08-09-2014

• Digital Electronics

→ Boolean Logical Ideas.

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

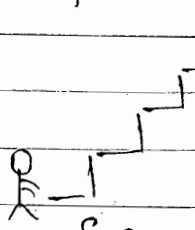
→ Boolean logical Ideas are categories in three ways.

(i) Producing the constant 0, 1 [Null, Identity] operations.

(ii) Unary operation: - Transfer, compliment. [Buffer, NOT]

(iii) Binary operation: - AND, OR, NAND, NOR, EX-OR, EX-NOR, Inhibition, Implication.

- $f_0 = 0 \Rightarrow$ NULL
- $f_1 = x.y \Rightarrow$ AND $x \cap y$
- $f_2 = x.y' \Rightarrow$ Inhibition x/y [x but NOT y]
- $f_3 = x \Rightarrow$ Buffer
- $f_4 = x'.y \Rightarrow$ Inhibition y/x [y but NOT x]
- $f_5 = y \Rightarrow$ Buffer
- $f_6 = x \oplus y$ ⇒ EX-OR $= x'y + xy'$ "ODD Function" also called Stair case logic



S_2	S_1	Bulb: f
0	0	0
0	1	1
1	0	1
1	1	0

$f_2 = S_2 \oplus S_1$

- $f_7 = x+y \Rightarrow$ OR $x \cup y$
- $f_8 = \overline{x+y} \Rightarrow$ NOR $x \downarrow y$
- $f_9 = x \odot y \Rightarrow$ EX-NOR $= x'y' + xy$ Even function

Co-incident logic gate, Equivalence logic gate

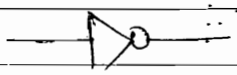
- $f_{10} = \bar{y} \Rightarrow$ NOT
- $f_{11} = x+y' \Rightarrow$ Implication $x \subset y$ (if y then x)
- $f_{12} = \bar{x} \Rightarrow$ NOT
- $f_{13} = x'+y \Rightarrow$ Implication $x \supset y$ (if x then y)
- $f_{14} = \overline{x \cdot y} \Rightarrow$ NAND $x \uparrow y$
- $f_{15} = 1 \Rightarrow$ Identity

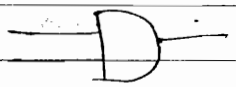
→ FOR n I/p variables :- 2^n combination

2^{2^n} possible function.

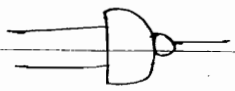
- Symbols for the logic gates

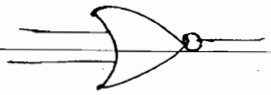

Buffer

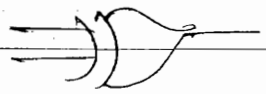

NOT


AND


OR


NAND


NOR


EX-OR


EX-NOR

→ NAND, NOR known as universal logical gates.

Shortcut:- NAND (4) NOR.

NOT	1	1
AND	2	3
OR	3	2
EX-OR	4	5
EX-NOR	5	4

- **Duality:-**

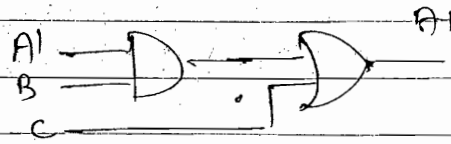
operator $B = \{\cdot, +\}$, Digit $B = \{0, 1\}$

Interchange :- (1) $(\cdot, +)$
(2) $(0, 1)$

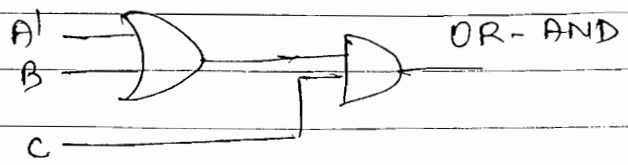
Step 1 :- Interchange the operator $(\cdot, +)$
Interchange the Identity $(0, 1)$

AND	OR
$x \cdot x = x$	$x + x = x$
$x \cdot 0 = 0$	$x + 1 = 1$
$x \cdot 1 = x$	$x + 0 = x$
$x \cdot \bar{x} = 0$	$x + \bar{x} = 1$

Ex:- $f = (A' \cdot B) + C$



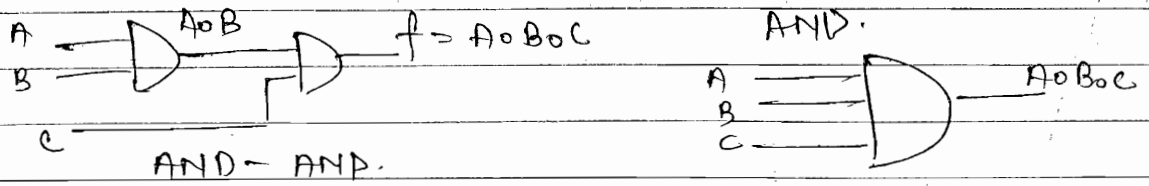
$f^D = (A' + B) \cdot C$



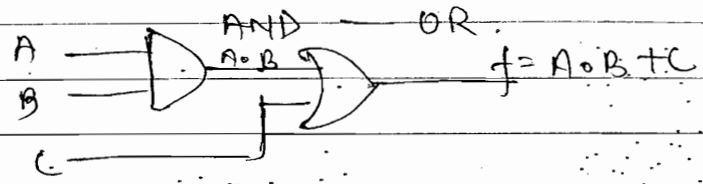
• De-generative form: (two stage only)

If a two-level logic gate sys. o/p can be expressed with a single logic gate. then the two level logic gate sys. is known as de-generative form for the single logic gate.

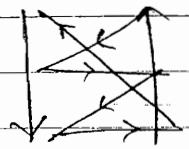
ex:- AND-AND is de-generated form. for the AND gate.



• NOR - De- generative form:-



Shortcut

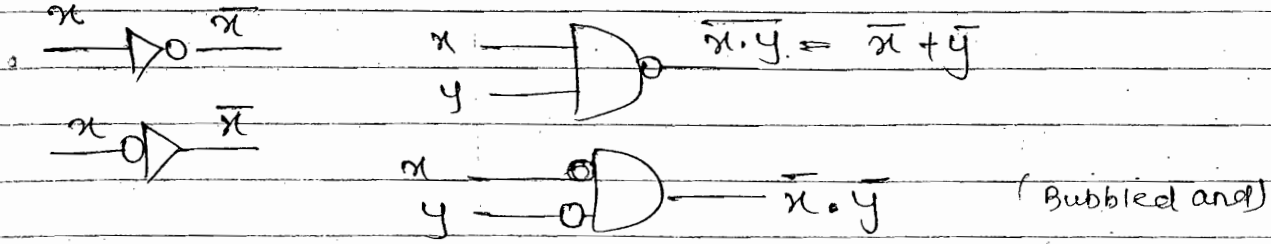


AND — OR — dual — OR — AND
NAND — NAND — NOR — NOR
NOR — OR — NAND — AND
OR — NAND — AND — NOR

→ Combinations of the same horizontal line are dual forms

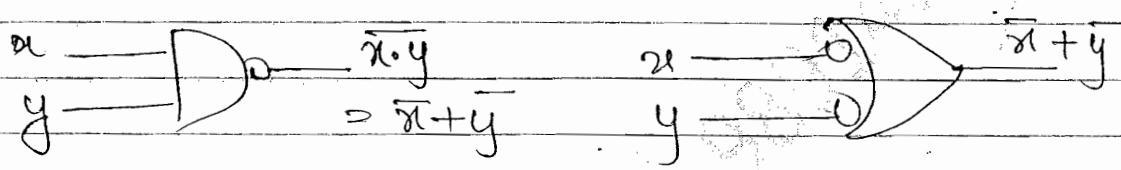
ex:- AND - OR → OR - AND.

NOTE:- Inversion Before Binary operation is not same as that of inversion after the Binary operation



But it can valid for uniary operation.

- Alternative logic Gate:-

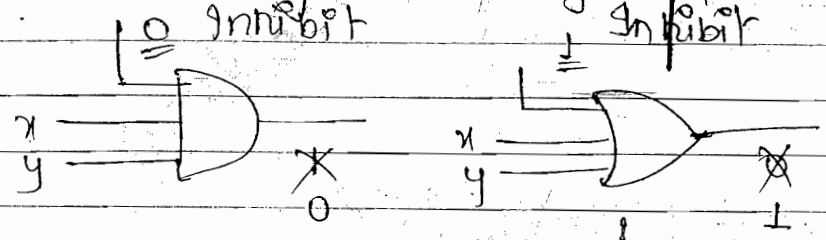


Shortcut

Bubbled

NAND	—	OR
NOR	"	AND
AND	"	NOR
OR	"	NAND

- Inhibiting the logic Gates:-



low → 0
High → 1

Short cut

Inhibit	Disable	Enable
NAND	low	High
NOR	high	low
AND	low	High
OR	High	low

- **Positive & Negative Logic:-**

+ve logic	-ve logic	+ve.
High → 1	How → 1	ex:- −2V → 1
How → 0	High → 0	−7V → 0

(+ve) AND logic

A	B	f = A·B
0	0	0
0	1	0
1	0	0
1	1	1

(−ve) AND = (+ve) OR

A	B	f = A+B
1	1	1
1	0	1
0	1	1
0	0	0

→ −ve AND equal to +ve OR & vice-versa
→ Same case for NAND and NOR.
→ −ve NAND = +ve NOR & vice-versa

- **Complimenting the Boolean Expression:-**

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

Step - (1) Dual form
Step - (2) compliment Individual variable.

→ $f(1) = (A' + B) \cdot C$
$(2) = (A + B') \cdot C' \equiv f'$

→ ORDER TO SOLVE BOOLEAN EXPRESSION:-
(1) [ ] Bracket
(2) NOT
(3) AND
(4) OR

- **Ex-OR Gate Specalister:-** **EX-NOR:-**

$x \oplus x$	$=$	0		$x \odot x$	$=$	1	
$x \oplus \bar{x}$	$=$	1		$x \odot \bar{x}$	$=$	0	
$x \oplus 1$	$=$	$\bar{x}$		$x \odot 1$	$=$	x	
$x \oplus 0$	$=$	x		$x \odot 0$	$=$	$\bar{x}$	

EX-NOR is even function for even No. of I/p variables. [EX-OR = EX-NOR]

$$\overline{x \oplus y} = x \odot y$$

$$x \oplus y \oplus z = x \odot y \odot z$$

$$\overline{x \oplus y \oplus z \oplus w} = x \odot y \odot w \odot z$$

EX-NOR is odd function for odd No. of I/p variables [EX-OR = EX-NOR]

- $\overline{x \oplus y}$, $\bar{x} \oplus y$, $x \oplus \bar{y}$ $\Rightarrow x \odot y$

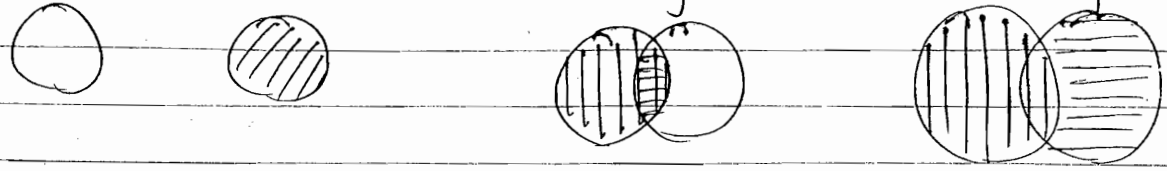
ex:- $\bar{x} \oplus y = (\bar{x})y + \bar{x}\bar{y}$

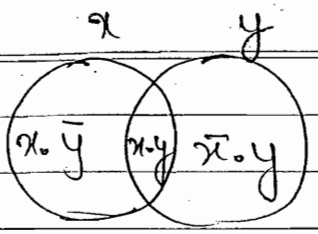
$= xy + \bar{x}\bar{y}$

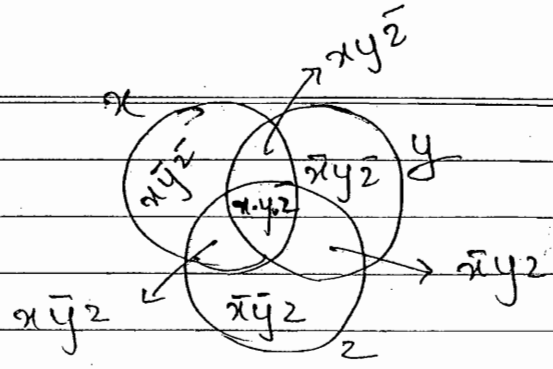
$= x \odot y$

- **Venn Diagram:-**

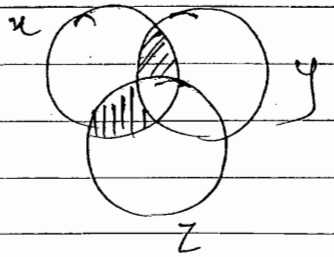
$x = 0$ $x = 1$ $x + xy = x$ $x + \bar{x}y = x + y$



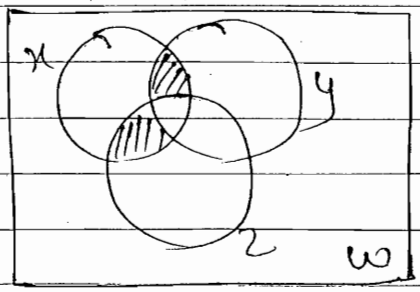




ex:



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



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

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

19/9/2014

- **Logic Minimization Technique:**

NOT if $x = 0$, $\bar{x} = 1$, $(\bar{x})' = x$

AND $x \cdot x = x$, $x \cdot 0 = 0$

$x \cdot 1 = x$, $x \cdot \bar{x} = 0$

OR $x + x = x$

$x + \bar{x} = 1$

$x + 1 = 1$

$x + 0 = x$

Commutative law

$x \cdot y = y \cdot x$

$x + y = y + x$

Associative law

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

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

Distributive law $x \cdot (y+z) = x \cdot y + x \cdot z$.

