



S. No.	Date	Title	Page No.	Sign / Remarks
		$\text{EX-OR} \rightarrow \text{odd fn} = \text{odd no. of 1's}$		
		$\text{EX-NOR} \rightarrow \text{even fn} = \text{even no. of 1's}$		
		<u>consensus law</u>		
		$(\bar{A}B + \bar{A}\bar{C}) + BC = AB + \bar{A}C$		
		$AB + \bar{A}C + BC (\cancel{\bar{A}B + \bar{A}\bar{C}})$		
		$= AB + \bar{A}C + B\bar{A} + \bar{A}BC (\cancel{B\bar{A} + \bar{A}BC})$		
		$= AB(\bar{C} + 1) + \bar{A}C(\bar{B} + 1) \quad \text{why } \bar{C} + \bar{B} = 1$		
		$= AB + \bar{A}C$		
		$\{ 1^{\text{st}} \text{ Complement} \quad \text{rest} \quad 0^{\text{th}} \}$		
		$- (2^{n-1}) \quad 2^0 + (2^{n-1})$		
		$\{ 0^{\text{th}} \text{ includes} \quad x^0 \}$		
		$x^0 \quad 1^{\text{st}}$		
		$\{ 2^0 \quad \text{is complement graph} \}$		
		$0^{\text{th}} \quad 2^0 \quad (2^{n-1})$		
		$2^0 \quad -2^0 \quad 1^{\text{st}}$		
		$1^{\text{st}} \quad -1^{\text{st}} \quad 0^{\text{th}}$		
		$n=2$		
		$\{ \text{Ex-OR} = \text{Ex-NOR if no. of v/p variables are odd.} \}$		
		$\{ \text{Ex-NOR} = \overline{\text{Ex-OR}} \text{ if no. of v/p variables are even.} \}$		
		$n=2$		

## Signed Binary numbers

- To represent -ve numbers in the binary system,  
 '0' is used for +ve sign.  
 '1' is used for -ve sign.
- The MSB bit is sign bit followed by magnitude bits

Q11 Express in 16-bit signed binary system

- (a) +8    b) -8    c) -165.    d) -1    e) +165

Soln (a)  $(+8)_{16\text{-bit signed binary equivalent}} = \underline{\hspace{10mm}}$

The binary number is 1000.

So the signed 16 bit binary number is

$$+8 = 0000\ 0000\ 0000\ 1000.$$

(b) -8

The signed 16-bit binary number is represented

by

$$-8 = 1000\ 0000\ 0000\ 1000.$$

$$\text{c)} \ (-165) = 1000\ 0000\ 1010\ 0101.$$

$$\text{d)} \ (-1) = 1000\ 0000\ 0000\ 0001.$$

$$\text{e)} \ (+165) = 0000\ 0000\ 1010\ 0101.$$

## Signed Decimal Number

→ The +ve decimal numbers are expressed in sign-mag form

→ The -ve decimal numbers are expressed in 2's complement form.

→ Taking the 2's-complement of a number is same as changing the sign of the given binary no.

→ If we take the 2's-complement twice, we get the original no.

→ The 2's complement of a number is same as its binary equivalent with appropriate number of bits.

Q11. What do the following represent in 2's Complement representation. a) 00011111 b) 11100101 c) 11110111

GATE

Soln a)  $00011111 \rightarrow +ve \text{ Binary number} = +31$

1. NOT

b)  $11100101 \rightarrow -ve \text{ Binary No.} = -27$

$$A = 11100101$$

$$1's \text{ comp. of } A = 00011010$$

$$2's \text{ comp. of } A = \frac{+1}{00011011} = 27$$

c)  $11110111 \rightarrow -ve \text{ Binary no.} = -9$

2. OR

$$\begin{array}{r} 11110111 \\ 00001000 \\ \hline 00001001 = 9 \end{array}$$

Q11.  $A = -24$ ,  $B = +16$ , ① represents A & B in 8 bit  
2's complement form ② Det A+B ③  $A-B=?$

Ans ①  $A = -24$ , sign = -ve, mag = 24.

3. Ans

$$(24)_{10} \leftrightarrow (00011000)_2$$

$$\begin{array}{r} 1's \text{ complement of } 00011000 = 11100111 \\ 2's \text{ comp. of } 00011000 = \frac{+1}{11101000} \end{array}$$

$$\rightarrow B = +16, \text{ sign = +ve, mag = 16}$$

As sign is +ve, 2's comp. of B = Binary equivalent of mag  
 $= (00010000)_2$

$$2's \text{ comp. of } (B=+16) = (00010000)_2$$

b)  $A+B = -24+16$

$$\begin{array}{r} 11101000 \\ 00010000 \\ \hline 11111000 \end{array}$$

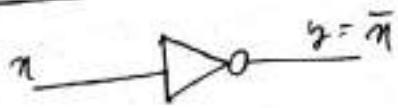
mag = ?

11 0  
21 00

$\therefore \text{ans} = -16$  (mag)

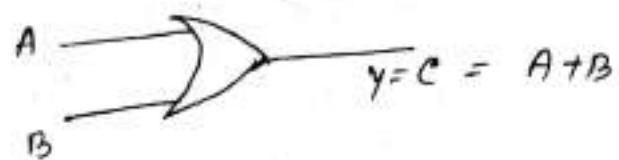
## LOGICAL GATES

### 1. NOT GATE:



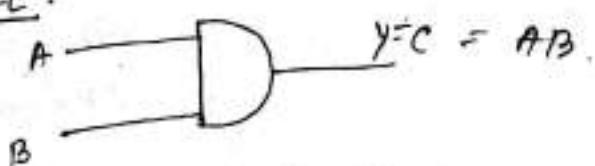
A	y = \bar{A}
0	1
1	0

### 2. OR GATE:



A	B	y = C
0	0	0
0	1	1
1	0	1
1	1	1

### 3. AND GATE:



A	B	y = C
0	0	0
0	1	0
1	0	0
1	1	1

mag = ?

1's 11111000

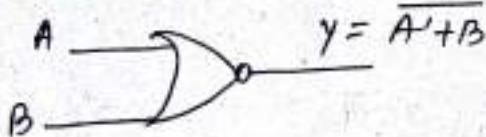
0's 00000111

2's 00001000

$\Rightarrow \text{mag} = 8$

$\Rightarrow A + B = -8$

#### 4. NOR GATE:

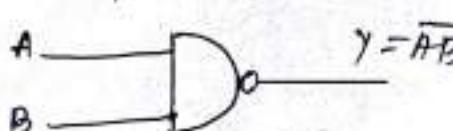


<u>A</u>	<u>B</u>	<u><math>Y = \overline{A} + \overline{B}</math></u>
0	0	1
0	1	0
1	0	0
1	1	0

1.  $Bw$

2.  $B$

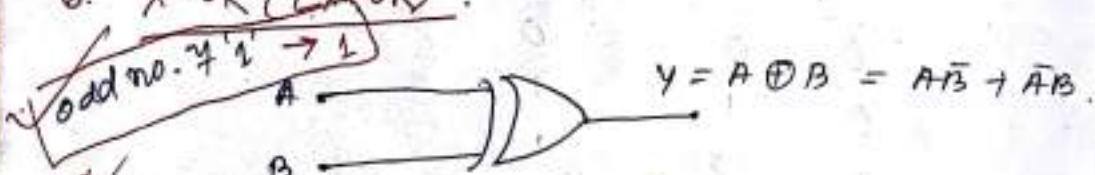
#### 5. NAND GATE:



<u>A</u>	<u>B</u>	<u><math>Y = \overline{AB}</math></u>
0	0	1
0	1	1
1	0	1
1	1	0

3.  $B$

#### 6. X-OR (EX-OR):

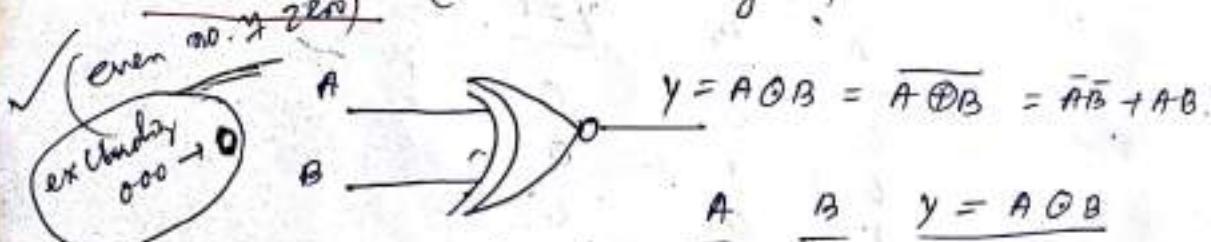


(Odd parity checker)

<u>A</u>	<u>B</u>	<u><math>Y = A \oplus B</math></u>
0	0	0
0	1	1
1	0	1
1	1	0

4.

#### 7. EX-NOR: (Complement gate)



(even parity checker)

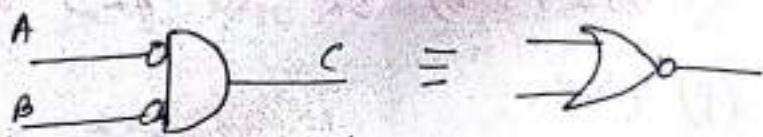
<u>A</u>	<u>B</u>	<u><math>Y = A \text{Q} B</math></u>
0	0	1
0	1	0
1	0	0
1	1	1

NAN

b'c

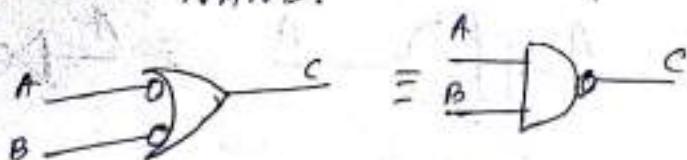
NAn

1. Bubbled AND = NOR:



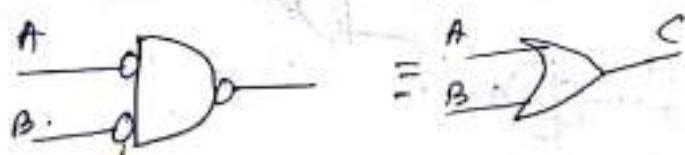
$$\bar{A}\bar{B} = \overline{A+B}$$

2. Bubbled OR = NAND.



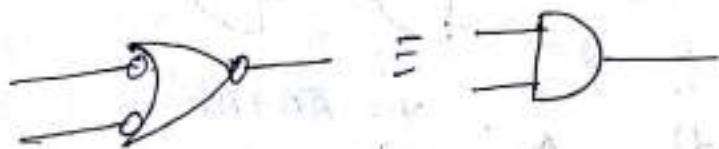
$$\bar{A} + \bar{B} = \overline{AB}$$

3. Bubbled NAND = OR



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

4. Bubbled NOR = AND.



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

5. NAND, NOR  $\rightarrow$  called as universal gate  
because we can design any gate using  
NAND; NOR.

Q Design following GATES Using NAND gate.

a) NOT b) OR c) AND d) EXOR e) EXNOR

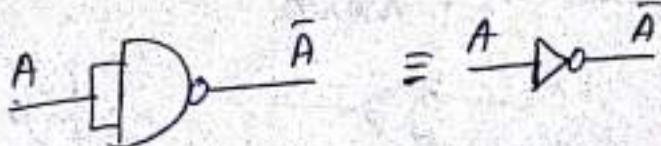
(f) NOR.

B - D  
A)

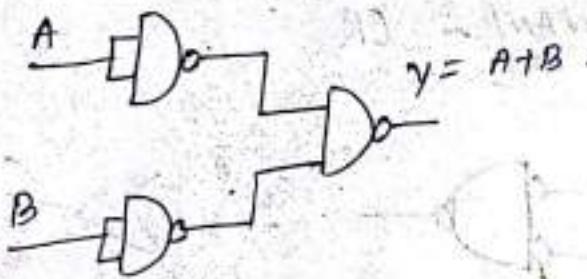
8th 9

8th

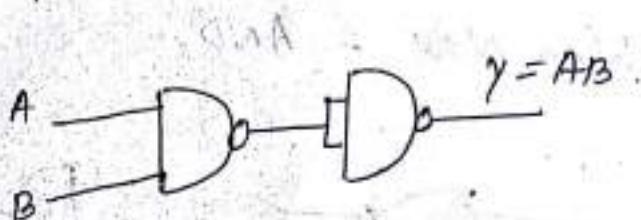
a)



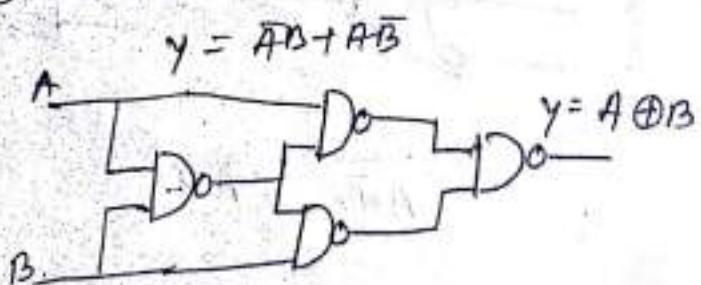
b)  $y = A + B$



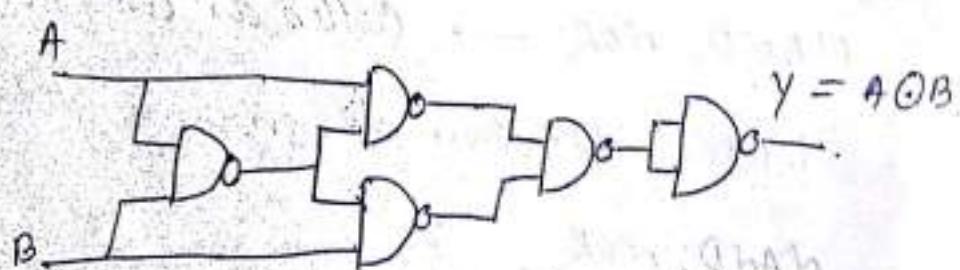
c)  $y = AB$



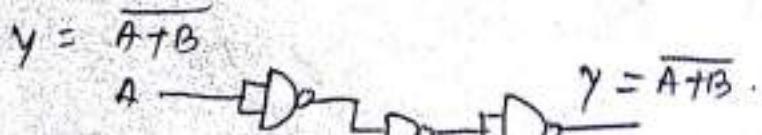
d)



e)

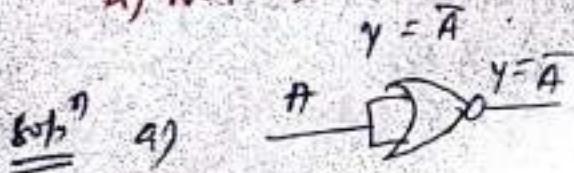


f)

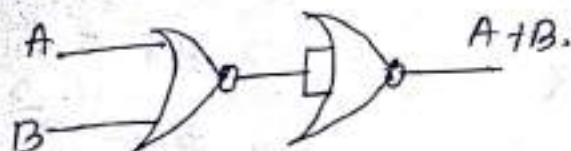


Q- Design the following gates using NOR Gate

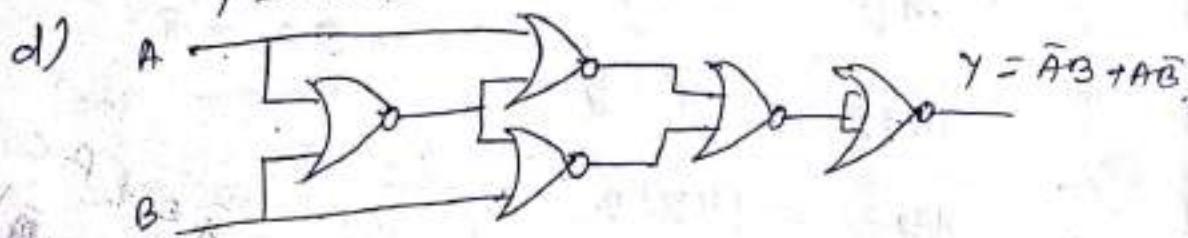
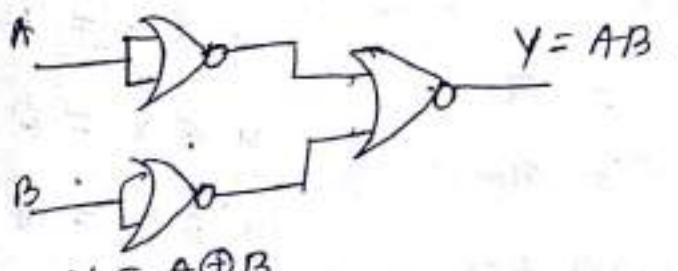
- a) NOT b) OR c) AND d) EX-OR e) EX-NOR (f) NAND.



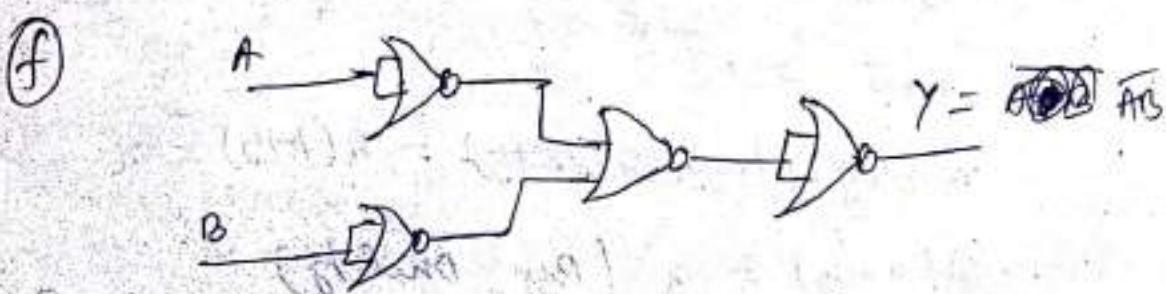
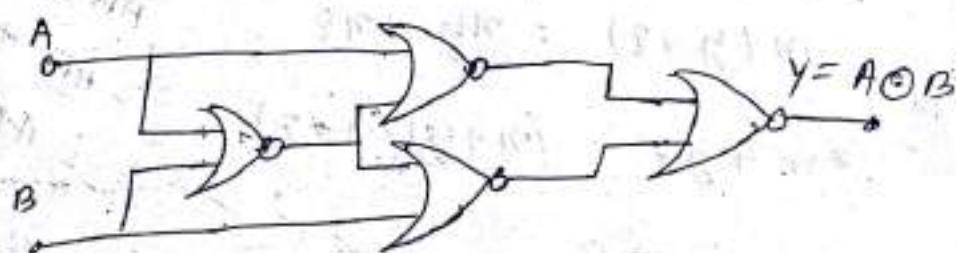
b)  $y = A + B$ .



c)  $y = AB$



e)  $y = A \oplus B = \overline{AB} + AB$ .



- Boolean Fun<sup>n</sup> :-

1 → called as multiplicative Identity.

0 → Additive Identity.

$n \cdot n' +$

Q

### Boolean Properties

$$n \cdot 0 = 0$$

$$n \oplus n = 0$$

Sols

$$n \cdot 1 = n$$

$$n \oplus \bar{n} = 1$$

$$n + 0 = n$$

$$n \oplus 1 = \bar{n}$$

Sols

$$n + 1 = 1$$

$$n \oplus 0 = n$$

$$n + n = n$$

$$n \oplus \bar{n} = 0$$

Sols

$$n \cdot n = n$$

$$n \oplus 1 = n$$

Sols

$$\overline{ny} = \bar{n} + \bar{y}$$

$$n \oplus 0 = \bar{n}$$

Sols

$$\overline{\bar{n}y} = \bar{n} \cdot \bar{y}$$

$$n(yz) = (ny)z$$

$$\begin{aligned} & AB \times \bar{A}C + \bar{A}C \times BC \\ & = AB + \bar{A}C + (\bar{A} + \bar{B})BC \\ & = AB + \bar{A}C + ABC + \bar{B}C \end{aligned}$$

Sols

$$n + (y+z) = (n+y) + z$$

$$\begin{aligned} & AB + \bar{A}C + (\bar{A} + \bar{B})BC \\ & = AB + \bar{A}C + ABC + \bar{B}C \end{aligned}$$

Sols

$$n(y+z) = ny + nz$$

$$\begin{aligned} & AB + \bar{A}C + (\bar{A} + \bar{B})BC \\ & = AB + \bar{A}C + ABC + \bar{B}C \end{aligned}$$

Sols

$$n + yz = (n+y)(n+z)$$

$$\begin{aligned} & AB + \bar{A}C + (\bar{A} + \bar{B})BC \\ & = AB + \bar{A}C + ABC + \bar{B}C \end{aligned}$$

Sols

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

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

$$n + ny = (\cancel{n} + \cancel{n})(\cancel{n} + \cancel{y}) = n(1+y) = n$$

$$n(n+y) = n \quad (\text{By Duality})$$

$$n + \bar{n}y = (n + n')(n + y) = n + y$$

Sols

$$n(n'y) = ny \quad (\text{By Duality})$$

Q  $f = n\bar{y} + \bar{n}\bar{y} + ny$ , find the min. no. of literals.

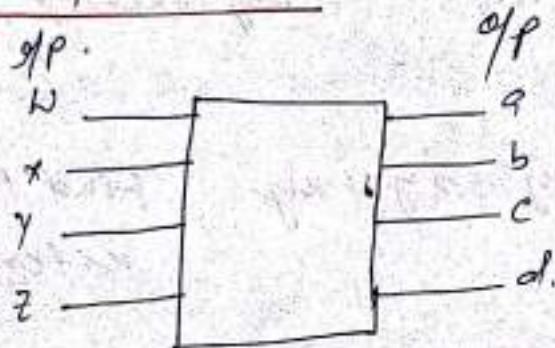
Sol<sup>n</sup>  $f = n\bar{y} + \bar{n}\bar{y} + ny \rightarrow 4 \text{ literals}$   
 $= ny + y$   
 $= (n+y)(y+\bar{y})$   
 $f = ny \rightarrow \text{no. of literals} = 2$ .

→ To reduce the complexity of hardware, for less propagation delay i.e. for faster, for less fault, for less cost, we have to minimize the ent. (i.e. by reducing the literals).

Q  $f = n\bar{y} + \bar{n}\bar{y} + ny + \bar{n}\bar{y}$ , find the min. no. of literals.

Sol<sup>n</sup>  $f = n\bar{y} + \bar{n}\bar{y} + ny + \bar{n}\bar{y} \rightarrow \text{no. of literals} = 4$   
 $= 1 \rightarrow \text{no. of literals} = 0$ .

## Design procedure



$\oplus$  Fin

Step-1 1st design 1st T-T

w	x	y	z	a	b	c	d
0	0	0	0				
1	1	1	1				

→ This table is known as functional table.

Step-2

Now translate the table into Boolean fun<sup>n</sup>.

$$a(w, x, y, z) = \checkmark$$

$$b(w, x, y, z) = \checkmark$$

$$c(w, x, y, z) = \checkmark$$

$$d(w, x, y, z) = \checkmark$$

: Duality Theorem:

Every algebraic expression reducible from Theorem of Boolean Algebra will remains valid even if the operators & identity elements are interchanged, i.e.  $\begin{cases} \cdot & + \\ 1 & 0 \end{cases}$  → To get the dual,

Q Find the duality of the given expression.

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

$$\Rightarrow x \cdot 0 = 0$$

$$2. x + x = x$$

$$x \cdot x = x$$

$$3. x + yz = (x+y)(x+z)$$

$$\Rightarrow x \cdot (y+z) = xy + xz$$

$$4. \overline{xy} = \bar{x} + \bar{y}$$

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

$$5. f(x, y, z) = xy\bar{z} + \bar{x}yz + x\bar{y}z$$

$$\text{Dual of } f(x, y, z) = (x+y+\bar{z})(\bar{x}+y+z)(x+\bar{y}+z)$$

: Complement Theorem :

To get the Complement of a given function  
1st find the dual of the func' & then comple  
each literal.

Ex-1. Find the complement of the given func'.

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

$$\text{Dual of } f(x, y, z) = (x+y+\bar{z})(\bar{x}+y+z)(x+\bar{y}+z)$$

$$\overline{f(x, y, z)} = (\bar{x}+y)(x+\bar{y})(\bar{x}+\bar{y})$$

Ex-2

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

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

Note the total no. of Boolean fun<sup>n</sup>/Expression

1. ~~by~~ the total no. of Boolean fun<sup>n</sup>/Expression  
that can be formed by 'n' variable  
 $= (2)^n$ .

2. Total no. of cells in a n-variable K-map  
 $= 2^n$  b/cse total no. of minterm  
or maxterm  $= 2^n$ .

Min term (Standard Product term):

Def<sup>n</sup> It is the product of all the binary variables either in complemented or uncomplemented form s.t. off will always

$$\begin{array}{cc} m & y \\ \hline 0 & 0 \end{array} \quad \text{Minterm (m)} \quad \bar{m}\bar{y} = m_0$$

1 → Active

$$\begin{array}{cc} 0 & 1 \end{array} \quad \bar{m}y = m_1$$

$$\begin{array}{cc} 1 & 0 \end{array} \quad m\bar{y} = m_2$$

$$\begin{array}{cc} 1 & 1 \end{array} \quad my = m_3$$

Max term (Standard Sum term):-

Def<sup>n</sup> It is the sum of all the variables either incomplemented or uncomplemented form so the off will be zero '0'

PO:

Note

<u>m</u>	<u>y</u>	<u>maxterm(M)</u>	$0 \rightarrow \text{not } y$
0	0	$\bar{x} + \bar{y} = M_0$	
0	1	$\bar{x} + y = M_1$	
1	0	$x + \bar{y} = M_2$	
1	1	$x + y = M_3$	

Sum of Product expression (SOP) :-

$$\text{SOP: } f(x, y) = x'y + xy'$$

$$f(x, y) = \sum m(1, 3)$$

$$\text{Ex } f(x, y) = \sum m(0, 1, 2, 3) \\ = 1 \text{ (off P)}$$

$$f(x, y, z) = \sum m(0, 1, 2, 3, 4, 5, 6, 7) \\ = 1 \text{ (off P)}$$

Product of Sum expression (POS):

$$\text{POS: } f(x, y) = (x + y)(x' + y') \\ \downarrow \qquad \qquad \qquad f(x, y) = (x' + y')(x + y) \\ = \prod m(0, 1)$$

$$\text{Ex } f(x, y) = \prod m(0, 3).$$

$$f(x, y, z) = \prod m(0, 1, 2, 3) = 0 \text{ (Ans)}$$

Note No. of minterms + No. of max terms = No. of 1's in the truth table

Standard Sum of Product (SSOP) = Canonical form

Q. Det. Standard SOP form for the given fcn

$$f(n, y, z) = n'y + z' + nyz$$

Soln

$$f(n, y, z) = \begin{array}{c} \text{not in SSOP} \\ n'y \\ \text{not in SSOP} \\ z' \\ \text{SSOP} \\ nyz \end{array}$$

$$= (n' + y \\ (n +$$

$$= n'y(z + \bar{z}) + \bar{z}(n + \bar{n})(y + \bar{y}) \\ + ny\bar{z}$$

Q/H

fin

Given

$$= n'yz + ny\bar{z}' + (n\bar{z}' + n'\bar{z}')/y \\ + ny\bar{z}$$

Soln

$$= n'yz + ny\bar{z}' + \cancel{n\bar{z}' + n'\bar{z}'} + n'y\bar{z}' + ny\bar{z}' + n'y\bar{z} \\ + ny\bar{z}$$

fin

$$= n'yz + ny\bar{z}' + n'y\bar{z}' + ny\bar{z}' + n'y\bar{z}' + ny\bar{z}$$

Q. Det. POS's. form for the given fcn

$$f(n, y, z) = n'(y' + z)$$

Soln

$$f(n, y, z) = n'(y' + z)$$

$$= (n' + yz + z\bar{z}) / (n + y' + z)$$

$$= \frac{(x' + y + z)(x' + y + z')(x' + \underline{y'} + z)(x' + y' + z')}{(x + y' + z)(x' + \underline{y'} + z)}$$

$$= \frac{(x' + y + z)(x' + y + z')(x' + y' + z')}{(x' + y' + z)} (x + y' + z)$$

Q7, find the Product of sum's form for 1<sup>h</sup>

Given ~~fun<sup>n</sup>~~  $f(x, y, z) = \sum_{\substack{o \\ 0 \\ 0}}^o \sum_{\substack{o \\ o \\ 1}}^o \sum_{\substack{o \\ 1 \\ 0 \\ 1}}^o \sum_{\substack{1 \\ 3}}^o$   
 $+ \sum_{\substack{1 \\ 1 \\ 1}}^o \sum_{\substack{1 \\ 0 \\ 1}}^o \sum_{\substack{1 \\ 0 \\ 0}}^o \sum_{\substack{1 \\ 3}}^o$

~~fun<sup>n</sup>~~ To get 1<sup>h</sup> result, complement the given

fun<sup>n</sup> two times  
 $\left[ [f(x, y, z)]' \right]'$

$$= \frac{(x + y + z)(x + y + z')(x + y' + z')(x' + y + z)}{(x' + y + z')(x' + y + z)}$$

$f(x, y, z) = \Sigma_m(0, 1, 3, 4, 5, 7)$

$f(x, y, z) = \Pi M(2, 6)$

$$= (x + y' + z)(x' + y' + z) \text{ (Ans)}$$

Q11 Find the POS form for the given fns<sup>3</sup>

$$f(m, y) = m'y + my'$$

Sol<sup>3</sup>  $f(m, y) = m'y + my' \rightarrow \text{sop.}$

$$\begin{aligned} [(f(m, y))']' &= \overline{(m+y') (m'+y)} \\ &= \overline{my} \quad \overline{m'y'} \\ &= (m'+y') (m+y) \end{aligned}$$

OR  $f(m, y) = \sum m(1, 2)$   
=  $\emptyset \Pi M(0, 3)$   
=  $(m+y) (m'+y')$

Q11 What is the equivalent other Canonical

form for the eq<sup>3</sup> given below

$$f(m, y, z) = \sum m(0, 1, 2, 3, 4, 5, 6, 7) = 1$$

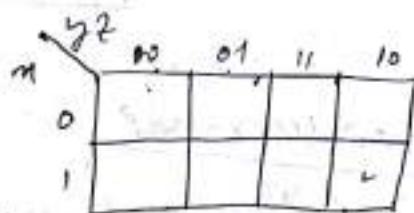
Sol<sup>3</sup> Does not exist, bcsce its op is Product  
of sets of ps.

$$\text{Also for } F(m, y, z) = \Pi M(0, 1, 2, 4, 5, 6)$$

Other canonical form does not exist.

## K-map

- K-map is the modified form of a truth table &  $n$ -variable K-map consists of  $2^n$  shells, representing all possible combination of these variables.
- Each shell of a K-map represents either a minterm or max-term.
- Grey code sequence is assign to shells to maintain adjacent property, b/cse if two shell's differs in just one variable value



$$ny^2 + ny\bar{2} = ny$$

④ For 2 variable K-map

Group of 4 shells	variable No. of literal eliminated	No. of literal Present in the result
—	2	0
2 shells	1	1, 2.
1 shell	0	—

for 3-variable K-map

8 shells	3	0
4 shells	2	1
2 shells	1	2
1 shell	0	3

Note

Sum of no. of eliminated of no. of literals present in the resulting system is always  $n - m$ .

→ In an  $n$  variable k-map, combining a group of '8' adjacent ~~cells~~ shells containing '1' as a single group will results a term of  $n-3$  literals by eliminating 3 variables.

		2-variable k-map	
		0	1
y	0	0	1
	1	2	3

		3-variable k-map				
		yz	00	01	11	10
w	0	0	1	3	2	
	1	4	5	7	6	

4-variable k-map

		yz	00	01	11	10
		wz	0	1	3	2
w	0	4	5	7	6	
	1	12	13	15	14	

		yz	00	01	11	10
		wz	00	01	11	10
w	0	1	1			
	1	1	1			

$$\rightarrow n = 4 = \text{no. of variables}$$

$$\rightarrow 4-\text{cells} = 2^2 \times 2^2$$

$$m = 2$$

$$\rightarrow \text{no. of literals}$$

$$= n - m$$

→ Each subcubes can be expressed by a product of containing  $(n-m)$  literals, where

$n = \text{no. of variables of the Boolean func}$

$m = \text{the collection of } 2^m \text{ cells}$ .

Ex

Consider 5 variable K-map.

$$\Rightarrow n = 5.$$

let's Considering 8 shells  $= 2^m = 2^3$ .

$$\Rightarrow m = 3.$$

No. of literals  $= n - m = 5 - 3 = 2$  ( ~~odd~~ <sup>present</sup> )

Q1) Minimize the Boolean func  $f(u, v, y, z)$

$$= \sum m(0, 4, 5, 7, 8, 9, 13, 15)$$

Soln

	yz	uv	00	01	11	10
vn	00	1				
01	01	1	1	1	1	1
11	11	1	1	1	1	1
10	10	1	1			

$$f(u, v, y, z) = u'v'z' + uv'y' + v'z$$

Q11 Minimize the function  
 w.r.t.  $w, n, y, z$   $f(w, n, y, z) = \Sigma^m(1, 5, 6, 7, 11, 12, 13, 15)$

$w'z'$	00	01	11	10
00	0	1	1	1
01	1	0	1	1
11	1	1	0	1
10	0	0	1	0

$$f_1(w) = \overline{f_1(w)}$$

Q11 Minimize the function  
 w.r.t.  $w, n, y, z$   $f_1(w) = \Sigma^m(1, 5, 6, 7, 11, 12, 13, 15)$

Q11 Minimize the function

$$f_1(w, n, y, z) = \Pi M(1, 5, 6, 7, 11, 12, 13, 15)$$

$$f_1(w, n, y, z) = \overline{f_1(w, n, y, z)}$$

$$= (w + y + z') (w' + n' + y) (w' + y' + z) (w + n' + y')$$

$$OR f_1(w, n, y, z) = \Sigma^m(0, 2, 3, 4, 8, 9, 10, 14)$$

$w'z'$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	0

$w'z'$	00	01	11	10
00	0	0	1	1
01	1	0	0	0
11	0	0	0	0
10	1	1	0	0

$$(w + n + y) (w + y + z) (w + \bar{n} + \bar{y}) (w + \bar{n} + \bar{y} + z)$$

$$f_1(w, n, y, z) = w'y'z' + w'n'y + w'n'y + w'nyz$$

$$f_1(u, v, y, z) = u'y'z' + u'v'y + u'v'z + v'yz \rightarrow \text{SOP}$$

After Booleanification  $f_1(u, v, y, z)$ , we get this exp.

$$\overline{f_1(u, v, y, z)} = (u+y+z') (u'+v'+y) (u'+v'+z')$$

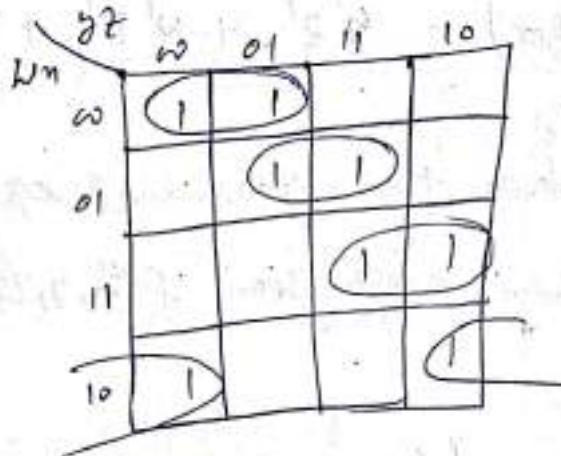
$(u+v'+z')$  → POS.

Q11 Obtain the minimal expression for given fns

~~Q11 P & S~~

$$f(u, v, y, z) = \Sigma(0, 1, 5, 7, 8, 10, 14, 15)$$

Soln



$$f(u, v, y, z) = u'y'z' + u'v'z + u'v'y + v'yz'$$

Q11 P & S obtain the minimal expression for

~~Q11 P & S~~ given fns  $f_1(u, v, y, z)' = \prod M(0, 1, 5, 7, 8, 10, 14, 15)$

Soln

$$f_1(u, v, y, z) = \overline{f(u, v, y, z)}$$

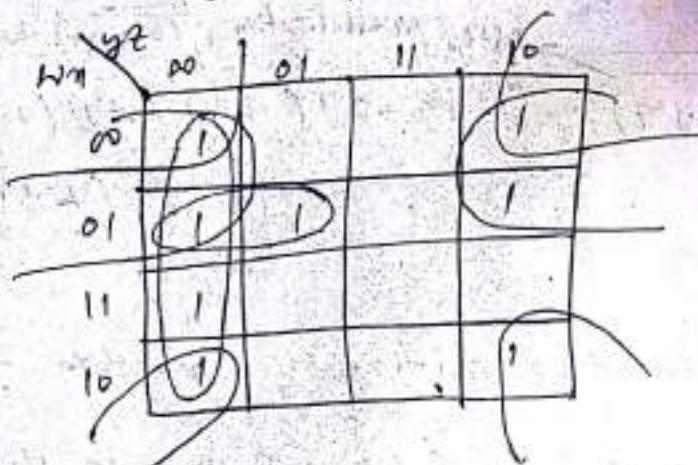
$$= (u+y+z) (u+v+y')$$

$$= (u'+v'+y) (u'+v+z)$$

(Ans)

Q11 Minimize the function  $f(w, y, z, t) = \Sigma(0, 2, 4, 5, 6, 8, 10)$

Sol:



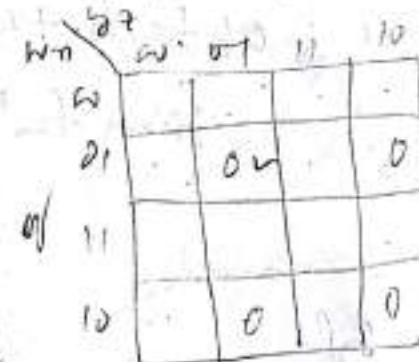
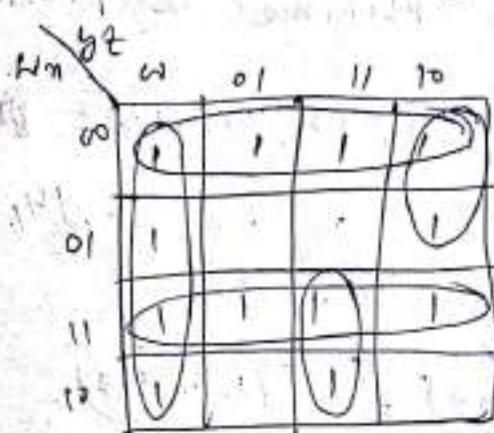
$$f(w, y, z, t) = y'z' + w'y' + w'z' + w'yz'$$

Q11 Obtain the minimal expression for the given expression  $f(w, y, z, t) = \Pi M(5, 6, 9, 10)$

Sol:

$$f(w, y, z, t) = \Pi M(5, 6, 9, 10)$$

$$= \Sigma m(0, 1, 2, 3, 4, 7, 8, 11, 12, 13, 15)$$



$$f(w, y, z, t) = y'z' + w'y + w'w + w'yz' + w'yz$$

$$(w'w + y + z')(w' + w'z')$$

OR  $f_1(u, n, y, z) = \Sigma m(5, 6, 9, 10)$

$u'$	$y'$	$w$	$n$	$z'$	$z$
$u$	$\cancel{w}$	$w$	$n$	$z'$	$z$
$n'$	$\cancel{y}$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$
$y'$	$w$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$
$z'$	$\cancel{w}$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$
$z$	$w$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$

$$f_1(u, n, y, z) = u'ny'z + u'nyz' + un'yz \\ + un'yz'$$

$\oplus$  4 terms of each term contains 4 literals  $\therefore$  Total terms  $= 4 \times 4 = 16$ .

$$f(u, n, y, z) = f_1(u, n, y, z)$$

$$= (u + n' + y + z') (u + n' + y' + z) \\ (u' + n + y + z') (u' + n + y' + z)$$

(Ans)

Don't care (X)

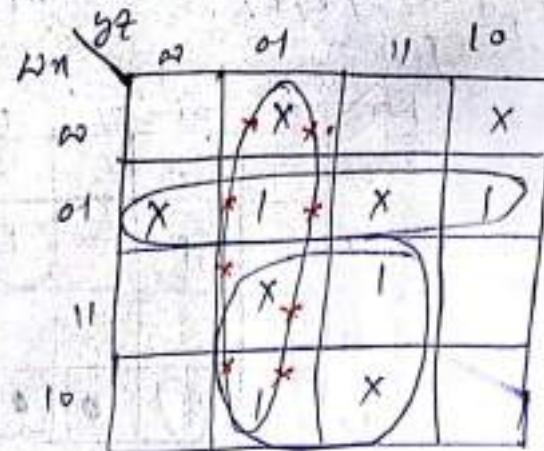
$\rightarrow$  Don't care can help to cover all the 1's, but it is not necessary to cover all the X.

Q11 Simplify the given K-map:

$u'$	$y'$	$w$	$n'$	$z'$	$z$
$u$	$\cancel{y}$	$w$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$
$n'$	$\cancel{y}$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$
$y'$	$w$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$
$z'$	$\cancel{w}$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$
$z$	$w$	$\cancel{w}$	$\cancel{n}$	$\cancel{z'}$	$\cancel{z}$

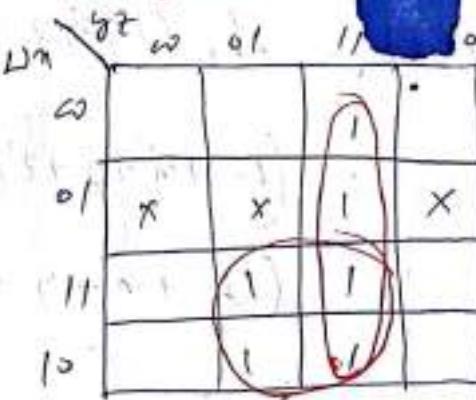
$$f(u, n, y, z) = n'z + n'z' = n \oplus z$$

Q11 Simplify the K-map.



$$f(w, x, y, z) = w'x + wz.$$

Q11 Minimize the given K-map.



$$f(w, x, y, z) = wz + yz.$$

Q11  $f(w, x, y, z) = \overline{\text{PI}} M(1, 5, 6, 7, 11, 12, 13, 15)$

	yz	wz	wy	wx	
	00				.
yz		0			
wz		0	0	D	
wy	0	0	0		
wx			D		

$$f(w, x, y, z) = (w' + x' + y)(w + y + z')(w + y + z)$$

$$\begin{aligned} \therefore f(w, x, y, z) &= \overline{\text{PI}} M(1, 5, 6, 7, 11, 12, 13, 15) = \overline{\text{PI}} M(1, 5, 6, 7, 11, 12, 13, 15) \\ &= \overline{wx'y' + w'y'z' + w'yz} \end{aligned}$$

$$\begin{aligned} \therefore f(w, x, y, z) &= \overline{\text{PI}} M(0, 2, 3, 4, 8, 9, 10, 14) \\ &\quad + \overline{w'yz} \\ &= (w + x' + y')(w + y + z') \\ &\quad + (w' + y' + z') \\ &\quad + \overline{w'yz} \end{aligned}$$

④ Simplify & finally get pos. form.  $(w' + y' + z') (w + x' + y')$

## Prime implicant

The prime implicant of a function  $f$  is a product term which is covered by  $f$ 's which will not further simplified.

$u_m \setminus y_t$	00	01	11	10
00	x	x		
01			1	
11		1	1	
10	1	1	1	1

All PIs are SPIs

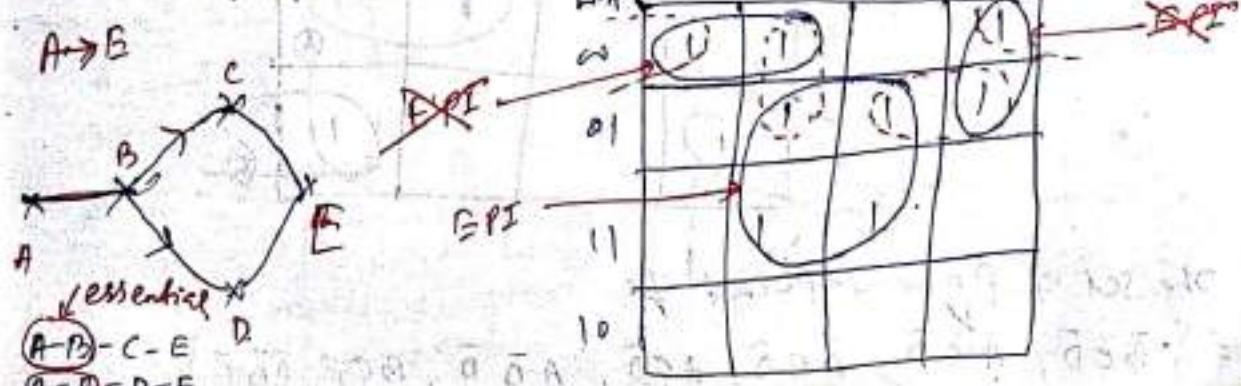
$$f(u, m, y, t) = \underline{u} \underline{y} + \underline{y} \underline{t}$$

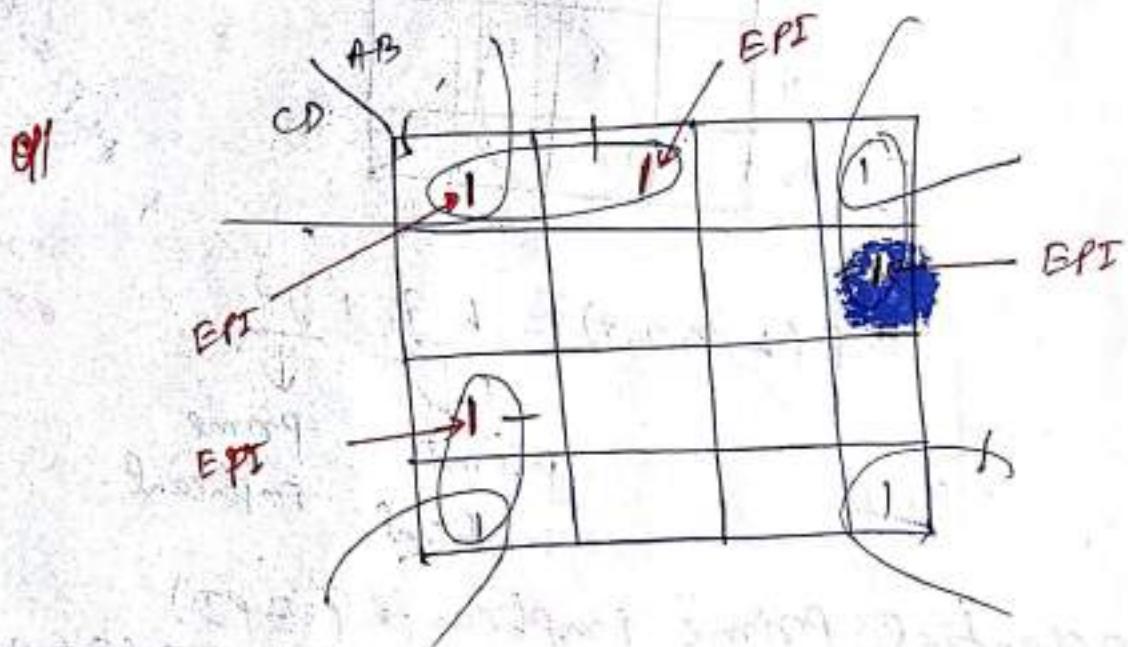
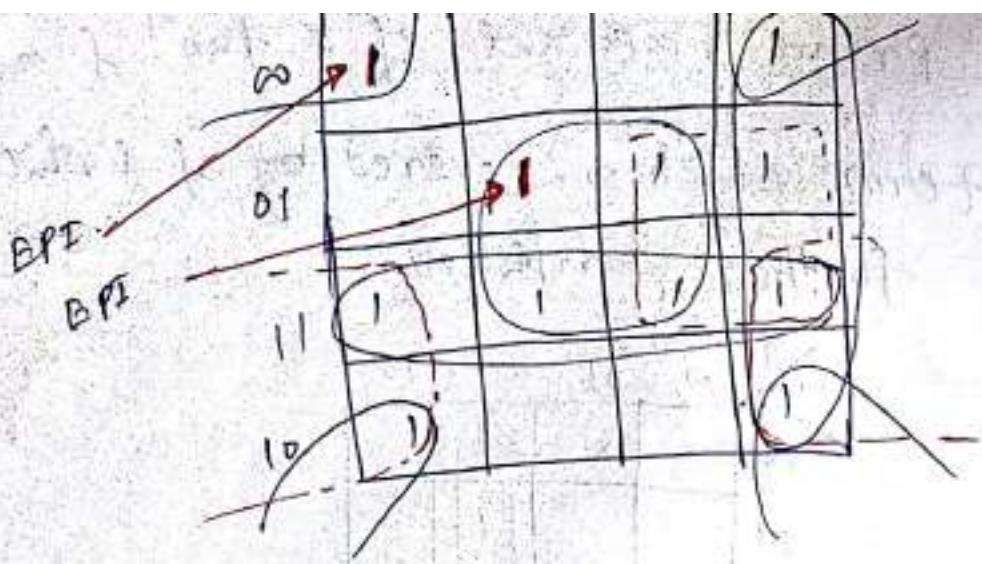
↓                    ↓  
prime implicant      prime implicant

## Essential prime implicant (EPI)

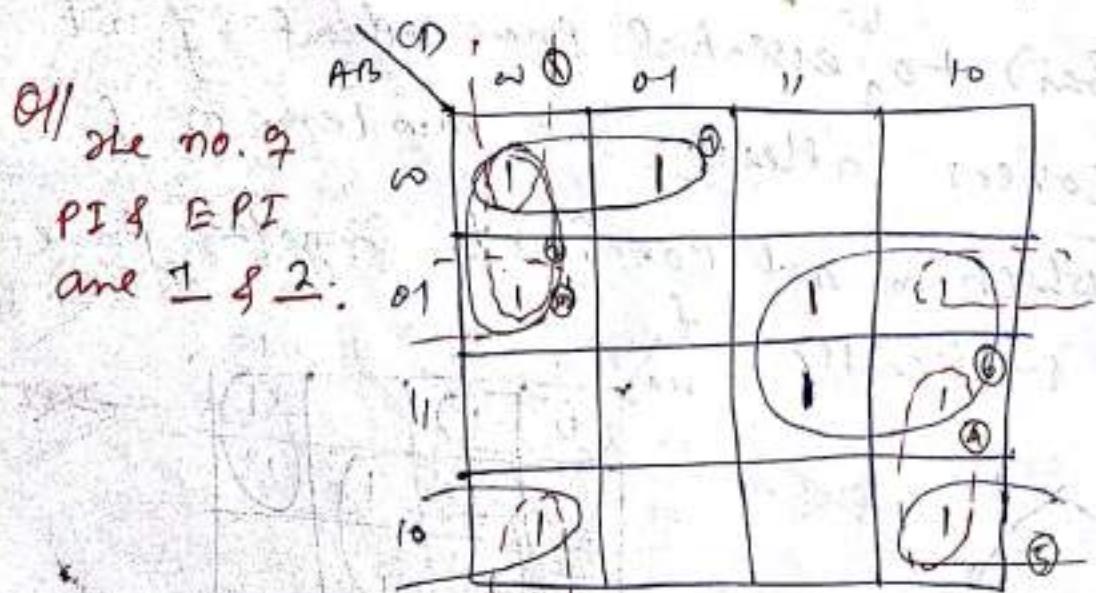
A prime implicant  $P_i$  in said to be essential prime implicant if it covers atleast one min term of  $f$

which is not covered by another prime implicants.





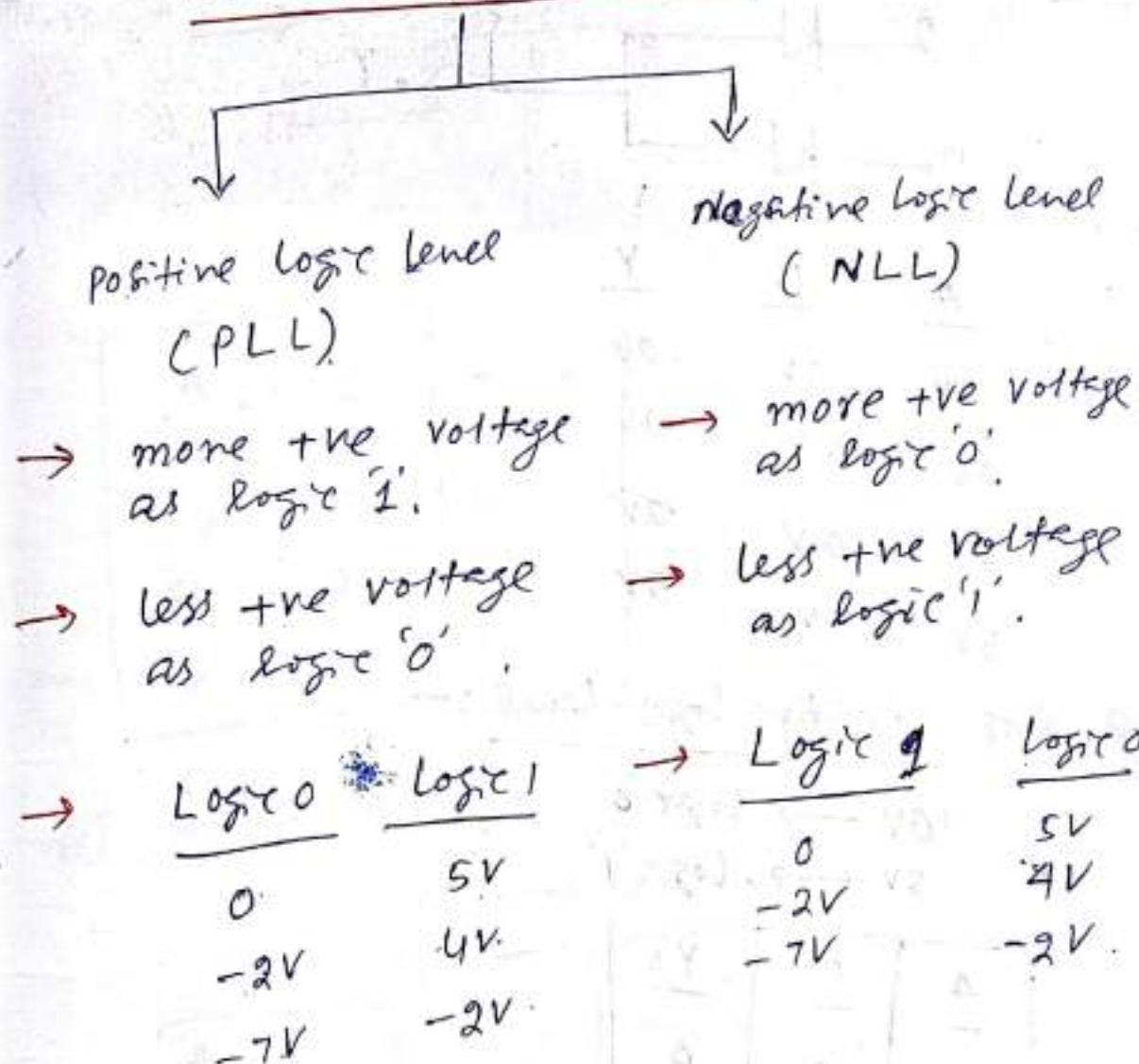
Q1 No. of essential Prime implicants = 4.



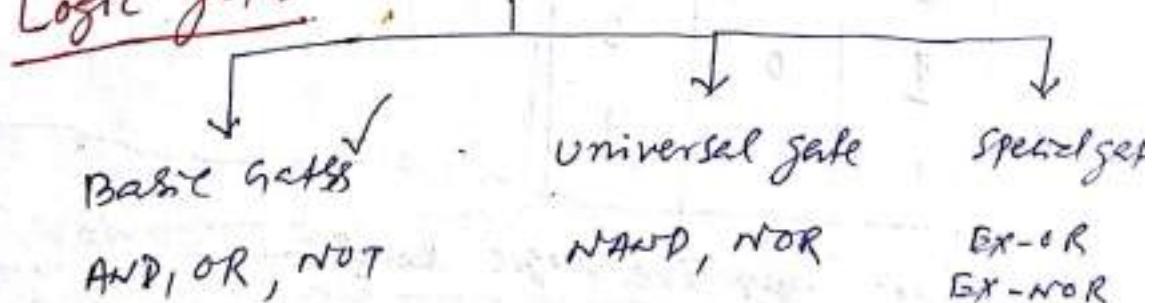
The set of prime implicants be

$$= \{\bar{B}\bar{C}\bar{D}, \bar{A}\bar{C}\bar{D}, \bar{A}\bar{B}\bar{D}, A\bar{C}\bar{D}, A\bar{D}\bar{D}, BC, \bar{A}\bar{B}\bar{C}\}$$

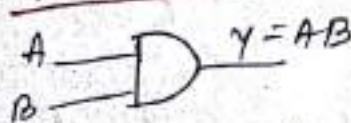
## Classification of Digital cat.



## Logic gates



### AND Gate

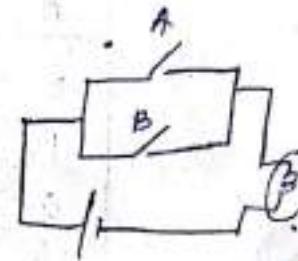


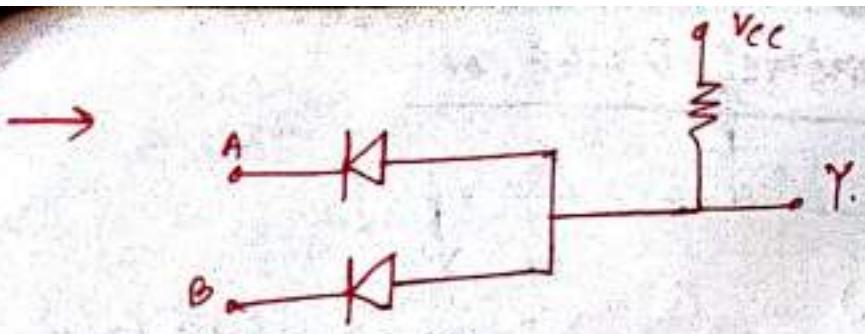
A	B	Y
0	0	0
0	1	0
1	0	0

### OR Gate



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





<u>A</u>	<u>B</u>	<u>Y</u>
0V	0V	0V
0V	5V	0V
5V	0V	0V
5V	5V	5V

\* for positive logic level :-

0V → logic 0  
5V → logic 1

<u>A</u>	<u>B</u>	<u>Y</u>
-	0	0
0	1	0
0	0	0
1	1	1

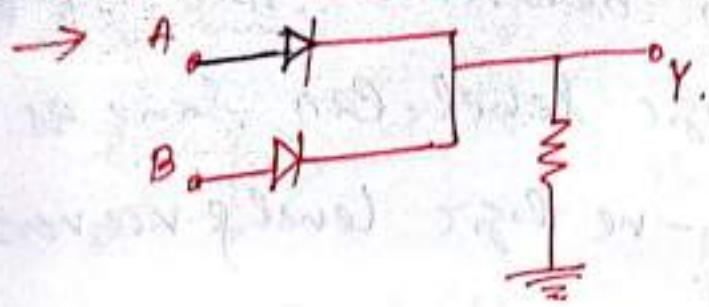
→ AND Gate

\* for negative logic level :-

0V → logic 1  
5V → logic 0.

<u>A</u>	<u>B</u>	<u>Y</u>
1	1	1
1	0	1
0	1	1

→ OR Gate



A	B	Y
0V	0V	0V
0V	5V	5V
5V	0V	5V
5V	5V	5V

① for positive logic level :-

$$\begin{aligned} 0V &\rightarrow \text{Logic '0'} \\ 5V &\rightarrow \text{Logic '1'} \end{aligned}$$

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

→ OR gate

② For negative logic level :-

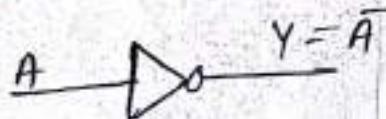
$$\begin{aligned} 0V &\rightarrow \text{Logic '1'} \\ 5V &\rightarrow \text{Logic '0'} \end{aligned}$$

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

→ AND gate.

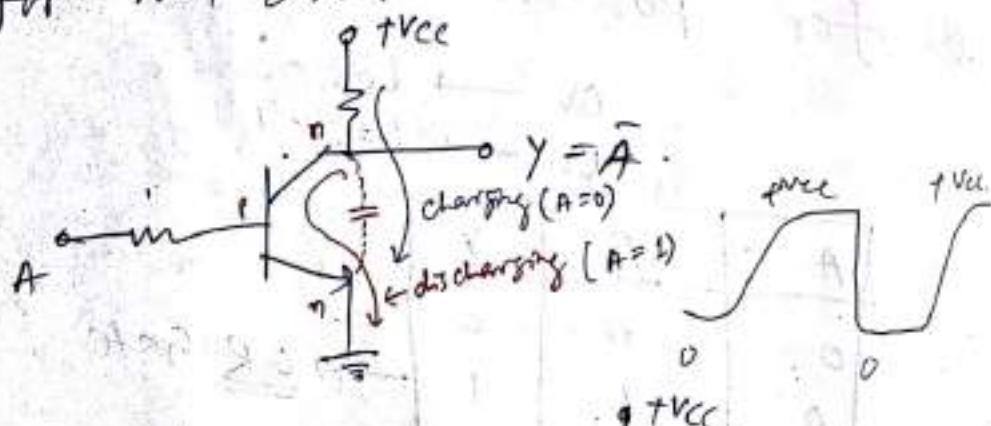
→ A cir. which behaves as AND gate  
 in positive logic level can behave as  
OR gate in -ve logic level vice versa

### NOT Gate



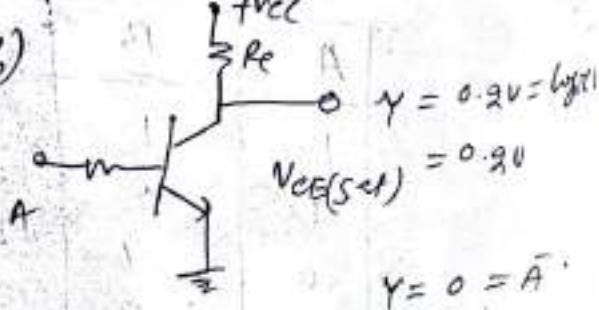
A	$Y = \bar{A}$
0	1
1	0

→ Both +ve & -ve logic level ~~for~~ same  
 for NOT gate.

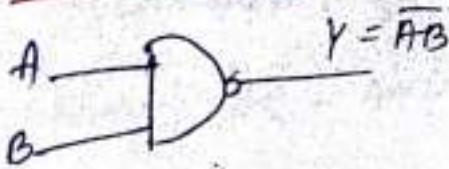


→ When  $A = 0$  (Low)

→ When  $A = 1$  (High)



### NAND Gate



$$\begin{array}{cc} A & B \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array} \quad Y = \bar{A}B$$

0 1  
1 0

1 0  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

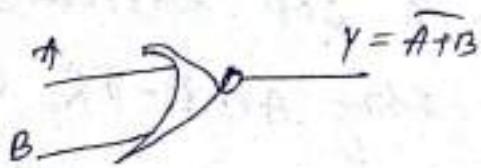
0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

### NOR Gate



$$\begin{array}{cc} A & B \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{array} \quad Y = \bar{A} + \bar{B}$$

0 1  
1 0

1 0  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

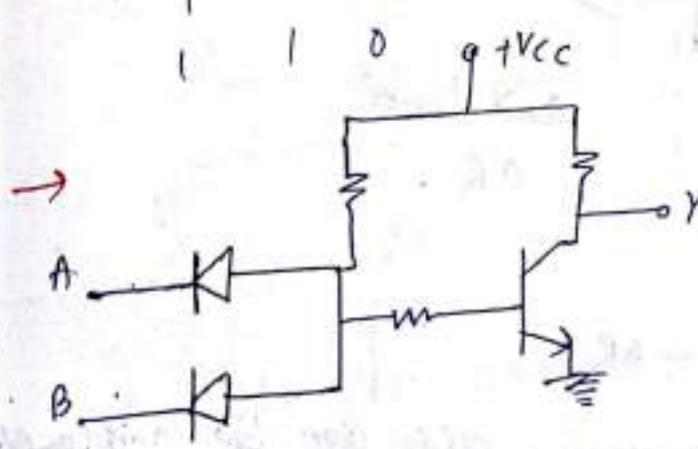
1 1  
1 1

0 1  
1 1

1 0  
1 1

0 0  
1 1

1 1  
1 1



NAND with P.L.L.

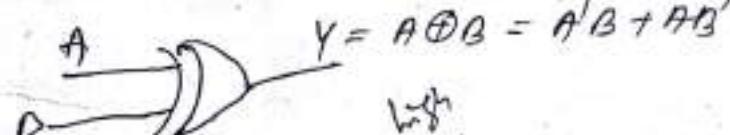
NOR with N.L.L.

NOR with P.L.L.

NAND with N.L.L.

→ A circ. which behaves as NAND in PLL  
is same as NOR in NLL & vice versa.

odd parity check  
X-OR



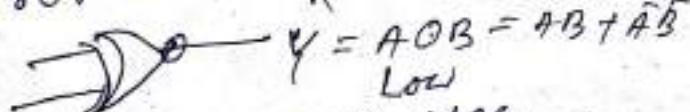
$$Y = A \oplus B = A'B + AB'$$

With

o/p is high, for odd no. of 1's.

o/p is low, for even no. of 1's.

Even parity  
checker.  
X-NR



$$Y = A \ominus B = AB + \bar{A}B'$$

Low

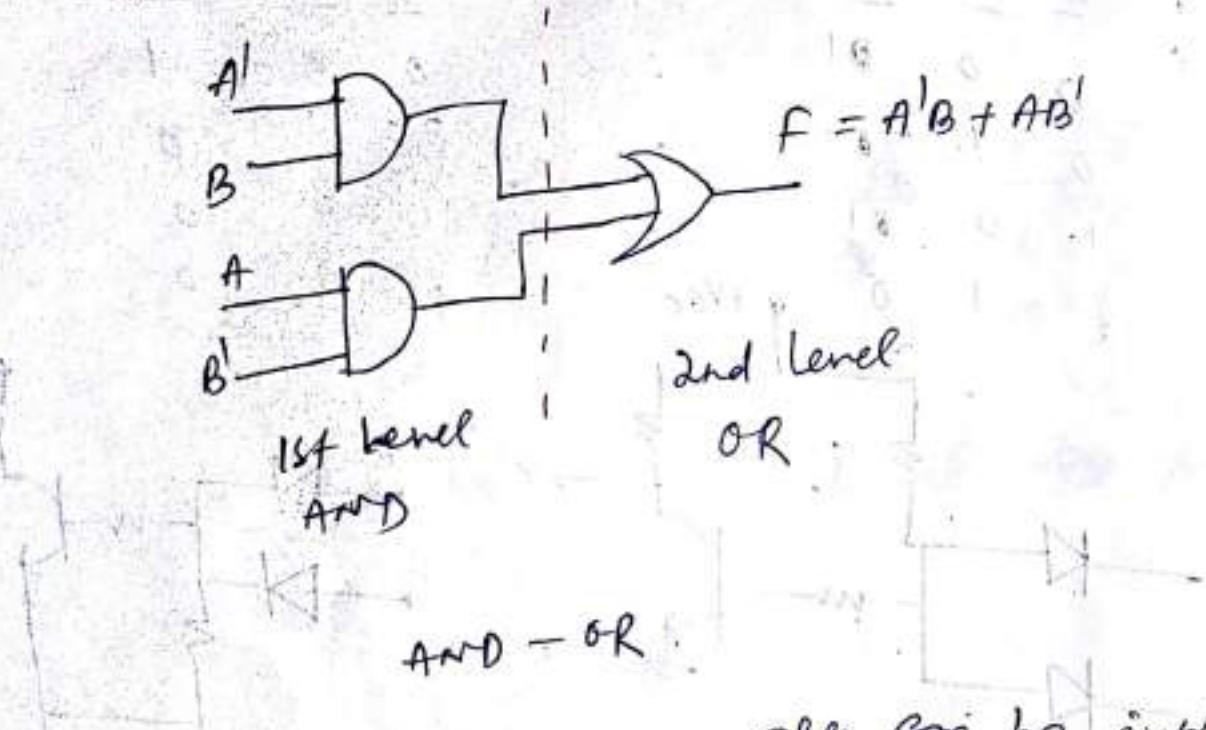
o/p is high, for even no. of 1's.

o/p is low, for odd no. of 1's.

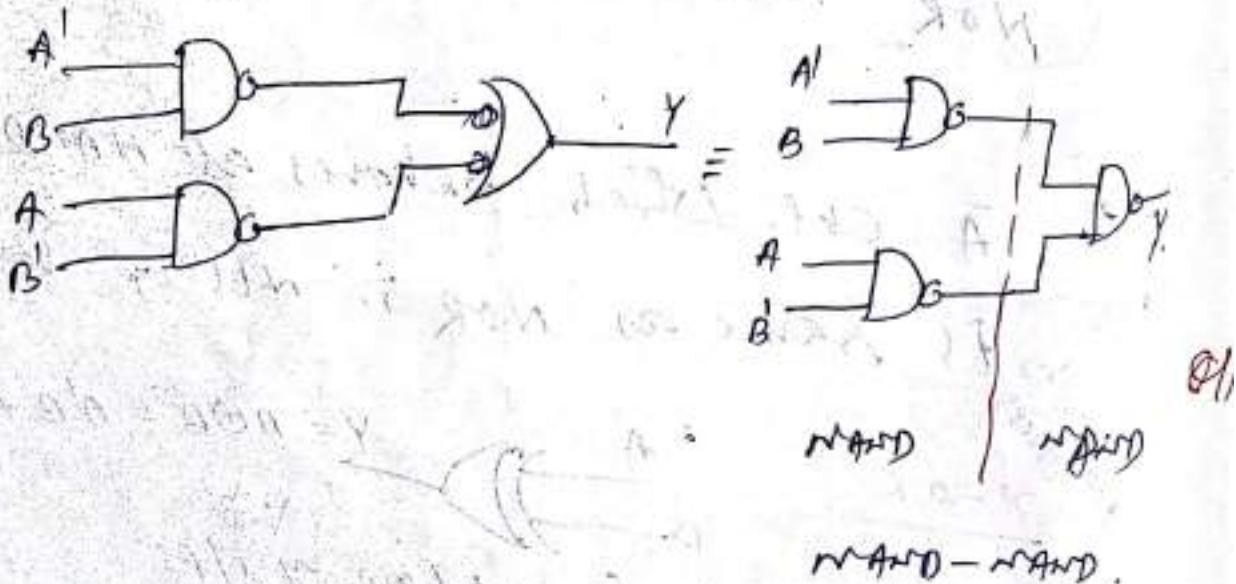
## SOP expression

→ A sop expression can be implemented by using AND-OR cells.

$$F = A'B + AB'$$



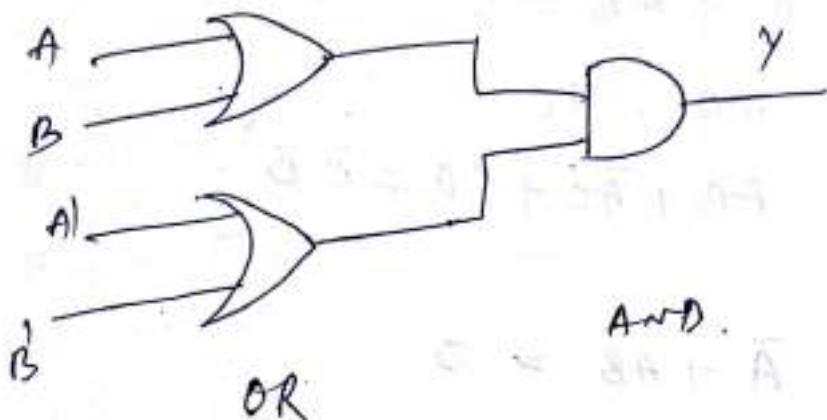
→ The SOP expression also can be implemented by using NAND-NAND cells.



