frac{V_r}{2^n - 1}}$$

% Resolution of ADC $= \frac{1}{2^n - 1} \times 100$

error $\leq$ Resolution

ADC



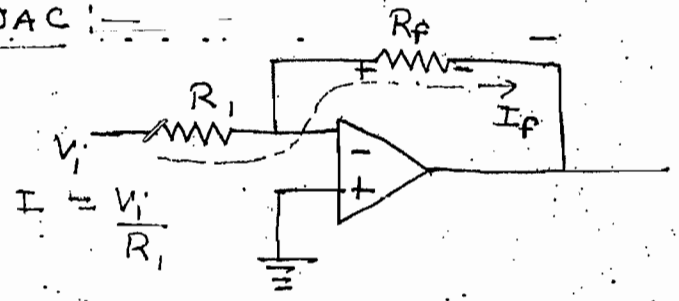
For ADC's

$V_1 > V_2 \longrightarrow +V_{ref} \longrightarrow$ logic 1

$V_1 < V_2 \longrightarrow -V_{ref} \longrightarrow$ logic 0

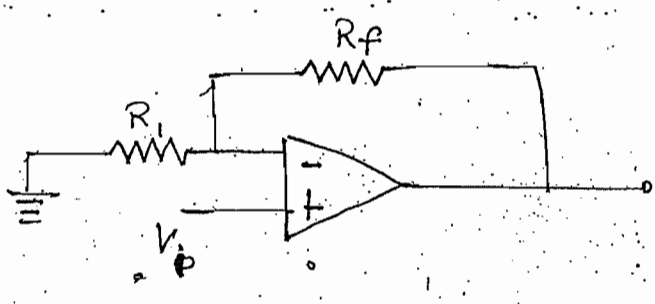
If $V_1 = V_2 \longrightarrow \pm V_{ref}$ (based on internal bias current)

For DAC :—



$V_0 = \left(-\frac{R_f}{R_1}\right) \cdot V_1$

$V_0 = (-R_f) \times I_f$



$V_0 = \left(1 + \frac{R_f}{R_1}\right) V_i$

4-bit Weighted Resistor DAC ckt :—

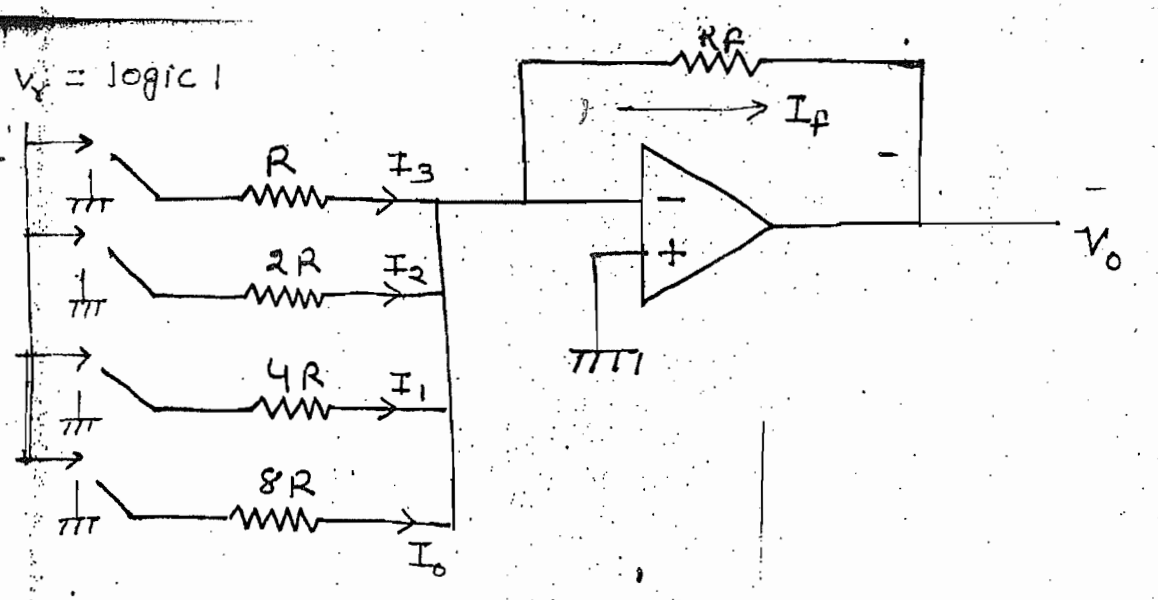
→ Accuracy is less due to different resistor

→ Assuming 4 bit to

b_3 b_2 b_1 b_0

↓ ↓

MSB LSB



$$I_3 = \frac{V_r}{R} \times b_3 \qquad I_2 = \frac{V_r}{2R} \times b_2$$

$$I_1 = \frac{V_r}{4R} \times b_1 \qquad I_0 = \frac{V_r}{8R} \times b_0$$

$$I_f = I_3 + I_2 + I_1 + I_0.$$

$$\Rightarrow V_0 = (-R_f) I_f$$

LSB resistance $= 2^{n-1} \times$ MSB Resistance

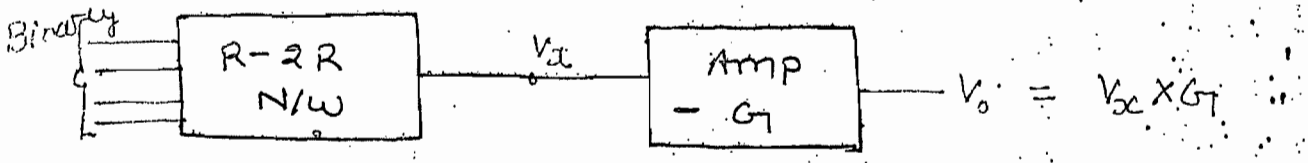
Disadvantage of Weighted Resistor :-

→ Accuracy is less due to diff use of different resistors

→ To overcome this R-2R ladder network is used.

R-2R ladder

- Normal ladder
  - Non inverting Amplifier
  - Inverting Amplifier
- Inverted



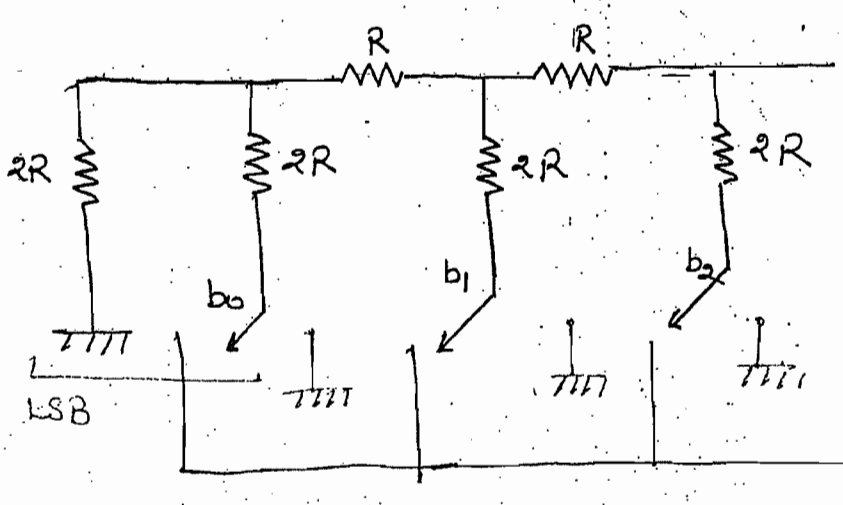
$$V_x = R \times B$$

$$= \frac{V_r}{2^n}\left(\sum_{i=0}^{n-1} 2^i b_i\right)$$

$$\boxed{V_o = \frac{V_r}{2^n} \times \left(\sum_{i=0}^{n-1} 2^i b_i\right) \times G_f}$$

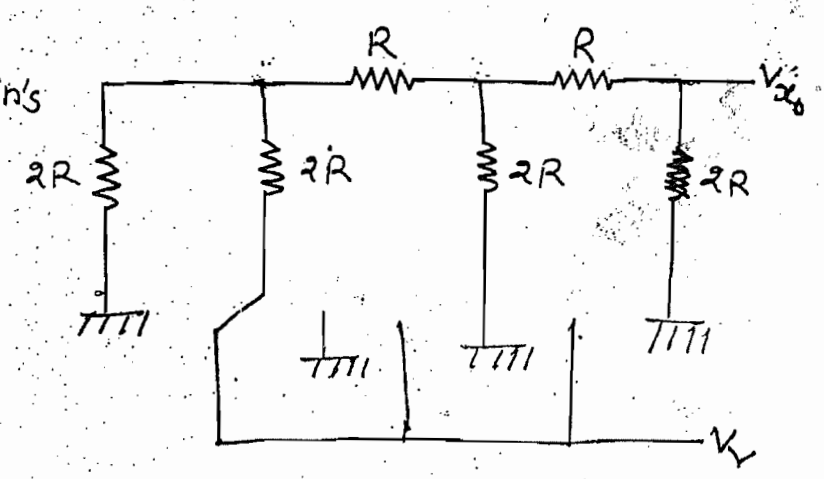
$$\boxed{V_o = R \times B \times G_f}$$

3-bit R-2-R Ladder N/W :-



$$V_x = V_{x_2} + V_{x_1} + V_{x_0}$$

Calculation of V_{x_0} :-



-By using S.T & thevenin's theorem

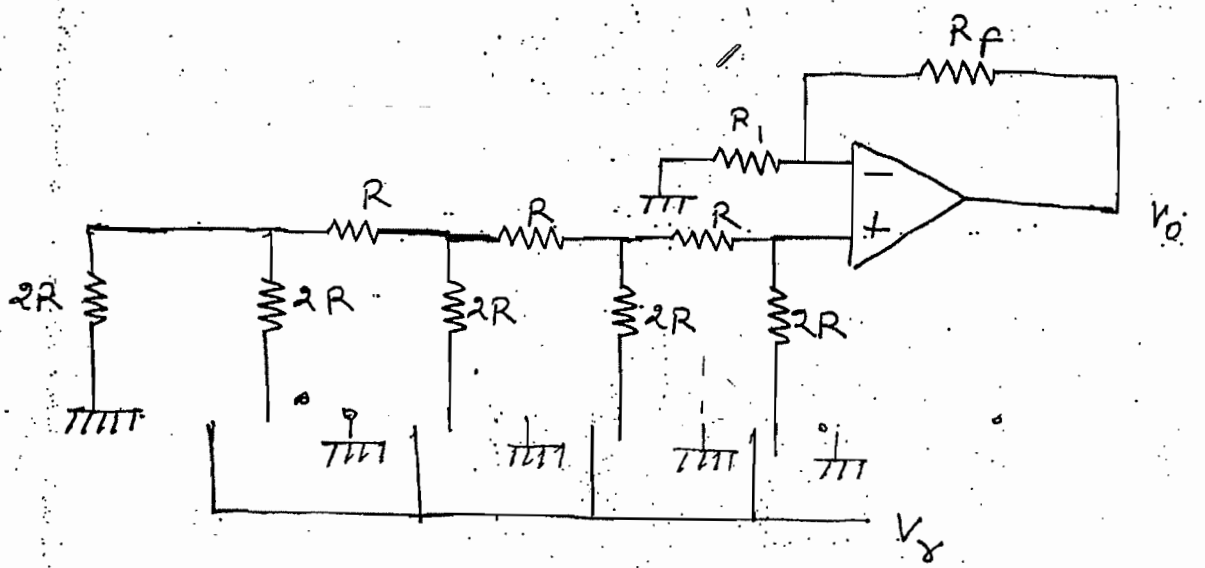
$$V_x = V_{x_2} + V_{x_1} + V_{x_0}$$

$$= \frac{V_r}{2} \times b_2 + \frac{V_r}{4} \times b_1 + \frac{V_r}{8} \times b_0$$

$$= \frac{V_r}{8} \left[4 \times b_2 + 2 \times b_1 + b_0 \right]$$

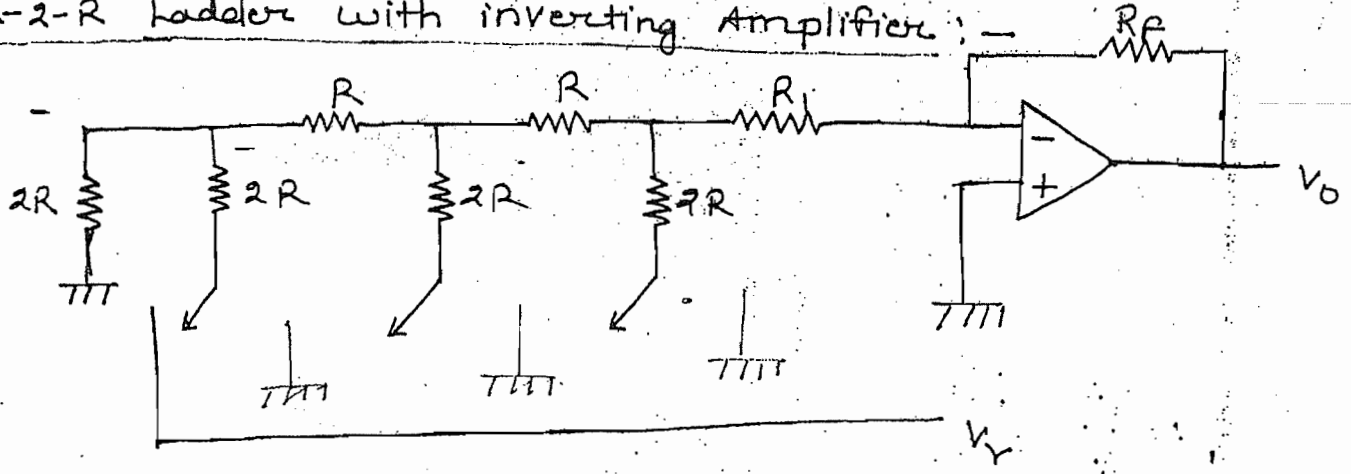
$$\boxed{V_x = \frac{V_r}{2^n} \times \left(\sum_{i=0}^{n-1} 2^i b_i \right)}$$

4-bit R-2-R Ladder DAC (With non-inverting Amp.) :-



$$\boxed{V_o = \frac{V_r}{2^n} \times \left(\sum_{i=0}^{n-1} 2^i b_i \right) \times \left(1 + \frac{R_F}{R_1} \right)}$$

R-2-R Ladder with inverting Amplifier :—

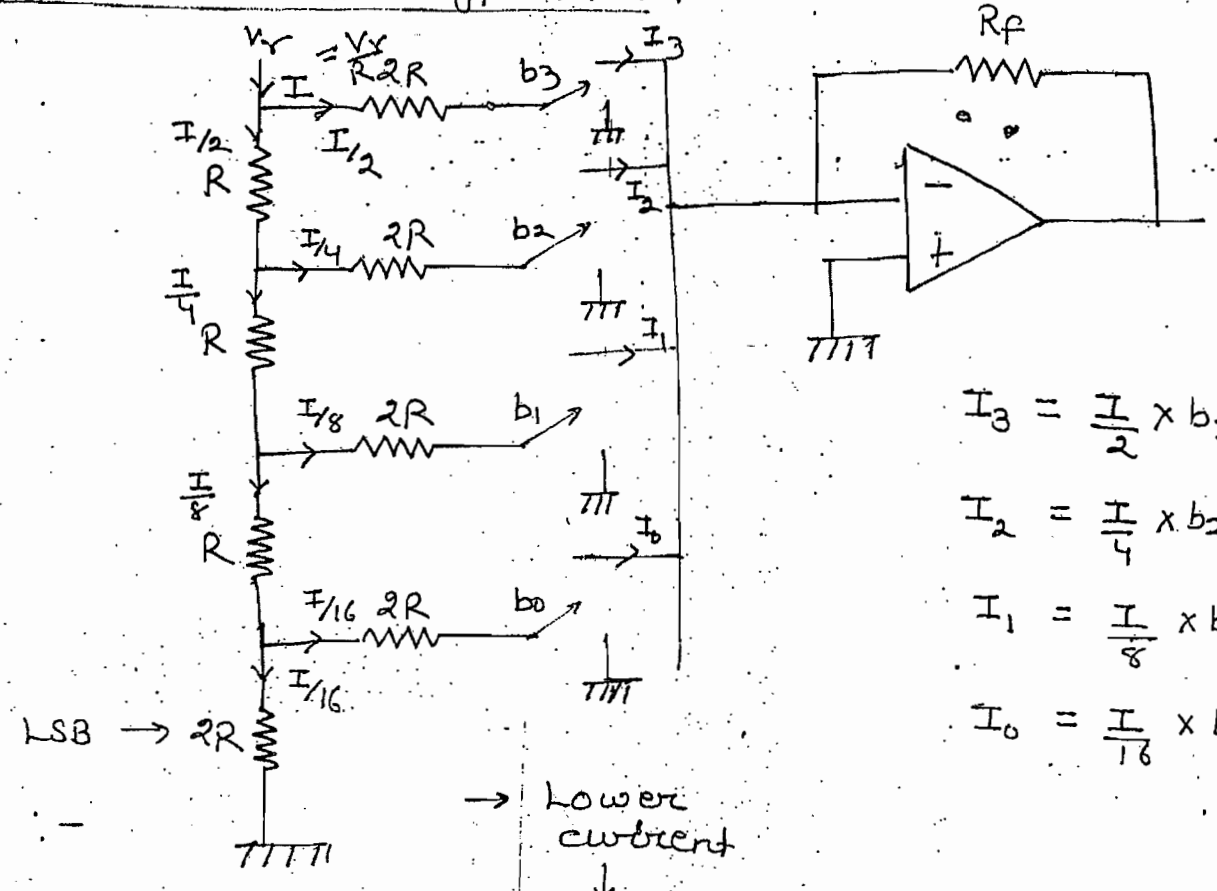


$$V_o = \frac{V_r}{2^n} \times \left(\sum_{i=0}^{n-1} 2^i b_i \right) \times \left(\frac{-R_f}{R_1 + R} \right)$$

$$V_o = (-R_f)\, I_f$$

$$I_f = \frac{V_r}{2^n} \times \left(\sum_{i=0}^{n-1} 2^i b_i \right) \left(\frac{1}{R_1 + R} \right)$$