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

De morgan's law $\overline{x \cdot y} = \bar{x} + \bar{y}$

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

Consensus theorem

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

↳ Redundent (any no. of variables)

(part of group NOT effect the o/p of sys).

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

ex:- $A \cdot B + \bar{A} \cdot C + B \cdot C \cdot D \cdot E = A \cdot B + \bar{A} \cdot C$

∵ B, C are complesary variable, and can ↑ some other variable upto any no. (A, B ...).

→ A variable is associated with variable, its compliment associated with other variable, Next term formed by the left over variables, the term is called redundent

→ Consensus theorem can be extended to any no. of variables.

Transposition Theorem :-

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

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

Operators and association can be interchange.

Absorption law: - ORing a variable with ANDing of that variable with other variable results in the same variable

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

Redundent literal Rule (RLR):

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

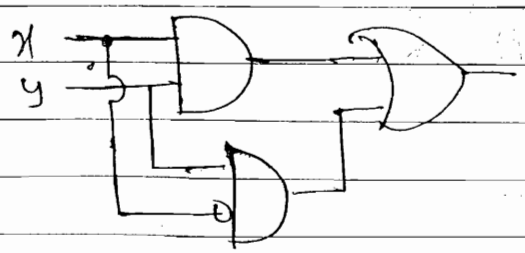
ORing a variable with ANDing of its compliment with another variable results in the ORing of these two variables.

- **SOP and POS form :-**

SOP (Minterm) (concentrate in 1

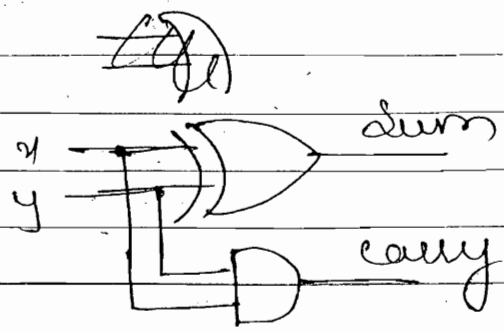
x	y	Minterm	f
0	0	$\bar{x} \cdot \bar{y}$	0
0	1	$\bar{x} \cdot y$	1
1	0	$x \cdot \bar{y}$	0
1	1	$x \cdot y$	1

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



x	y	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

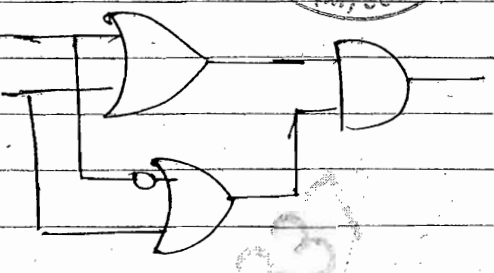
$$\text{Sum} = \bar{x} \cdot y + x \cdot \bar{y}$$
$$= x \oplus y$$



$$\text{carry} = x \cdot y$$

POS (Maxterm) Concentrate in zero.

x	y	Maxterm.	f	= (x+y)·(x̄+y)
0	0	x+y	0	
0	1	x+ȳ	1	
1	0	x̄+y	0	
1	1	x̄+ȳ	1	



- If 0 are less in o/p table concentrate in 0 (POS) and make circuit, If 1 are less concentrate in 1 (SOP) and make circuit.

- $\Sigma m \Rightarrow$ SOP $\qquad \Pi M \Rightarrow$ POS

ex:- (1) $\Sigma m(0,2) = \Pi M(1,3)$ too 0/p.

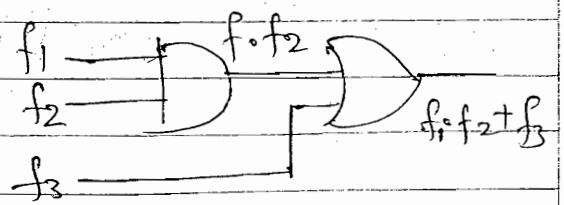
(2) $\Sigma m(0,2) = \bar{x}\cdot\bar{y} + x\cdot\bar{y}$ (00, 10)

(3) $\Pi M(1,5) = \Sigma m(0,2,3,4,6,7)$ (001, 101)

(4) $\Pi M(0,2) = (x+y)(\bar{x}+y)$ (00, 10)

(5) $\Sigma m(1,5,9) = \Pi M(0,2,3,4,6,7,8,10,11,12,13,14,15)$

Q $f_1 = \Sigma m(0,1,2,4,6)$
$f_2 = \Sigma m(2,3,4,6,7)$
$f_3 = \Sigma m(3,4,5,6)$



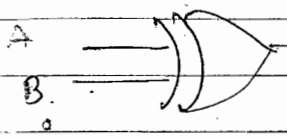
Solution $f_1 \cdot f_2 \rightarrow$ Present in both
$= \Sigma m(2,4,6)$

$f_1 \cdot f_2 + f_3 \Rightarrow \Sigma m(2,4,6) + \Sigma m(3,4,5,6)$
$= \Sigma m(2,3,4,5,6)$

(8421) → code to convert binary NO.

ex:- 0101 = 5
0010 = 2

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



$f = A \oplus B$

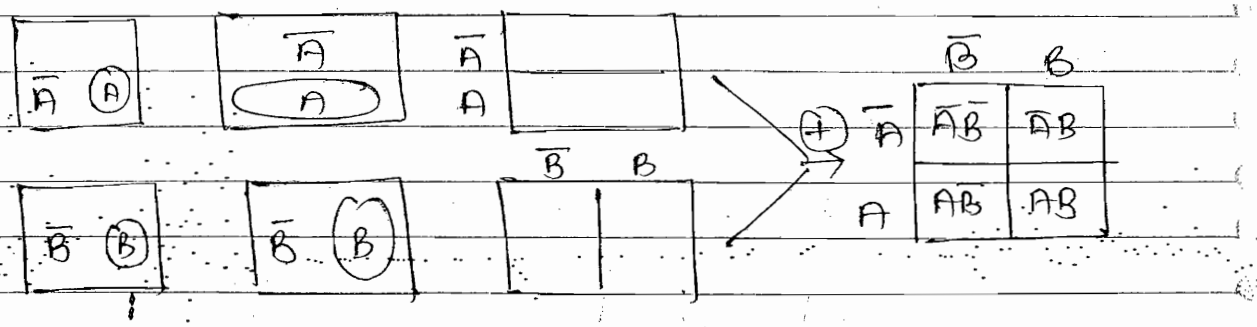
Solution $A \oplus B = \bar{A} \cdot B + \bar{B} \cdot A$

$\bar{A} \cdot B = \Sigma m(3,5,7)$, $\bar{B} \cdot A = \Sigma m(0,2,4)$

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

$= \Sigma m(0,2,3,4,5,7)$.

- **K-MAP :-** It is a modification of Venn diagrams is a group of adjacent cell, Each cell is represented by minterm. Minterm is a group of literal.



(1)

	$\bar{B}$	B
$\bar{A}$	$\bar{A}\bar{B}$ 0 0 0	$\bar{A}B$ 0 1 1
A	$A\bar{B}$ 1 0 2	AB 1 1 3

for 2 variables

Must be one literal change in K-Map in neighbour cell).

(2)

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

for 3 variables

$f = (A\ B\ C)$

MSB LSB

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

DO Not write like $C\bar{A}B$ Because C is LSB always comes last

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

$f = (C\ A\ B)$ — LSB, MSB

लो लो

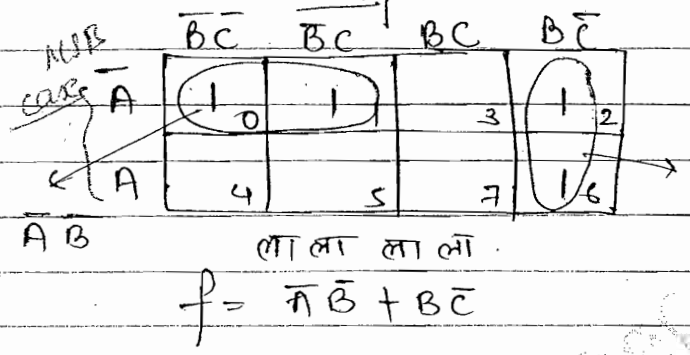
Q $f = \Sigma m(0, 1, 2, 6)$ → 000, 001, 010, 110

$f = (ABC)$ — MSB, LSB

By default.

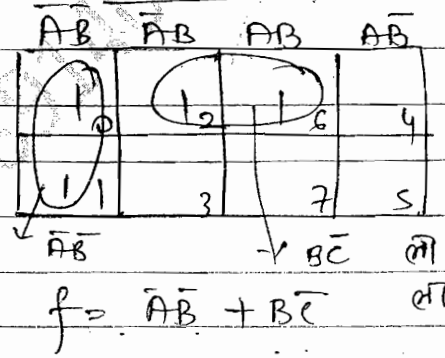
$f = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C}$

Solution KMAP



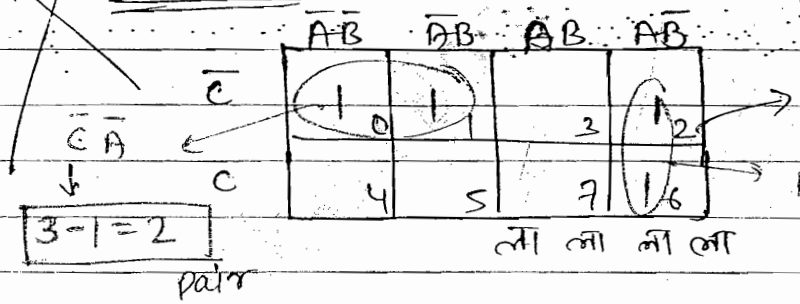
$f = \bar{A}\bar{B} + B\bar{C}$

K-MAP



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

Mistake (what we do)!!

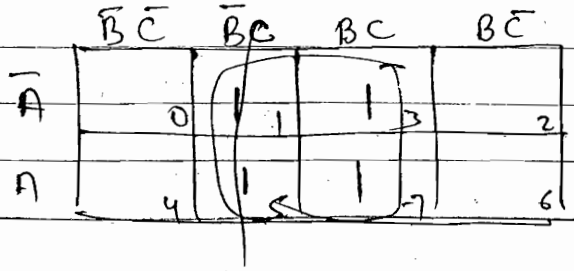


No. are wrong written

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

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

$f = (ABC)$ — MSB, LSB.