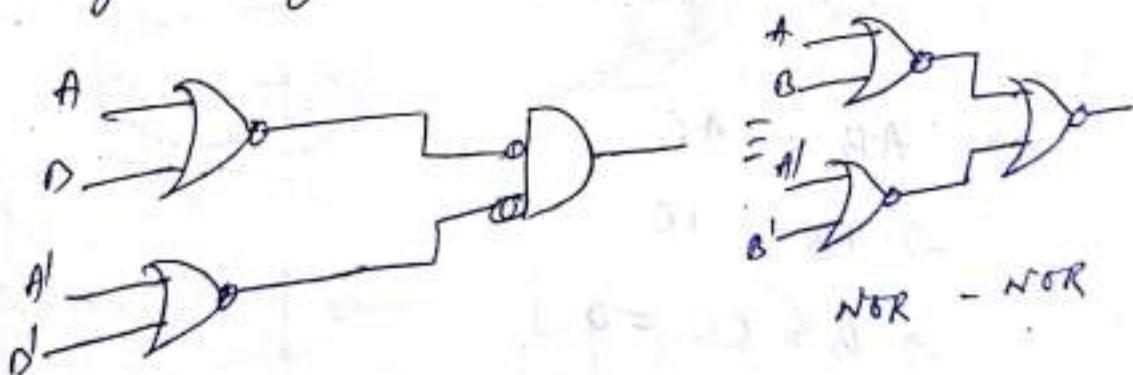
## POS expression

→ The pos expression can be implemented by using OR - AND ~~OR~~<sup>AND</sup>.

$$y = (A+B)(A^T+B^T)$$



→ The pos expression also can be implemented by using NOR-NOR cuts.



Q11 What happens when a Bit string is XORed itself n-times.

$$\text{Ans} \quad [B \oplus \underbrace{B \oplus (B \oplus (B \oplus \dots))}_{n \text{ times}}]$$

If  $n$  is even bit string remains unchanged  
 If  $n$  is odd bit string ~~is changed~~ changes

\* Sfonet at '1' means fixed at '1'.

Q11 What value of A, B, C & D satisfy  
the simultaneous Boolean eqns.

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

$$AB = AC$$

$$AB + \bar{AC} + CD = \bar{C} \bar{D}$$

$$\stackrel{\text{from}}{\rightarrow} \bar{A} + AB = 0$$

$$\stackrel{\text{from}}{\rightarrow} \bar{A} + B = 0$$

$$\Rightarrow A = 1, B = 0$$

$$\stackrel{\text{from}}{\rightarrow} AB = AC$$

$$\Rightarrow 1B = 1C$$

$$\Rightarrow B = C = 0$$

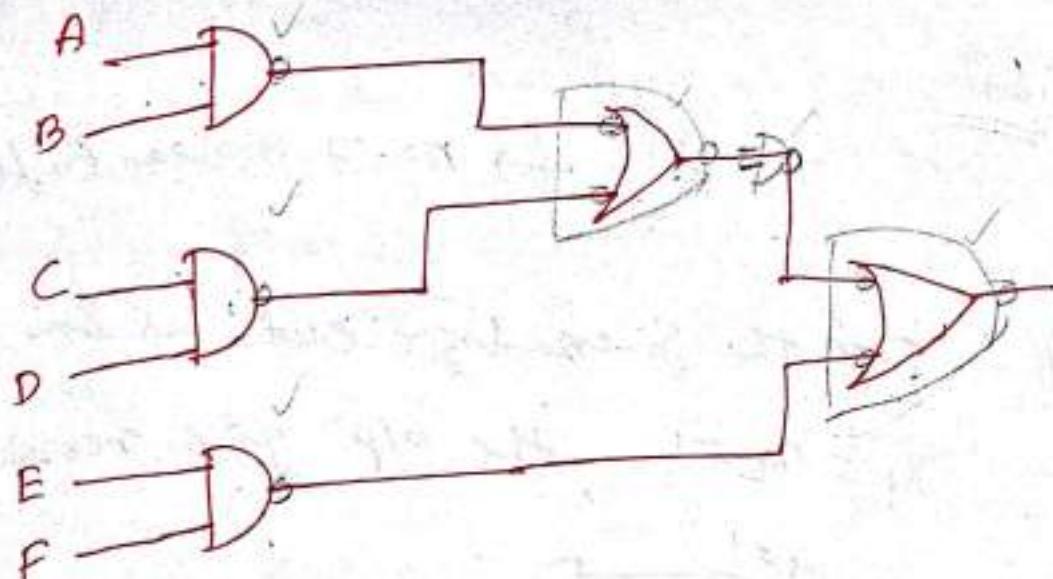
$$\stackrel{\text{from}}{\rightarrow} AB + \bar{AC} + CD = \bar{C} \bar{D}$$

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

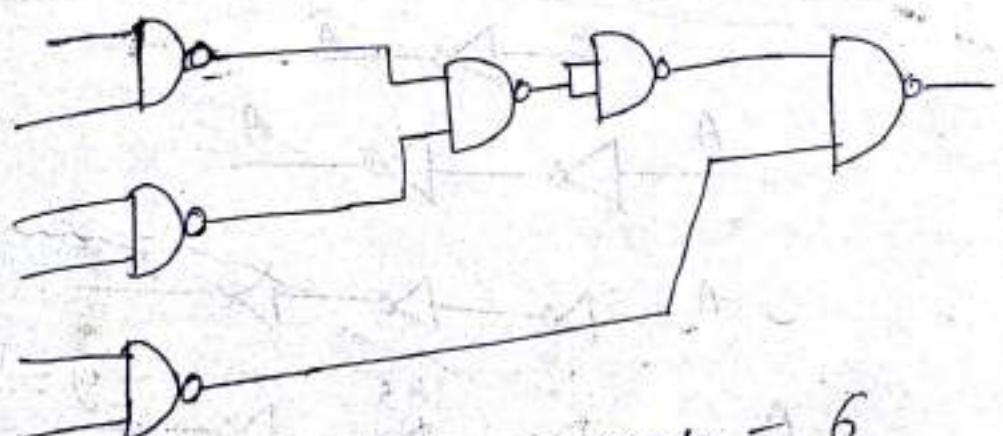
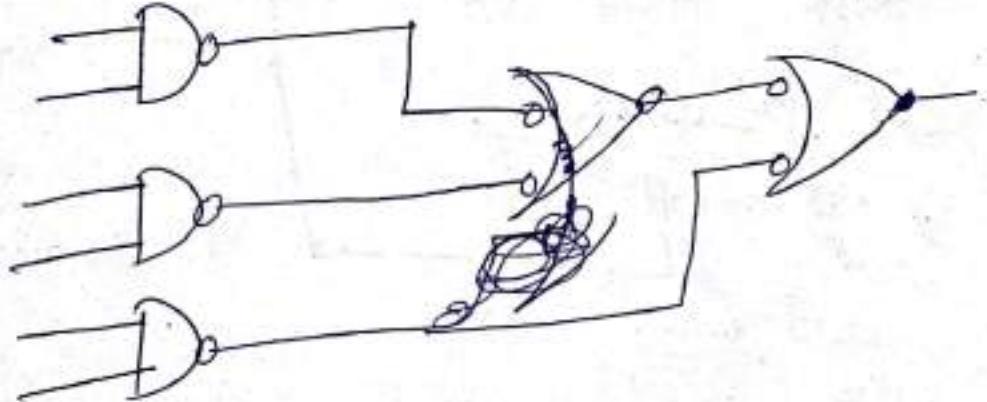
$$\Rightarrow D = 0$$

$$A = 1, B = 0, C = 0, D = 0 \quad (\text{Ans})$$

Q11 How many 2 input NAND gate require to implements the following logic & cells.



Ans

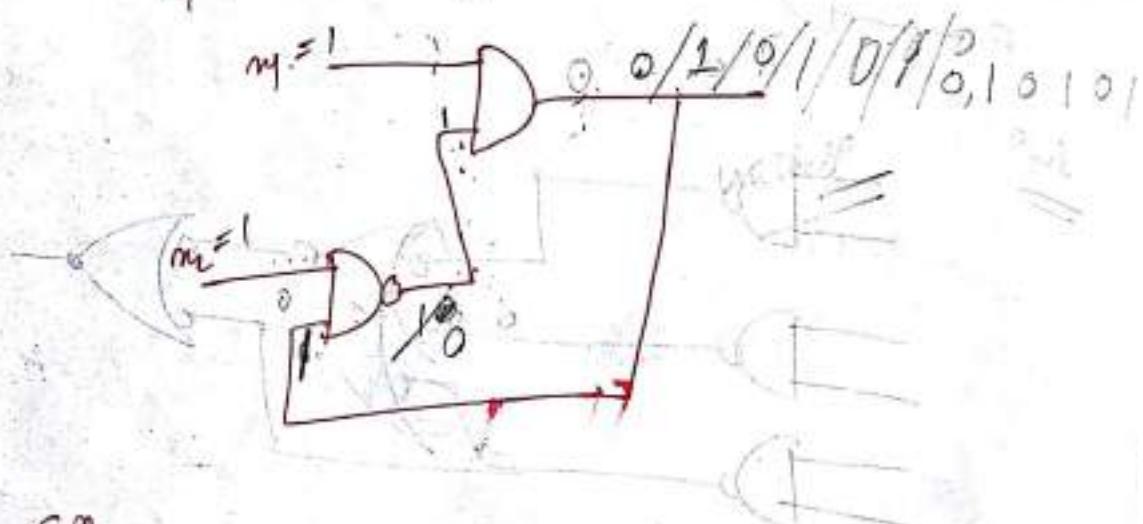


No. of NAND gate = 6.

Off By using 'n' variables we can have,  
Boolean fun / expression.

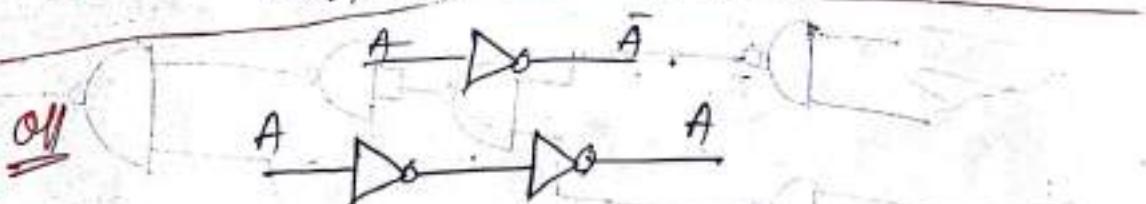
Surj  $(2^n)^n \rightarrow$  no. of Boolean fun/Bxpr.

Off For the given logic circuit as long as  
 $x_1 = x_2 = 1$ , the off gives remains unstable

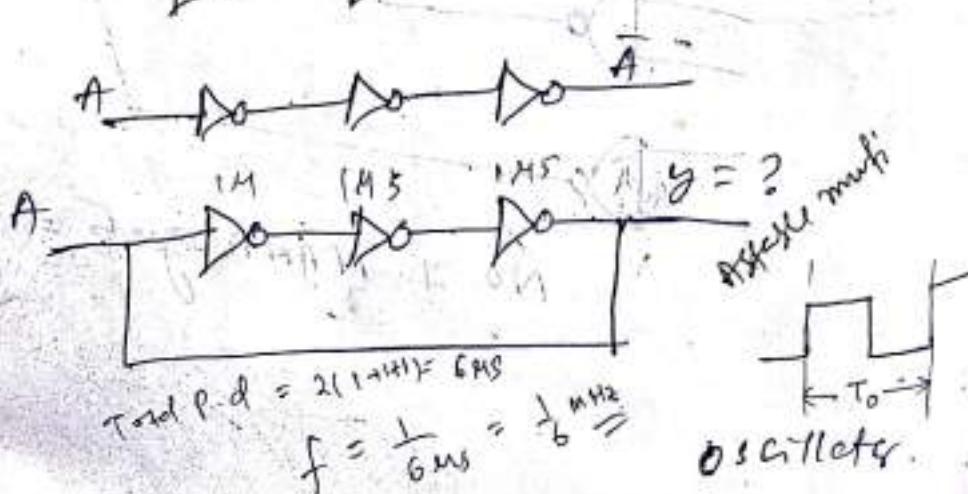


Surj

unstable.



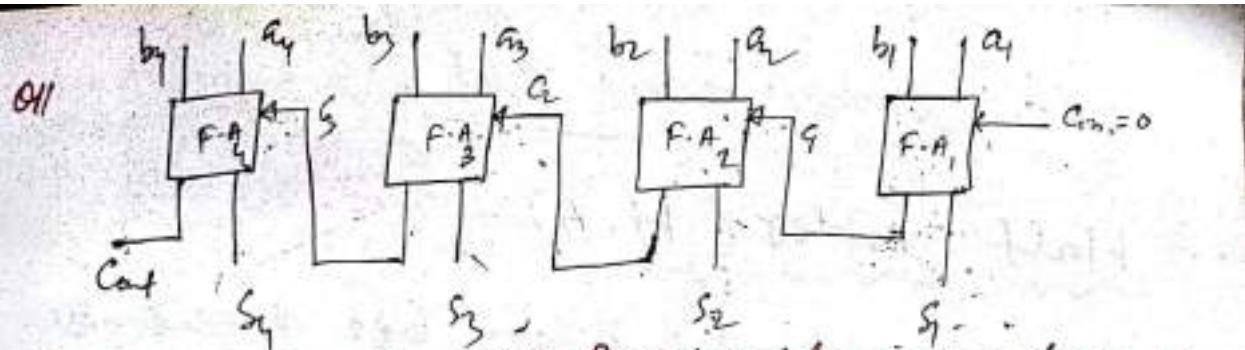
Off



$$\text{Total P-d} = 2(1M + 1M5) = 6\text{ms}$$

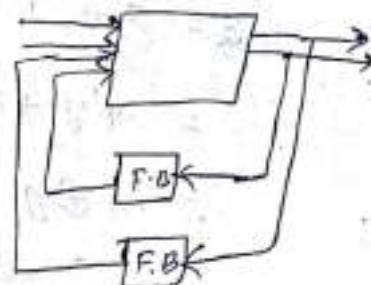
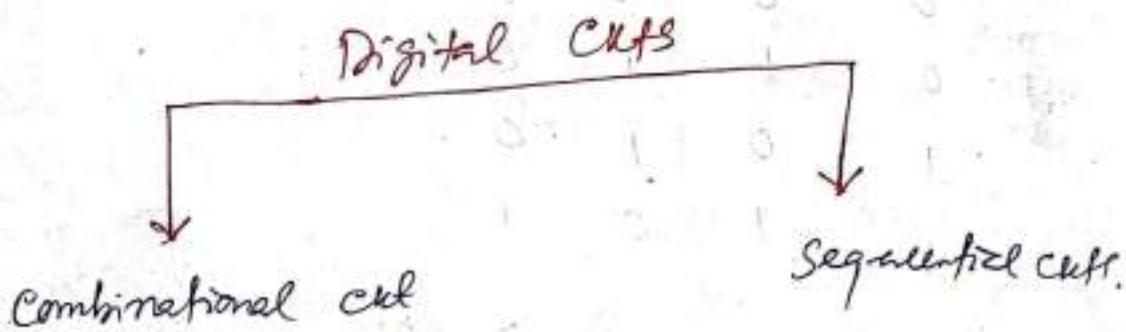
$$f = \frac{1}{T} = \frac{1}{6\text{ms}} = 166.66 \text{ Hz}$$

oscillator.



In the above parallel Binary adder if F.A takes 32 nsec to produce the sum of 14 nsec to produce carry. So the total time taken for addition.

$$\begin{aligned} & 32n \\ & 42 + 32 \quad 28 + 32 \quad 14 + 32 \\ & \text{Total} = 74 \text{ nsec} \end{aligned}$$



→ op depends on only present i/p.

Ex: F.A., F.S., H.A., H.S

Decoder, Encoder, MUX,  
demux, ROM, Binary Parallel Adder, <sup>mag</sup> Comparator etc.

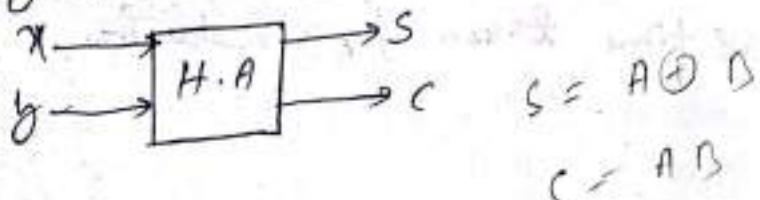
→ op depends on present i/p & previous op also.

Ex: Register, Counter,  
Serial adder,  
Sequence generator,  
Sequential machine etc.

## Combinational Circ.

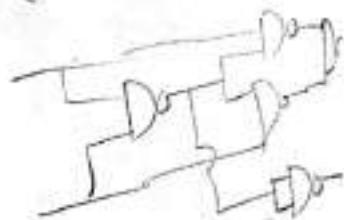
-: Half Adder (H.A) :-

It adds two binary bits & generates sum & carry.



T. T

<u>x</u>	<u>y</u>	<u>s</u>	<u>c</u>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



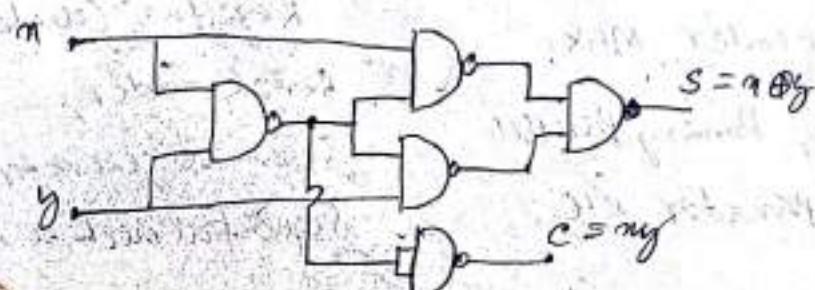
$$S = \sum m(1, 3), \quad C = \sum m(3)$$

$$= \bar{x}y + xy' \quad = xy.$$

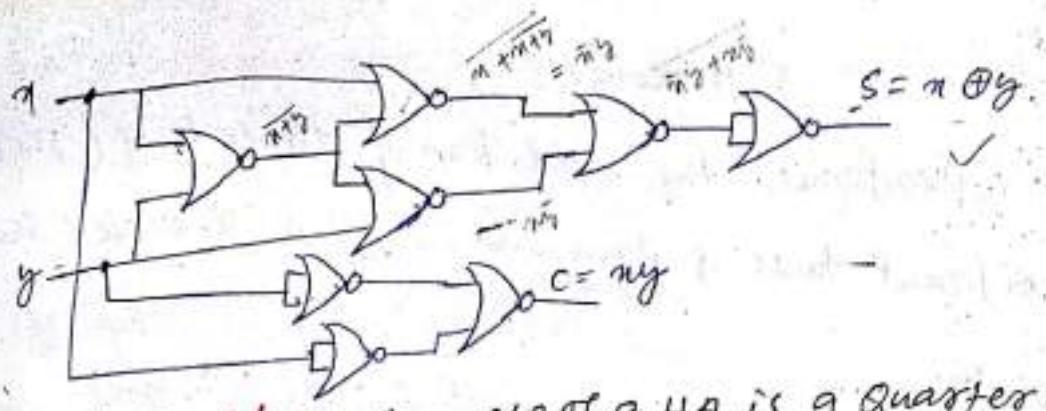
$$= x \oplus y$$

$$x \rightarrow \text{Inverter} \rightarrow \text{OR gate} \rightarrow S = x \oplus y$$

$$y \rightarrow \text{Inverter} \rightarrow \text{AND gate} \rightarrow c = xy$$



Minimum no. of NAND gate required to design H.A :-



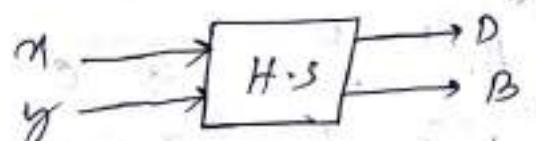
quarter adder: sum opp of a HA is a Quarter adder.

Half Subtractor (H.S.):

It subtracts two binary bits to generate difference (D) & Borrow (B).

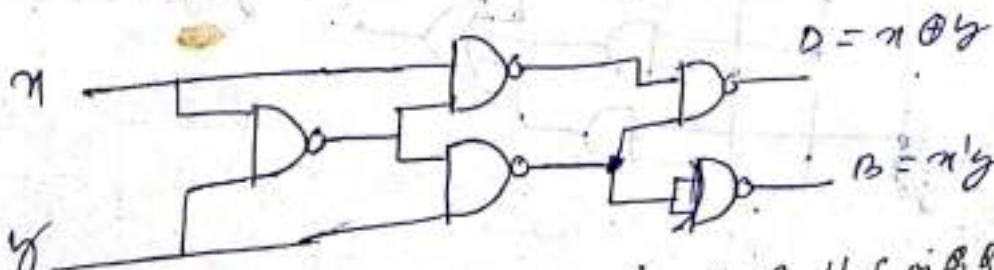
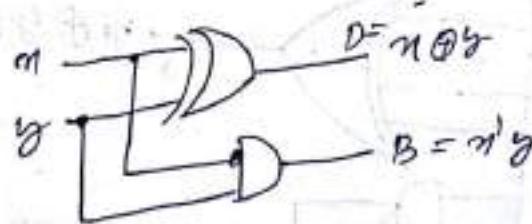
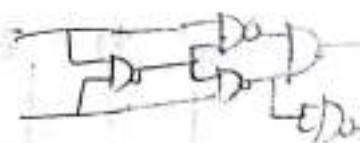
$$S = \bar{x} \oplus y$$

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



$$\begin{array}{c} T, T \\ \hline x & y & D & B \\ \hline 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{array} \quad D = x'y + xy' = x \oplus y$$

$$B = x'y$$



Quarter Subtractor: Difference opp. of a H.S in Q subtraction

- ④ H.S can be converted to H.A. by using an extra inverter.

## Full adder

It performs the addition of three bits (Two significant bits & previous carry) & generates sum & carry.

T. T			$x \oplus y \oplus z$		$S = \sum m(1, 3, 7, 7)$			
$n$	$y$	$z$	$S$	$C$	$n$	$y$	$z$	$S = \sum m(1, 3, 4, 7)$
0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1	1
0	1	0	1	0	1	1	0	0
0	1	1	0	1	1	0	1	1
1	0	0	0	1	1	0	0	0
1	0	1	0	1	1	1	0	1
1	1	0	1	1	1	0	1	1
1	1	1	1	1	1	1	1	1

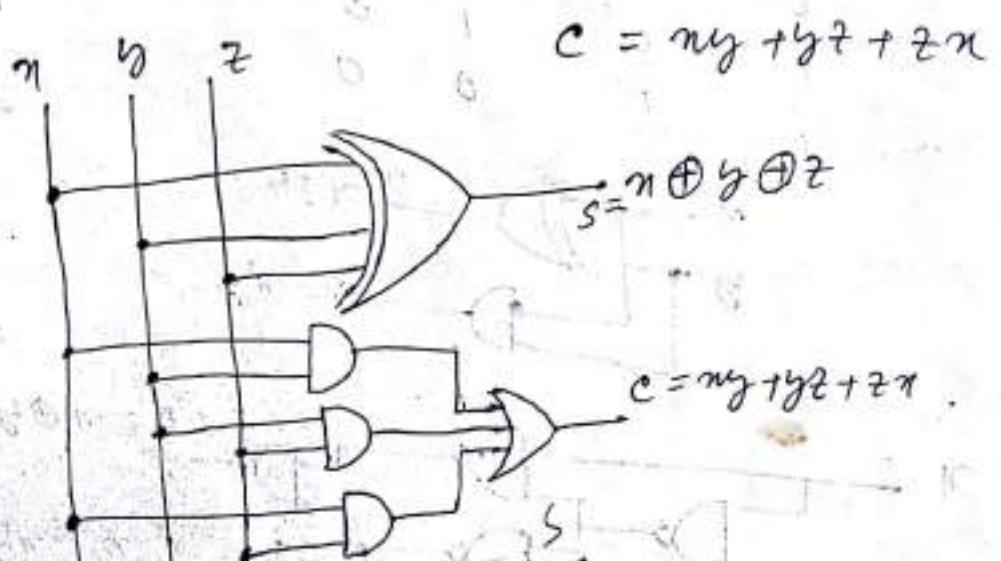
$$S = n \oplus y \oplus z$$

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

$$\textcircled{*} \quad C = \sum m(3, 5, 6, 7)$$

$n$	$y$	$z$	$C$
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

$$C = ny + yz + zx$$



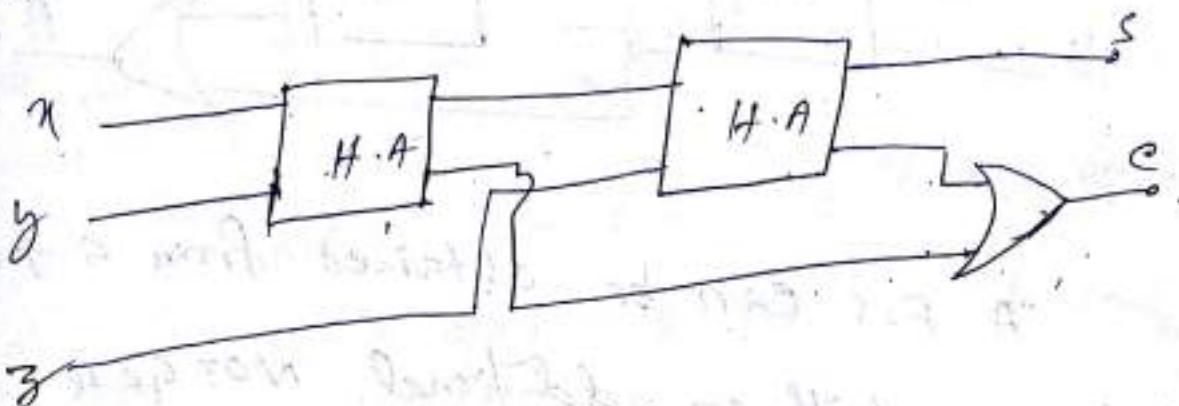
$$C = ny + yz + zx$$

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

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

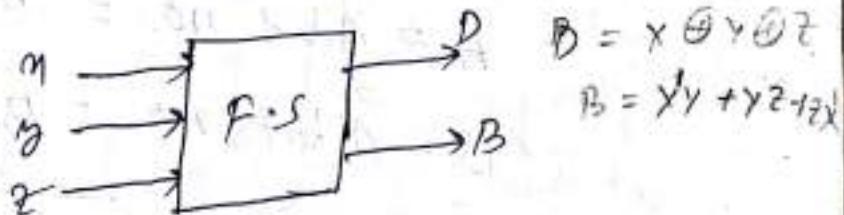
## full adder using Half adder

A full Adder can be implemented using 2 H.A & one OR gate.



## Full Subtractor

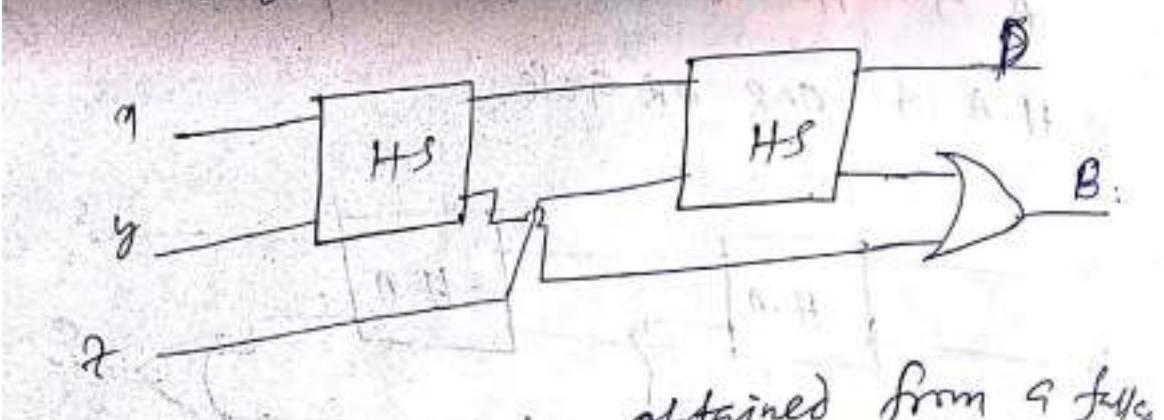
This circuit performs the subtraction of 3 binary variables & generates the diff. & Borrow as opps.



T. T

$m$	$y$	$z$	$D$	$B$	$D = \sum m(1, 2, 4, 7)$
0	0	0	0	0	$B = \sum m(1, 2, 3, 7)$
0	0	1	1	1	
0	1	0	1	1	$D = m \oplus y \oplus z$
0	1	1	0	1	$B = m'(y+z) + yz$
1	0	1	1	0	$= m'y + yz + m'z$
1	0	0	0	0	
1	1	0	0	1	
1	1	1	1	1	$\sum m(1, 2, 3, 7)$
					$D = \sum m(1, 2, 3, 7)$

→ A F.S can be design using two H.s &  
OR gate.

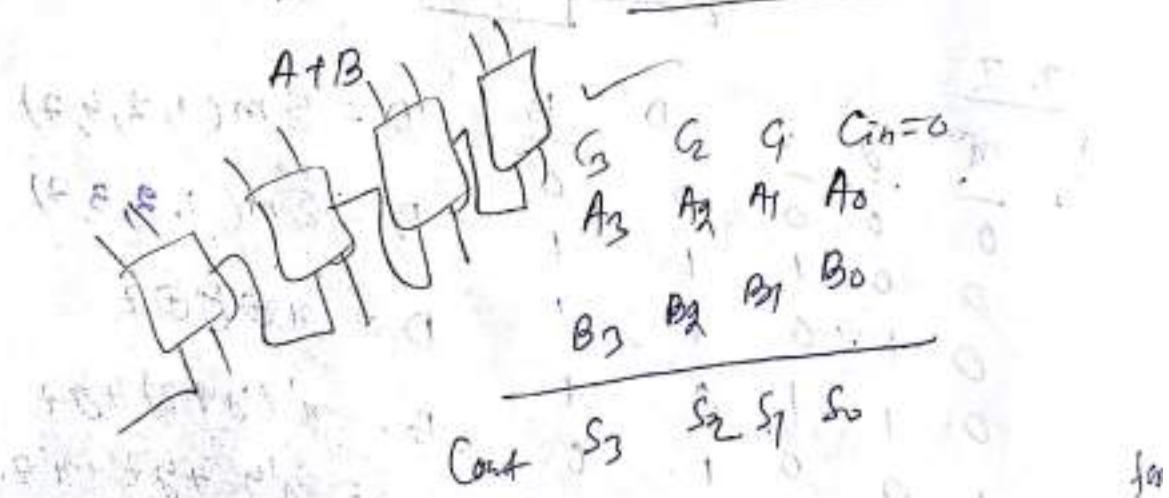


\* A F.S can be obtained from a full  
ckt with an additional NOT gate.

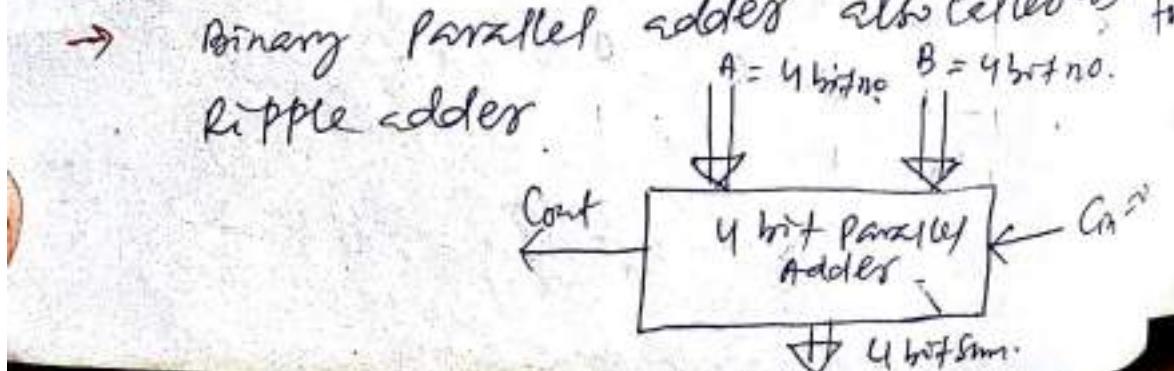
→ Binary parallel Adder :-

$$A = 4 \text{ bit no.} = A_3 A_2 A_1 A_0$$

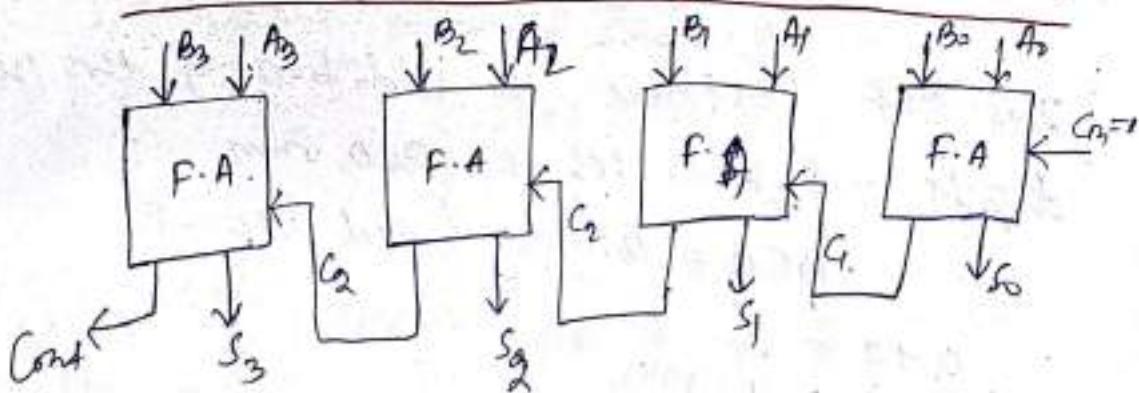
$$B = 4 \text{ bit no.} = B_3 B_2 B_1 B_0$$



→ Binary parallel adder also called as  
Ripple adder.



## Design a 4-bit Binary Parallel Adder

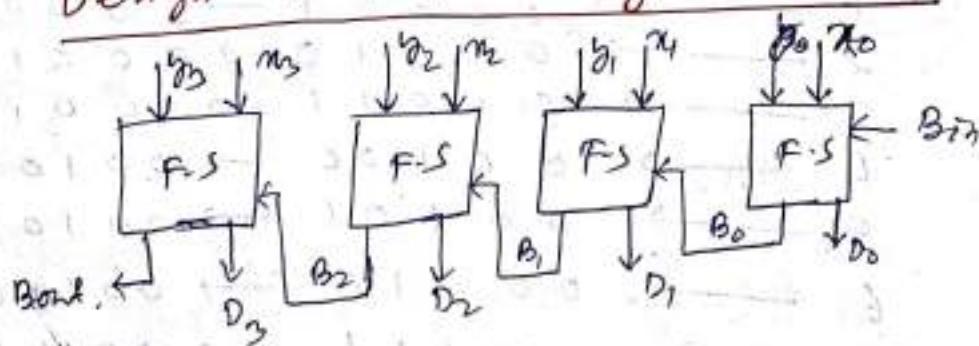


No. of i/p m = 9: ( $C_0, A_0, A_1, A_2, B_3, B_0, B_1, B_2, B_3$ )

No. of o/p r = 5: ( $S_0, S_1, S_2, S_3, \text{Cont}$ )

→ Carry Look ahead adder is faster adder.

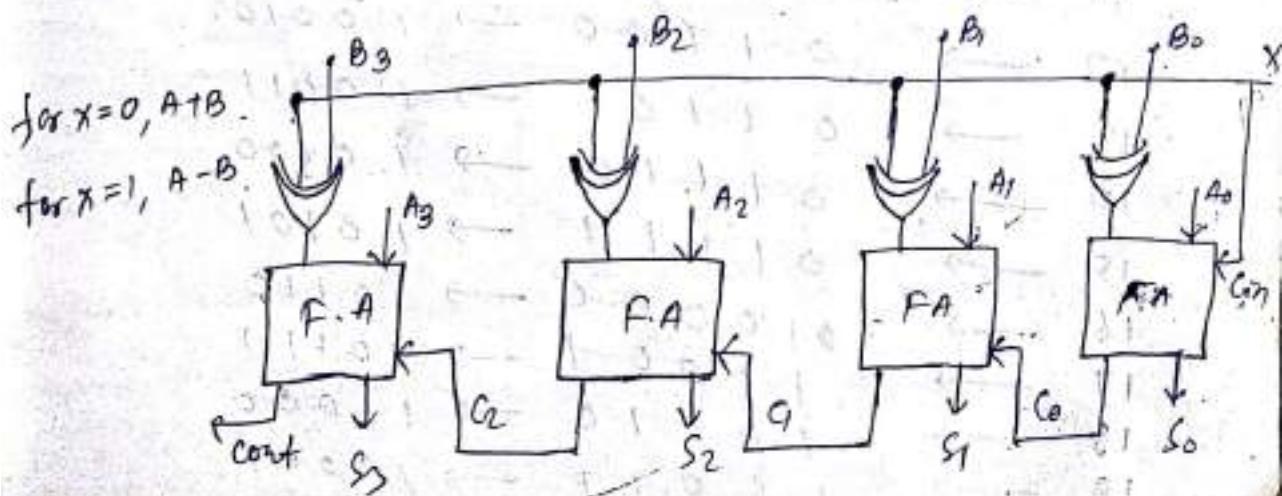
## Design a 4-bit Binary Subtractor



## Design a 4-bit Binary adder/Subtractor

$$A + B, \quad A - B = A + \bar{B} + 1$$

$$B_0 \oplus 1 = \bar{B}_0 \quad \& \quad B_0 \oplus 0 = B_0$$



- BCD adder :-

This circuit performs the addition of two decimal digits & generates the BCD sum.

BCD = Binary Coded Decimal.

$$0+0=0 \quad 9+6 \rightarrow (1001) + (0110) = 1111$$

$$\begin{array}{r} \text{BCD} \\ 0110 \\ + 0101 \\ \hline \end{array}$$

$$0+1=1$$

$$0+2=2$$

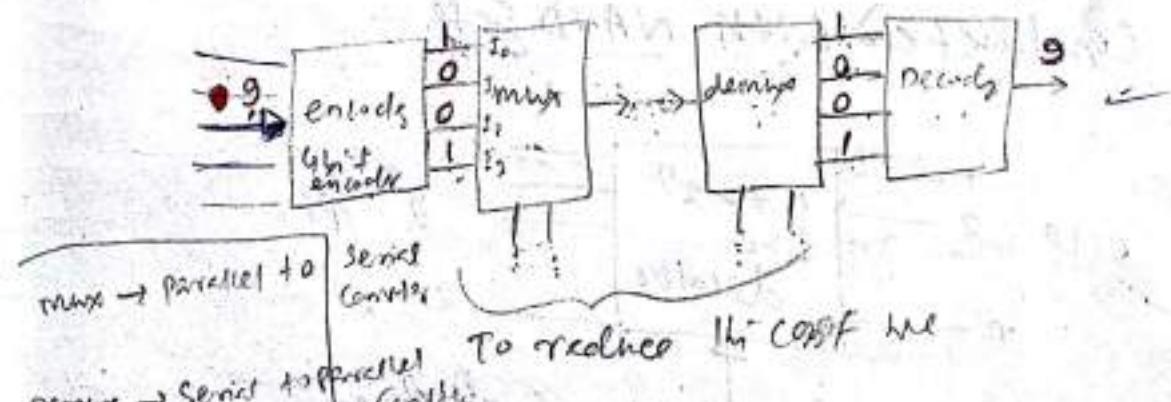
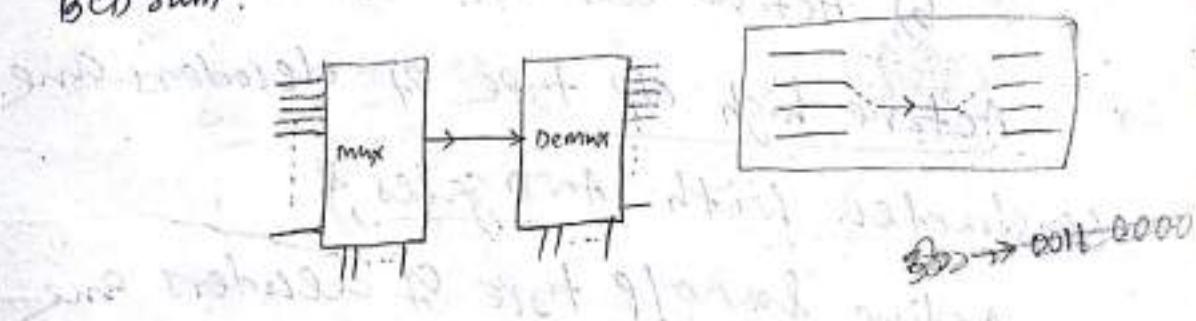
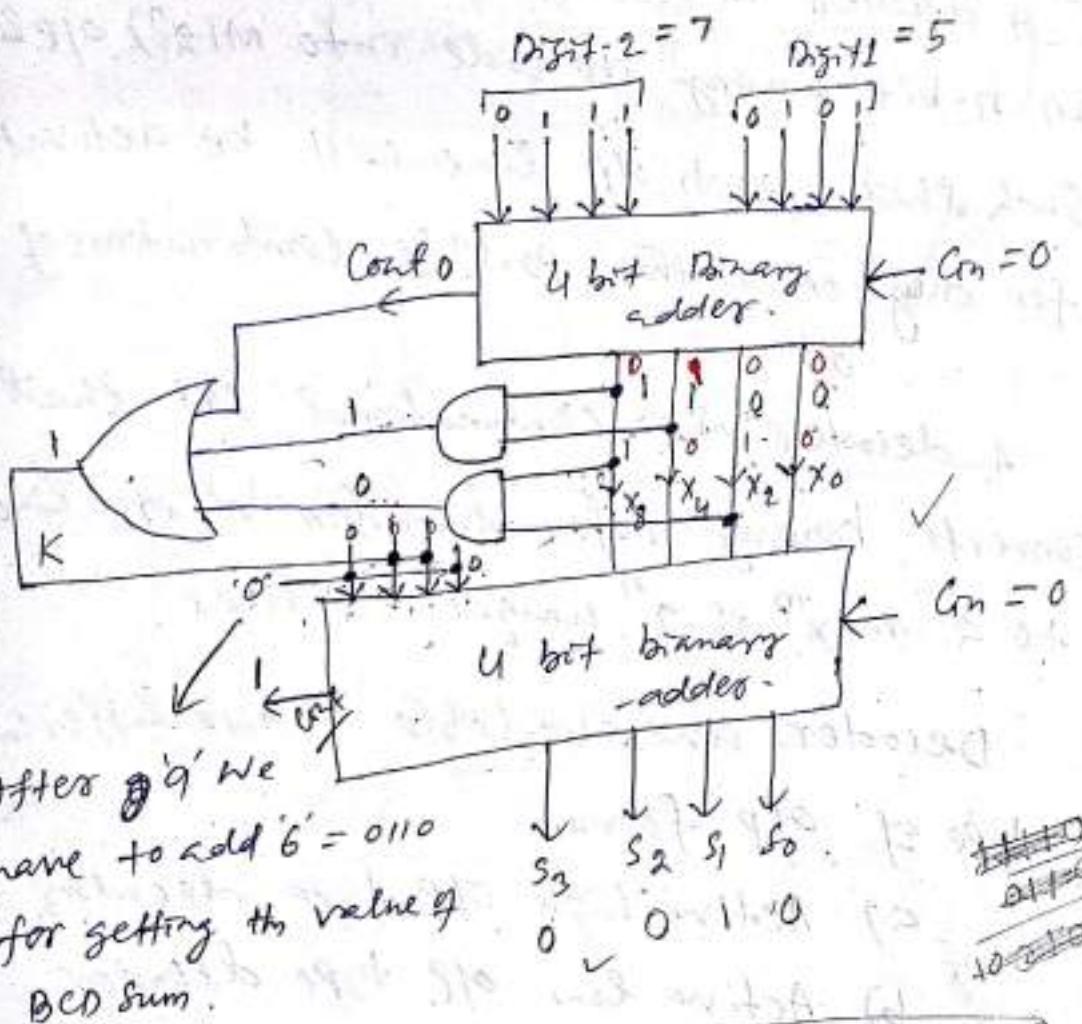
$$9+9=18 \quad (1001) + (1001) = 10010$$

$$\begin{array}{r} \text{BCD} \\ 1001 \\ + 1001 \\ \hline 0001 \end{array}$$

$$9+9+1=19.$$

Decimal Sum	Binary Sum	BCD Sum
	Coat \$x_8\ x_7\ x_4\ x_2\ x_0\$	\$K\ S_3\ S_2\ S_1\ S_0\$
0	0 0 0 0 0	0000 0 0 0 0
1	0 0 0 0 1	0000 0 0 0 1
2	0 0 0 1 0	0000 0 0 1 0
3	0 0 0 1 1	0000 0 0 1 1
4	0 0 1 0 0	0000 0 0 1 0 0
5	0 0 1 0 1	0000 0 0 1 0 1
6	0 0 1 1 0	0000 0 0 1 1 0
7	0 0 1 1 1	0000 0 0 1 1 1
8	0 1 0 0 0	0000 0 1 0 0 0
9	0 1 0 0 1	0000 0 1 0 0 1
10	0 1 0 1 0	0000 0 1 0 1 0
11	0 1 0 1 1	0001 0 0 0 1
12	0 1 1 0 0	0001 0 0 1 0
13	0 1 1 0 1	0001 0 0 1 1
14	0 1 1 1 0	0001 0 1 0 0
15	0 1 1 1 1	0001 0 1 0 1
16	1 0 0 0 0	0001 0 1 1 0
17	1 0 0 0 1	0001 0 1 1 1
18	1 0 0 1 0	0001 1 0 0 0
19	1 0 0 1 1	0001 1 0 0 1

$$K = \text{Const} + X_8 X_4 + X_8 X_2$$



Why not encoder & decoder serial to parallel connection  
Parallel to serial conversion instead of taking connection of parallel wires between the encoder and decoder

## Decoder

Def<sup>n</sup>

A decoder is a logic circuit that converts an  $n$ -bit binary I/P code into  $M(2^n)$  O/P lines such that each O/P line will be activated for only one of the possible combinations of I/P.

or

A decoder is a combinational circuit that converts binary information from ' $n$ ' I/P lines to a max<sup>m</sup> of  $2^n$  unique O/P lines.

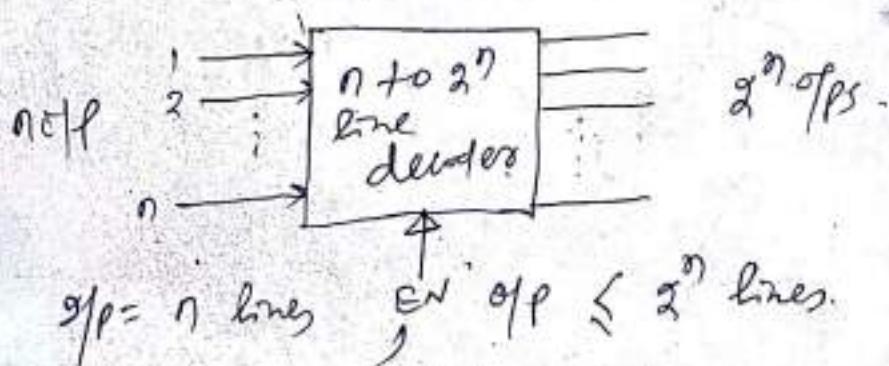
→ Decoders are available in two different type of O/P forms.

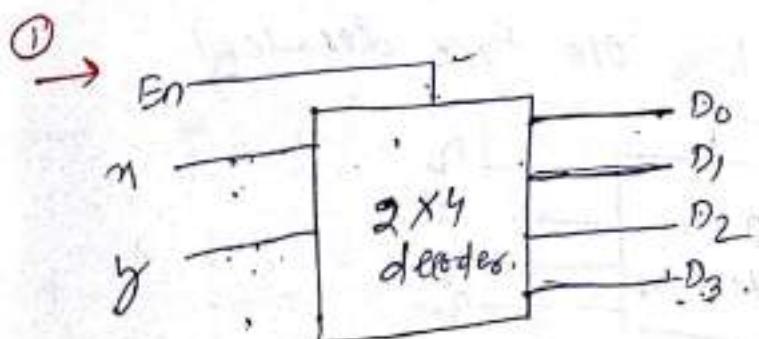
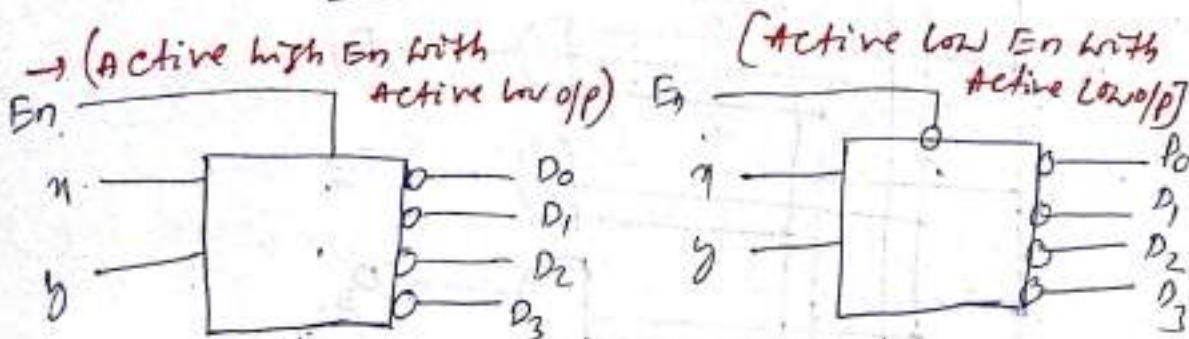
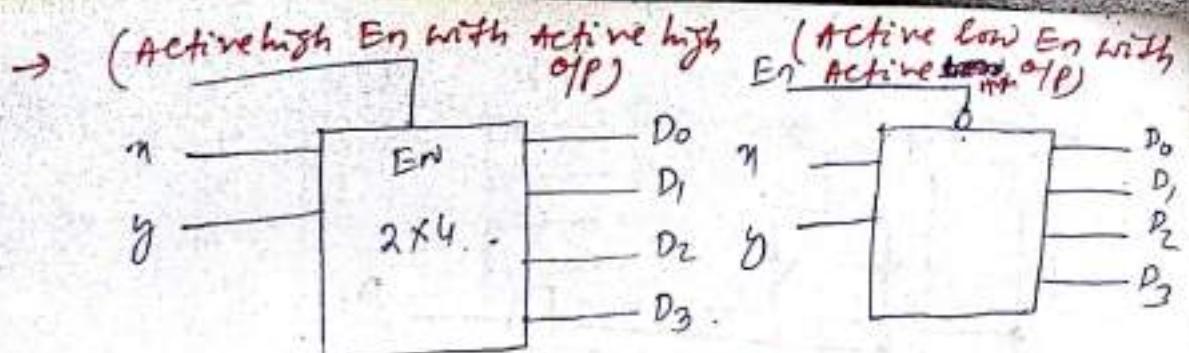
a) Active high O/P type decoder

b) Active low O/P type decoder

→ Active high O/P type of decoders are constructed with AND gates, &

Active low O/P type of decoders are constructed with NAND gates.





01111111  
— — 3132

<u>T.T</u>	<u>m</u>	<u>y</u>	<u>D<sub>0</sub></u>	<u>D<sub>1</sub></u>	<u>D<sub>2</sub></u>	<u>D<sub>3</sub></u>
En	0.	0.	1	0	0	0
1	0	1	0	1	0	0
1	0	0	0	0	1	0
1	1	0	0	0	0	1
0	X	X	0	0	0	0

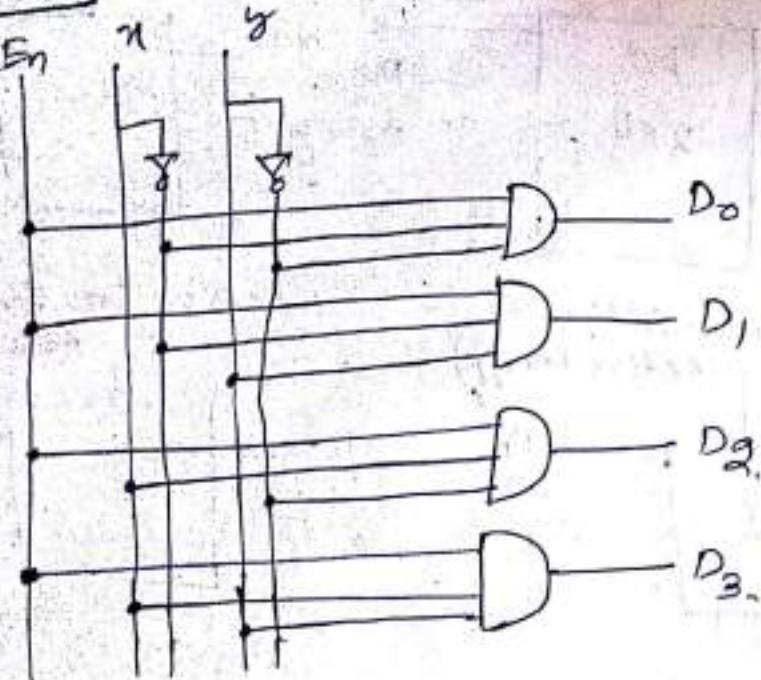
$$D_0 = \bar{m} \bar{y} = m_0 = \text{minterm}$$

$$D_1 = \bar{m} y = m_1 = \text{minterm}$$

$$D_2 = m \bar{y} = m_2 = \text{minterm}$$

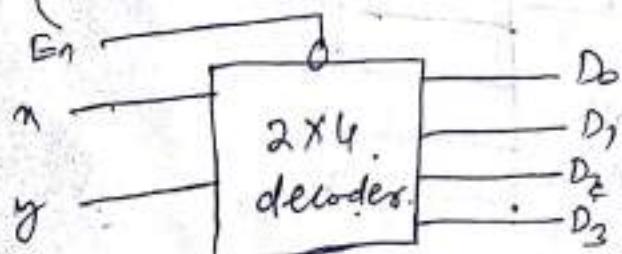
$$D_3 = m y = m_3 = \text{minterm}$$

Design



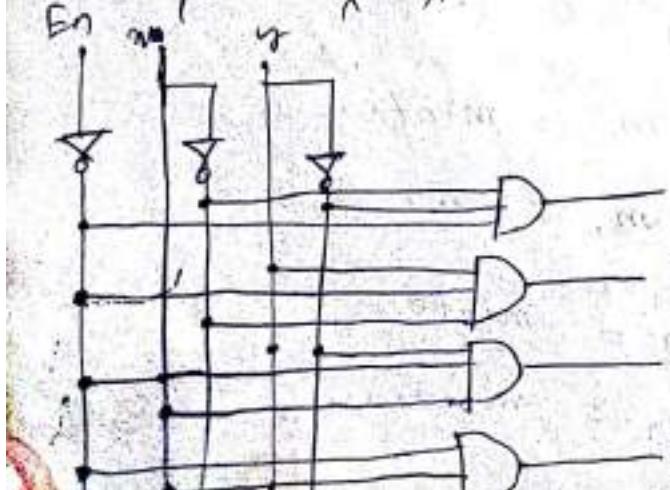
(Active high ORP type decoder)

②



En	x	y	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>
0	0	0	1	0	0	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	1	0	0	0	1

En      x      y      x̄      ȳ

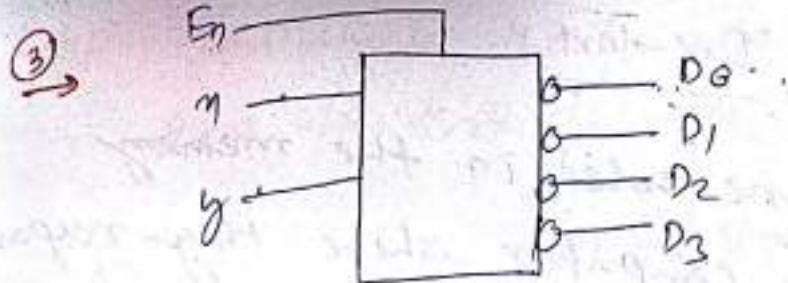


$$D_0 = \bar{m}_0 = m_0 = m_3 t_{EN}$$

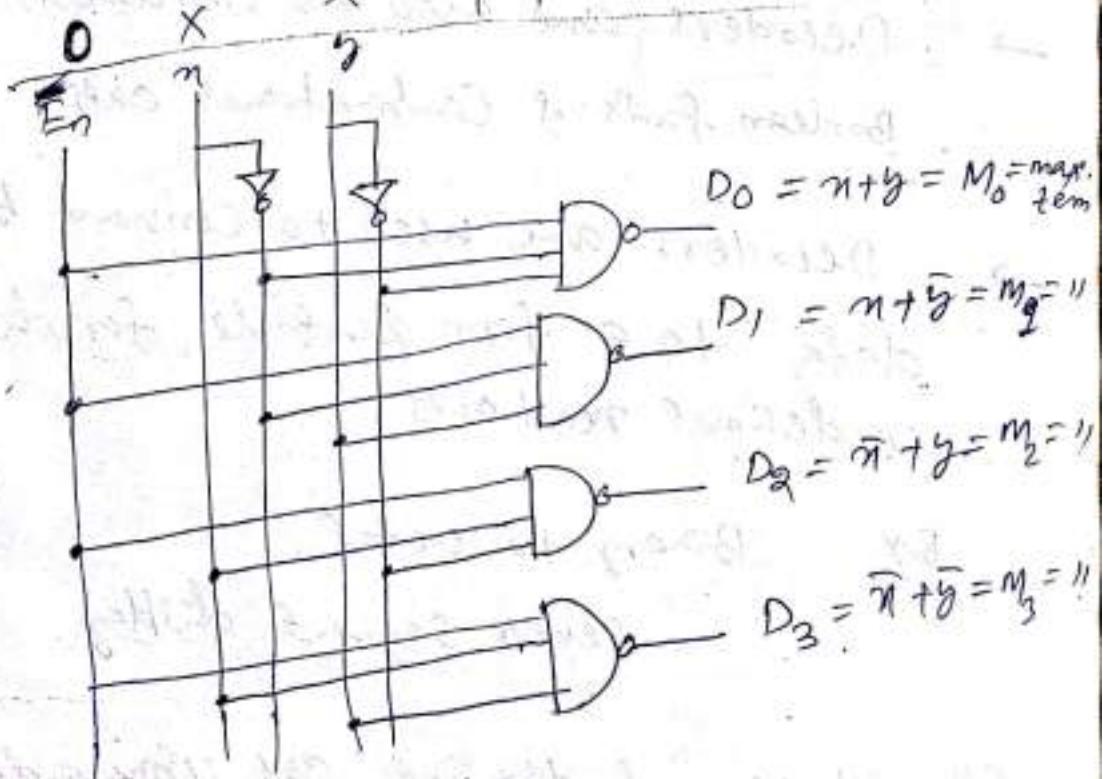
$$D_1 = \bar{m}_1 = m_1 = "$$

$$D_2 = \bar{m}_2 = m_2 = "$$

$$D_3 = \bar{m}_3 = m_3 = "$$



<u>En</u>	<u>n</u>	<u>y</u>	<u>D<sub>0</sub></u>	<u>D<sub>1</sub></u>	<u>D<sub>2</sub></u>	<u>D<sub>3</sub></u>
0	0	0	0	1	1	1
1	0	1	1	0	1	1
1	1	0	1	1	0	1
1	1	1	1	1	1	1



→ For Active low decoder has the o/p's are max. term.

→ 3-to-8 line decoder = Binary to octal decoder  
= 1-of-8 decoder  
b'cs. only one of the 8 o/p's is activated at a time

## Applications of Decoders

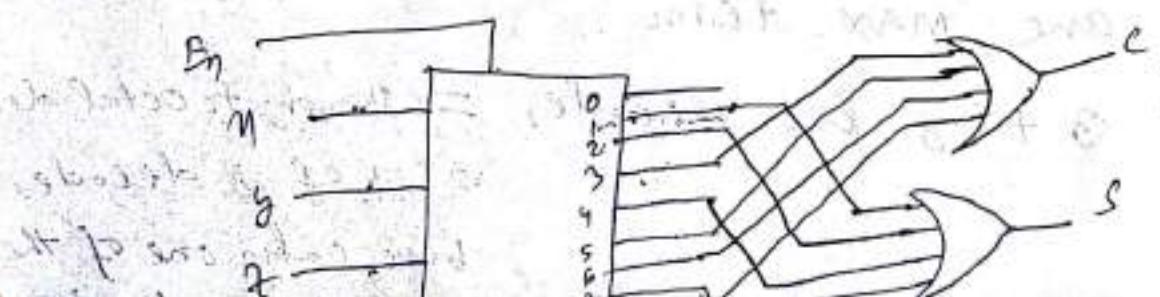
- Decoders are used in the memory system of a computer, where they map to the address code from the CPU to access the memory storage location specified by the address code, i.e. Decoders are used to identify a memory location.
- Decoders are used to implement Boolean func's & Combinational circ.
- Decoders are used to convert binary data to a form suitable for display on decimal need outs.

B/X Binary to octal.

seven segment display.

Q1) Implement the F.A circuit using a decoder

Soln F.A       $S = \sum m(1, 2, 4, 7)$ ,  $C = \sum m(3)$

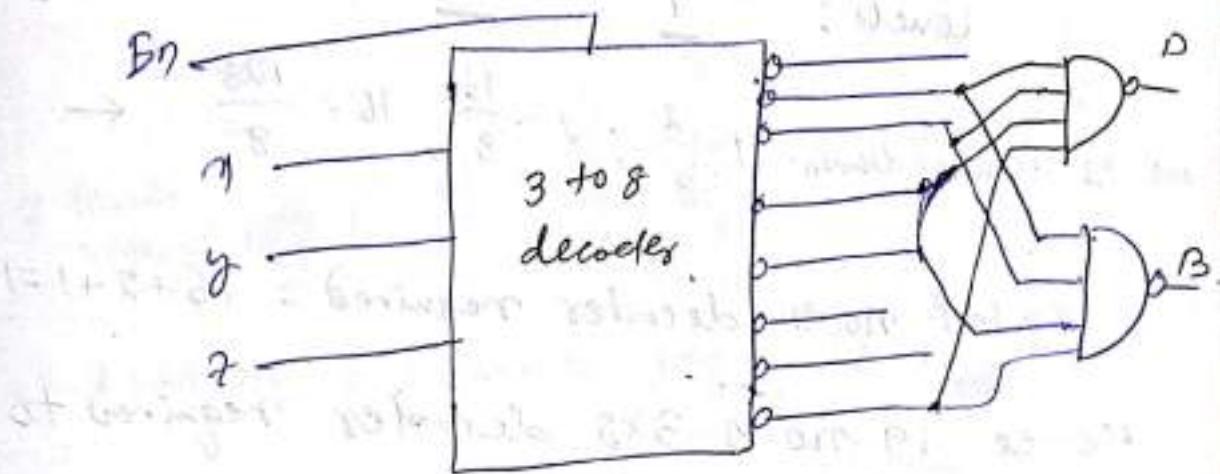


Q11 Implement the f.s cuts using active low op type decoder.

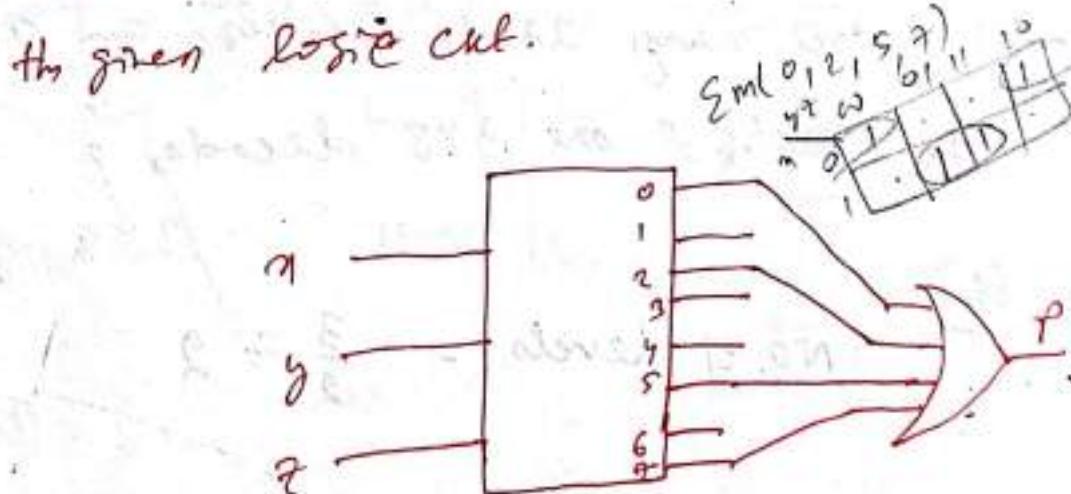
Soln

$$D(n, y, z) = \sum m(1, 2, 4, 7)$$

$$B(n, y, z) = \sum m(1, 2, 3, 5)$$



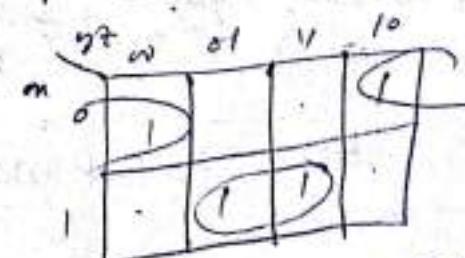
Q11 obtain the expression for o/p 'P' for the given logic cut.



Soln

$n$	$y$	$z$	$P$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0

$$P = \Sigma m(0, 2, 5, 7)$$



$$P = m_0' z' + m_2' z = m_0' z$$

Q11 How many  $3 \times 8$  decoder required to  
construct  $7 \times 128$  decoders

$$\frac{7}{3} \approx 3$$

soln No. of levels =  $\frac{7}{3} \approx 3$

levels :  $\begin{array}{c} 1 \\ 2 \\ \hline \end{array}$   $\leftarrow \frac{7}{3} = \frac{16}{8} = \frac{128}{8}$

No. of required decoders:  $1 = \frac{2}{8}$ ,  $2 = \frac{16}{8}$ ,  $16 = \frac{128}{8}$   $\leftarrow$

Total no. of decoder required =  $16 + 2 + 1 = 19$

Hence 19 no. of  $3 \times 8$  decoder required to  
design  $7 \times 128$  decoders.

Q11 How many  $2 \times 4$  decoders are required  
to construct one  $3 \times 8$  decoder?

soln

$2 \times 4$        $3 \times 8$

No. of levels =  $\frac{3}{2} \approx 2$

$$\frac{3}{2} \approx 2$$

level :  $\begin{array}{c} 1 \\ 2 \\ \hline \end{array}$   $\leftarrow \frac{3}{2} = \frac{9}{4}$

No. of decoder reqd:  $1 = \frac{9}{4}$ ,  $2 = \frac{8}{4}$   $\leftarrow \frac{2}{4} = \frac{1}{2}$

Hence total no. of decoder reqd =  $2 + 1 = 3$

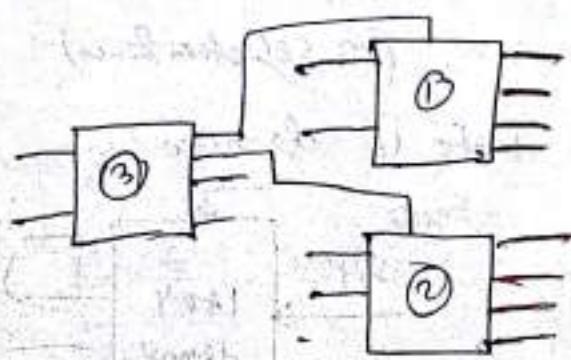
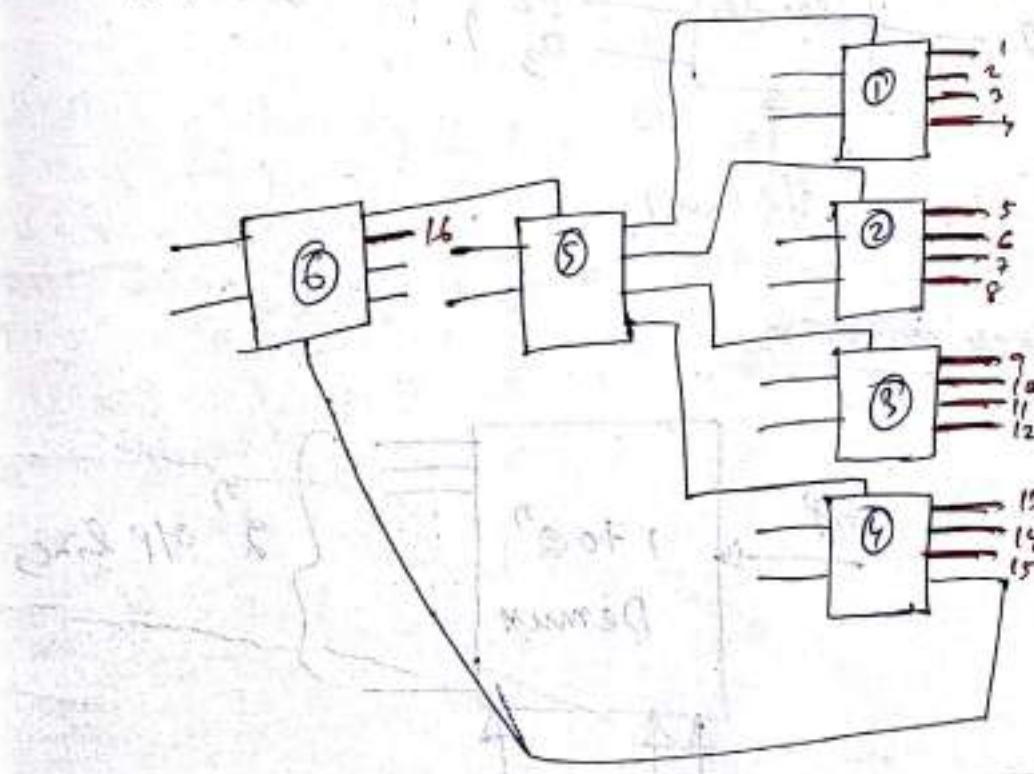
Q11 How many 2 to 4 decoders are required to construct a 4 to 16 decoder.

Soln no. of levels =  $\frac{4}{2} = 2$ .  $\frac{4}{2} = 2$

levels 1 2  
 $\frac{16}{4}$  1 =  $\frac{16}{4}$ , 4 =  $\frac{16}{4}$  1 =  $\frac{16}{4}$

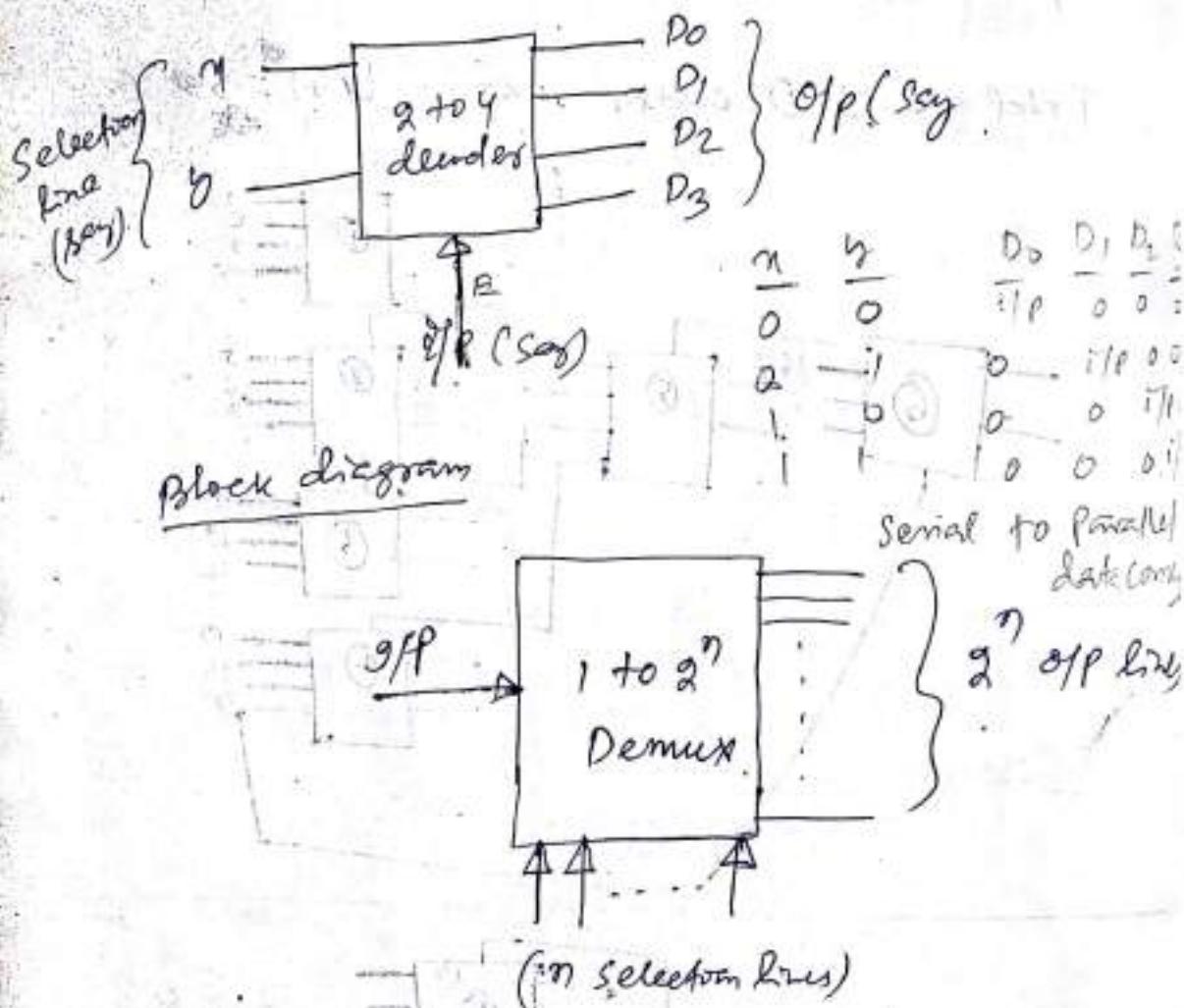
No. of decoders required = 1 =  $\frac{4}{4}$ , 4 =  $\frac{16}{4}$

Total no. of Decoder req =  $4 + 1 = 5$  (Ans)



## Demultiplexer (Demux)

- A Decoder with enable as i/p is known as demultiplexer.
- A Demultiplexer is a ckt. that receives information on a single line & transmits that information to one of the  $2^n$  output lines based on n' selection lines.