Inverted ladder type DAC :—



→ Lower current → LSB

$$I_3 = \frac{I}{2} \times b_3$$

$$I_2 = \frac{I}{4} \times b_2$$

$$I_1 = \frac{I}{8} \times b_1$$

$$I_0 = \frac{I}{16} \times b_0$$

$$I_f = I_3 + I_2 + I_1 + I_0$$

$$= \frac{I}{2} \times b_3 + \frac{I}{4} \times b_2 + \frac{I}{8} \times b_1 + \frac{I}{16} \times b_0$$

$$= \frac{I}{16} \left[8 \times b_3 + 4 \times b_1 + 2 b_1 + b_0 \right]$$

$$\boxed{I_f = \frac{V_r}{2^n} \left(\sum_{i=0}^{n-1} 2^i b_i \right) \times \frac{1}{R}}$$

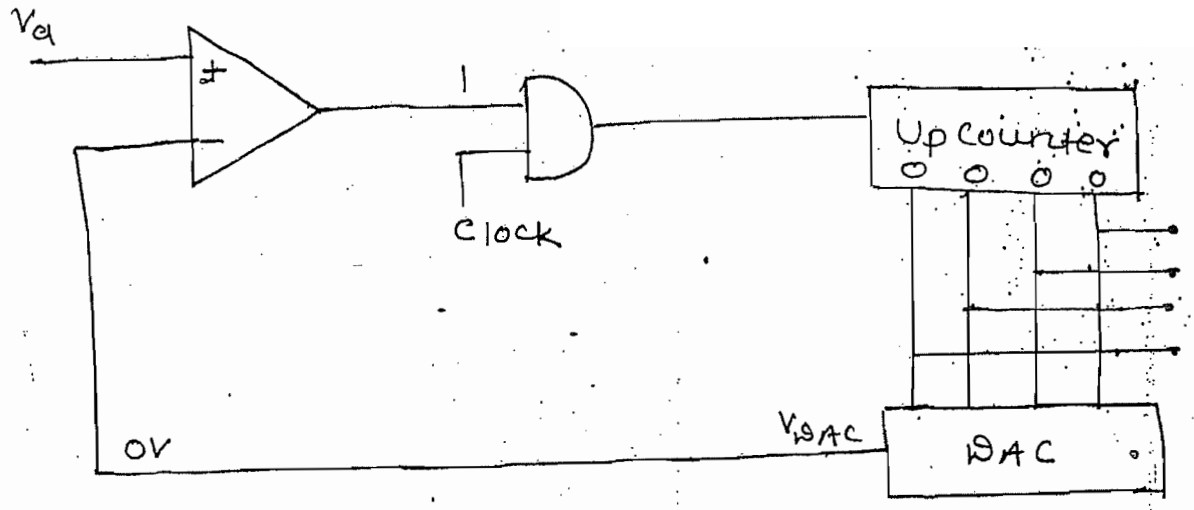
$$V_0 = (-R_f) \times I_f$$

$$\boxed{V_0 = \frac{V_r}{2^n} \times \left(\sum_{i=0}^{n-1} 2^i b_i \right) \times \left(-\frac{R_f}{R} \right)}$$

$$= R \times B \times G_1$$

ADC :-

Counter type ADC :-



Resolution

→ In counter type ADC shown in figure, a comparator is used in i/p to compare i/p analog voltage V_A with feedback reference voltage provided by DAC

→ Up counter is used to convert no. of clock pulse applied through AND gate.

Operation :-

→ When analog voltage is greater than feedback voltage, the o/p of comparator is logic 1 and the counter will continously increment

→ When analog voltage is less than feedback voltage, o/p of comparator is zero and counter will stop at this time and o/p of counter will provide binary value corresponding to input analog voltage.

eg :- Resolution = 1V i/p = V_a = 4.7V

comparing with 4.7V ⇒ $V_a > V_{DAC}$

counter increment & count = 0001

Process is continue and atlast if V_{DAC} = 5V then $V_a < V_{DAC}$ & i/p to AND gate = 0 & counter stop

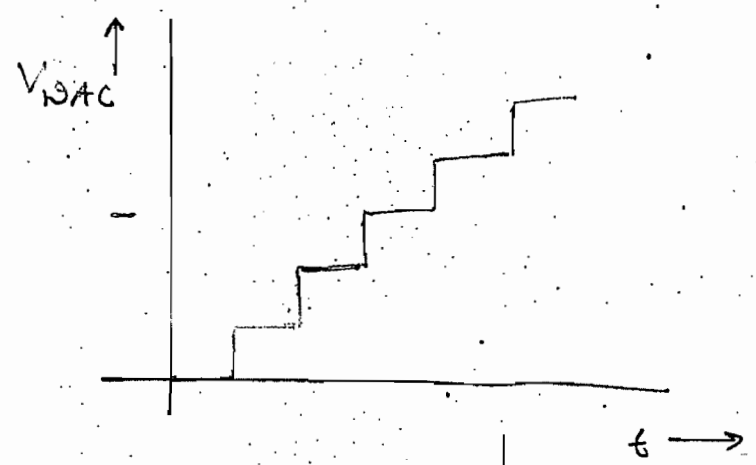
Error = 5 - 4.7 = 0.3V ≤ Resolution (1V)

→ In counter type ADC for n-bit conversion max. no of clock pulses used is $2^n - 1$.

→ Max. conversion time = $(2^n - 1)\, T_{CLK}$

→ Min " " = 1

→ Avg. " " = $\frac{2^n - 1 + 1}{2} = 2^{n-1}$

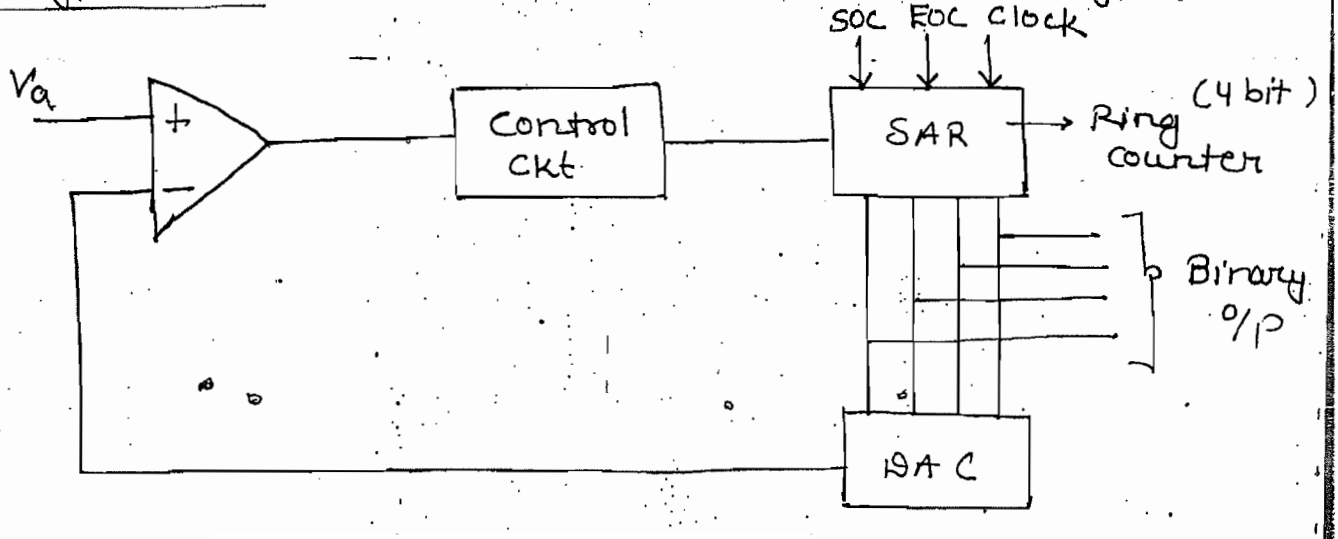


→ Counter type ADC is also known as Ramp type ADC

→ If up/down Counter is used in counter type ADC then it is known as Tracking type ADC

→ In this conversion time depends on i/p analog voltage

SAR type ADC :— (Successive Approximation type)



4-bit Ring counter

1. 1000
2. 0100
3. 0010
4. 0001

→ In SAR type ADC shown in fig, comparator used at i/p to compare i/p analog voltage with feedback voltage provided by DAC

→ In SAR ~~reg~~ register ring counter is used to set successively one by one bit from MSB to LSB with clock

→ control ckt is used to reset current or present set bit in SAR when analog voltage is less than feedback voltage

5.1V → SOC → 1000 → 0110 → 0101 → 0101 → EOC

12.3V → SOC → 1000 → 1100 → 1110 → 1101 → 1100 → EOC

→ In SAR type ADC, For n-bit conversion max no. of clock pulses used is 2^n

→ Max conversion time = $n \times T_{CLK}$

→ In this conversion time is independent of i/p analog voltage

→ SAR type is faster than counter type ADC

Flash type ADC :-

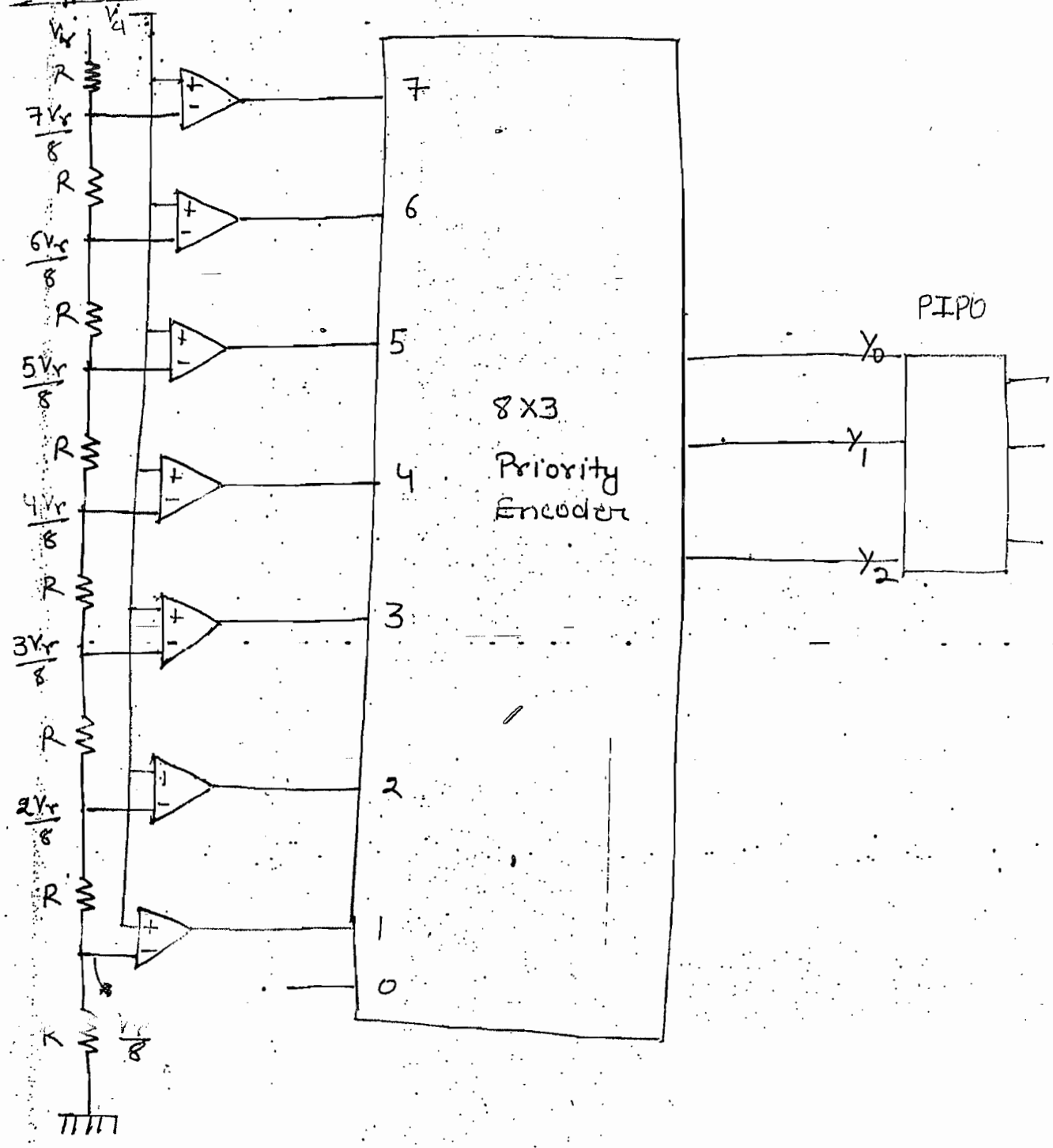
→ No of comparators are connected in parallel to increase speed of operation. Hence it is known as parallel compare type ADC

→ In this for n-bit conversion min no. of comparat required = $2^n - 1$

→ No of resistor = 2^n to provide reference voltage

→ $2^n \times n$ priority encoder

- 3-bit flash type ADC :-



→ Max. no. of clock pulses required for flash type → 1

→ Fastest type ADC among all

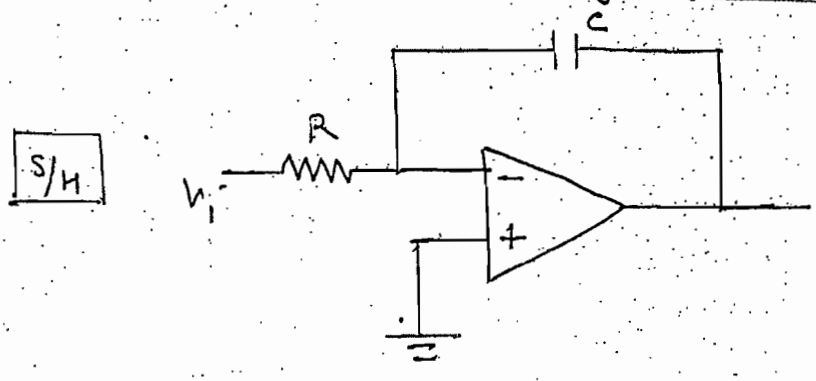
→ Range of analog

Range of analog voltage	Binary o/p $Y_2\ Y_1\ Y_0$
$V_a < \frac{V_r}{8}$	0 0 0
$\frac{V_r}{8} < V_a < \frac{2V_r}{8}$	0 0 1
$\frac{2V_r}{8} < V_a < \frac{3V_r}{8}$	0 1 0
$\frac{6V_r}{8} < V_a < \frac{7V_r}{8}$	1 1 0
$V_a > \frac{7V_r}{8}$	1 1 1

Disadvantage :-

→ large hardware is required.