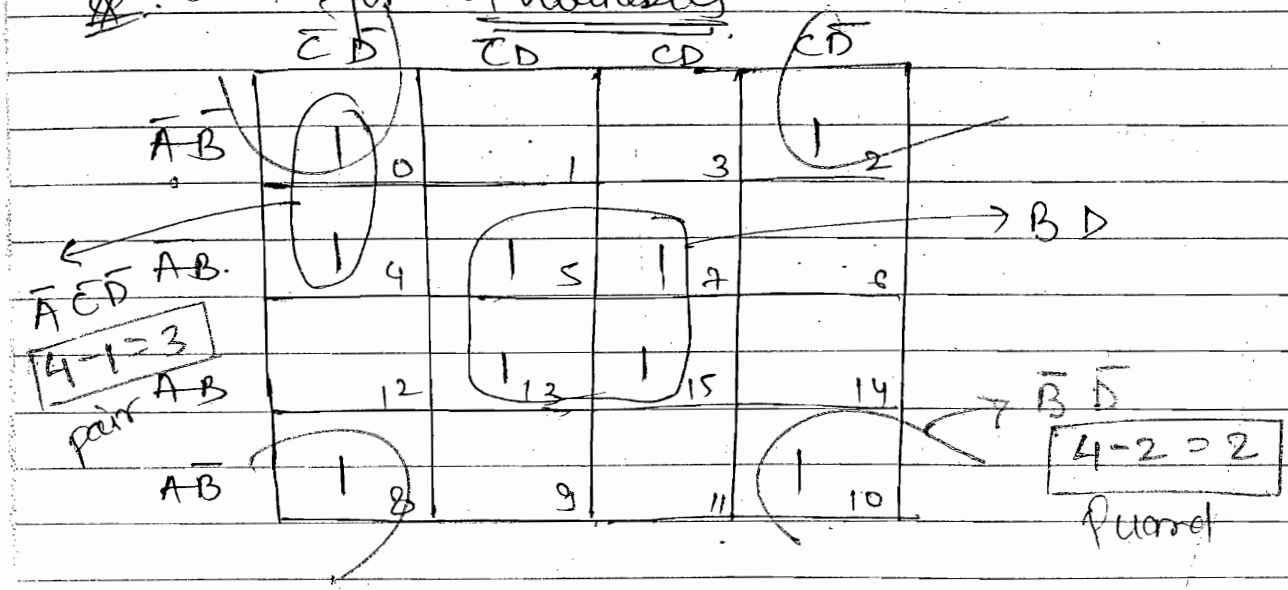


$f = C$

3-2=1

Quad

for 4 variables



Shortcut Pair → 1 Subtract

Quad → 2 Subtract

Octet → 3 Subtract.

① Variable − Shortcut = O/p in No. of literal

Ex:- No. of I/p variable = 5

Octet pair = 3

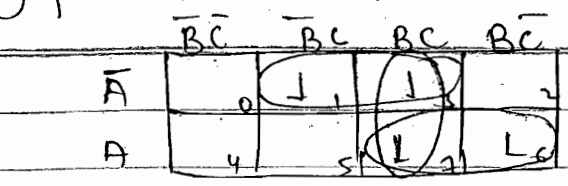
O/p variable = 5 − 3 = 2

- Redundant Group:- It is that part of circuit which doesnot effect the circuit result.

→ While solving the K-Map we should follow the technique of highes order grouping to lower order grouping.

Ex

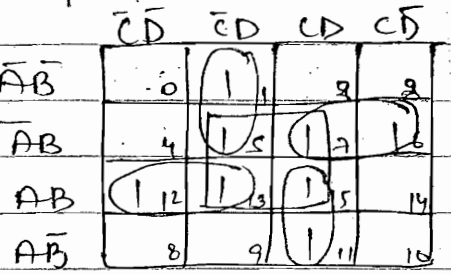
① $f = \Sigma m(1,3,6,7)$



P.I.

$f = \underbrace{\bar{A}C + AB}_{\text{E.P.I}} + (BC)$ ↓ Redundent

② $f = \Sigma m(1,5,6,7,11,12,13,15)$.



E.P.I.

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

$+ (BD)$ ↓ Redundent.

R.P.I. ← leave quard.

: If (15) → 1 is not present then we have to consider quard. (It will give less # literal)

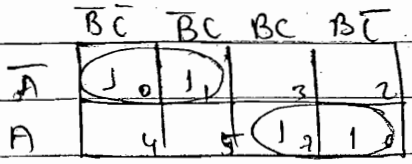
- Must consider all the 1's in group.

20/9/2014

- Implicant :- It is minterm corresponding to that cell which is having 1 in the K-map.
- Prime Implicant :- All possible grouping of K-maps known as prime implicant.
- Non Prime Implicant :- This are those implicant which are not able to get possible grouping.
- Essential Prime Implicant :- E.P.I is prime implicant (P.I) only which is having atleast a single 1 which is only one time group. (uniquely grouped)
- Selective Prime Implicant :- These are prime implicant which are under the process of selection.

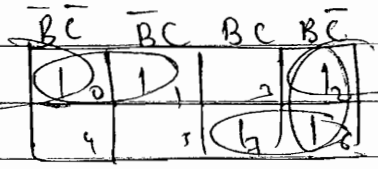
all E.P.I are P.I. But reverse is Not ~~false~~ True

- **Redundent P.I:** It is the P.I. which does effect the circuit result.

Ex:- 

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

E.P.I. also P.I — E.P.I. also P.I

Ex:- 

$f = Em(0,1,4,6,7)$

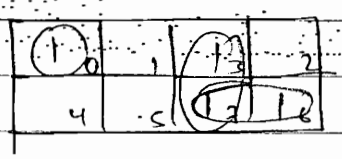
$f =$ $\bar{A}\bar{B}$ (0,1) — E.P.I ✓

$\bar{A}\bar{C}$ (0,2) — P.I. } S.P.I (selectively prime implicant)

$B\bar{C}$ (2,6) — P.I. }

AB (6,7) — E.P.I ✓

I.M.P → 5 (Implicant), P.I → 4, E.P.I → 2, Non P.I → 0,

ex- 

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

Implicant = 4, Non Prime Implicant → 1

Prime Implicant - 2, E.P.I = 2

⇒ **DON'T CARE Conditions:** - The o/p corresponding to the unspecified I/p is known as don't care condition.

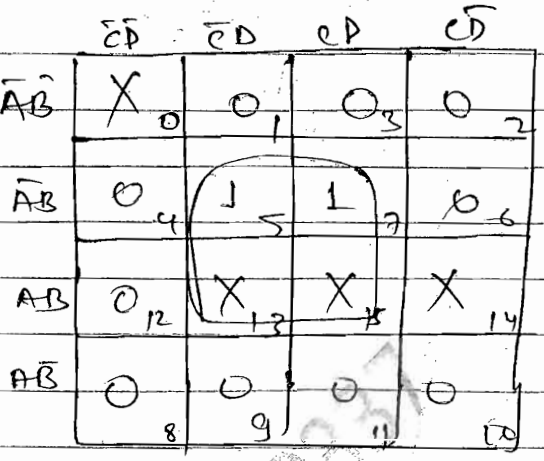
Q Design a ckt which give alarm at 5 and 7 o'clock.

Solve four $/p ABCD

0000 → 0 → X
Never arises $/p.

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

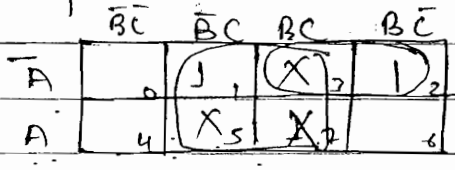
9 → 0
10 → 0
11 → 0
12 → 0
~~11 01~~ 13 → X
1110 14 → X
1111 15 → X
Not arises in $/p



$\bar{A}\bar{B}D \rightarrow BD$

X → Blocks are empty so we have a chance to take don't care.

Ex: - $f = \Sigma m(1,2) + \phi(3,5,7)$

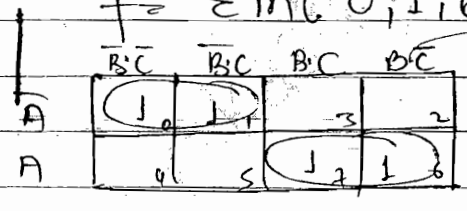


$f = \bar{A}B + C$

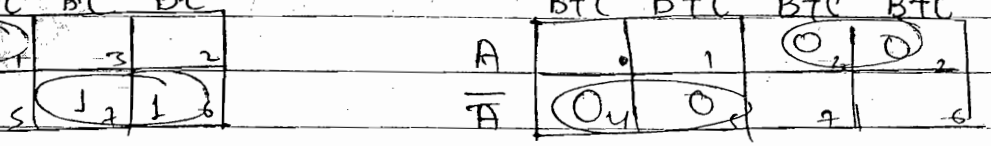
• POS forms In K-Maps

(conversion of SOP into POS)

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



$f = \pi M(2,3,4,5)$



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

$f = (A+\bar{B}).(\bar{A}+B)$ ← (1)

$\boxed{POS = \overline{SOD}}$

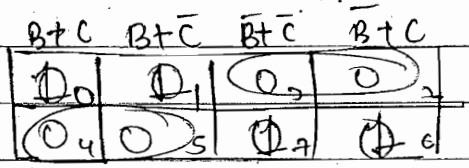
(1) Dual form ⇒ $(\bar{A}+\bar{B}).(A+B)$

(2) Compliment variable ⇒

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

Not making with eqn (1) But drawing in K-Map Identity must be also change.

(1) change operator
(2) complement Individual variable.



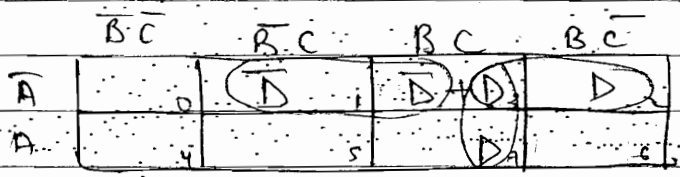
$f = (\bar{A}+B).(A+\bar{B})$

→ Blue colour 0 convert into red colour 1 and remaining are 0 (2,3,4,5), pair the 0's both POS form.

- Variable Entering Method :- (VEM)

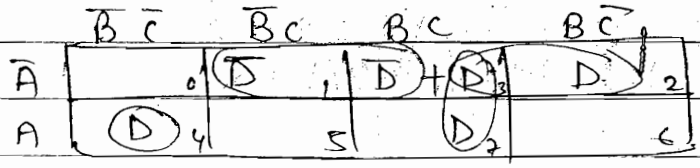
4→9|P → Using 3 Variable K-Map.

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



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

If $f = \bar{A}B\bar{C}\bar{D} + \bar{A}BCD + \bar{A}BC\bar{D} + \bar{A}B\bar{C}D + ABCD + AB\bar{C}D$



$f = \bar{A}C\bar{D} + \bar{A}BD + BCD + AB\bar{C}D$

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

(2) $f = BC$ Ans

Conventional Variable Entering Method :—

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0 (0)	A (1)	1 (3)	$\bar{A}$ (2)
x	B (4)	0 (5)	X (7)	C (6)

Step 1 :- Keep all the variables as zero and get the simplified ~~and~~ expression.

NOTE :- Check weather the given 1 in K-map is fully covered or Not ~~Reason~~ If it fully covered then it becomes redundent

Step 2 :- Select any variable and keep 1 in its position and other variables should be kept as 0. and given 1 replaced by don't care (X).

Step 3 :- Repeat the above step By selecting each variable.

Step 1

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

Real colour → change
Black colour → same

$f_1 = yz$ Redundent

for A

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

$f_2 = A\bar{x}z$

for $\bar{A}$

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

$f_3 = \bar{A}\bar{x}y$

for B.

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

$f_4 = B x \bar{y} \bar{z}$

for C.

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

$f_5 = C x y$

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

→ Redundent.

If A → cell is not present then consider the redundent 2n and. It will not Redundent anymore.

Shortcut

cell is fully covered.

	$\bar{y}\bar{z}$	$\bar{y}z$	yz	$y\bar{z}$
$\bar{x}$	0	$\bar{A}$	$A+\bar{A}$	$\bar{A}$
x	B	0	X	C

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

Q

1st Method

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

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

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

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

2nd Method

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

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

$\bar{A}B$

Not fully covered.

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

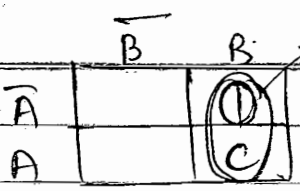
CB

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

$A, \bar{A}, 1$ with X can be grouped together (guard)
don't care

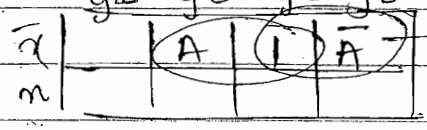
make guard

3rd Method



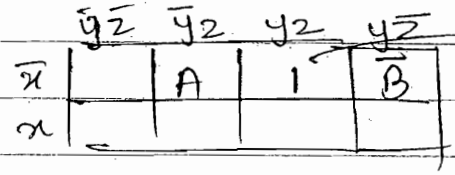
$f = \bar{A}B + BC$

- Cell having 1 must have neighbour cell of variable and its compliment for fully covered.