→ Enable decides if it is demux or high impedance state

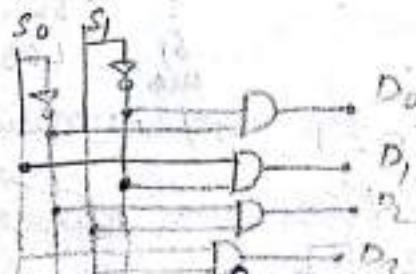
T. T

for demux, $\leftarrow I/P$	$S_1$	$S_0$	$D_0$	$D_1$	$D_2$	$D_3$
from decoder	<u><math>E_n</math></u>	$n$	$o$	$o$	$o$	$o$
	0	0	0	0	0	0
	0	0	01	0	0	0
	0	1	0	0	0	0
	0	1	1	0	0	0
	0	1	0	1	0	0
	1	0	0	0	1	0
	1	0	1	0	0	0
	1	1	0	0	0	1
	1	1	1	0	0	1

$\rightarrow S_0$  Enable  $S_1$   $S_0$   $D_0$   $D_1$   $D_2$   $D_3$

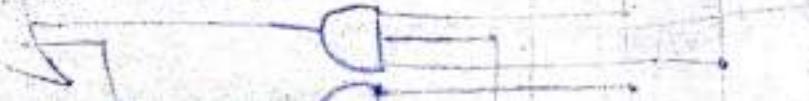
wence demux  $S_1$   
are designed by  
only  $W_L$  or  
 $A_{ND}$  gates.

0	0	$I/P$	0	0	0	$D_0 = S_0 S_1$
0	1	0	$I/P$	0	0	$D_1 = S_1 S_0$
1	0	0	0	0	$I/P$	$D_2 = S_0 S_1$
1	1	0	0	0	$I/P$	$D_3 = S_0 S_1$



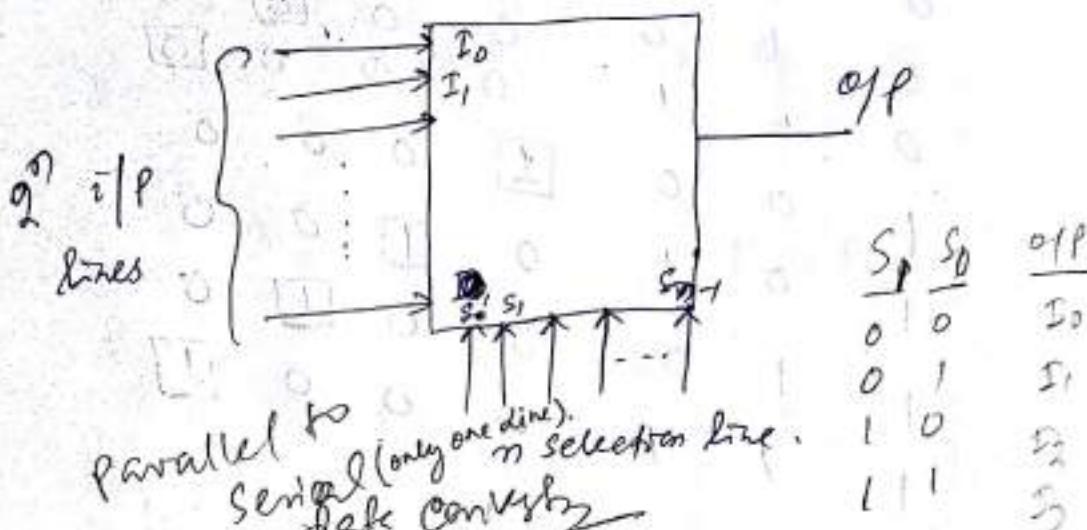
APP1

→ Demux is used for serial to parallel data converters.

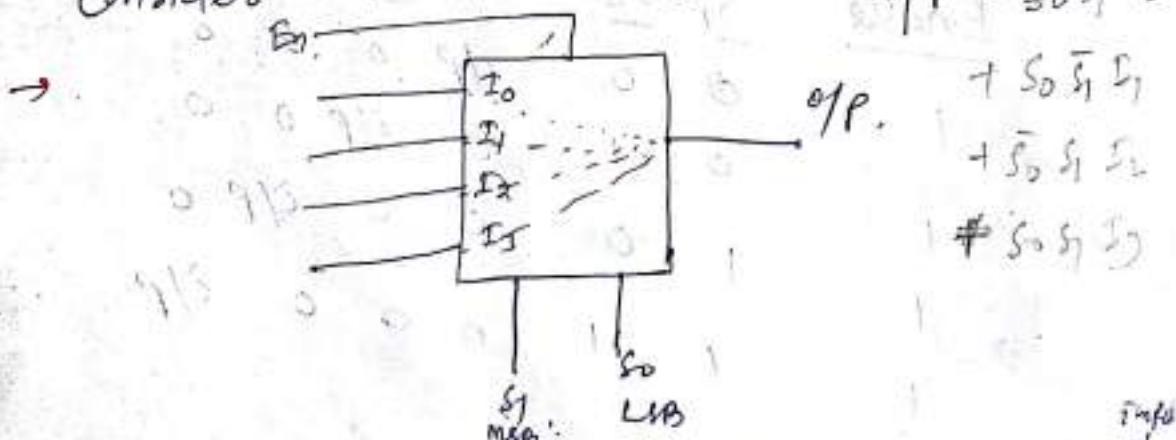


## Multiplexers

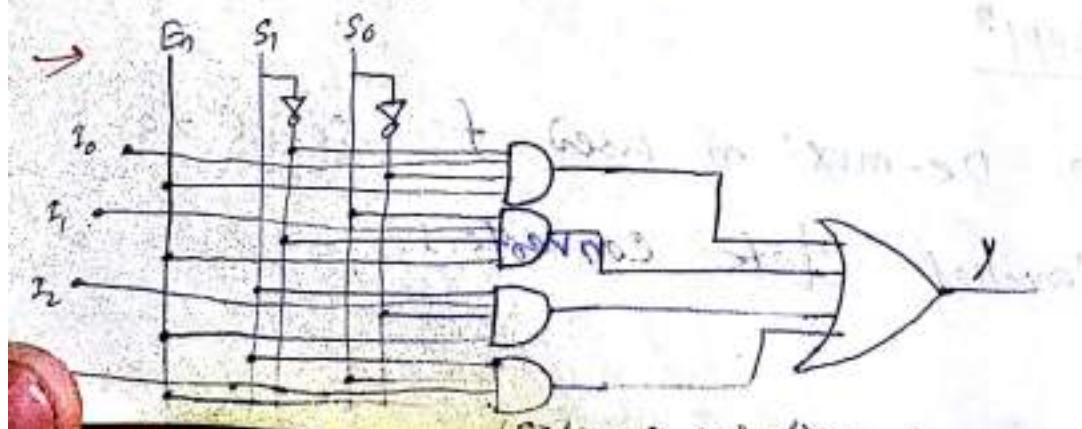
→ A multiplexer selects binary information from one of the  $2^n$  i/p lines and directs it on a single o/p line. The selection of a particular line is controlled by a set of selection lines.



Consider a  $4 \times 1$  mux,



When  $E=1$ , it is mux & when  $E=0$ , high  $q_M$



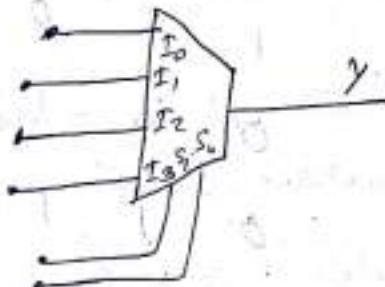
→ T · T

En	$S_1$	$S_0$	$Y$
1	0	0	$I_0$
	0	1	$I_1$
1	1	0	$I_2$
1	1	1	$I_3$
0	x	x	0

→ characteristics eqn<sup>n</sup>

$$Y = \bar{S}_0 \bar{S}_1 I_0 + S_0 \bar{S}_1 I_1 + \bar{S}_0 S_1 I_2 + S_0 S_1 I_3.$$

→ Standard logic symbol.



⇒ mux are also referred as data selector.  
i.e. select the data from 2<sup>0</sup>:1P to  
0P side.

### Applications

- multiplexers are used for parallel to serial conversion operation.
- Muxes are used to implement the Boolean fun's.

Q11 Simplify the Boolean function  
 $f(x_1, y, z) = \sum m(0, 2, 4, 5)$  using maps.

Sol:  $x_1, y, z \rightarrow$  are selection lines for m<sub>0</sub>

so i/p lines be  $= 2^3 = 8$  lines



<u>m</u>	<u>y</u>	<u>z</u>	<u>f</u>
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

8-to-1 m<sub>0</sub>

$f = \overline{x_1} \cdot \overline{y} \cdot \overline{z} + x_1 \cdot \overline{y} \cdot z + x_1 \cdot y \cdot \overline{z} + x_1 \cdot y \cdot z$

Q11 Simplify  $f(x_1, y, z) = \sum m(0, 2, 4, 5)$ , using 4:1

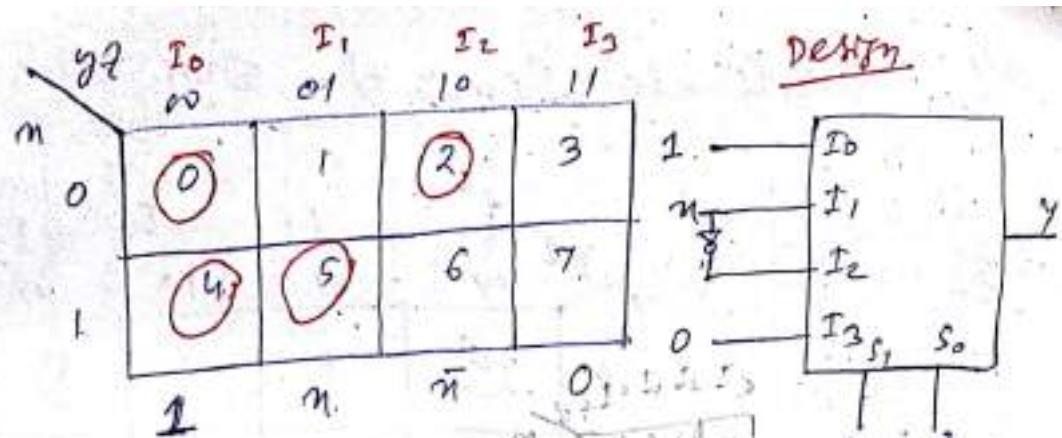
<u>m</u>	<u>y</u>	<u>z</u>	<u>f</u>
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Sol:  $x_1$  is the selection line.

$m=0 \left\{ \begin{array}{l} I_0 \\ I_1 \\ I_2 \\ I_3 \end{array} \right.$

$m=1 \left\{ \begin{array}{l} I_0 \\ I_1 \\ I_2 \\ I_3 \end{array} \right.$

A 4x4 grid representing a 4-to-1 decoder. The columns are labeled x1, y, z. The rows are labeled I0, I1, I2, I3. The output f is connected to I3.



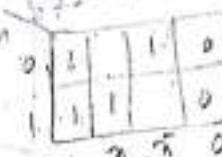
$$y = y_2 \quad s_0 = 2 \quad \text{and} \quad n = 2/p.$$

When  $y = 0, z = 0, I_0 = 0$

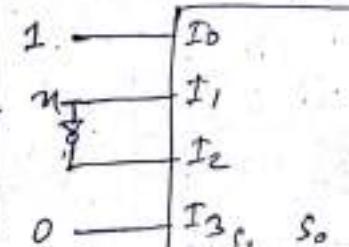
$y = 0, z = 1, I_1 = 1$

$y = 1, z = 0, I_2 = 1$

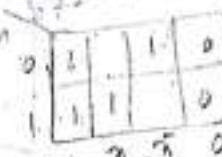
$y = 1, z = 1, I_3 = 0$



Design



y



Sum

$I_0 \text{ is selected}$

n

1

0

1

0

→ for  $y = 0, z = 0, \text{Op} = \underline{0} = I_0 \text{ when } n = 0$

$y = 0, z = 0, \text{Op} = \underline{1} = I_0 \text{ when } n = 1$

so  $I_0 = 1 \text{ independent of } n$

→ for  $y = 0, z = 1, \text{Op} = \underline{0} = I_1 \text{ when } n = 0$

$y = 0, z = 1, \text{Op} = \underline{1} = I_1 \text{ when } n = 1$

so  $I_1 = n$ .

→ for  $y = 1, z = 0, \text{Op} = \underline{1} = I_2 \text{ when } n = 0$

$y = 1, z = 0, \text{Op} = \underline{0} = I_2 \text{ when } n = 1$

so  $I_2 = \bar{n}$ .

→ for  $y = 1, z = 1, \text{Op} = \underline{0} = I_3 \text{ when } n = 0$

$y = 1, z = 1, \text{Op} = \underline{0} = I_3 \text{ when } n = 1$

so  $I_3 = 0 \text{ independent of } n$

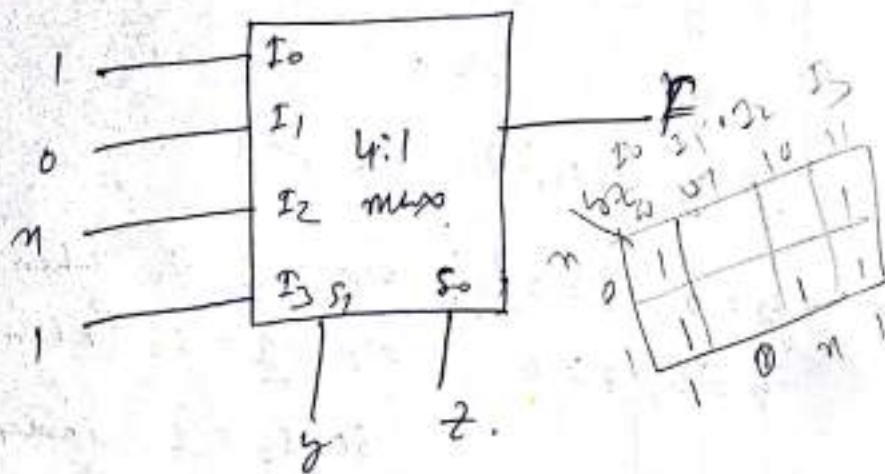
→ if both are circled put 1 → if 1st one is uncircled  
if 2nd one is circled put 0

Q11. Implement  $f(m, n, z) = \sum m(0, 3, 4, 6, 7) u_z$

Sol:

	$y_2$	$m$	$I_0$	$I_1$	$I_2$	$I_3$
0	0	0	0	1	2	3
1	1	5	6	7		

~~$m \rightarrow i/p$~~ ,  $y_2, z \rightarrow$  Selection line.



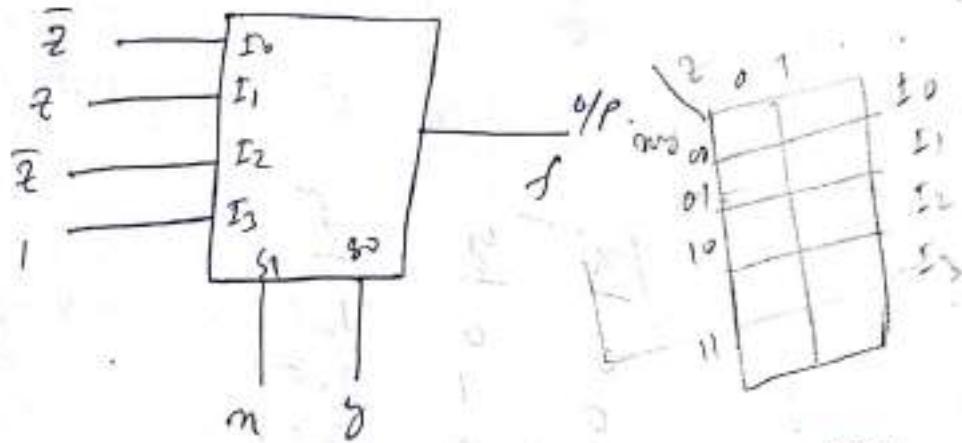
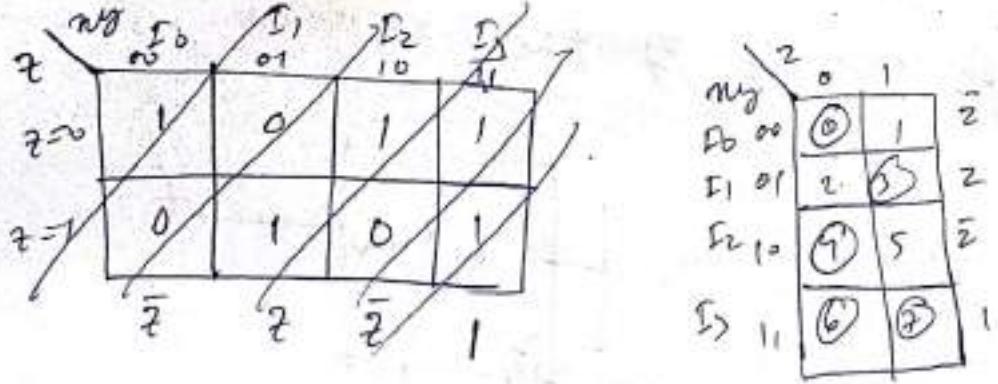
Q11. Implement the above func' using 4x1 multiplexers

using one as Selection lines of  $z$  as i/p

selecting i/p line

Soln:

$m$	$y$	$z$	$f$
0	0	0	0 $\rightarrow D_0$
0	0	1	1 $\rightarrow D_1$
0	1	0	0 $\rightarrow D_2$
0	1	1	1 $\rightarrow D_3$
1	0	0	1 $\rightarrow D_4$
1	0	1	0 $\rightarrow D_5$
1	1	0	1 $\rightarrow D_6$
1	1	1	0 $\rightarrow D_7$



Q11 Implement the F.A. *at using  $4 \times 1$  mao.*

Soln.  $m \rightarrow$  if p,  $n, z \rightarrow$  selection line.

$$c(m, n, z) = \Sigma m(3, 5, 6, 7)$$

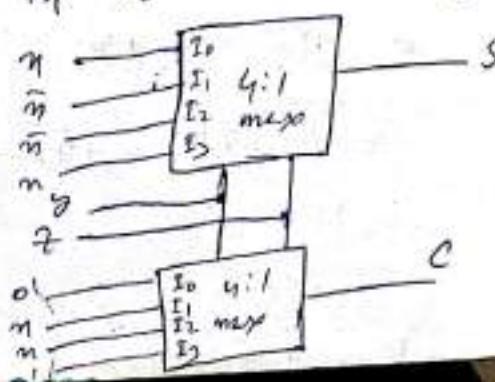
$$s(m, n, z) = \Sigma m(1, 2, 4, 7)$$

for carry

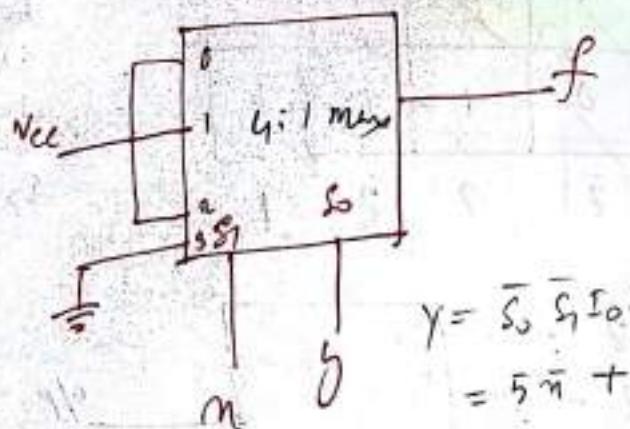
	$I_0$	$I_1$	$I_2$	$I_3$
$m$	0	1	2	③
$n$	4	⑤	⑥	⑦

for sum

	$I_0$	$I_1$	$I_2$	$I_3$
$m$	0	①	②	3
$n$	④	5	6	⑦



Q11. Find the off of the  $4 \times 1$  max, shown in fig.



$$y = \bar{S}_0 \bar{S}_1 I_0 + \bar{S}_0 \bar{S}_2 I_1 + \bar{S}_0 \bar{S}_3 I_2 + S_1 S_2 S_3$$

$$= \bar{n} \bar{s} + \bar{n} \bar{s} + \bar{n} s$$

Sol

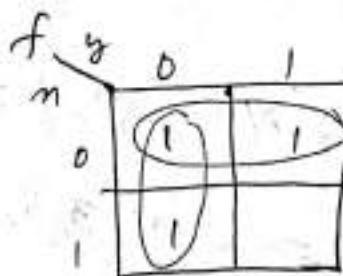
<u>n</u>	<u>y</u>	<u>f</u>
0	0	$I_0 = 1$
0	1	$I_1 = 1$
0	0	$I_2 = 1$
1	1	$I_3 = 0$

$$= \bar{n} + n \bar{s}$$

$$= (\bar{n} + n)(\bar{n} + \bar{s})$$

$$= \bar{n} + \bar{s}$$

Ans



$$f = \bar{n} + \bar{s}$$

By characteristic eqn method.

$$\text{OR } f = \bar{S}_0 \bar{S}_1 I_0 + \bar{S}_0 \bar{S}_2 I_1 + \bar{S}_0 \bar{S}_3 I_2 + S_1 S_2 S_3$$

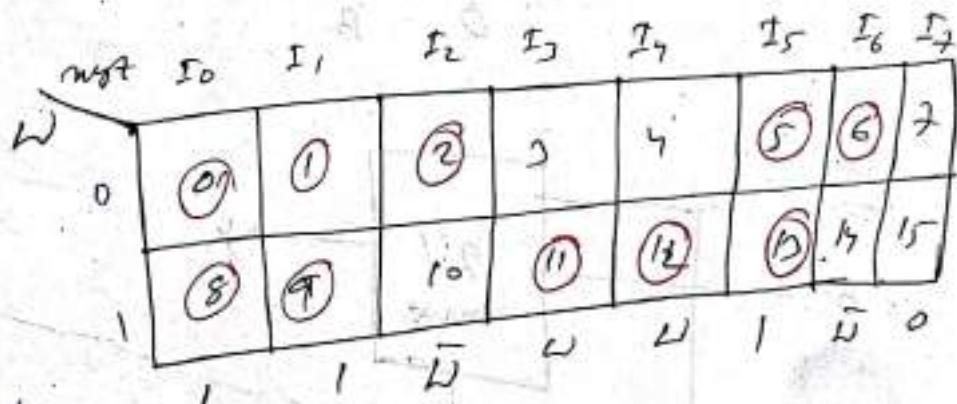
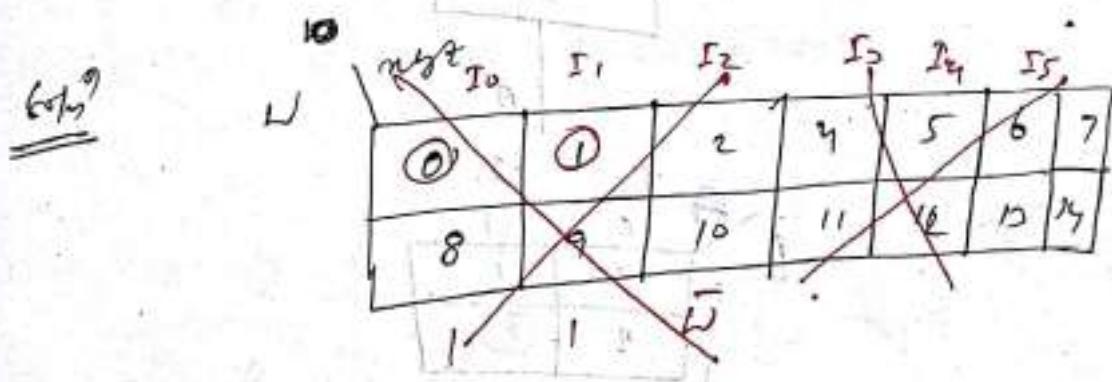
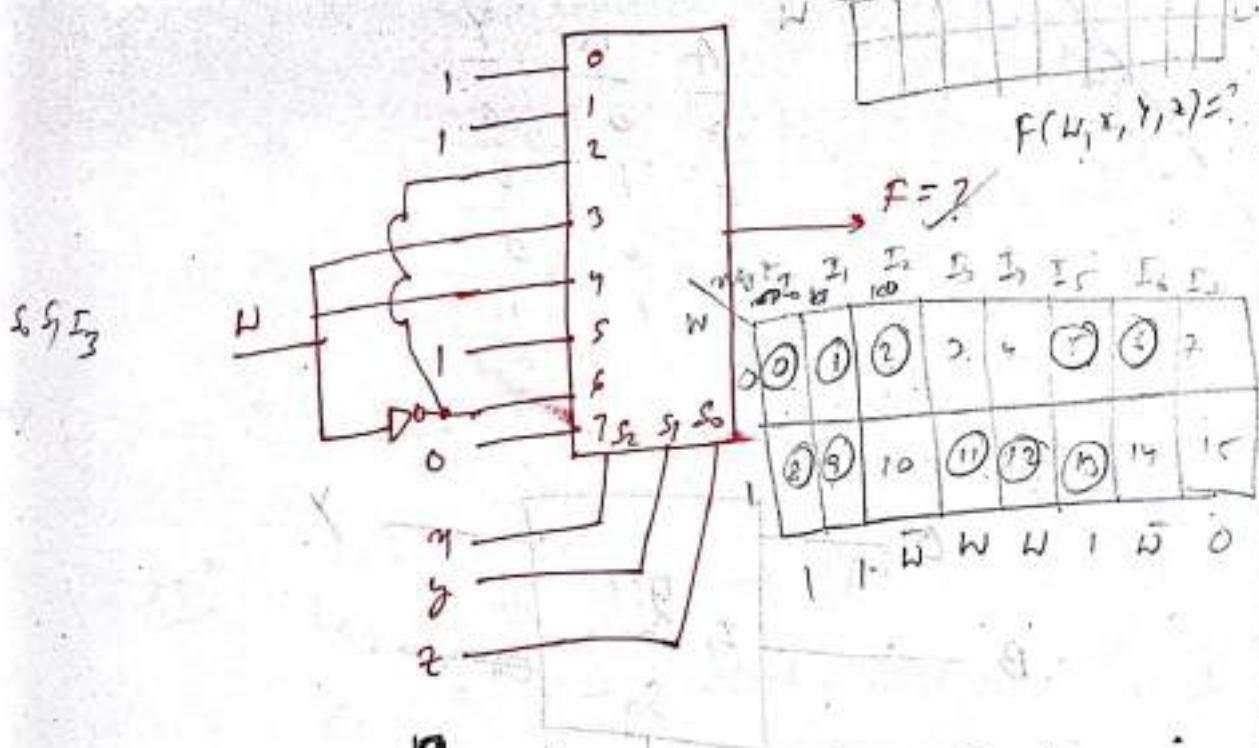
$$= \bar{n} \bar{s} + \bar{n} \bar{s} + \bar{n} s + n s$$

$$= (\bar{n} + n)(\bar{s} + s)$$

Ans



off the off of the  $8 \times 1$  max as shown below



$\cancel{\text{First}} \rightarrow$

$$F(u, x, y, z) = \sum_m (0, 1, 2, 3, 4, 5, 6, 7)$$

$\cancel{\text{max } I_0, I_1, I_2, I_3, I_4, I_5, I_6, I_7} \rightarrow 3, 12, 13) \text{ Ans.}$

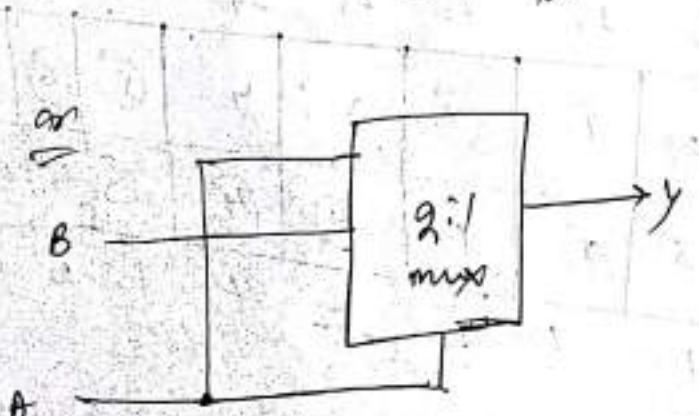
Q11 Implement 2:1 P and get LUTs 2x1 m

Ans

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



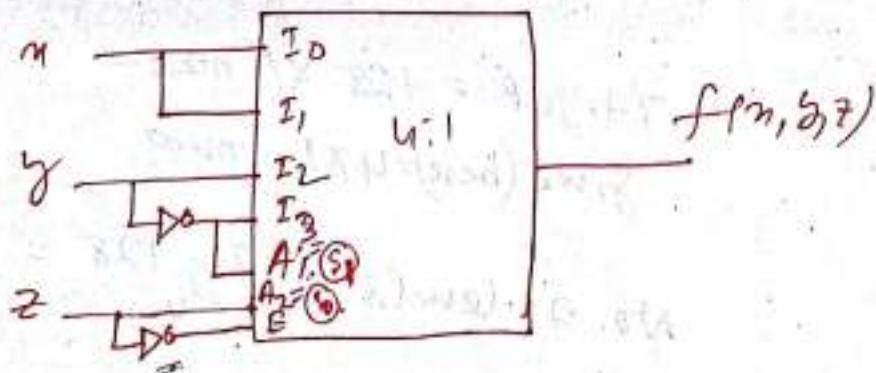
A		$\Sigma_0$	$\Sigma_1$
B		0	1
0	1	0	1
1	0	2	3



$$\begin{aligned}
 Y &= \bar{S}_0 \bar{S}_1 I_0 + S_0 \bar{S}_1 I_1 + \bar{S}_0 S_1 I_2 + S_0 S_1 I_3 \\
 &= \bar{S}_0 \bar{S}_1 I_0 + \cancel{\bar{S}_0 S_1 I_1} + \cancel{S_0 \bar{S}_1 I_2} + \cancel{S_0 S_1 I_3} \\
 D &= S_0 I_1 + \cancel{S_0 \bar{S}_1} \\
 &= S_0 I_1
 \end{aligned}$$

Q1 Consider the following max the fun<sup>n</sup>

$f(n, y, z)$  implemented by ~~by~~ - is



Sol<sup>n</sup>  $I_0, I_1, I_2, I_3 \rightarrow$  g/p/s.

$A_1, A_2 \rightarrow$  selecting line.

$E \rightarrow$  enable lines.

$n$	$y$	$z$	$f$
0	0	0	0
0	0	1	x
0	1	0	0
0	1	1	x
0	1	0	0
1	0	0	x
1	0	1	x
1	1	0	1
1	1	1	x



or  $f = \bar{s}_0 \bar{s}_1 I_0 + s_0 \bar{s}_1 I_1 + \bar{s}_0 s_1 I_2 + s_0 s_1 I_3$ .

$$= \bar{s}_0 \bar{s}_1 n + s_0 \bar{s}_1 n + \bar{s}_0 s_1 m + s_0 s_1 m$$
$$= \cancel{m} \bar{s}_0 \bar{s}_1 n + \cancel{m} s_0 \bar{s}_1 n + \cancel{m} \bar{s}_0 s_1 + \cancel{m} s_0 s_1$$
$$= \cancel{m} \cancel{n} + \cancel{m} \cancel{n} = \cancel{m} \cancel{n}$$

$\oplus$  How many  $4 \times 1$  mixers are required to construct  $128 \times 1$  mixer.

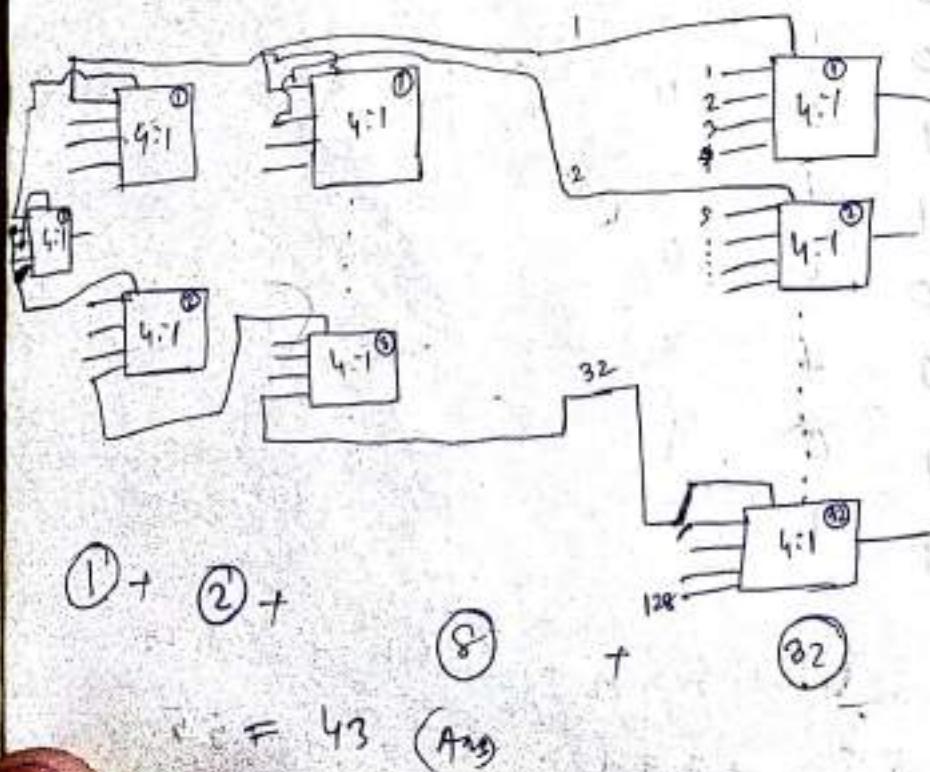
Ans Target =  $128 \times 1$  mixer  
Given (base) =  $4 \times 1$  mixer.

$$\text{No. of levels} = \log_4 128 = 3.5 \approx 4$$

levels: 1 2 3 4

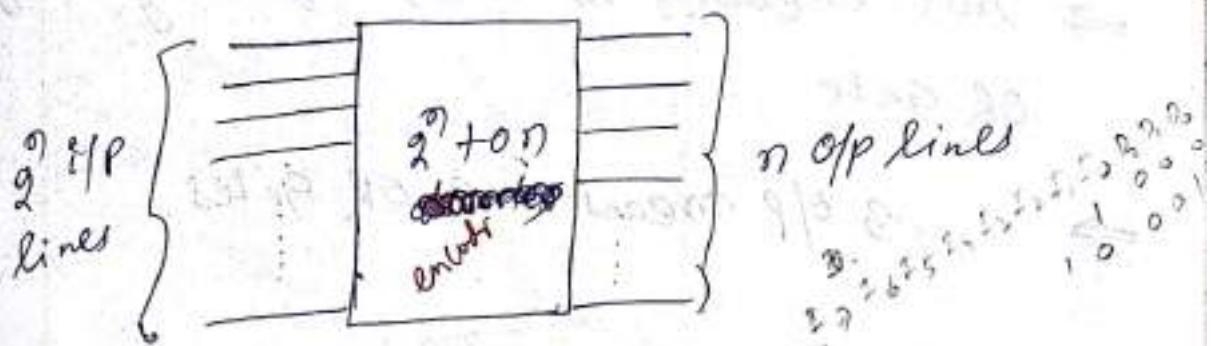
No. of mixers	$\frac{128}{4^1}$	$\frac{128}{4^2}$	$\frac{128}{4^3}$	$\frac{128}{4^4}$
$= \sum \frac{128}{4^k}$	32	8	2	1
	"	"	"	"
	32	8	2	1

$$\text{Total no. of mixers} = 32 + 8 + 2 + 1 = 43$$

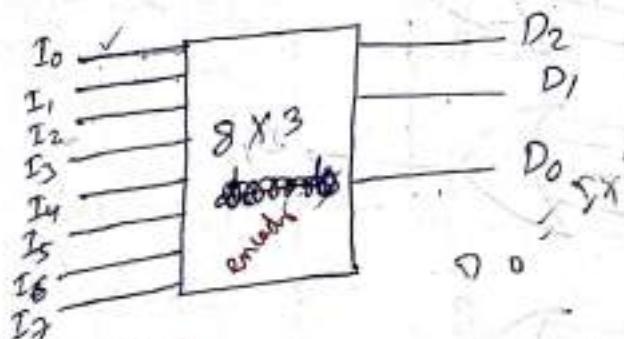


## Encoders

An encoder has  $2^n$  of i/p lines of which only one of which is the active high state. & an n-bit code is generated upon which of the i/p's is excited..



Example



(Octal to Binary)

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

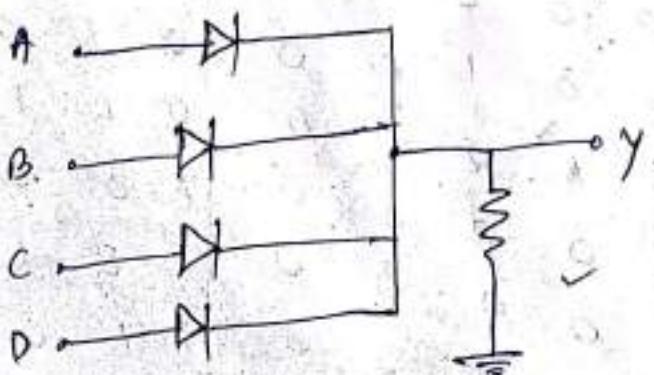
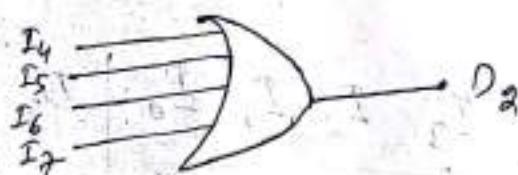
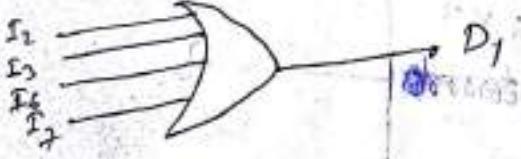
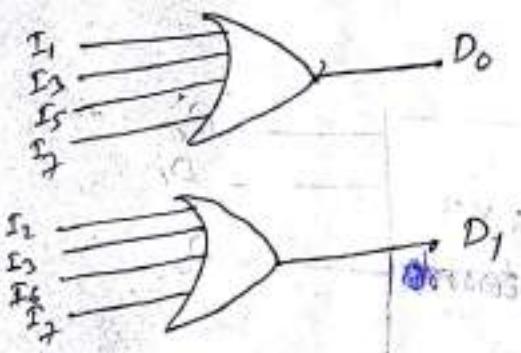
$$D_0 = I_1 + I_3 + \cancel{I_4} + I_5 + I_7$$

$$D_1 = I_2 + I_3 + I_6 + I_7$$

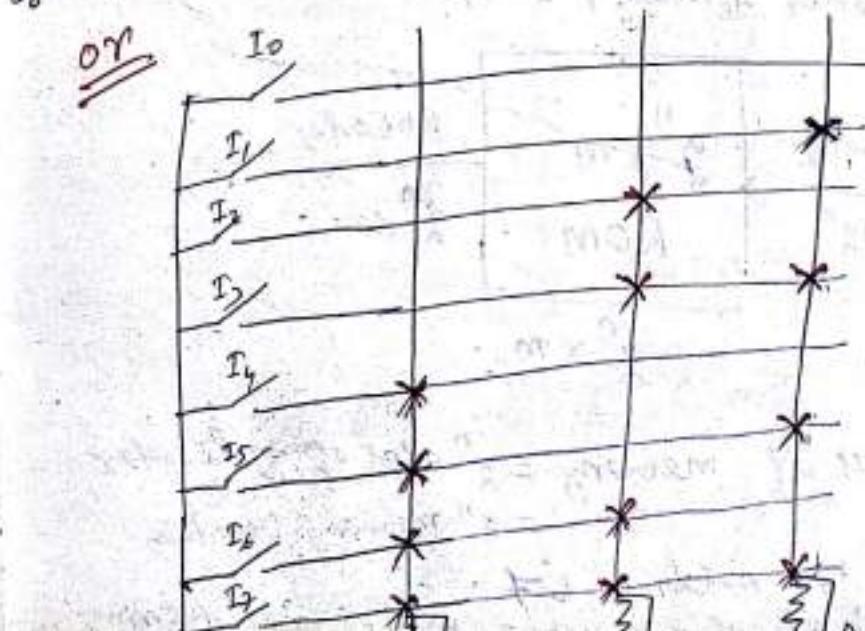
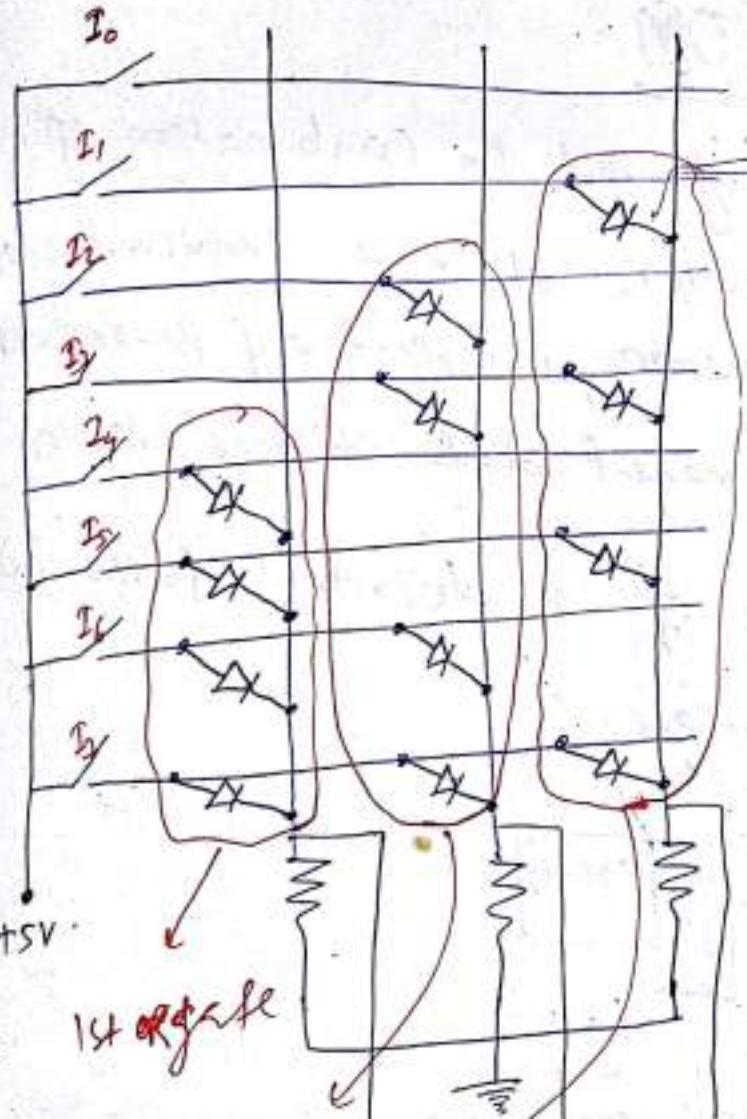
$$D_2 = I_4 + I_5 + I_6 + I_7$$

→ This encoder is design by using only OR gate

⇒ 3 opp means = 3 OR gates.



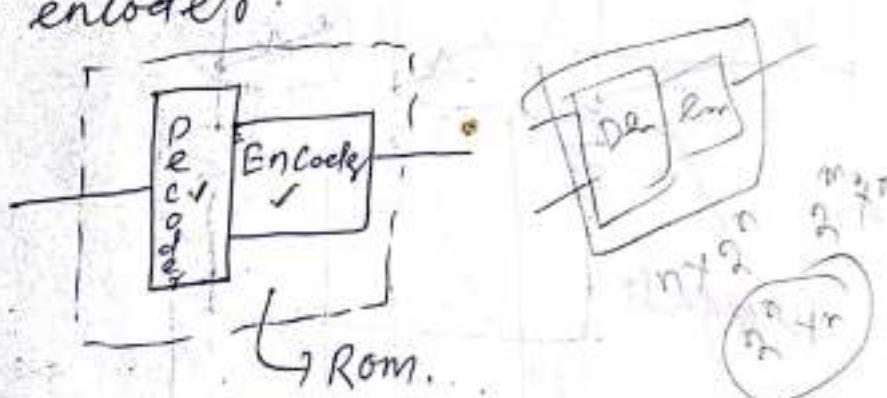
(4 input OR gate)



# ROM

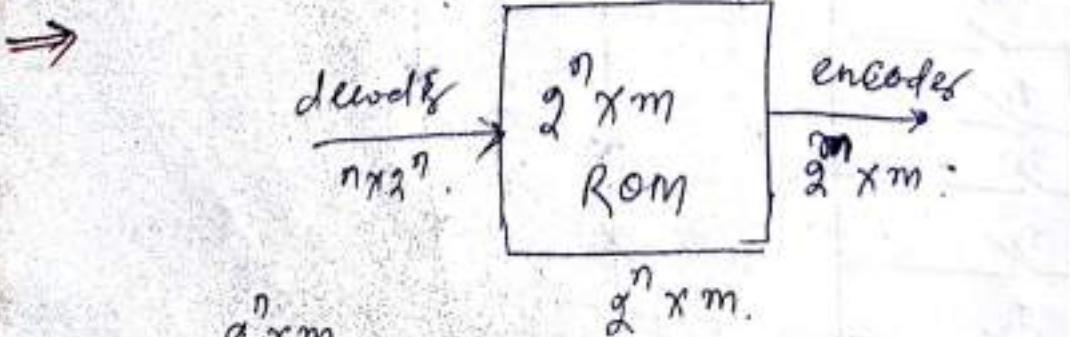
⇒ ROM is nothing but the combination of decoder & encoder. It is a semiconductor non-volatile memory, irrespective of power supply. It is a permanent data storage device.

⇒ ROM consists of a decoder followed by an encoder.



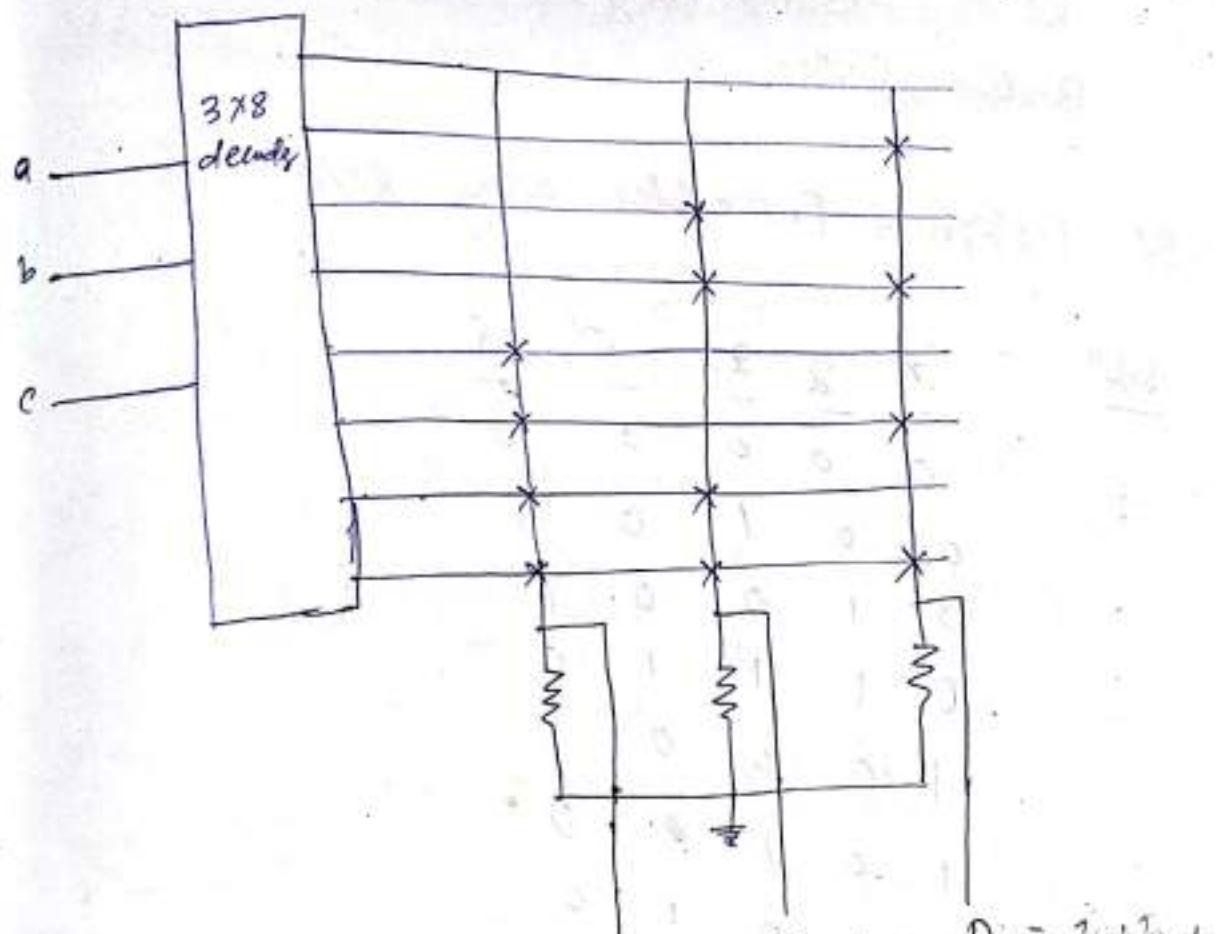
AND  $\leftarrow$  I/P. OR  $\rightarrow$  O/P.

⇒ A ROM consists of AND gates at one side & OR gates at other side.  
 Decoder + Encoder = ROM.



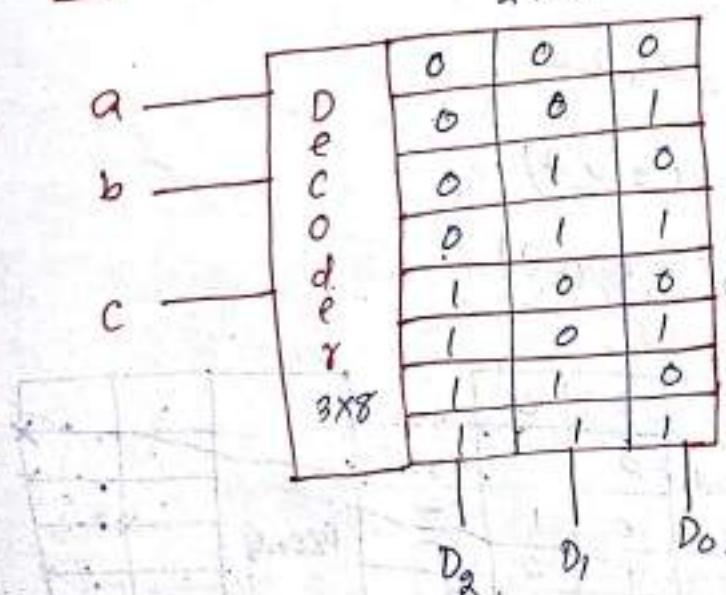
$2^n$  = Depth of memory =  $2^n$  ops of a decoder  
 $= 2^n$  memory location

$n$  = no. of address bits



$$D_2 = \overline{a} \cdot \overline{b} \cdot \overline{c} \\ D_1 = \overline{a} \cdot \overline{b} \cdot c + \overline{a} \cdot b \cdot \overline{c} + a \cdot \overline{b} \cdot \overline{c} \\ D_0 = a \cdot b \cdot c$$

OR



no link = 0  
link = 1.

Size of Rom. =  $2^3 \times 3$ .

$2^3$  = output of decoder

3 = output of encoder

size of Rom. =  $2^3 \times 3$ .

⇒ ROMs are used to implements the Boolean fun's.

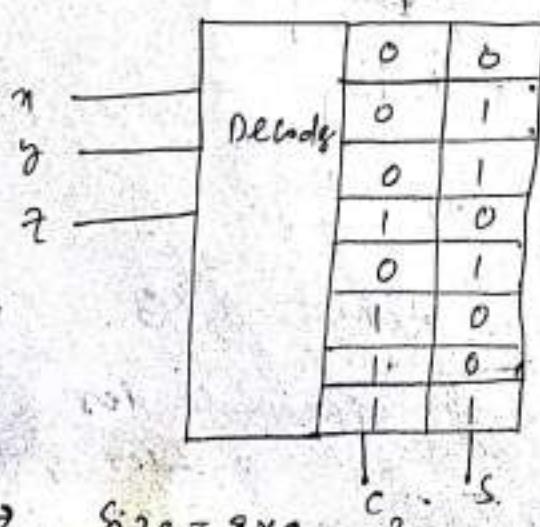
Q) Design a Full adder using ROM.

<u>Sols</u>	<u>n</u>	<u>y</u>	<u>z</u>	<u>C'</u>	<u>S</u>
	0	0	0	0	0
0	0	1	0	1	
0	1	0	0	1	
0	1	1	1	0	
1	0	0	0	1	
1	0	1	0	0	
1	1	0	1	0	
1	1	1	1	1	

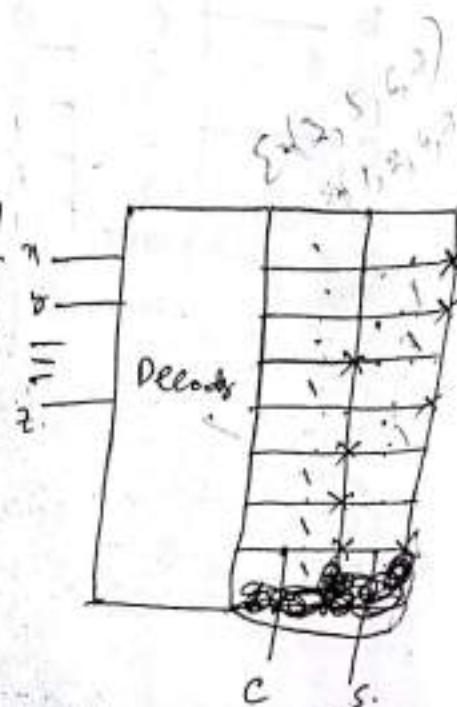
$$\sum m(n, y, z) = 1, 2, 4$$

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

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



$$\rightarrow \text{Size} = 8 \times 2$$



Q11 What is the size of ROM required to implement following set of Boolean fun?

$$f_1(x_1, x_2, x_3) = \Sigma m(1, 2, 4, 7)$$

$$f_2(x_1, x_2, x_3) = \Sigma m(3, 5, 6, 7)$$

$$f_3(x_1, x_2, x_3) = \Sigma m(0, 2, 4, 7)$$

Sol

$x_1 x_2 x_3$	0	0	1
d	0	-	0
e	0	1	1
c	1	0	0
o	0	1	-
d	1	0	0
e	1	-	0
r	1	1	1

$$\begin{aligned} f_1 &= \Sigma m(3, 5, 6, 7) \\ f_2 &= \Sigma m(1, 2, 4, 7) \\ f_3 &= \Sigma m(0, 2, 4, 7) \end{aligned}$$

$$\text{Size} = 2^3 \times 3$$

Q12 What is the min<sup>n</sup> size of ROM required to implement following set of eq's.

$$f_1(x_1, x_2, x_3) = \Sigma m(1, 2, 4, 7), f_2(x_1, x_2, x_3) = \Sigma m(3, 5, 6, 7)$$

$$f_3(x_1, x_2, x_3) = \Sigma m(0, 2, 4, 7), f_4(x_1, x_2, x_3) = \Sigma m(0, 1, 3, 4, 5, 6, 7)$$

$x_1 x_2 x_3$	0	0	1
d	0	0	1
e	0	1	0
c	0	1	1
o	1	0	0
d	0	1	1
e	1	0	0
r	1	0	0
r	1	1	1

$$\begin{aligned} V_{CC} &= \text{logic 1} \\ f_4 &\Rightarrow \text{Size} = 2^3 \times 3 \end{aligned}$$

Q11 Implement the following Boolean func.  
by using a Rom.

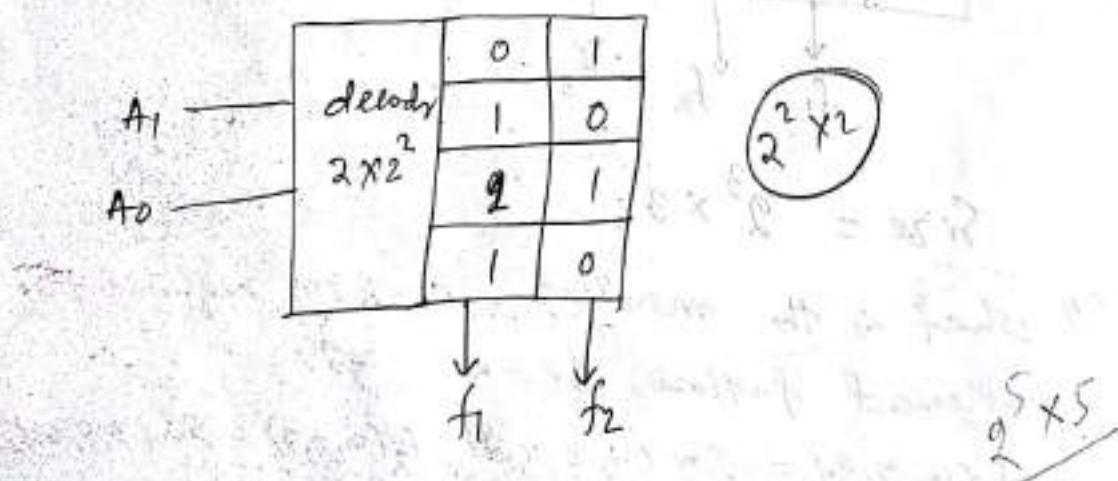
$$f_1(A_1, A_0) = \Sigma m(1, 3, 5)$$

$$f_2(A_1, A_0) = \Sigma m(0, 2)$$

Sol → The variable ~~of~~ of Boolean func's represent the ips for the Rom.

→ The no. of Boolean func's represents the no. of ips of the Rom.

$$\text{so size of Rom} = 2^2 \times 2$$



Q11 A ROM is used to implement 5 Boolean func's each with 5 variables; (Let it be 5)

Sol size of decoders =  $5 \times 2^5 = 5 \times 32$ .

size of encoder =  $2^5 \times 5$ .

size of Rom =  $2^5 \times 5 = 32 \times 5$  ROM

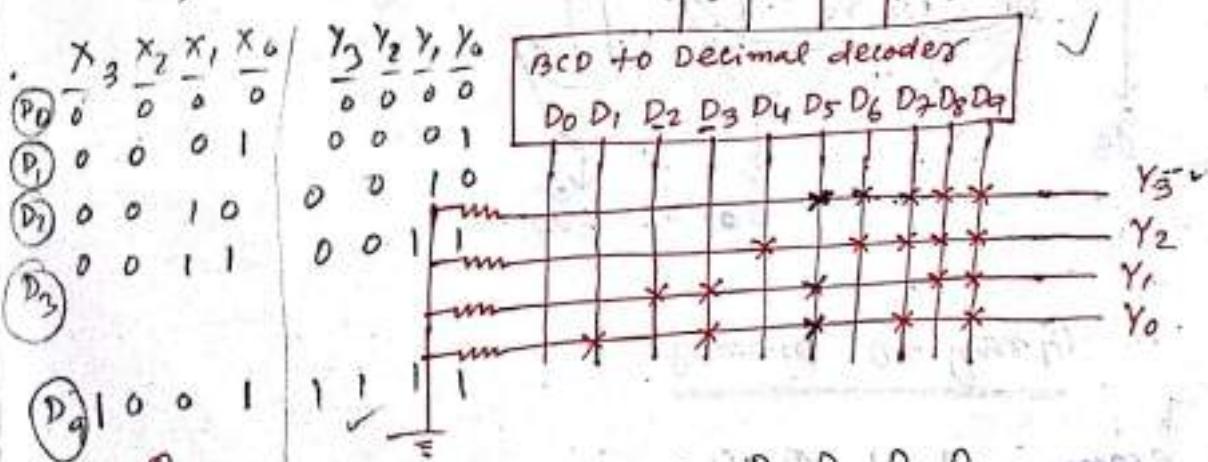
Q1 How many  $128 \times 8$  bits Rom requires to design  $1M \times 8$  bits Rom.

$$\text{Ans} \quad \text{Total no. of } 128 \times 8 \text{ bits Rom Required} = \frac{1M \times 8}{128 \times 8}$$

$$= \frac{1024 \times 1024}{128} \times \frac{8}{8} = \frac{2^{20}}{2^7} = 2^{13} \quad \checkmark$$

Q1 If inputs  $x_3 x_2 x_1 x_0$  to Rom shown in the fig. are 8421 BCD numbers then the outputs  $y_3 y_2 y_1 y_0$  are.

- a) Gray code no. b) Excess-3 code no.  
 c) 2421 code no. d) 6421 code no.



$D_0 \rightarrow$	0	0	0	1	0	0	0	0	1
$D_1 \rightarrow$	0	0	1	0	0	0	0	1	0
$D_2 \rightarrow$	0	0	1	1	0	0	1	1	1
$D_3 \rightarrow$	0	0	1	0	0	0	1	0	0
$D_4 \rightarrow$	0	1	0	0	1	0	0	1	1
$D_5 \rightarrow$	0	1	0	1	1	0	1	0	0
$D_6 \rightarrow$	0	1	1	0	1	1	0	0	1
$D_7 \rightarrow$	0	1	1	1	1	1	1	1	0
$D_8 \rightarrow$	0	1	1	1	1	1	1	1	1
$D_9 \rightarrow$	1	0	0	1	1	1	1	1	1

2421 code no.  $\textcircled{C} \text{ (Ans)}$

# Code converters

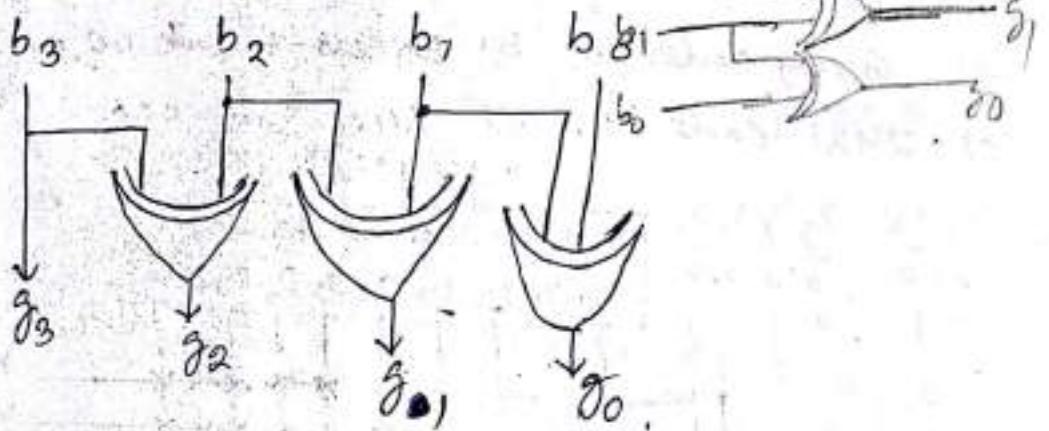
## Binary to Gray Code converter

binary  $\rightarrow$  1. 0. 1. 1.  
 gray  $\rightarrow$  1 1 1 0.

B	1 0 1 0 0 0 1
G	1 0 1 1

$$g_n = b_n$$

$$g_i = b_i \oplus b_{i+1}$$



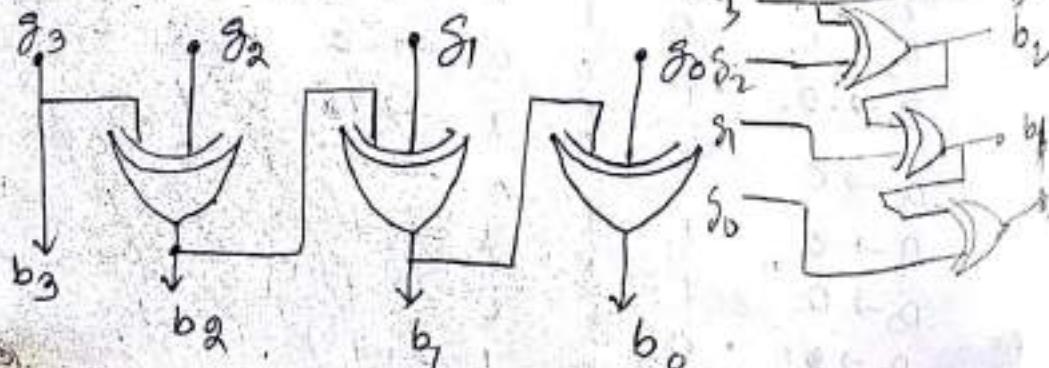
## Gray to binary

gray      1 1 0 0  
 binary    1 0 0 0

<u>G - B</u>	
1	0
1	0

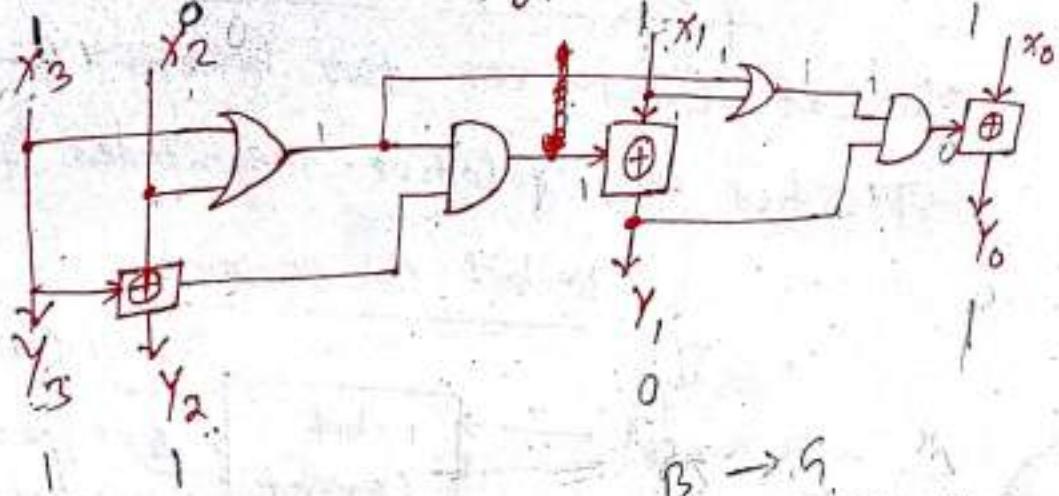
$$b_n = g_n$$

$$b_i = g_i \oplus b_{i+1}$$



(Q1)

The circ. shown in fig. converts

Solu<sup>n</sup>

$$y_3 = x_3$$

$$y_2 = x_3 \oplus x_2 = y_3 \oplus x_2$$

$$y_1 = x_3 \oplus x_2 \oplus x_1 = y_2 \oplus x_1$$

$$y_0 = [(x_1 + x_2 + x_3)(x_3 \oplus x_2 \oplus x_1)] \oplus x_0$$

$$= [x_3 \oplus x_2 \oplus x_1] \oplus x_0$$

$$= y_1 \oplus x_0$$

$$y_3 = x_3$$

$$y_2 = y_3 \oplus x_2$$

$$y_1 = y_2 \oplus x_1$$

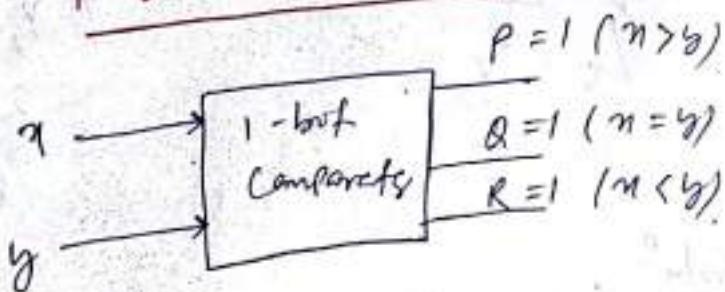
$$y_0 = y_1 \oplus x_0$$

Gray code to Binary code (Ans) //

## Magnitude Comparison

This cat. Compares two binary numbers & specifies the relative magnitudes of these no.