Dual slope Integrating type ADC :-

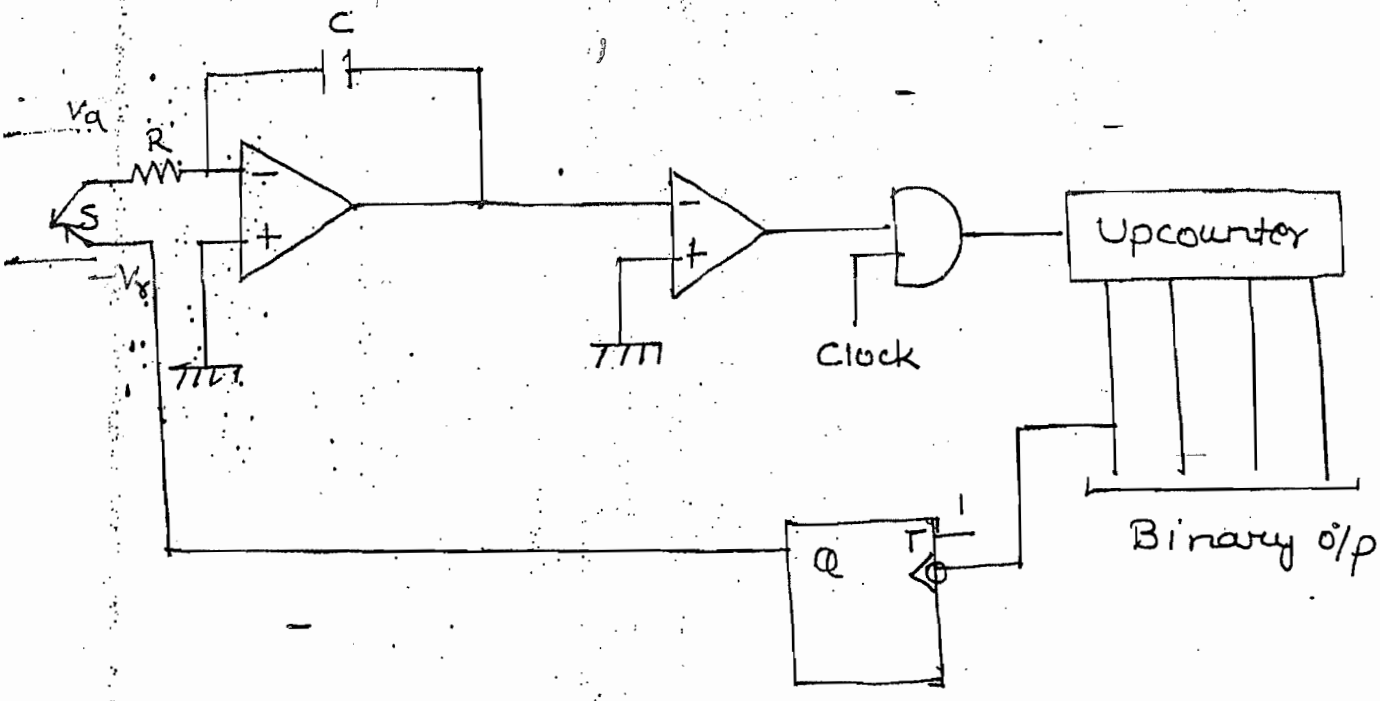


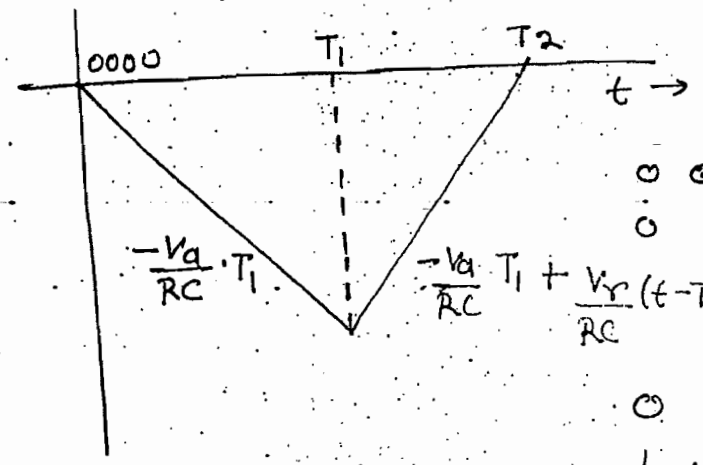
$$V_o = -\frac{1}{RC}\int V_i\,dt$$

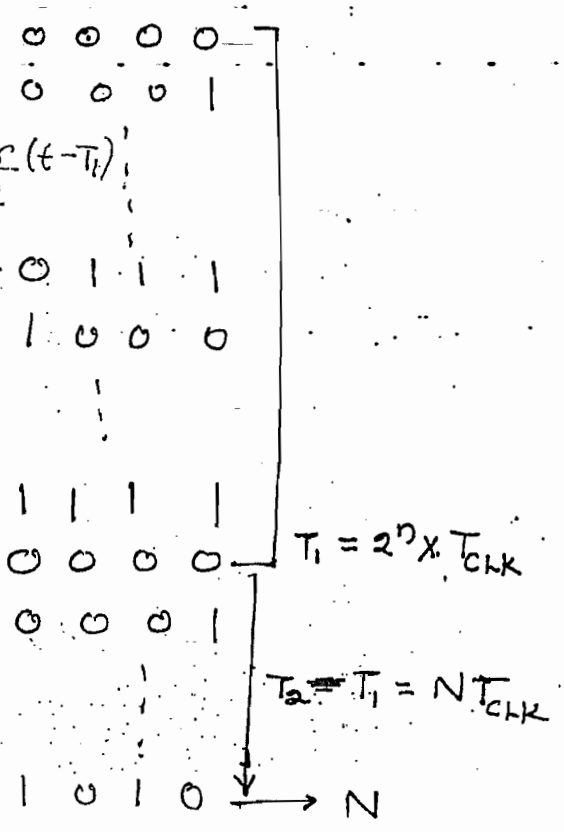
$$= \frac{-V_i}{RC}\int dt$$

$$= \left(\frac{-V_i}{RC}\right)t$$

Ramp







→ In Dual slope ADC shown in figure, an integrator is used to integrate analog voltage V_a and ref. voltage $-V_r$ ($|V_r| > V_a$)

→ When conversion is started counter & t-FF are reset to zero and the switch is connected to analog voltage V_A

→ During analog voltage integration o/p of the integrator is -ve voltage have

After 2^n clock pulse counter will reach count 0 again

→ At this time ($T_1 = 2^n$ clk), TFF toggles to logic 1 (0→ and switch is connected to reference voltage

→ During reference voltage integration, counter will continue clock pulses upto time T_2

→ At $T = T_2$ integrator o/p is +ve voltage and counter will stop

→ Let 'N' is the count when counter has stopped

$$\boxed{T_2 - T_1 = N \times T_{CLK}}$$

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

At $t = T_2$; $V_o = 0$

$$0 = \frac{-V_a}{RC} T_1 + \frac{V_r}{RC}(T_2 - T_1)$$

$$\frac{V_a}{RC} T_1 = \frac{V_r}{RC}(T_2 - T_1)$$

$$V_a \times 2^n T_{CLK} = V_r \cdot N T_{CLK}$$

$$\Rightarrow \boxed{V_a = \left(\frac{V_r}{2^n}\right) N}$$

N → count

$$V_a \propto N$$

→ Most accurate ADC among all

→ Ripple will not present

→ Mostly used in digital voltmeter

→ Slowest ADC among all

→ No. of clock pulses = $2^n + N$

→ Max. no. of clock pulse = $2^n + (2^n - 1)$
$\simeq 2^{n+1}$

	Clock Time	
Counter type	$\longrightarrow (2^n - 1)\, T_{CLK}$	$= T_S$
SAR type	$\longrightarrow (n)\, T_{CLK}$	$= T_S$
Flash type	$\longrightarrow 1\, T_{CLK}$	$= T_S$
Dual slope	$\longrightarrow (2^{n+1})\, T_{CLK}$.	$= T_S$

Logic families :-

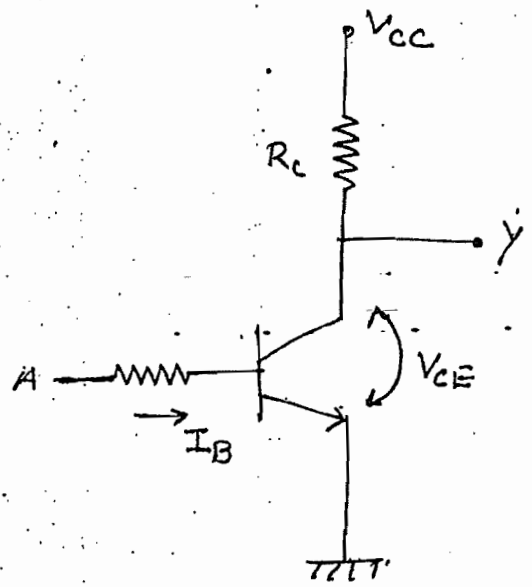
Transistor

J_{EB}	J_{CB}			
RB	RB	→ Cutoff	→ OFF	} switch
FB	FB	→ saturation	— ON	} switch
FB	RB	→ Active		
RB	FB	→ Rev. active		

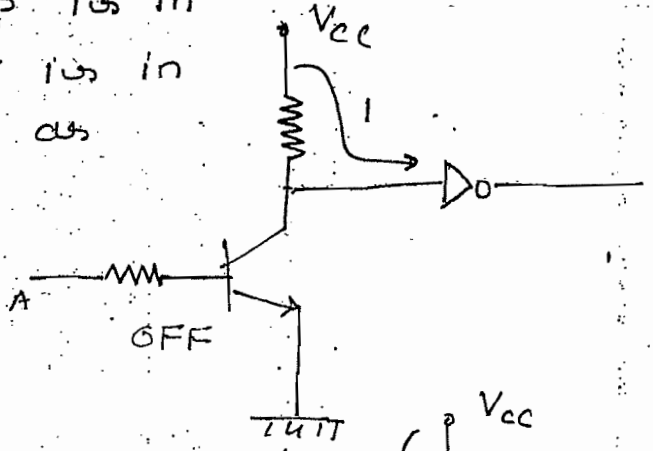
→ $V_{cc} = I_C R_C + V_{CE}$

→ $V_{CE} = V_{cc} - I_C R_C$ (↓ 0)

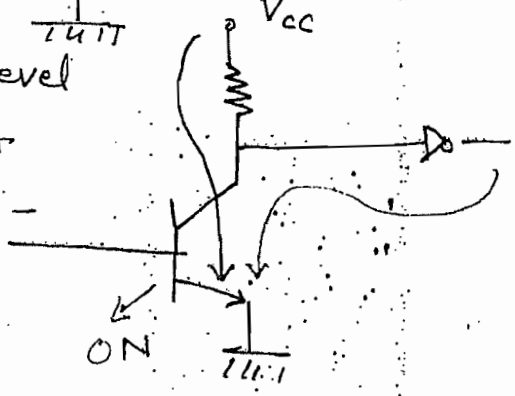
A	T_1	o/p Voltage	Y
0	OFF	$\approx V_{cc}$	1
1	ON	$V_{CE\,sat}$	0



→ When a logic gate o/p is in logic '1' level (transistor is in OFF state) then it act as current source.



→ When logic gate o/p is zero level (transistor is in ON state) then it will act as current sink



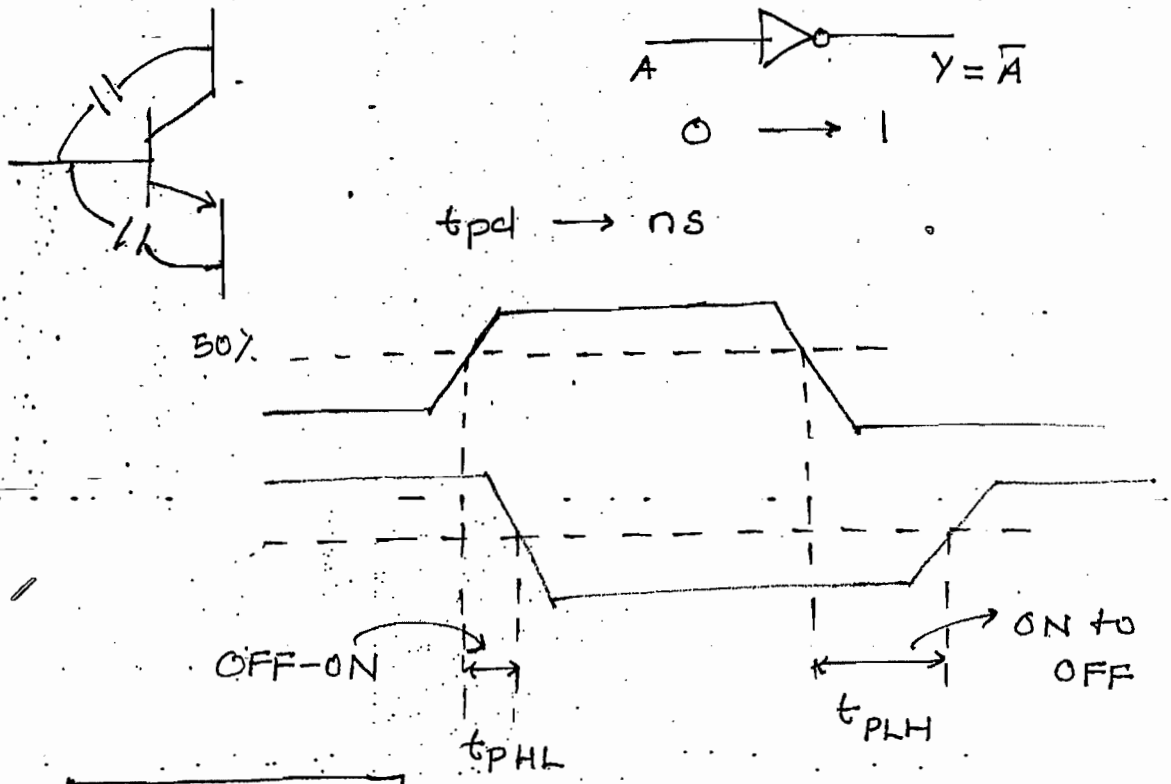
Characteristics of logic families :-

(i) t_{pd} (ii) P_{diss} (iii) Figure of Merit

(iv) fanout (v) Noise Margin (vi) Wired logic

(v) fan in.

(i) Propagation Delay (t_{pd}) :-



$$t_{PLH} > t_{PHL}$$

→ In transistor $t_{PLH} > t_{PHL}$ or $t_{ON-OFF} > t_{OFF-ON}$ due to storage time or saturation delay

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

Ques:- TTL, $t_{PHL} = 9$ ns | $t_{PLH} = 11$ ns, $t_{pd} = ?$

Soln:- $t_{pd} = \frac{9+11}{2} = 10$ ns.

(II) Power dissipation :-

→ indicates power dissipation by 1 logic gate

→ mW

→ $P_{diss} = V_{CC} I_{Cavg}$ Power diss. in bipolar

$$I_{Cavg.} = \frac{I_{C_{ON}} + I_{C_{OFF}}}{2}$$

→ $P_{diss} = V_{DD} I_{Davg.}$ Power diss in unipolar

(III) Figure of Merit :-

$$FOM = t_{pd} \times P_{diss}$$

ns x mW = pJ

→ Speed power product

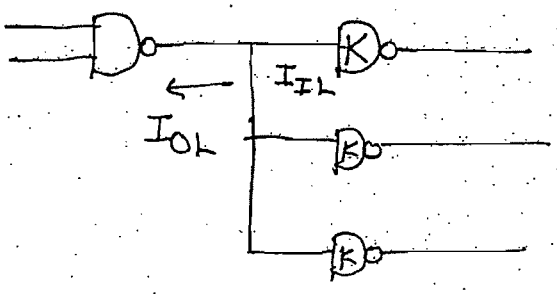
Level of integration :-

SSI	→	1-12 gates	Bipolar
MSI	→	13-99 gates	
LSI	→	100-1000 gates	MOS
VLSI	→	> 1000 gates	

(IV) Fanout :-

→ Max. no. of logic gate that can be driven by a logic gate output without effecting its operation.

→ Fanout depend on current source & current sink property of logic gate



$$\text{fanout}_H = \frac{I_{OH}}{I_{IH}}$$

$$\text{fanout}_L = \frac{I_{OL}}{I_{IL}}$$

$$\text{fanout} = (\text{fanout}_H, \text{fanout}_L)\ \text{min}$$

Ques:- In TTL logic families, $I_{OH} = 400\,\mu A$

$I_{LH} = 40\,\mu A$

$I_{OL} = 16\,mA$ $I_{IL} = 1.6\,mA$

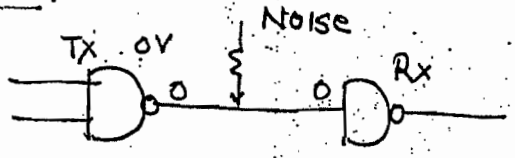
Fanout = ?

Soln:- $\text{Fanout}_H = \frac{400}{40} = 10$

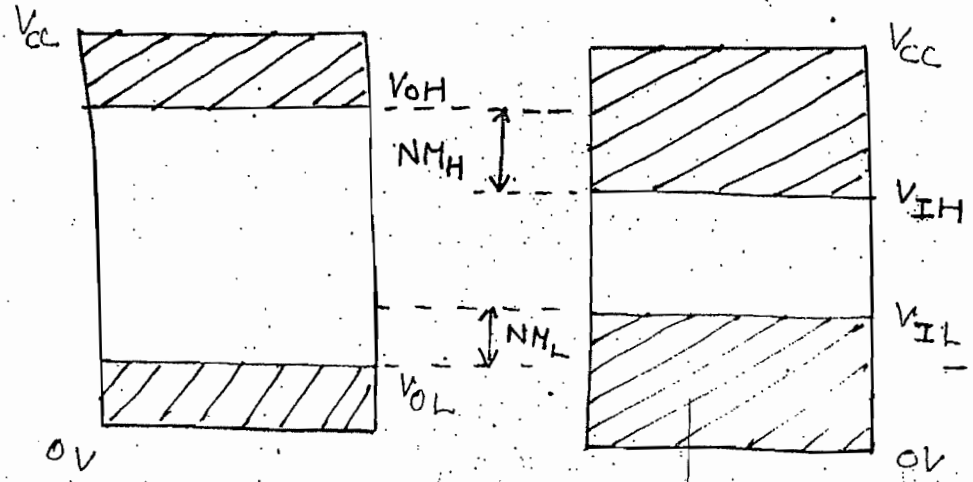
$\text{Fanout}_L = \frac{16}{1.6} = 10$

Fanout = 10

Noise Margin:-



→ Noise Margin is the max. noise voltage that can be added to a logic gate input which will not effect output