→ fully covered.

If →



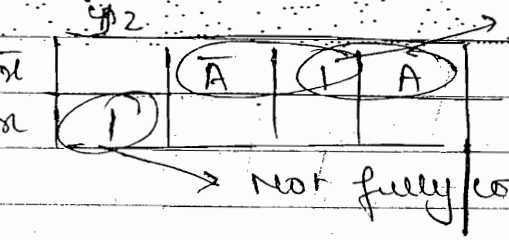
Then Not fully covered.

→ pairing of variable with don't care and 1 only.

(A Ā) ✓ (A B) ✗ (A X) ✓ (A 1) ✓

→ All 1's in K-Map must check → fully covered or not.

Ex:-



→ this 1 is fully covered

→ Not fully covered, → $x\bar{y}\bar{z}$

- Tabulation Method OR Mc Clusky:-

Step 1:- Get the circuit expression in the form of SOP.

Step 2:- Make the different groups depending on the number of 1's in the minterms.

Step 3:- Compare the successive groups until we get the different group and get the simplified answer.

→ In don't Care Condition :- Consider them. but remove at table.

⇒ Solving the don't Care terms in Tabulation Method Consider the don't care no. along with given no. and get the final minterms but don't this no. in E.P.I. Table.

$f = \Sigma m(0, 1, 3, 7, 8, 9, 11, 14, 15)$

Group	Minterm	Variable A B C D
0	0	0 0 0 0
1	1	0 0 0 1
	8	1 0 0 0
2	3	0 0 1 1
	9	1 0 0 1
3	7	0 1 1 1
	11	1 0 1 1
	14	1 1 1 0
4	15	1 1 1 1

Implicant Table

Group	Minterms	Variable A B C D
0	0, 1	0 0 0 –
	0, 8	– 0 0 0
1	1, 3	0 0 – 1
	1, 9	– 0 0 1
	8, 9	1 0 0 –
2	3, 7	0 – 1 1
	3, 11	– 0 1 1
	9, 11	1 0 – 1

Pairs Table

3	7, 15	- 1 1 1	
	11, 15	1 - 1 1	
	14, 15	1 1 1 -	ABC

Group	Minterm	Variable	
0	0, 1, 8, 9	- 0 0 -	
	0, 8, 1, 9	- 0 0 -	$\bar{B}\bar{C}$
1	1, 3, 9, 11	- 0 - 1	
	1, 9, 3, 11	- 0 - 0	$\bar{B}D$
2	3, 7, 11, 15	- - 1 1	
	3, 11, 7, 15	- - 1 1	CD

Quads Table

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

→ Redundent

	0	1	3	7	8	9	11	14	15
E.P.I $\bar{B}\bar{C}$	(X)	X			(X)	X			
R.P.I $\bar{B}D$		X	X			X	X		
E.P.I CD			X	(X)			X		(X)
E.P.I ABC							(X)		X

single in column.

No need to circle prime this one already having circle in this row

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

ABC

Redundent ($\bar{B}D$)

→ Row having cross become E.P.I.

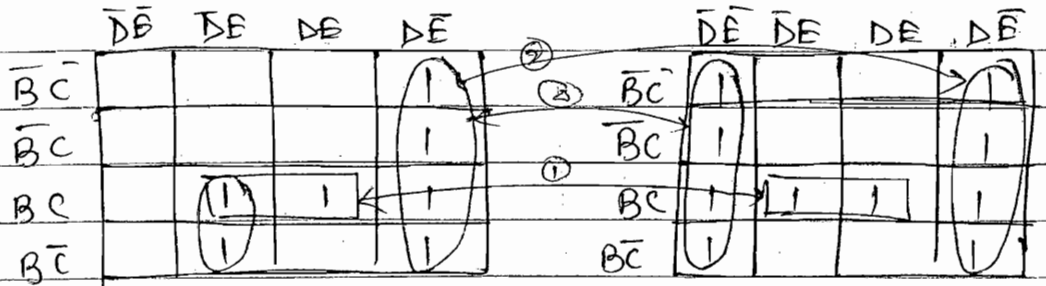
→ Suppose column 11 don't have two cross only $\bar{B}D$ row having cross then $\bar{B}D$ Become P.I, Not R.P.I.

→ Circle only those X which are single in column

21/9/2014

5-Variable K-Map

- For grouping only one literal change required.



1. $\bar{A}BCE + ABCE = BCE$ (Grouped together)
2. $\bar{A}D\bar{E} + AD\bar{E} = D\bar{E}$ (" ")
3. $\bar{A}D\bar{E} + A\bar{D}\bar{E}$ = Two literal change. NO grouped together [Grand node → two]
4. $BCE + \bar{A}B\bar{D}E$

- Standard SOP (SSOP):- Must having all literals. If any literal is missing it called SOP.

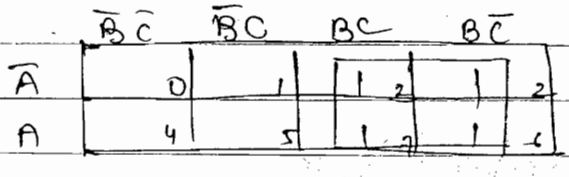
SOP to SSOP:-

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

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

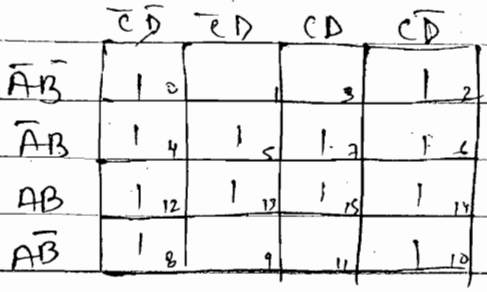
$= \bar{A}BC + ABC + AB\bar{C} + AB\bar{C} + \bar{A}B\bar{C}$

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



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

Ex:-



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

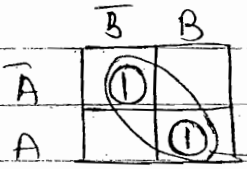
Diagonal Group (EX-NOR, EX-OR)

- Not the circuit simplified Technique → Only to write the answer quickly. (save time)
- Same Nature → EX-NOR $(\bar{A}\bar{B})$ [Both bar, Both without Bar]
- Different Nature → EX-OR $(\bar{A}, B)$ [one bar, other without Bar]

(making grouping in diagonal).

ex:-

(1)

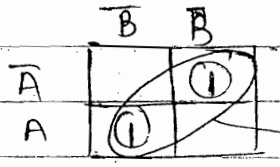


$\bar{A}\bar{B} + AB$.

→ direct → $A \odot B$ [EX-NOR]

(same Nature)

(2)

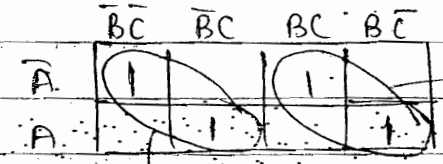


$\bar{A}B + A\bar{B}$

→ $A \oplus B$ [EX-OR]

(different Nature)

(3)



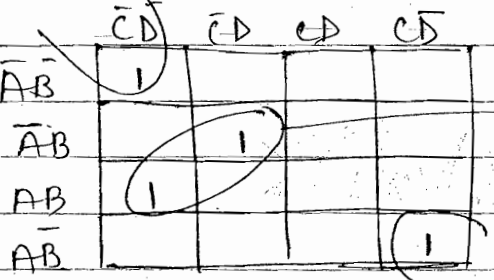
→ B constant, A & C different Nature

$B[A \oplus C]$

B constant, $(\bar{A}, \bar{C})$, (A, C) same Nature

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

(4)



→ B, $\bar{C}$ constant, $(\bar{A}D)$, $(A\bar{D})$ different Nature

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

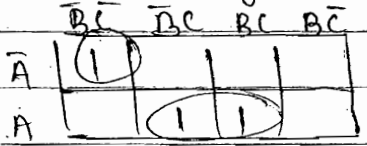
→ $\bar{B}\bar{D}$ constant $(\bar{A}\bar{C}, A, C)$ same Nature

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

⇒ Only apply when given in option :-

- Only apply when simplified is not possible. [only present in diagonal].

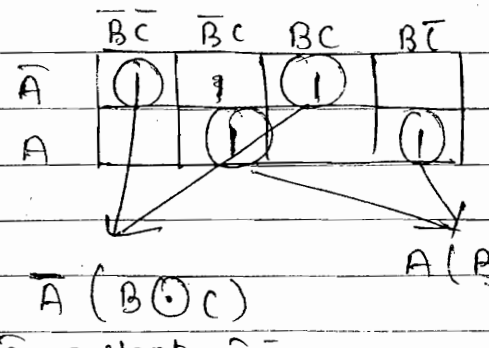
⇒ If



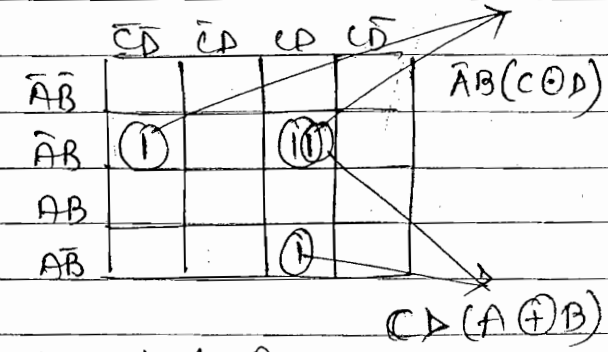
taking this pairing
$AC + \bar{A}B\bar{C}$

- Other Method for simplifying:-

OFFSET IN K-MAP:-



$\bar{A}$ constant, $\bar{B}\bar{C}$, BC → same Nature ∴ Ex-NOR



A constant, $\bar{B}C$, $B\bar{C}$ → different Nature ∴ Ex-OR

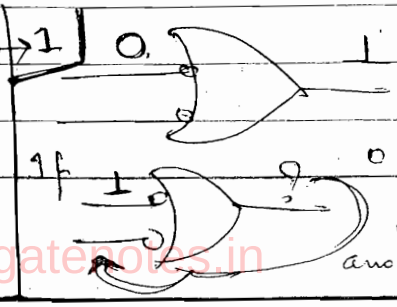
- Sequential Circuit:-

present O/p depend upon not only present I/p but also past O/p.

Latch :- O/p store the value (0 or 1) until the I/p change ∴ called latch.

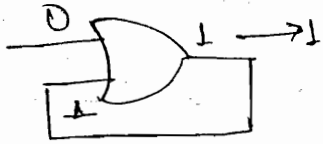
Active low:- Input ($S = 0$), affect the O/p.

→ If I/p is 0 → O/p is away → 1



S R
0 0 → Don't care.

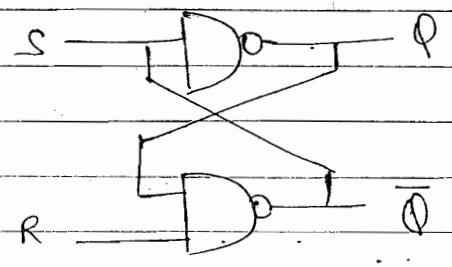
If $S \to 0$ $Q \to 1$



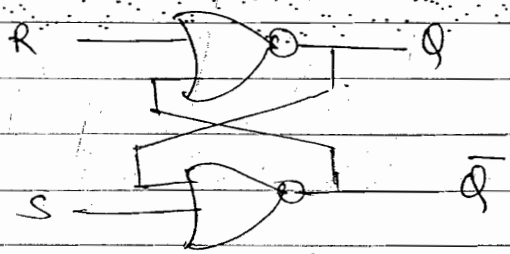
NAND — low
NOR → high

- $R \to 0$, $\bar{Q} \to 1$, $Q \to 0$ So Reset, $S \to 1$
- $R \to 1$, $\bar{Q} \to 0$, $Q \to 1$ So Set, $S \to 0$.
- Apply o/p again-again to get stable o/p (Microscopic method).
- S-R. → NAND Gate → Active low
  → NOR Gate → Active High.
- Logic gate with feedback connection known are sequential circuits. ex:- Latches.