## 1-bit Comparator



<u>T</u>	<u>T</u>	$m^{70}$	$m^7$	$m^{20}$	<u>n</u>	<u>y</u>	<u>P</u>	<u>R</u>
<u>n</u>	<u>y</u>	<u>P</u>	<u>Q</u>	<u>R</u>	<u>n</u>	<u>y</u>	<u>P</u>	<u>R</u>
0	0	0	1	0	0	0	0	0
0	1	0	0	1	0	1	0	1
0	1	1	0	0	1	1	1	0
1	0	0	1	0				
1	1							

$$P = ny^!, \quad Q = n'y^! + ny = n(Oy)^!, \quad R = n'y.$$

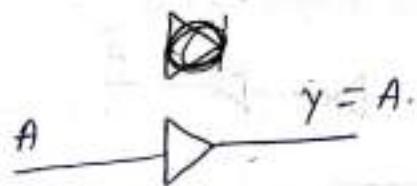
a bit mag. compass for



<u>N</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>P</u>	<u>Q</u>	<u>R</u>
0	0	0	0 → 0	0	1	0
0	0	0	1 → 0	0	0	1
0	0	1	0 → 0	0	1	
0	0	1	1 → 0	0	0	1
0	1	0	0 → 1	0	0	
1	1	0	1 → 0	1	0	
1	1	1	0 → 0	0	1	

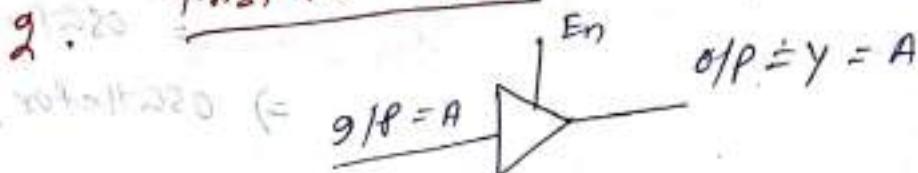
## Some Special Gates

### Buffer:



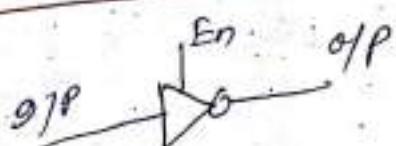
<u>T.</u>	<u>A</u>	<u>Y</u>	<u>o/p eqs?</u>
0	0	0	$y = A$
1	1	1	

### Tristate Buffer:

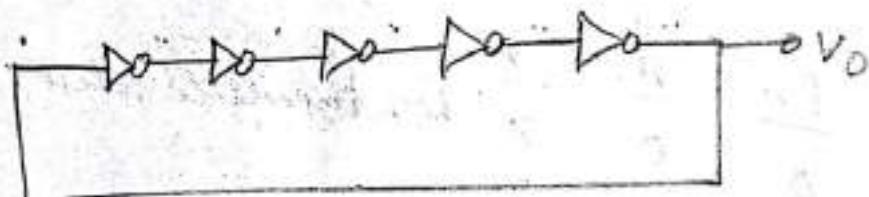
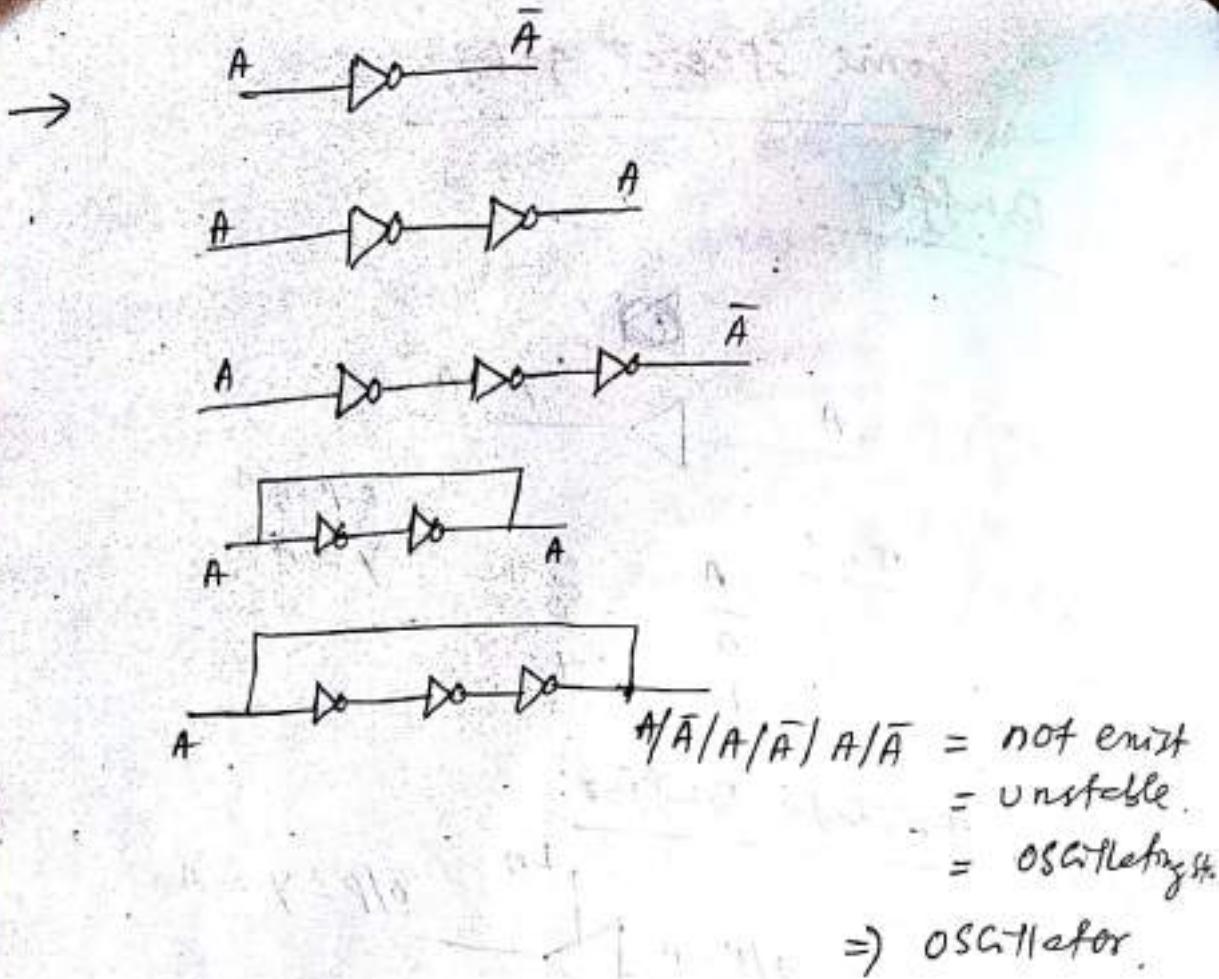


<u>En</u>	<u>A</u>	<u>Y</u>	<u><math>Z = \text{high impedance state}</math></u>
0	0	$Z$	
0	1	$Z$	
0	0	0	
1	1	1	
1	1	$Z$	

### Tristate inverter:



<u>En</u>	<u>o/p</u>	<u>o/p</u>
0	0	$Z$
0	1	$Z$
0	0	1
1	1	0
1	1	$Z$



p.d of each inverter  $\approx 100$  psec.

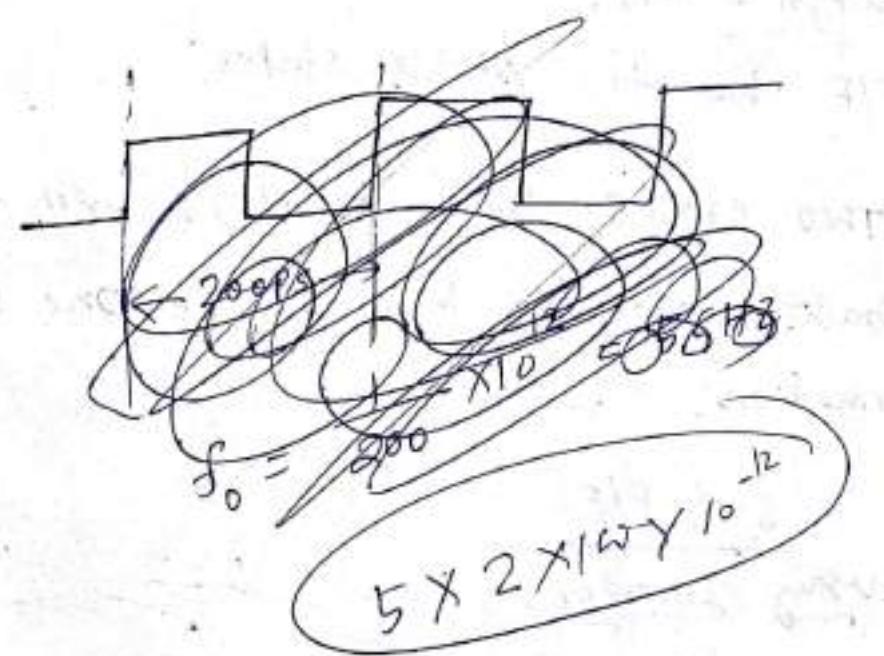
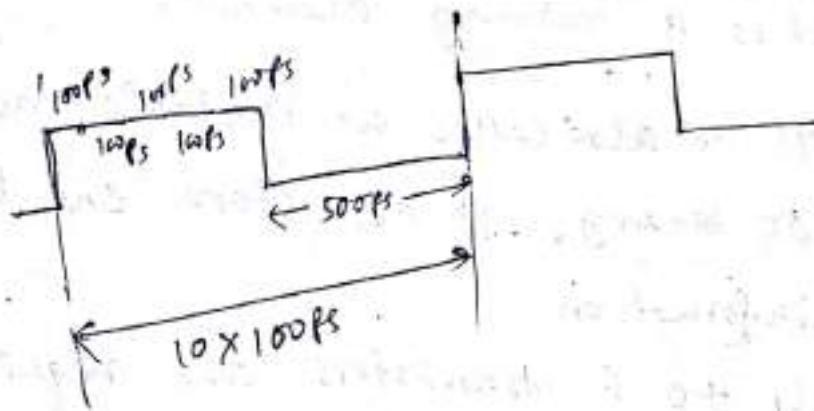
(f) fundamental freq of osc. o/p = ?

$$5 \times 100 \times 10^{-12} \times 2 \begin{cases} \rightarrow \text{ON} \\ \rightarrow \text{OFF} \end{cases}$$

$$= 10^{-9}$$

$$f = \frac{1}{10^{-9}} = 10^9 = \frac{10^3 \times 10^6}{10^2 \text{ MHz}}$$

$$= 1.947 \text{ MHz}$$



## - : Sequential Ckt : -

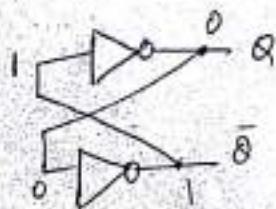
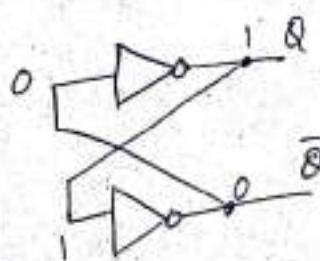
In this ckt's it's off at any time not only dep. the present ips but also previous ips as well as previous off.

### Flip flop

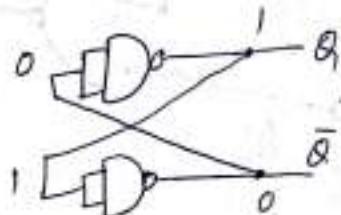
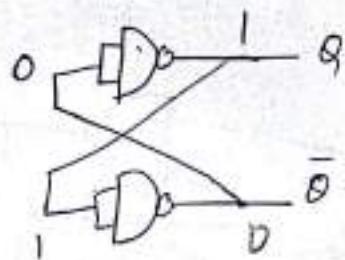
- ⇒ It is a memory element.
- ⇒ F/F is also called as bistable multivibrator or binary. It can store one bit of information.
- ⇒ 4 to 6 transistors are required to design a f/f.
- ⇒ F/F has two stable states.
- ④ Two cross coupled inverters will form a basic latch which can store one bit of information.

### S-R F/F

#### a) Using Inverter.



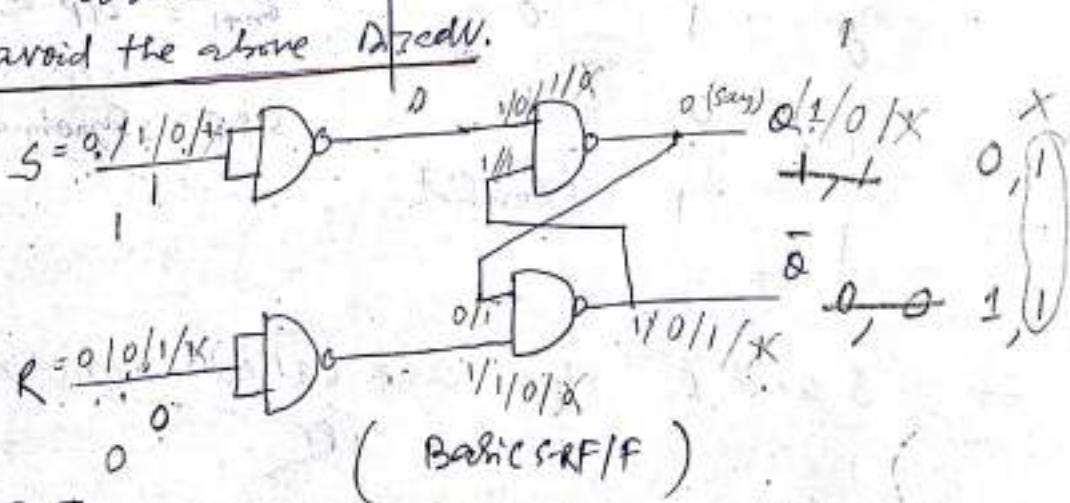
b) using NAND Gate.



Disadv.

To change the off we have to first disconnect the off & then give the off again we connect then give the off again for changing the off the above step also repeated.

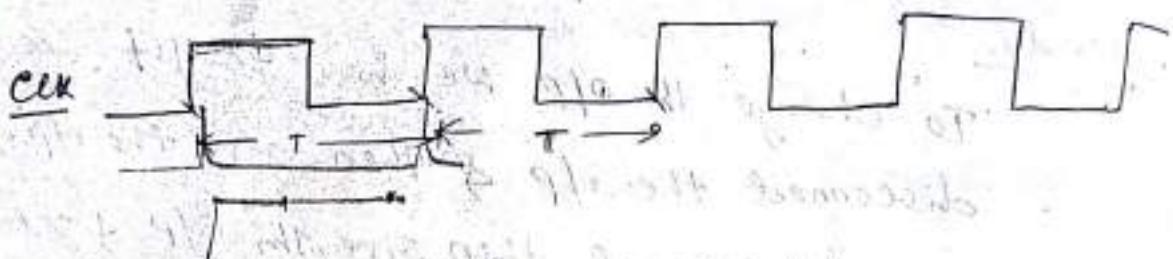
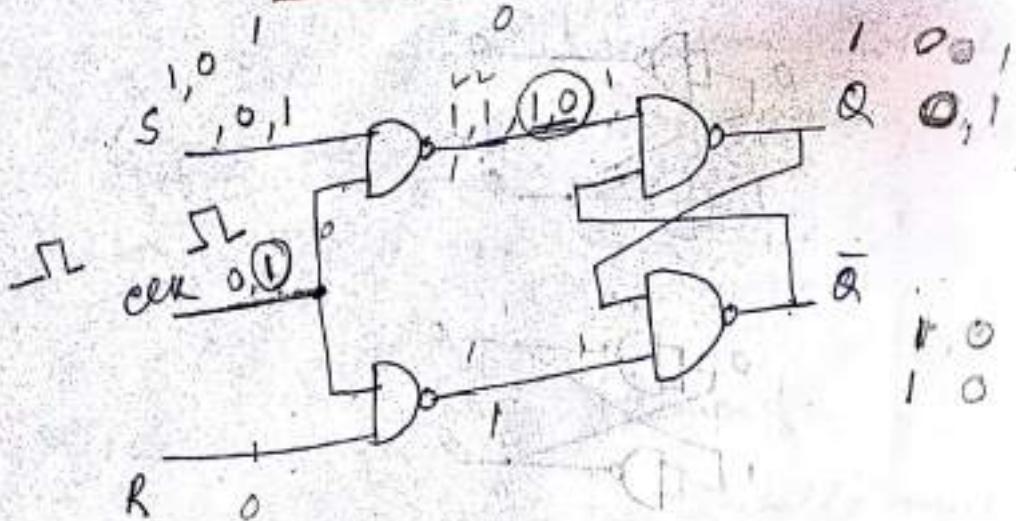
To avoid the above Disadv.



T. T

$\frac{S}{0}$	$\frac{R}{0}$	$\frac{Q}{\text{No. change}}$
0	1	0 (reset)
0	0	1 (set)
1		invalid state
1	1	

### Clocked S-R F/F



S      R       $Q_{n+1}$

0      0       $Q_n$

0      1      0

1      0      1

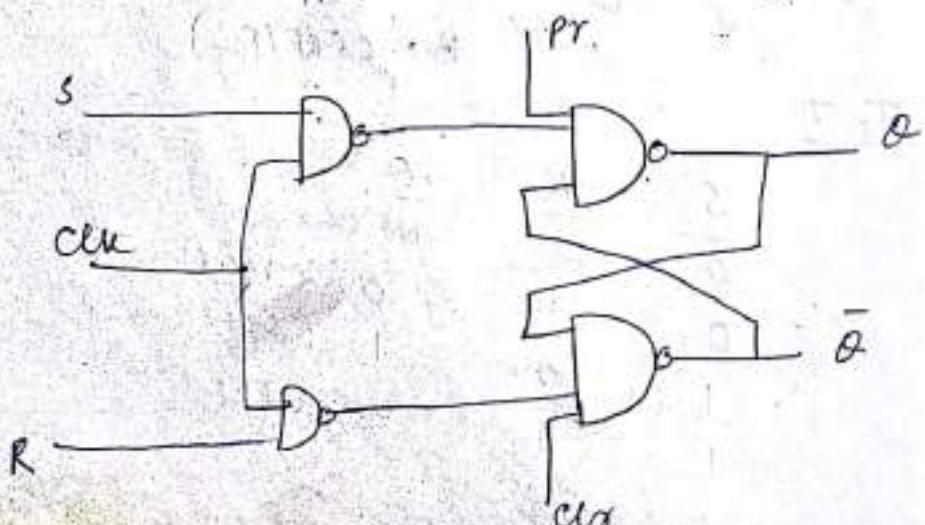
1      1      invalid.

$Q_n$  = output at just now

$Q_{n+1}$  = output after one clk

S & R = Synchronous inputs.

→ 'S' and 'R' inputs are called synchronous



→  $\overline{pr}$ ,  $\overline{clr}$  are called as direct or asynchronous inputs.  
 $\overline{pr}$ ,  $\overline{clr}$  are active low inputs.

T. T

Pr      cl      off

1 -

1.

off

Normal operation.

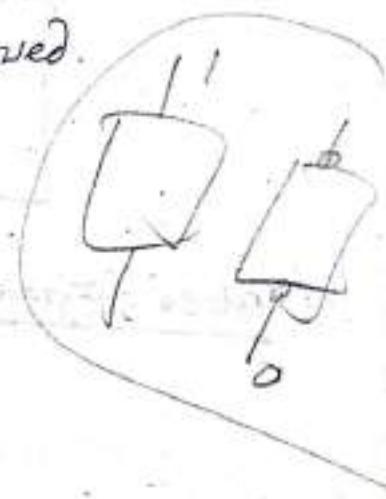
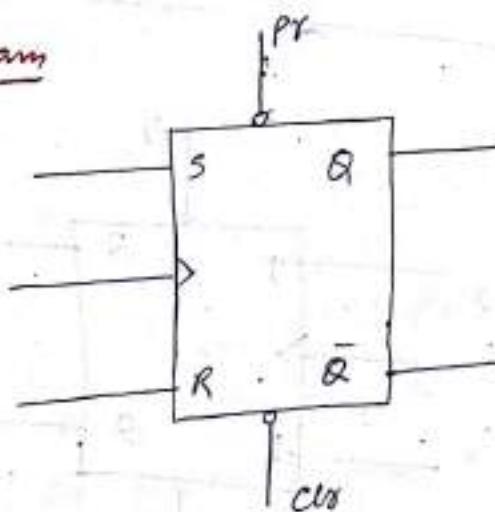
→ 0.      1      1.

→ 1.      0      0

0      0

not allowed.

Block diagram



→ The off of F/F changes only during the clock pulse. On best clock pulse the off of the F/F does not change.

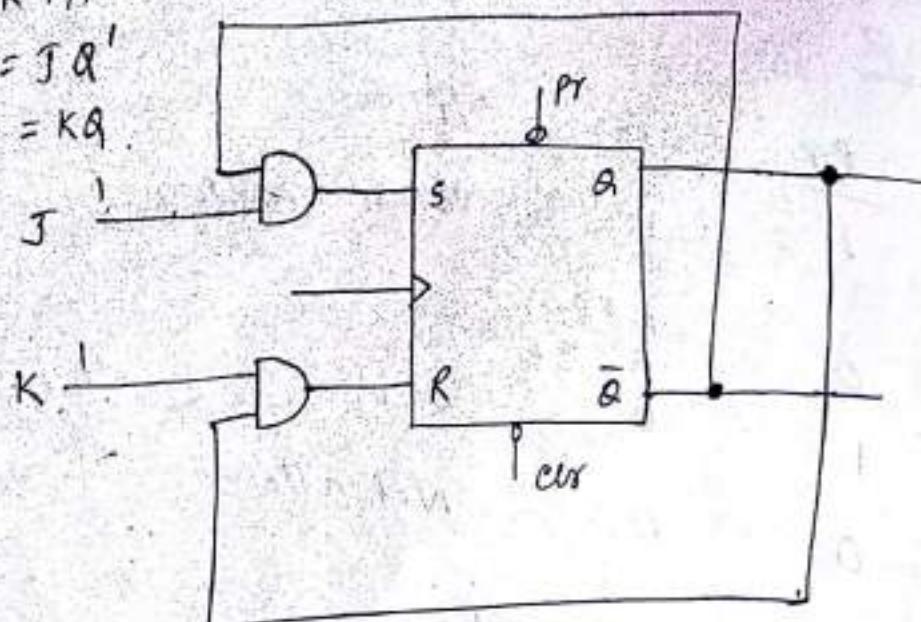
→ The disadvantages of S-R F/F is when  $S=1=R$ , then off can't be determined. This can be eliminated in J-K F/F.

## J-K F/F

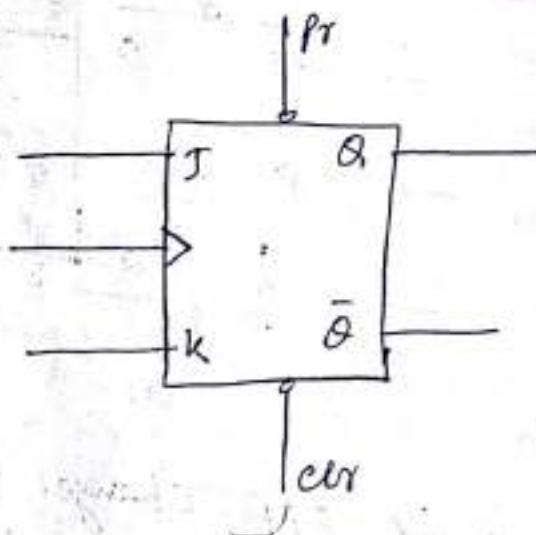
S-R F/F can be converted to J-K F/F by using two

$$S = JQ'$$

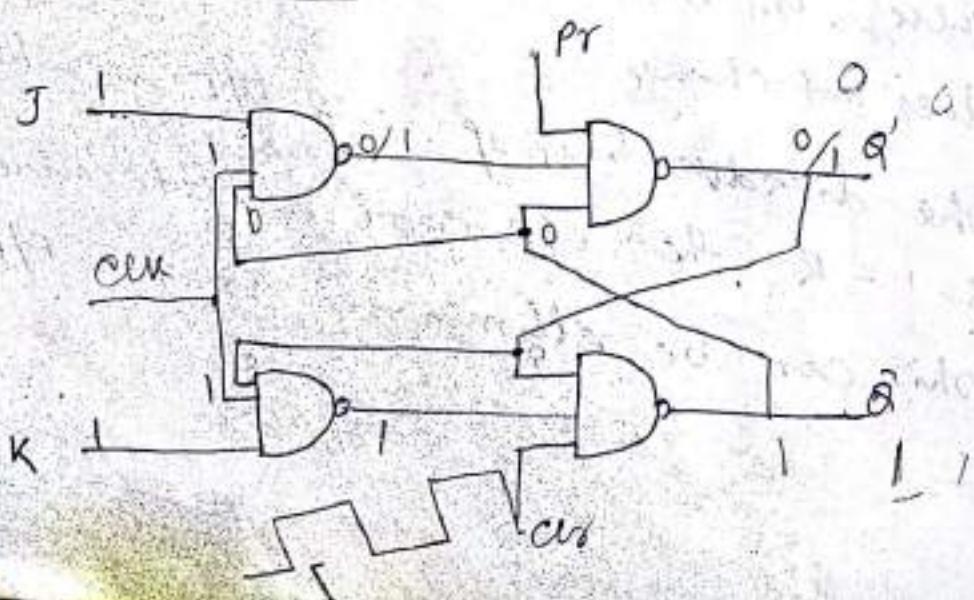
$$R = KQ$$



## Block diagram



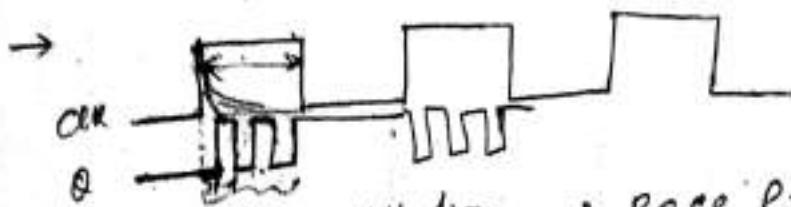
## Internal diagram



T. T

J	K	$\frac{Q_{n+1}}{N.C} = Q_n$
0	0	
0	1	0
0	0	1
1		Toggled = $\bar{Q}_n$
1		

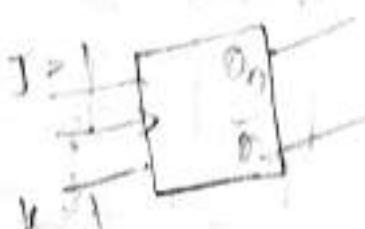
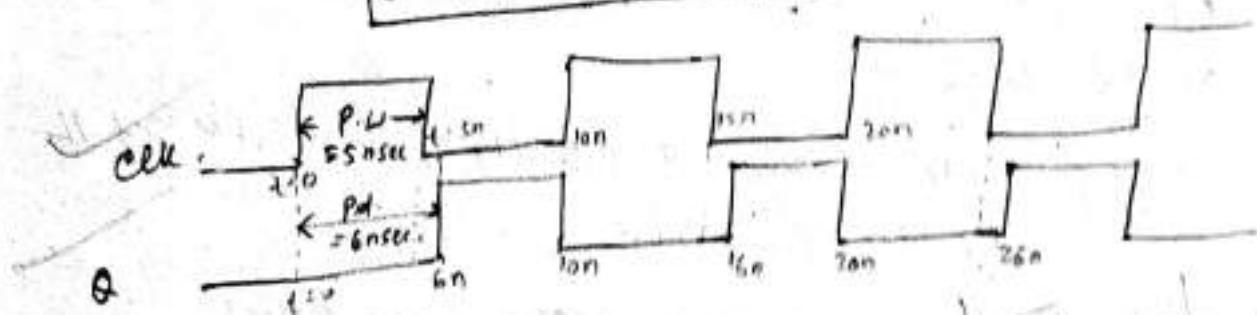
- Disadvantage, when  $J = K = 1$ , the o/p toggles more than once during the clock pulse is called "Race around problem".



Oscillation → Race Problem.

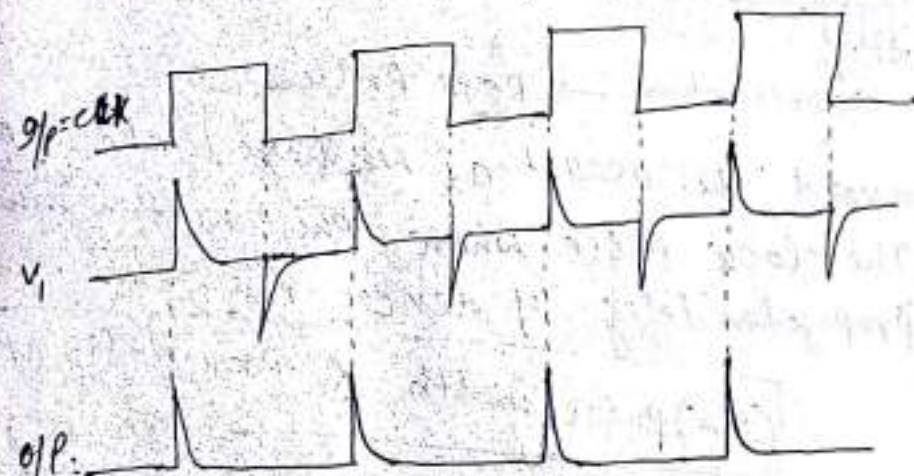
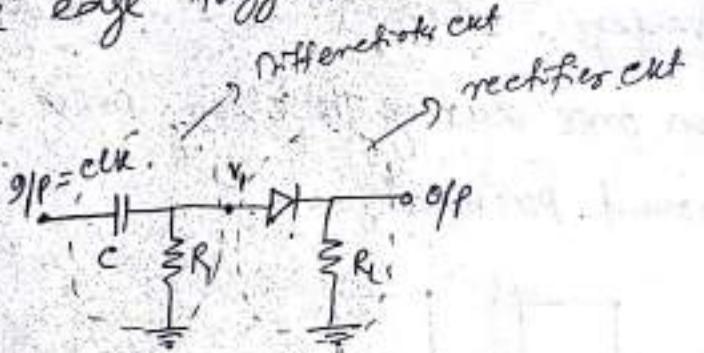
- To avoid this oscillation, we have to choose  
① The clock pulse whose pulse width is less than equal to propagation delay of device, i.e

$$(P.W) \text{ pulse width} \leq \text{prop. delay (P.D)}$$



②

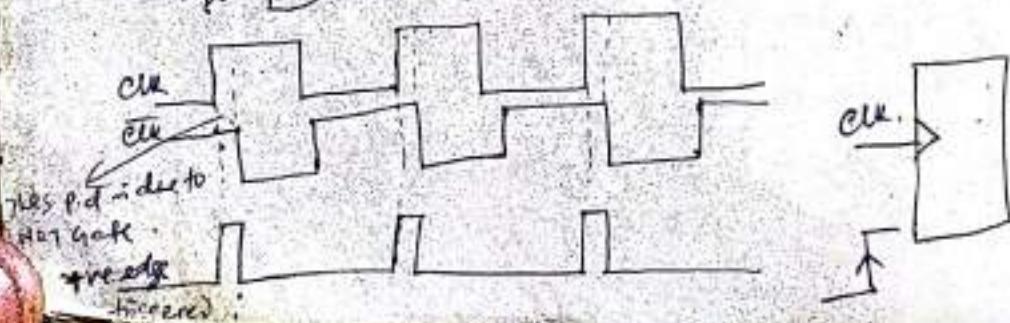
- By using edge trigger J/K FF:-  
(This process only meant for decreasing the pulse width.)  
The clock pulses are applied to a differentiator circuit followed by clamping to generate either +ve edge trigger or -ve edge trigger clock pulse.



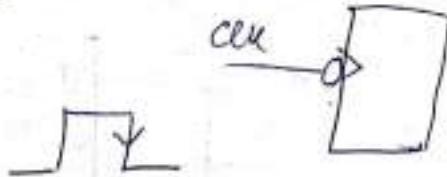
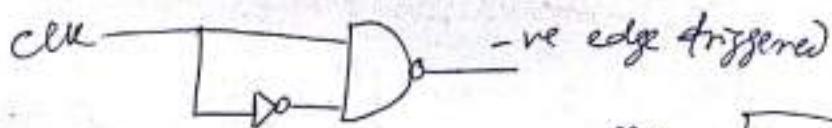
= +ve edge triggered.

③ If we change the polarity of Diode we get the -ve edge triggered pulse.

OR  $\text{clk}$  -ve edge triggered (+ve pulse)

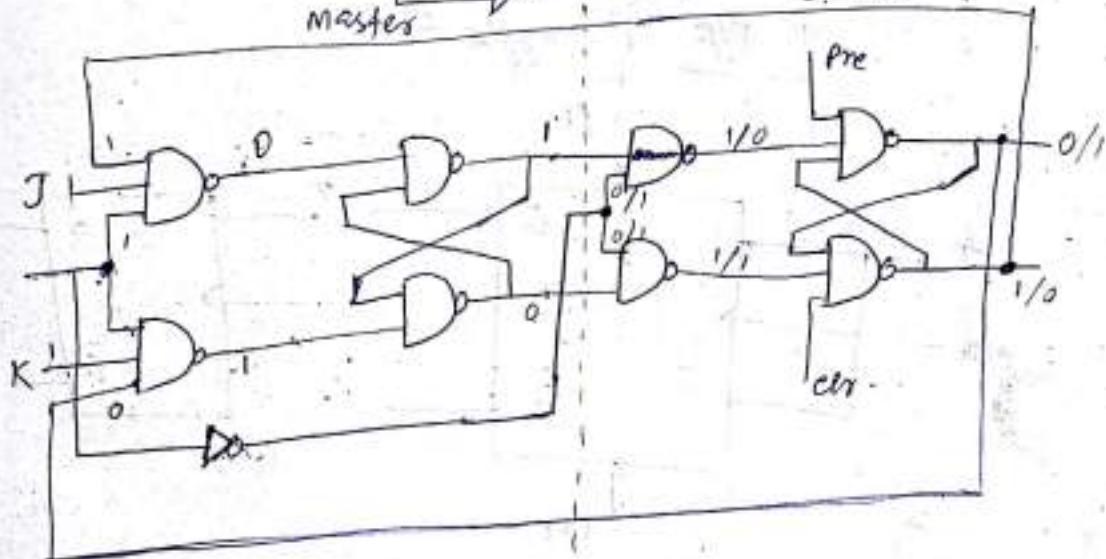
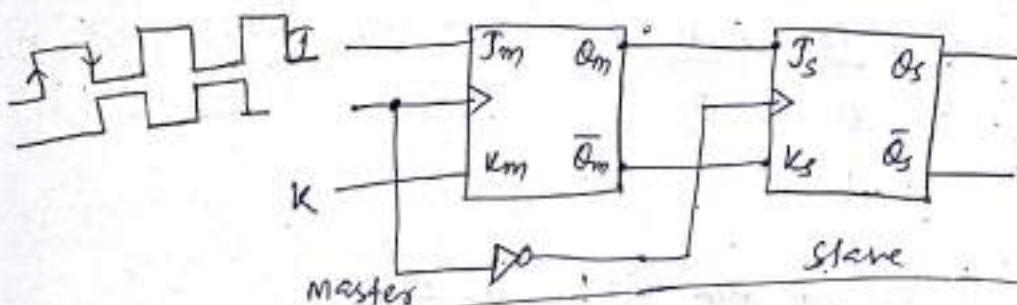


→ To generate -ve edge triggered



③

### master-slave J-k F/F



$$J = K = 1$$

clock pulse width is long master-slave f/f only complementary or

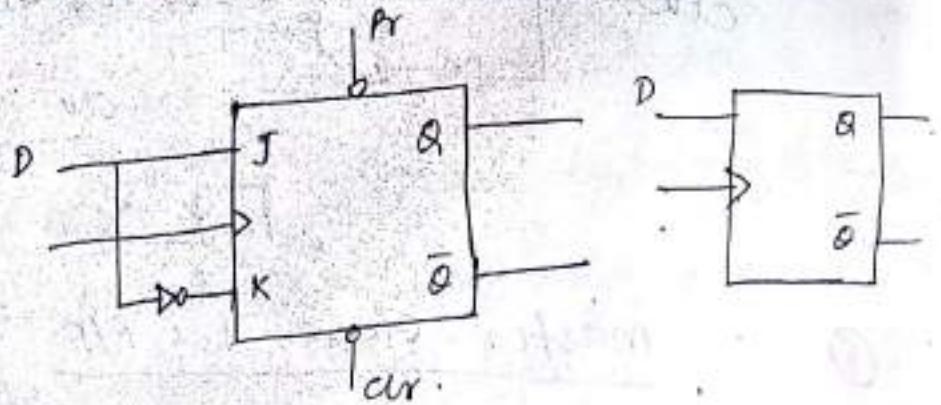
→ irrespective of clock pulse width the master f/f will be complemented ~~more times~~, but

→ Q/P of slave f/f will never be  $J=1$ , so Q/P slave toggles more times, but

→ When clock goes -ve the slave f/f will be complemented

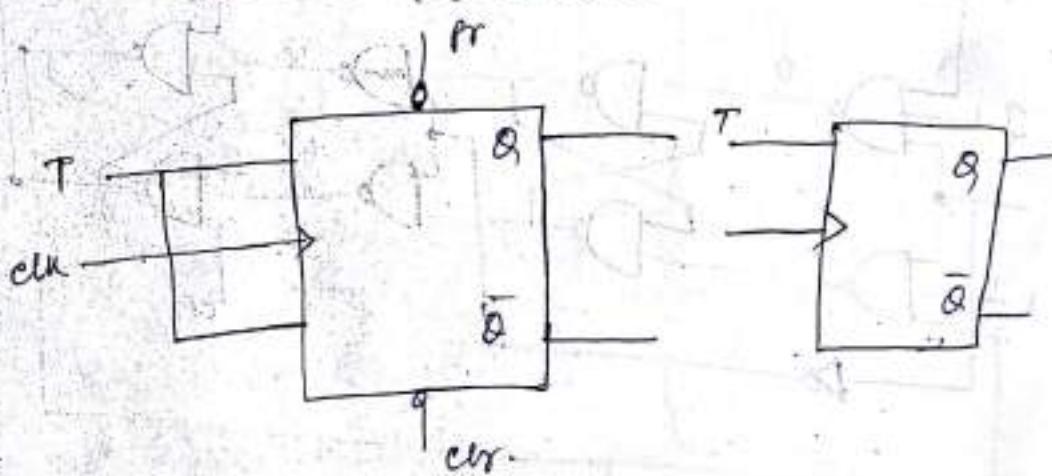
→ M/S J-k F/F can be referred as -ve edge triggered J-k F/F.

### D F/F (Delay / delay F.F)



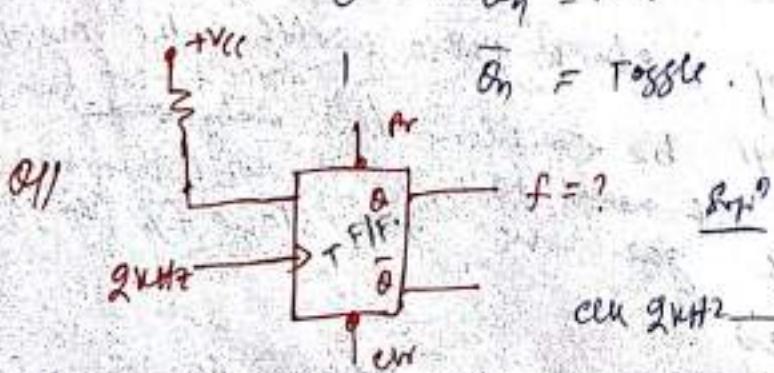
<u>D</u>	<u><math>Q_{n+1}</math></u>
0	0
1	1

### T F/F (Toggle F/F)



<u>T</u>	<u><math>Q_{n+1}</math></u>
0	$Q_n = n \cdot c$

$\bar{Q}_n = \text{Toggle}$ .



$$f = ? \quad \frac{f_{in}}{2}$$

clk 2Vcc

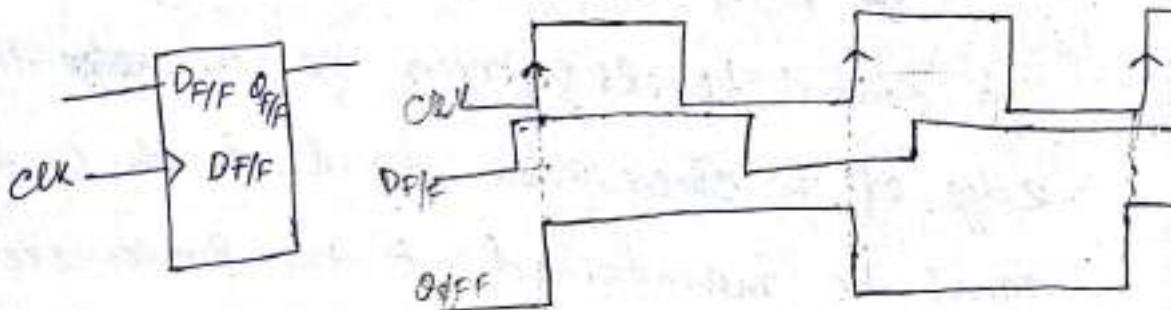
Q 1Vcc

$\Rightarrow$  The operation of f/f ~ dividing freq.

$D \xrightarrow{\text{Preferable}} \text{Register}, T \xrightarrow{\text{Preferable}} \text{counter}.$

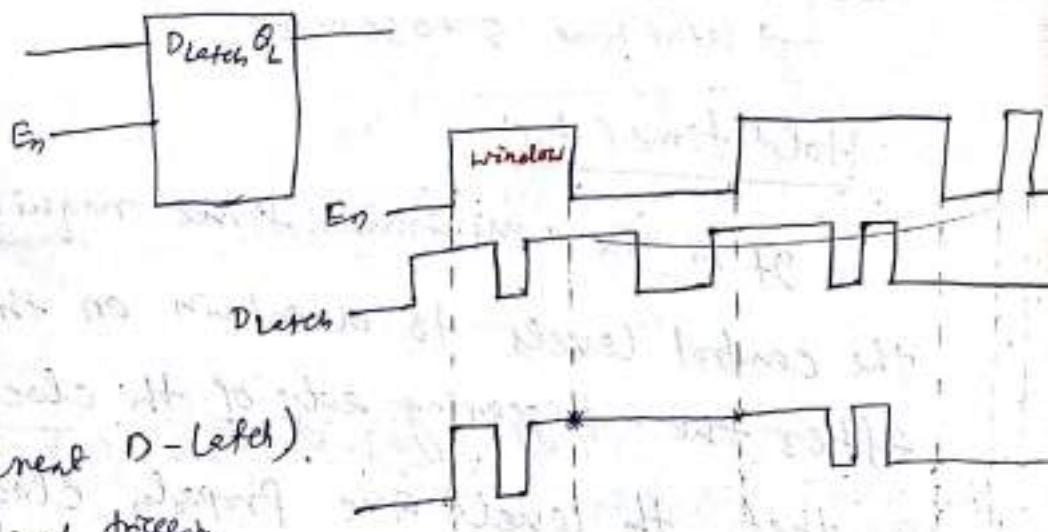
Q11 What is the difference b/w D-F/F & D-Latch.

Soln)



④ It can capture the i/p value at just the edge triggering.

For D-latch, we give the Enable i/p instead of clk

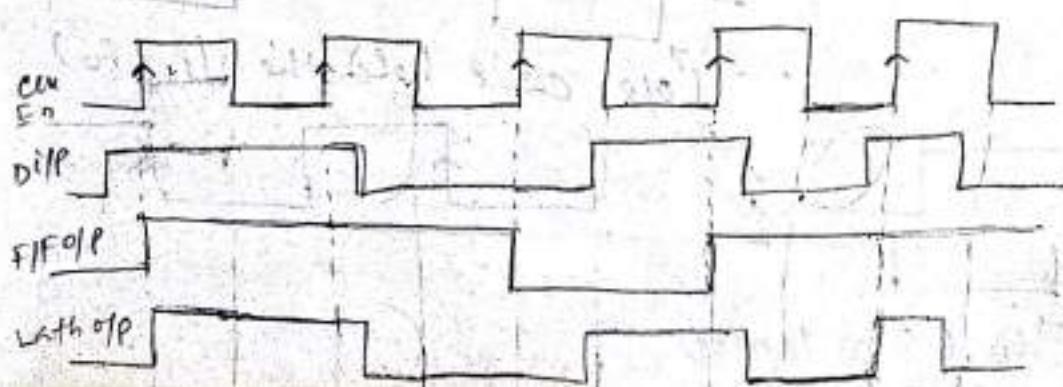


(Transparent D-latch).

D-latch level triggering

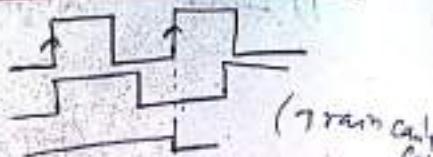
D-F/F edge triggering

Latch <sup>now</sup> hold the value.



## Setup time ( $t_s$ ) & Hold time ( $t_H$ )

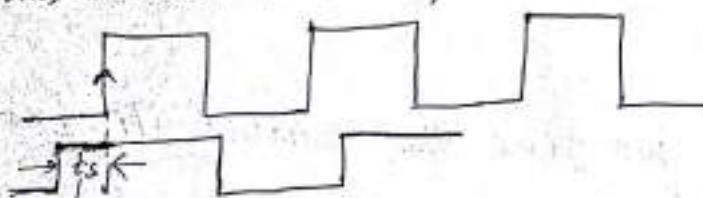
~~Setup time ( $t_s$ ):~~



$t_s$  is the minimum time required for i/p control levels, prior to the triggering edge of the clock pulse so that the Control must be maintained at the proper level.

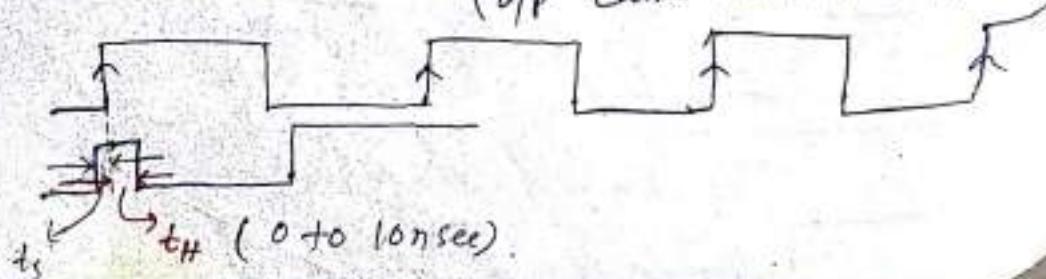
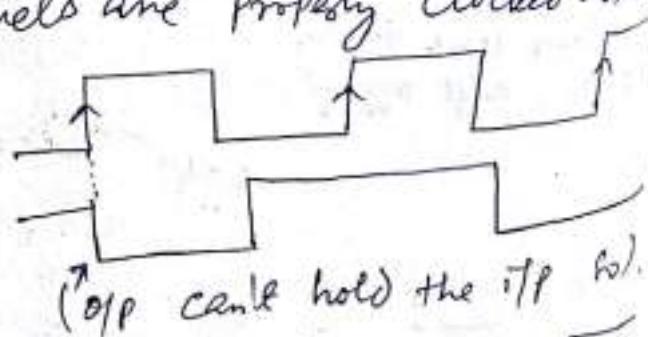
( Train can't catch at 7 P.M if you just arrive at 7 P.M )

④ This is due to presence of stray capacitance at i/p.



## Hold time ( $t_H$ )

$t_H$  is the minimum time required for the control levels to maintain on the i/p's after the triggering edge of the clock pulse so that the levels are properly clocked into the flip-flop.

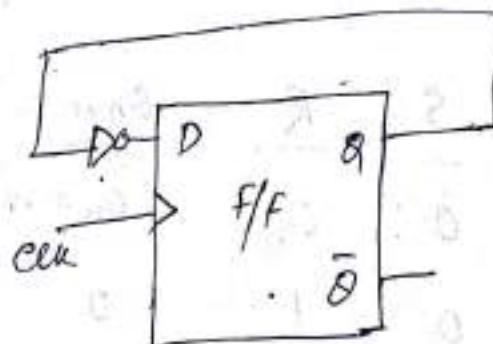


## propagation delay (P.D)

It is due to the presence of stray capacitance at the output of f/f. P.D is the time taken by IC to respond to the i/p signal.

Time required to get the so<sup>l</sup> of the i/p  
Maximum clk freq ( $f_{max}$ ) or Determination of minimum clock period :-

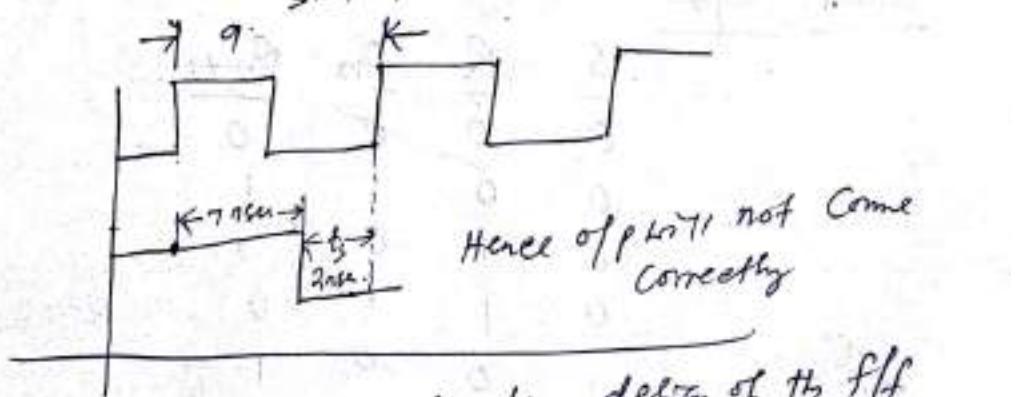
It is the highest rate at which the f/f can be properly triggered. If clock freq. is more than the max<sup>on</sup> specified value, the f/f may not be able to respond properly.



propagation delay of f/f = 5 nsec.

" " NOT gate = 2 nsec

" " setup time for f/f = 2 nsec.



↳ for design of the f/f  
 we have minimum CLK period = 10 ns

~~S-R~~ F/F

T-T:  $\frac{\text{Truth table}}{g_f}$  is defined as, ~~for~~ for the combination  
of op. What will be the op.

: Excitation Table: (Exci. T)

$g_f$  is defined as ~~for~~ for the combination  
of op. What will be the g/f.

: Characteristic eqn:

The set the relation of  $Q_{n+1}$   
in func<sup>n</sup> of  $Q_n$  & Synchronous op of f/f.

		S-R F/F		$Q_{n+1}$
		S	R	$Q_{n+1}$
		0	0	$Q_n = n.c$
		0	1	0
		1	0	1
		1	1	Invalid

Char. eqn

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

$$Q_{n+1}(S, R, Q_n) =$$

$Q_n$	00	01	11	10
0	1			
1	1	1	x	x

$$Q_{n+1}(S, R, Q_n) = S + \bar{R} Q_n \rightarrow \text{char. eq.,?}$$

Exci. Table.

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

$Q_{n+1}(S, R, Q_n)$	$Q_n$	00	01	11	10
S	0	1			
R	0	1	X	X	X

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

J-K FF

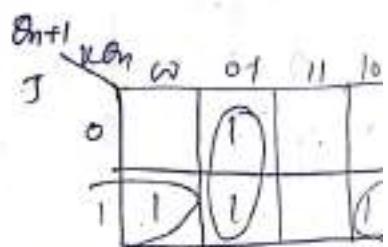
T-T

<u>J</u>	<u>K</u>	<u>Q<sub>n+1</sub></u>	<u>Q<sub>n</sub></u>	<u>J</u>	<u>K</u>	<u>Q<sub>n+1</sub></u>
0	0	0	0	0	0	0
0	1	1	0	1	0	1
1	0	0	1	1	1	0
1	1	1	1	0	0	1

$Q_n = \text{Toggle}$ .

char. eq's

<u>J</u>	<u>K</u>	<u>Q<sub>n</sub></u>	<u>Q<sub>n+1</sub></u>	<u>J</u>	<u>K</u>	<u>Q<sub>n</sub></u>	<u>Q<sub>n+1</sub></u>
0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0
1	0	1	0	1	0	1	0
1	0	1	0	1	1	1	0
1	1	0	0	0	0	0	1
1	1	1	0	0	1	1	1



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

char. eq's.

Exc. Test

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

<u>Q<sub>n</sub></u>	<u>Q<sub>n+1</sub></u>	<u>J</u>	<u>K</u>	<u>Q<sub>n</sub></u>	<u>Q<sub>n+1</sub></u>	<u>J</u>	<u>K</u>
0	0	0	X	0	0	0	0
0	1	1	X	1	0	1	1
1	0	0	X	1	0	0	1
1	0	0	0	1	1	1	0
1	X	X	0	1	1	1	0

<u>D F/F</u>		
<u>T.T</u>	<u>D</u>	<u><math>Q_{n+1}</math></u>
	<u>0</u>	<u>0</u>
<u>q</u>		<u>1.</u>

<u>char. eqs?</u>	<u>D</u>	<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u><math>Q_{n+1} = D</math></u>	<u>char. eqs?</u>
	<u>0</u>	<u>0</u>	<u>0</u>		
	<u>0</u>	<u>1</u>	<u>0</u>		
	<u>1</u>	<u>0</u>	<u>1</u>		
	<u>1</u>	<u>1</u>	<u>1</u>		

<u>Exci. Table</u>	<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>D</u>	<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u><math>Q_{n+1} = D</math></u>
	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>
	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>
	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>
	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>
	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>	<u>1</u>

<u>T.T</u>	<u>I</u>	<u><math>Q_{n+1}</math></u>	<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u><math>Q_n</math></u>	<u><math>Q_{n+1} = I \oplus Q_n</math></u>
	<u>0</u>	<u><math>Q_n</math></u>	<u>N.C</u>	<u>0</u>	<u><math>Q_n</math></u>	
	<u>1</u>	<u><math>\bar{Q}_n</math></u>	<u>Toggle</u>	<u>1</u>	<u><math>\bar{Q}_n</math></u>	<u>Toggle</u>

<u>char. eqs</u>	<u>D</u>	<u>T</u>	<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u><math>Q_{n+1} = T'Q_n + Q_n T</math></u>
	<u><math>Q_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>0</u>	<u>0</u>	<u><math>Q_{n+1} = T'Q_n + Q_n T</math></u>
	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>	<u><math>Q_{n+1} = T'Q_n + Q_n T</math></u>
	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u><math>Q_{n+1} = T'Q_n + Q_n T</math></u>

Exci. Table  $\approx T \oplus Q_n$

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

## Conversion of F/Fs

\* Convert D F/F to J-K F/F

given  $F/F = D F/F$

target  $F/F = J - K F/F$

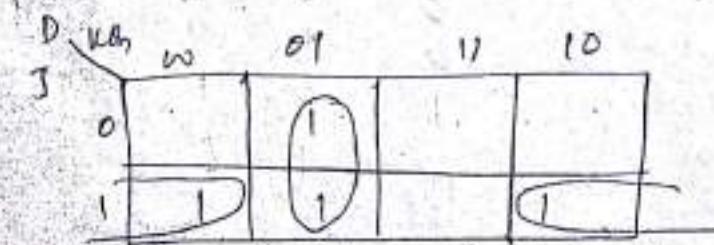
Procedure

→ Write T.T expansion of Target F/F.

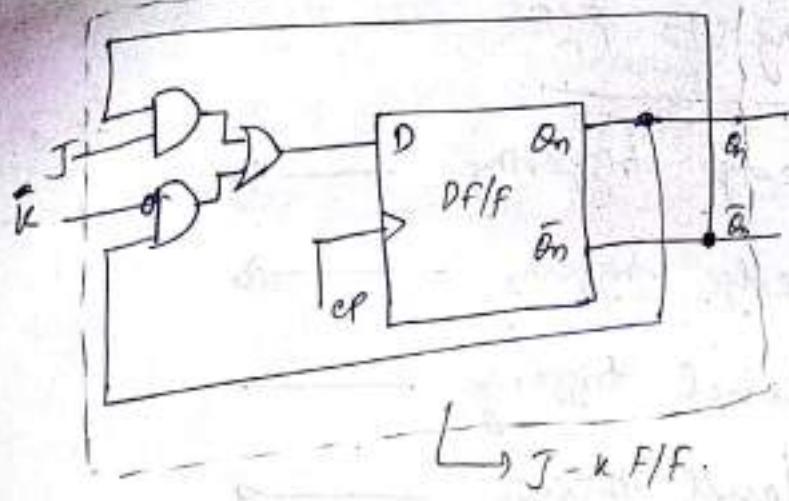
→ From that Target F/F T.T, write the excitation table for given F/F.

→ Now write the eqn. & Finally design.

<u>J</u>	<u>K</u>	<u><math>\bar{Q}_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>D</u>
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0



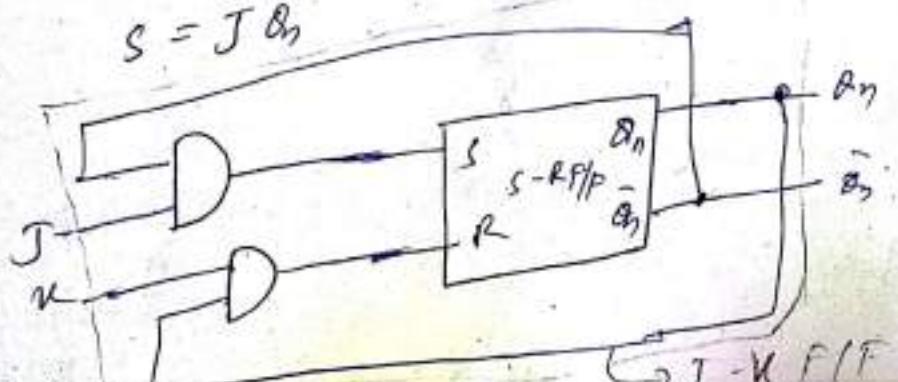
$$D = \bar{J} \bar{Q}_n + \bar{K} \bar{Q}_n$$



\* Convert S-R FF to J-K FF  
Given S-R FF, Target = J-K FF

<u>J</u>	<u>K</u>	<u><math>\bar{Q}_n</math></u>	<u><math>Q_{n+1}</math></u>	<u>S</u>	<u>R</u>
0	0	1	1	X	0
0	0	0	0	0	X
0	1	0	0	0	1
0	1	1	0	0	0
1	0	0	1	X	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	0	1	0

S =  $J \bar{Q}_n$       R =  $K \bar{Q}_n$



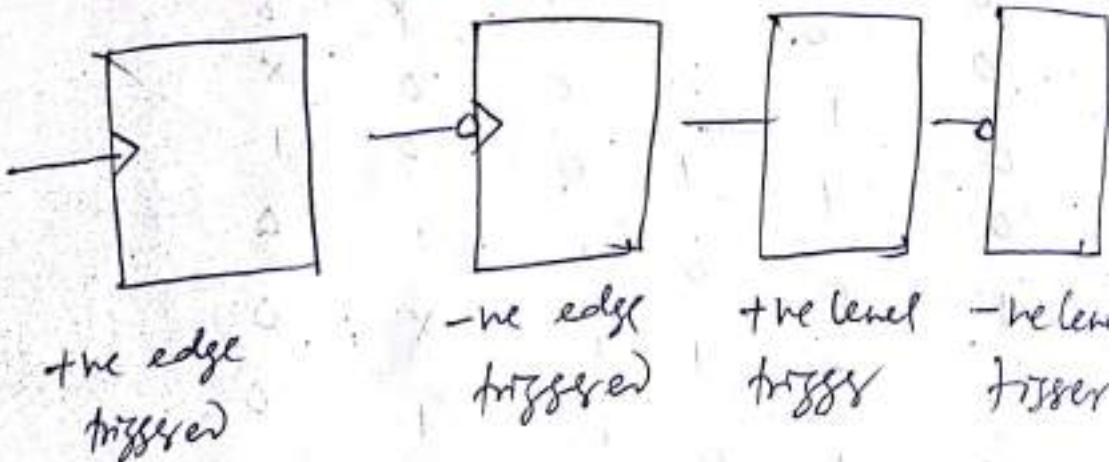
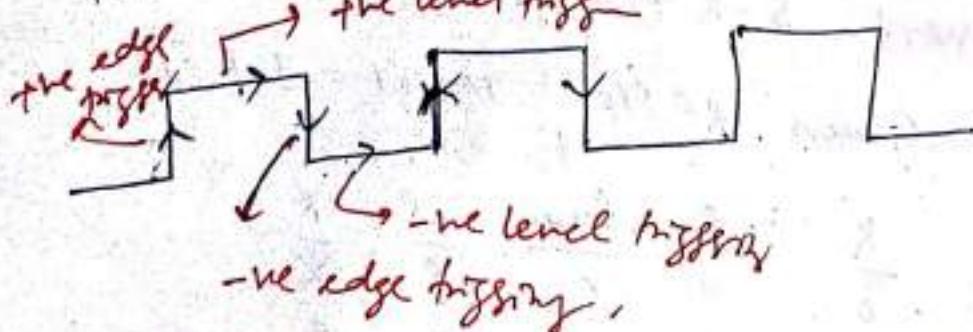
## Triggering to P/F

1. +ve edge triggering →

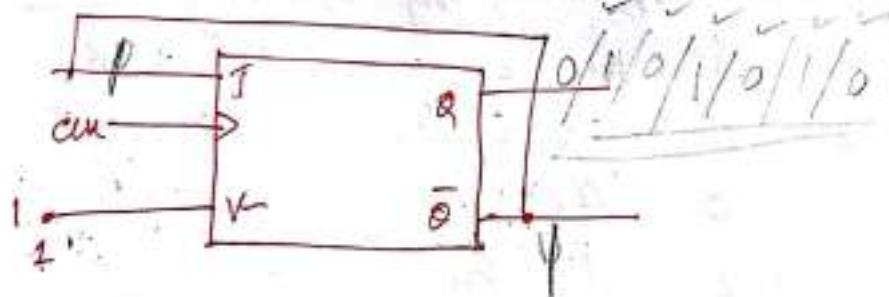
2. -ve edge triggering →

3. +ve level triggering →

4. -ve level triggering →



Q11 On a J-K FF  $J=Q$  &  $K=1$  assuming  
the f/f was initially closed and then closed  
for 6 pulses. Then the sequence at O/P Q will be

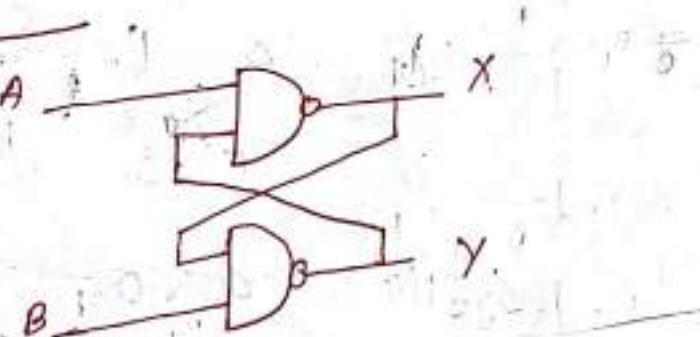


Soln

clk	Q
1	1
2	0
3	1
4	0
5	1
6	0

101010 (Ans)

Q11 If  $A=1$  &  $B=1$  & the ip B is replaced by  
a sequence 101010 --- then the op X & Y  
will be



Soln

X & Y are fixed at 0 & 1 respectively

Off the XY f/f whose char. table is given below to be implemented by using J-K f/f, we can be done by making \_\_\_\_\_.

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

$$\text{Target f/f} = X Y F/F$$

$$\text{Given f/f} = J K F/F \quad J = \checkmark \\ K = \checkmark$$

$X$	$Y$	$Q_n$	$Q_{n+1}$	$J$	$K$
0	0	0	1	1	0
0	0	1	1	*	0
0	1	0	0	0	0
0	1	1	1	x	0
1	0	0	1	1	x
1	0	1	0	*	1
1	1	0	0	0	x
1	1	1	0	x	1

$J$	$y_{Q_n}$	$W$	$O1$	$W$	$O1$	$W$	$O1$	$W$	$O1$
0	0	1	X	X					
1	1	X	1	X	X	1	X	X	1

$J = Y$        $K = X$

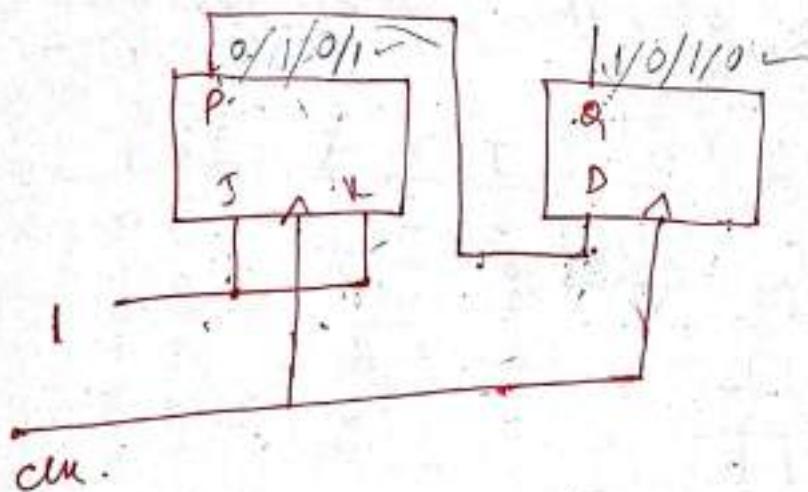
It can be done by making

$$J = \bar{Y}, K = X \quad (\text{Ans})$$

Q11 The following arrangement of m-s F/F has the initial state of P & Q as 0 & 1 respectively.

After 3 clk cycles the opp state P & Q will be -

80

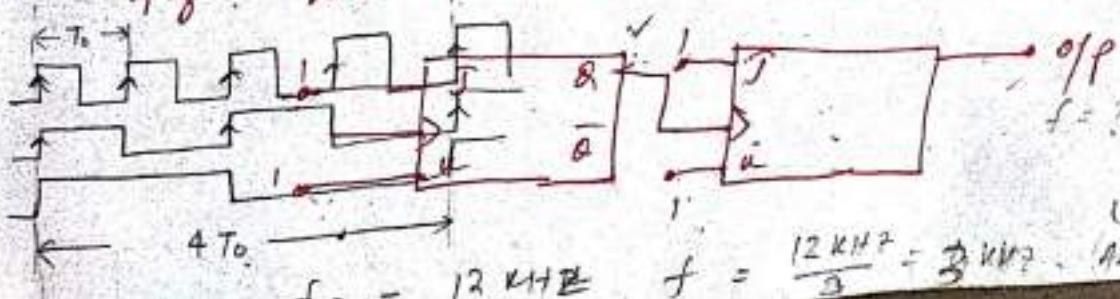


Sohg

clk	P	Q
-	0	1
↑	1	0
↓	0	1
↑	1	0
↓	1	0

$$P = 1, \quad Q = 0. \quad (\text{Ans})$$

Q11 An i/p freq. of 12 kHz is applied to 145 J-K F/F arrangement shown in the given fig. The resultant o/p freq will be -



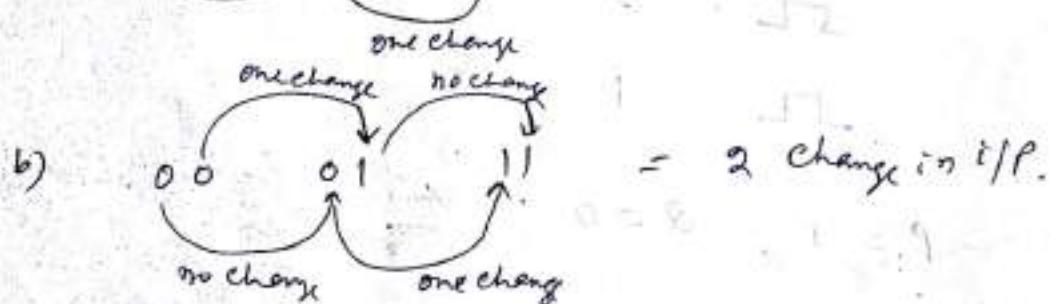
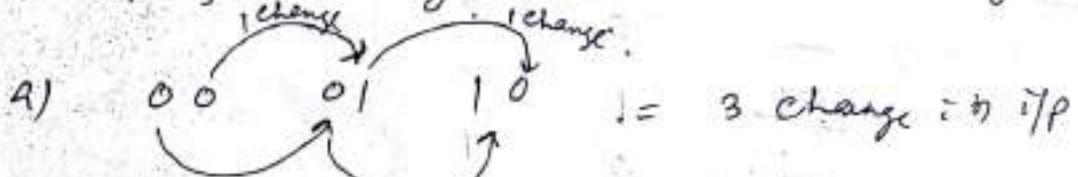
Q11 A -ve edge trigger J-K F/F whose initial off = 0 & in order to have the off states 0, 0, 1 in the next 3 successive CLK pulses. the J-K i/p states required would be

- a) 00, 01, 10 ✓
- b) 00, 01, 11 ✓
- c) 00, 10, 01 X
- d) 01, 10, 11 . X

Sol:

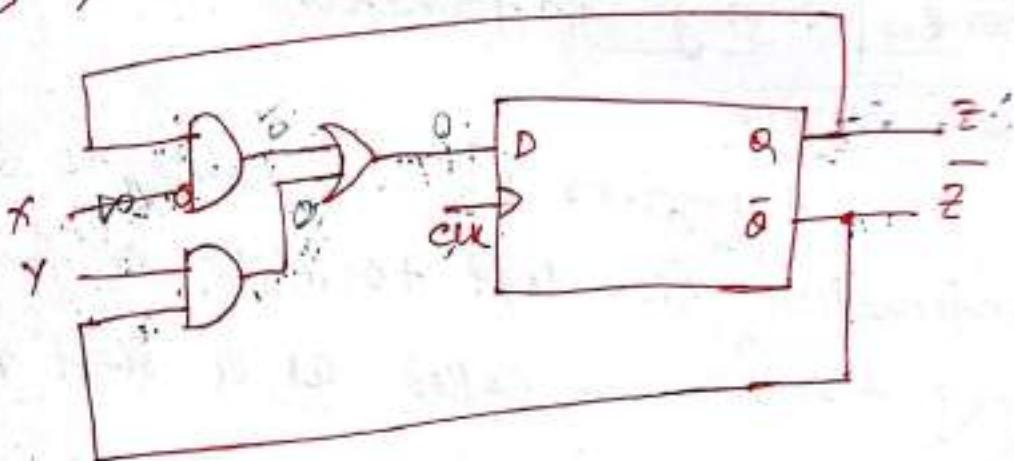
<u>CLK</u>	<u>θ</u>	<u>J</u>	<u>K</u>	<u>J K</u>
-	0	-	-	{ 0 0      0 X 0 1      0 X
↑	0	0	1 ] NC RS	{ 0 0      1 X 0 1      1 X
↑	0	0	1 ] NC RS	{ 1 1      1 X
↑	1	1	0 ] NC toggle set	{ 1 0      0 X

For perfect we go for minimum change in i/p



So Ans is b.

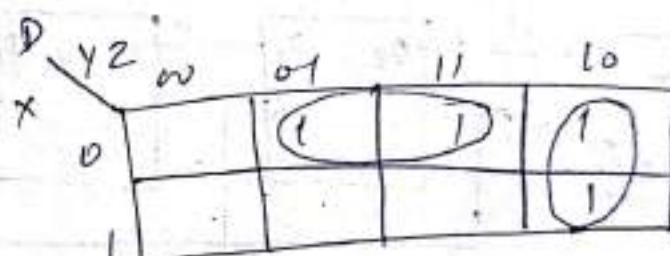
Q11 A sequential circuit using D FF & logic gates is shown in the fig. where  $x$  &  $y$  are inputs &  $z$  is the output. The out. is \_\_\_\_\_.



Sol:

$\bar{x}$	$\bar{y}$	$\bar{z}$	$\frac{z_{n+1}}{D}$	$\frac{D}{\bar{z}}$	$\bar{x}$	$\bar{y}$	<del><math>\bar{z}</math></del>
0	0	0	0	0	0	0	1
0	0	1	1	1	0	1	0
0	1	0	1	1	0	1	0
0	1	1	1	1	1	0	1
1	0	0	0	0	1	0	0
1	0	1	0	0	1	1	1
1	1	0	1	0	1	1	1
1	1	1	0	0			

] Toggle



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

It is JK FF with J = Y, K = X

Ans

## Registers

A register is a group of flip-flops used to store binary information.

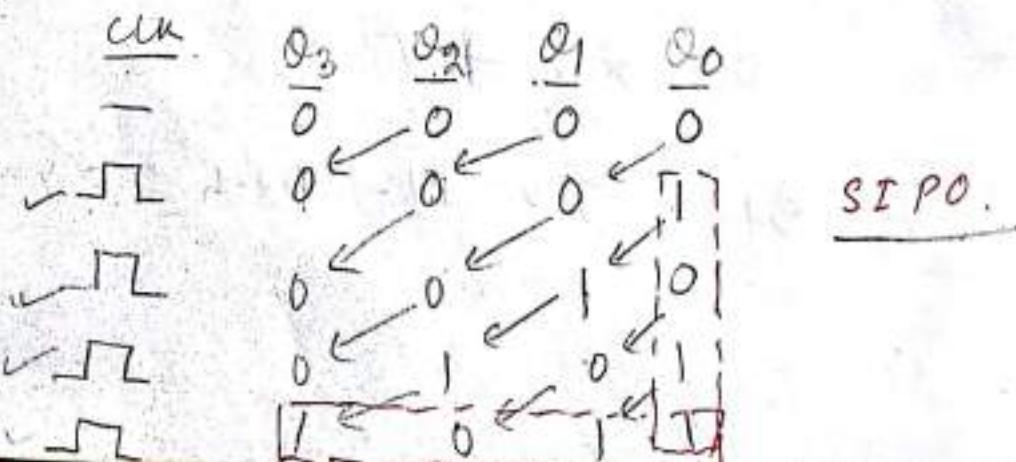
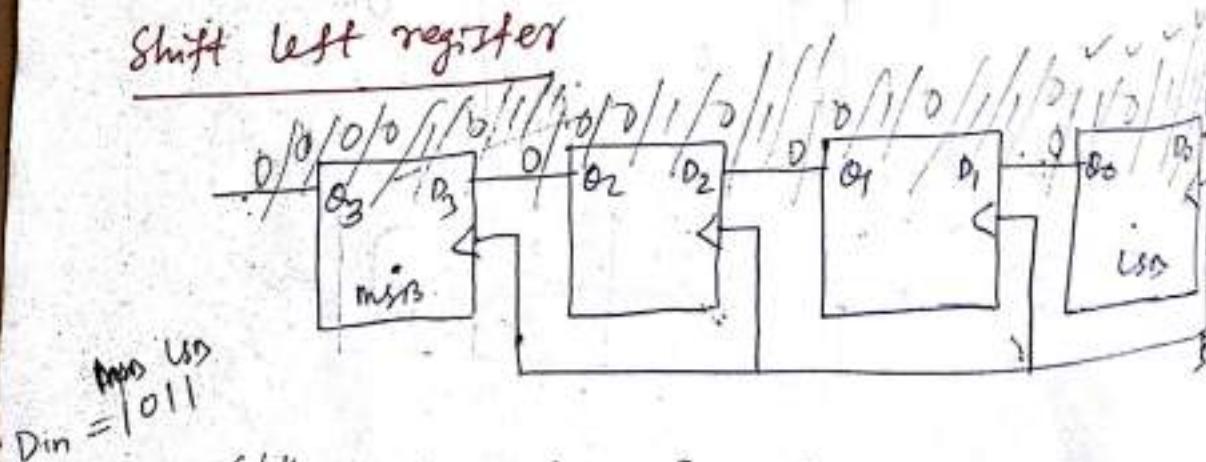
### Shift register

A register which is able to shift the information either from left to right or from right to left is called as a shift register.