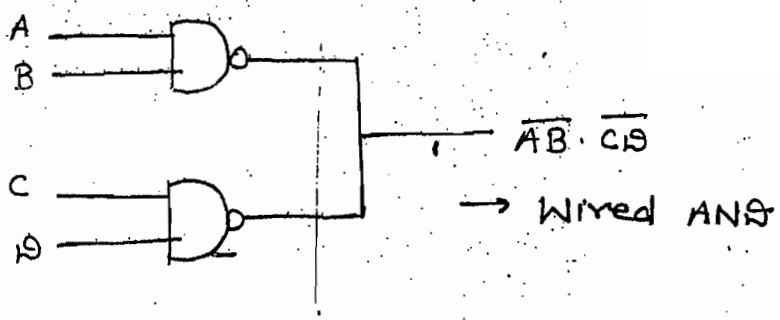


$$NM_H = V_{OH} - V_{IH}$$

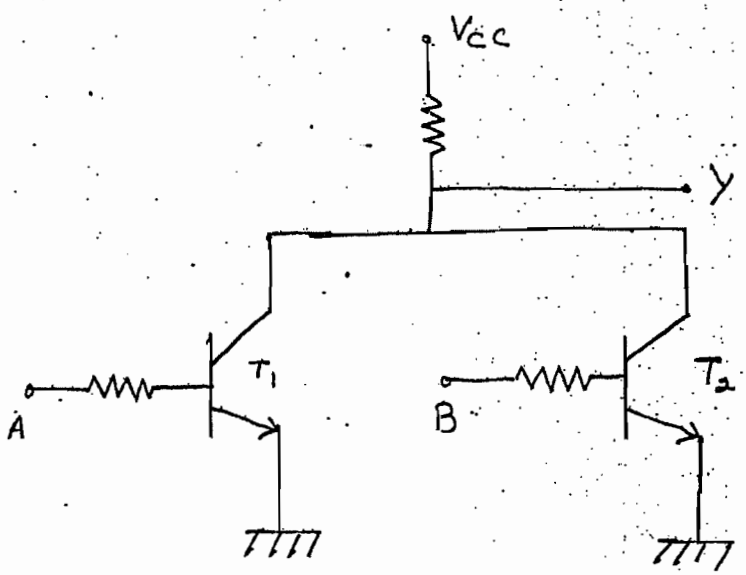
$$NM_L = V_{IL} - V_{OL}$$

$$V_{OH} > V_{IH} > V_{IL} > V_{OL}$$

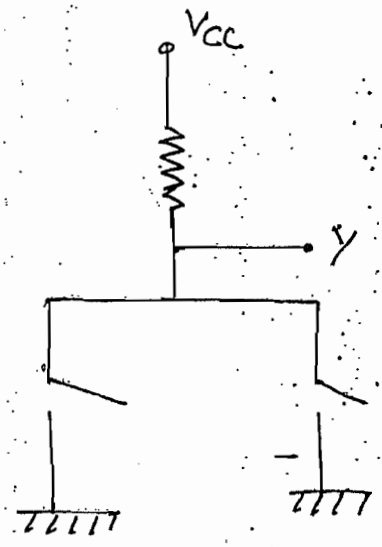
(VI) Wired Logic :-



Resistor Transistor Logic (RTL) :-

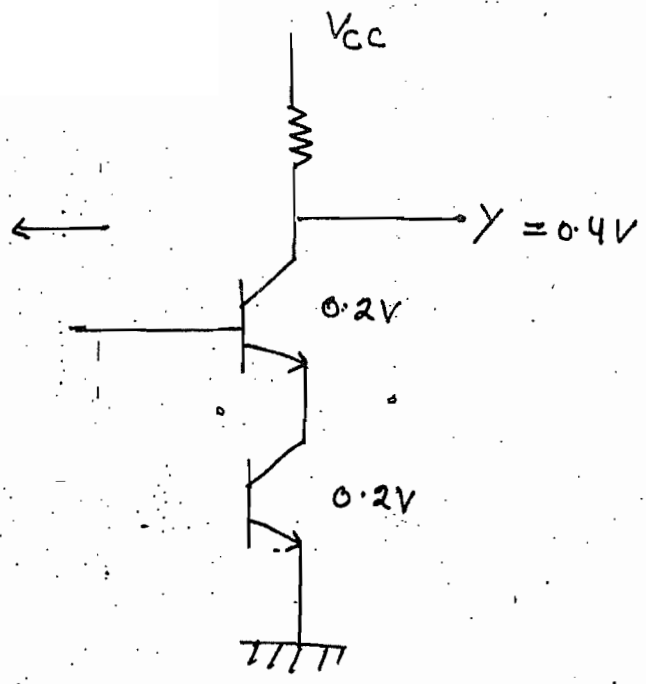


A	B	T_1	T_2	Y
0	0	OFF	OFF	1
0	1	OFF	ON	0
1	0	ON	OFF	0
1	1	ON	ON	0



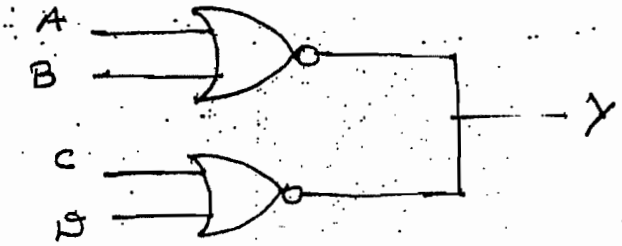
logic 0 = 0.2V

It is not used because $Y = 0.4V$



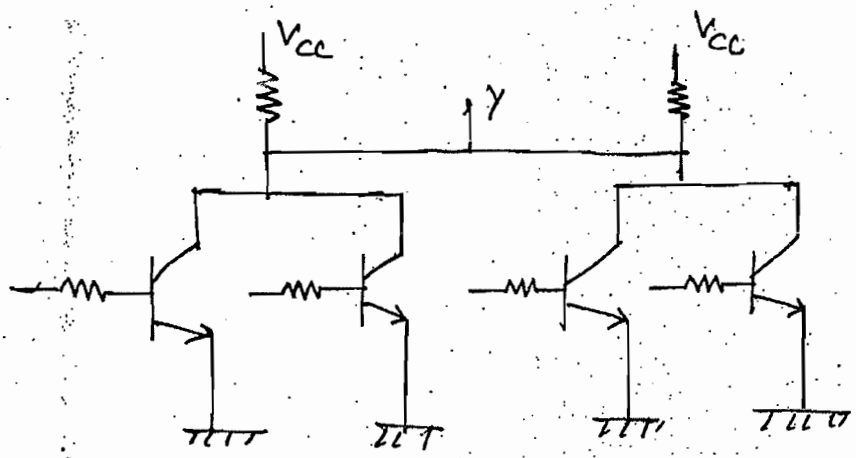
→ Basic gate ⇒ NOR
→ t_{pd} = 50 ns
→ P_{diss} = 10 mW
→ FOM = 500 pJ
→ Noise Margin = 0.2V
→ fanout = 3
→ Wired AND operation.

Ques:- The circuit shown in figure contains 2 RTL NOR gate then for given i/p, output Y is



Soln:- $Y = (\overline{A+B}) \cdot (\overline{C+D})$

A



A	B	C	D	y	
0	0	0	0	1	$\rightarrow \bar{A}\cdot\bar{B}\cdot\bar{C}\cdot\bar{D}$
0	0	0	1	0	
:					
1	1	1	1	0	

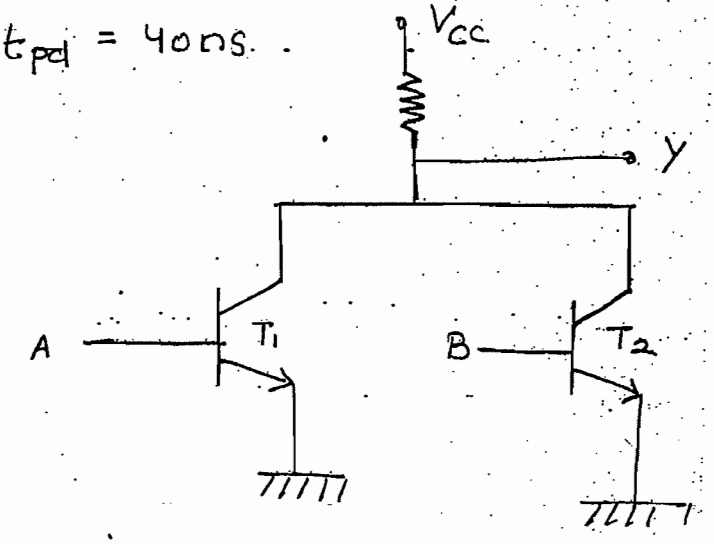
$\Rightarrow \quad Y = \bar{A}\cdot\bar{B}\cdot\bar{C}\cdot\bar{D} = \overline{(A+B)}\cdot\overline{(C+D)}$

Disadvantage :-

→ Speed of operation is less
→ Noise Margin is less
→ Fanout very less

Direct Coupled Transistor Logic (DCTL) :-

→ t_{pd} = 40ns.



Speed of operation depends on

(I) RC Time constant

(II) Storage time / saturation delay

→ In DCTL logic family base resistance is removed due to this RC time constant dec and speed of operation of Gate will improves when compared with RTL logic gate.

Disadvantage :-

Current Hogging :-

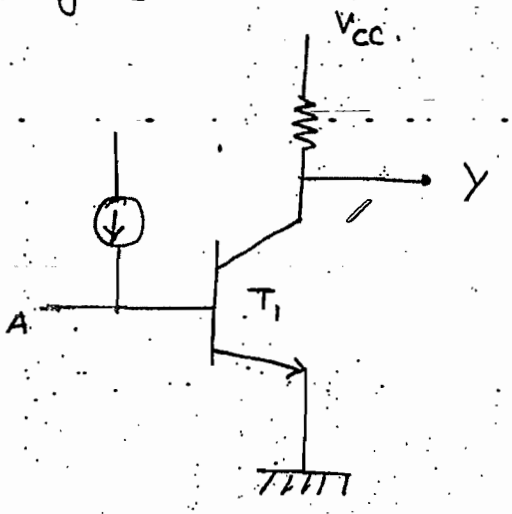
A	B	T_1	T_2	Y
0	0	OFF	OFF	1
0	1	OFF	ON	0
1	0	ON	OFF	0
1	1	ON	OFF	0

Current Hogging

To avoid current hogging I^2L logic family is used

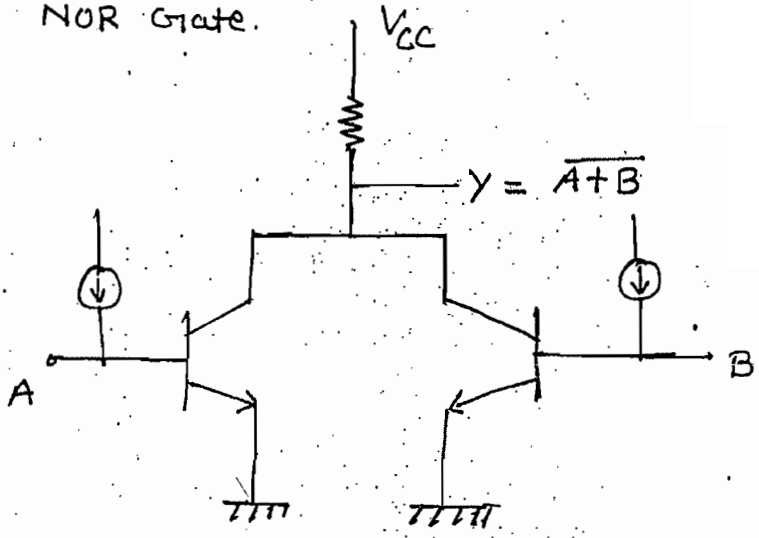
I^2L (Integrated Injection logic) :—

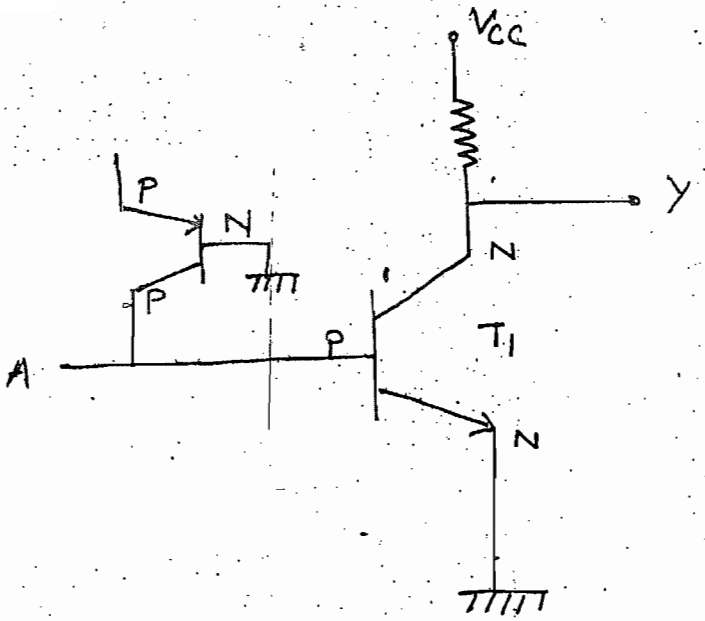
→ I^2L NOT gate



A	T_1	Y
0	OFF	1
1	ON	0

→ I^2L NOR Gate.





High level of integration

→ In I^2L logic family, PNP, NPN transistors are integrated together during IC fabrication. due to this transistor occupies less area (small geometry, low gain

→ I^2L is also called as MTL logic family

↓

Merged transistor logic

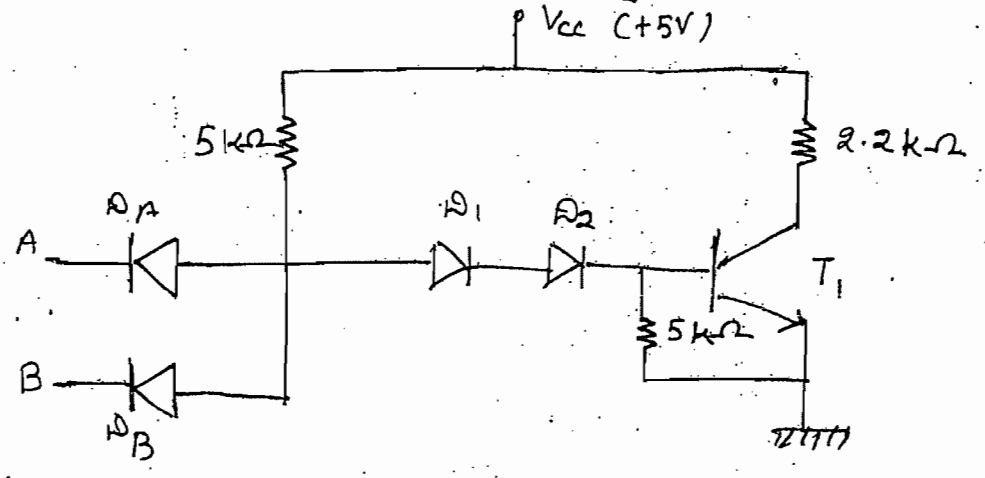
→ I^2L is mostly used in MSI and LSI circuit due to its higher level of integration

→ I^2L is main advantage is having best FOM among all logic families.

$$\boxed{FOM = 0.1PJ - 0.7PJ}$$

→ I^2L is available with multicollector o/p

Diode Transistor Logic (DTL) :—



A	B	D_A	D_B	D_1	D_2	T_1	Y
0	0	ON	ON	OFF	OFF	OFF	1
0	1	ON	OFF	OFF	OFF	OFF	1
1	0	OFF	ON	OFF	OFF	OFF	1
1	1	OFF	OFF	ON	ON	ON	0

→ Diode D_2 is used to improve Noise Margin

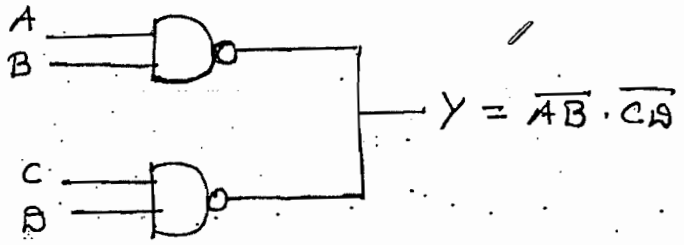
→ If one more diode D_3 is connected in series with D_2 then noise margin will improves whereas fanout will decreases.

→ t_{pd} - 30ns → Basic gate is NAND

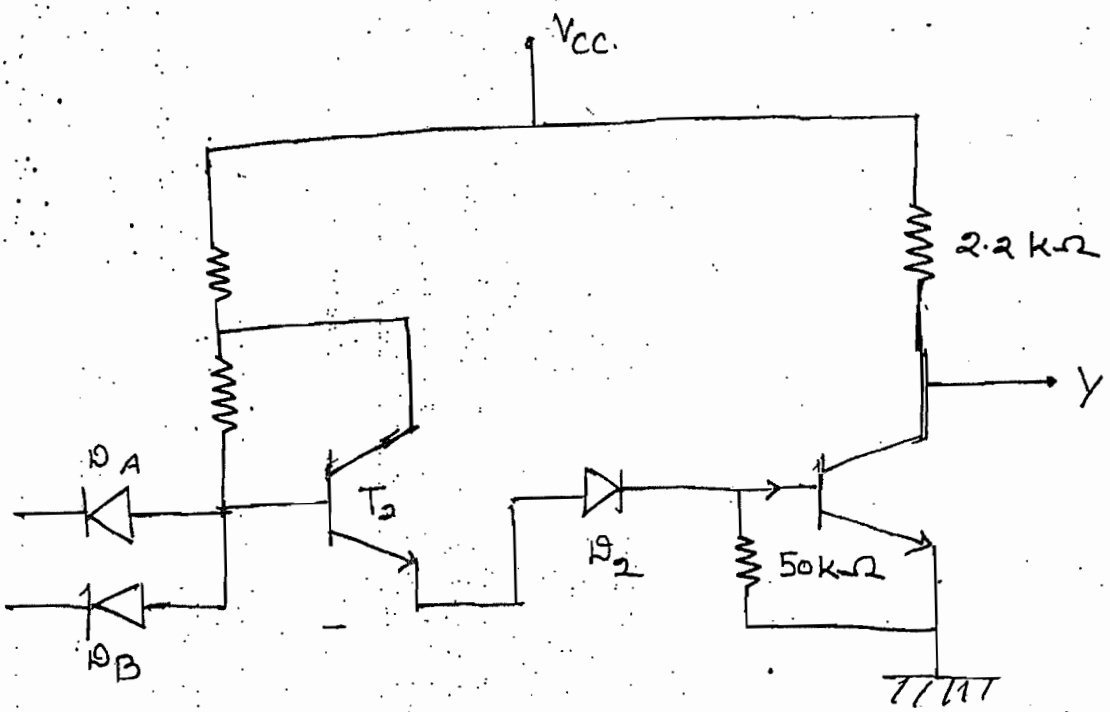
→ P_{diss} - 8 mW → fanout 3 → Basic RTL