Latches are two type: ① NAND gate latch (active-low)

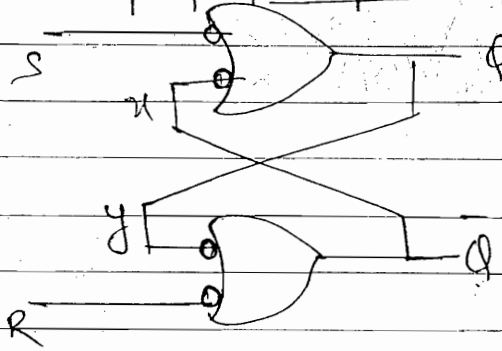


② NOR gate latch: (active-high)



S, R, position change to get complete diff. table from SR(NAND) above,

- S-R Flip-flop operation (Active-low)



Case 1:- S=1, R=1

✓ Initially if Q=0
When the feedback o/p will be x=1, y=0, Then we get the next stage $Q^+=0$

$Q^+ = 0$.

✓ Initially if $Q = 1$:- Then the feedback I/p will be $x = 0$, $y = 1$. Then then the next stage will be $Q^+ = 1$

So By observation we can say that we $S = 1$, $R = 1$ we get Q^+ = No change.

Case 2 :- $S = 1$, $R = 0$

✓ Initially $Q = 0$:- Then the feedback I/p will be $x = 1$, $y = 0$ so we get the next stage $Q^+ = 0$

✓ Initially $Q = 1$:- Then the feedback I/p will be $x = 0$, $y = 1$ so we get the next stage $Q = \bar{Q} = 1$ only. which is an intermediate stage. It is not the stable stage. So that the internal feedback is operate until we get stable O/p.

By the observation we can say that when $S = 1$, $R = 0$ we get $Q^+ = 0$ "Reset"

case 3 :- $S = 0$, $R = 1$

The same above processor is conducted for initially $Q = 0$ and for initially $Q = 1$ By.

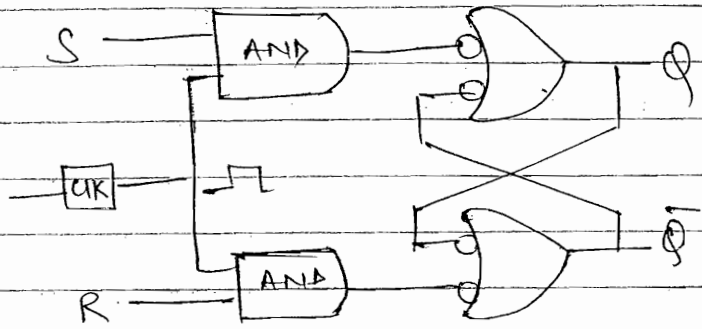
By observation we can say when $S = 0$, $R = 1$ we get $Q^+ = 1$ "Set".

case 4 :- When $S = 0$, $R = 0$

By the observation we got $Q^+ = \bar{Q}^+ = 1$ which is unuseful condition known as don't care condition.

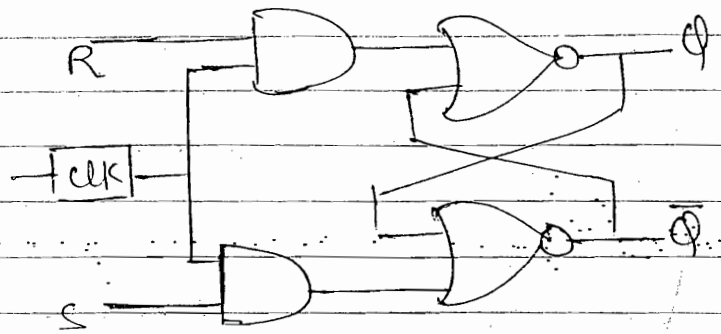
NOTE:- ① The S-R latch can be controlled with the help of a switch known as gated SR latch

② The controlling can also be done by using clock generator known as clocked SR flip-flop.



Active low Table

CLK	S	R	Q^+
⊓	1	1	No change
⊓	1	0	0 Reset
⊓	0	1	1 Set
⊓	0	0	x Don't care



Active high Table

CLK	S	R	Q^+
⊓	0	0	No change
⊓	1	0	1 Set
⊓	0	1	0 Reset
⊓	1	1	x Don't care

- Difference b/w flip-flop and latch is clock generator
  No CLK → latch. controlling done → flip-flop.

- I/p given, changed internally By NOT gate and O/p comes → Active high (Mean. Low change into high I/p. in Active low → O/p comes → equal to Active high (internally changed)

→ Modified SR → JK

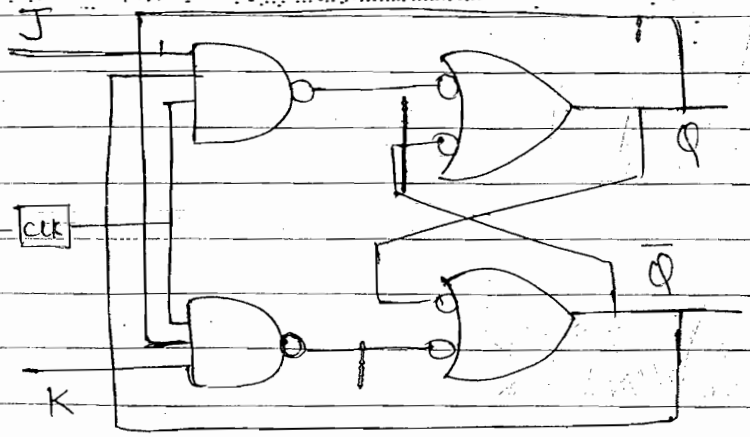
- **J-K Flip-Flop :-** The modification of SR flip-flop with external feedback connection is known as J-K flip-flop.

When J=1, K=1, and clock pulse is applied By observation we can say that- $Q^+ = \bar{Q}$ **toggle** condition.

NOTE: ① Internal feedback should be operated until we get the stable O/p. This operation doesn't depend on clock pulse.

② External feedback operated only one time for the clock pulse.

If [output $[Q, \bar{Q}]$ applied to feedback to I/p, it will Not effect the I/p because clock is Now OFF, it will wait until the Next clockpulse will applied]



Table

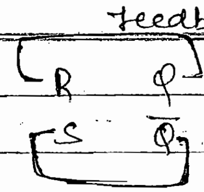
clk	J	K	Q^+
⎍	0	0	No change
⎍	1	0	1 Set
⎍	0	1	0 Reset
⎍	1	1	$\bar{Q}$ Toggle

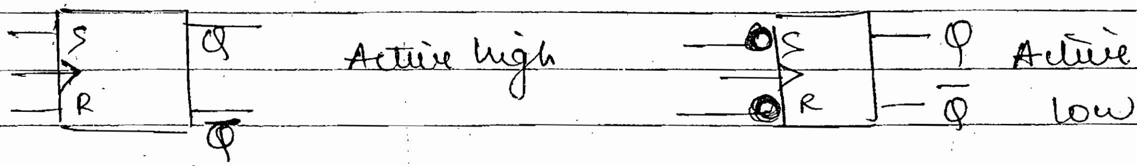
S R ⟶ 99% ⟶ High Active high (By default)

S	R	J	K	
0	0	0	0	N.C
0	1	0	1	set
1	0	1	0	Reset
1	1 → X	1	1	Toggle.

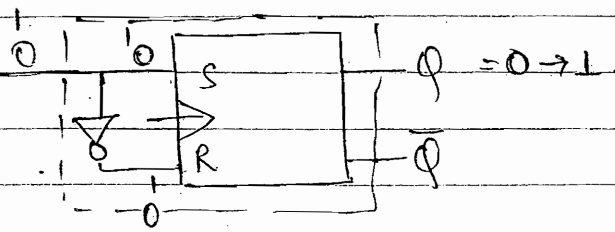
C active high

→ To make $\frac{R}{S}$ $\frac{\bar{Q}}{Q}$ J.K





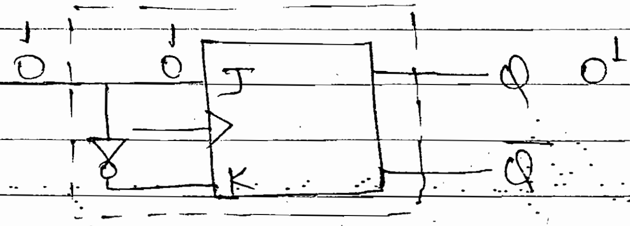
- <u>D - Flip Flop and T - Flip Flop:-</u>



Table

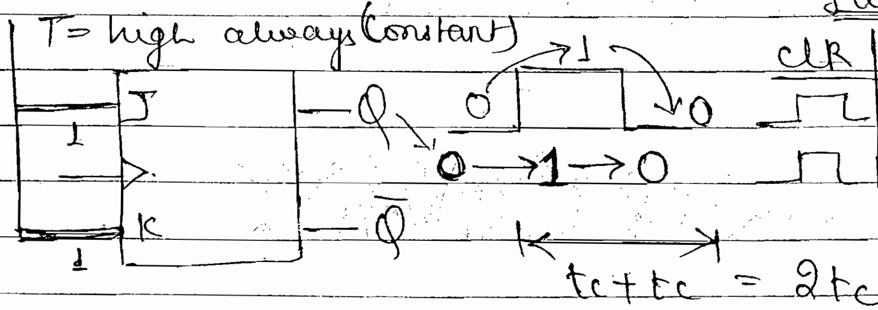
CLK	D	Q^+
⎍	0	0
⎍	1	1

→ By J-K flip-flop. → Not use this one, Because of external feedback connectⁿ, size ↑, But asked in exam.



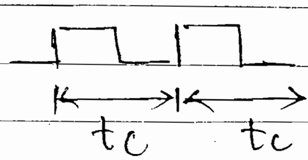
- <u>T - flip-flop</u> (Toggle)

T = high always (Constant)



Table

CLK	T	Q^+
⎍	0	No change
⎍	1	Toggle.



I/p x Hz.   O/p $\frac{x}{2}$ Hz

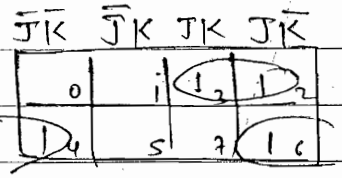
→ I/p is always set at high, for 1 clock pulse it will toggle 0 → 1 and for 2nd clock pulse it will toggle 1 → 0

→ 2 I/p pulse, 1 O/p pulse with 1/2 freq.

22/9/2014.

- **Characteristic Equations of Flip-Flop:**

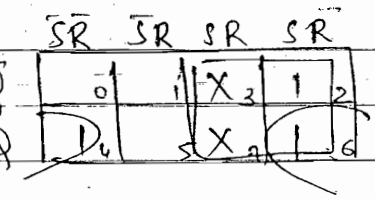
Q	J	K	Q^+	
0	0	0	0	N.C
0	0	1	0	Reset
0	1	0	1	Set
0	1	1	1	Toggle $\bar{Q}$
1	0	0	1	N.C
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	0	Toggle $\bar{Q}$



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

- **Characteristic Equations of S-R Flip Flop:-**

Q	S	R	Q^+	
0	0	0	0	N.C
0	0	1	0	Reset
0	1	0	1	Set
0	1	1	X	Don't care
1	0	0	1	N.C.
1	0	1	0	Reset
1	1	0	1	Set
1	1	1	X	Don't care



$$Q^+ = S + \bar{R}Q$$

- **Characteristic eq^n D - Flip Flop :-**

Q	D	Q^+
0	0	0
0	1	1
1	0	0
1	1	1

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

$$Q^+ = D$$

- **Characteristic eq^n of T - Flip Flop**

Q	T	Q^+	
0	0	0	NoC
0	1	1	Toggle
1	0	1	NoC
1	1	0	Toggle

	$\bar{T}$	T
$\bar{Q}$		1
Q	1	

$$Q^+ = \bar{Q}T + Q\bar{T}$$

$$Q^+ = Q \oplus T$$

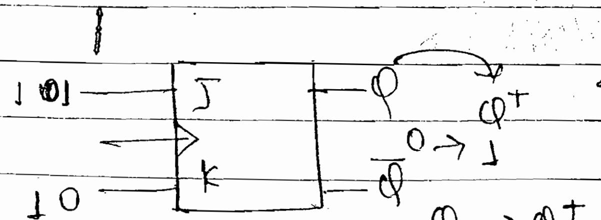
- **Excitation Table:** To get particular O/p with possible ways of I/p.
  → consist of possible ways of I/p to get particular O/p

For JK flip-flop.