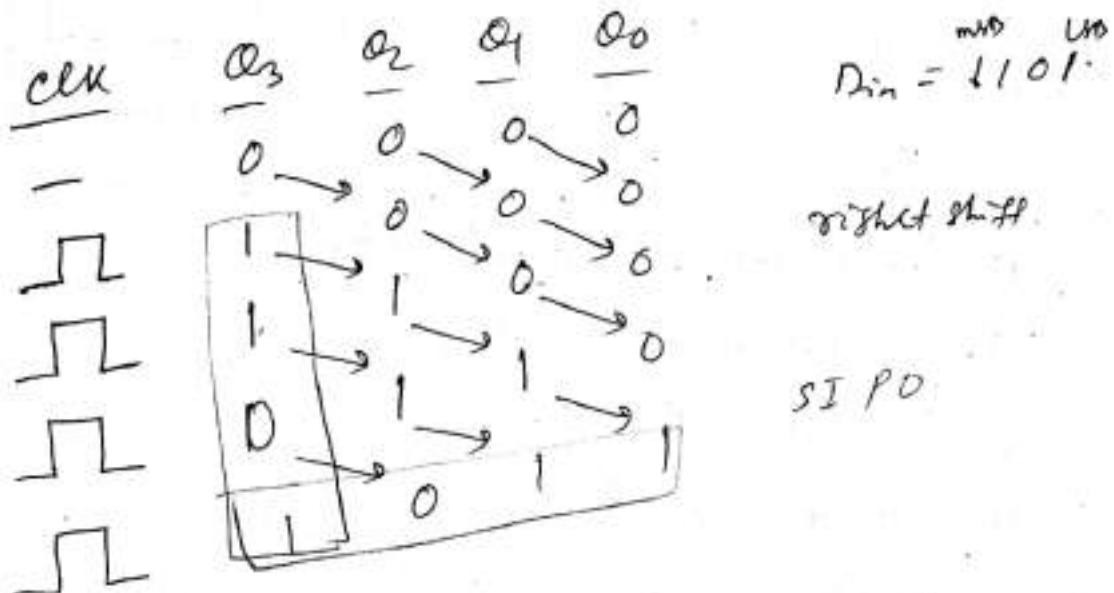
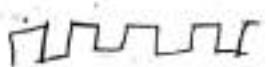
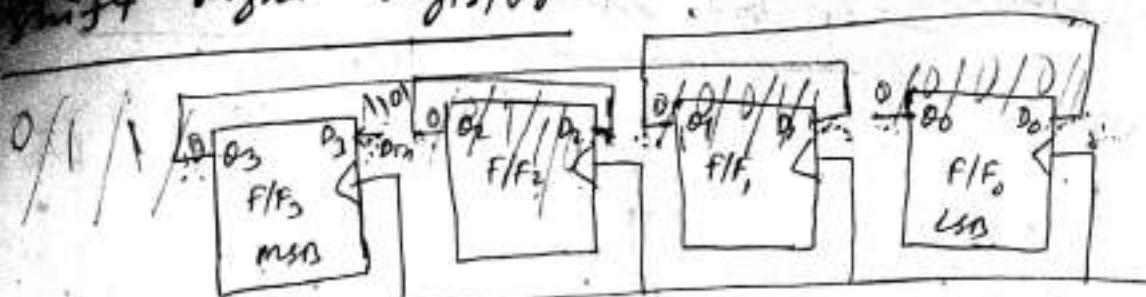
→ A shift register can perform four different operations.

1. SI PO
2. SIS
3. PI PO
4. PISO.

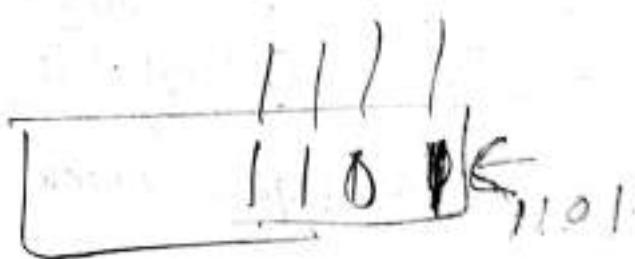
### Shift left register

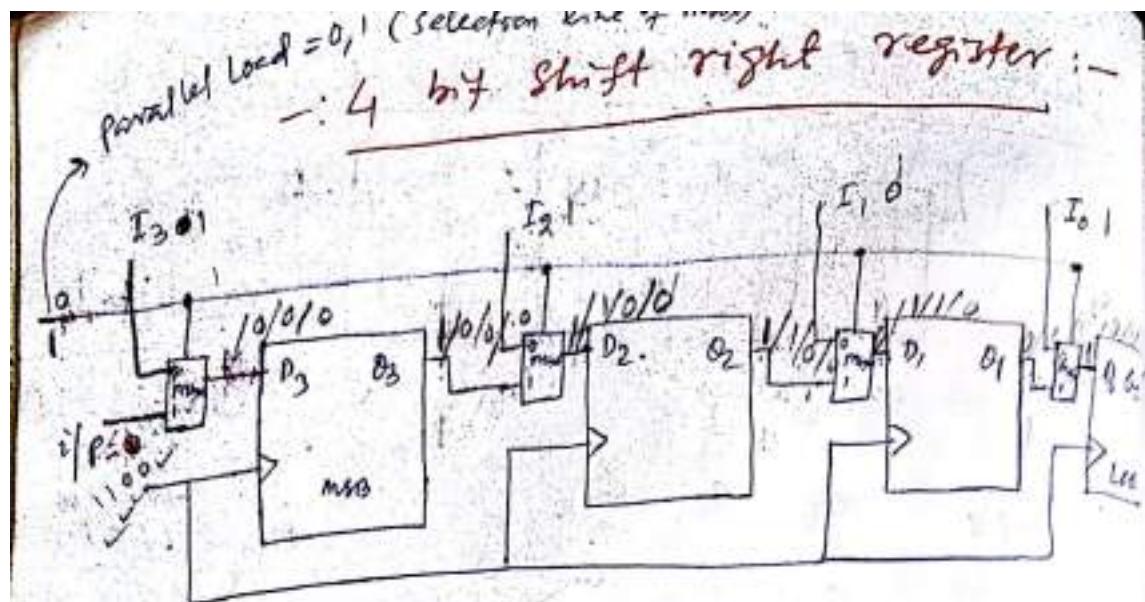


shift right register



0/0/0  
1011





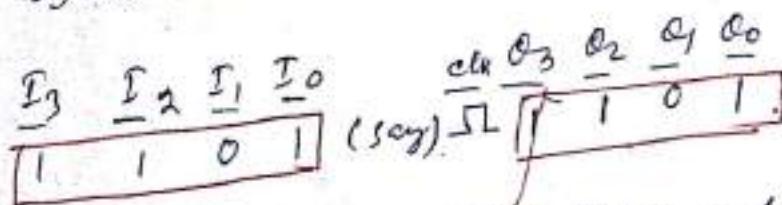
- 3)
- (1) parallel-in - parallel-out
  - (2) P. in - serial out.
  - (3) S. in - P. out
  - (4) S. in - S. out.

#### 1. PIPO

✓ when parallel load = 0.

$I_3 \ I_2 \ I_1 \ I_0 \rightarrow \text{SOP. } \checkmark$

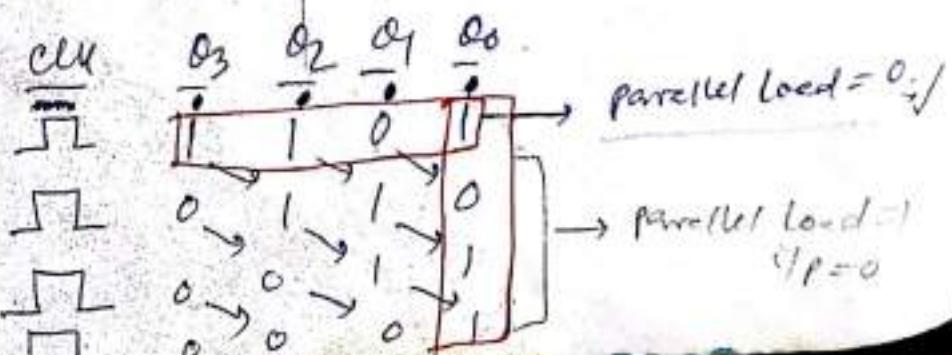
$Q_3 \ Q_2 \ Q_1 \ Q_0 \rightarrow \text{O/P. } \checkmark$



For PIPO required clock pulse = 1.

#### 2. PISO

when parallel load = 1,  $i/P = 0$ :

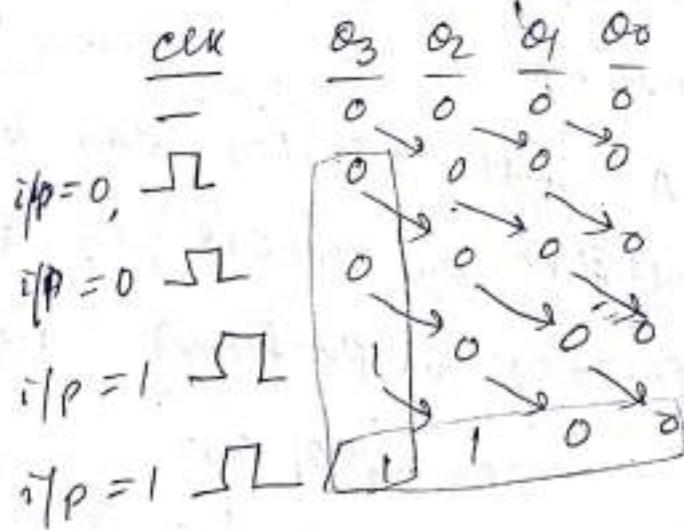


$\Rightarrow$  No. of clock pulses required = 4 =  $n$ .

3. SI PO. Parallel load = 1.

$i/p = 1100$

10/11,



$\Rightarrow$  No. of clock pulses required = 4 =  $n$ .

4. SI SO:

$I_3, I_2, I_1, I_0 = 1101$

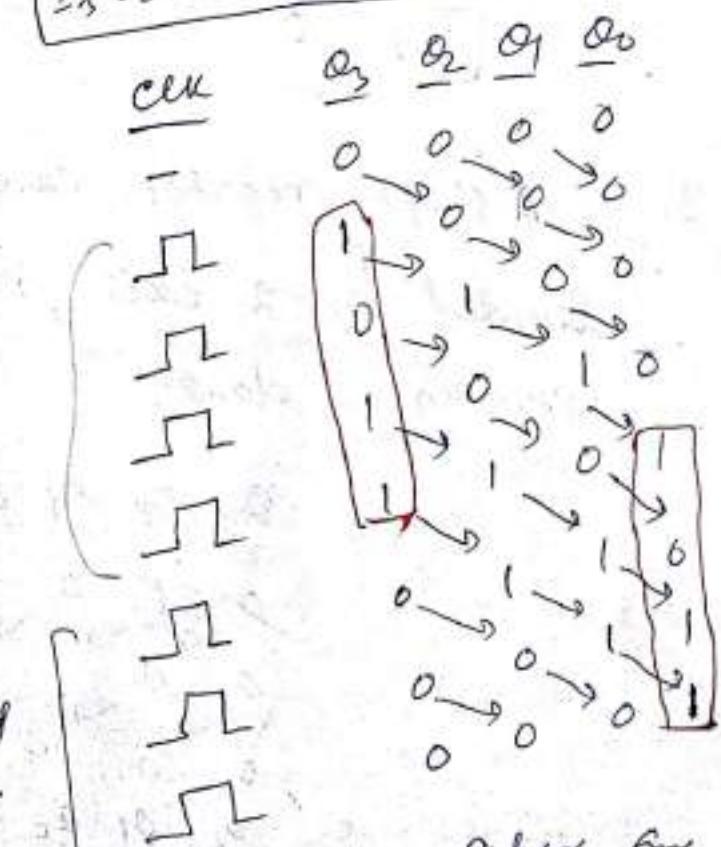
$i/p = 1$

Parallel load = 0

(2n-1)

$i/p = 0$

Parallel load  
= 1.



$\Rightarrow$  For 4 bit shift register 7 clock pulses are required to perform - SI SO, operations ( $2n-1$  all pulses for  $n$ -bit shift register).

### Application



1. Shift registers are used to perform serial to parallel & parallel to serial conversion.
2. A shift register can be used as multiplier by two ckt., if shift left operation is performed. i.e  $\times 2$

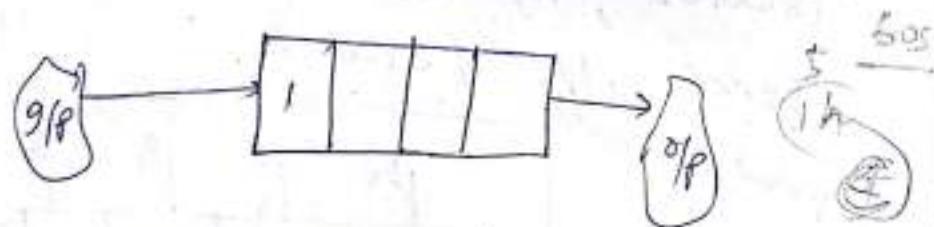
$$\begin{array}{cccc}
 & Q_3 & Q_2 & Q_1 & Q_0 \\
 & 0 & 0 & 1 & . & . & = 3 \\
 & \swarrow & \swarrow & \swarrow & & & \\
 0 & 1 & 1 & 0 & = 6 & = 3 \times 2 \\
 & \swarrow & \swarrow & \swarrow & & & \\
 0 & 1 & 0 & 0 & = 12 & = 6 \times 2 \\
 \text{o.f.} & 1 & 1 & 0 & 0 & = 24 & = 12 \times 2 \\
 \boxed{1} & 1 & 0 & 0 & 0
 \end{array}$$

3. A shift register can be used as a divider by 2 ckt., if shift right operation is done.

$$\begin{array}{cccc}
 & Q_3 & Q_2 & Q_1 & Q_0 \\
 & 0 & 1 & 0 & 0 & = 4 \\
 & \searrow & \searrow & \searrow & & \\
 0 & 0 & 1 & 0 & = \frac{4}{2} = 2 & = 4 \div 2 \\
 & \searrow & \searrow & \searrow & & \\
 0 & 0 & 0 & 1 & = 2 \div 2 = 1 &
 \end{array}$$
  

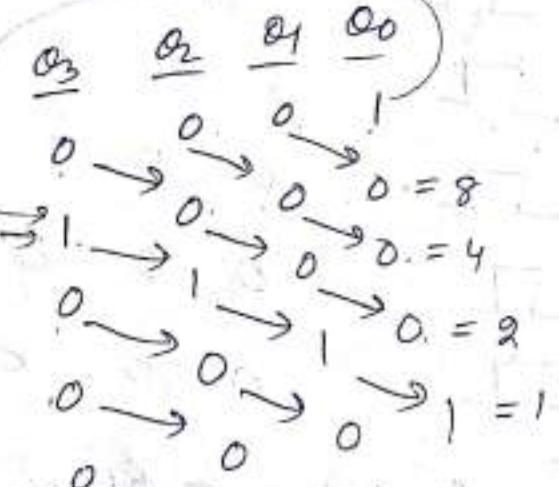
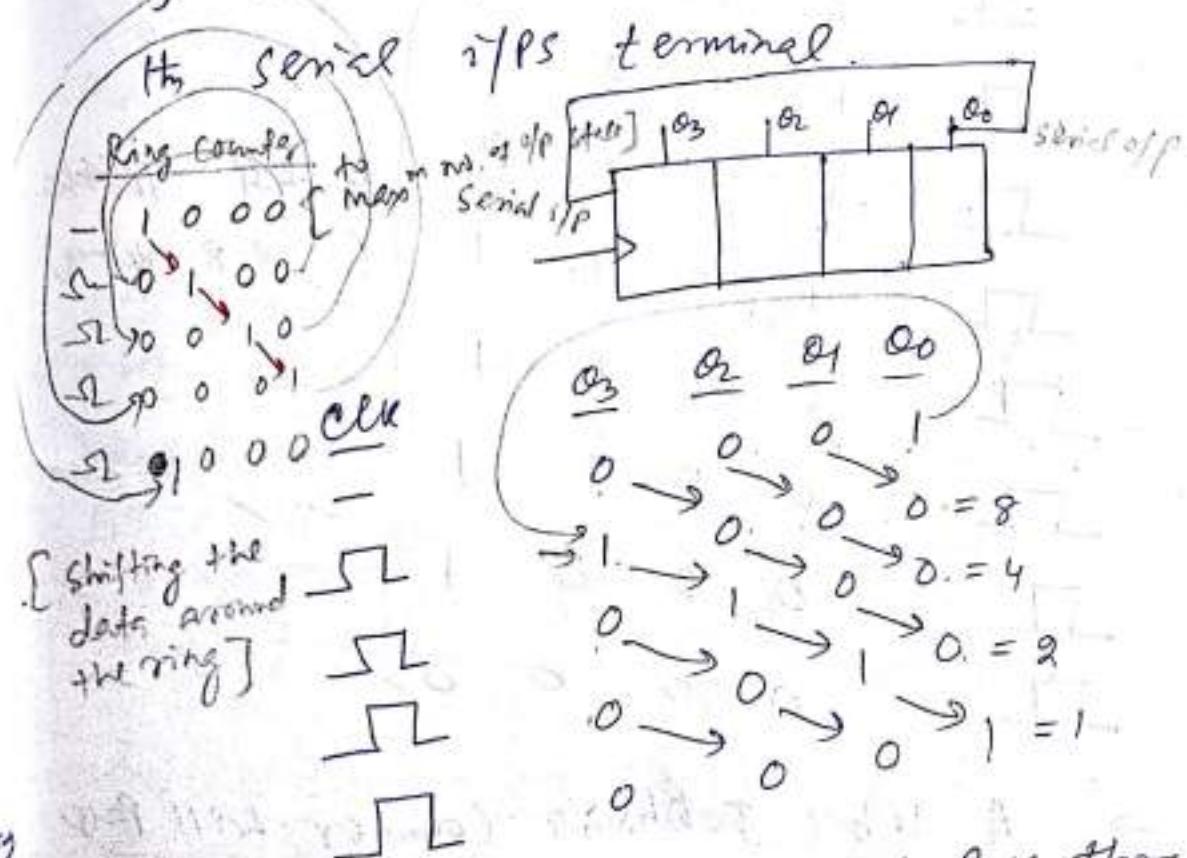
$$\begin{array}{cccc}
 \text{Bx} & Q_3 & Q_2 & Q_1 & Q_0 \\
 0 & 1 & 0 & 1 & = 5 \\
 0 & 0 & 1 & 0 & = 5 \div 2 = 2
 \end{array}$$

4. A shift register can be used as Digital delay Line (like street light lifetime for 12 Hrs)



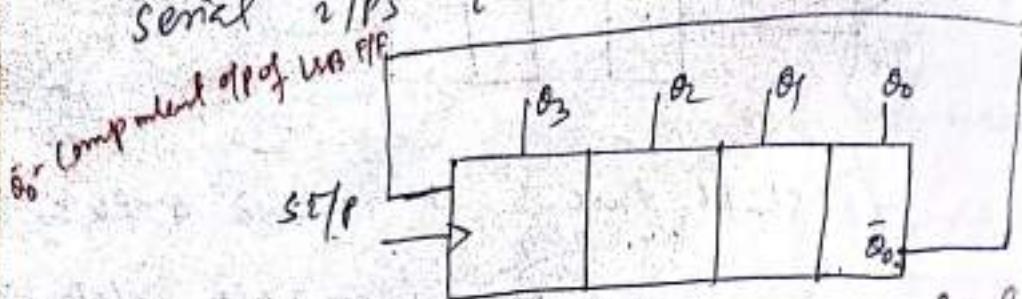
To shift this 1, we required 4 clk so  
9/8 can go to off after 4 clk - delay.

5. A shift register can be used as Ring Counter  
if serial off terminal is connected to



- A 4-bit ring counter will pass through a max<sup>m</sup> of 4 different off states (for n-bit, max<sup>m</sup> n-diff. off state require)
- For ring counter non-used state =  $16 - 4 \cdot \frac{8,4,2,1}{4 \text{ less used}} = 12$ .

6. A shift register can be used as a Twisted ring / Johnson's ring counter / switch counter, if  $\bar{Q}_0$  terminal is connected to serial  $\bar{I}PS$  terminal.



<u>CW</u>	$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
-	0	0	0	0	1	1	1	1
$\sqcap$	1	0	0	0	0	0	0	1
$\sqcap$	1	1	0	0	0	0	0	0
$\sqcap$	1	1	1	0	0	0	0	0
$\sqcap$	1	1	1	1	0	0	0	0
$\sqcap$	1	1	1	1	1	1	1	0
$\sqcap$	0	1	1	1	1	0	1	1
$\sqcap$	0	0	1	1	1	1	0	1
$\sqcap$	0	0	0	1	1	1	1	0
$\sqcap$	0	0	0	0	0	0	0	0

→ A 4-bit Johnson's Counter will pass through 8 diff. opp states. (i.e for  $n$  bit Johnson's Counter, max  $2^n$  opp states)

→ No. of unused state =  $16 - 8 = 8$ .

Note (1) Counters have max<sup>m</sup>  $2^n$  o/p states.

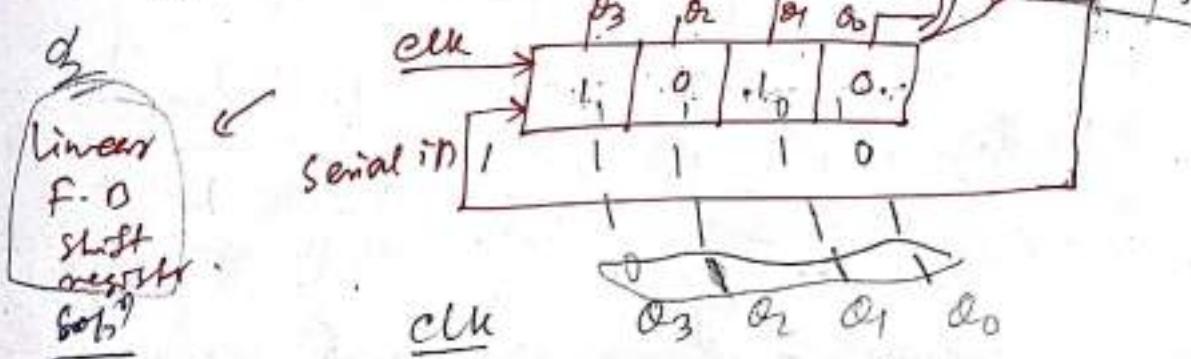
(2) Shift register ring counters have max<sup>m</sup>  $n$  o/p states.

(3) Shift register twisted ring counters have max<sup>m</sup>  $2n$  o/p states.

Q11 The initial contents of 4-bit serial-in-parallel-out right shift register shown in fig. is 1010.

After 4 clock pulses are applied, the contents of the shift register will be.

(a) 1111 (b) 1110 (c) 0111 (d) 1010

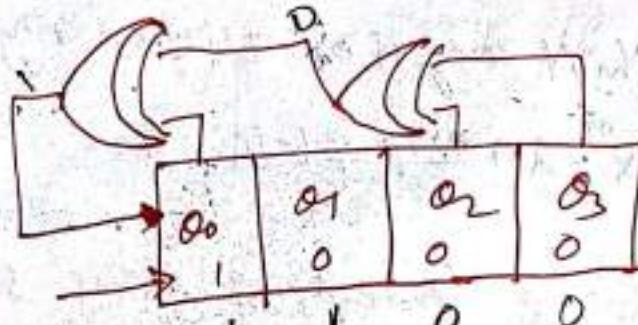


{ based on the  
E.B connected  
the pattern also  
changed }

CLK	Q3	Q2	Q1	Q0
-	1	0	1	0
↑	1	1	0	1
↑	1	1	1	0
↑	1	1	1	1
↑	0	1	1	1

①

Off A 4-bit shift register is initialize  
a value 1000. After how many CLK  
pulse the initial pattern re-appears.



<u>bit</u>	<u>clk</u>	<u>Q0</u>	<u>Q1</u>	<u>Q2</u>	<u>Q3</u>
1000	-	1	0	0	0
1100✓	↑	1	1	0	0
1110✓	↑	1	1	1	0
0111✓	↑	0	1	1	1
0011✓	↑	0	0	1	1
1000✓	↑	0	0	0	1
	↑	1	0	0	0

Ans After 6 CLK. (Ans) //

## Counters

Def' A Counter is a register capable of counting the no. of clock pulses which have arrived at clock i/p.

→ Counters are also referred as freq. divider cells.

### Counter.



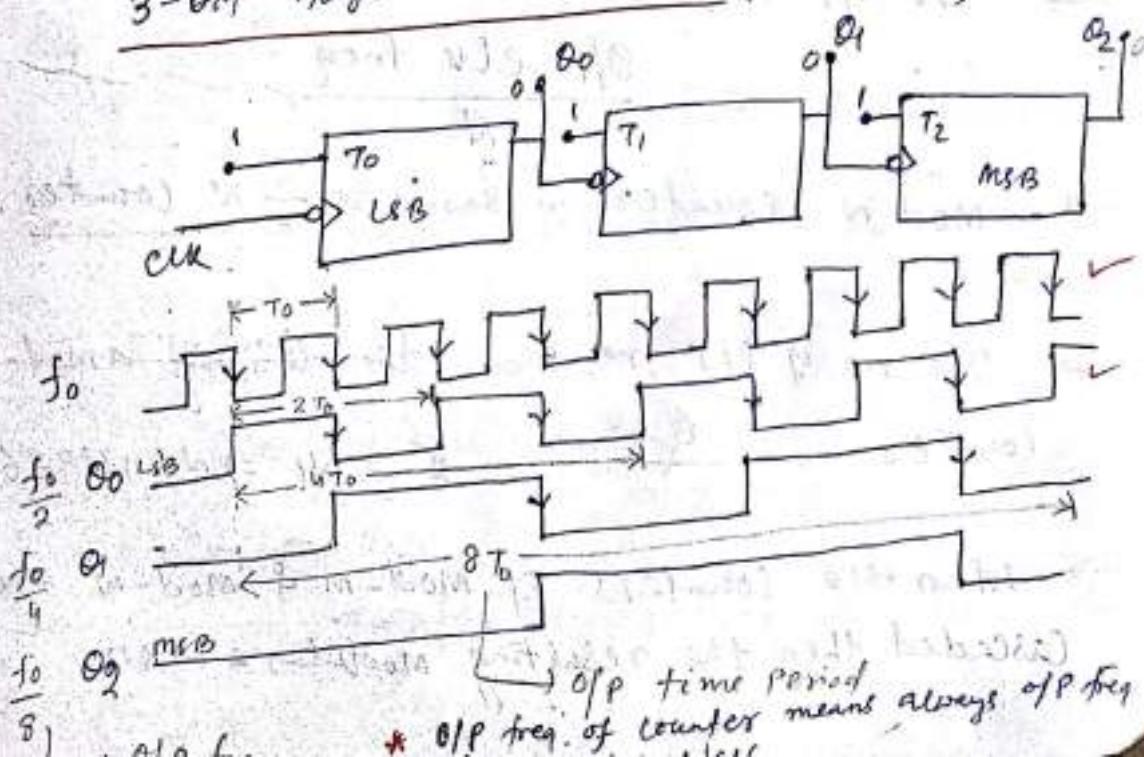
Asynchronous Counter  
or Ripple Counter  
or Serial Counter

Synchronous Counter  
or Parallel Counter.

### A Synchronous Counter

The o/p of one flip flop is connected to the clock i/p of the next flip flop and so on.

### 3-bit Asynchronous Counter (up Counter)



→ Off time period consists of 8 clock time period

$$\rightarrow \text{Off freq} = \frac{\text{Off clk freq}}{8}$$

→ therefore the previous ckt is known as

divided by 8 ckt b'cse the off freq is getting divided by the off freq by 8. so it is called as mod-8 counter.

→ modulus of a counter is the no. of states that can be generated by counter.

Ex Mod- $N$  counter means it has  $= N$  off states.

→ n bit counter has max $^n$   $2^n$  off states.

→ No. of f/f required for Mod- $N$  counter  
 $= \log_2 N$ .

→ The off freq. of a Mod- $N$  counter

$$= \frac{\text{Off clk freq}}{N}$$

→ Mod- $N$  counter is known as  $\frac{1}{N}$  counter.

→ The no. of f/f required for designing a mod- $N$  counter is  $\frac{6}{N}$ .  $2^6 = 64 \Rightarrow$  no. of f/f = 6.

→ When two counters of mod- $m$  & mod- $n$  are cascaded then the resulting modulus =  $m \times n$ .

→ Propagation delay of Asynchronous Counter ( $T_{pd}$ ) is given by

$$T_{pd} = N * T_{f/f} + t_{combination}$$

$$N = \text{No. of f/f}$$

→ Propagation delay of Synchronous Counter ( $T_{pd}$ ) is given by

$$T_{pd} = T_{f/f} + t_{combination}$$

\* Hence Synchronous Counter more faster than Asynchronous Counter

→ Twisted Ring & Ring Counter belong to Shift Counter (Synchronous Counter).

→ Free running modulus of Ring Counter & Twisted Ring (Johnson's Counter) are

✓ Ring counter  $\Rightarrow$  Mod-N

✓ Twisted ring counter  $\Rightarrow$  Mod- $2N$ .

✓ Normal counter  $\Rightarrow 2^N$ .

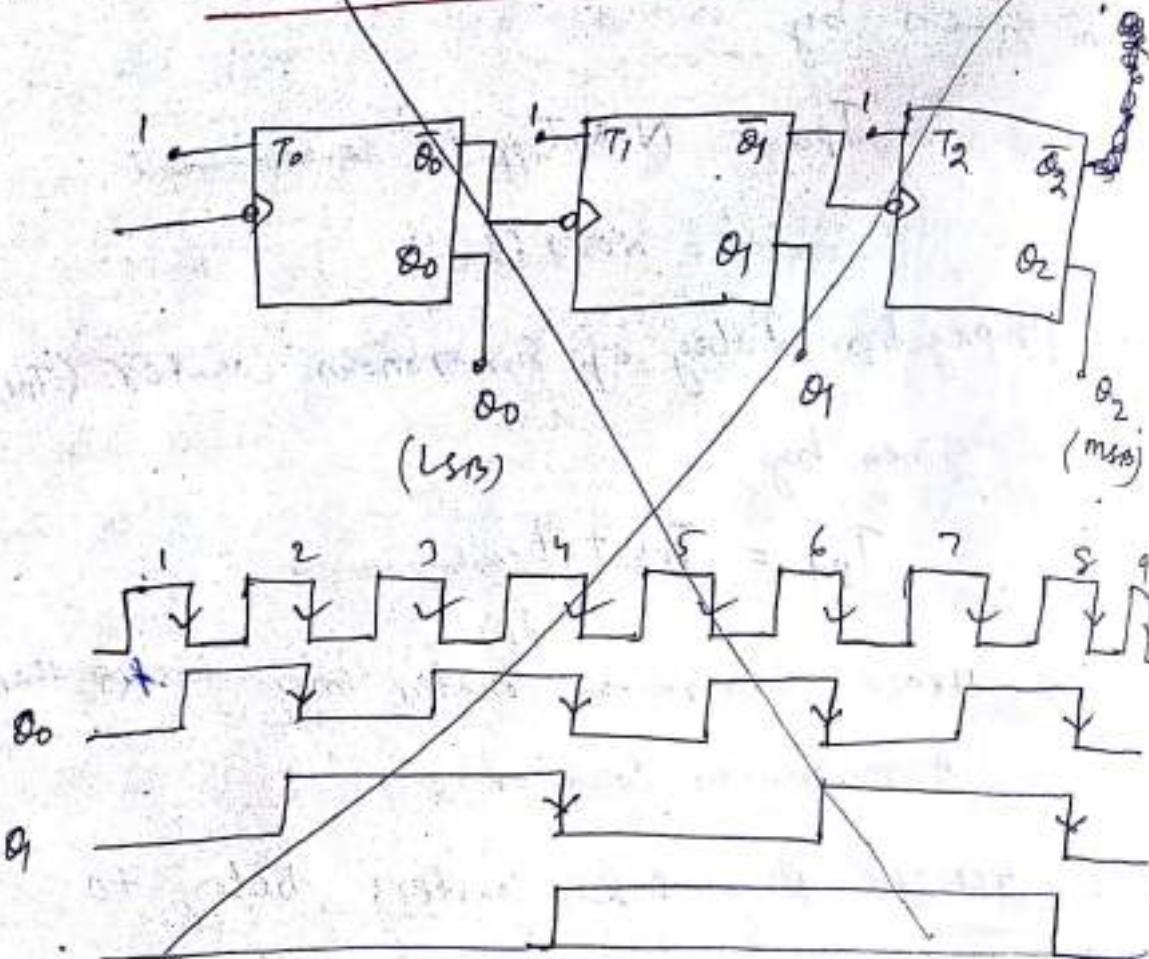
→ No. of f/f required for mod-N counts

For Normal Counter =  $\log_2 N$ .

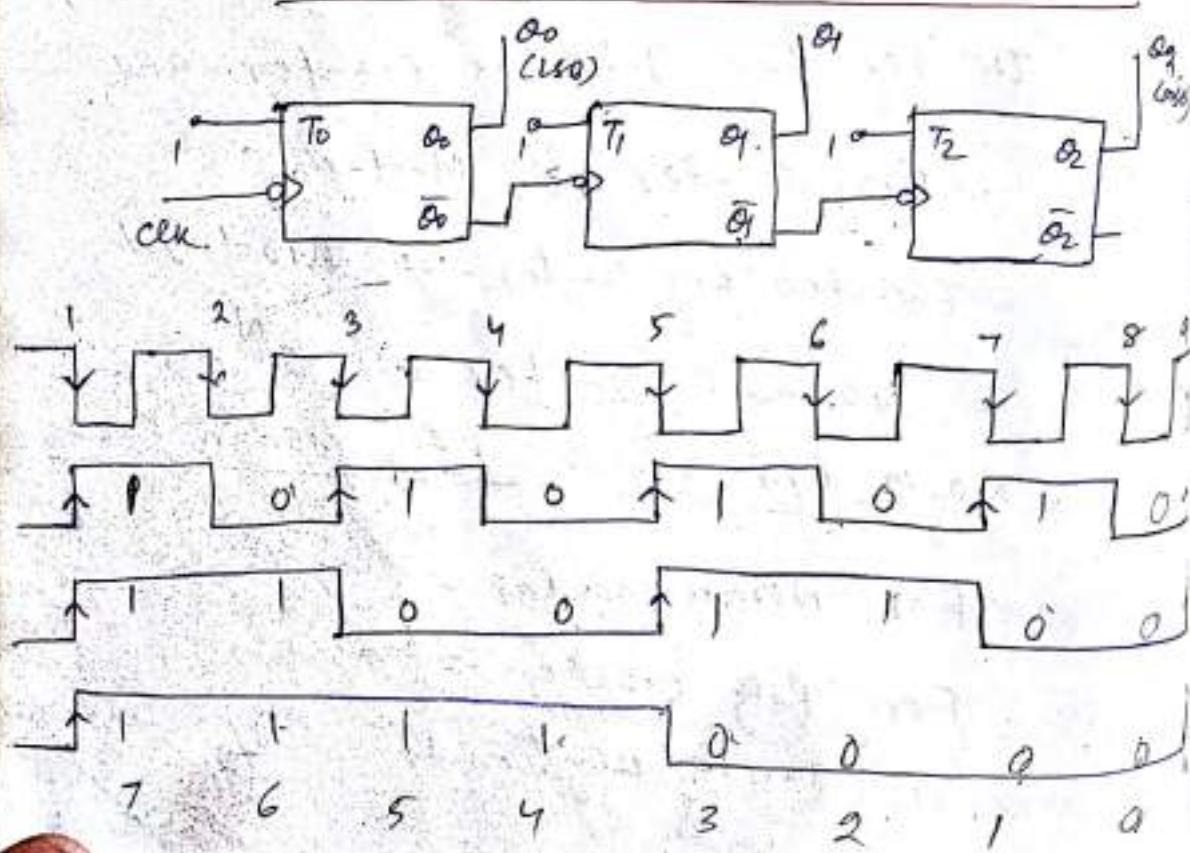
For Ring Counter =  $n$

ii Twisted Ring Counter =  $N/2$ .

3-bit Asynchronous Down Counter

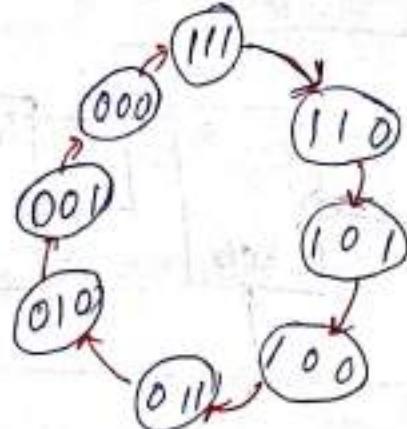


- 3-bit Asynchronous Down Counter:



→ If any counter contains all the states in called as a binary counter.

✓ starting pt.

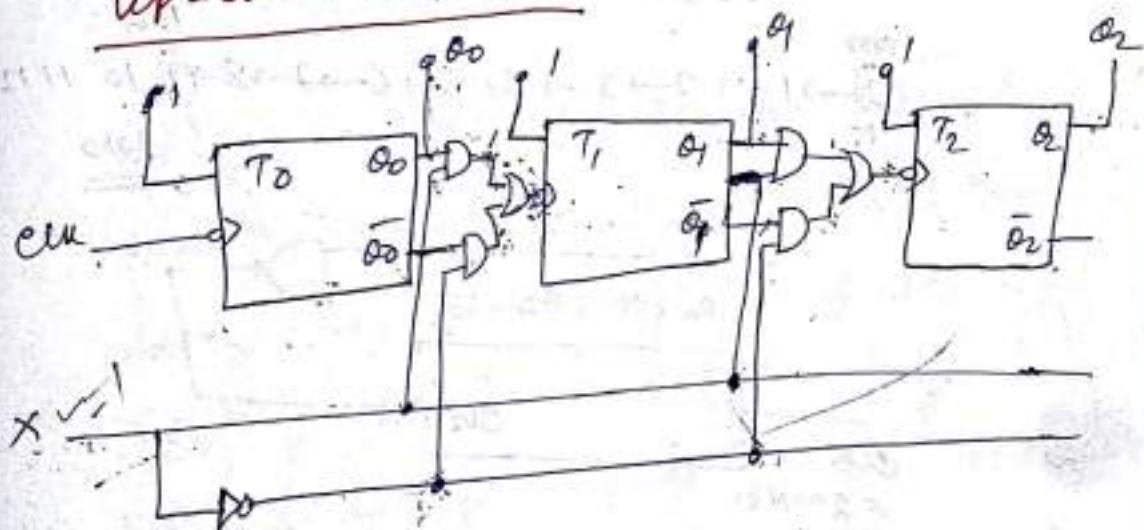


$$\text{Mod-8} = \frac{1}{8}$$

counter.

= 3 bit binary counter.

### Up-down counter

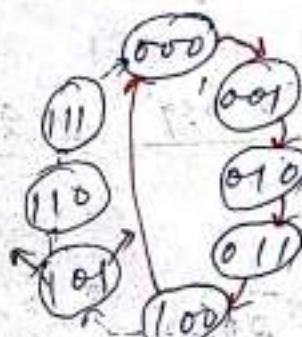


(b)

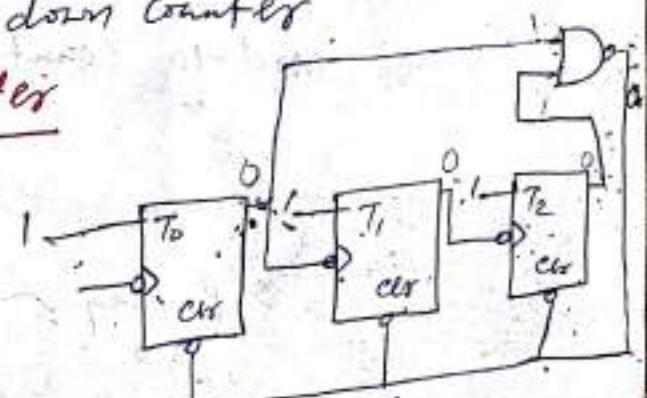
When  $X = 1$ , up counter.

& when  $X = 0$ , down counter.

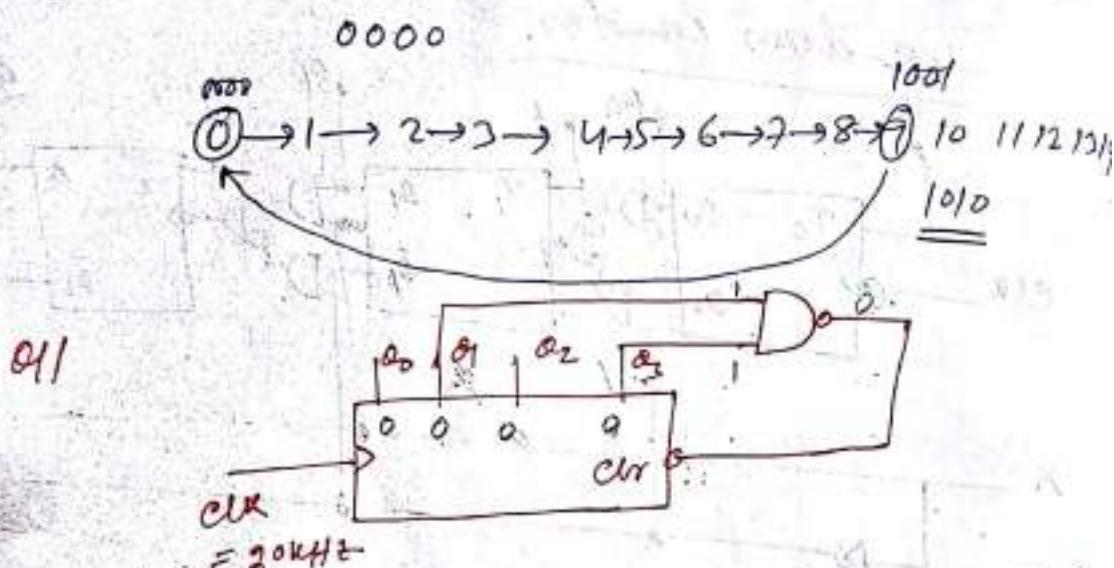
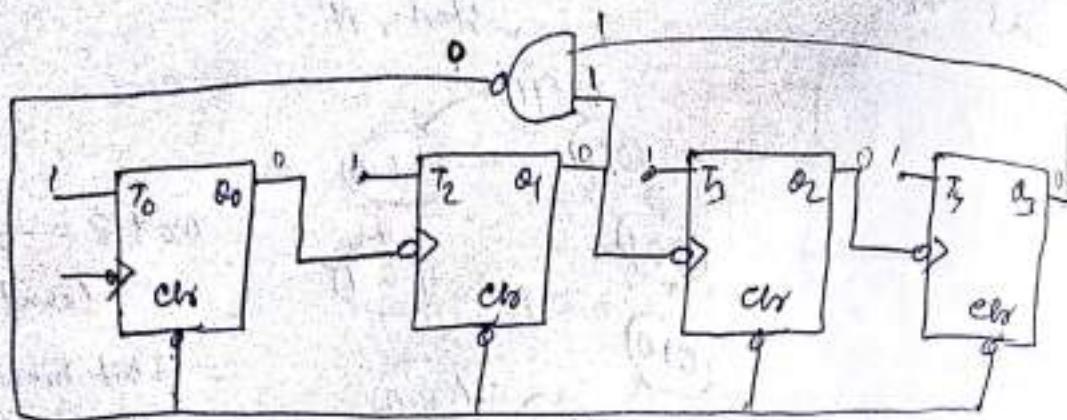
### Mod-5 counter



When  $m30=1$ ,  $l30=1$  then



## Mod-10 Async. counter (decade counter)



What it represents, what will be output.

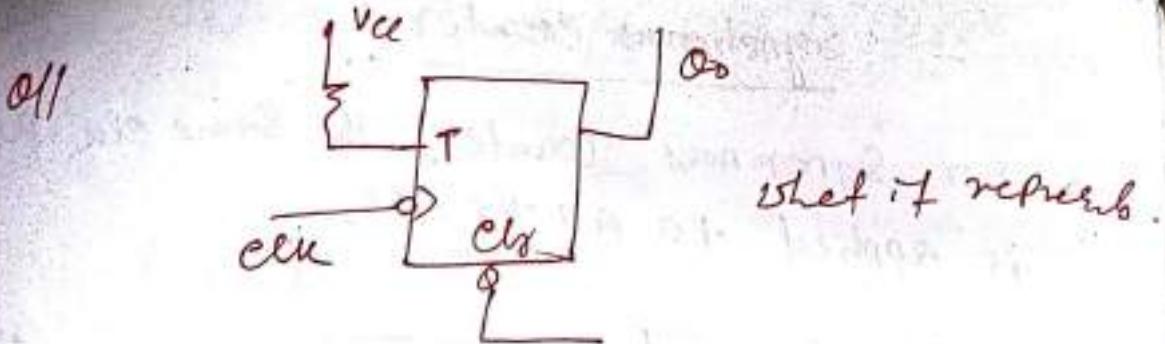
Ex 9

Mod-10 Counter = decade counter

=  $\frac{1}{10}$  Counter

$$\text{off. freq} = \frac{\text{clk freq}}{10} = \frac{20\text{kHz}}{10} = 2\text{kHz}$$

$$\text{Off time period} = \frac{1}{2\text{kHz}}$$



What it represents.

Sol? Mod-2 Counter =  $\div 2$  Counter

→ For Asynchronous Counter it has no specific design procedure.

→ There is a limit to the highest operating freq. In a ripple counter the delay time are additive & the total setting time for the counter is approximately equal to the product of propagation delay & no. of flfs.

$$\text{Total setting time} = P.d \times N$$

where  $N = \text{no. of flfs}$ .

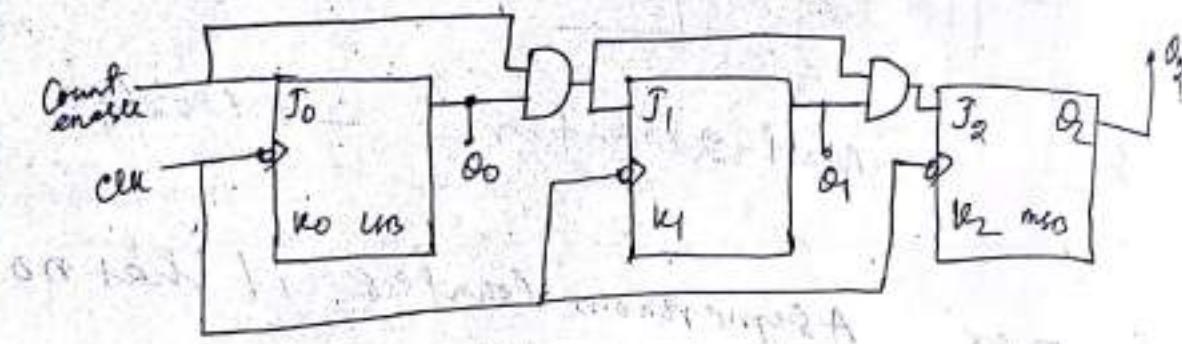
$$\text{Highest clk freq} \leq \frac{1}{N \times P.d}$$

$$(f_{cl})_{\text{highest}} = \frac{1}{N \times (P.d_{\text{prop}} + P.d_{\text{setup}})}$$

→ This can be overcome by Synchronous Counter

## - : Synchronous Counter :-

→ for Synchronous Counter, the same clk pulse is applied to all the P/Fs.



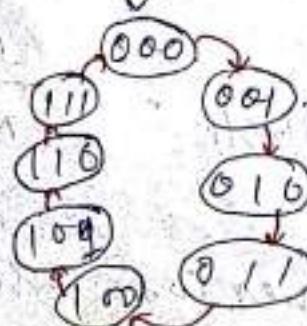
clk	Q0	Q1	Q2	
-	0	0	0	$J_0 = k_0 = 1, \quad J_1 = k_1 = 0, \quad J_2 = k_2 = 0$
<del>clk</del>	1	0	1	$J_0 = k_0 = 1, \quad J_1 = k_1 = 1, \quad J_2 = k_2 = 1$
<del>clk</del>	0	1	0	$J_0 = k_0 = 0, \quad J_1 = k_1 = 0, \quad J_2 = k_2 = 0$

Total setting time for the Synchronous Counter is the P.d. through 1 P/F. Also all the clks are same for all.

The max<sup>m</sup> clock freq is  $\frac{1}{\text{P.d of } P/F + P.d}$

Q11 Design mod-8 Syn. Counter by using TFL

Step 1

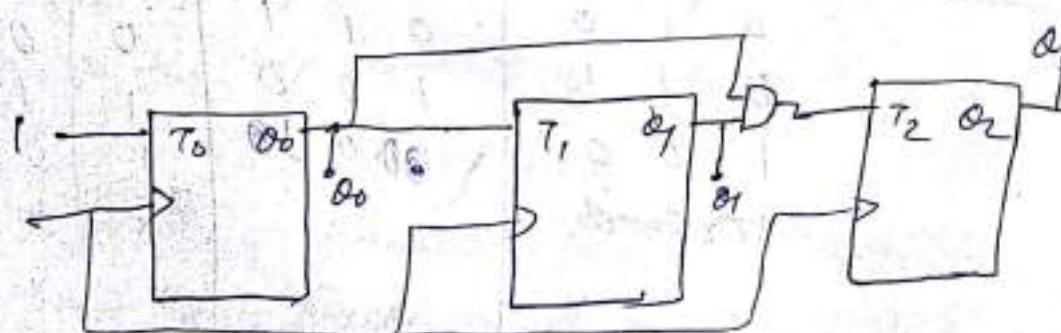
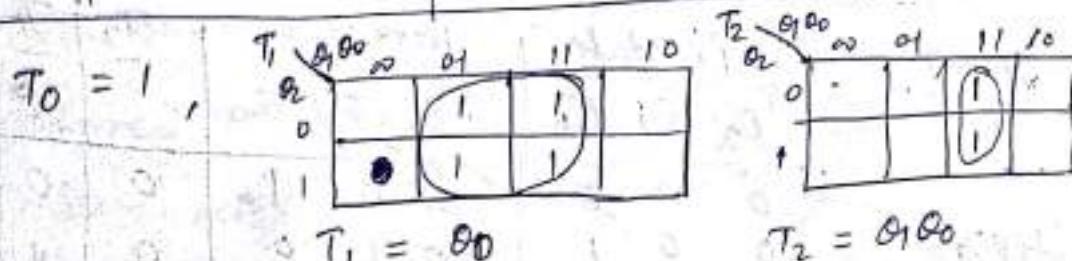


~~Step 2~~ Take the ~~exact~~ EXCitation table for T F/F

<u><math>D_n</math></u>	<u><math>D_{n+1}</math></u>	T
0	0	0
0	1	1
0	0	1
1	1	0

EXCitation matrix

Present State ( $D_n$ )	Next State ( $D_{n+1}$ )			<u><math>T_2</math></u>	<u><math>T_1</math></u>	<u><math>T_0</math></u>
	<u><math>D_2</math></u>	<u><math>D_1</math></u>	<u><math>D_0</math></u>			
0	0	0	0	0	0	1
0	0	1	0	0	1	1
1	0	1	1	0	0	1
1	1	0	0	1	1	1
1	1	1	1	1	0	1
0	1	0	0	0	0	1
1	0	1	1	1	0	1
1	1	0	1	0	1	1
0	1	1	0	0	0	1
1	1	1	1	0	0	1

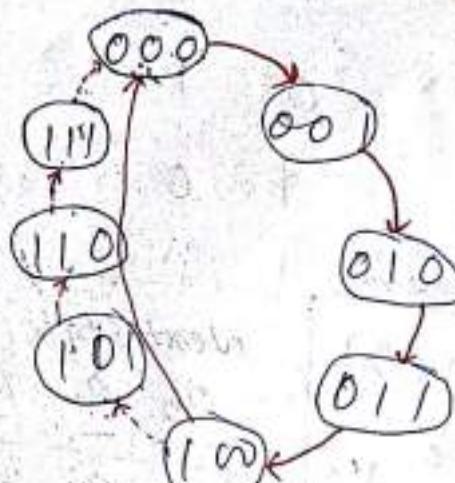


Q11-2 Design 3-bit syn. counter by using  
 a) SR-PF (b) J-K PF (c) D PF.

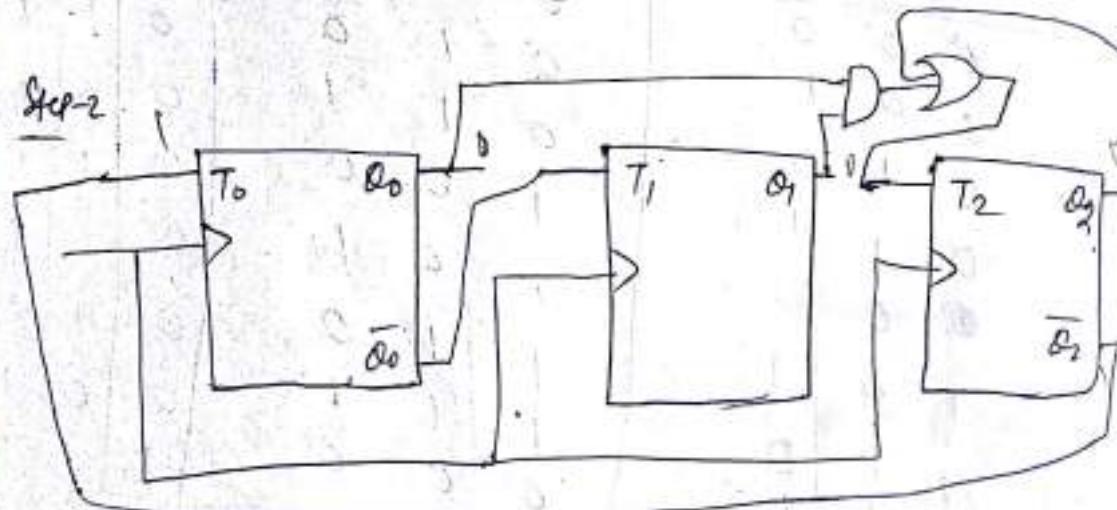
Q11-3 Design syn. mod-5 Counter.

Step-1

Step-1



Step-2



Step-3

<u>Step-3</u>	Pr. state			Next state			T <sub>0</sub>	T <sub>1</sub>	T <sub>2</sub>
	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>			
000	0	0	0	0	0	1	0	0	1
000	0	0	1	0	1	0	0	1	1
000	0	1	0	0	1	1	0	1	1
000	0	1	1	1	0	0	0	0	1
000	1	0	0	0	0	0	1	0	0
000	1	0	0	0	0	0	x	x	x
000	1	0	0	0	0	0	x	x	x
000	1	0	0	0	0	0	x	x	x

		0	1	11	10
		0	1	X	X
T <sub>2</sub>	0	.	.	0	-
	1	(1)	X	(X)	X

$$T_2 = \bar{Q}_2 + Q_1 \bar{Q}_0$$

		0	1	11	10
		0	1	11	10
T <sub>1</sub>	0	.	0	-	-
	1	(1)	(1)	-	-

$$T_1 = \bar{Q}_0$$

		0	1	11	10
		0	1	11	10
T <sub>0</sub>	0	0	0	-	-
	1	0	X	X	X

$$T_0 = \bar{Q}_2$$

→ In this design there is a problem, when ~~present~~ previous state may be 101, off can be any and also it will land on any one of the unused state (101, 110, 111). This problem is called as lockout problem.

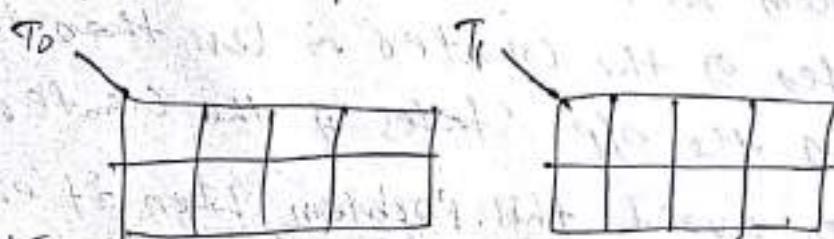
→ When the counter enters into one of the unused state & if it will never come back into the used state, it is referred to as lockout problem, this problem will occur when the no. of off states of the counter is less than the max. no. of the off states of the counter.

→ To avoid this problem, when it enters into the unused state the counter must be forced back to one of the used states.

To eliminate the lockout problem  
 (1) when it lands on any unused state  
 then applying by clock pulse we have  
 to enter into used state.

(2) To eliminate the lockout problem,  
 put the <sup>All</sup> next state off = 000 for  
 unused state of present state. Then only

<u>present state</u>	<u>next state</u>	<u><math>T_2</math></u>	<u><math>T_1</math></u>	<u><math>T_0</math></u>
000	001	0	01	
001	010	0	11	
010	011	0	01	
011	100	1	11	
100	000	1	00	
101	001	1	01	
110	010	1	10	
111	000	1	11	

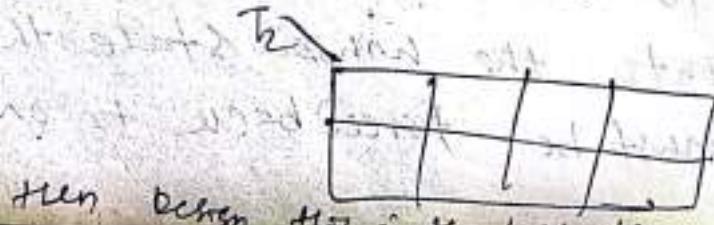


$$T_0 = \underline{\quad}$$

$$T_2 = \underline{\quad}$$

$$T_1 = \underline{\quad}$$

$$T_1 = \underline{\quad}$$



Note lockout problem will appear in the counter only if the counter is having at least one unused state.

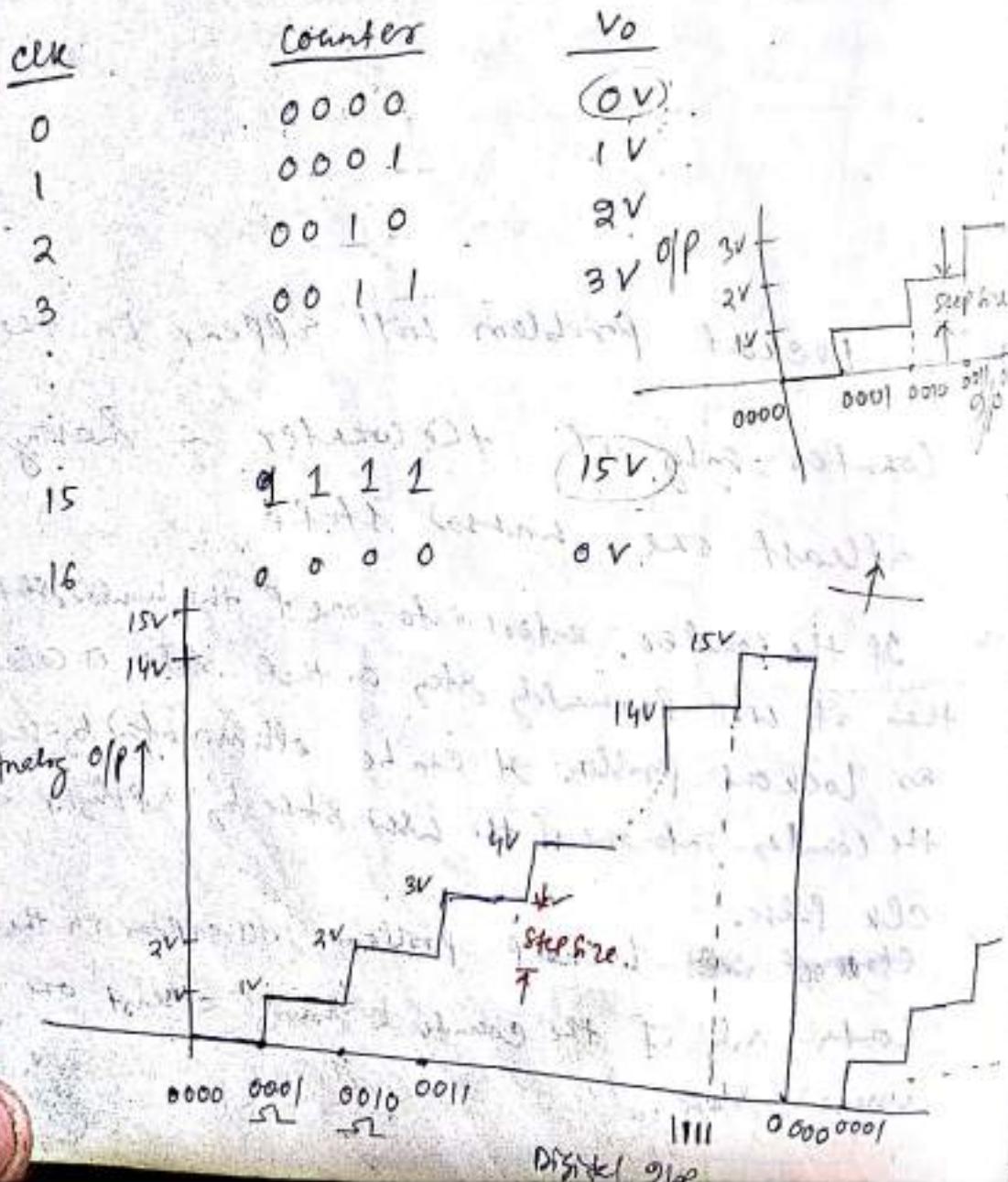
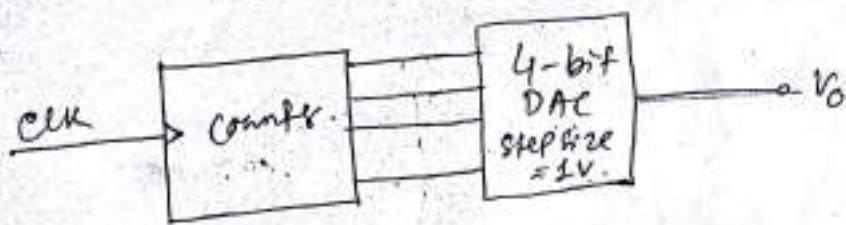
→ If the counter enters into one of the unused state then it will permanently stay in that state is called as lockout problem. It can be eliminated by sending the counter into one of the used state by applying a clk. pulse.

→ ~~lockout~~ lockout problem will appear in the counter only if the counter is having at least one unused state.

## Data Converters

1. Digital to Analog converter (DAC)
2. Analog to Digital converter (ADC)

### 1. DAC



1. No. of Steps = 15.

For n-bit DAC the no. of steps =  $2^n - 1$ .

2. Full Scale O/P [FSO]

Step size =

FSO = No. of steps  $\times$  Step size.

$$= (2^n - 1) \times \text{Step size}.$$

where  $n$  = size of the DAC.

Resolution:

$\Delta$  is the smallest possible change at the O/P of the DAC for any change in the i/p.

So, Resolution = Step size (i.e. in terms of voltages)

$$\Delta = 1V \text{ (here)}$$

$$\underline{\% \text{ Resolution}} = \frac{\text{Step size}}{\text{Full scale off}} \times 100$$

$$\therefore \frac{1}{(2^n - 1)} \times 100$$

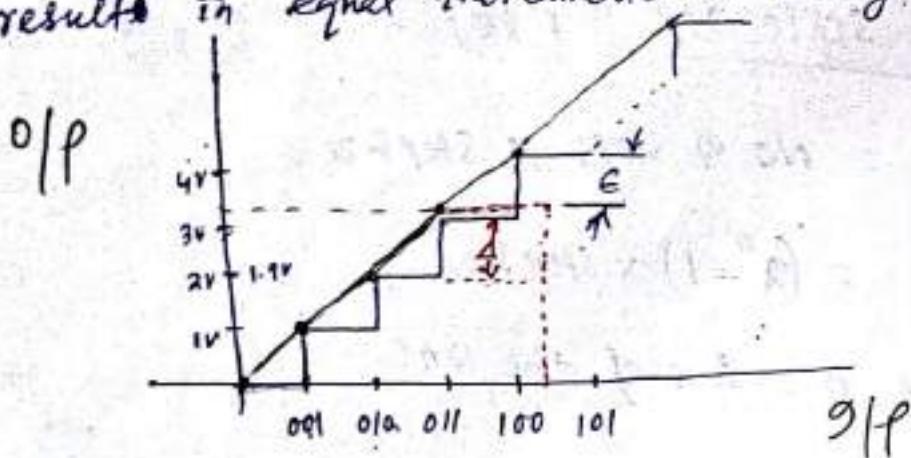
$$= \frac{1}{2^n - 1} \times 100 \%$$

resolution (clarity)

The smallest O/P that we can detect with certainty or clarity is called as resolution.

- Linearity :-

For Linearity, Equal increments in the digital i/p should result in equal increments in analog o/p.



In actual case, the i/p vs. o/p relation is not linear. So

→ The difference bet<sup>n</sup> actual o/p voltage & measure o/p voltage is referred as linearity error ( $E$ )

$$|E| < \frac{1}{2} \Delta \quad (\text{no linearity error})$$

$$|E| > \frac{\Delta}{2} \quad (\text{linearity error})$$

- Accuracy :- (Actual o/p voltage - expected o/p voltage)  
The accuracy of DAC is a measure of the difference bet<sup>n</sup> the actual o/p voltage & the expected o/p voltage.

→ It is specified as a percentage of full scale or max<sup>m</sup> o/p voltage.

Q11 A DAC with 10V. Full scale o/p voltage & an accuracy of  $\pm 0.2\%$ , then the max<sup>m</sup> error of any o/p voltage will be \_\_\_\_.

Ans? F.S.O = 10V. , Accuracy =  $\pm 0.2\%$ .  $\therefore \text{Error} = \pm 0.2\% \times 10 = \pm 0.2$

$$\begin{aligned} \text{max}^m \text{ error} &= 102\% \times 10V \\ &= \frac{0.2}{10} \times 10 = 0.02V = 20mV \text{ (Ans)} // \end{aligned}$$

Q11 How many bits are required at the o/p of DAC if it is necessary to resolve voltage to  $5mV$  & DAC has  $10V$  Full scale o/p voltage.

$$\begin{aligned} \underline{\text{Soln}} \quad n = ? & \quad \text{Resolution} = \text{Step size} = 5mV \\ \text{Step size result} = 5mV & \quad \% \text{ Resolution} = \frac{FSO}{(2^n - 1)} \\ FSO = 10V & \end{aligned}$$

$$\begin{aligned} \underline{\text{Soln}} \quad n = ? & \quad \text{Resolution} = \frac{\text{Step size} \times 100}{FSO} \\ \% \text{ Resolution} & = \frac{5mV \times 100}{(2^n - 1) 5mV} \end{aligned}$$

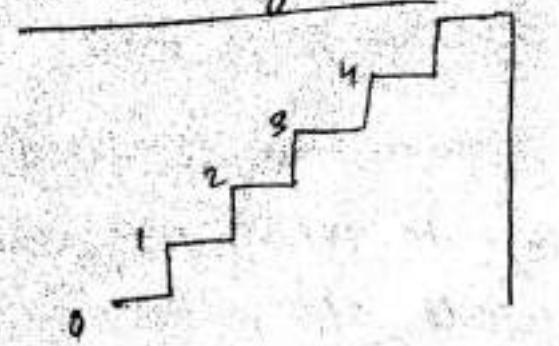
$$\begin{aligned} \text{Full scale o/p} &= (2^n - 1) \times \text{Step size} \\ \frac{10V}{5mV} &= (2^n - 1) \times 10^3 \end{aligned}$$

$$\begin{aligned} \text{Accuracy granted} &\Rightarrow 2000 + 1 = 2^n & FSO = 10 \\ \text{Actual error} &\Rightarrow 2^n = 2001 & \text{Accuracy} = 1\% \\ \text{Max error} &= 1 \times 10^3 \times 10^{-3} = \frac{1}{100} \times 10 = 10mV \end{aligned}$$

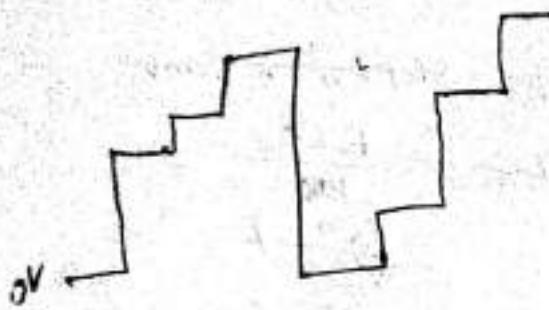
Q11 For the above problem what happens if accuracy is 1%.

$$\begin{aligned} \underline{\text{Soln}} \quad FSO = 10V, \text{ Accuracy} = 1\% &= \frac{1}{100} \\ \text{max}^m \text{ error} &= 1\% \times 10 = \frac{1}{100} \times 10 = 0.1V = 100mV \end{aligned}$$

### monotonicity test



monotony



Not monotony.

For a Proper monotonicity the steps of the staircase waveform must be equally spaced & of exact same amplitude. Any deviation in staircase waveform will specify the "MAL" functioning of DAC.

### Setting time:

If the msb is 1 (100) & remaining bits are '0' ~~remaining~~ it represents the 1/2 of the full scale  $t_f$ .

$t_f \Rightarrow$  Full Scale off voltage

= Full Scale off voltage - resolution.

Q11 What is the resolution of a 8 bit DAC whose max<sup>m</sup> full scale o/p voltage range 0 to 10V.

Sol<sup>n</sup>  $n = 8$ .  $FSO = 10 - 0 = 10V$ .  $FSO = \frac{10}{2^8} = \frac{10}{256}$

Resolution = Step size =  $\frac{10}{2^8} = 0.039$ .  
 $Step size = \frac{10}{2^8} = \frac{10}{256}$

→ if the LSB bit is 1 & remaining bits are zero '0', it refers <sup>as</sup> the resolution.

Q11 What is the resolution of a 8-bit ADC whose operating voltage range is 0 to 10V.

Sol<sup>n</sup> Resolution = Step size =  $\frac{10 - 0}{2^8} = \frac{10}{256}$

Q11. What is the resolution of a 8-bit ADC whose operating voltage range is -10V to 10V.

Resolution =  $\frac{10 - (-10)}{2^8} = \frac{20}{256} (Ans)$

Q11 A 5-bit DAC has a current off. When the digital i/p is 10000, the off current is 8mA. Find the off current of digital o/p is 11111.

Sol<sup>n</sup>  $(10000)_2 = (16)_{10}$   $16 \times Step size = 8mA$   
 $Step size = \frac{8mA}{16} = 0.5mA$

$\Rightarrow Step size = \frac{8mA}{16} = 0.5mA$   
 $(11111)_2 = (31)_{10}$ ,  $O/P = 31 \times 0.5 = 15.5mA (Ans)$

Q11 A D/A converter gives an O/P of 3.8V  
 for a digital i/p of 10011. Find the step size  
 what is the o/p if digital i/p is 11111.  
 $19 \times \text{step size} = 3.8$

Soln  $(10011)_2 = (19)_{10}$  Step size  
 $19 \times \text{step size} = 3.8$  ✓  
 $\Rightarrow \text{step size} = \frac{3.8}{19} = 0.2 \text{ V.}$

$$(11111)_2 = (31)_{10}$$

$$\text{o/p} = 31 \times 0.2 = \cancel{6.2} \text{ V.} \quad (\text{Ans})$$

Q11 An 8-bit D/A converter has a step size of 6mV. Find full scale o/p voltage & percentage resolution.