→ FOM - 24 PJ → Wired AND

→ NM - 0.75V ↓



Standard RTL :-

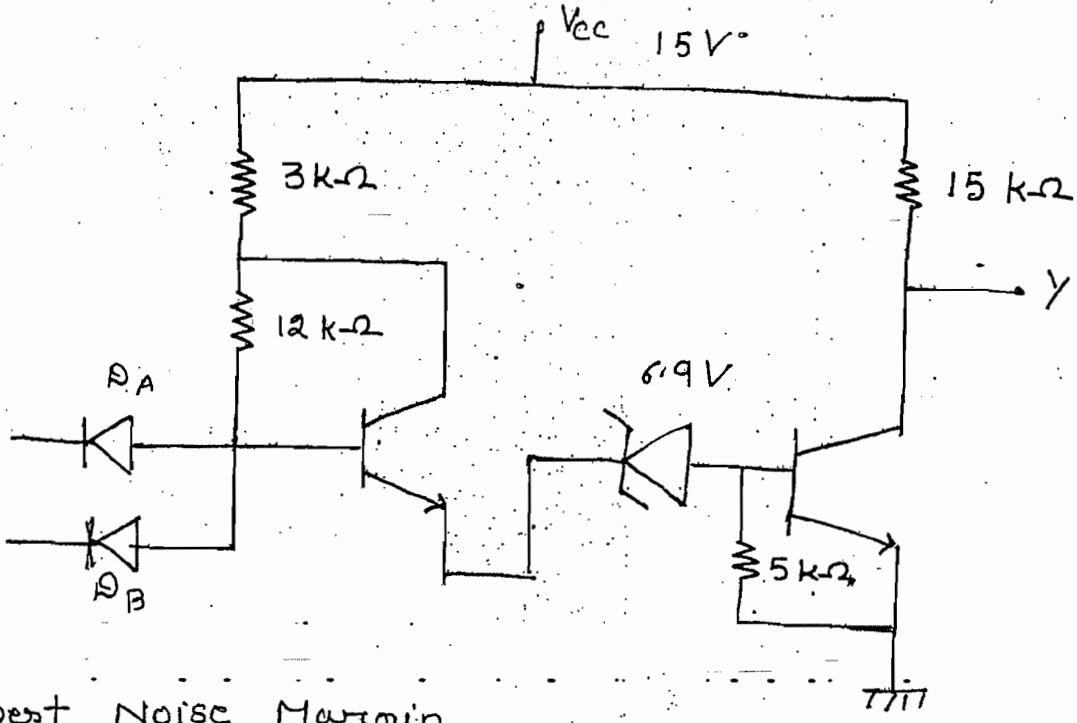


→ In RTL logic family to improve fanout diode D_1 is ~~or~~ replaced with transistor T_2 and resultant family

is known as standard TTL

→ DTL logic family is mostly used in monolithic IC fabrication

High Threshold Logic family (HTL) :-



→ Highest Noise Margin

NM ⇒ 4V − 5V

High noise immunity

→ logic 0 → 2V
logic 1 → 14V
] Higher voltage swing

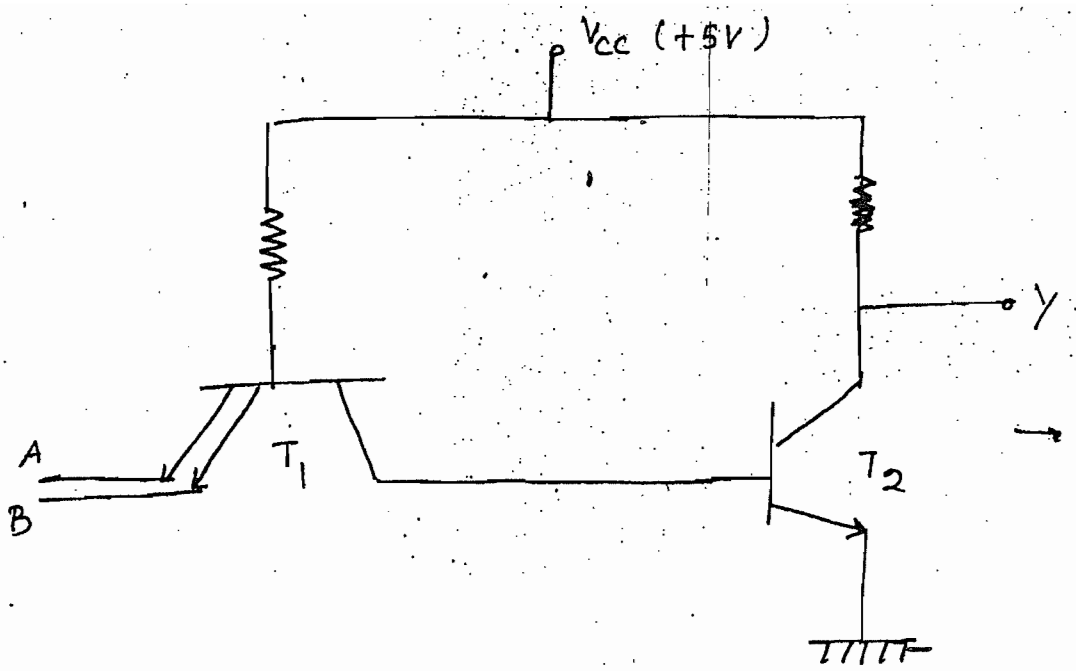
→ fanout − 10

→ Wired AND operation

→ t_{pd} = 90 ns

→ P_{diss} = 55 mW

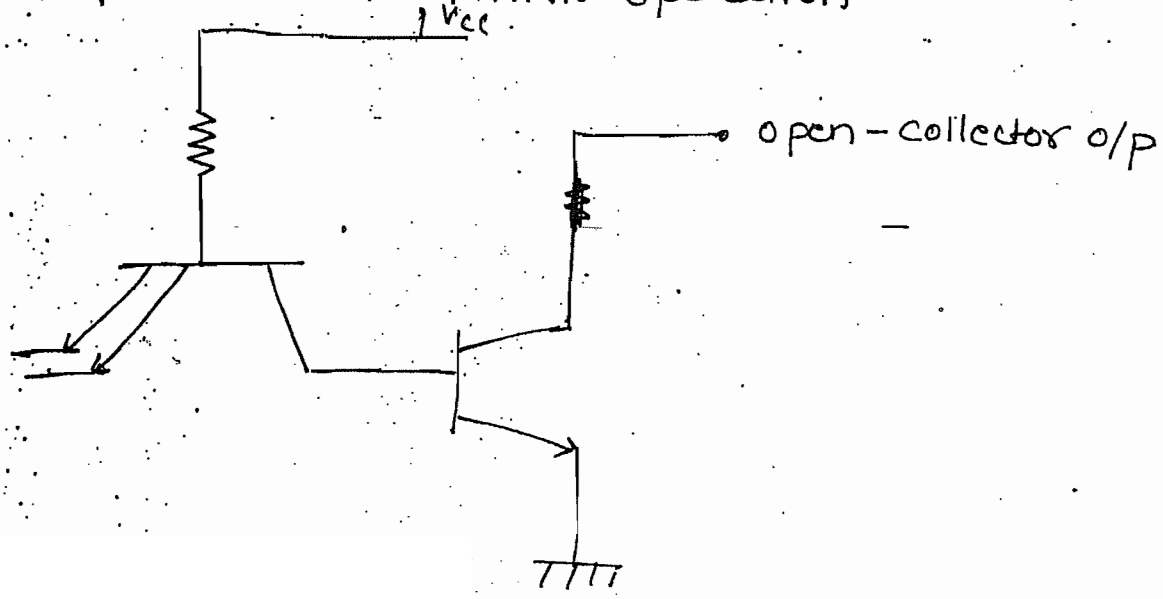
→ FOM ≃ 5000PJ



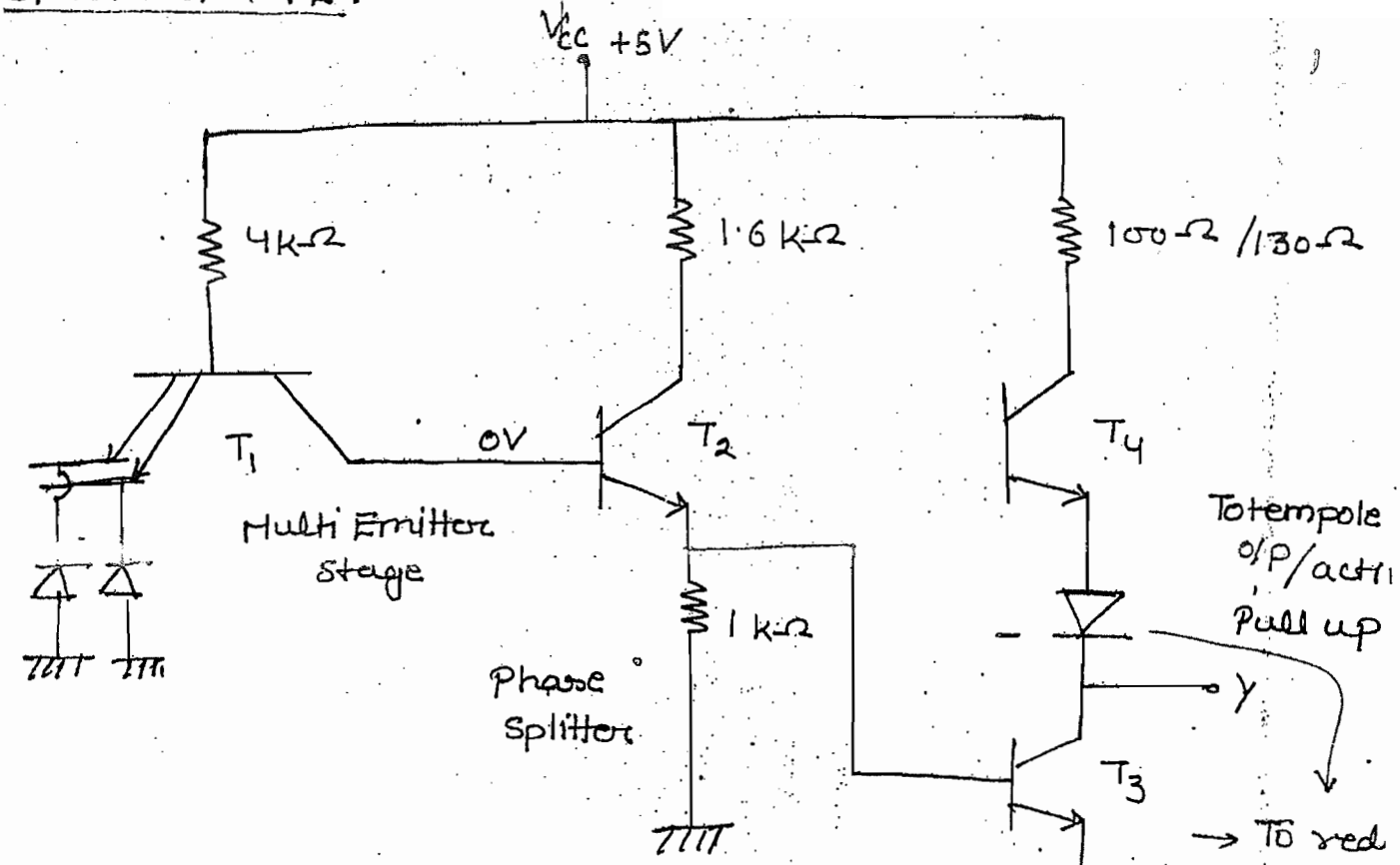
→ Basic TTL

A	B	T_1	T_2	Y
0	0	A	C	1
0	1	A	C	1
1	0	A	C	1
1	1	RA	S	0

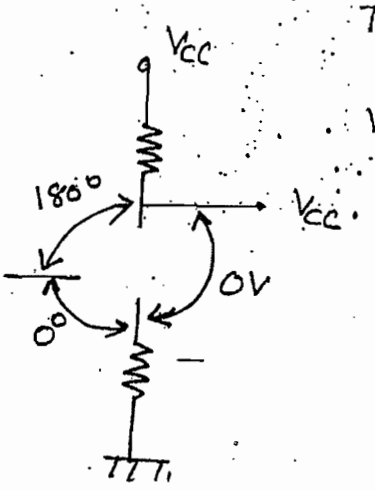
→ In TTL logic family open collector outputs are used to provide wired AND operation



Standared TTL :-



A	B	T_1	T_2	T_3	T_4	Y
0	0	A	C	C	S	1
0	1	A	C	C	S	1
1	0	A	C	C	S	1
1	1	RA	S	S	C	0



(Phase splitter)

→ NAND

→ t_{pd} – 10ns

→ P_{diss} – 10mW

→ fanout – 10

→ FOM – 100PJ

→ NM – 0.4V

**

$V_{OH} = 2.4V$

$V_{IH} = 2V$

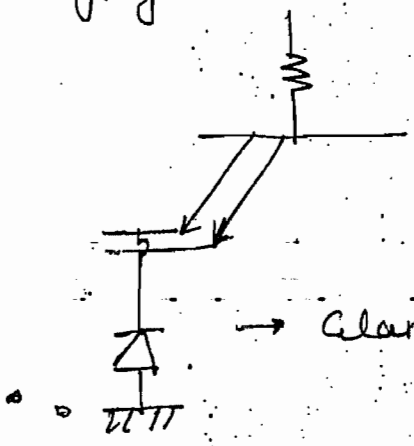
$V_{IL} = 0.8V$

$V_{OL} = 0.4V$

Advantage of totempole o/p :-

→ Higher speed of operation
→ Low power dissipation
→ Higher fanout
→ Totempole o/p or active pull up o/p is not used for wired operation.

→ Diode is used in totempole stage to cut-off transistor T_4 when T_3 is ON
→ Clamping diodes are used in i/p states/stages to protect Transistor T_1 during high freq transition or spikes or Ringing.

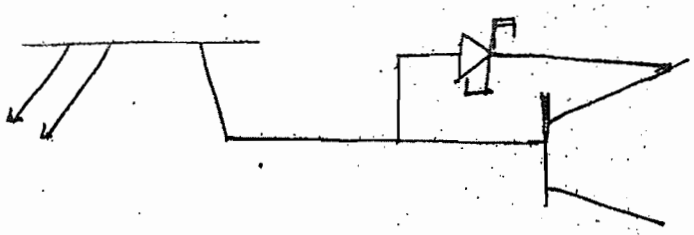


→ Clamping diodes.

TTL :-

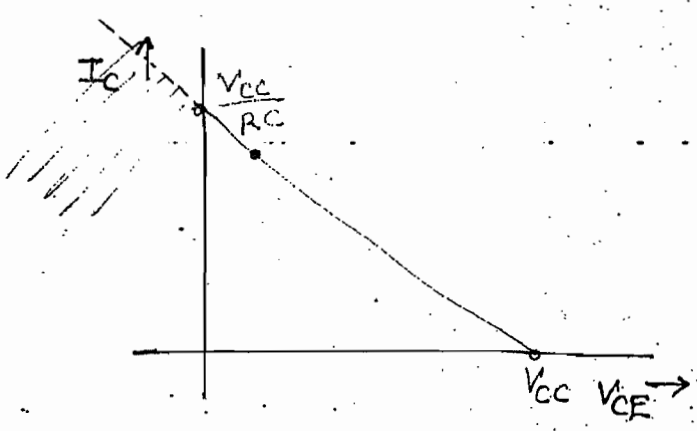
(I) Standard TTL → 10ns → 7400
(II) High speed TTL → 74H → 6ns / High Power diss.
(III) Low power TTL → 74L → 22ns
(IV) Schottky TTL → 74S → 2ns

→ In TTL logic family if external resistors values are reduce then RC value will decrease and speed of operation increase & resultant logic family known as high speed TTL
→ In low power TTL, resistor values are increased to reduce power dissipation.
→ If schottky diode is placed b/w base and collector terminal of each transistor in TTL then it remove storage time or saturation delay. Due to this speed of operation will increase and resultant logic family known as schottky TTL.

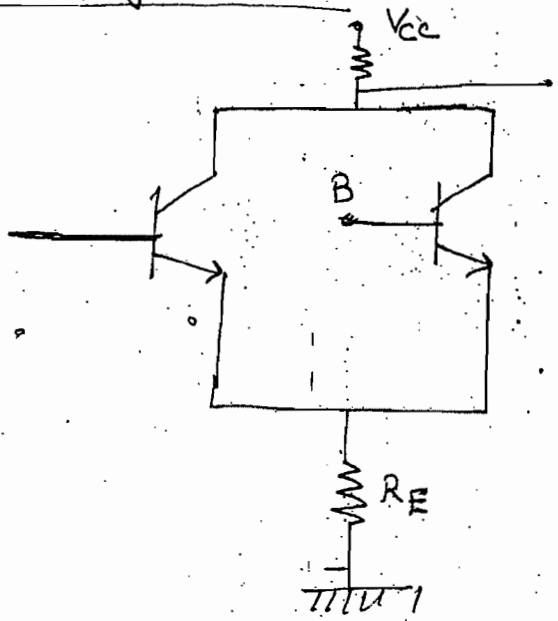


→ ALS (Advanced low power schottky)

→ LS (Low power schottky)



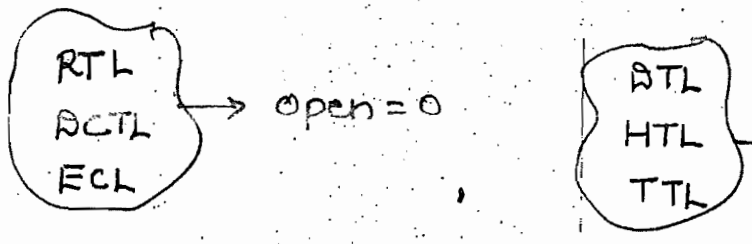
Emitter coupled logic (ECL):-

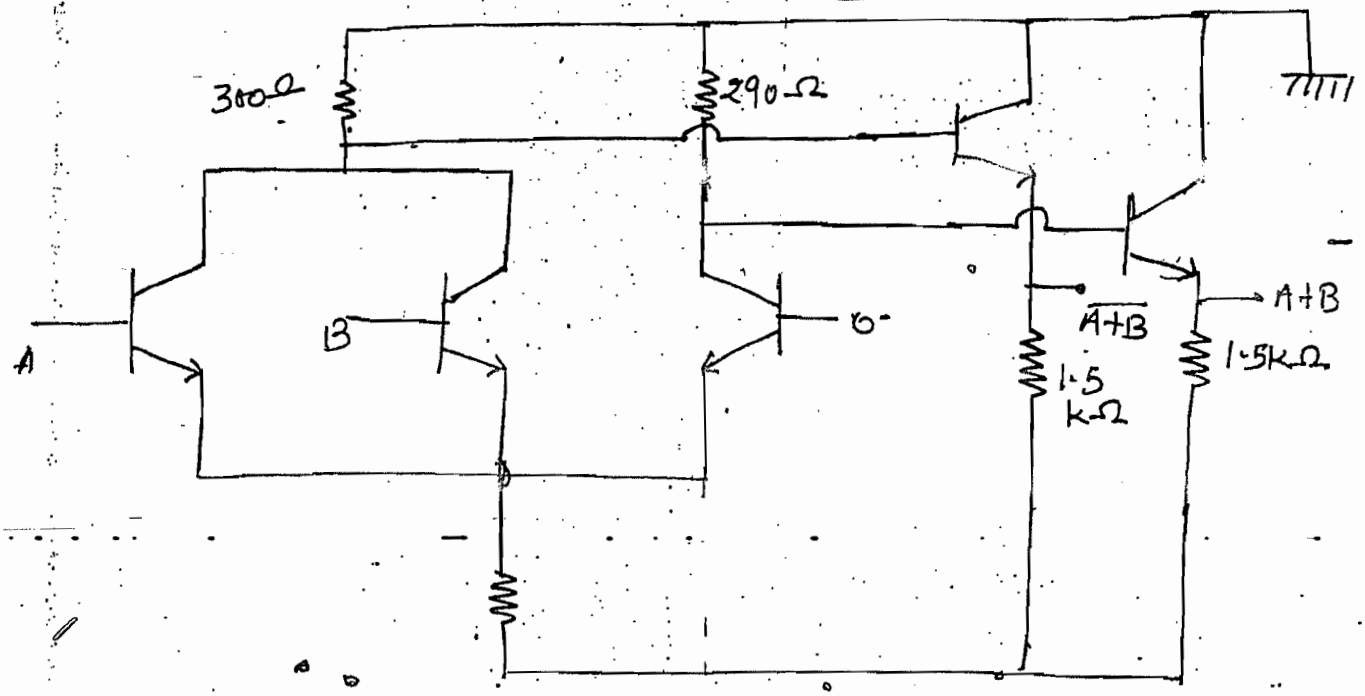


→ Simple ECL

→ In ECL logic family transistor are operated cut-off and active mode.