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

For SR flip flop

Q	Q^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0



$Q \to Q^+$

J	K		J K	
1	0		0	0
1	1		1	0
fixed	X		X	0

For D flip flop.

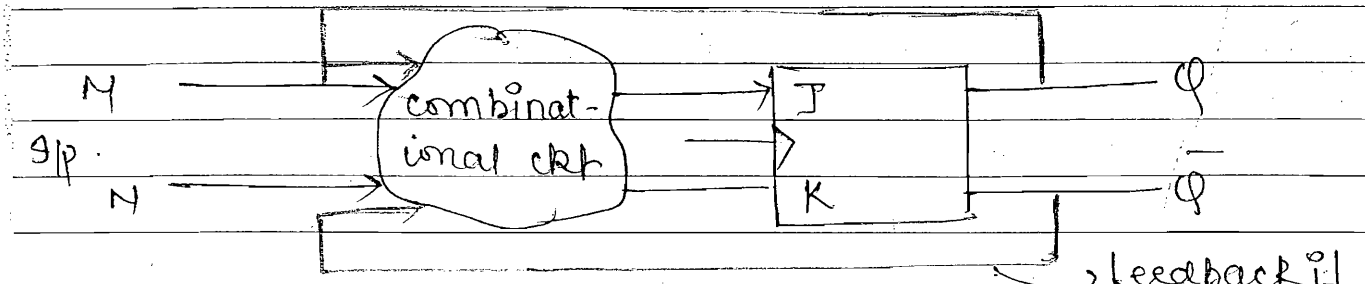
Q	Q^+	D
0	0	0
0	1	1
1	0	0
1	1	1

Truth Table :- O/p with desire o/p.

For T flip flop.

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	0

- Designing of flip flop :- (By using JK)



o/p Q	M	N	client Requirement Q^+	J	K
0	0	0	1	1	X
0	0	1	0	0	X
0	1	0	1	1	X
0	1	1	1	1	X
1	0	0	X	X	X
1	0	1	0	X	1
1	1	0	1	X	0
1	1	1	X	X	X

- J, K → o/p are written by excitation table results.
- If we don't fill from excitation table but X, X ex. this line
- Then draw K-map of J, K

Excitation Table of JK

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

For J

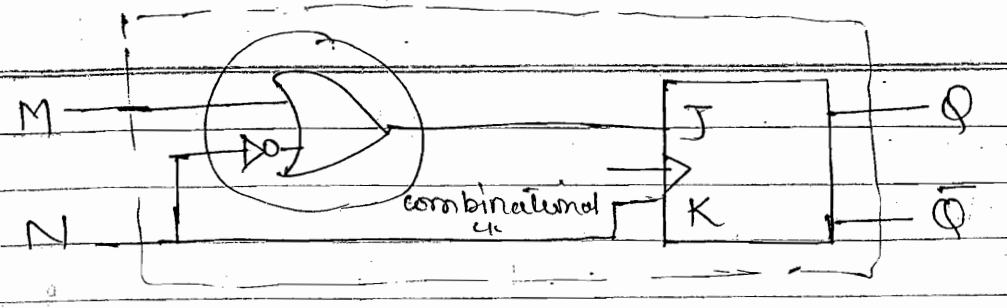
Q	$\bar{M}\bar{N}$	$\bar{M}N$	MN	$M\bar{N}$
$\bar{Q}$	1		1	1
Q	X	X	X	X

$J = M + N'$

For K

Q	$\bar{M}\bar{N}$	$\bar{M}N$	MN	$M\bar{N}$
$\bar{Q}$	X	X	X	X
Q	X	1	X	

$K = N$



- **Designing of flip-flop using SR :-**

Truth Table of JK → Client Requirement

Q	A→J	B→K	Q^+	S	R
0	0	0	0	0	X
0	0	1	0	0	X
0	1	0	1	1	0
0	1	1	1	1	0
1	0	0	1	X	0
1	0	1	0	0	1
1	1	0	1	X	0
1	1	1	0	0	1

SR excitation Table

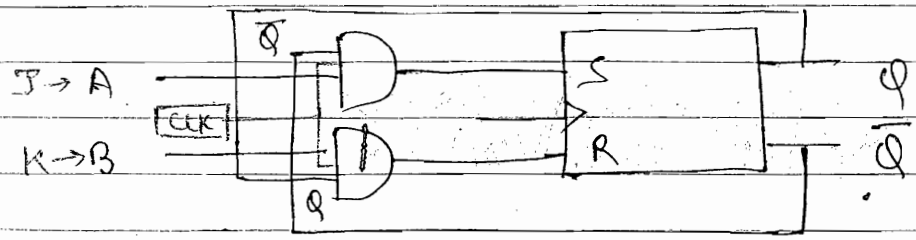
Q	Q^+	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

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

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

$S = \bar{Q}A$

$R = QB$



S-R flip-flop convert into JK flip-flop = CLR is externally applied.

Requirement → Truth Table
By using → Excitat^n Table.

→ All excitation Table are for → Active High
→ This JK flip-flop and previous JK are same. because previously having NAND gate SR → Active low converted into high Active. But in this AND are used Because SR → is already Active high (excitat^n Table).

Q Design ~~Ex~~ D-F.F By using S-R:-

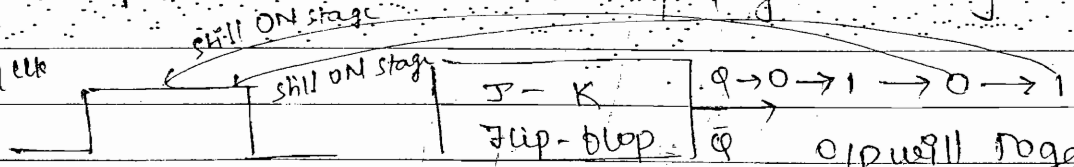
Ans

Delay → Truth Table
S-R → Excitation Table.

Q How?

① S-R ⇒ J-K, T, D
② D ⇒ S-R, J-K, T
③ J-K ⇒ S-R, T, D
④ T ⇒ J-K, S-R, D

• clk pulse width ↑ ⟶ propagation delay ↓

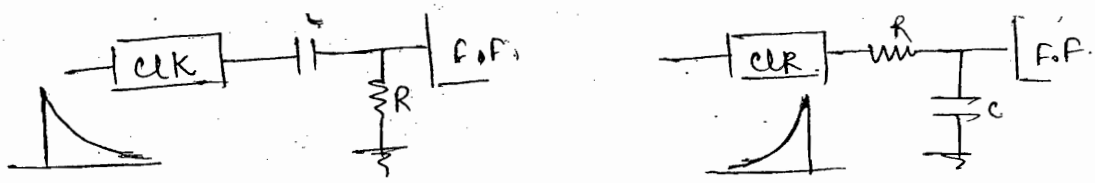


o/p will toggle (repetition occur in single pulse)

Because after changing o/p from initially 0→1 clk is still ON stage so again change 1→0, ⟶ 0→1,

⇒ problem called Race Around problem.

⇒ To Avoid is ① Reduce pulse width, OR
② Use filter (RC).

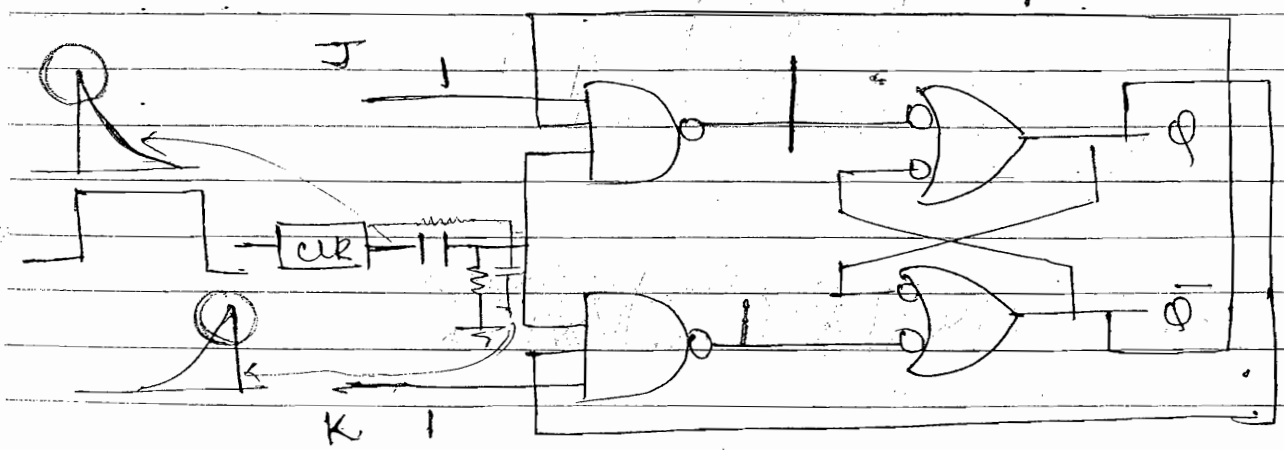


- **Race Around Problem:-** When J=1, K=1 and clock pulse is applied. Then we get the toggle condition. But when the o/p is occuring at the instant of time when the i/p pulse is in ON state then their is repetition of Toggle at the output for the single pulse of i/p known as race around problem.

- **Methods to avoid Race Around Problem:-**
  (i) Reducing the pulse width.
  (ii) By using the edge triggering.
  (iii) By using MASTER SLAVE J-K Flip-Flop.

NOTE Edge triggering two types:-
(i) POSITIVE EDGE TRIGGERING (leading edge triggering) (rising edge)

(ii) NEGATIVE EDGE TRIGGERING (trailing edge triggering) (falling edge)



- Green portion having sufficient magnitude to trigger remaining NOT having —u— —u— to —u—

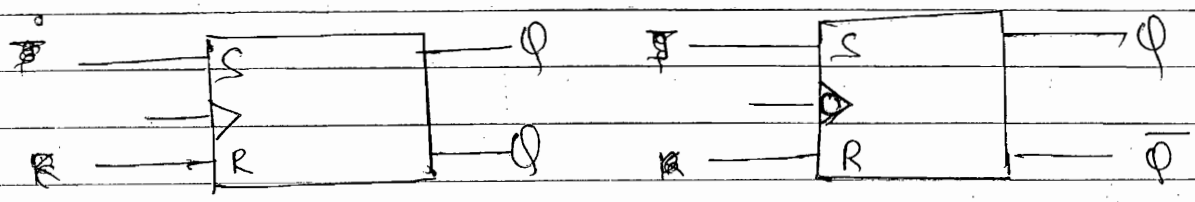
So. clk → 0

edge triggering level triggering or pulse

+ve −ve

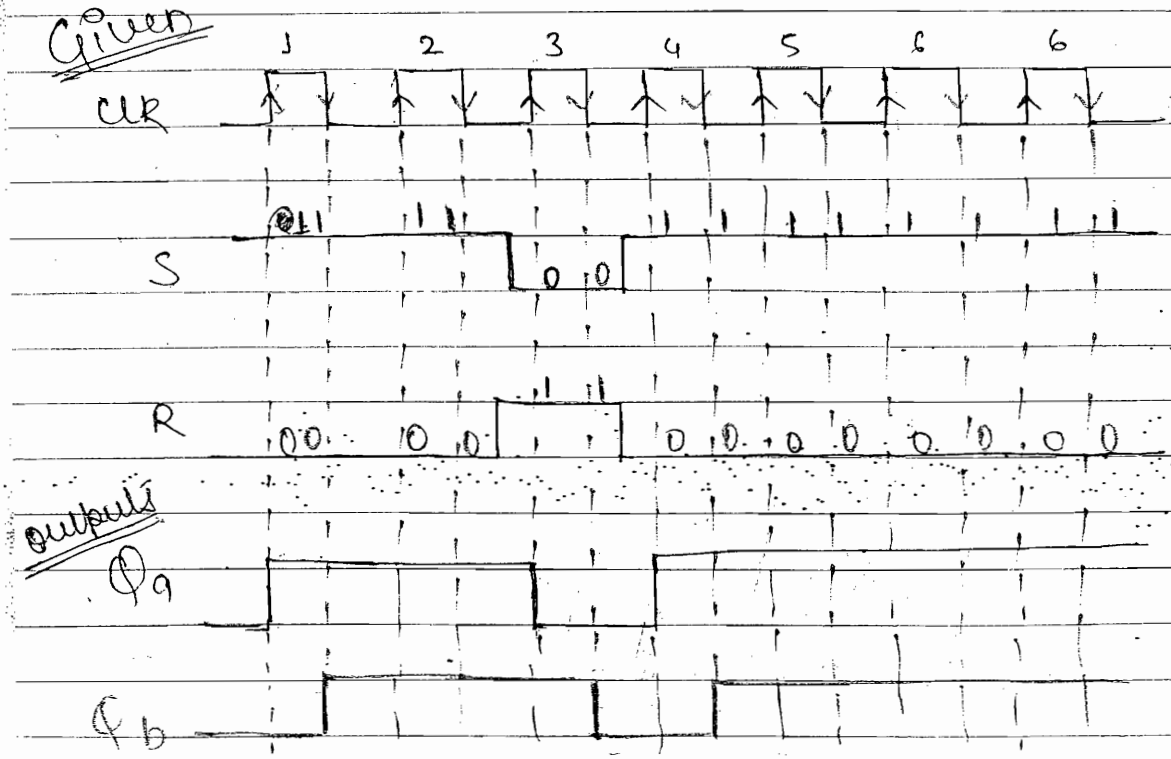
- Timing Diagram :-

Edge Triggering



−ve edge triggering. −ve E.T.