Soln  $n = 8$ , Step size = 6mV.  
~~Full Scale Output =  $(2^8 - 1) \times \text{Step size}$~~   
~~Full Scale Output =  $255 \times 6 \text{ mV} = 1.53 \text{ V.}$~~   
~~1. % Resolution =  $\frac{\text{Step size}}{\text{F.S.O.}} \times 100$~~   
 $= \frac{6 \text{ mV}}{255 \times 6 \text{ mV}} \times 100$   
 $= 0.392\%$ .

Q11 In a stair step A/D converter, clock freq. is 1MHz.  
 the threshold voltage  $V_T$  is 0.1mV. the full scale  
 o/p of D/A converter (having 10bit i/p) is 5.115V.

Find (a) digital equivalent of analog i/p of 2.693V.

b) conversion time.

c) Resolution.

$$FSO = 5.115V$$

$$(2^{10}-1) \text{ Step} = 5.115$$

Step Total no. of steps  $= 2^{10} - 1 = 1023$

$$\text{Step size} = \frac{5.115}{1023} = 5mV$$

Step B

a)  $(x)_2 = (y)_{10}$

$$y \times (\text{Step size}) = 2.693$$

$$y = \frac{2.693}{5 \times 10^{-3}} = 524.6 \approx 525$$

$$(525)_{10} = (x)_2 = (10000 \overset{1106432}{0} \overset{163421}{1} 10001)_2 \\ = (10000011001)_2$$

b) Total 525 clock pulses are required

to get complete conversion.

$$\text{clock freq } f = 1MHz$$

$$\Rightarrow \text{Time of one clk pulse} = 1\mu\text{sec}$$

$$\text{Total conversion time} = 525 \mu\text{sec}$$

c) Resolution = Step size = 5mV.

$$\text{y. Resolution} = \frac{\text{Step size}}{FSO} \times 10^3 \\ = \frac{1}{2^{10}-1} \text{ V/mv} = 0.09775 \text{ V. (Ans)}$$

Q11 An 8 bit successive approximation A/D converter has a resolution of 10mV. If analog input is 1.592, find the digital output.

Solution

Resolution = 10mV = Step size.



$$n = 8$$

$$\text{Total no. of step} = 2^8 - 1$$

~~1.592 × 1000 / (2^8 - 1)~~

10mV

Step number, which gives an opp of 1.592

$$= \frac{1.592}{\text{Step size}} = \frac{1.592}{10 \times 10^{-3}} = 159.2 \approx 159$$

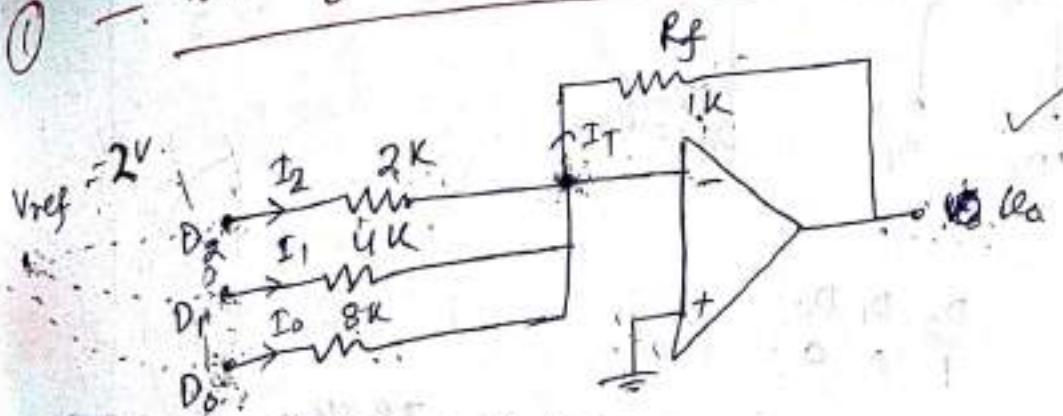
$$(159)_{10} = (1001111)_2$$

## DAC

Types of DAC

- (1) Binary weighted Register type DAC.
- (2) R-2R Ladder N/W.

① :- Binary weighted Register type DAC:-



→ for  $D_2 D_1 D_0 = 111$

$$I_T = I_0 + I_1 + I_2$$

$$= \frac{V_{ref}}{8K} + \frac{V_{ref}}{4K} + \frac{V_{ref}}{2K}$$

$$= \frac{V_{ref}}{1K} \left[ \frac{1}{8} + \frac{1}{4} + \frac{1}{2} \right] = 7/8 \text{ mA.}$$

$$V_o = -I_T R_f = -7/8 \text{ mA} \times 1K = -7/8 \text{ V.}$$

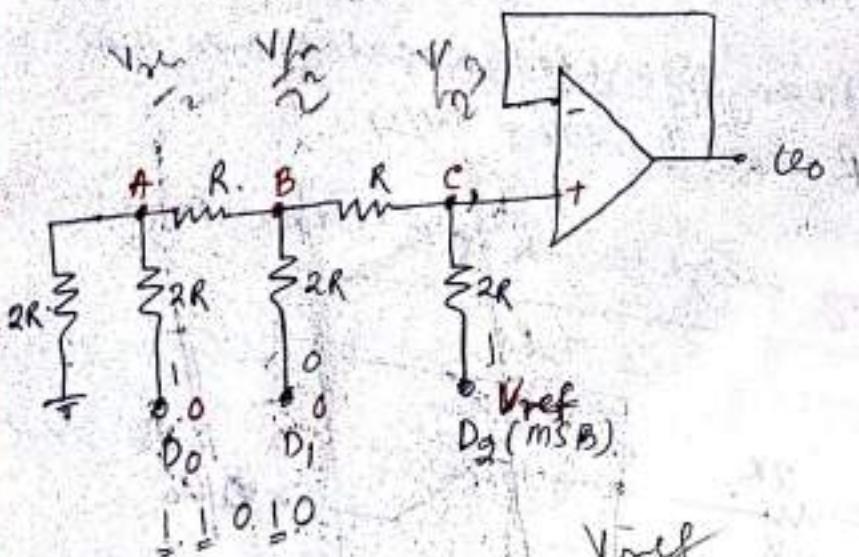
→ if  $D_2 = 1, D_1 = D_0 = 0$ .

$$I_T = \frac{V_{ref}}{2K} = 0.5 \text{ mA.}$$

Disadv:  
Requirement of various precision resistors i.e  
 $2K, 4K, 8K, 16K, \dots, 256K$  ---

②

## R-2R Ladder N/W DAC



Let

$$\begin{matrix} D_2 & D_1 & D_0 \\ 1 & 0 & 0 \end{matrix}$$

$$V_o = V_C = V_T = V_- = \frac{2R \times V_{ref}}{2R+2R} = \frac{V_{ref}}{2}$$

Let

$$\begin{matrix} D_2 & D_1 & D_0 \\ 0 & 1 & 0 \end{matrix}$$

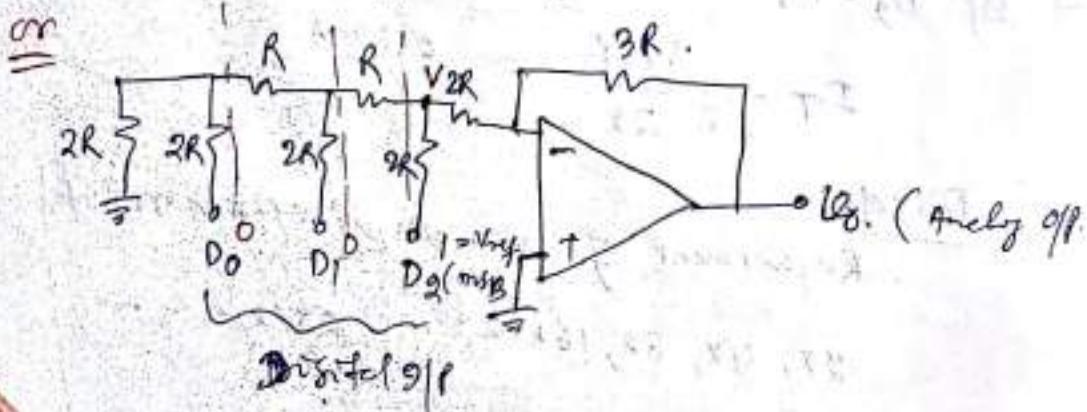
$$V_o = \frac{V_{ref}}{4}$$

Let

$$\begin{matrix} D_2 & D_1 & D_0 \\ 1 & 1 & 0 \end{matrix}$$

$$V_o = \frac{V_{ref}}{2} + \frac{V_{ref}}{4}$$

or



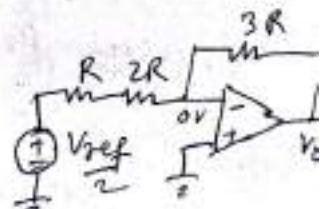
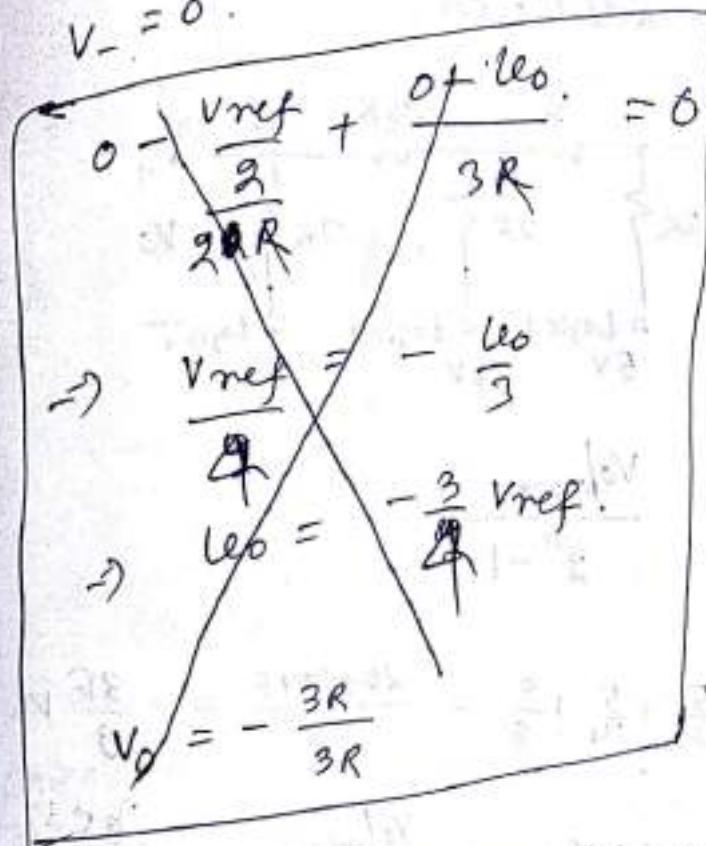
Digital off

Idea

$D_2$	$D_1$	$D_0$
1	0	0

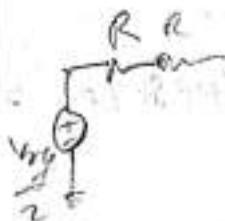
$$V = \frac{V_{ref}}{2}$$

$$V_- = 0$$



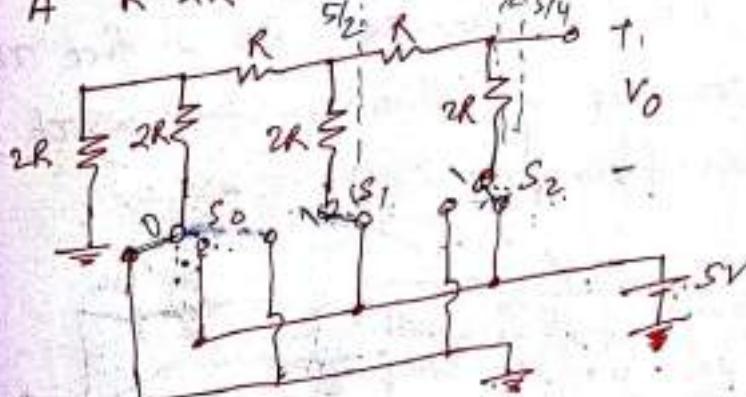
$$0 - \frac{V_{ref}}{2} + \frac{O - V_O}{3R} = 0$$

$$\Rightarrow V_O = -\frac{V_{ref}}{2}$$



$$2) V_O = -\frac{3R}{3R} \left( \frac{V_{ref}}{2} \right) = -\frac{V_{ref}}{2}$$

Q1) A R-2R ladder type DAC shown below,



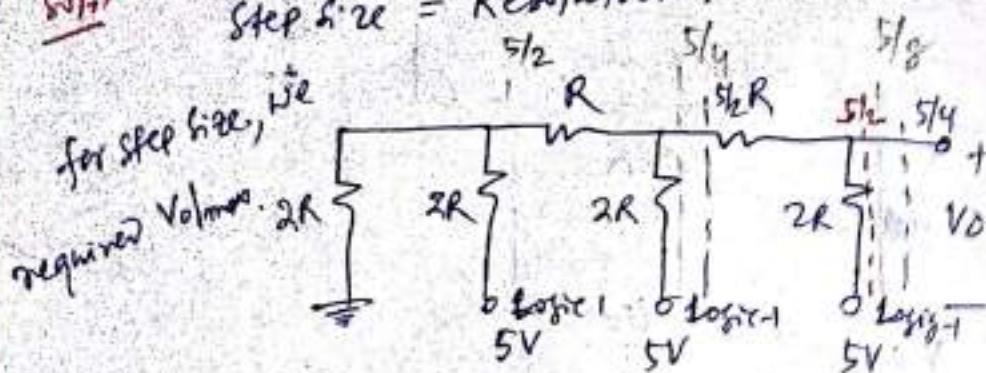
What is off voltage  $V_O$  for the switch states

$$S_0 = 0, S_1 = 1, S_2 = 1$$

$$V_+ = \Sigma + \Sigma = \frac{10 + 5}{4} = \frac{15}{4} V \text{ (Ans)}$$

~~Q1~~ Q1 What is the step size of the DAC ckt. Given in previous problem.

Soln? Step size = Resolution.



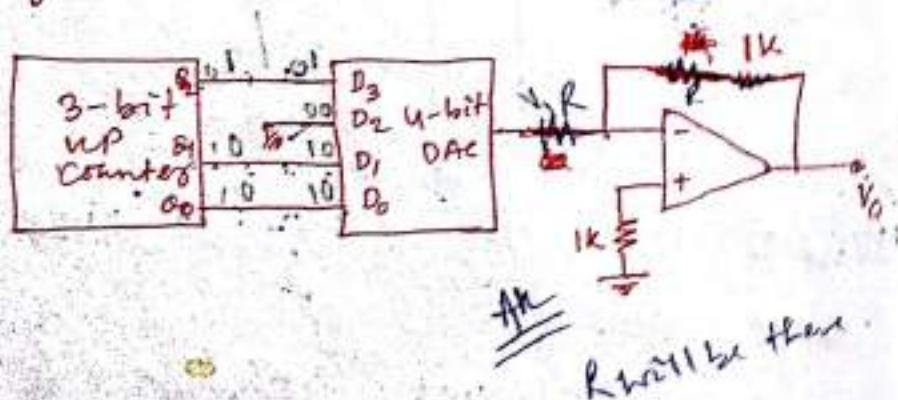
$$\text{Step size} = \frac{V_{o\max}}{2^n - 1}$$

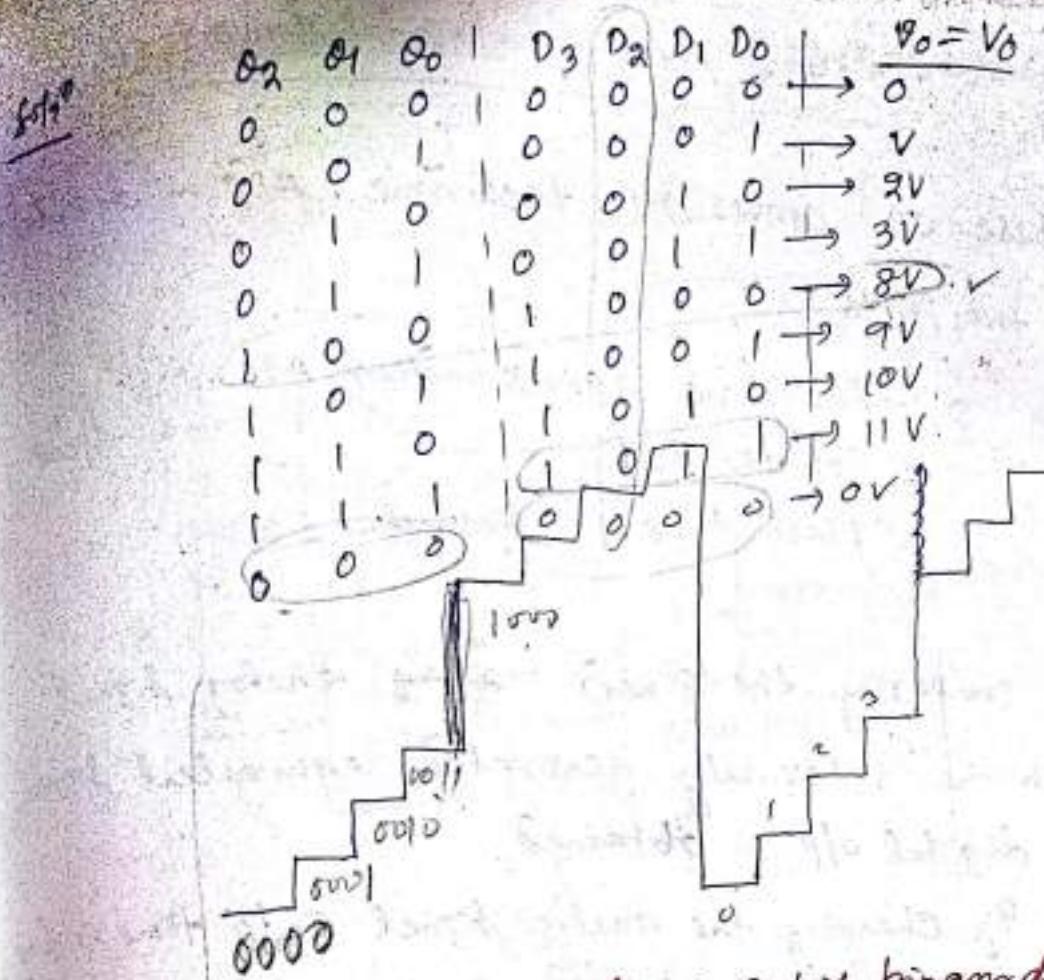
$$V_{o\max} = \frac{5}{2} + \frac{5}{4} + \frac{5}{8} = \frac{20+10+5}{8} = \frac{35}{8} V$$

$$\text{Step size} = \text{Resolution} = \frac{V_{o\max}}{2^n - 1} = \frac{\frac{35}{8}}{7} = \frac{5}{8} = 0.625 V$$

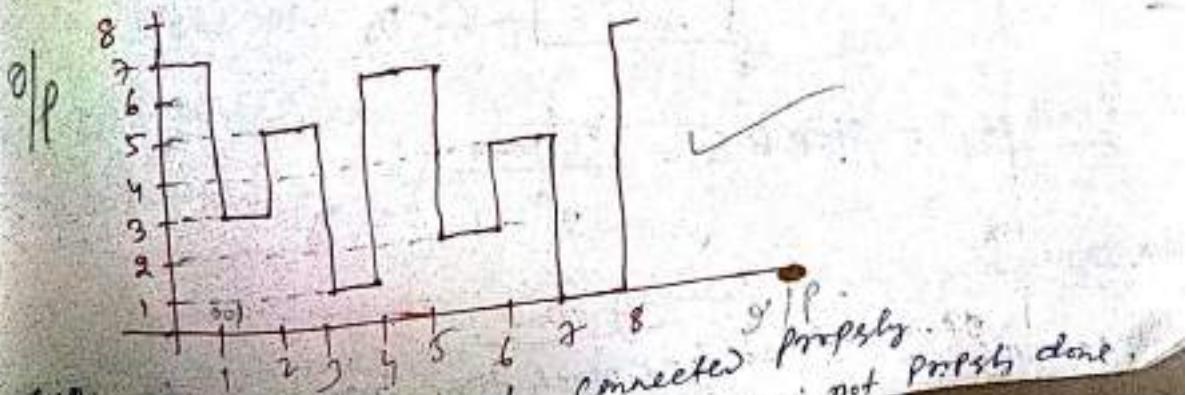
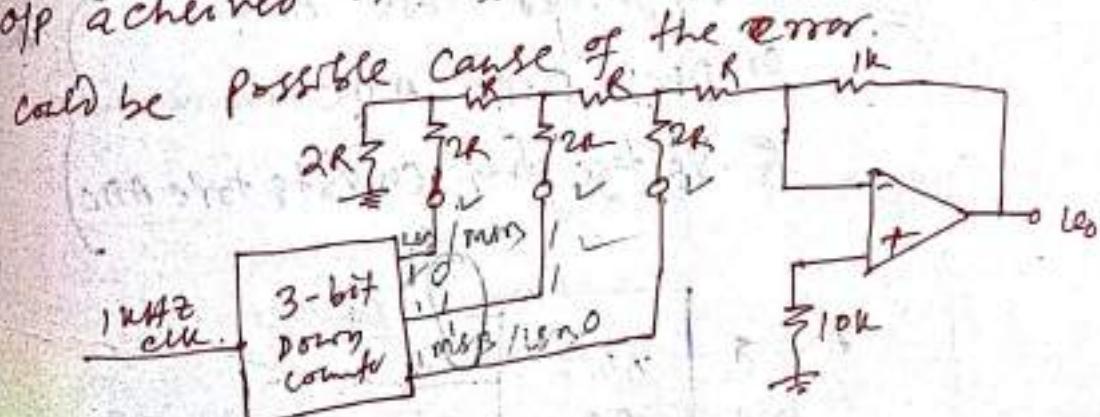
(Ans)

Q2 A 4-bit DAC is connected to a free running 3-bit upcounter as shown in fig. Which of the following waveform will observe at  $V_o$ .





Q. A student has made a 3-bit binary down counter connected to  $R=2R$  ladder type DAC as shown in fig. to generate a staircase waveform. The output achieved is different as shown in fig. What could be possible cause of the error.



## A DC (Analog to Digital Converter)

Base on conversion technique ADC is divided to two parts.

NT  $\stackrel{\text{I.}}{=}$  Successive approximation ADC.  
 Cont.  $= \left(\frac{n}{2}\right)T$  2. Counter type  
 3. flash type.  $\rightarrow$  fastest. = 0

Speed.

By comparing the given ~~analog~~ analog signal with the internally generated equivalent signal, the digital off is obtained.

By changing the analog signal into time or ~~new~~ and comparing these ~~less~~ parameters with the known values, the digital off is obtained.

II. 4. Integrator type ADC      Accur.  
 or Dual slope ADC  $\rightarrow$  more accurate.  
 5. Voltage freq. convert type ADC



$$\text{Resolution} = \text{Step size} = \frac{V_{\text{ref max}}}{2^n - 1}$$

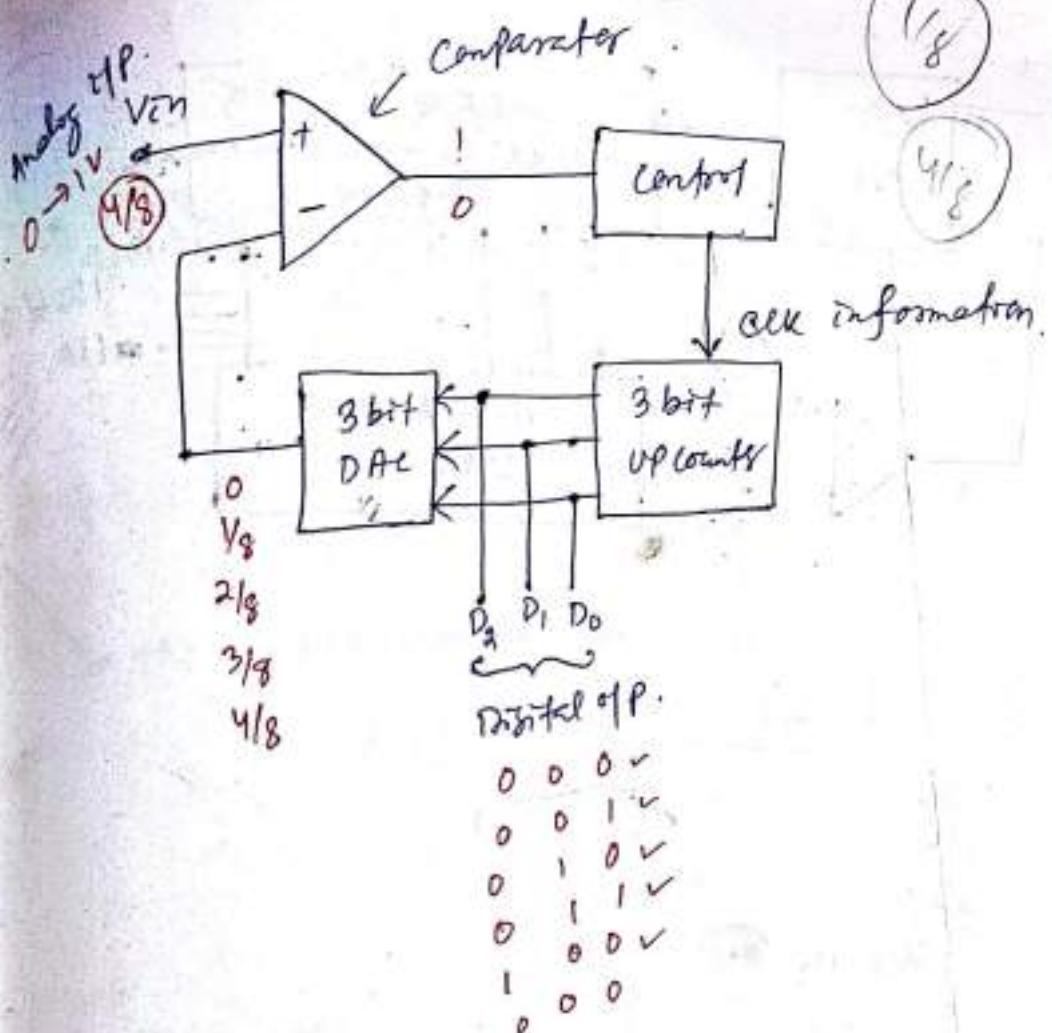
$$F.S.D = \frac{V_{\text{ref max}}}{2^n}$$

Max. Conversion time  $\approx$  Settling time :-

It is defined how much time required from off to all 1's.

$$\text{max. conversion time} = (2^n - 1) \times T, \text{ where } T = \text{clock period.}$$

### Counter type ADC

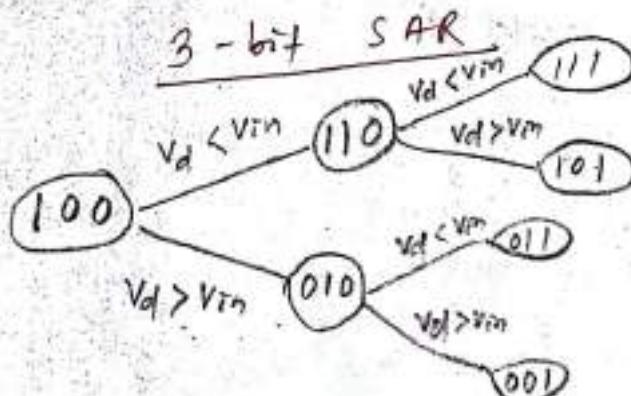
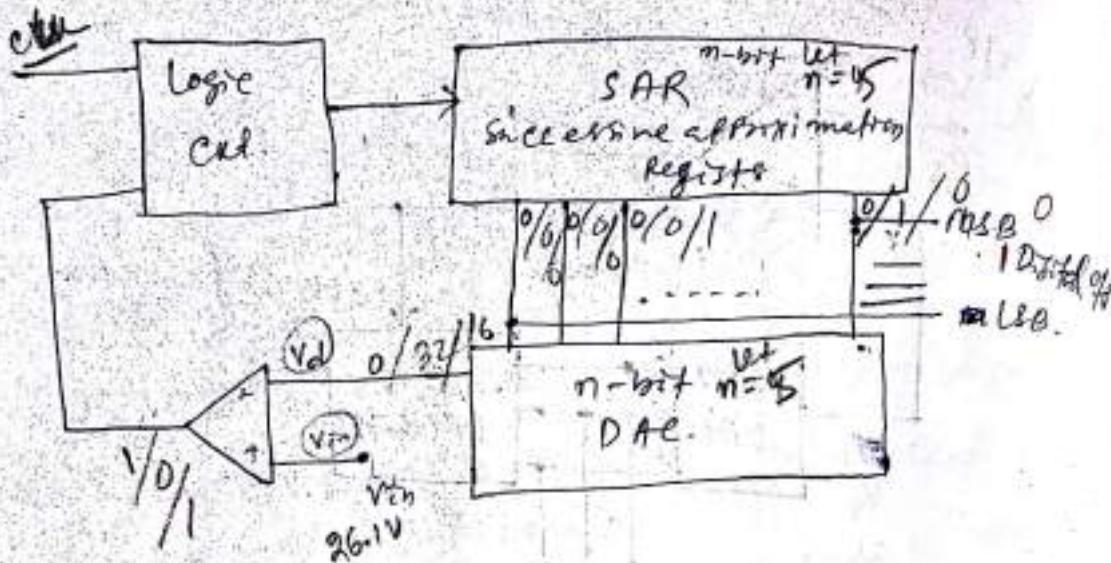


$$\text{max. conversion time} = (2^n - 1) T, \quad T = \text{clock period.}$$

Hence the max<sup>m</sup> conversion time is too large.  
So to reduce it we go for successive approximation type ADC.

## Successive approximation type ADC

→ Here the max conversion time =  $N \times T$ . (less).

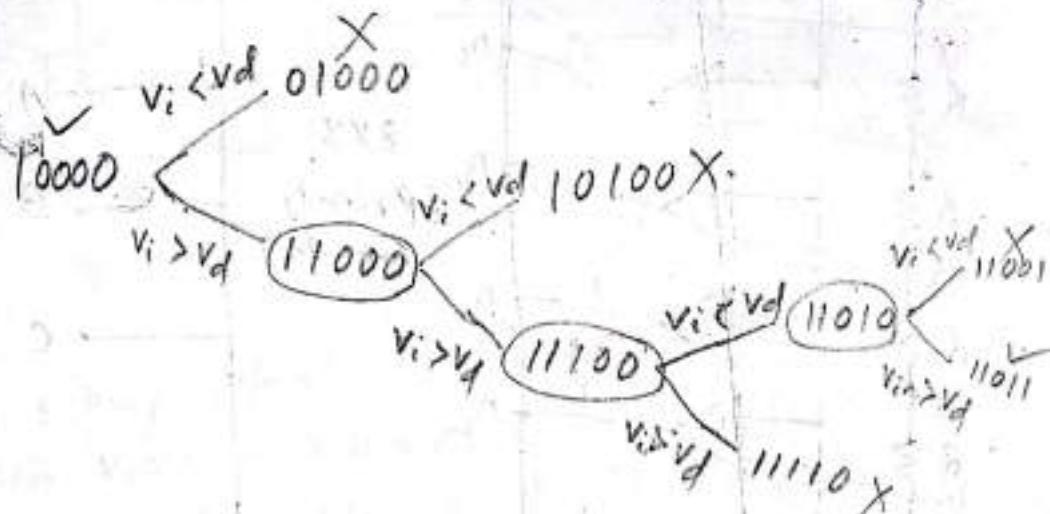


for 3-bit 3 UK Pulse req.

- $V_{in}$  = Analog signal. 1st set SA register to zero.
- then place '1' in MSB. This is fed to D/A converter whose o/p goes to Comparator. If o/p of Comparator is 1, then fixed '1' in the MSB position of Digital o/p. If 0, then fixed '0' in the 1st MSB position of Digital o/p.
- then place '1' in next MSB. Continue the process. . . .
  - If  $V_d > V_m \rightarrow$  o/p of comparator = 0
  - If  $V_d < V_m \rightarrow$  " " = 1.

Q11 The final value of a 5-bit SAR is 11011.  
What are the intermediate values.

Soln



So the intermediate values are

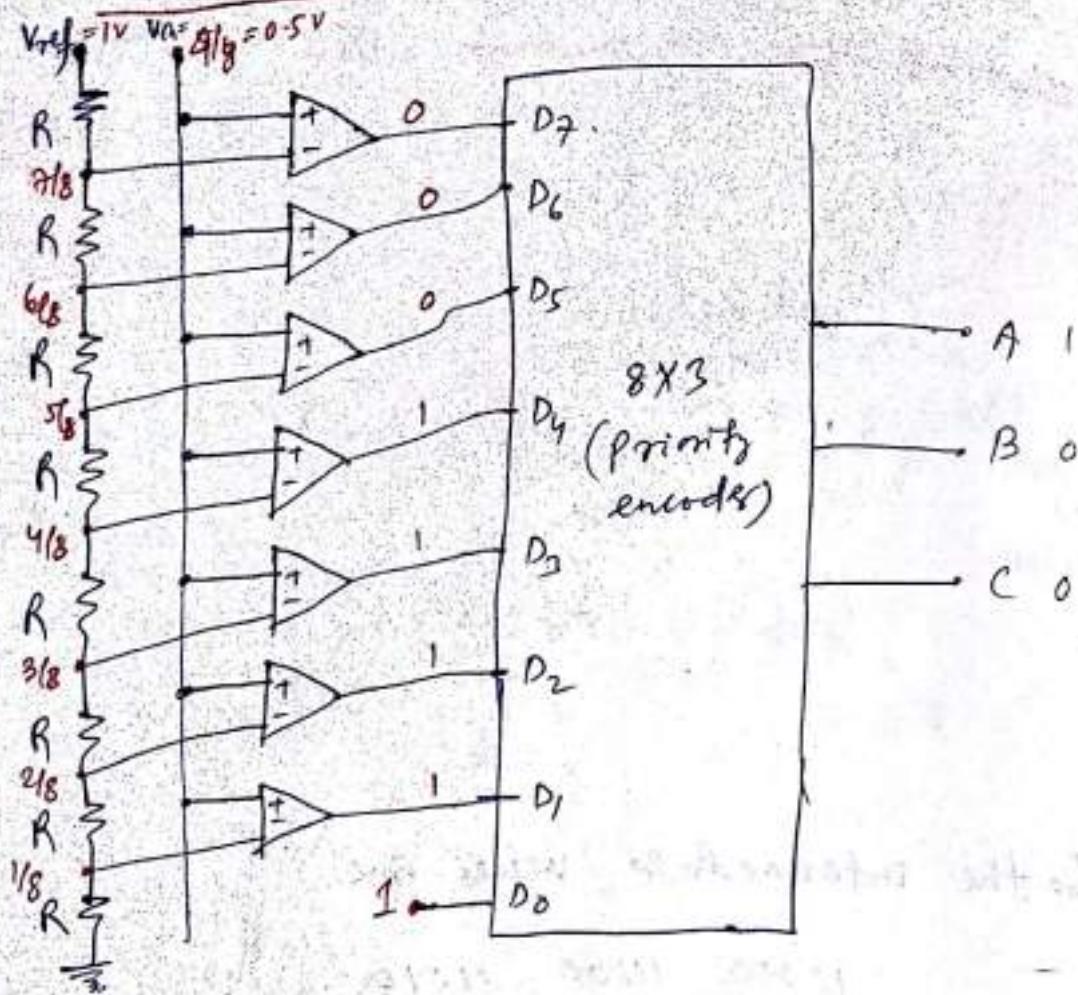
11000, 11100, 11010.

### flash type ADC

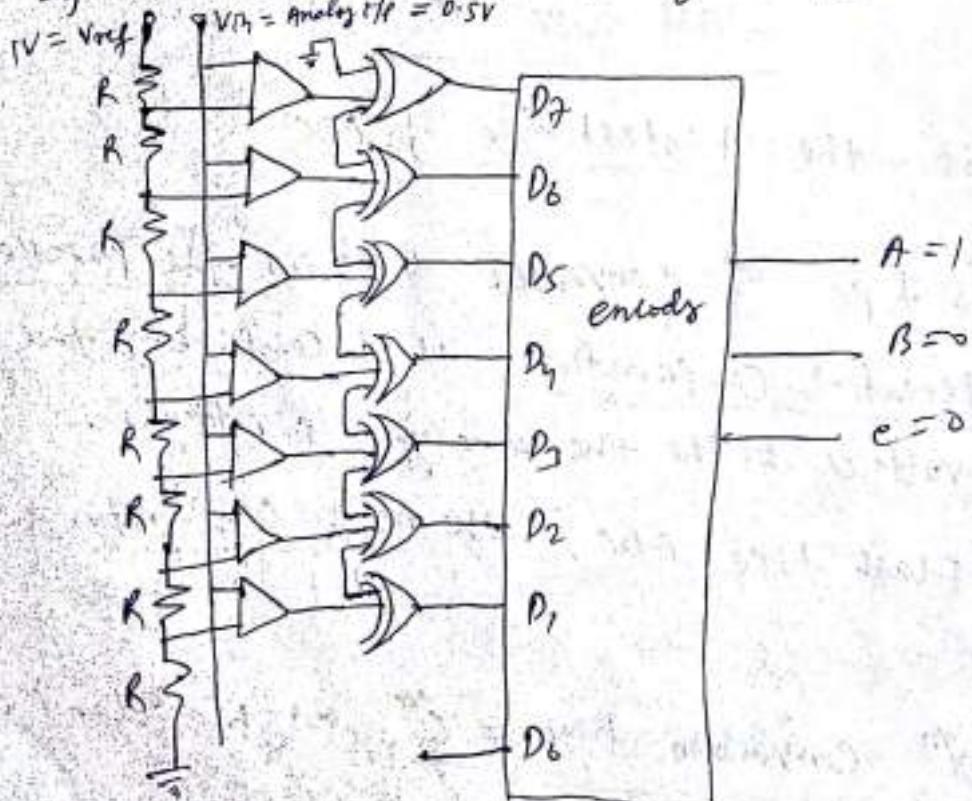
- This is the fastest type of ADC.
- This type of converter utilizes the parallel differential comparators, that compares the ref voltage with the analog input voltages.
- In flash type ADC, the no. of comparators are  $= 2^n - 1$ .

- Max<sup>n</sup> conversion time = 0 as we "don't use clk here."

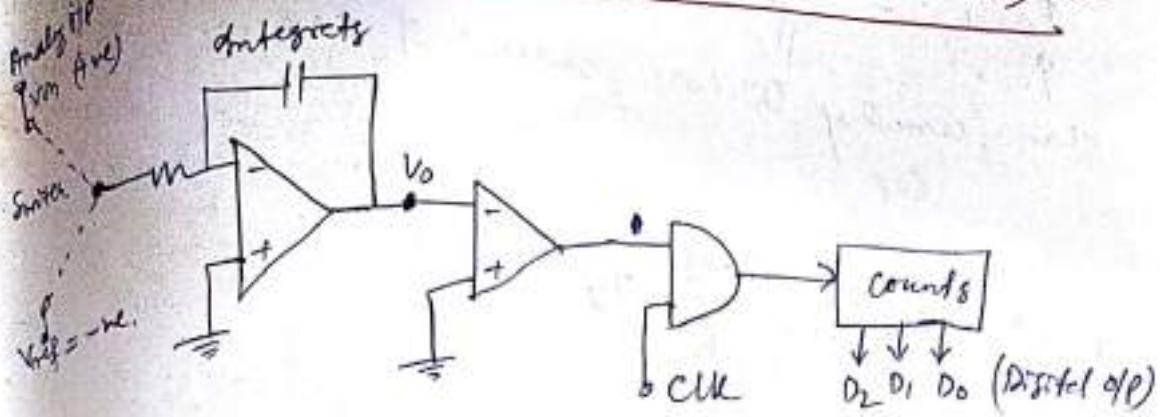
### 3-bit flash type ADC



If we want without parity encoder then see below



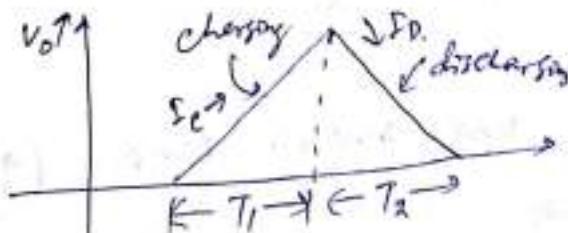
## Integrator type ADC ( Dual slope ADC ) :-



- $V_m$  &  $V_{ref}$  polarities should be opposite.
- when  $V_o > 0 \rightarrow$  Counter Counts, when  $V_o < 0 \Rightarrow$  Counter stops.
- Switch is connected to ' $V_m$ ' for fixed time  $T_1$ , then it goes to ' $V_{ref}$ ' & Counter starts.

→ Initially the counter is clear.

- else:-
- Time constant for integrator is very large so the capacitor is charging linearly, when connected to  $V_m$ .
  - Cap. ~~get~~ discharges, when connected to  $-V_{ref}$  for time  $T_2$



$$T_1 = n_1 T, \quad T_2 = n_2 T$$

$\therefore T = \text{clk time period}.$   
 $n_1, n_2 \rightarrow \text{no. of clk pulses required for charging & discharging of cap.}$

→ Whatever the value of IIP, the value of  $T_1$  &  $T_2$  are fixed.

$$I_C T_1 = I_D T_2$$

↑                   ↑  
charging current of   discharging current of cap.  
Cap.

$$\Rightarrow \frac{V_m}{R} T_1 = \frac{V_{ref}}{R} T_2$$

$$\Rightarrow V_m T_1 = V_{ref} T_2$$

$$\begin{aligned}\Rightarrow V_m &= V_{ref} \left( \frac{T_2}{T_1} \right) \\ &= V_{ref} \left( \frac{n_2 T}{n_1 T} \right) \\ &= V_{ref} \left( \frac{n_2}{n_1} \right).\end{aligned}$$

const.      const.

$$\Rightarrow V_m = k n_2$$

$$\Rightarrow \boxed{V_m \propto n_2}$$

$$\rightarrow \text{Max. Conversion time} = (g^n - 1) T$$

Adv

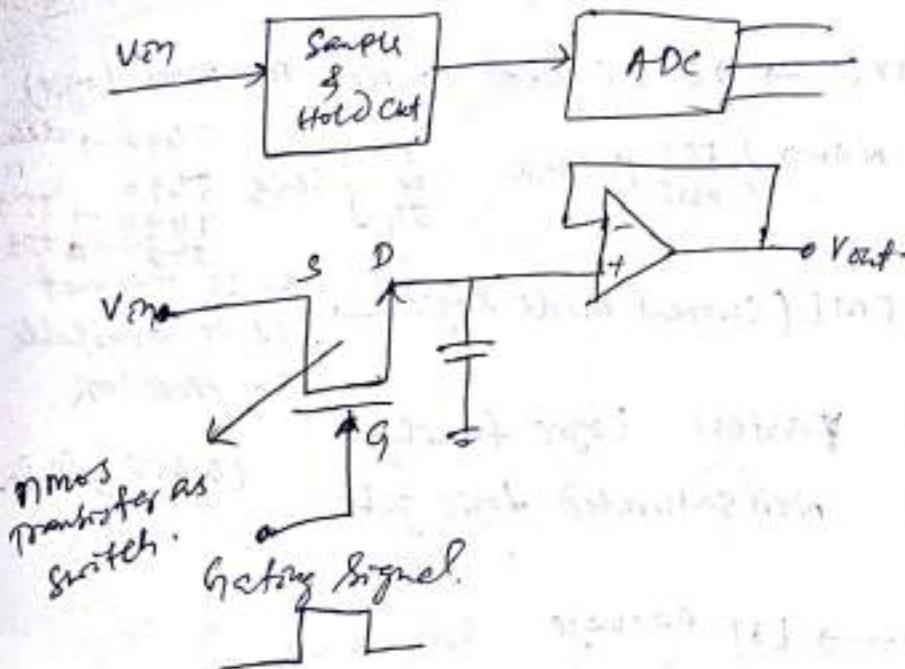
- It is very accurate & used in digital voltmeters
- The integrator at the i/p eliminates the power supply noise, called as "soft hum"

Disadv

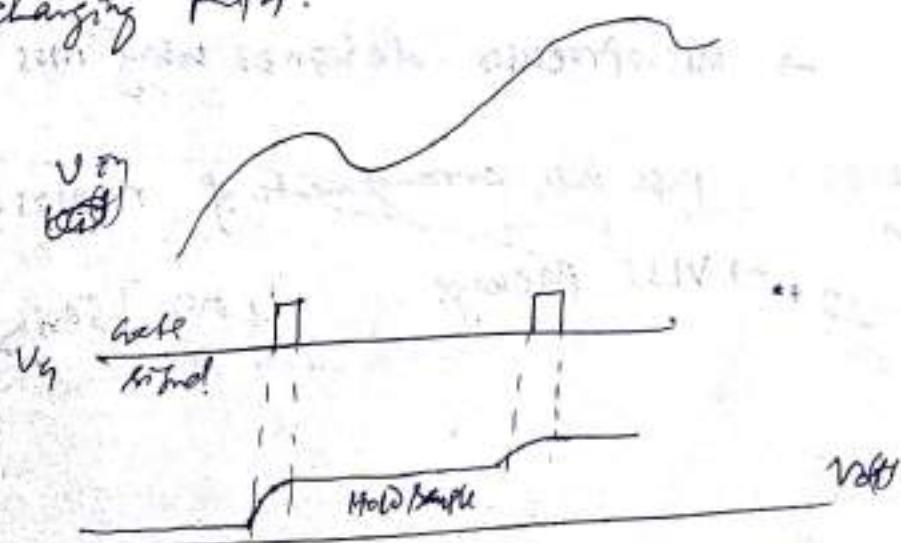
It is very slow in operation.

## Sample & Hold Circ.

When a sinusoidal signal is i/p to ADC to convert a digital signal then we use the Sample & Hold Circ.



When switch is closed the capacity is charged & after some time if switch is open the capacitor hold this sample value b'cse no discharging path.



logic family

- $SSI \rightarrow$  Small Scale Integration.
  - $MSI \rightarrow$  Middle Scale Integration.
  - $L\&I \rightarrow$  Large Scale Integration.
  - $VLSI \rightarrow$  Very Large Scale Integration
  - $ULSI \rightarrow$  Ultra Large Scale Integration
  - $GSI \rightarrow$  Giant Scale Integration.

1. RTL → DTL → DCTL (Diode Coupled Transfer Logic)
  2. TTL → NAND } SSI package      74 } series      7490 → decade  
TTL → NOR } SSI package      74 } series      5490 → "      7490 → TS41  
TTL → AND } SSI package      54 } series      5490 → +5V  
TTL → OR } SSI package      54 } series      5490 → +5V
  3. ECL or CML (Current Mode logic) → In IC format  
it is available in NOR/OR.  
→ Fastest Logic family      10,000 ] series  
→ Non saturated logic gate.
  4. MOS → LSI Package.
  5. CMOS : Push-Pull arrangements of n-MOS & p-MOS.

VLSI → VLSI package. [4000] series

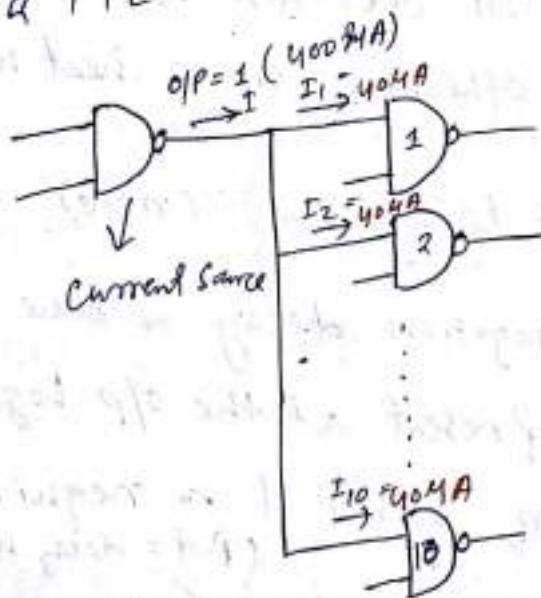
## Important characteristics of logic gates

### 1. Fan-out:

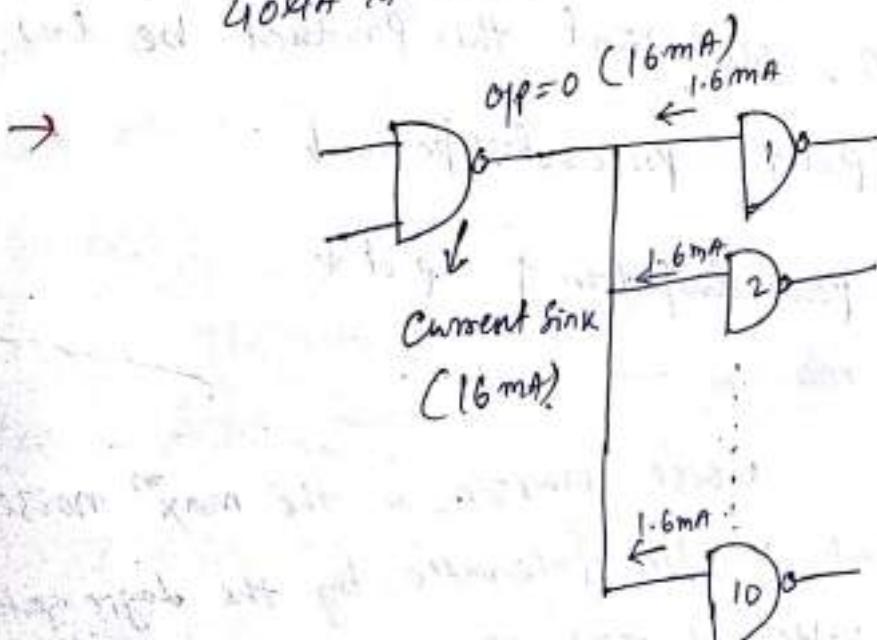
the no. of ops. that can be connected to the op of a logic gate is called as fan-out, without affecting its operat.

Ex consider a TTL NAND.

cond  
, 5°C to 70°C  
, -55°C to 125°C



fan-out = 10.  
40mA is sufficient to drive 10 gate.



Fan-out = 10.

## 2. Fan-in :

The max<sup>m</sup> no. of inputs that a logic gate can have is called as fan-in.

## 3. Power dissipation (mw) :

For well operation less power dissipation is required, otherwise more heat will be generated.

## 4. Propagation delay (P.D.) : (nsec)

Propagation delay is due to the stay capacitors present at the output logic gates. For well operation, less P.D. is required. (P.D = delay time =  $t_d$ ) ( $0 \rightarrow 50\%$ )

## 5. Speed power product (S.P.M)

It is the product of P.D & power dissipation. We want this product be low.

P.D  $\uparrow$  Power dissipation  $\downarrow$

Power dissipation  $\uparrow$  P.D  $\downarrow$

## 6. Noise margin : —

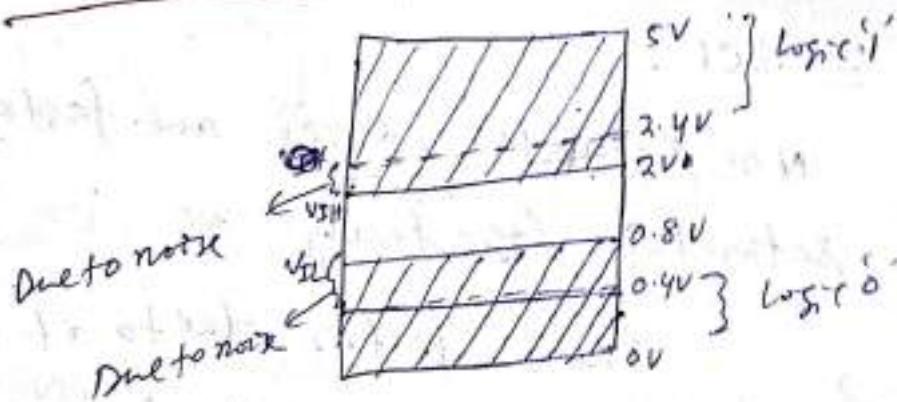
Noise margin is the max<sup>m</sup> noise voltage that can be tolerated by the logic gates.

OPP voltage of TTL NAND

$V_{OL}$  = max<sup>m</sup> opp voltage in the logic Low level.

$V_{OH}$  = minimum opp voltage in the logic High level.

### O/P voltage of TTL NAND



$$\text{High state noise margin} = V_{OH} - V_{IH}$$

$$\text{Low state noise margin} = V_{IL} - V_{OL}$$

- The ckt. can tolerate any noise signal i.e less than or equal to  $V_{OH} - V_{IH}$  &  $V_{IL} - V_{OL}$ .

### Breadth of logic family:

The no. of different logic fam<sup>y</sup>s available for a logic gate is called as Breadth of logic fam<sup>y</sup>

Ex ECL  $\rightarrow$  NOR  $\vee$  OR  $\vee$  Breadth of logic fam<sup>y</sup> = 2.

### Saturated logic:

A form of logic gate in which one opp state is the saturation voltage level of the transistor.

Ex RTL, DTL, TTL.

High  
↑ (required large time  
to go from Low to High)

## 9. Non saturated logic

A form of logic with transistor operating outside the saturation region is called as non-saturated logic.

Ex ECL.

→ Non saturated logics, are faster than saturated logic family.

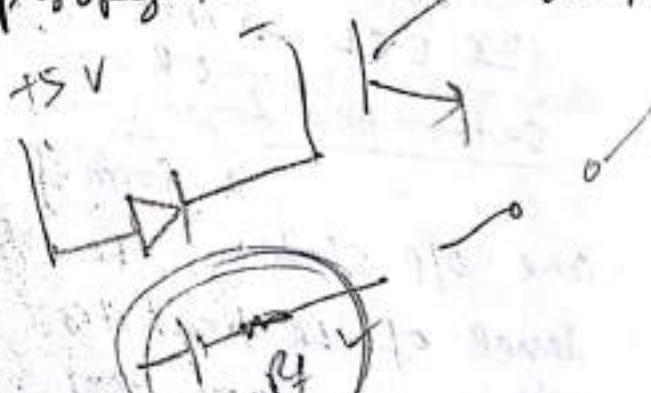
→ Ecl is faster due to it is from active to cut off directly.

1. ECL  $\Rightarrow$  CML  $\Rightarrow$  fastest logic gate family  
 $\Rightarrow$  consume more power..

2. CMOS  $\Rightarrow$  slowest logic gate family.  
 $\Rightarrow$  consume less power  
 $\Rightarrow$  highest fanout.

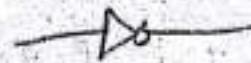
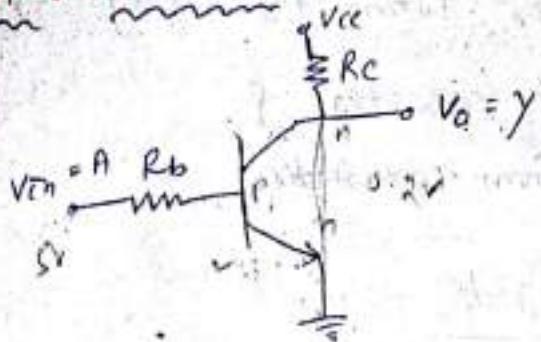
Figure of merit

$$F.O.M = \text{propagation delay} \times \text{power dissipated}$$



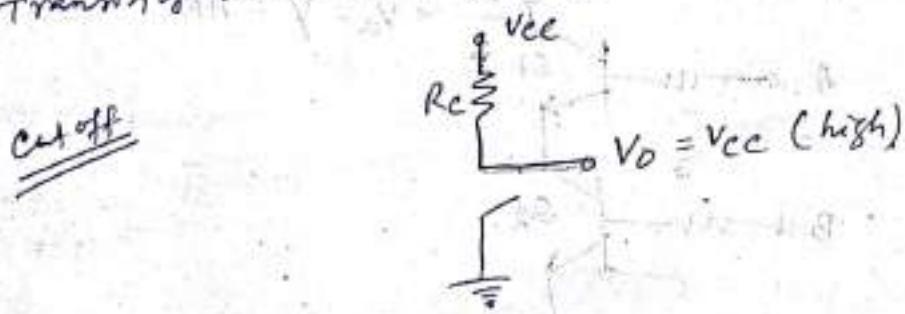
## Transistor as switch.

### Transistor Inverter:

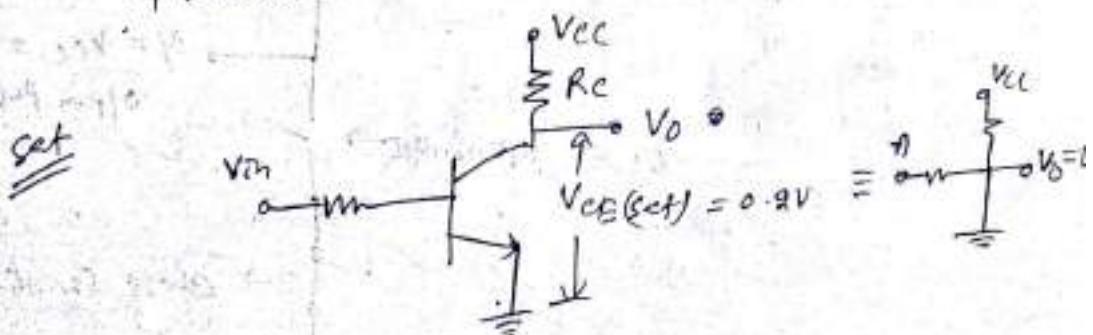


Cond<sup>n</sup> for sat.  $|I_B| > \left| \frac{I_C}{B} \right|$

→ When  $V_{in} = 0$  (Low),  
transistor is in cut off region. (as  $\beta_E = R \cdot B, \beta_C = F \cdot B$ )



→ When  $V_{in} = 5V$  (High),  
transistor is in sat. (as  $\beta_E = F \cdot B, \beta_C = F \cdot B$ )



<u><math>V_{in}(A)</math></u>	<u><math>V_o</math></u>
0	1
1	0

⇒ inverter

→ only CE config used as inverter not CC, CB.  
... look where this inverter.

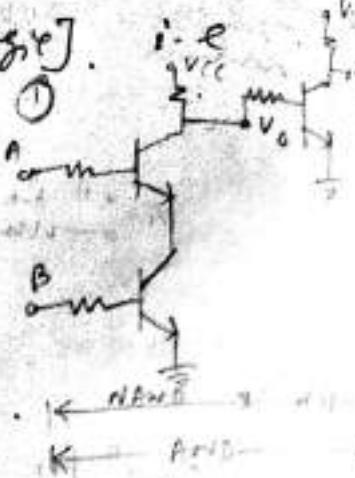
### R-T-L

R-T-L [Resistor Transistor logic].

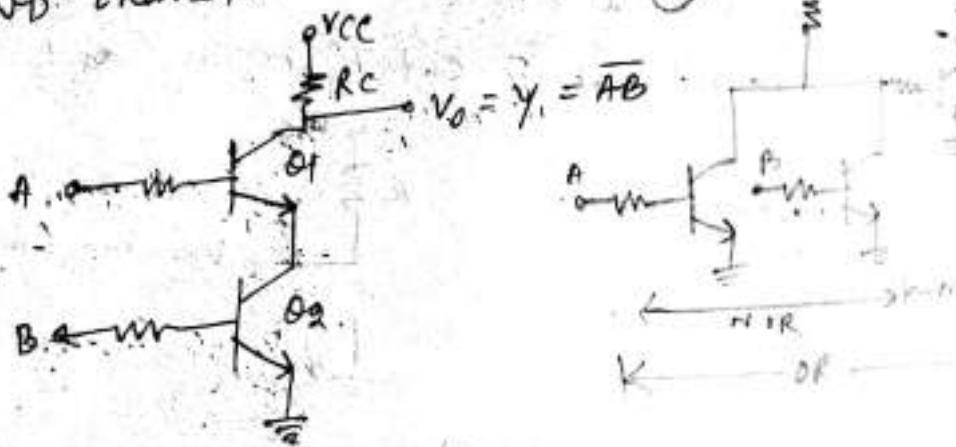
g/p is given to resistor. f  
o/p is taken from transistor.

2 input NAND gate

$$y = \overline{AB} \quad A \text{---} D \text{---} B \quad y = \overline{AB}$$



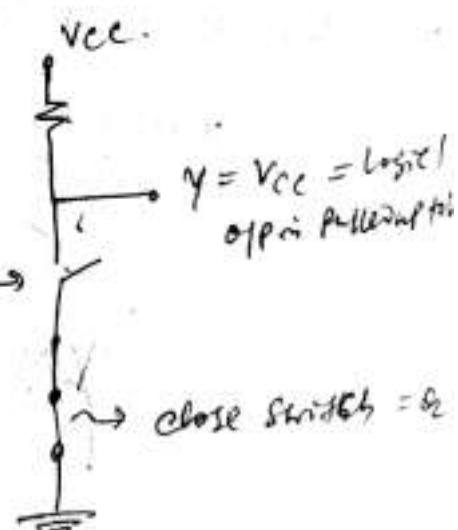
~~W.W.M~~ (AND indicates series switch).



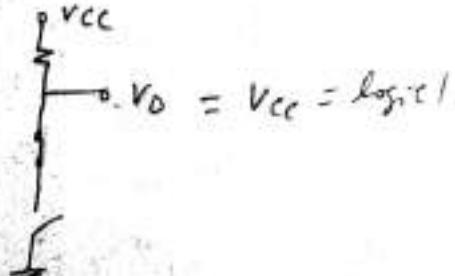
Explaining

when  $A=0, B=1$ .

$\alpha_1 = \text{open switch}$



when  $A=1, B=0$ .



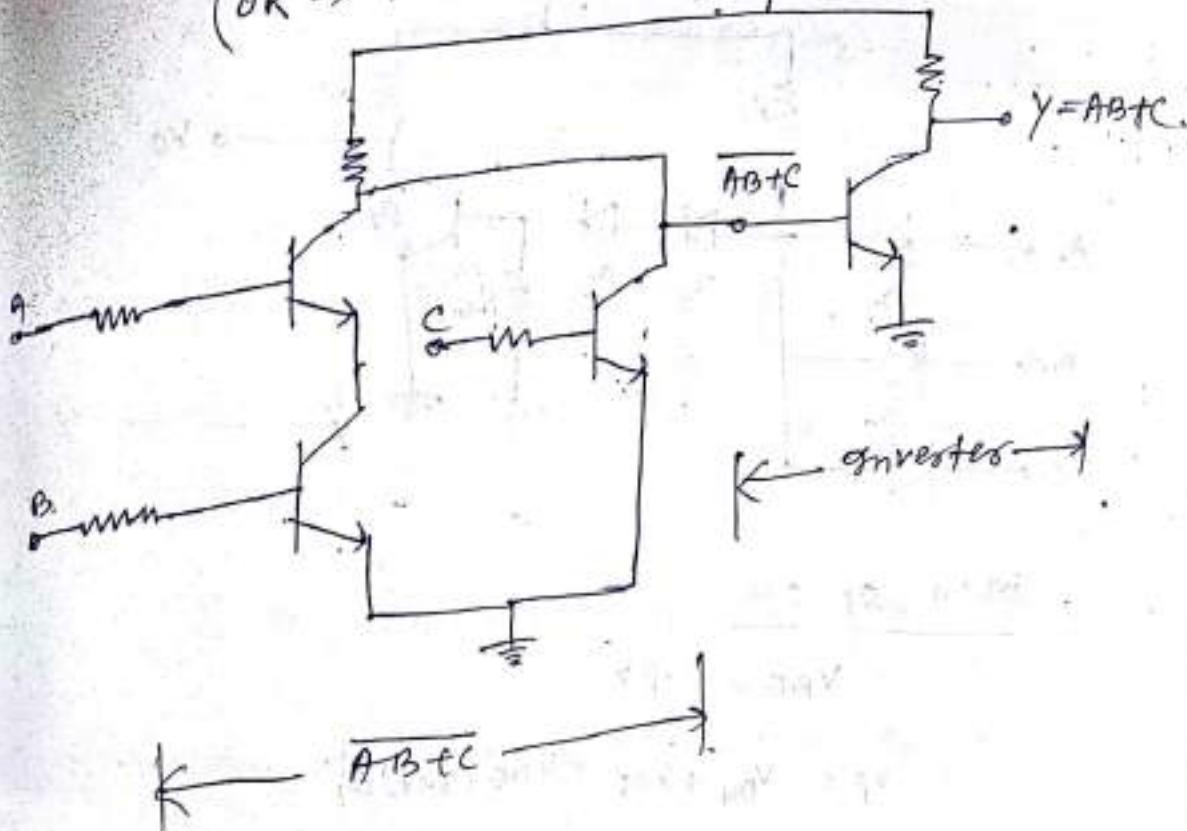
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

→ (NAND)

Three IP Gates

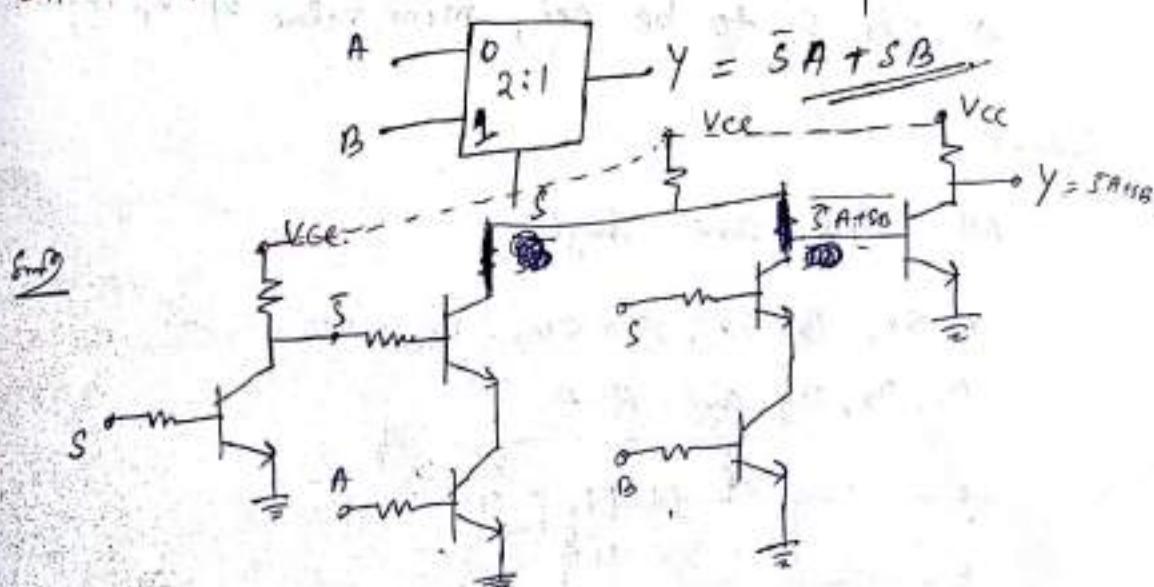
Off Design  $y = AB + C$ .

(OR  $\Rightarrow$  parallel switch)



Off Design  $\Rightarrow$  2:1 max using transistors.

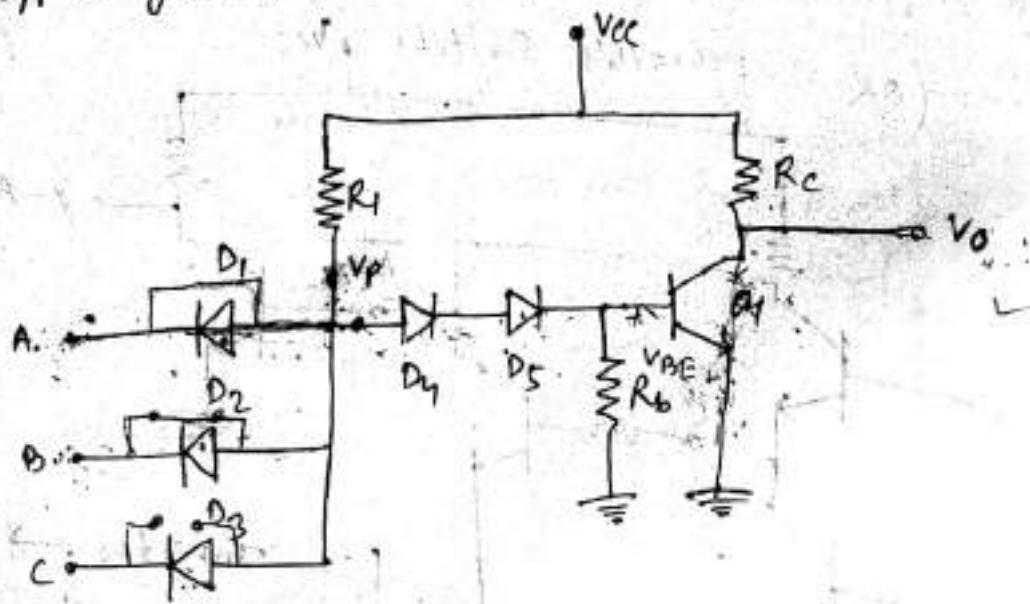
$$y = \overline{S}A + SB$$



Off Design on BX-OR gate  $y = \overline{A}B + A\overline{B}$ .

(Learning of the ~~inverter~~ inverter) = VLSI.

D.T.L (H.T.L) (High Threshold Logic)  
O/P is given to Diode & O/P is taken across tank.



When Q1 ON.

$$V_{BE} = 0.8V$$

$$V_p = V_{D4} + V_{D5} + V_{BE}(\text{sat}) Q_1$$

$$V_p = 0.7 + 0.7 + 0.8 = \underline{2.2V}$$

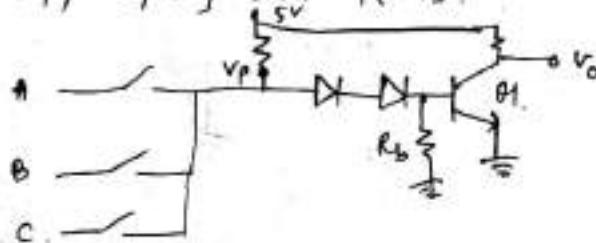
$\Rightarrow$  For 'Q1' to be ON, min. value of  $V_p$  required:

Case-1

All inputs are high.

$$A = 5V, B = 5V, C = 5V. V_{CC} = 5V$$

$D_1, D_2, D_3$  are  $R \cdot B$ .



$V_p$  is connected to 5V, so  $Q_1$  goes to sat.

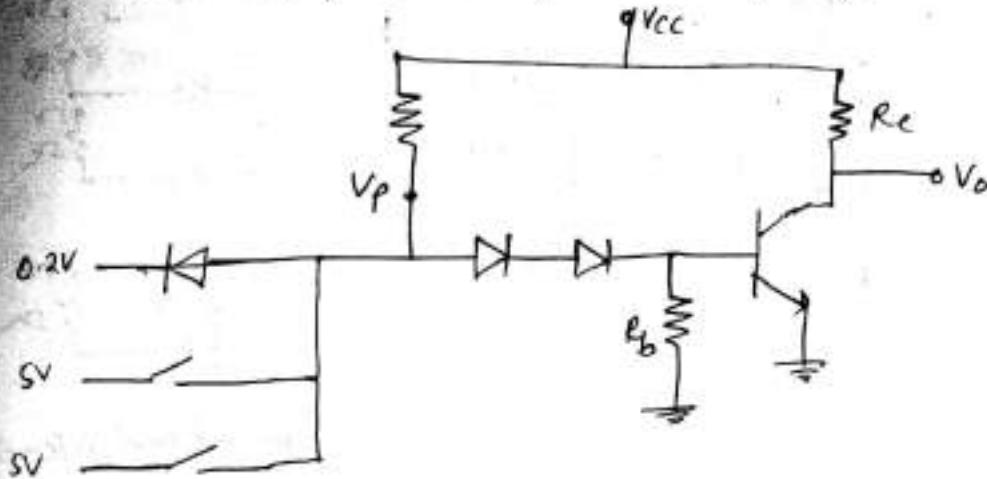
$$V_0 = V_{CE}(\text{sat}) = 0.2V = \text{logic '0'}$$

④  $B$  value is taken such a way that it will go to sat.

$$|I_B| > \left| \frac{I_C}{\beta} \right|$$

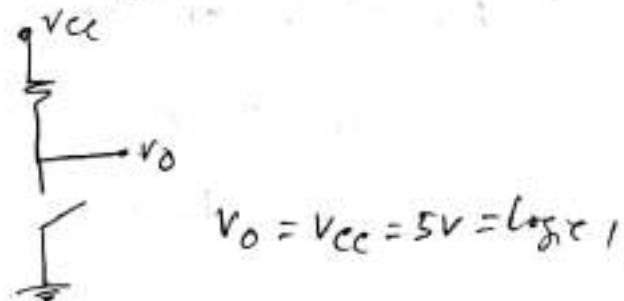
If any one of the i/p is given low voltage.

$$A = 0.2V, B = 5V, C = 5V \text{ (say)}$$



$$V_P = 0.2V + 0.7V \text{ so } Q_1 = \text{off.} \quad (\text{as min of } 2.2V \\ = 0.9V)$$

(necessary for Q1 to be on) Hence Q1 = off.