→ Fastest logic family because is non-saturated





→ ECL basically contains two stages:-

(i) Differential Amplifier stage

(ii) CC (or) Emitter follower

→ To use differential amplifier, ~~component~~ complement o/p are available in ECL

→ NOR/OR Gate → ~~P_{diss} = 55 pJ~~ P_{diss} = 55 ~~mJ~~ mW

→ t_{pd} = 1ns → FOM → 55 pJ

→ fanout = 25

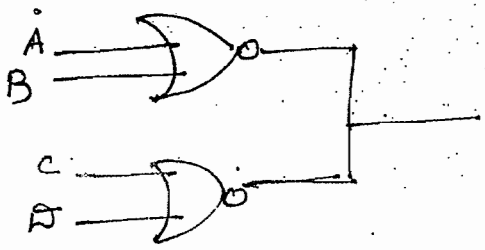
→ ECL uses (-ve) power supply. Due to this power power supply ripple / noise / spikes / Glitches will not effect operation of logic gate.

→ logic 0 = -1.7V
logic 1 = -0.8V
} +ve logic

V_{OL}	$= -1.7V$
V_{IL}	$= -1.4V$
V_{IH}	$= -1.1V$
V_{OH}	$= -0.8V$

→ ECL will provides wired OR operation.

Ques:- The ckt shown in figure use ECL, O/P Y is



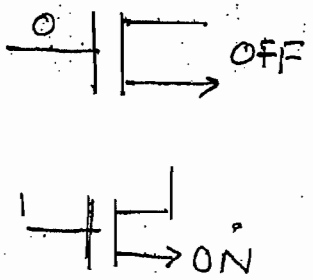
Ans:- $Y = \overline{A+B} + \overline{C+D}$

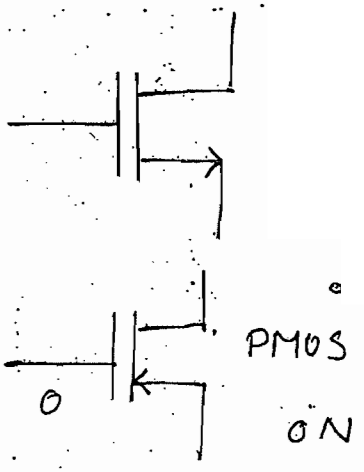
→ ECL is also known as current mode logic family (CML)

Disadvantage:-

→ High power dissipation.

NMOS:-





1 → OFF

PMOS

logic 0 → ON

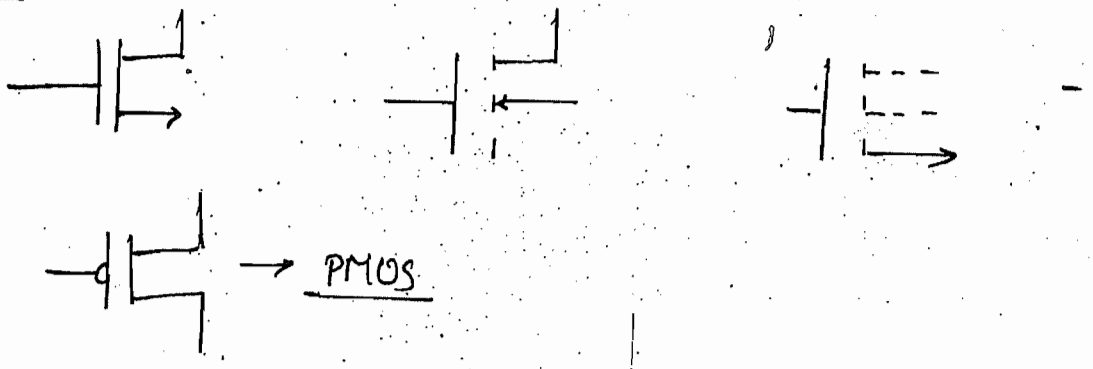
1 → OFF

N-MOS

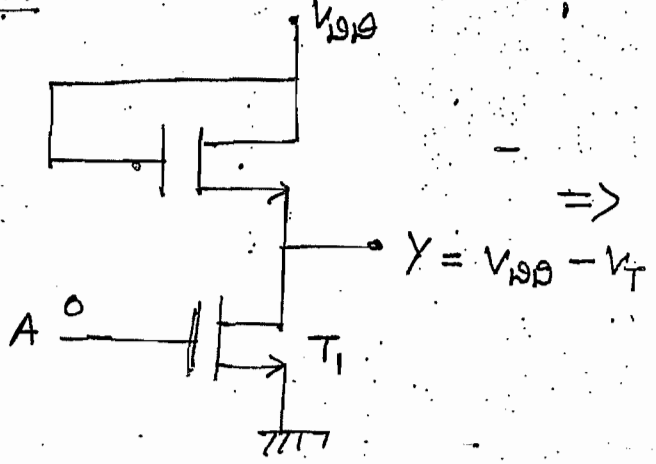
0 → OFF

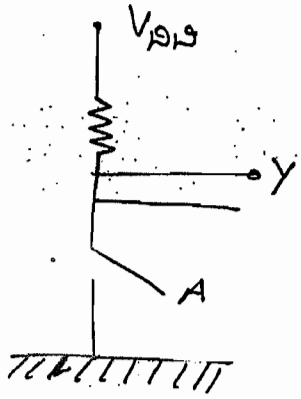
1 → ON

NMOS :-



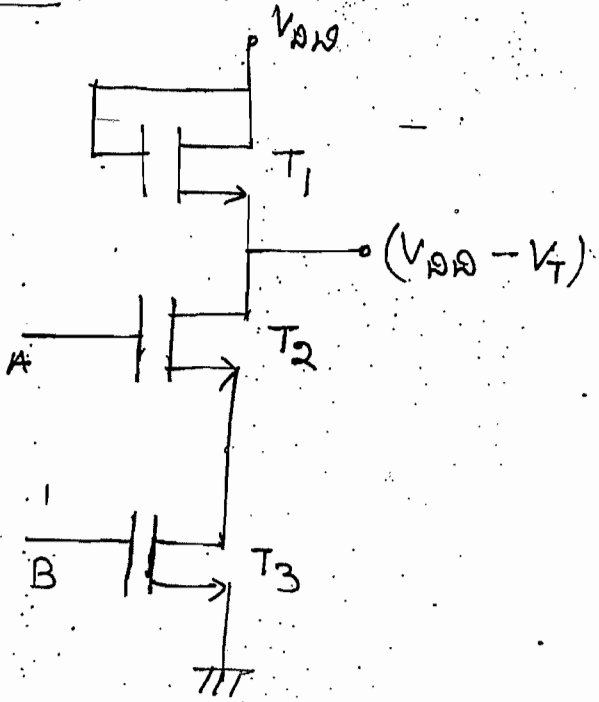
NOT :-

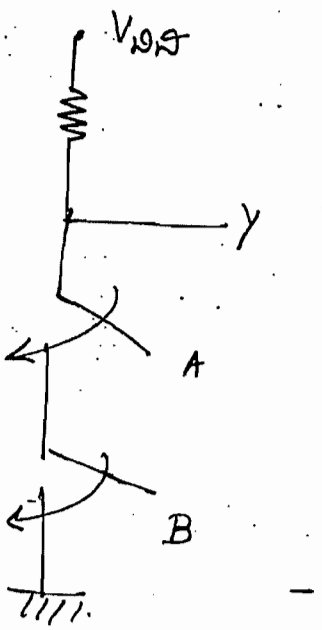




A	T_1	Y
0	OFF	1 → $V_{DD} - V_T$
1	ON	0 → (0 V)

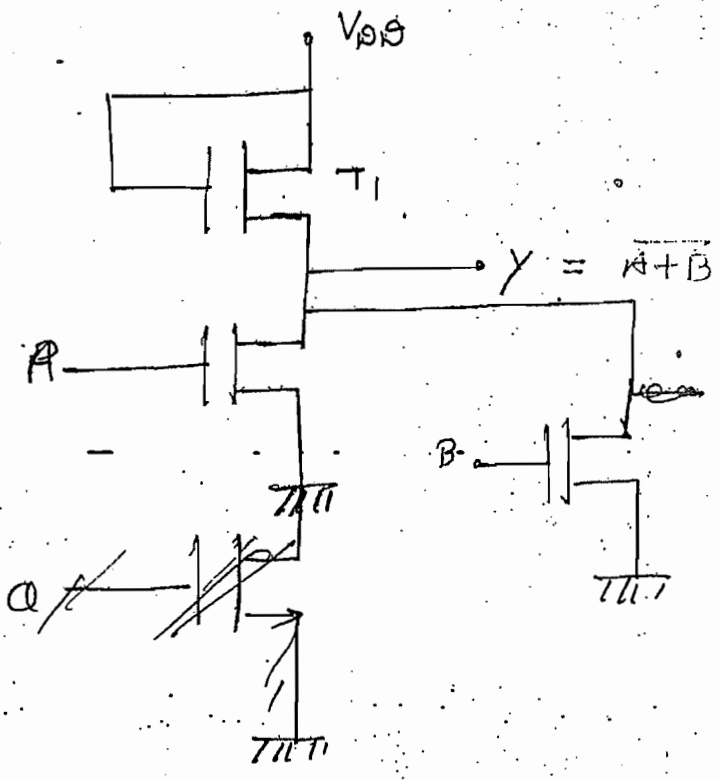
NAND Gate :-





A	B	T_2	T_3	Y
0	0	OFF	OFF	1
0	1	OFF	ON	1
1	0	ON	OFF	1
1	1	ON	ON	0

NOR :-



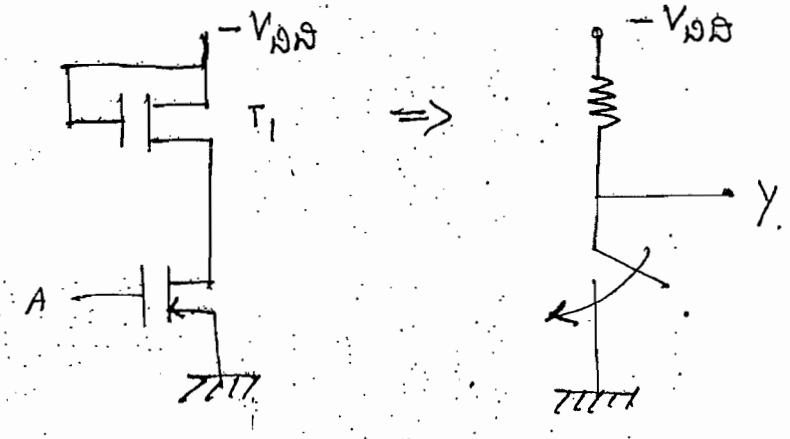
NMOS :-

→ t_{pd} ⇒ 250 ns → P_{diss} = 1 mW

→ FOM ⇒ 250 → fanout - 5

→ NM = 1.5V

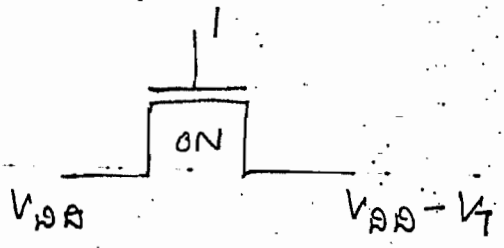
PMOS :-

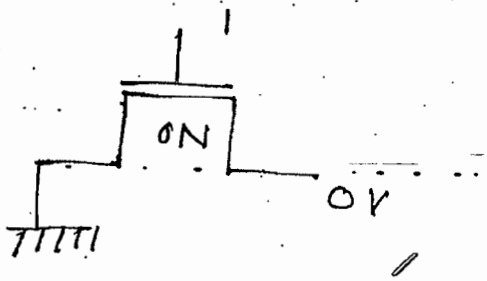


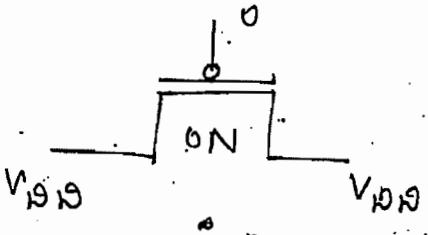
A	T_2	O/p Voltage	Y
0	- ON	GND (V_T)	1
1	OFF	$-V_{DD}$	0

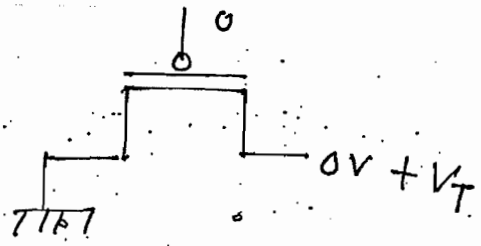
NMOS → Strong '0'
NMOS → Weak '1'

PMOS → Strong '1'
PMOS → Weak '0'

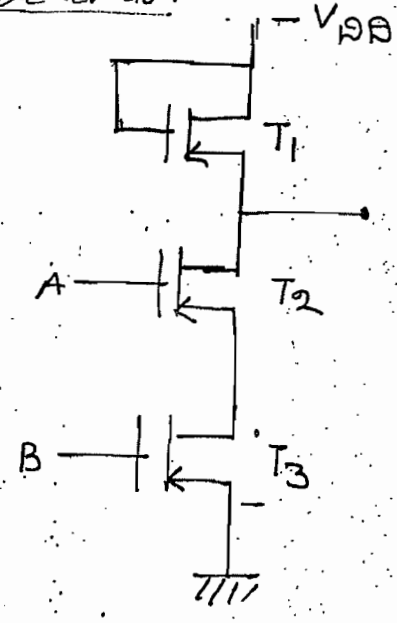








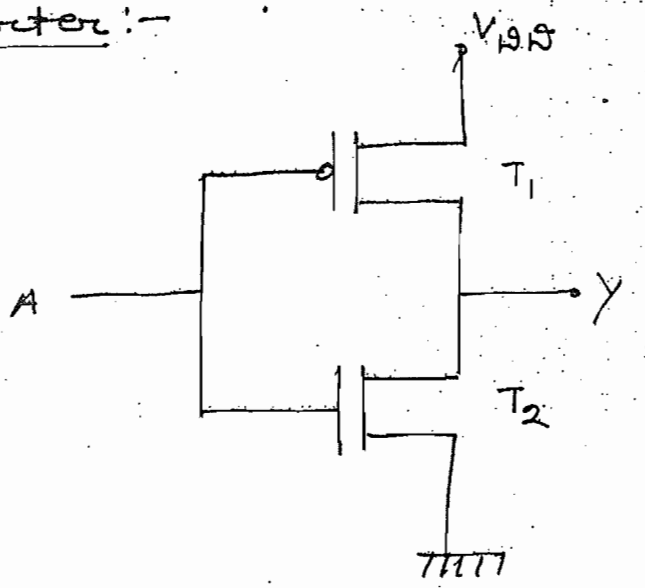
NOR ~~NAND~~ (CMOS) :—

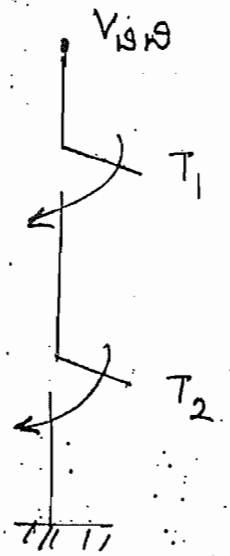


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

CMOS :-

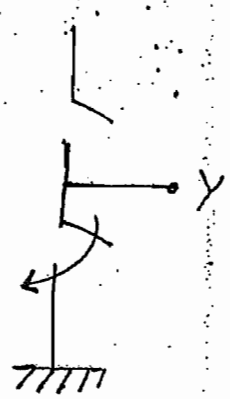
Inverter :-





(logic 0)

logic '1'



A	T_1	T_2	Y
0	ON	OFF	1 (V_{DD})
1	OFF	ON	0 (0V)

→ Lowest Power dissipation

$$P_{diss} = 0.01\ mW$$

→ $t_{pd} = 70\ ns$

→ FOM = 0.7 PJ

→ fanout = 50

→ $NM \approx \frac{V_{DD}}{2}$

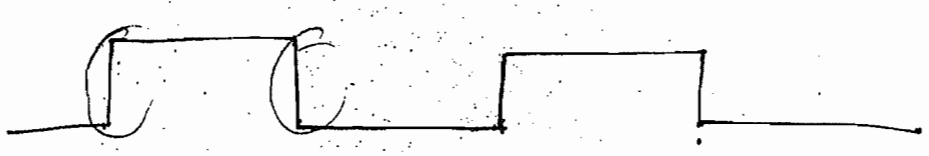
→ In CMOS there are two types of power dissipation (1) Static Power dissipation :-

↓

is present during logic 0 or logic '1' input

→ Dynamic Power dissipation :-

is present during 0 to 1 / 1 to 0 transition.