Given

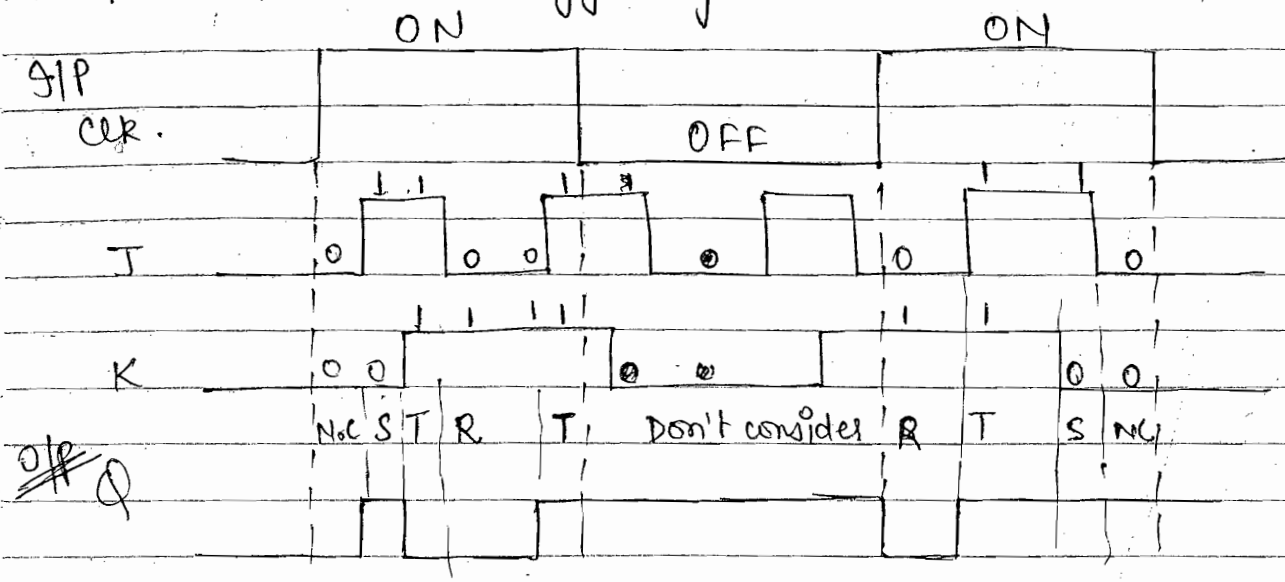


→ By using the result of S-R flip-flop, 0 0 → NC, 0 1 → Reset, 1 0 → Set and 1 1 → X we draw the timing o/p.

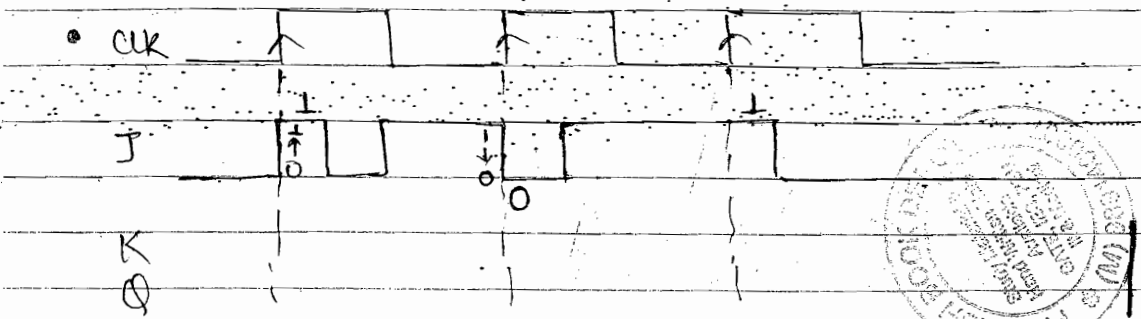
→ Befor drawing, first write the S, R state

- • → +ve triggering
- • → −ve triggering

Pulse OR level Triggering :-



→ OFF level triggering period don't consider, the ON state remain continous upto Next ON stage, In toggle period stage will chang (0→1, 1→0).



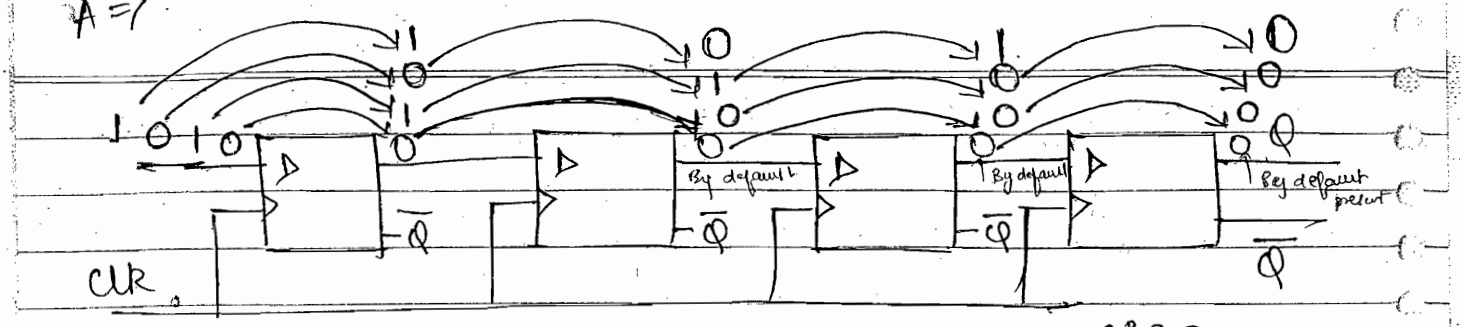
- Parallel in Binary Shift Register :-

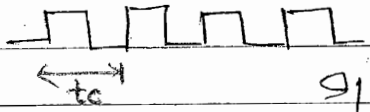
Flip flop connected in cascade manner is known as Binary Shift Register.

- Types

(1) Serial In Serial Out (SISO)
(2) Serial In parallel Out (SIPO)
(3) Parallel In Serial Out (PISO)
(4) Parallel In Parallel Out (PIPO)

A ⇒ 1 0 1 0





		I/p clk.	O/p clk.
(1)	SI SO	n	(n-1)
(2)	SI PO	n	0
(3)	PI SO	1	(n-1)
(4)	PI PO	1	0

- No. of I/p Bit ⇒ No. of clock pulse required to store these Bit in SI. and only one clk pulse to input the bit in F.F. (Data is already present at O/p. SO (n-1) Bit to come out data for SO and 0 for PO

- Application of Binary Shift Register:
(1) They are used for temperary data storage.

(2) These will provide the time delay.

Time Delay $\Delta t = N_0 T_c$ N → No. of clk pulse

$= N_0 \frac{1}{f_c}$

(3) These are used for data conversion ex:- SIPO convert serial form of data into parallel form and PISO convert parallel data into serial form.

→ left Shift ⇒ Multiplication by 2.

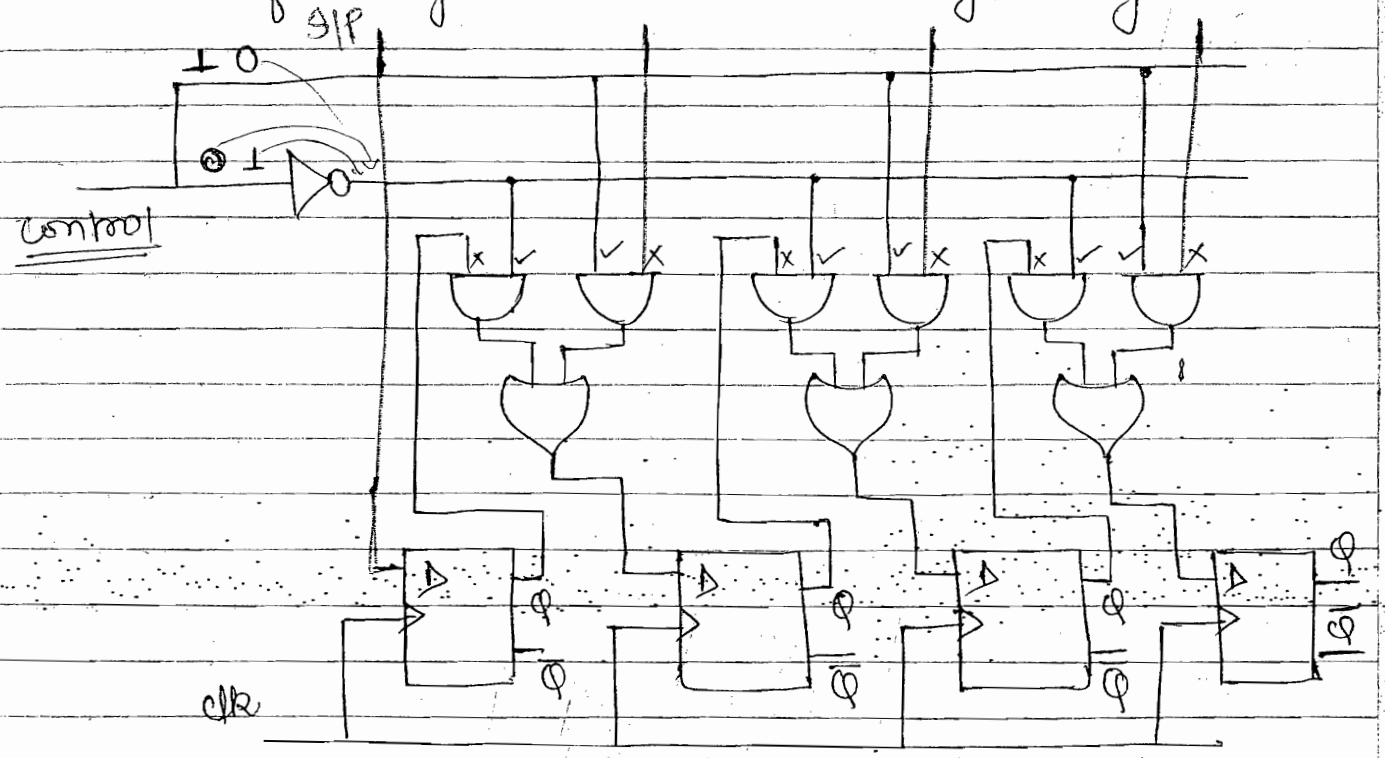
(4) These are used for arithmetic operation.

Ex:- left shift in register is multiplication by 2

Right shift register is → Division by 2

NOTE:- Right shift Register having error of 0.5 for odd Numbers

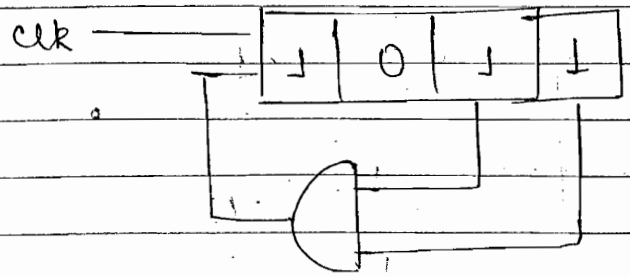
(5) Shift Registers are used to design Ring Counters.