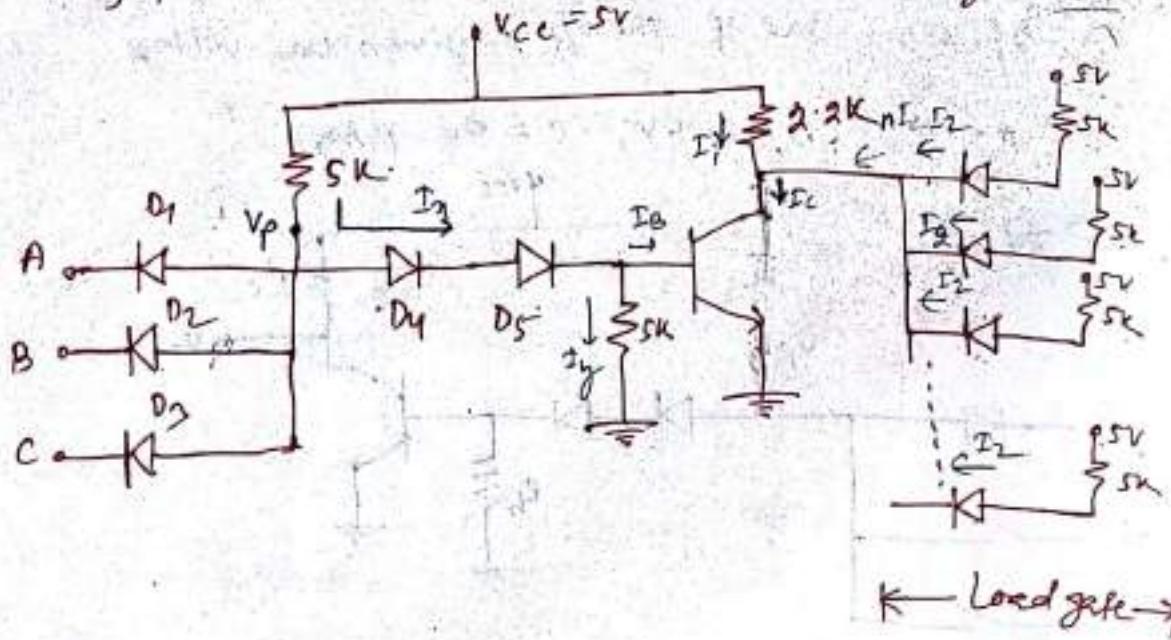


$$V_0 = V_{CC} = 5V = \text{logic 1}$$

A	B	C	V0
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

$\Rightarrow$  NAND Gate.

Q1 Find the no. of (load gates) that can be connected  
 if  $\beta = 30$ . Such that the transistor is working in sat.



Soln Cond<sup>n</sup> for Sat.

$$|I_B| > \left| \frac{I_C}{\beta} \right|$$

→ For all inputs are high for Sat.

$$V_P = 2.2V$$

$$I_{\eta} = I_B + I_Y$$

$$\Rightarrow \frac{V_{CC} - V_P}{5k} = I_B + \frac{0.8}{5k}$$

$$\Rightarrow \frac{5 - 2.2}{5k} - \frac{0.8}{5k} = I_B$$

$$\Rightarrow \frac{2.8}{5k} = I_B \Rightarrow I_B = 0.4mA$$

$$I_C = I_B + N I_D$$

$$|I_B| > \left| \frac{I_C}{\beta} \right|$$

$$\Rightarrow |I_B| \beta > I_C$$

$$\frac{5 - 0.2 - 0.7}{2.2 \text{ k}} + N \cdot \frac{5 - 0.2 - 0.7}{2.2 \text{ k}} \leq 30 \times 0.4 \text{ mA}$$

$$2.18 \text{ mA} + N \cdot 0.86 \text{ mA} \leq 30 \times 0.4 \text{ mA}$$

$$\Rightarrow N \leq 12.$$

choice of 10' is always better design.

$$I_B = 0.4 \text{ mA}$$

$$I_{CQ} = \beta I_B = 1.2 \text{ mA}$$

$$\begin{array}{c} 5 - 0.2 \\ \diagdown 2.2 \\ 5 - 0.7 - 0.2 \\ \hline 5 \end{array} \text{ mA} = 2.18 \text{ mA}$$

$$\begin{array}{c} 5 - 0.2 \\ \diagdown 2.2 \\ 5 - 0.7 - 0.2 \\ \hline 5 \end{array} \text{ mA} = 0.86 \text{ mA}$$

$$I_{CQ} = n I_2 + I_1$$

$$1.2 \text{ mA} = n \left( \frac{5 - 0.2 - 0.7}{2.2 \text{ k}} \right) + \left( \frac{5 - 0.2}{2.2 \text{ k}} \right)$$

$$\cancel{1.2 \text{ mA}} = n \times 0.82 \text{ mA} + 2.18 \text{ mA}$$

$$\underline{n \leq 12}$$

Wired AND

### Different TTL OpPS

i) open collector TTL OpPS

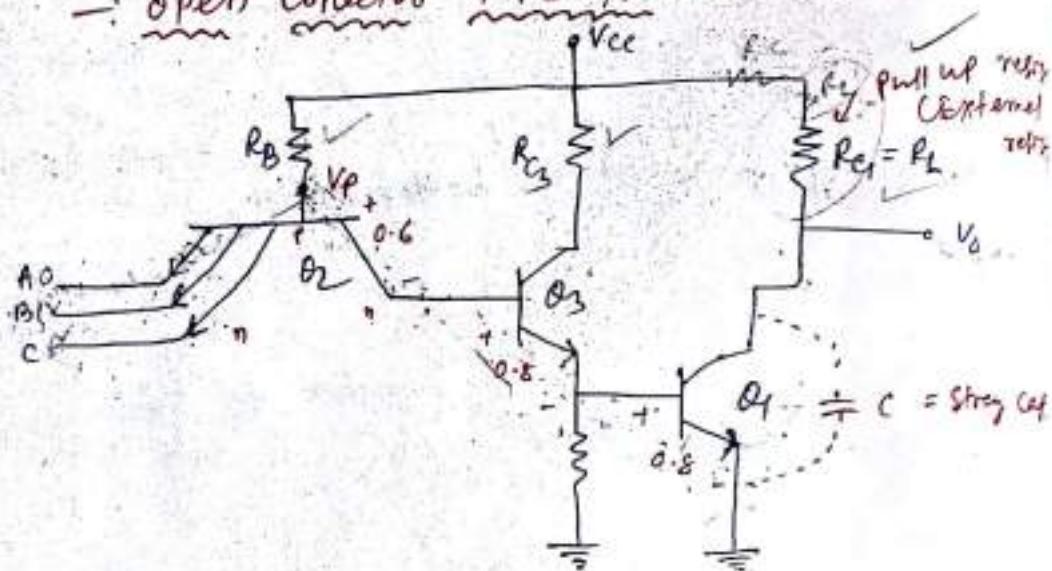
ii) Totem pole TTL OpPS.

iii) Tristate TTL OpPS.

\* Floating i/p's are allowed in TTL family & the floating i/p is always = logic 1.

\* Floating i/p's are not allowed in CMOS.

i) -: open collector TTL OpPS :-



→ For transistor Q1 to be on,

$$\begin{aligned}V_p &= V_{BEQ_1(\text{sat})} + V_{BEQ_3(\text{sat})} + V_{BCQ_2} \\&= 0.8 + 0.8 + 0.6 \\&= 2.2 \text{ V}\end{aligned}$$

Case-1

When all the i/p's are high, i.e.  $A=B=C=5V$ .

$V_p$  directly connected to  $V_{CC}$  so

$$I_{EQ_2} = R \cdot B, \quad I_{CO_2} = f \cdot B.$$

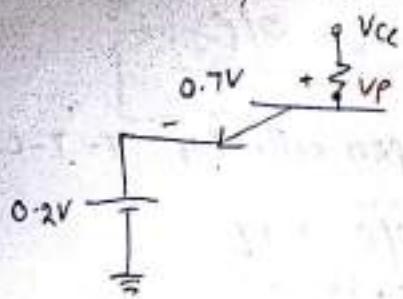
⇒ Transistor Q2 is said to be working in inverse active mode.

$V_{p\min} = 2.2 \text{ V}$  for Q1 to be on, but  $V_p$  is connected to 5V. Hence Q1 is comfortably go to sat<sup>(n)</sup>

$$V_O = V_{CC(\text{sat})} = 0.2 \text{ V} = \text{logic 0.}$$

Case-2 When any one of  $V_P$  is low.

$$A = 0.2V, \quad B = 5V = C.$$

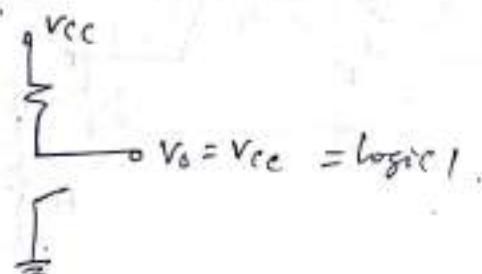


\* floating inputs are allowed & in Schottky  
\* floating inputs are always '0'.

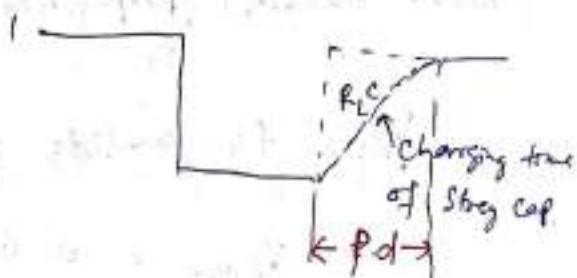
$$V_P = 0.2 + 0.7 = 0.9V.$$

But  $V_{P\min} = 2.2V$ , for  $Q_1$  to be ON. Here  $V_P = 0.9V$

so  $Q_1$  is OFF



$$\begin{array}{cccc} A & B & C & V_o \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array} \quad \text{so } V_o = \overline{ABC} \text{ (NOT AND)}$$



Suppose  $R_L$  is very large

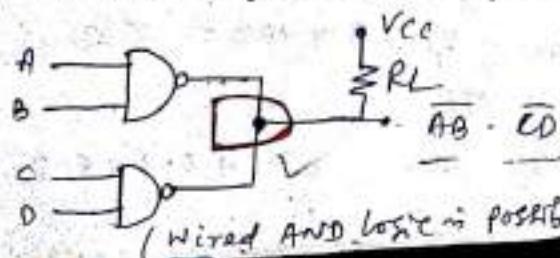
$\rightarrow R_L C$  will be large

$\rightarrow$  propagation delay will be large.

Dissadv  $\rightarrow$  the propagation delay of  $T = T_L + L$  is more.

Applications

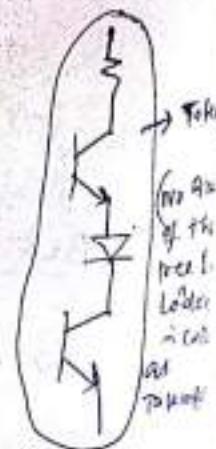
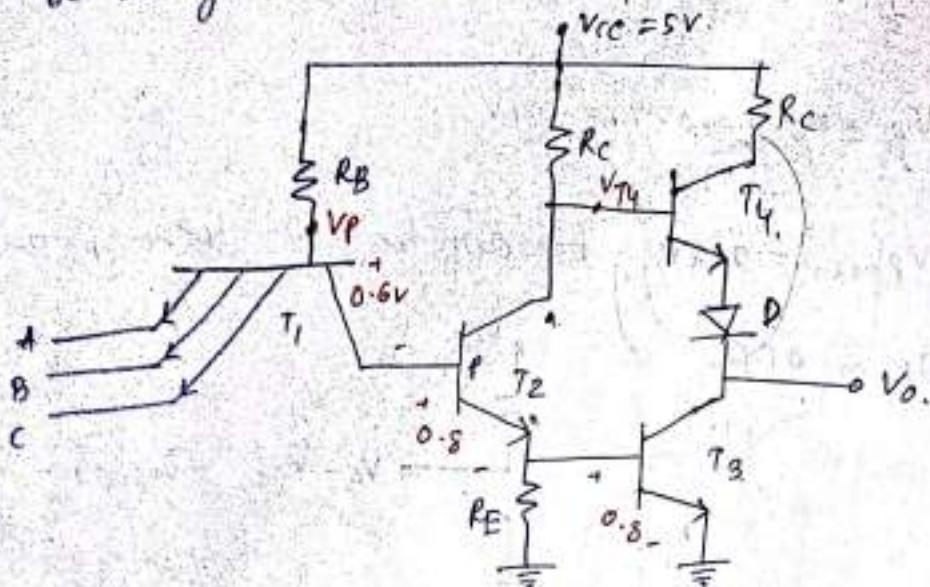
(1)  $\rightarrow$  Wired logic is possible.



(2) Common Bus Lines are design by using open-collector TTL.

### TTL with Totem pole O/P :-

The propagation delay of open collector T-T-L is more so we go to Totem pole off TTL.



The propagation delay of T-T-L (open collector) off gate during turn off time is 35 nsec. but due to active pull (Bursts transits), propagation delay is reduced to 10 nsec.

$V_{p,\min}$  for transistors  $T_3$  ON,

$$V_{p,\min} = 0.6 + 0.8 + 0.8 = 2.2V.$$

Case-1 When all the O/Ps are high.

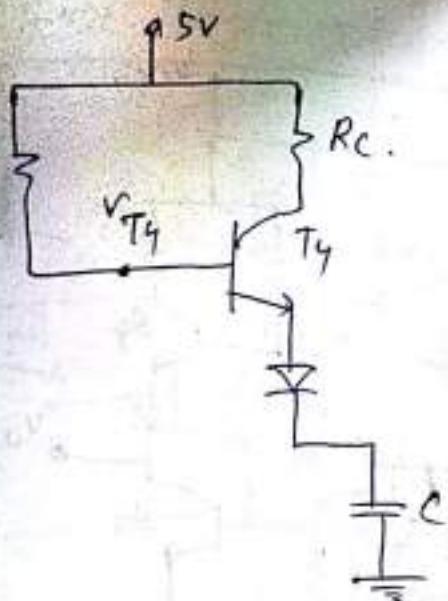
$$V_p \underset{\min}{=} 2.2V, V_{out} = V_o = V_{CE(\text{sat})} = 0.2V = 0$$

Case-2 If any one of the O/P is low:

$$T_1 \approx \text{ON}, V_p = 0.7 + 0.2 = 0.9V.$$

$T_2 \approx \text{off}$

$$V_{T_4} \text{ should be } \min = 0.7 + 0.7 + 0.2 = 1.6V$$



$V_{T4} = 5V \Rightarrow T_4$  is on. & Diode is on.

Hence  $T_4$  & Diode are on & then off <sup>After the</sup> <sub>Completion of  $V_{T4}$</sub> .

→ Cap. charged upto  $3.6V = 5 - 1.4 =$  ~~5.0~~ logic 1

propagation delay  $\tau = RC$

$R$  is less means ( $\tau \downarrow$ )

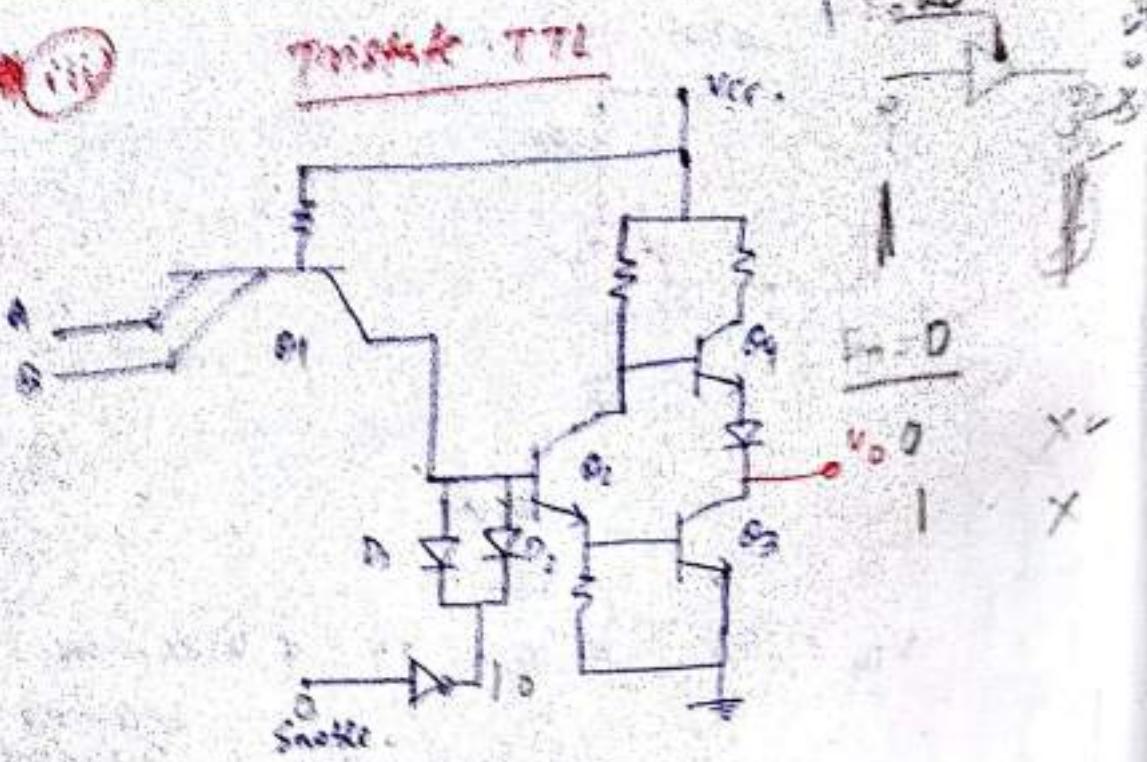
if  $R_C = 0$  then transistors will be burnt.

→  $T_2$  Transistor is used as Phase Splitter.

→ Diode 'D' protects the transistor  $T_3$  from large current.

Adv.

Totem Pole TTL is faster than open collector TTL



→  $\text{Enable} = 0$ .

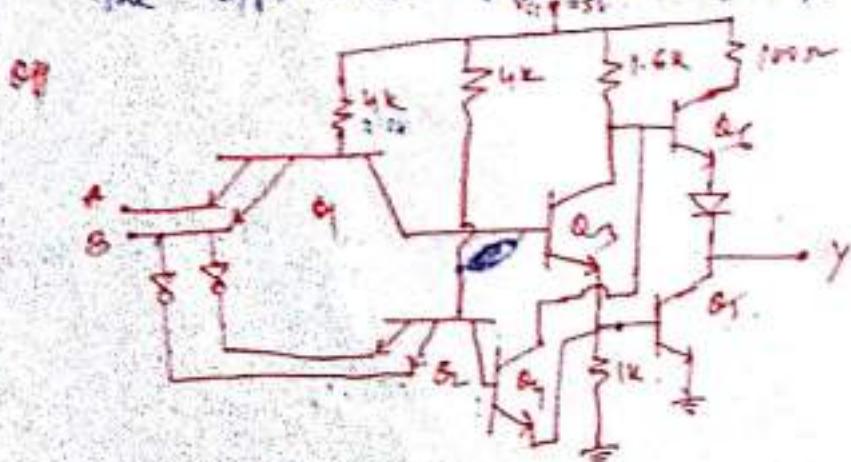
Diodes  $D_1$ ,  $S$ ,  $D_2$  are R.B of off state i.e.

open drain TTL operation.

→  $\text{Enable} = 1$ ,  $D_1$  &  $D_2$  F.B of on state i.e.

shorted &  $Q_2$  base is connected to ground.  
 $Q_2$  off.

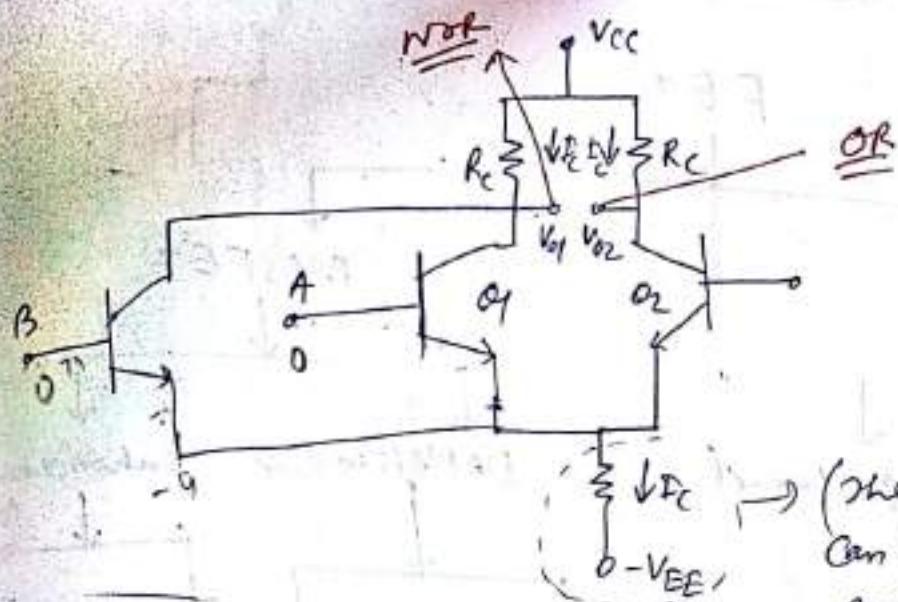
transistor  $Q_3$ ,  $Q_4$  &  $Q_5$  are in off state i.e.  
the ops are open circled & offering high if  $y = 1$ .



the off  $\Rightarrow Y = 1$

- $A\bar{B}$
- $\bar{A}B$
- $A\bar{B} + \bar{A}\bar{B}$  or  $\bar{A}B + A\bar{B}$

- ECL logic gates :-



(The entire arrangement  
can be replaced by  
constant current source)

Assume Logic 0  $\rightarrow -1.6V$ .

(Logic 1  $\rightarrow -0.7V$ )

<u>VA</u>	<u>VB</u>	<u>V<sub>O1</sub></u>	<u>V<sub>O1</sub></u>	<u>V<sub>O2</sub></u>
0	0	-5	1	0
0	1	0		1
1	0	0		1
1	1	0		1

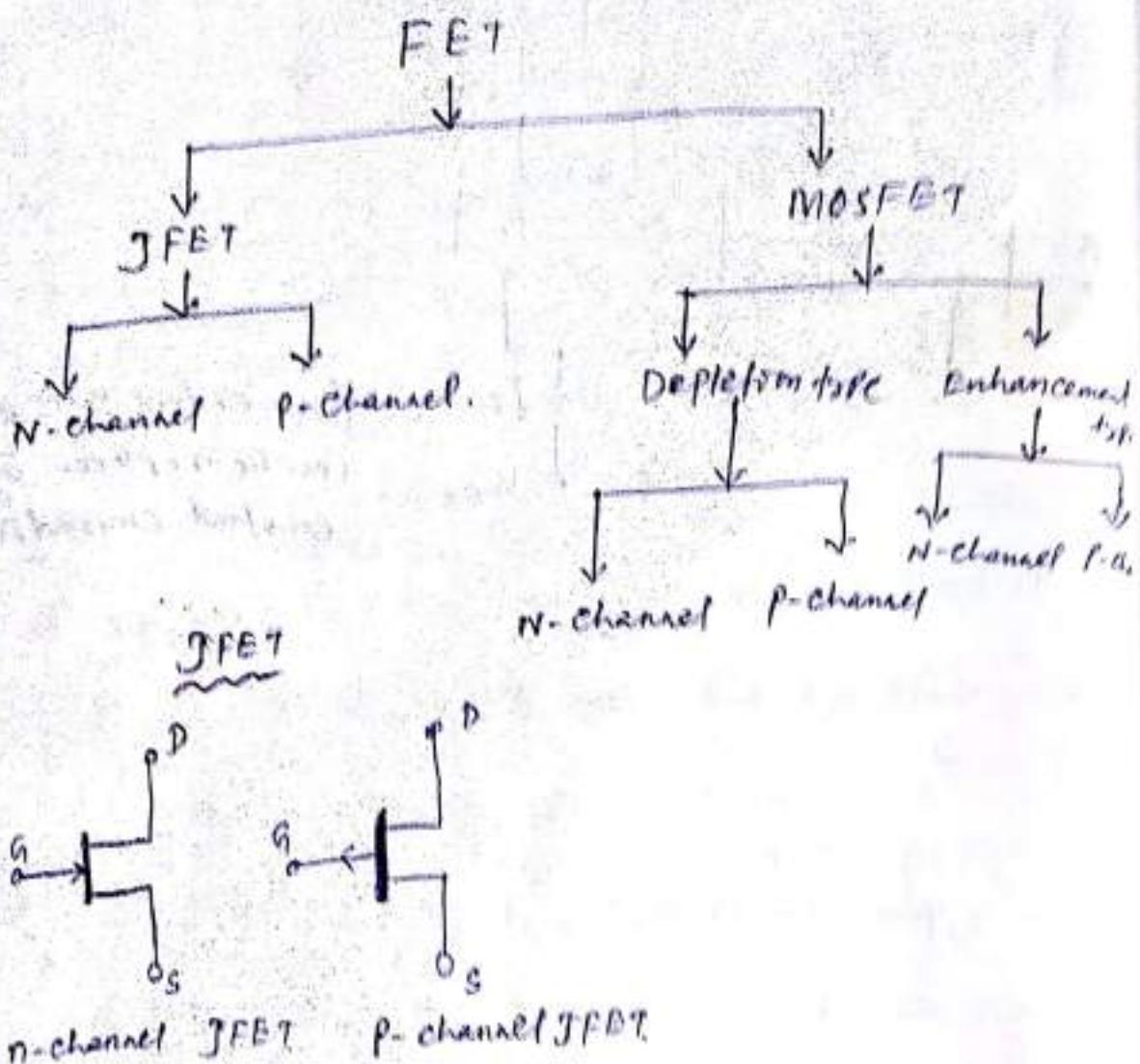
- Fastest
- lowest possible propagation delay
- max<sup>n</sup> freq. of operation.
- Active to cut off (other than BCL all other ckt. operates from cut-off to set. & set. to cut-off)

→ Applicability

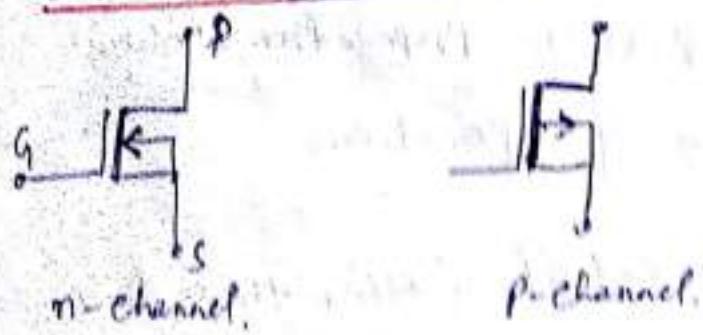
Another logic is possible for ECL i.e. wired-OR logic

NOR OR

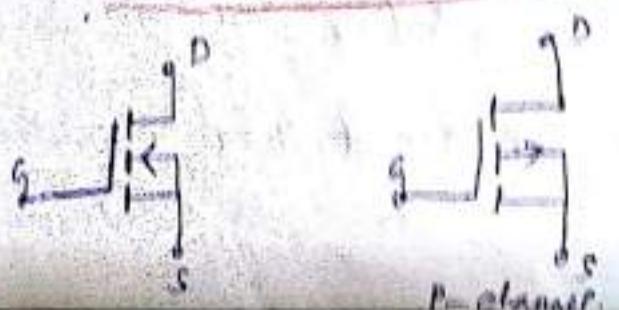
## MOS Logic family

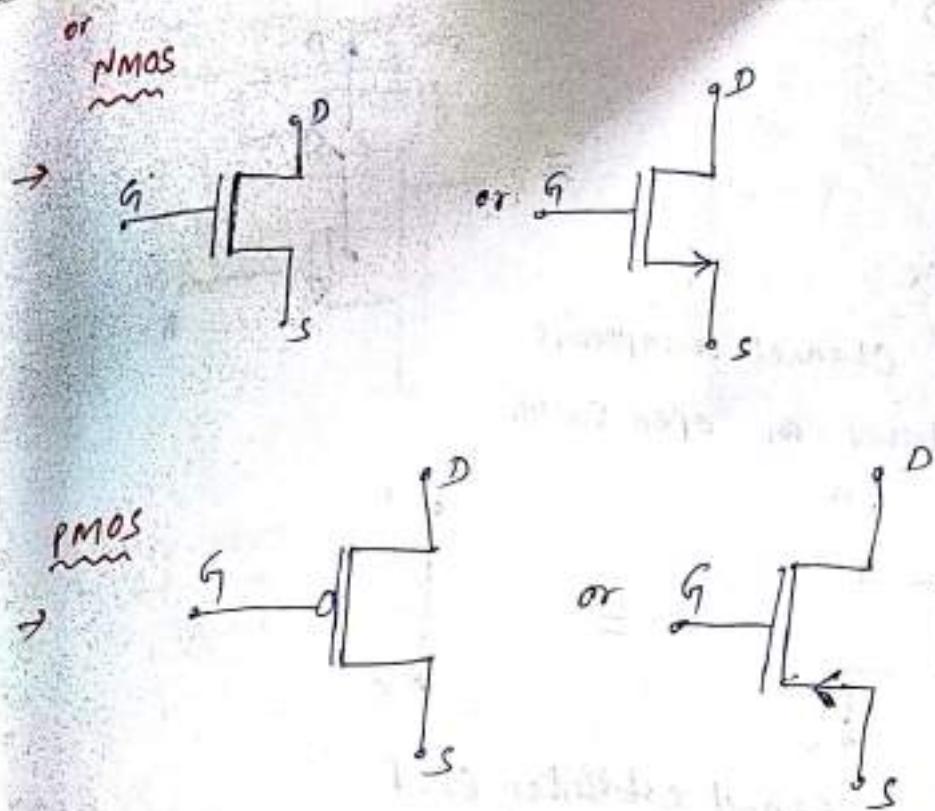


### Depletion type MOSFET



### Enhancement type MOSFET





→ Threshold voltage ( $V_t$ ):

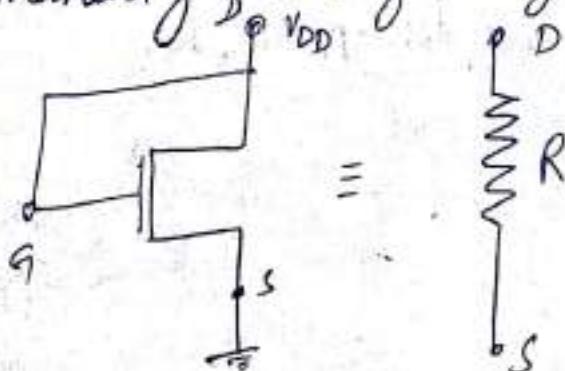
the voltage at which the MOS transistor conducts is called as  $V_t$ .

→ For pmos,  $V_t < -2V$ .

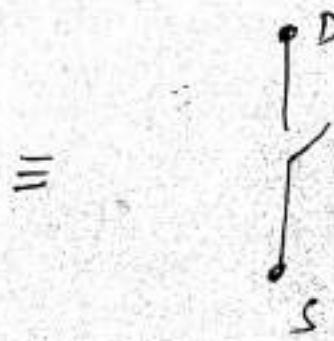
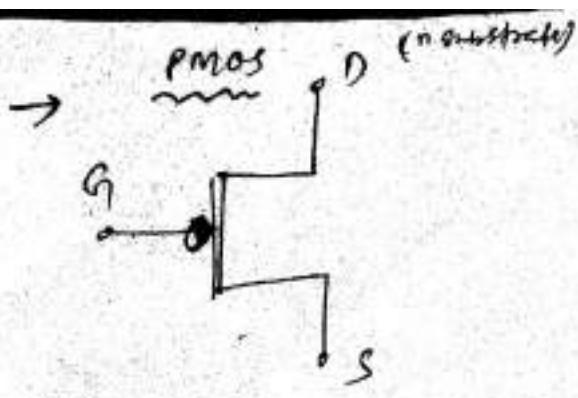
for nmos,  $V_t > 2V$ .

→ If  $V_{GS} = 0$  then both types of MOSFET are turned off.

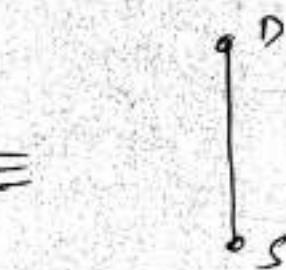
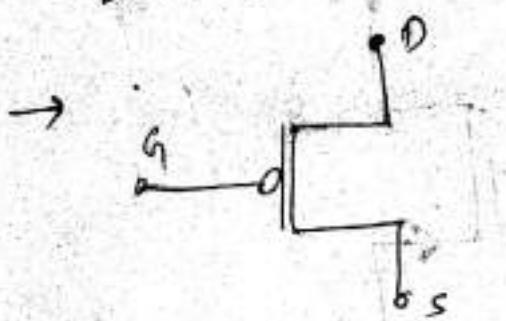
→ MOS Devices can be used as resistor by permanently biasing the gate terminal.



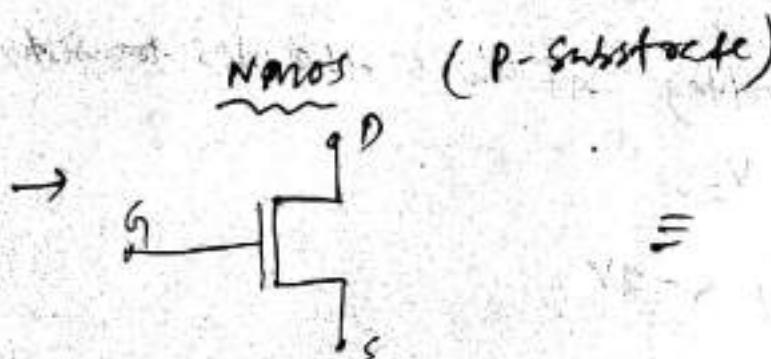
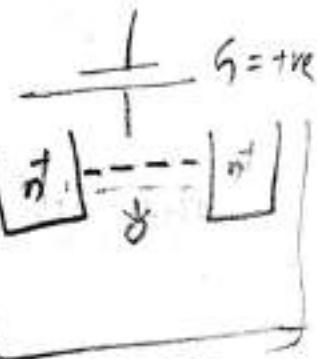
$$R = \frac{V_{DS}}{\text{channel current}}$$



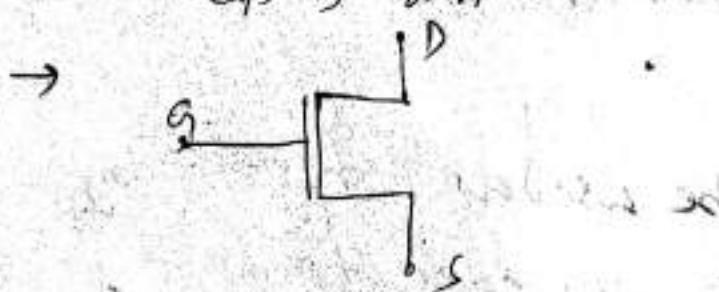
$G = 1 \Rightarrow$  channel disappears.  
so it behaves as open switch.



$G = 0$ , channel establishes. so if acts as short circuit.

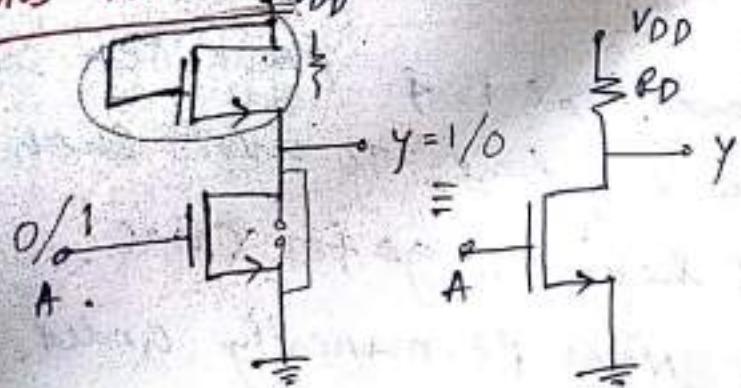


$G = 1$ , channel establishes. so if acts as short circuit.



If  $G = 0$ , channel disappears. so if acts as open circuit.

### N-MOS NOT Gate

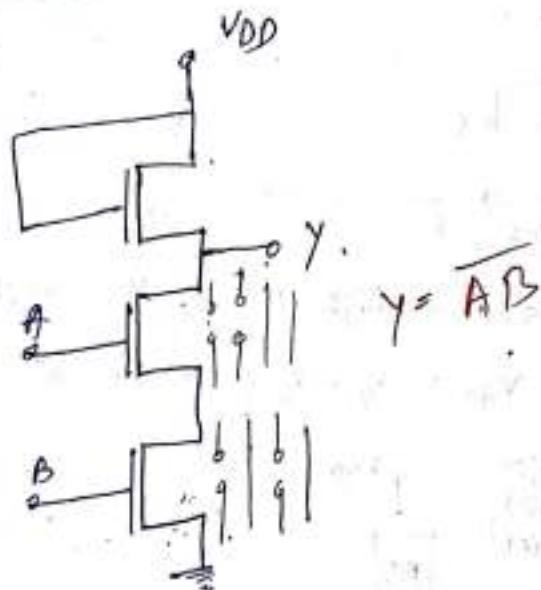


$$\begin{array}{c} A \\ \hline 0 & 1 \end{array} \quad \begin{array}{c} Y \\ \hline 1 & 0 \end{array} \Rightarrow Y = \bar{A}$$

### N-MOS NAND Gate

(AND = series)

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

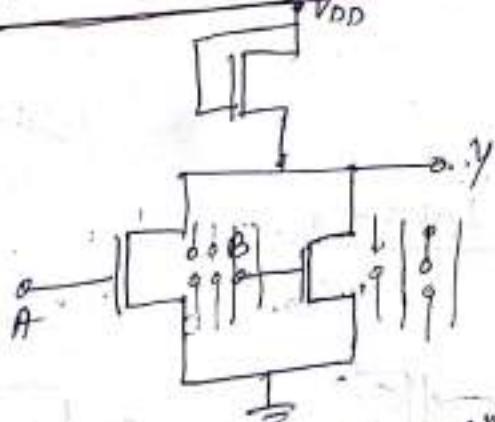


$$Y = \overline{AB}$$

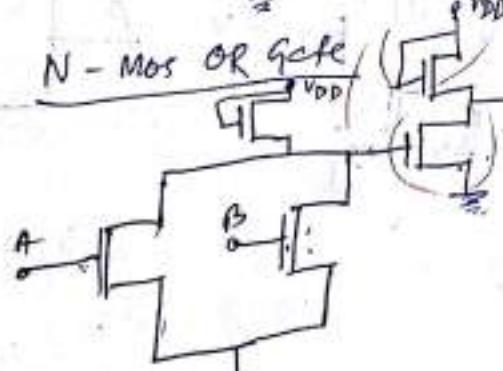
### N-MOS NOR Gate

$$\overline{A+B}$$

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

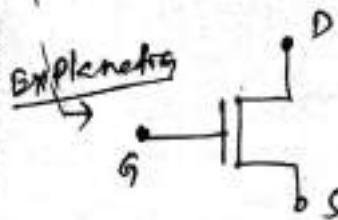


$$N - Mos OR Gate \quad \begin{array}{c} V_{DD} \\ \parallel \\ \text{Transistor} \end{array} \quad Y = A + B$$



### C-MOS

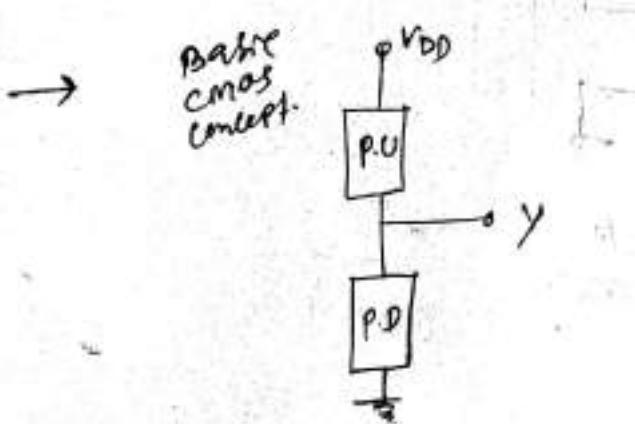
- NMOS can't send logic 1 & PMOS can't send logic 0 for a long time. To avoid such instability we have to go for CMOS.
- In CMOS, NMOS permanently connected to ground & PMOS to V<sub>DD</sub>.



$$g_f = g_d = 1 \Rightarrow V_g = \text{High} \Rightarrow S \xrightarrow{\quad} D$$

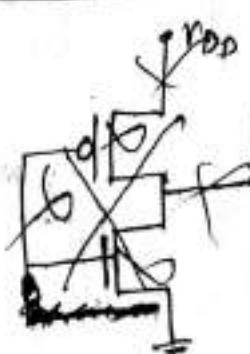
If Data is high '1'  $\Rightarrow V_g = \text{High}$ .

Now  $V_{GS} = V_g - V_S = \text{High} - \text{High} = \text{Low}$ , so MOSFET will OFF.



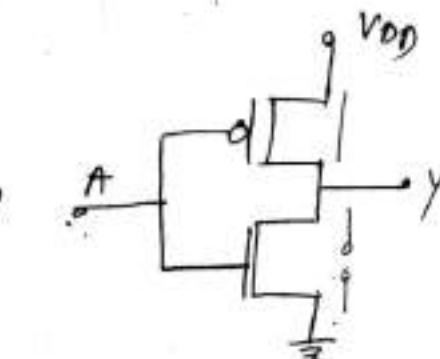
### CMOS Inverter

- A = 0  
PMOS ON  $\Rightarrow Y = 1$   
NMOS OFF



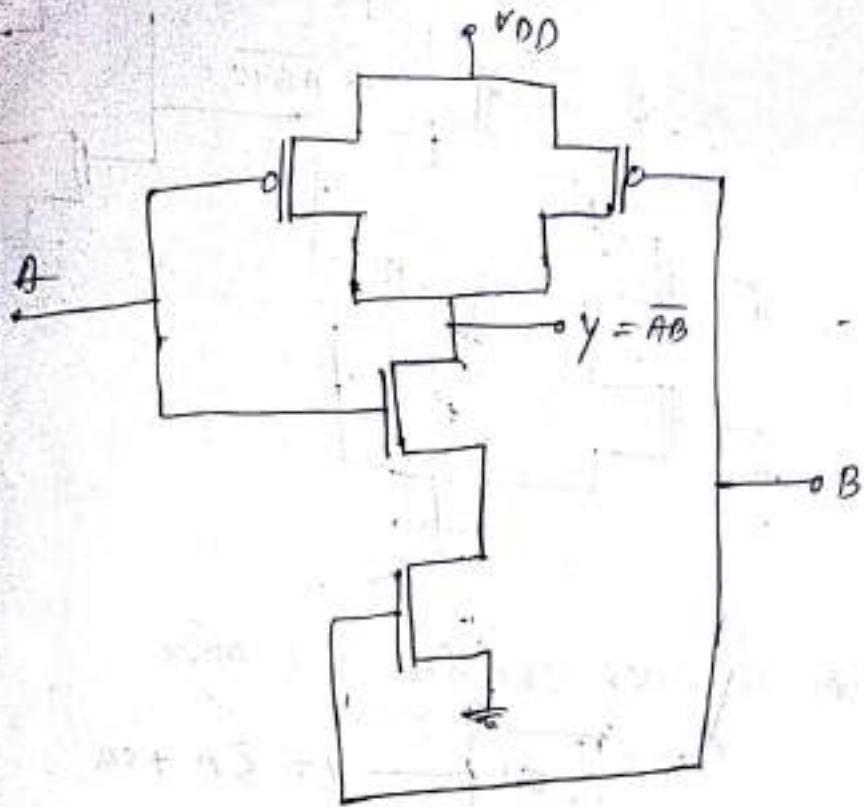
- A = 1

- PMOS OFF  $\Rightarrow Y = 0$   
NMOS ON  $\Rightarrow Y = \bar{A}$ .



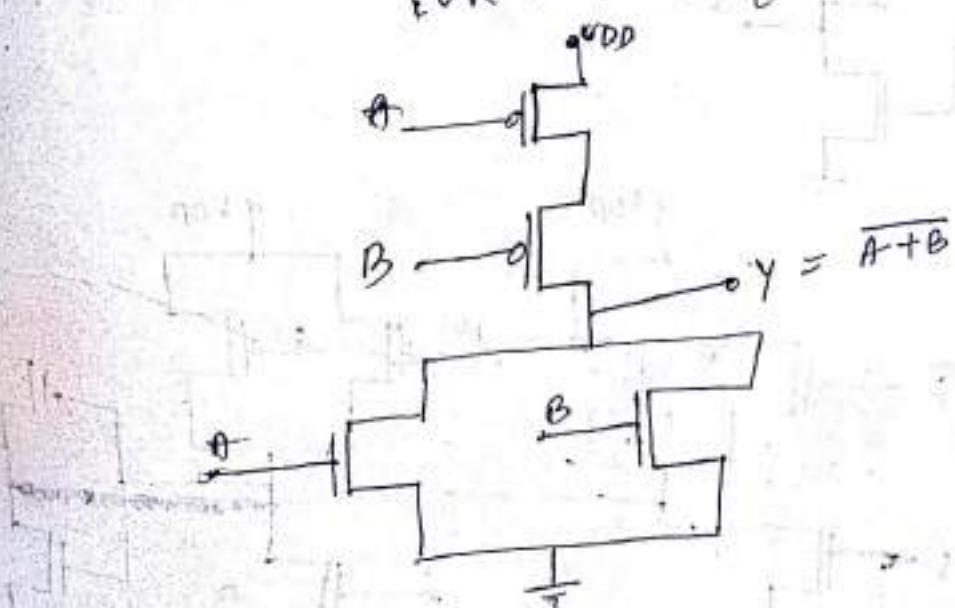
## CMOS TWO i/p NAND Gate

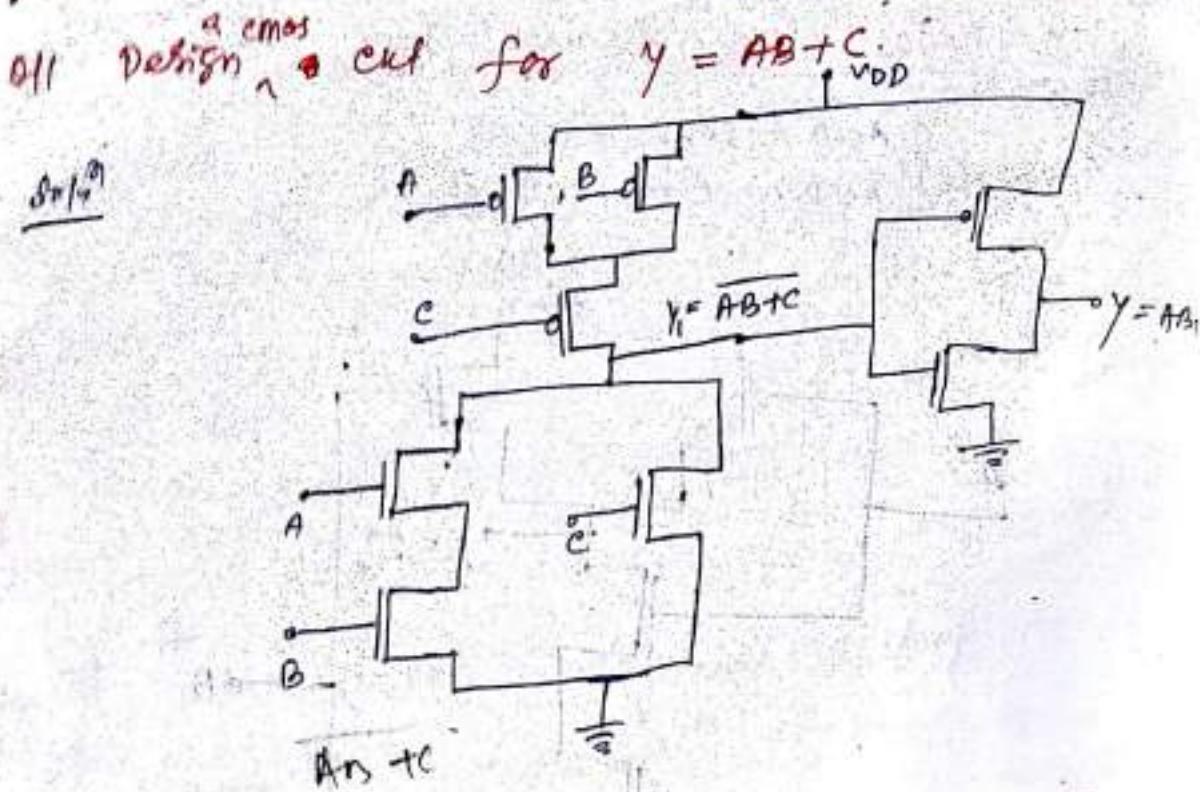
{ AND = Series [Nmos]  
AND = Parallel [PMos]



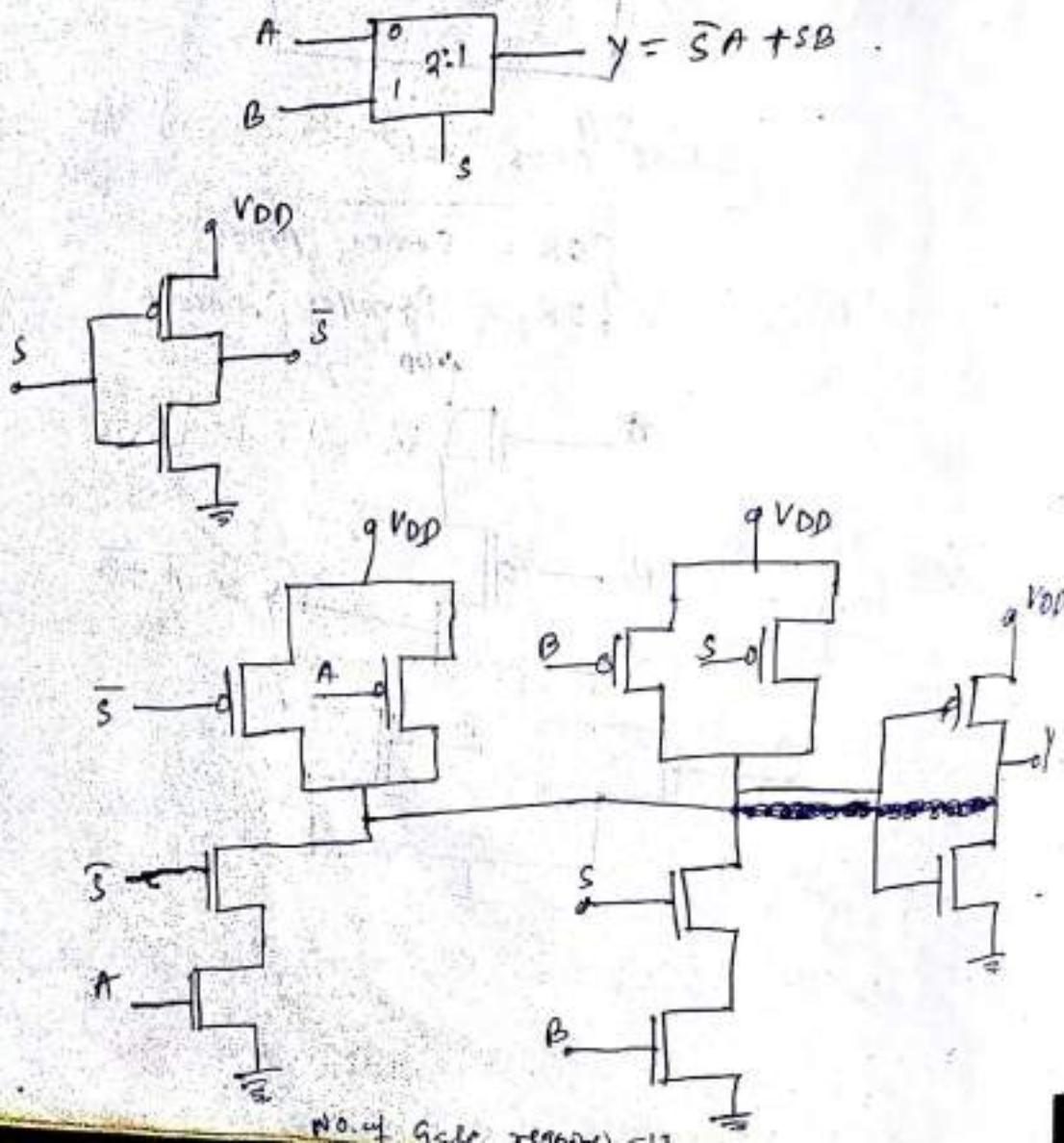
## CMOS 2*i*/P NOR gate

{ OR = Series [PMos]  
OR = Parallel [NMos]





Q11 Design a cmos cut for 2:1 mux.



## Def<sup>n</sup> & classification of Sequential Ckt.

### Sequential Ckt:

Present O/P is a fun<sup>n</sup> of not only the present i/p but also past i/p's.

→ In Synchronous Sequential Ckts the time is discretised using clock i/p.

State of a machine  
→ State captures the 'relevant' history of i/p's in a compact form.

\* Once the digital system is alive i.e once switch 'on' to digital system & if we want to store all the previous i/p over a time, we have to store increasingly larger amount of i/p's, which is not possible. So "State of the machine captures the required i/p's from the history".

→ Here memory is used to store the i/p's only for the system. So all Finite State machine, needs the finite <sup>amount</sup> of memory.

### Finite memory system

only a finite no. of past i/p's are required to generate the present O/P.  
Ex: Pattern Recognition.

## Infinite memory Systems

All the past inputs are required  
to generate the present O/P.

Ex parity generator.

Note  
Both systems can be implemented using finite state  
machine (FSM).  
Representation

→ A synchronous sequential Ckt (machine M)  
is a five tuple. Finite state ~~on~~ machine,  
consider a synchronous seq.  $M = \{ I, O, S, f, g \}$

→  $\checkmark I = \text{I/O set}$

$\checkmark O = \text{O/P set}$

$\checkmark S = \text{state space}$

$\checkmark f$  is a fun<sup>n</sup> mapping  $I \times S \Rightarrow O$

$$f: I \times S \rightarrow O$$

$\checkmark g$  is a fun<sup>n</sup> mapping  $I \times S \Rightarrow S$ .

$$g: I \times S \rightarrow S$$

Ex - 1

Parity generator

→ Define two states ( $s_0, s_1$ )

① 10101101

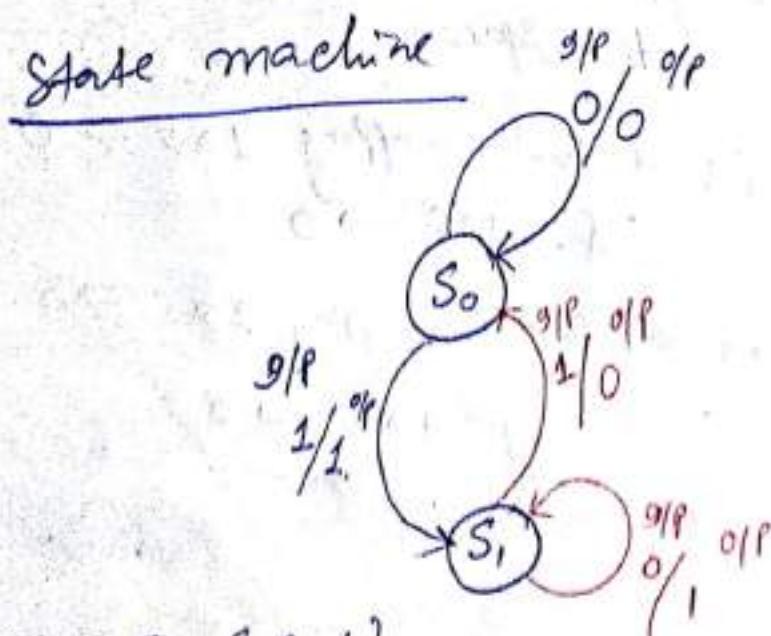
No. of '1' is = 5  $\Rightarrow$  odd.  $P = 1$   
 $o/p = P = 1$

② 101101101

No. of '1' is = 6  $\Rightarrow$  even.  $P = 0$ .  
 $o/p = P = 0$ .

$s_0$ : no. of 1's received till now is even.

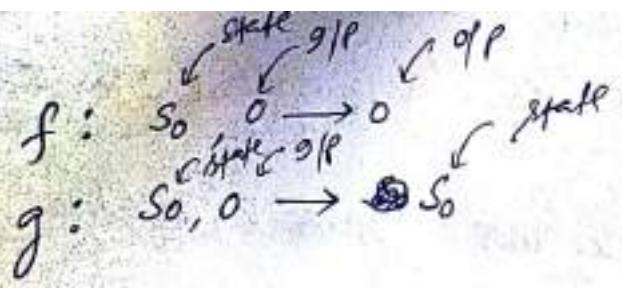
$s_1$ : " " " " " " odd.



$$g/p \text{ set} = I = \{0, 1\}$$

$$o/p \text{ set} = O = \{0, 1\}$$

$$\text{State set} = S = \{s_0, s_1\}$$



Again when  $g$  am in state  $S_1$ , if I get  $i/p = 0$ ,  
 the no. of '1' remains odd  $\Rightarrow$  ~~off~~ <sup>also</sup>  $i/p = 1$  & moves  
 to state  $S_1$ . and so on.

Ex-2

## Pattern Recognition

$$P = 1101$$

$S_0$ : No match till time  $t$ .

$S_1$ : 1-bit match till time  $t$ .

$S_2$ : 2-bit match till time  $t$ .

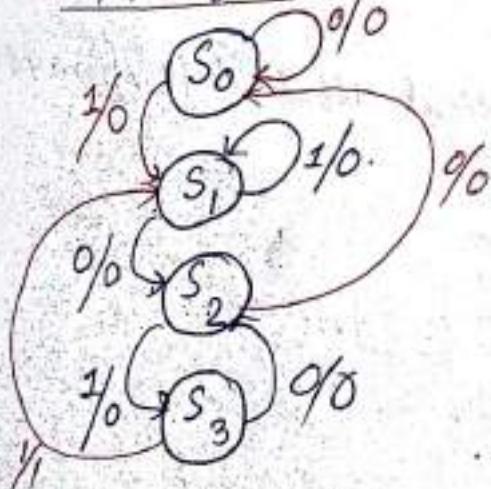
$S_3$ : 3-bit match till time  $t$ .

Specification

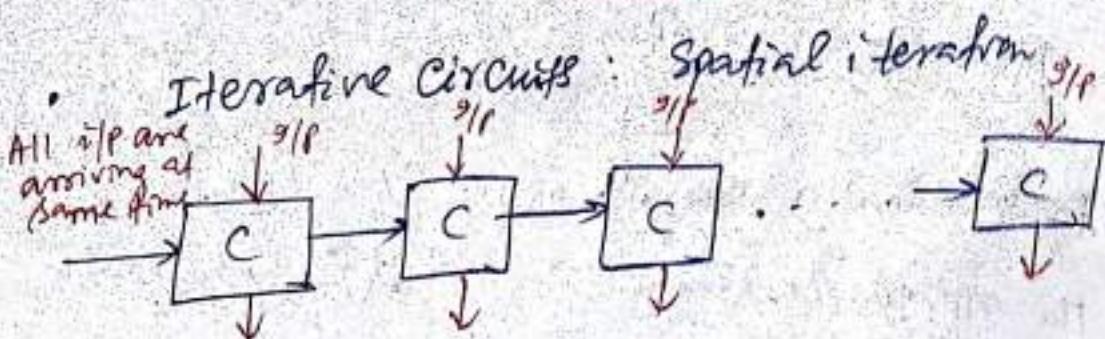
$$y(t) = \begin{cases} 1, & \text{if } \langle n_t, n_{t-1}, n_{t-2}, n_{t-3} \rangle \\ & = 1101 \\ 0, & \text{otherwise} \end{cases}$$

$n_t$   $\rightarrow$  current  $i/p$   
 $n_{t-1}$   $\rightarrow$  previous  $i/p$

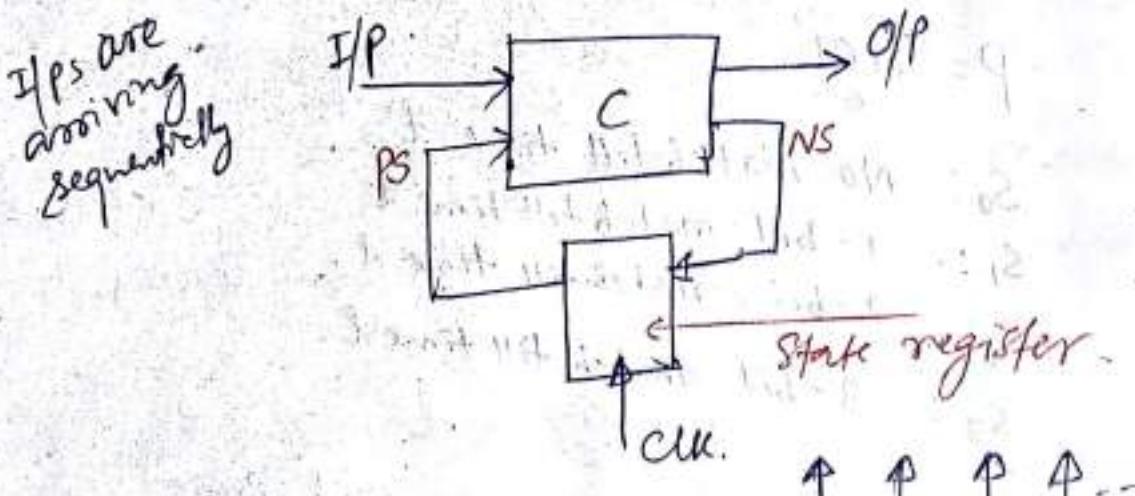
State diagram



## Temporal Iteration vs. Spatial Iteration



State machines : Temporal Iteration



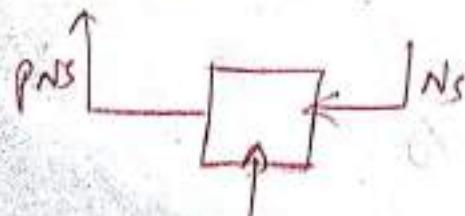
$$f: I/XS \rightarrow O$$

$$g: I/XS \rightarrow S$$



State register

It produces the present Next State from NS when it is clocked.



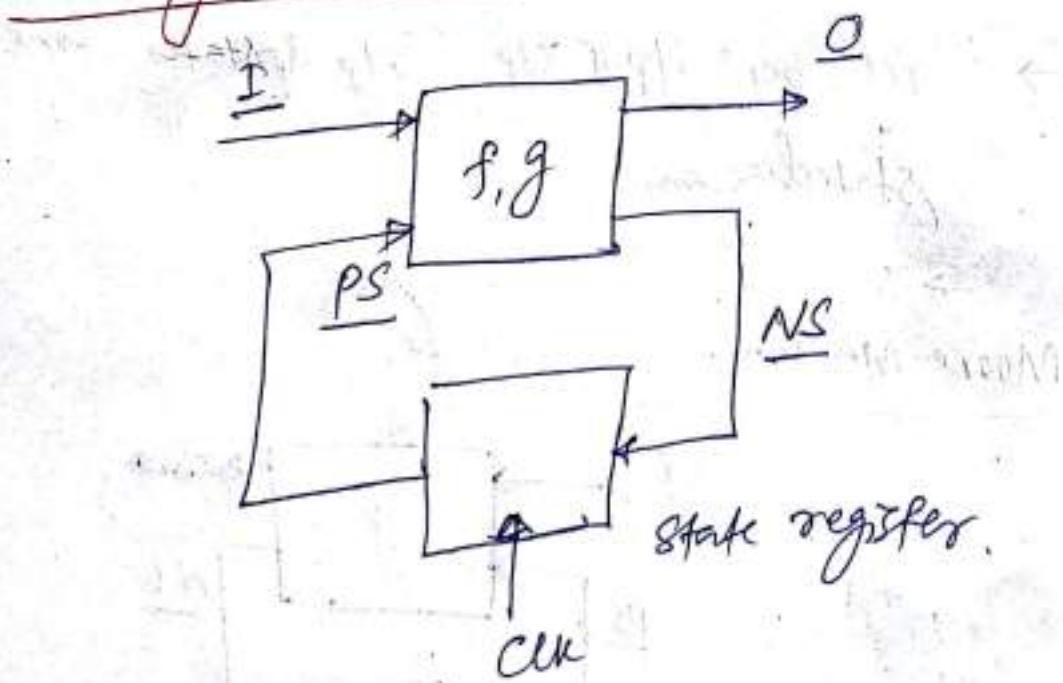
## Implementation of Sequential Ctl.

There are two different ways to implement sequential Ctl.

1. Mealy Machine

2. Moore Machine

Mealy Machine



$$f: I \times S \rightarrow O$$

$$\Rightarrow O(t) = f(i(t), s(t))$$

$$\Rightarrow O(t) = f(i(t), ps(t)) \Rightarrow O = f(i, ps)$$

$$g: I \times S \rightarrow S \quad \left( \text{generation of next state} \right)$$

$$s(t+1) = g(i(t), s(t))$$

$$ns = g(i, ps)$$

$f = \text{Opp fun}^n$   
 $g = \text{Next state fun}^n$

$f$  &  $g$  are implemented

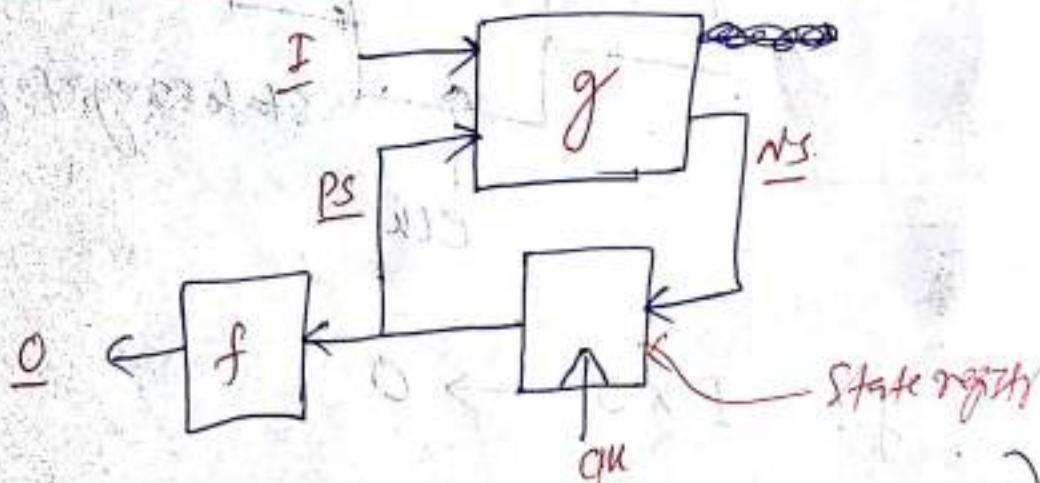
by Combinational cat

State registers = Store the "inf" regarding the state

→ Designing a seq. cat = Design of Combinational cat  
+  
Design of State register.

→ rel<sup>n</sup> bec<sup>n</sup> if  $p \neq 0$ ,  $p \neq 1$  states are from  
state diagram.

Moore Machine



→ In this case Present opp  $\neq f(\text{Present iff})$

$$f: S \rightarrow O$$

$$O(t) = f(S(t))$$

$$S = f(P)$$

$$g : I \times S \rightarrow S$$

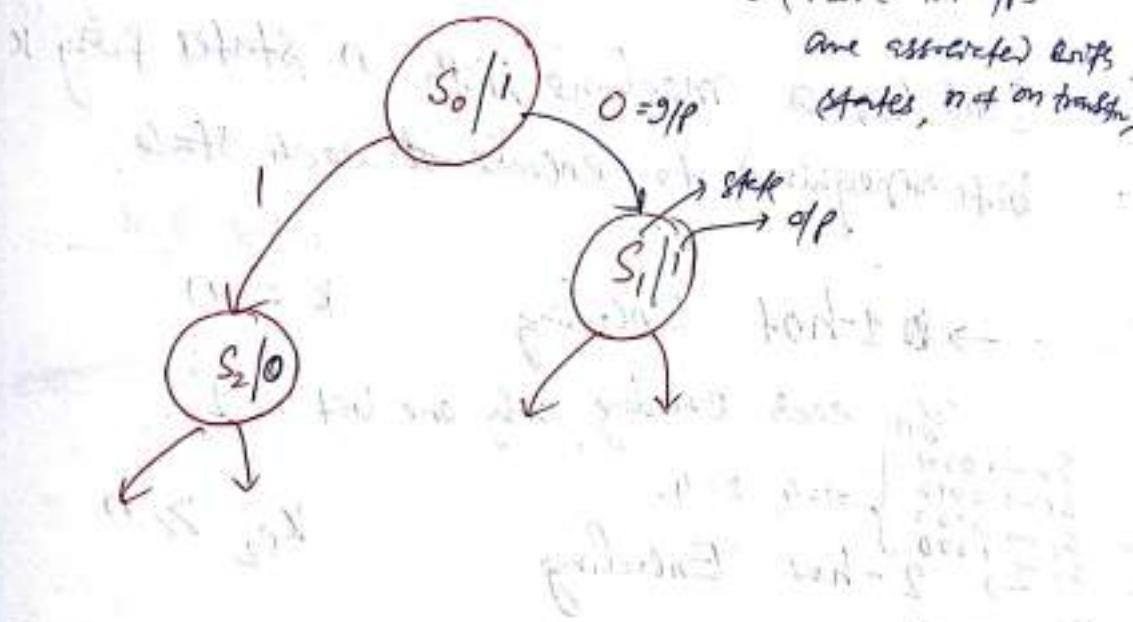
$$S(t+1) = g(i(t), S(t))$$

~~$$q_{ns} = g(i, p_s)$$~~

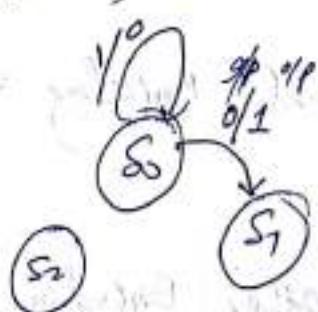
$\Rightarrow$  here also  $f$  &  $g$  are implemented by Combinational Circ.

### State machine for Moore Machine

① (here the ops  
are associated with  
states, not on transi-



In Mealy Machine, the ops  $S$  are associate  
with transition.



Note

In both Mealy machine & Moore Machine,  
 $f$  &  $g$  are implemented using Combinational Circ.  
(use T.T & design).

→ Designing of State registers.

\* Designing of State registers depends on State Encoding (i.e. the size of State registers depends on State Encoding).

State Encoding

Consider a machine with 'n' states & say 'k' bits are required to encode all each state.