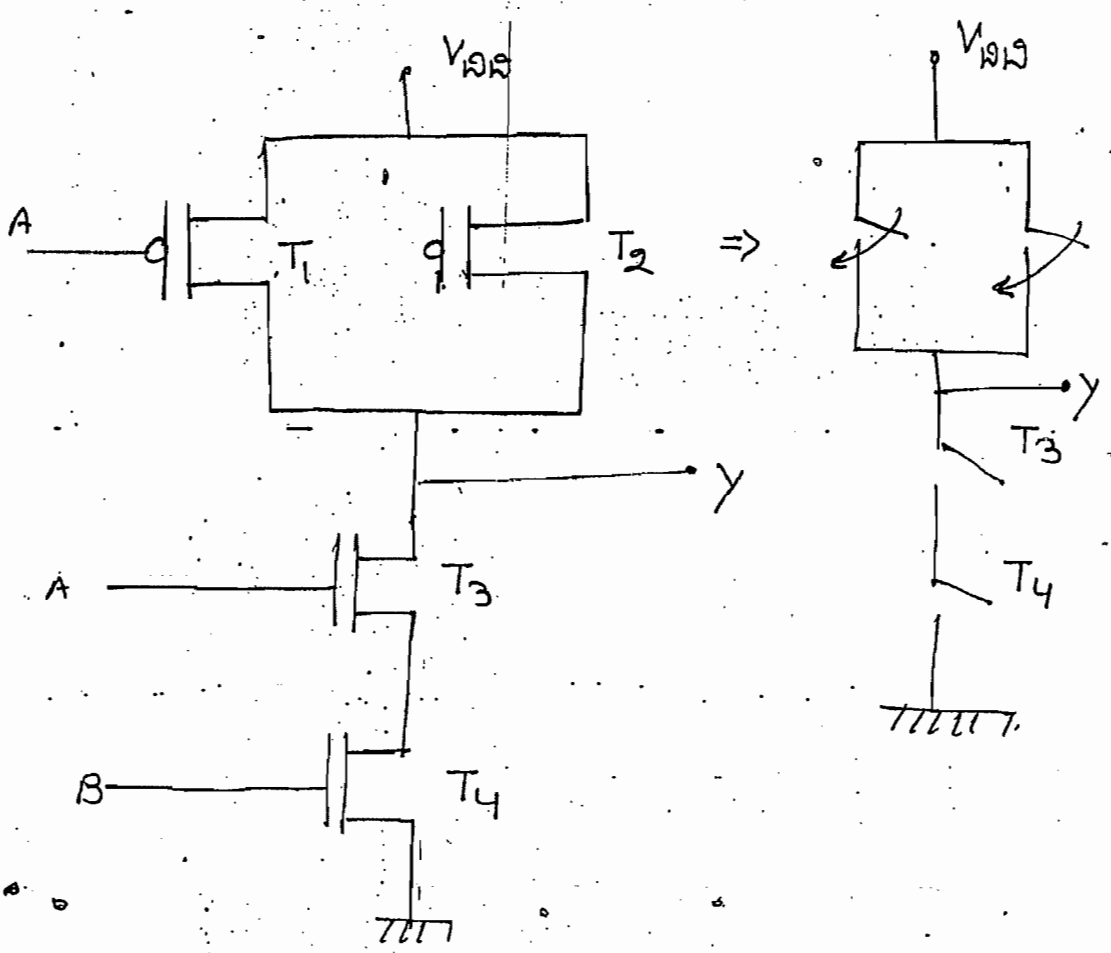


Dynamic
Power diss

Dynamic Power dissipation	$= \frac{1}{2} C f V_{DD}^2$

CMOS :-

NAND :-

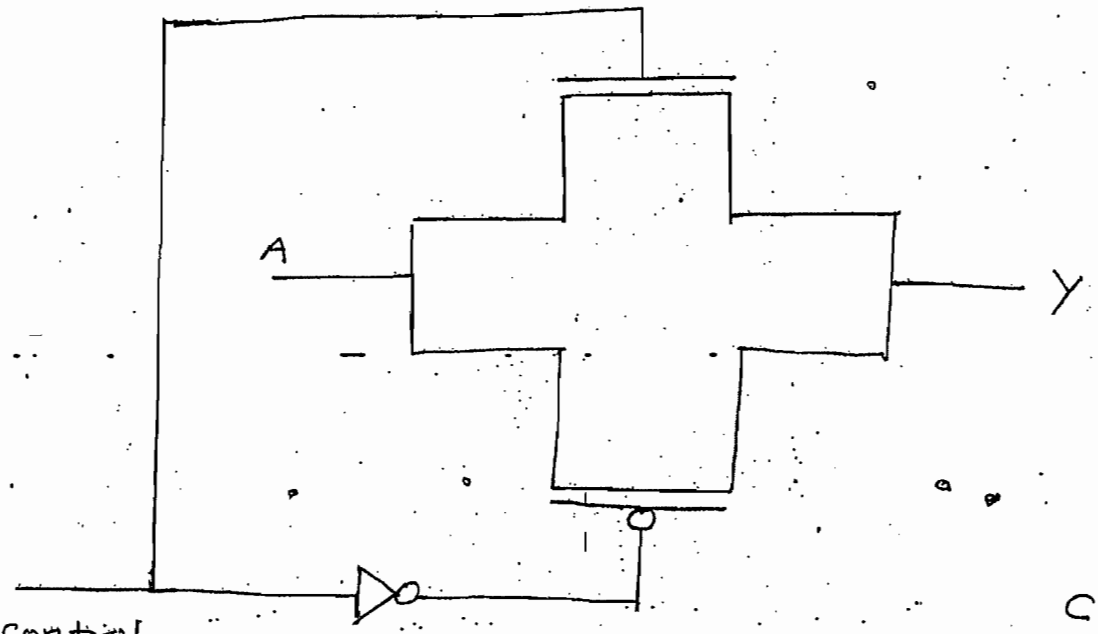


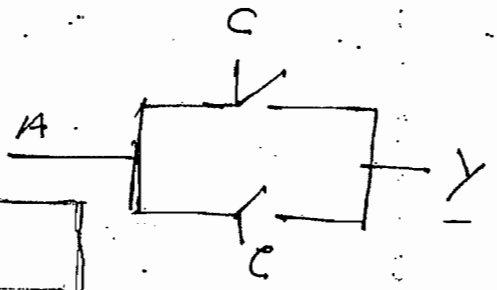
A	B	T_1	T_2	T_3	T_4	y
0	0	ON	ON	OFF	OFF	1
0	1	ON	OFF	OFF	ON	1
1	0	OFF	ON	ON	OFF	1
1	1	OFF	OFF	ON	ON	0

CMOS :-

	Transistor Required
NAND	4
NOR	4
AND	6
OR	6
NOT	2

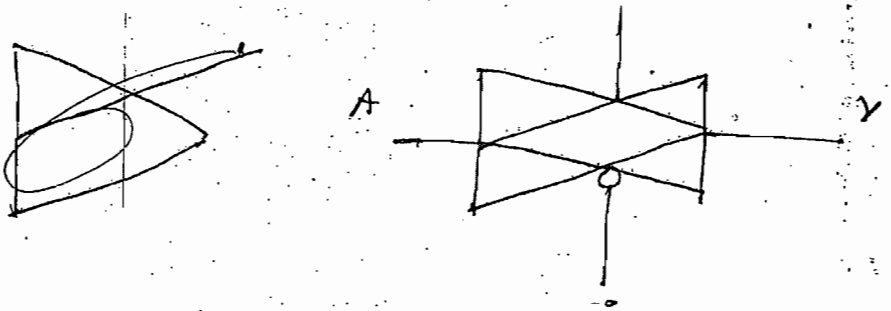
Transmission Gate :-



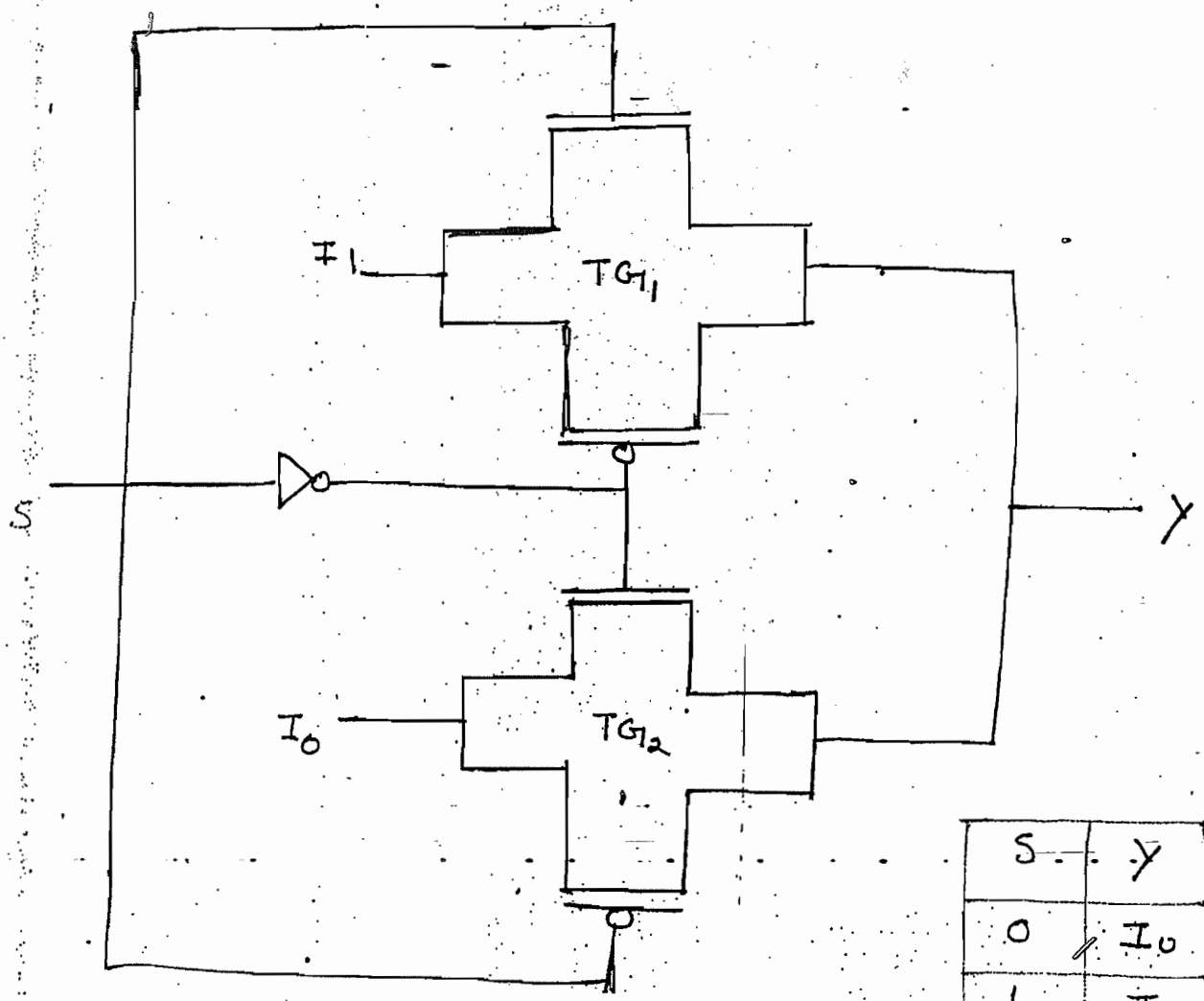


Control	Y
0	High impedance
1	A

Symbol :-



Implement 2x1 MUX using Transmission Gate :-



S	Y
0	I_0
1	I_1

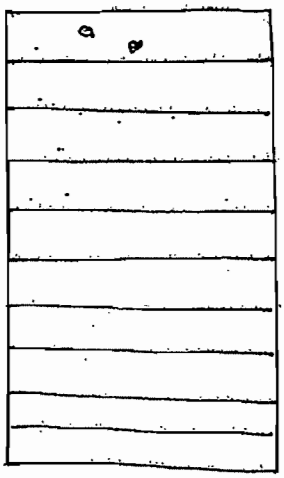
Memories

- Registers
- Primary Memories
- Secondary Memories

Primary Memories:
- RAM
- ROM

	RAM	ROM
→	Read/Write	Read
→	Volatile	Non-volatile
→	temporary storage	Permanent storage
→	User programs / Data	System Programs

Address ⇒



No. of location = 2^n

Memory Capacity = $2^n \times m$

Secondary Memory

- Serial access memory or sequential access memory
- Semi random access

Serial Access Memory

eg:- Magnetic tape

→ Magnetic bubble

→ Ferrite core memory

↓

Distributive Read out

Semi Random Access

eg:- All disk memories

↓

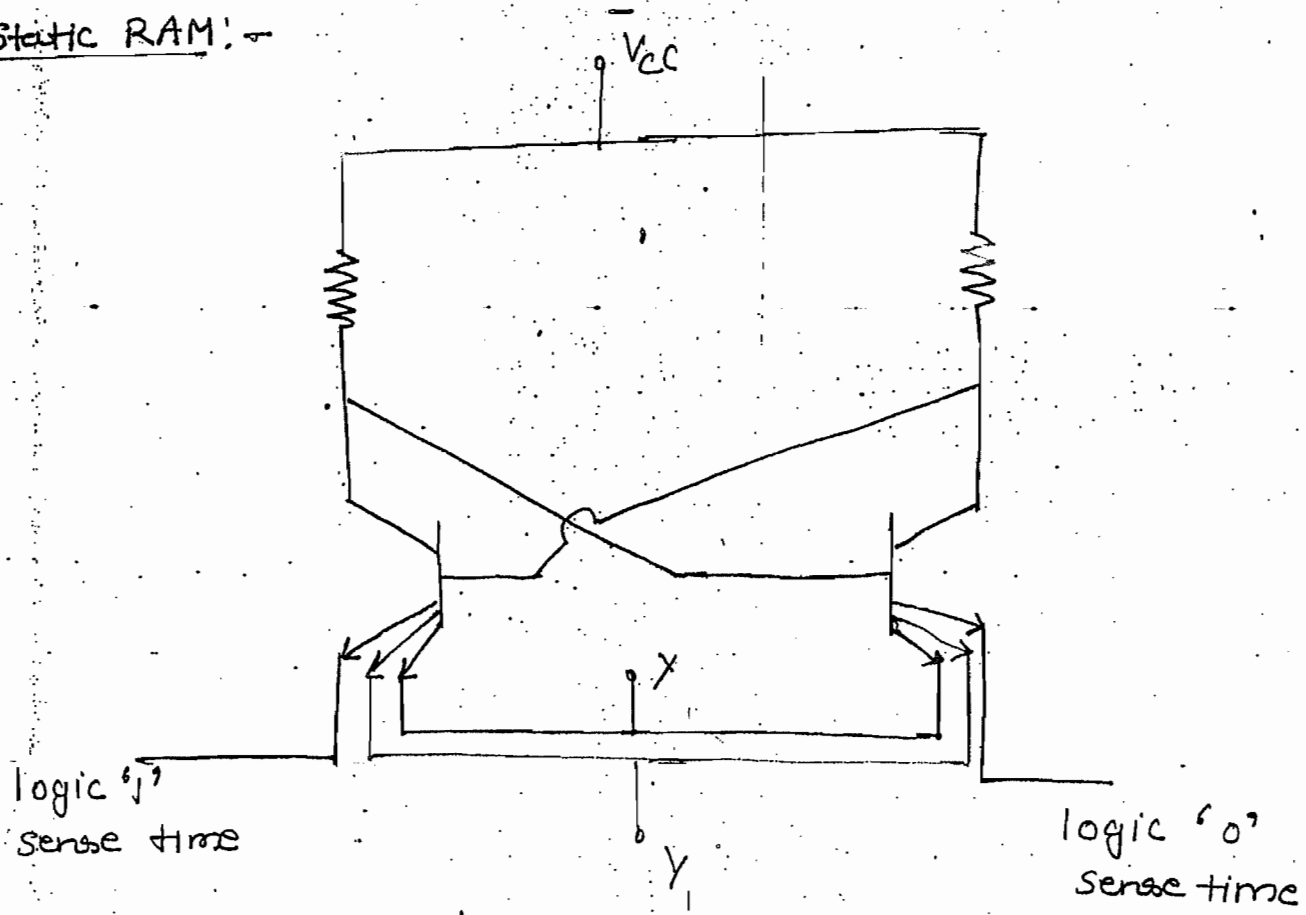
CD, DVD, HD

RAM:-

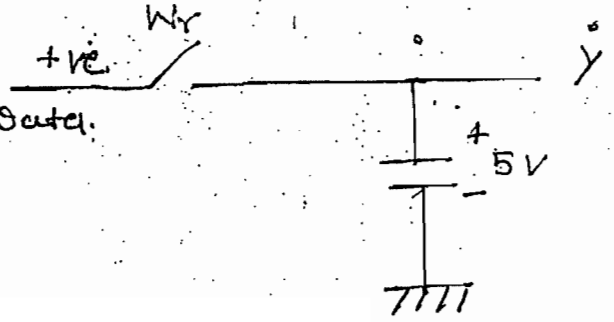
(I) Static RAM

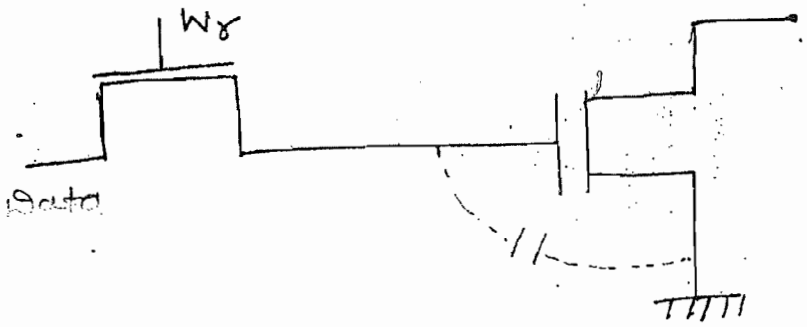
(II) Dynamic RAM

Static RAM:-



Dynamic RAM:-



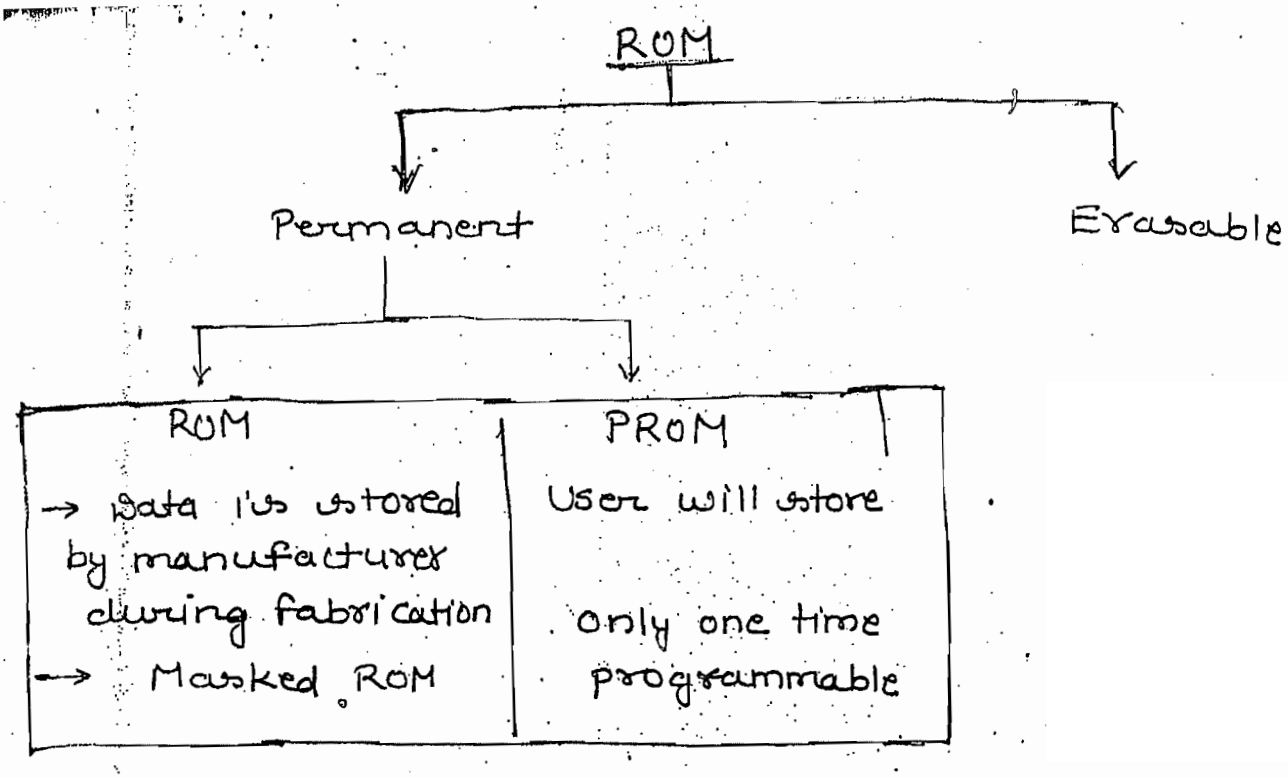


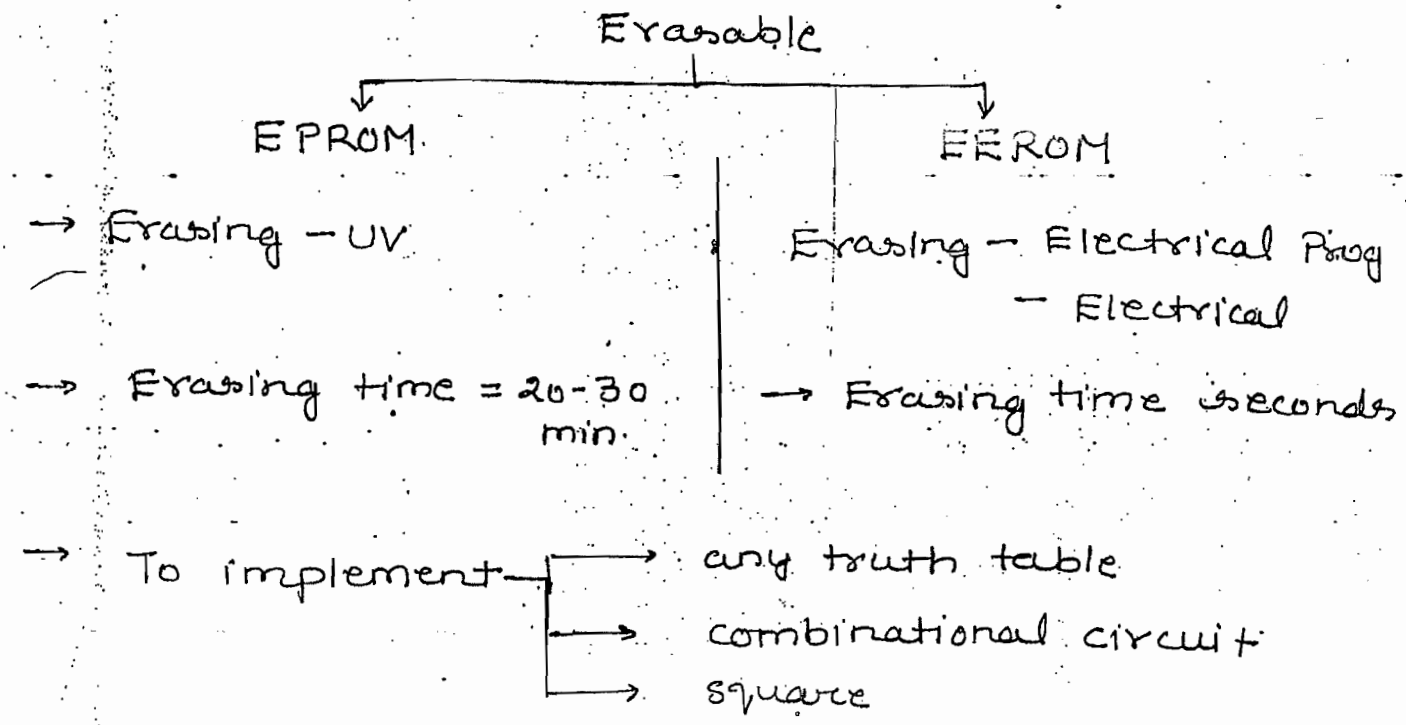
→ Refreshing

	Static RAM	Dynamic RAM
1.	Data is stored in cross coupled structure	Data is stored in MOS Capacitor
2.	BJT, MOSFET	MOSFET
3.	Faster	Slower
4.	Mor dissipate more power	Dissipate less power
5.	For storage of 1-bit, 6 transistor are required	1-bit => 2 transistor
6.	Memory capacity is less	Memory capacity is more
7.	It is used as a cache memory	Used as a primary memory
8.	No refreshing is req.	Refreshing is required (every 2ns)
9.	Volatile	Volatile

ROM:—

→ combinational circuit
→ Decoder - Encoder
AND — OR
→ Permanent programs
→





Ques:- Implement following logic expression using ROM circuit.

$Y_3 (A, B, C) = \Sigma m (1, 2, 4, 7)$

$Y_2 (A, B, C) = \Sigma m (0, 1, 4, 5)$

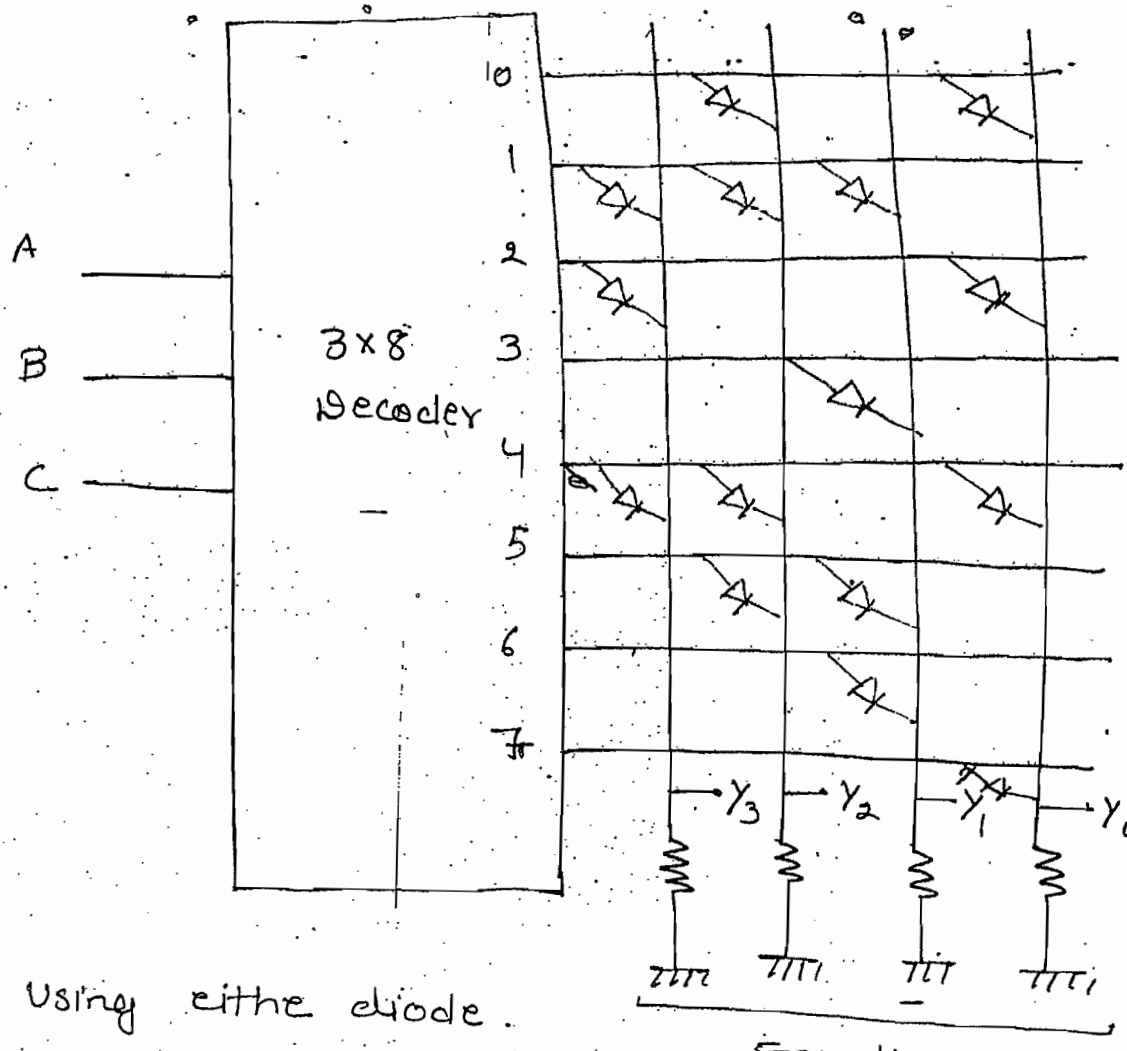
$Y_1 (A, B, C) = \Sigma m (1, 3, 5, 6)$

$Y_0 (A, B, C) = \Sigma m (0, 2, 4, 7)$

Truth Table

Data Bus

A	B	C	Y_3	Y_2	Y_1	Y_0
0	0	0	0	1	0	1
0	0	1	1	1	1	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
1	0	0	1	1	0	1
1	0	1	0	1	1	0
1	1	0	0	0	1	0
1	1	1	1	0	0	1



→ Using either diode or cross marks

Encoding

→	ROM	Fixed AND	Fixed OR
→	PROM	Fixed AND	Prog OR
→	PAL	Prog. AND	Fixed OR
→	PLA	Prog. AND	Prog. OR

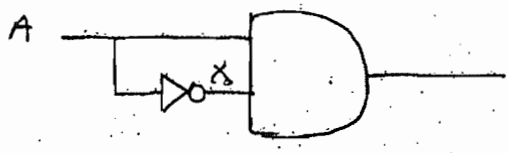
ques:- To implement 2-bits squaring function no. of address line and data lines used in ROM?

soln:-

A	B	Data		Data
Add.				
0	0	0	→	0 0 0 0
0	1	1	→	0 0 0 1
1	0	4	→	0 1 0 0
1	1	9	→	1 0 0 1

Hazards :-

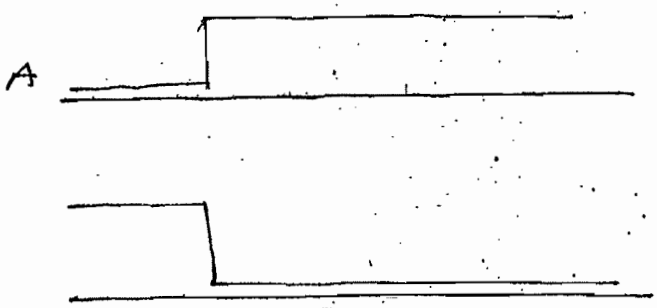
→ Hazards are unwanted change in o/p due to propagation delay.



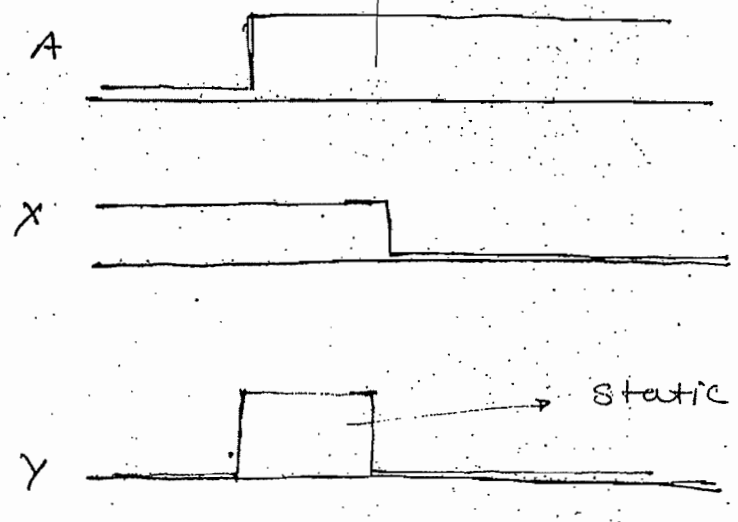
Sometimes it is 1 due to propagation delay

Ques:- Draw o/p waveform for given i/p with no delay in logic gate

Soln:- (i)

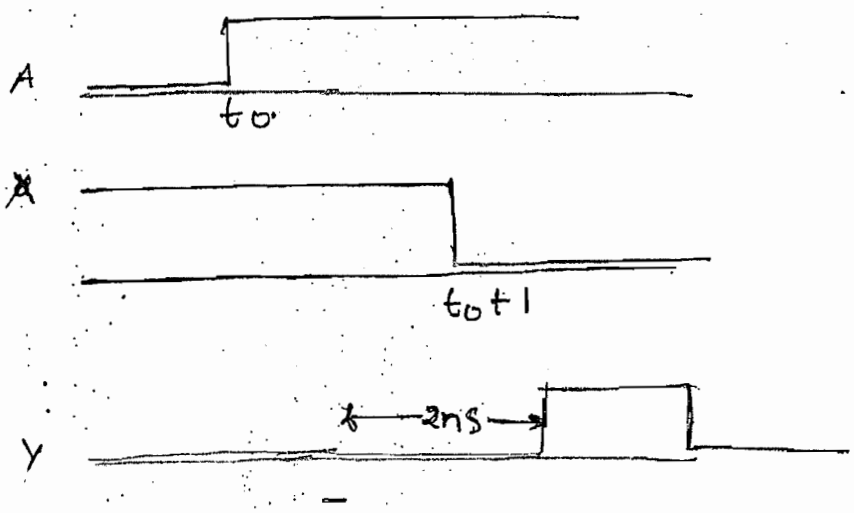


(ii) Consider $t_{pd\,NOT}$ = 1 ns $t_{pd\,AND}$ = 0 ns



(iii) $t_{pd\ NOT} = 1ns$

$t_{pd\ AND} = 2ns$



Hazard

- Static
  - occurs in two level combination
    - Static 1 Hazard
      - two level AND-OR SOP
    - Static 0
      - two level OR-AND
- Dynamic
  - Multi level comb
- Essential
  - Essential Hazard occurs in asynchronous, sequential

→ Static and dynamic hazards can be avoided by