→ 1-hot Encoding

$$\frac{\text{Long } \neq k}{K = n}$$

in each Encoding, only one bit is '1'.

$$\left. \begin{array}{l} S_0 \rightarrow 0001 \\ S_1 \rightarrow 0010 \\ S_2 \rightarrow 0100 \\ S_3 \rightarrow 1000 \end{array} \right\} n=4, K=4.$$

→ 2-hot Encoding

$$K_{C2} > n$$

$$\left. \begin{array}{l} S_0 \rightarrow 0011 \\ S_1 \rightarrow 0101 \\ S_2 \rightarrow 1001 \\ S_3 \rightarrow 1100 \end{array} \right\}$$

→ Minimal Encoding

$$K = \log_2 n$$

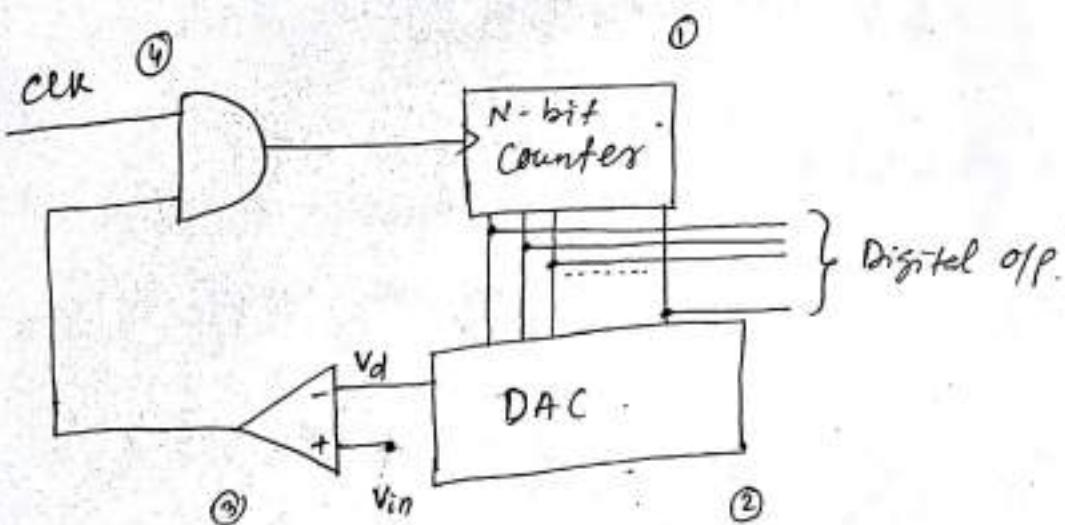
→ Any other Encoding

$$n > k > \log_2 n$$

- ADC
1. Counter type
  2. Successive Approximation
  3. Flash ADC
  4. Dual slope ADC (Integrating type)

①

Counter type



Counter followed by DAC = Stair Case O/P.

In the above figure O/P of  $N$ -bit Counter is parallelly given as i/p to DAC. The O/P of DAC ( $V_d$ ) given to inverting terminal of the Comparator & other terminal to Comparator is the analog i/p ( $V_{in}$ ).

Whenever  $V_d \geq V_{in}$ , O/P of Comparator is 1 and the O/P of Comparator is given as i/p to AND & other i/p to AND gate is the clk. If O/P of Comparator = 0, then the O/P of AND gate will be '0' i.e. no clock to Counter. So Counter will stop & that value of O/P of Counter

will be the digital opp of the given Analog  
( $V_{in} = 9V$ ).

Analysis

Let  $V_{in} = 5V$  & initially the counter is clear.

so  $V_d = 0$ .

$V_{in} > V_d \Rightarrow$  O/p of Comparator = 1

so O/p of AND Gate =  $clk \cdot 1 = clk$  &

This is clocked to Counter & Counter  
will increase its opp by 1. (i.e. 000...1)

& DAC opp will be 1. once again

$\rightarrow V_{in} > V_d \Rightarrow$  O/p of Comparator = 1

$\Rightarrow$  O/p of AND Gate =  $clk \&$  Counter will  
increase its opp by 1 (i.e. 000...10) &  
DAC opp will be '2'.

$\rightarrow$  And again  $V_{in} > V_d$ ,

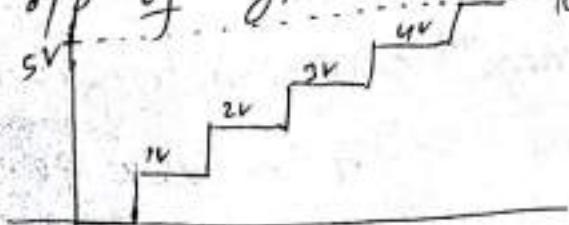
it will continue till  $V_{in} = V_d$  & when

$V_{in} = V_d$ , O/p of Comparator = 0 & O/p AND Gate = 0,

$\Rightarrow$  no clk to Counter  $\Rightarrow$  Counter will stop now.

And that value of opp of Counter will be 11-  
and that value of opp of Counter will be 11-

digital opp of given Analogue if  $V_p = V_{in}$ .



→ If  $V_{in} = 5.2V$ , it will be quantized to next integer value = 6V. & then converted into digital value.

Q11 Is conversion time dependent on i/p or not?

→ In this ADC conversion time depends on the magnitude of the i/p. If i/p is more then conversion time, it will take more.

Q What is the max<sup>m</sup> conversion time?

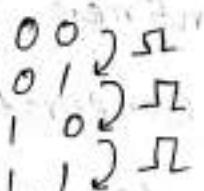
max<sup>m</sup> conversion time is defined as the total time taken by ADC to convert all '0' to all '1'.

$$0V \rightarrow 0000$$

$$1V \rightarrow 0001$$

$$15V \rightarrow 1111$$

Ex For 2-bit Counter type ADC,

  
i.e. 3 clk pulses are required, i.e.  
max<sup>m</sup> conversion time = 3T.

→ For N-bit Counter type ADC,

$$\text{max}^m \text{ conversion time} = (2^N - 1) T$$

T = clk period

N = size of ADC.

Disadv.

As max<sup>m</sup> conversion time is too large ~~so~~ so very slow  
To avoid this will go for other ADC i.e Successive Approximation ADC.

②

-: Successive Approximation type ADC:-

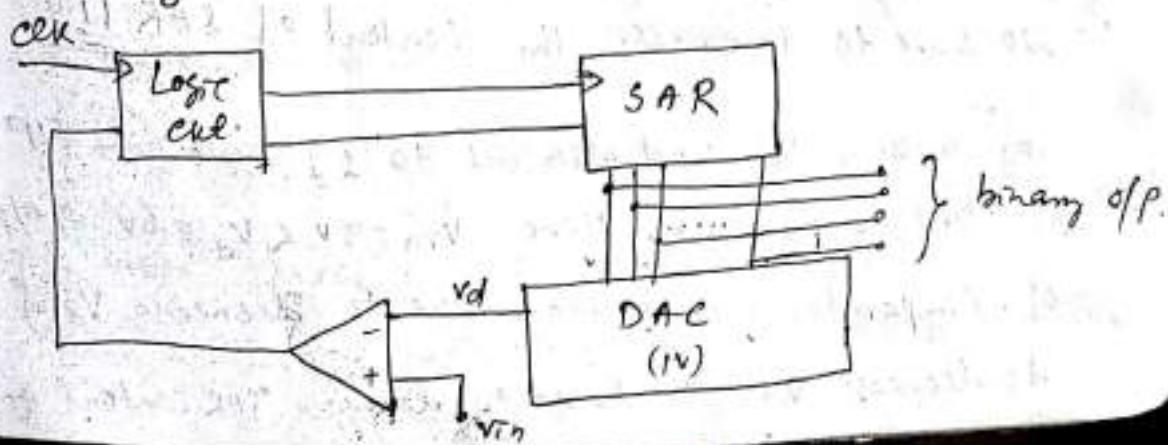
Successive Approximation type ADC has a fixed conversion time which is not dependent on the value of the analog i/p.

AND replaced by logic clk.

Counter  $\rightarrow$  Internal Register "SAR".

The block diagram of Successive Approximation type ADC consists of a DAC, Successive Approximation Register (SAR), a Comparator, Logic clk.

The basic operation is as follows, the bits of DAC are enabled one at a time, starting with the MSB ( b'cse successive Approp. type ADC is always starting at the middle & going up & going down)



- As each bit is enabled, the Comparator produces an o/p that indicates whether the analog input voltage ( $V_{in}$ ) is greater or less than the o/p of DAC ( $V_d$ ).
- If  $V_d > V_{in}$ , o/p of Comparator = 0, causing the bit in the control register to reset.
- If  $V_d < V_{in}$ , o/p of Comparator = 1, causing the bit is retained in the control register.

→ The method is best explained by taking an example, let's consider a 3-bit SAR ADC. If the o/p of the DAC ranges from 0V to 7V (000 to 111) Let's consider the unknown analog input voltage be 5V.

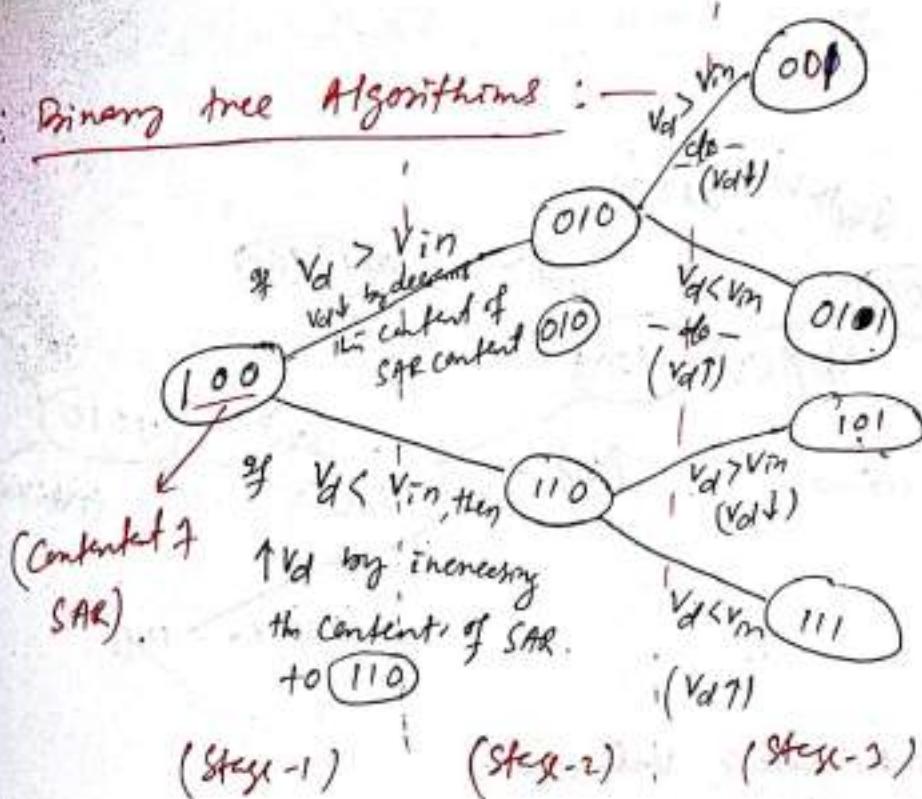
→ On the 1st CLK, the o/p of SAR is loaded with 100 the middle value (bcse we are using here Binary tree Algorithm) which is connected to 4V by DAC. So  $V_d = 4V \Rightarrow V_{in} = 5V > V_d = 4V$ . Thus the o/p of Comparator = 1. To increase  $V_d$ , ~~we~~ we have to increase the content of SAR 110.

(by making the 2nd MSB bit to 1). Now the o/p of DAC =  $V_d = 6V$ . Here  $V_{in} = 5V < V_d = 6V \Rightarrow$  o/p of Comparator = 0 so we have to decrease  $V_d$ , to decrease  $V_d$  we have to decrease SAR content to 101.

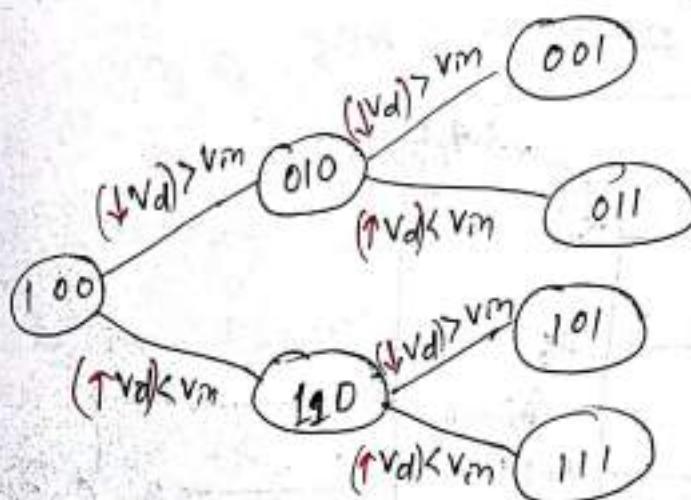
Now the o/p of DAC =  $V_d$ .  $V_d = V_m$  so  $V_m = 5V$ 's

Digital equivalent is 101.

### Binary tree Algorithms :-



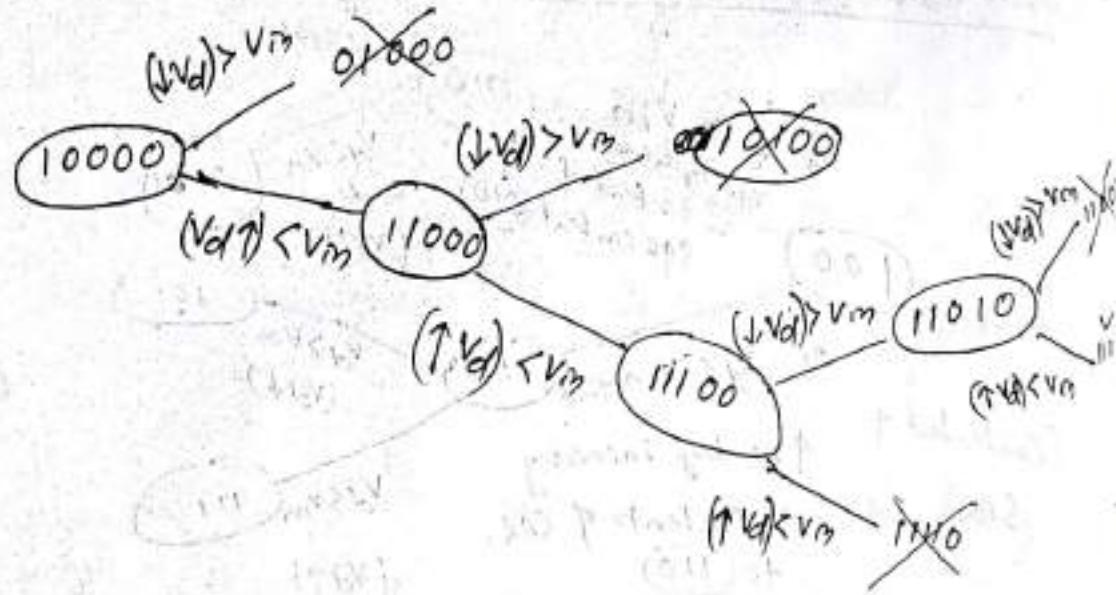
{ if i/p is  $V_m = 0$ , SAR = 001 }



⇒ Conversion time does not depend on the magnitude of the i/p. It only depends on the no. of stages.

⇒ Max<sup>m</sup> conversion time =  $N \times T$ .  
N = No. of bits. = ?  
 $T$  = clock period.

Q1 The final value of a 5-bit SAR is  $11011$ . What are the intermediate values?

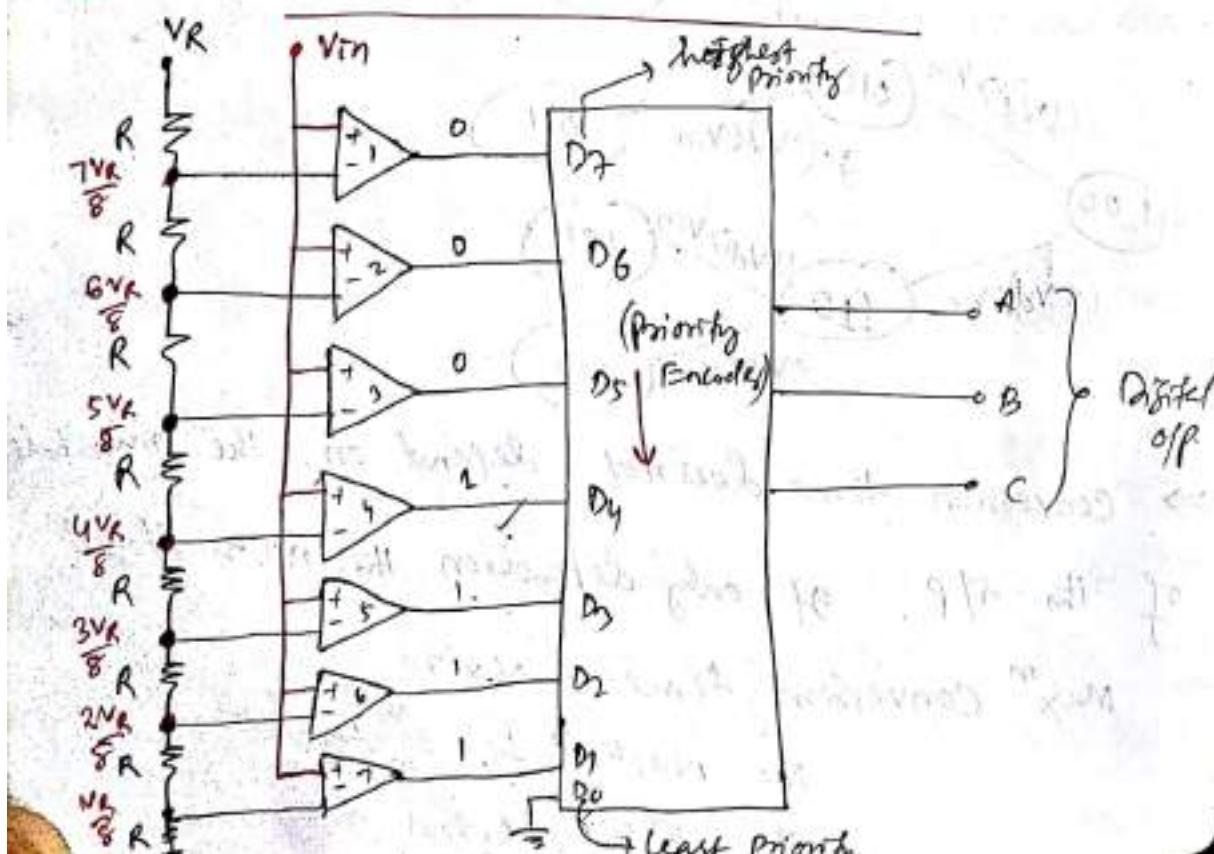


So the intermediate values are

10000, 11000, 11100, 11010, 11011

③

3-bit Flash type ADC



As the input impedance of Comparators is very high  
 so all 8 'R' resistors are connected in series, so the voltage  
 will be divided at each node as  $\frac{7V_R}{8}, \frac{6V_R}{8}, \frac{5V_R}{8}, \frac{4V_R}{8},$

$$\frac{3V_R}{8}, \frac{2V_R}{8}, \frac{V_R}{8}$$

$$\rightarrow \text{Let's consider, } \frac{4V_R}{8} < V_m < \frac{5V_R}{8}$$

$$\text{so } V_m < \frac{7V_R}{8} \Rightarrow \text{O/p of 1st Comparator} = 0$$

$$V_m < \frac{6V_R}{8} \Rightarrow \text{, " 2nd, " } = 0$$

$$V_m < \frac{5V_R}{8} \Rightarrow \text{, " 3rd, " } = 0$$

$$V_m > \frac{4V_R}{8} \Rightarrow \text{, " 4,5,6,7th, " } = 1,1,1$$

$V_m > \frac{4V_R}{8}$  encoder

So the I/P to Priority Decoder is

$$\begin{matrix} D_7 & D_6 & D_5 & D_4 & D_3 & D_2 & D_1 & D_0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{matrix}$$

$\rightarrow$  The O/P of Priority Decoder, will consider the highest priority I/P  $D_4 = 1$  & others will be zero.

So for I/P  $\frac{4V_R}{8} < V_m < \frac{5V_R}{8}$ , Digital

O/P will be 100.

$\rightarrow$  If  $V_R = 8V$ ,  $\Rightarrow 4 < V_m < 5$  i.e. O/P = 100.

∴ Vm is quantised to 4V.

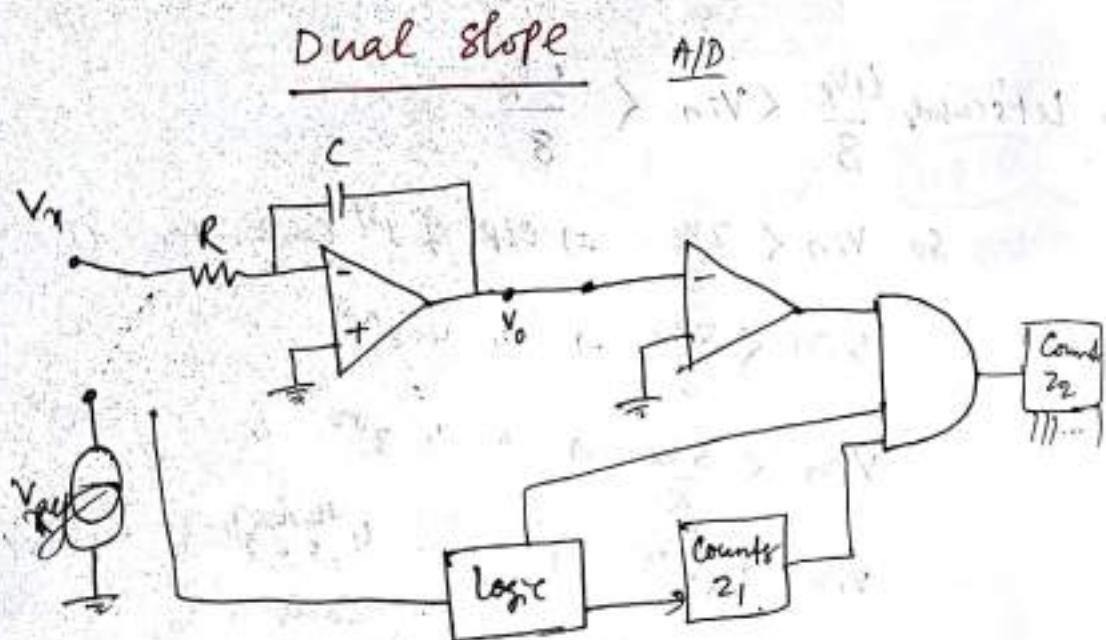
Adv.  
 → Conversion time does not depend on the magnitude of conversion time = 0  $\propto$  propagation delay

→ As clock is not required here so it is the fastest

and hence it is called parallel AtoD converter.

### Disadv.

For a  $N$ -bit flash type ADC, the no. of comparators required =  $(2^N - 1)$ .



$$V_{O_1} = ?$$

$$V_{O_1} - V_C - V_O = 0 \Rightarrow V_{O_1} - V_C = - \frac{I_c(t)}{C} = - \frac{V_R}{CR}$$

$$V_O(t) = - \frac{V_R}{RC} t.$$
$$\Rightarrow V_O \Big|_{t=T_1} = - \frac{V_R}{RC} T_1 = - \frac{V_R}{RC} T_1$$

$$T_1 = \frac{V_O \cdot RC}{V_R}$$

$$V_O = \frac{V_{ref}}{RC} T_2$$

$$\Rightarrow T_2 = \frac{V_O}{V_{ref}} \cdot RC$$

$$\boxed{\frac{T_1}{T_2} = \frac{V_{ref}}{V_R}}$$

$$\boxed{T_2 = \frac{V_R}{V_{ref}} T_1}$$

$$Z_2 = \text{no. of counts} = T_2 \cdot f_{clock} = \frac{V_R}{V_{ref}} T_1 \cdot f_{clock}$$

$\hookrightarrow$  Total time  $T_2$

~~Q2~~ Q3

$$Z_1 = T_1 \cdot f_{clock}$$

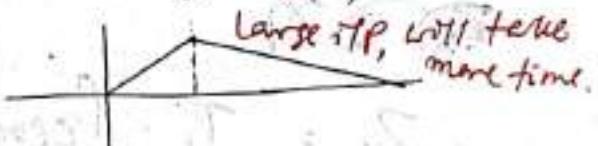
$$Z_2 = Z_1 \cdot \frac{V_R}{V_{ref}}$$

O/P is independent of RC

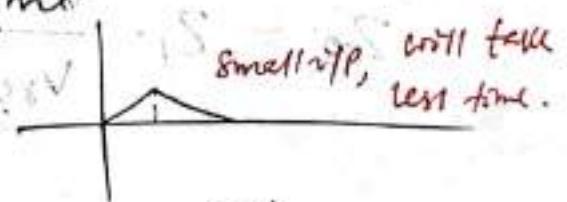
Time required is more.

→ For Dual slope ADC;

① The conversion time depends on the magnitude of r/p. i.e. for large r/p, the conversion time is more.



& for small r/p, the conversion time is less  
because it will take less time



② Max<sup>m</sup> Conversion time =  $(2^N - 1) T$ , as  
a register is interlaced inside it [000 i.e.  
all zero to 111 i.e. all '1']

### Adv.

1. It is very accurate & hence it is used in Digital voltmeter.
2. The integrator, at the r/p of ADC will eliminate the effect of noise.

Diseadv. Area under the Gaussian Signal = 0 =  $\int \text{Gaussian Signal}$

Q11 What is the resolution of 8-bit ADC whose I/P voltage range -10V to 10V.

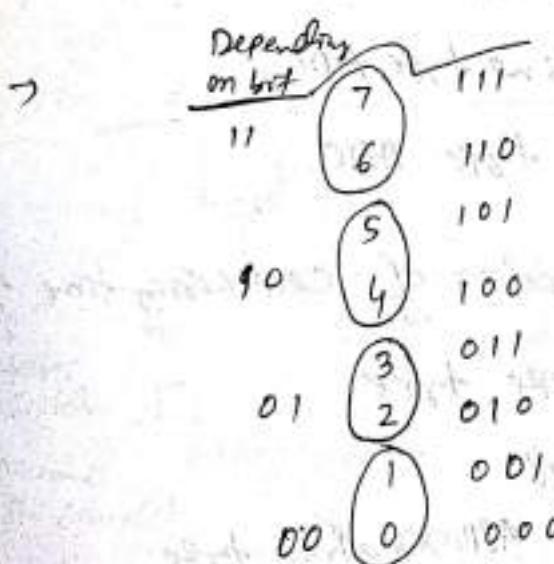
Sol: For ADC

$$\text{Resolution} = \frac{\text{I/P Voltage range}}{\text{No. of Quantization level}}$$

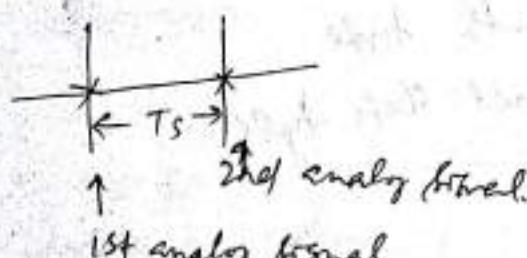
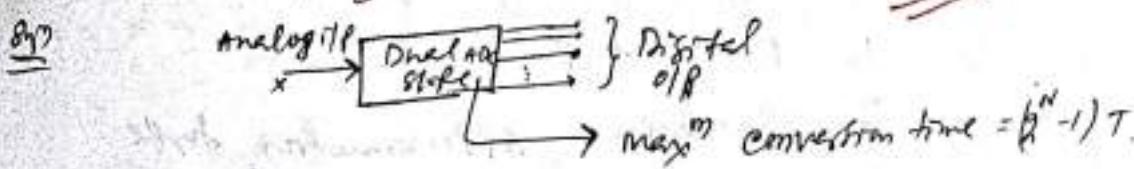
$$= \frac{10 - (-10)}{2^N} = \frac{20}{2^8} = \frac{20}{256} \text{ (Ans)}$$

Note → For a 2-bit ADC;

$\begin{matrix} 00 \\ 01 \\ 10 \\ 11 \end{matrix} \rightarrow$  there are 4 quantization levels.



Q11 What is the conversion rate of Dual Slope ADC if the clk freq is 1MHz. bit 8 bits



$T_S$  = Duration between 1st and 2nd analog terminals.

$$T_S > (2^N - 1) T$$

$$T_S = (2^N - 1) T = (2^8 - 1) 1 \text{usec} = 255 \text{usec.}$$

$$\therefore \text{Conversion rate} = \frac{1}{T_S} = \frac{1}{255 \text{usec}} = \frac{1}{255 \times 10^6} \text{sec.}^{-1} \text{ps/sec.}$$

### Note

~~of ADC~~

#### 1. Decreasing order of Accuracy of ADC

1. Dual slope type ADC
2. Counter type ADC
3. Successive type ADC
4. Flash type ADC.

#### 2. Descending order of conversion time of ADC

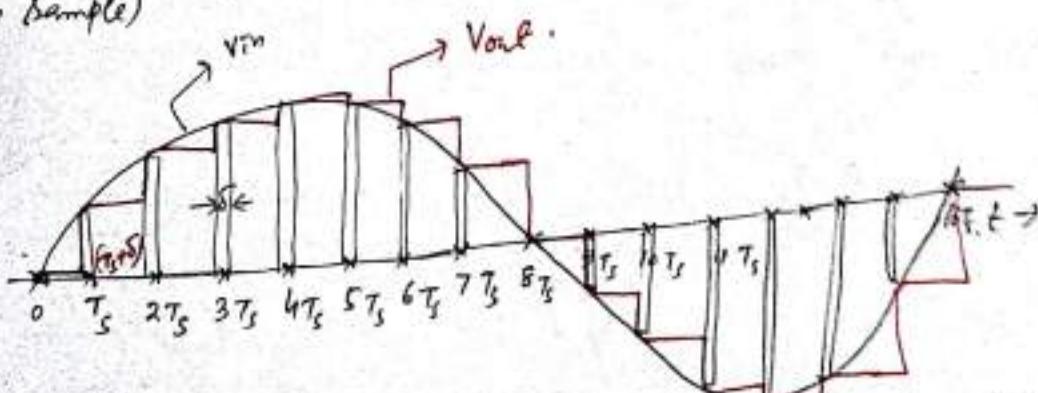
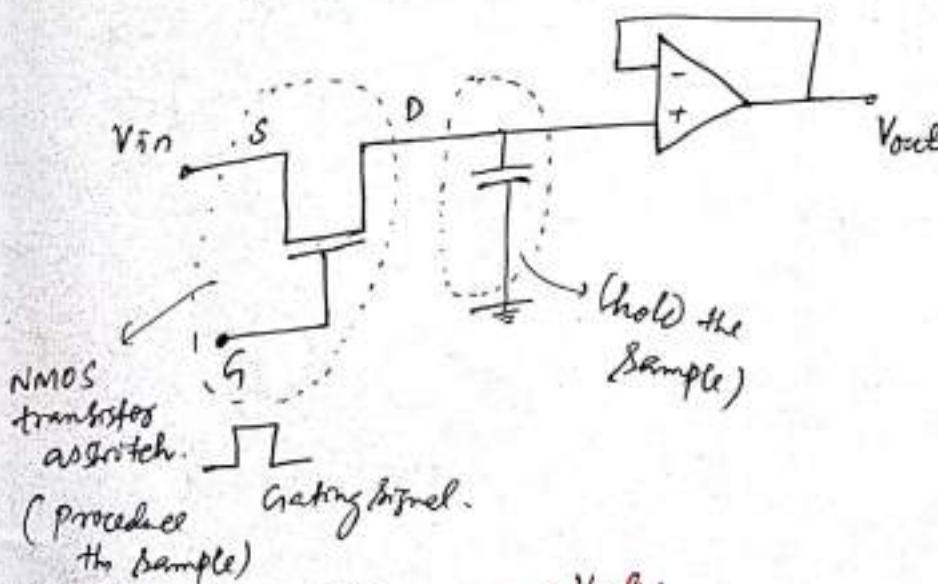
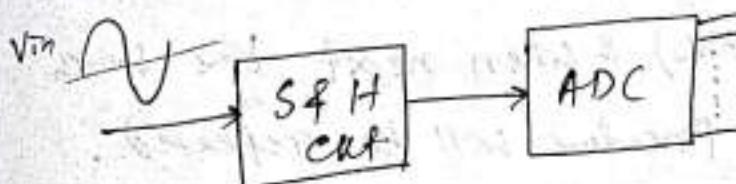
1. Dual slope type
2. Counter type
3. Successive Approximation type
4. Flash type ADC.

#### 3. Descending order of Speed of ADC.

1. Flash type
2. Successive Approximation type
3. Counter type
4. Dual slope type.

## Sample & Hold Circ.

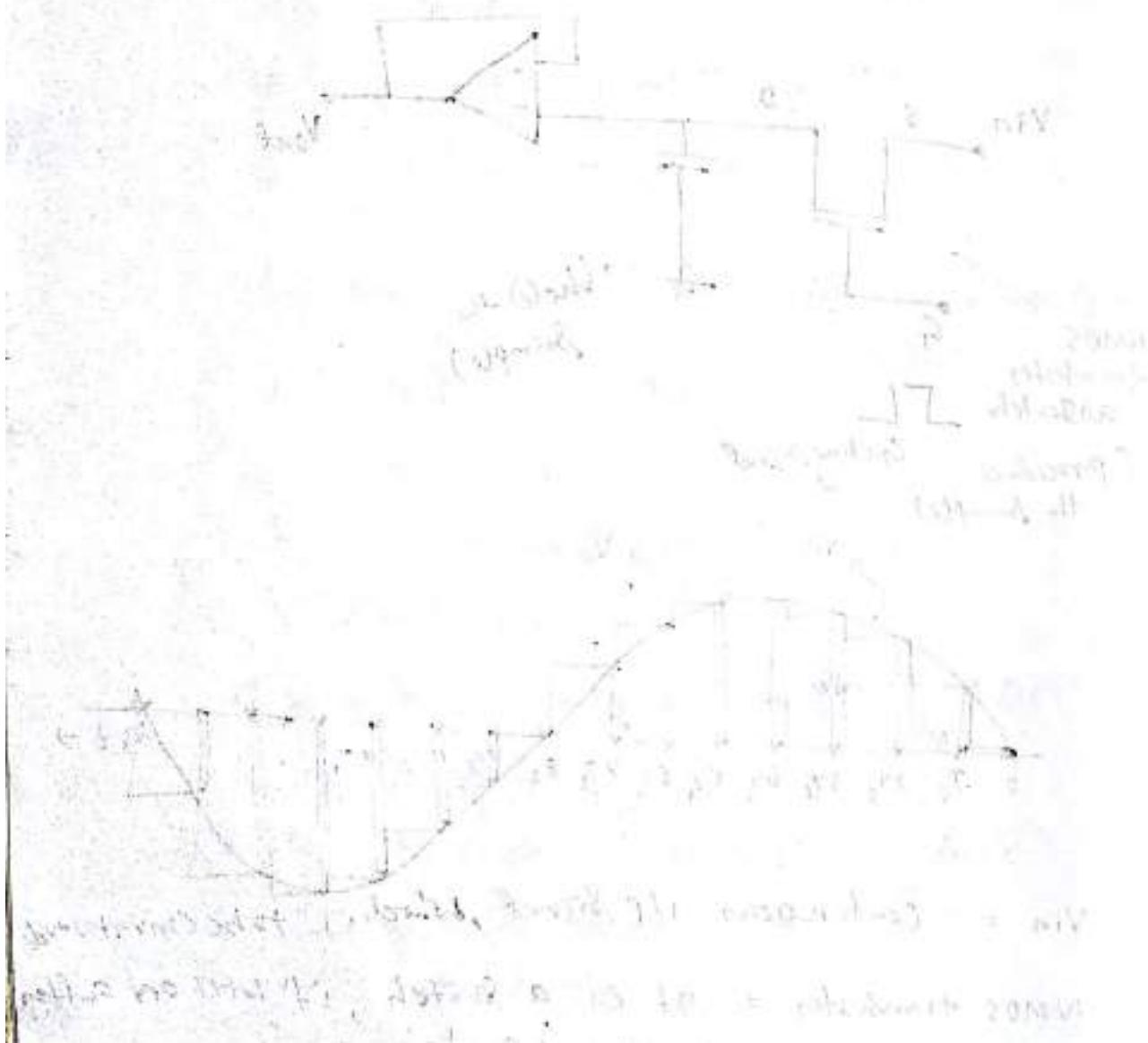
- ① → If i/p Signal  $V_{in}$  continuously changes then we can't directly convert it into digital signal.
- To convert a continuous time signal to piece-wise cont. signal we will take the help of Sample & Hold Circ preceded by ADC.



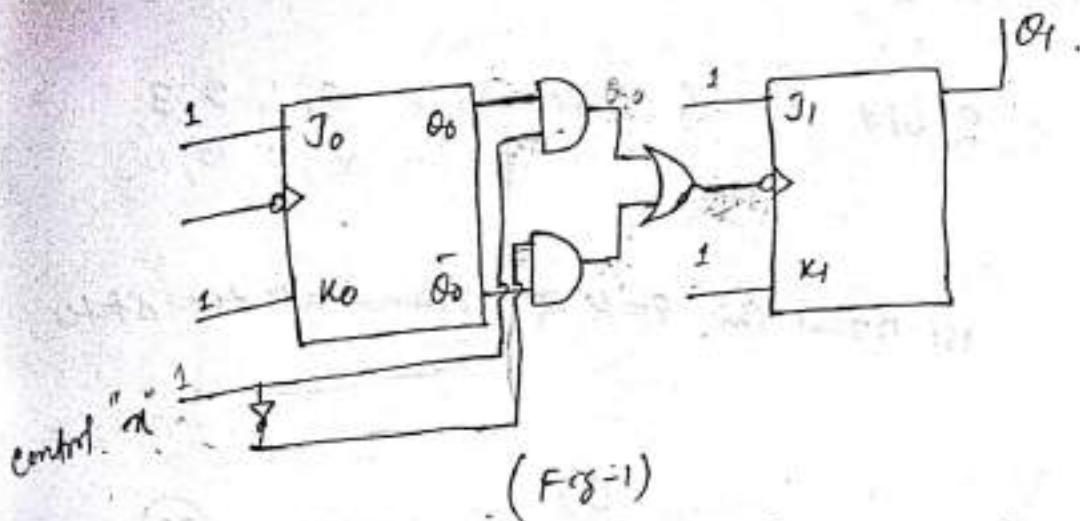
- $V_{in}$  = Continuous i/p signal, which is to be converted.
- NMOS transistor = if it is a switch, it will turn off after each  $T_s$  sec. & it will on for  $\delta t$  duration.
- During on period of switch, the capacitor

will charge immediately to i/p voltage  $V_{in}$  of during "on" duration of switch. The cap. charge follow the i/p voltage of  $V_{out}$  is through a voltage follower so  $V_{out}$  is same as i/p  $V_{in}$ .

When the switch will turn off, the cap. will not get the path for discharging<sup>and</sup> that previous peak i/p voltage will be remains constant capacitor of also at  $V_{out}$ . (This period is known as holding period = from one clock to ~~next~~ next clock) & when next, the switch turn on same procedure will be repeated.

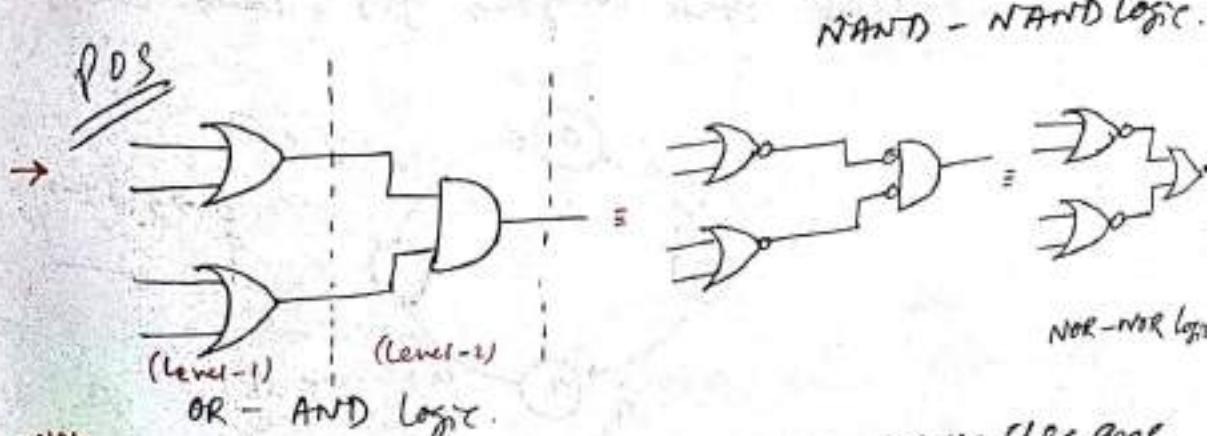
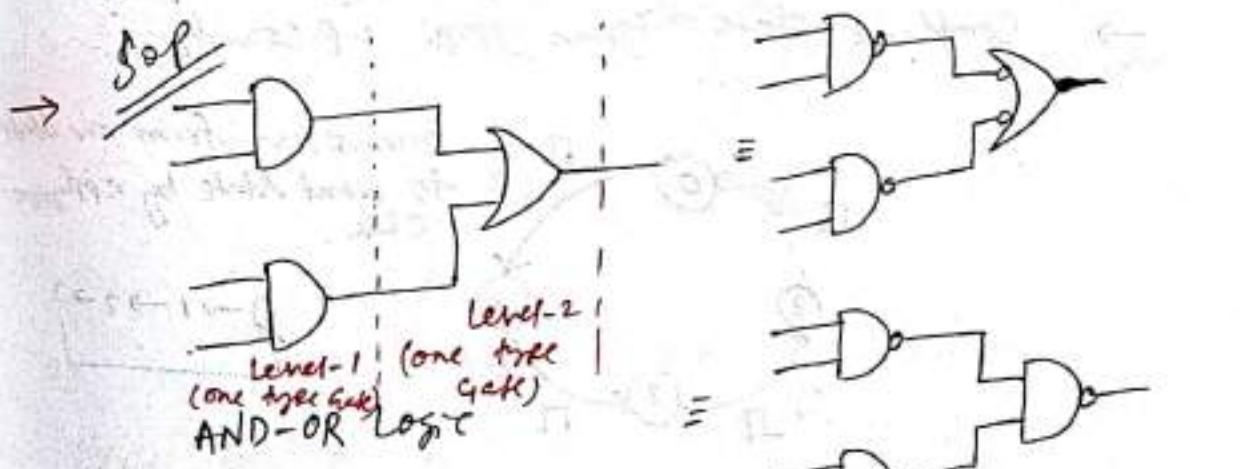


## - 2 Bit Asynchronous Up/Down Counter :-



$n=1 \Rightarrow \bar{Q}_0 = \underline{\text{clk}} \Rightarrow \text{Up Counter} (\underline{00}, \underline{01}, \underline{10}, \underline{11}, \underline{00}, \underline{01}, \dots)$

$n=0 \Rightarrow \bar{Q}_0 = \underline{\text{clk}} \Rightarrow \text{Down Counter} (\underline{00}, \underline{11}, \underline{10}, \underline{01}, \underline{00}, \dots)$

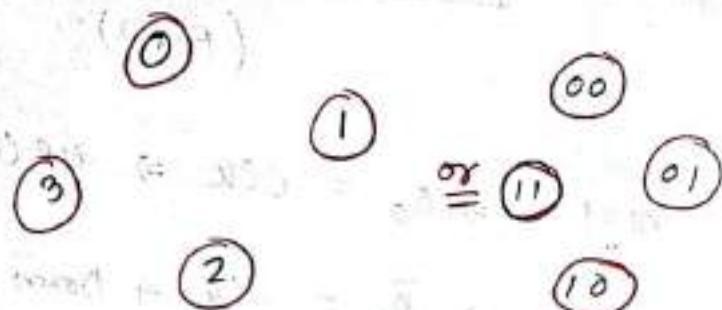


Note In the above ~~up~~ up/down counter, if the f/f's are edge trigger then for  $n=1$ , it acts as down counter.

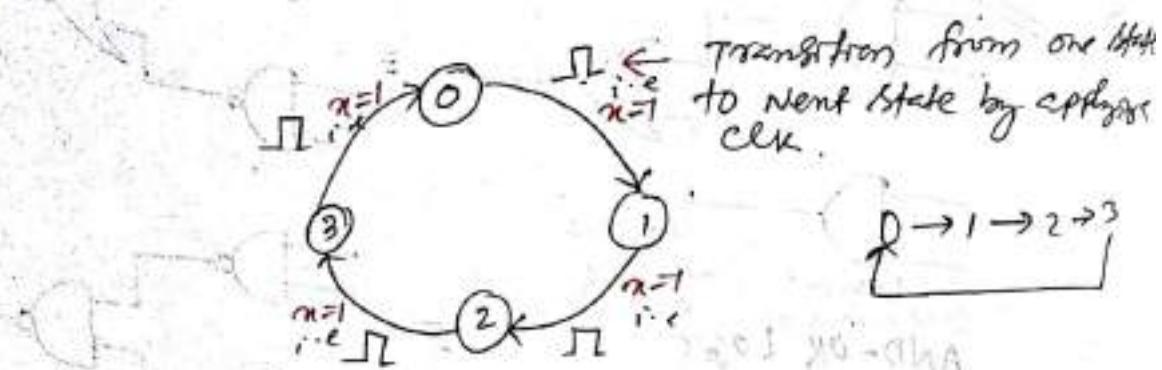
Q11 Draw the state diagrams of the following  
(i) a 2-bit sync. up/down counter

Ans. 2 bit  $\Rightarrow$  4 states i.e. 0, 1, 2, 3  
00, 01, 10, 11.

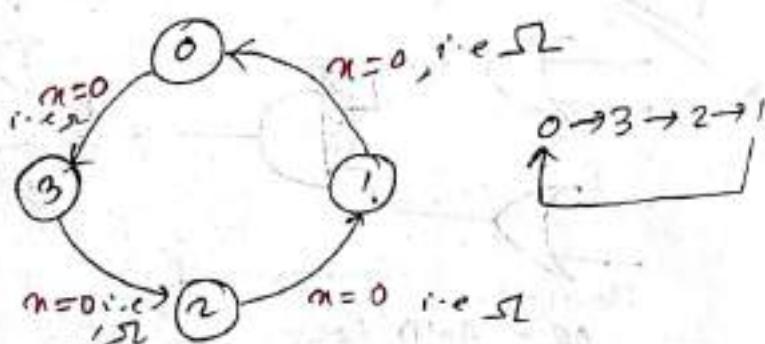
1st draw the state & round all the states.



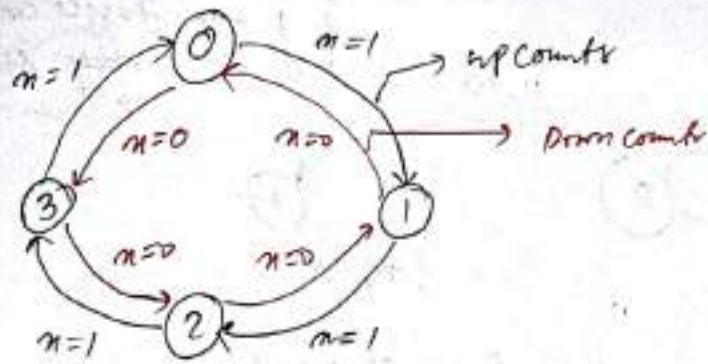
→ Complete state diagrams for up-counts.



→ Complete state diagrams for Down-counts.



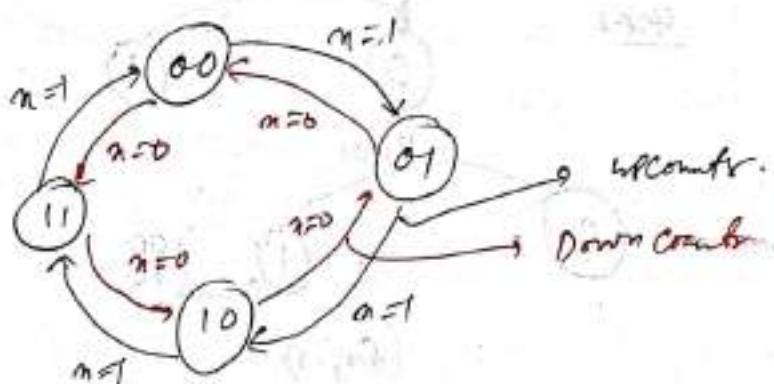
→ Complete state diagram for up/down counter.



$m = 1$ , means a  $\text{Clk} = \bar{Q}_1$  when  $n = 1$  in fig-1 [Up contr]

$m = 0$ , means a  $\text{Clk} = \bar{Q}_0$  when  $n = 0$  in fig-1, [Down contr]

or



Q) Draw the state diagram for (i) J-K FF.

(ii) D-FF (iii) T-FF (iv) S-R FF.

J-K FF

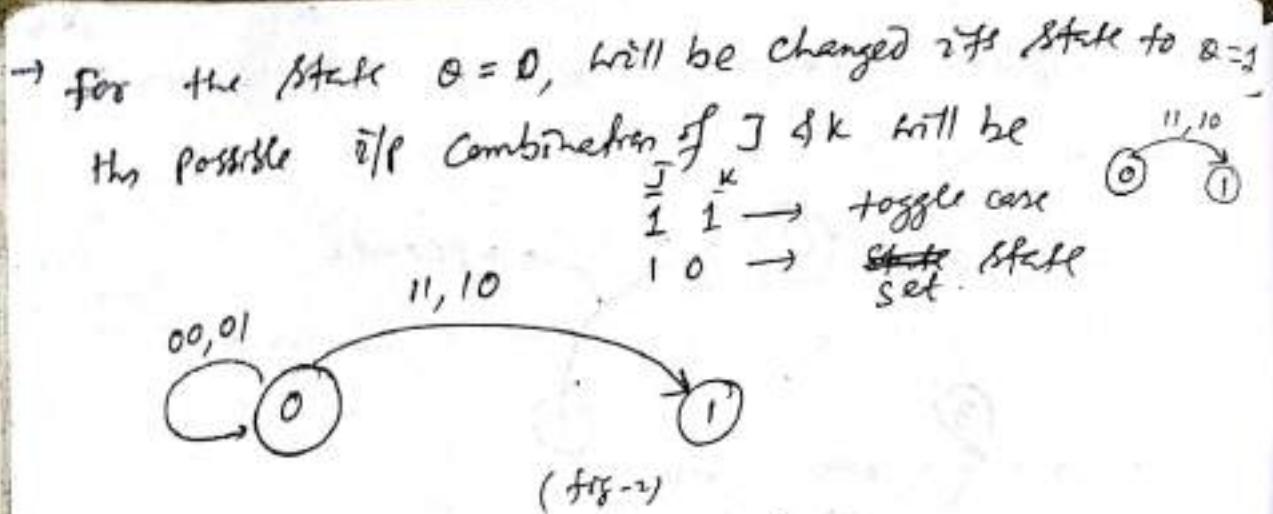
It has two states 0, 1. i.e.  $Q = 0$  or  $Q = 1$ .

0

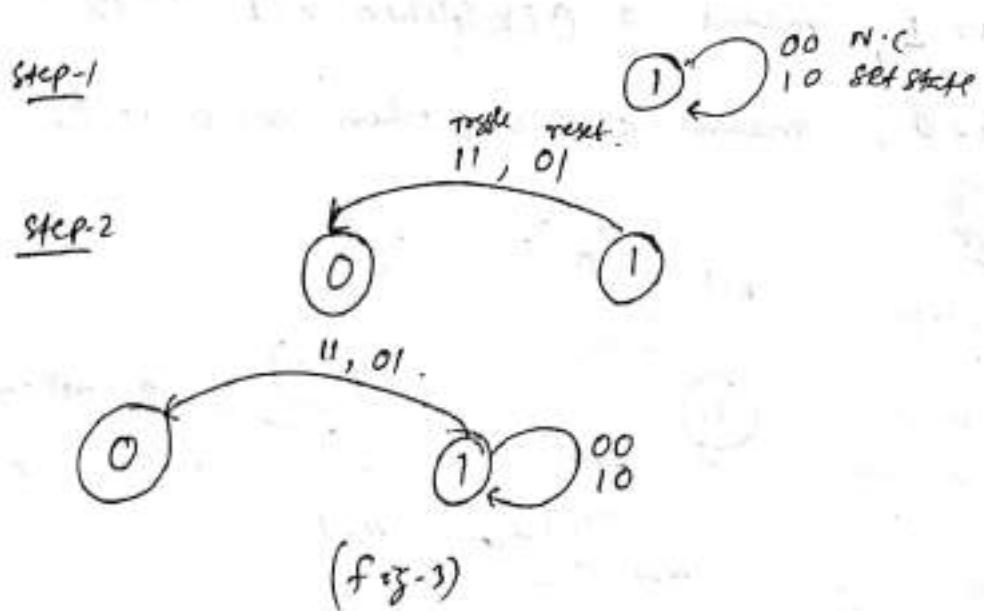
1

for the state  $Q = 0$ , will be in same state, the possible  
trips of J & K will be  $00 \rightarrow$  no change state  
 $01 \rightarrow$  next state.

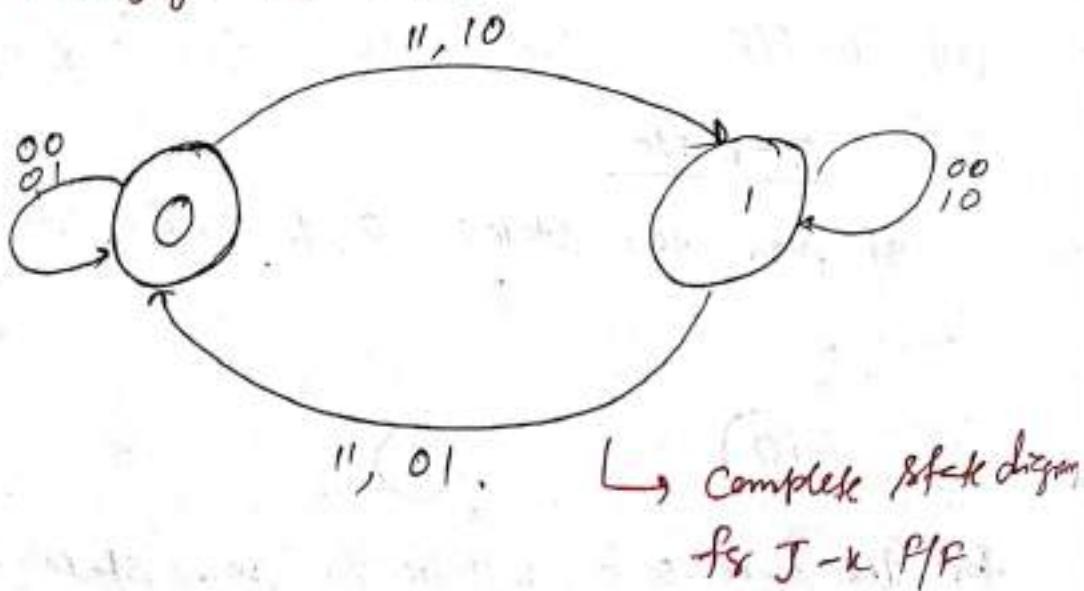
00,01  
00



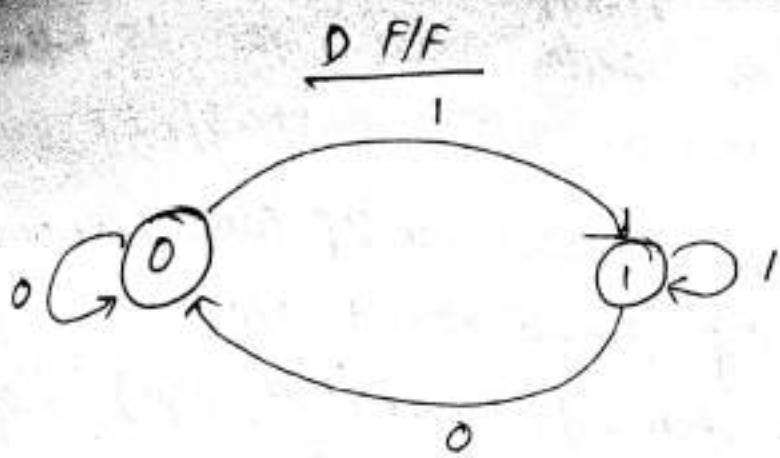
→ similar steps for state  $Q=1$ ,



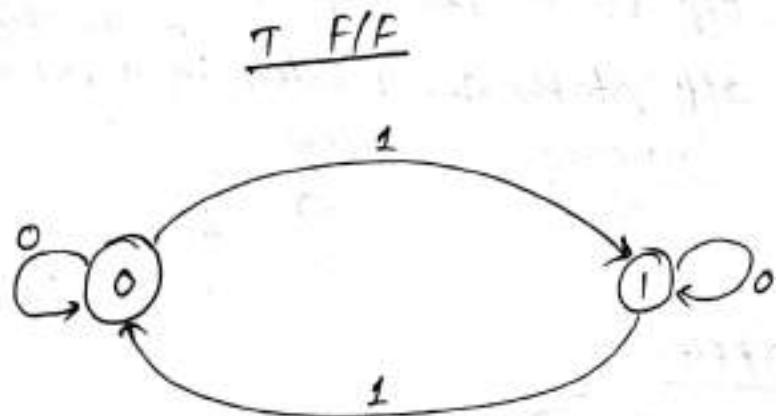
→ Combiningly fig-2, fig-3



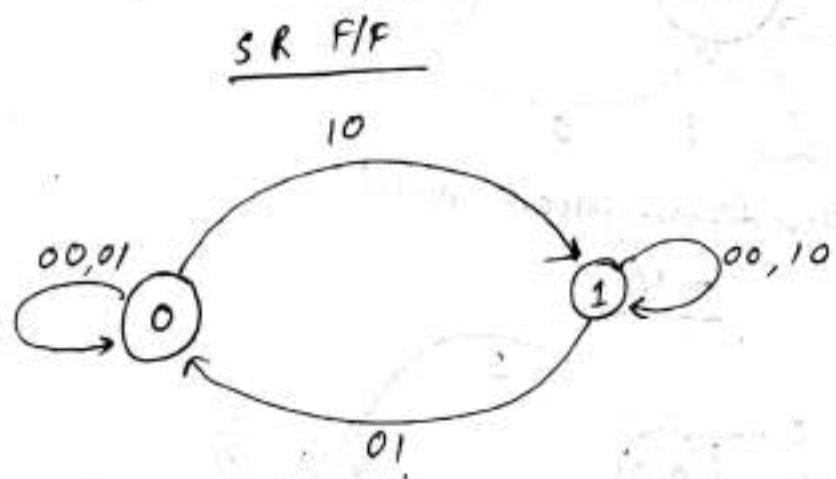
(ii)



(iii)



(iv)



Q11 Draw the ~~Mealy~~ model for the following.

- (i) D F/F (ii) T F/F (iii) J/K F/F (iv) S/R F/F

Sol In moore machine model, (i) the next state is a function of present state & i/p.

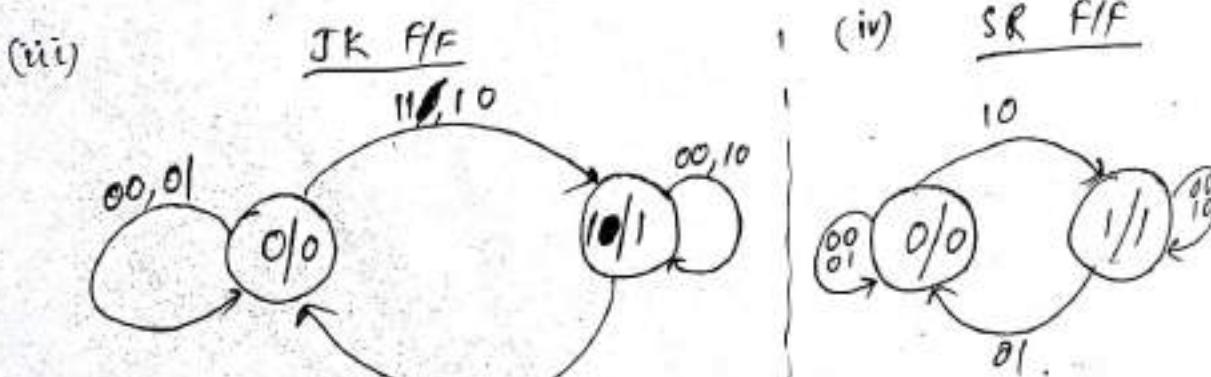
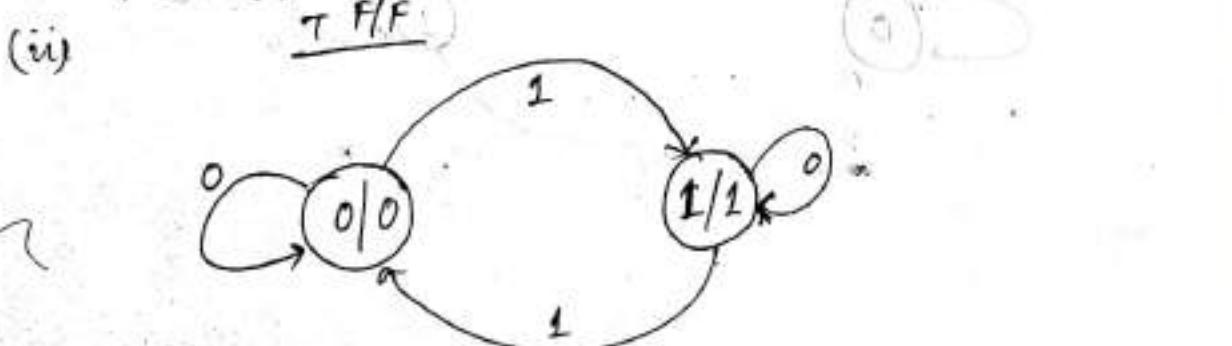
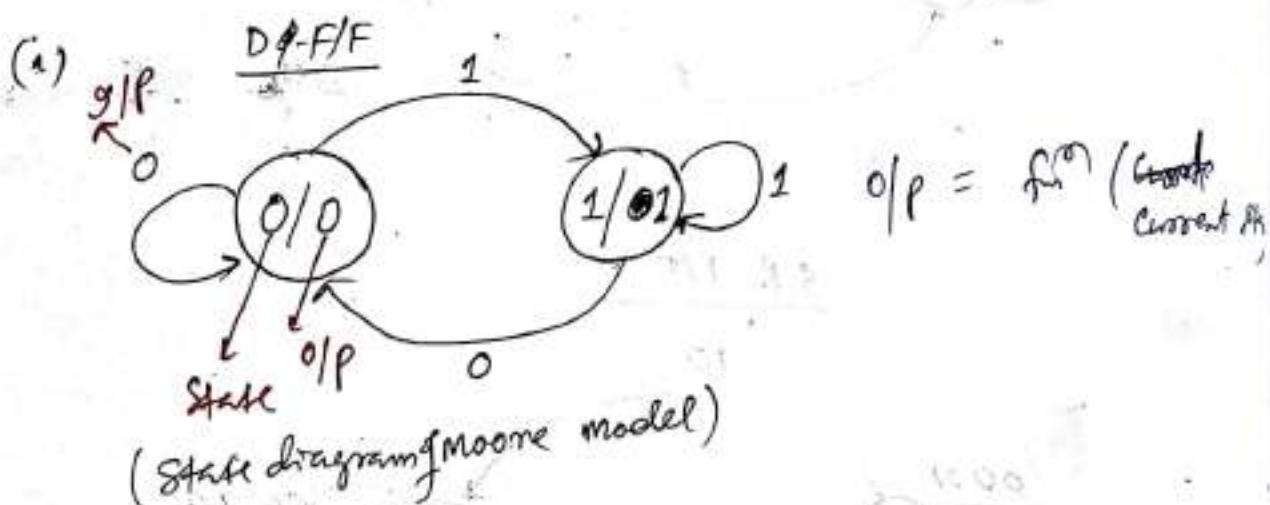
$$\text{Next state} = f(\text{Present state}, i/p)$$

(2) O/p is a function of Present State. (O/p are written along with the present state)

(3) O/p states are written in a separate column

for D F/F

Present State	Next State		O/p
0	0	1	0
1	1	0	1



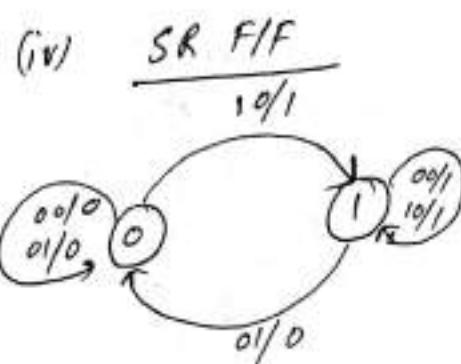
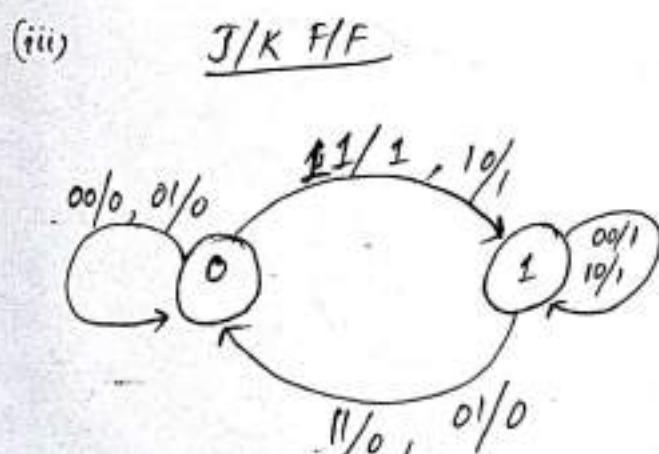
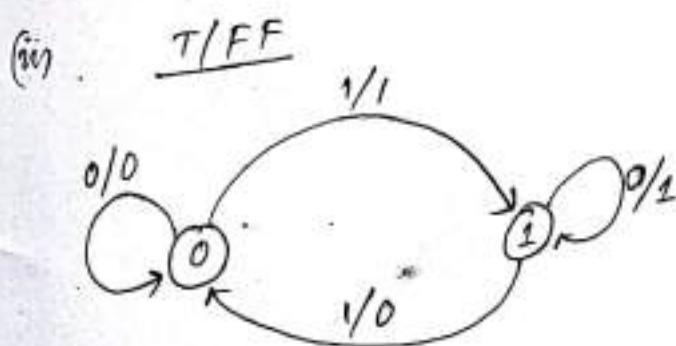
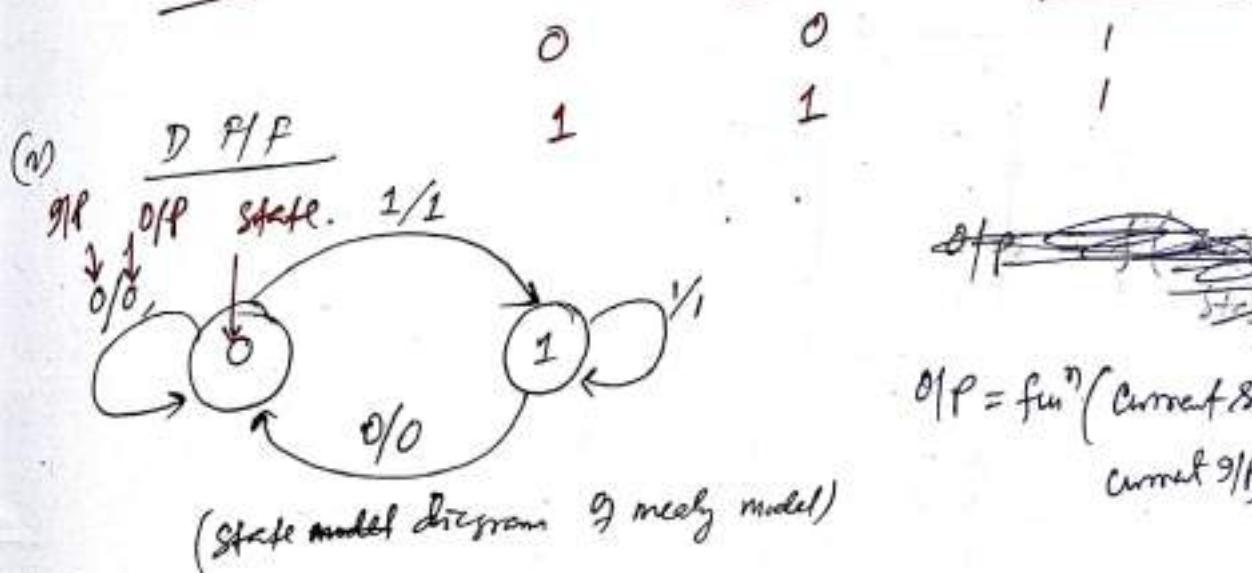
Q11 Draw the State diagram of mealy model for 1h following (i) D P/F (ii) T P/F (iii) J/K F/F (iv) S R F/F.

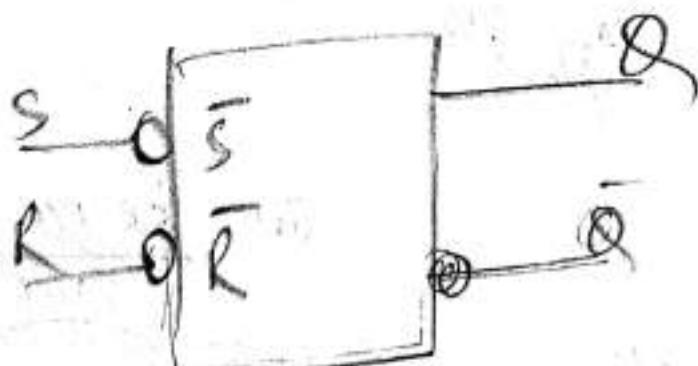
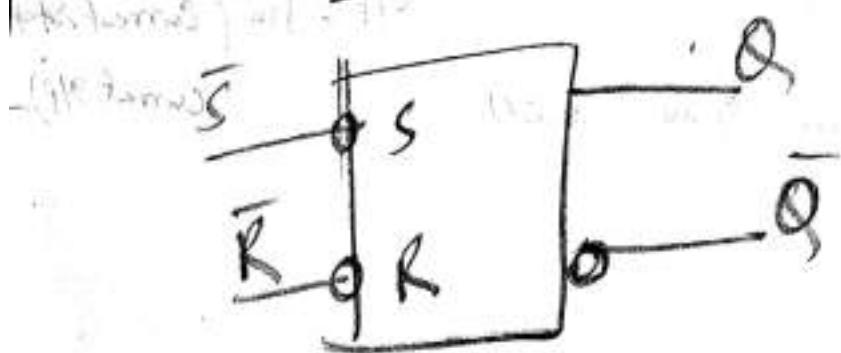
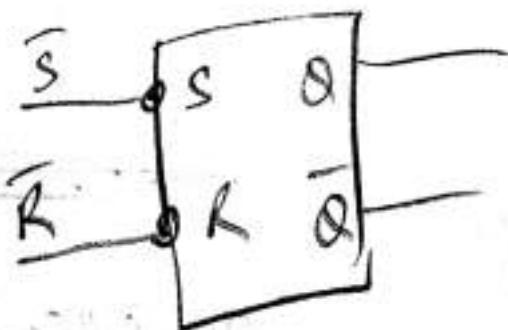
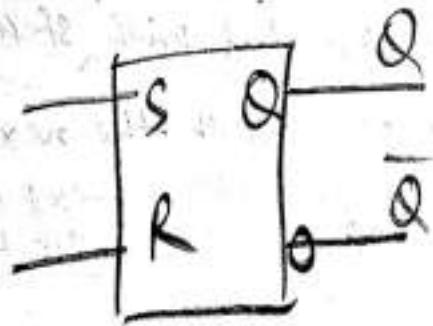
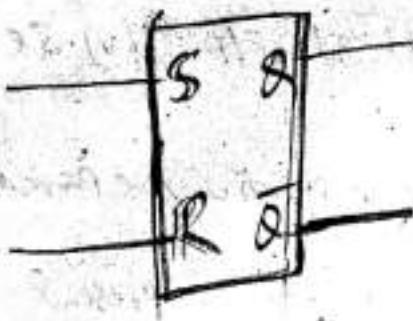
Sol For mealy machine, (i) there is no separate column for opps.

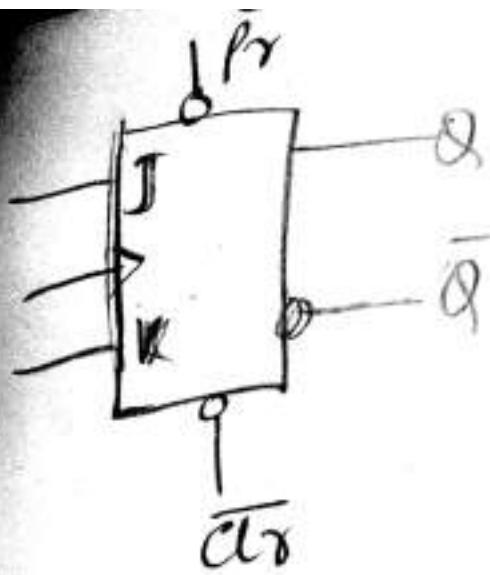
(ii) opps are not written along with states.

(iii) opps are written along with the next states.

for D P/F.	Present state	next state 9/P E=0	next state 9/P E=1
------------	---------------	-----------------------	-----------------------







$\bar{P} \cdot \bar{Q}$	$\bar{Q}$	On + Input operat
0	1	0 1
0	0	0
1	1	Normal operat
1	1	slf