- When control pulse is 0 → SISO
- When control pulse is 1 → 0 → PISO

To make Serial o/p o/p. control will change to 0.

Q After 3 pulse what is O/p?



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

1011 → O/p.

23/9/2014

- Special type of Syn. Counter → Ring counter.
- No. of possible state → Mod value.
  ex: 8 — → Mod 8

$n = 2^n$ → Mod

n: No. of flip flop used

ex: Mod 10 → No. of flip flop used → 5

- Timing diagram → Take column and draw in row.

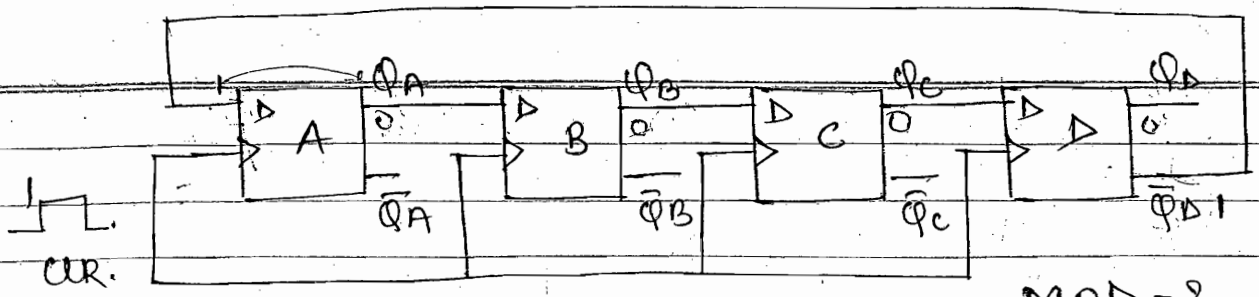
Counters: It count the no. of clk pulses applied to it. Counters are of two type.

(i) Synchronous Counter (Parallel Counter)
(ii) Asynchronous Counter (Ripple Counter, Wave)

→ Ring Counters are special category of Syn. counters. Categories in two ways.

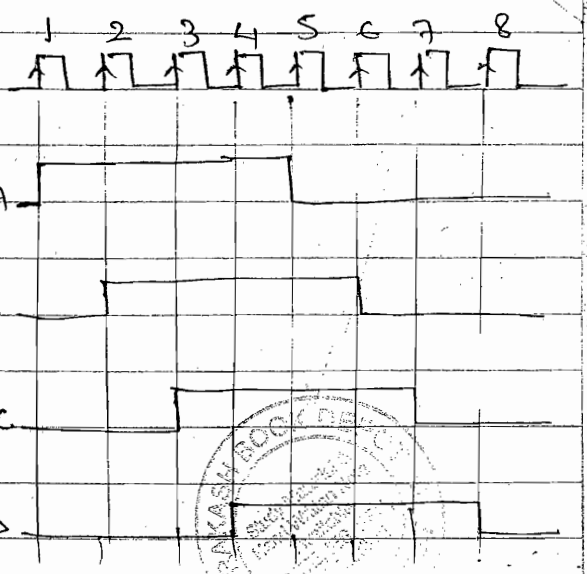
(i) Johnson's Ring Counter or (Twisted Ring Counter)
" Feedback is given from complimentary O/p of last flip flop.

Ex. MOD-10 ⟶ > flip flop.



MOD-8

CLK		Q_A	Q_B	Q_C	Q_D	
1	8	1	0	0	0	0→8
2	12	1	1	0	0	0→12 Q_A
3	14	1	1	1	0	0→14
4	15	1	1	1	1	1→15 Q_B
5	2	0	1	1	1	1→7
6	3	0	0	1	1	1→3 Q_C
7	1	0	0	0	1	1→1
8	0	0	0	0	0	0→0 Q_D
9		1	0	0	0	



Repeated. Not used.

formula - $n \rightarrow 2^n$ → possible states; → CLK pulses

no. of F.F. ; Mod value

- Initially Q_A, Q_B, Q_C, Q_D are at 0 states. When applied clk pulse. $Q_A = 1$, and its Binary shifted shown in table.

Q. What are the used and un-used stages of 4-bit Jonshon's Ring counter and what will be the Next stages for un-used state and check wheather this self corrected counter or NOT.

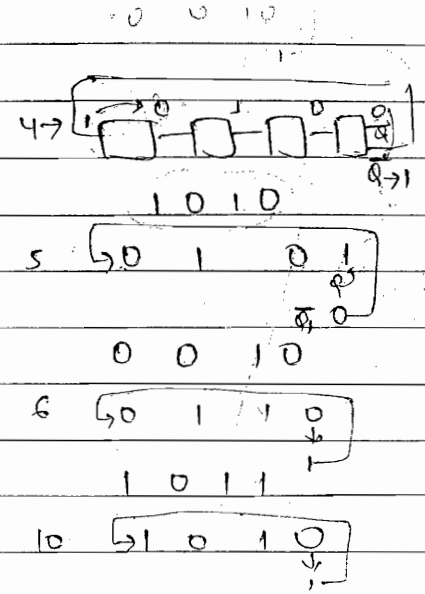
Ans. possible state are $2^n \rightarrow 16$ but getting 8 Because of such connection. So used other counter.

comes due to voltage fluctuation

used states :- → 8 → 12 → 14 → 15 → 7 → 3 → 1 → 0

un-used states :- 2, 4, 5, 6, 9, 10, 11, 13.

un used states		Next stage
2	→	9
4	→	10
5	→	2 ✓
6	→	11
10	→	13
11	→	5 ✓
13	→	6
9	→	4



- If goes to un-used states like 11, Next state 11 → 5, 5 → 2, 2 → 9, 9 → 4, 4 → 10, 10 → 13, 13 → 6, 6 → 11, 11 → 5 ...
  Repeated, Locked

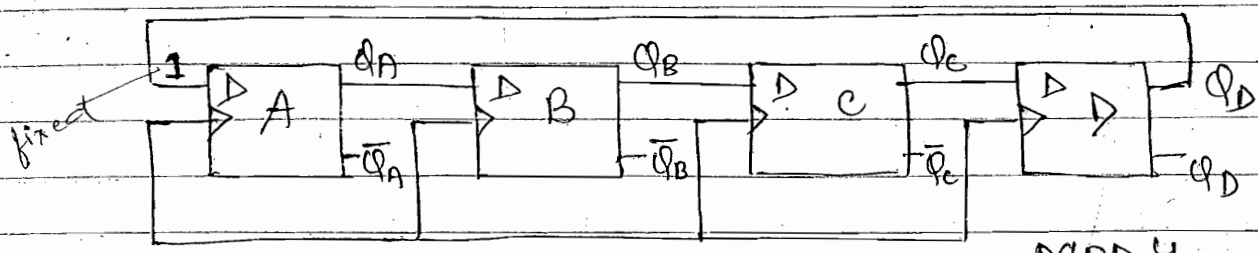
- If we checked Next state of unused state, we can say that it is Not a self corrected counter because it is not coming to its original states after any No. of pulse. So known as "locked out condition"

  So get o/p. 0 0 0 0, to any pulse.

~~Rearrangement done → first Ref. B → 9/p → A always.~~

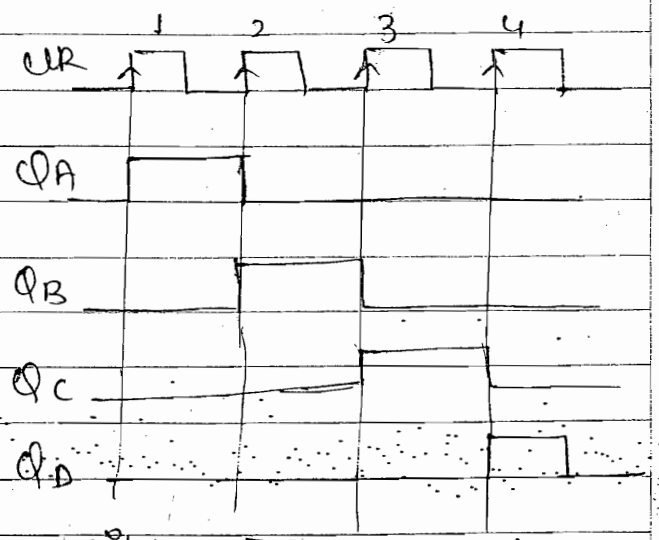
(4) Ordinary Ring Counter or Ring Counter :-

- Feedback is given from un complimentary o/p of last flip-flop.

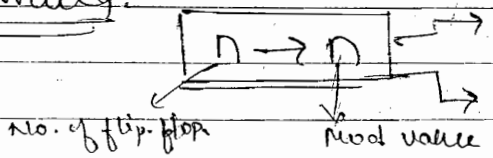


MOD 4

CLR	Q_A	Q_B	Q_C	Q_D
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	1	0	0	0



Formula:-

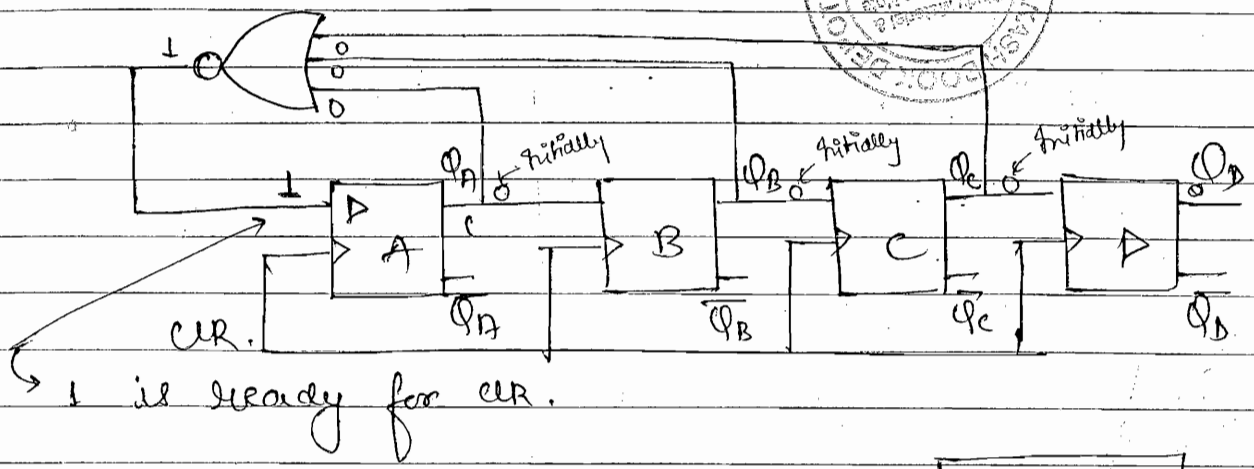


→ First F.F. set always to 1 to start it. If 0 is applied always, get 0000, NO o/p..

⟹ NOT A SELF STARTED.

- To Make it Self. Started Add. NOR Gate we get same Table. So called Self Started Ring Counter

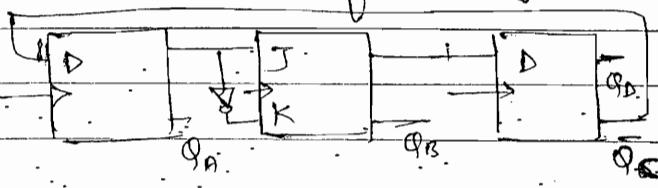
- Self started Ring Counter



1 is ready for clr.

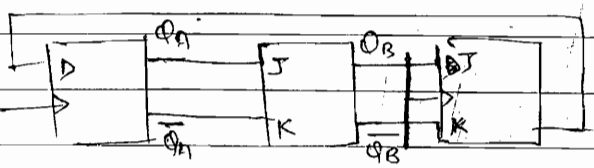
- unused state for Jhonson's ⇒ $\boxed{2^n - 2n}$
- Un-used state for Ring counter ⇒ $\boxed{2^n - n}$

Ex:

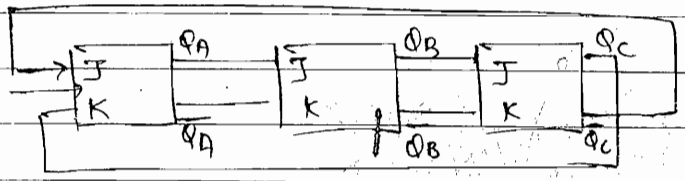


Still Jhonson's.

Other Names of Jhonson's :- ① Switch Tail
② Walking
③ Creeping.

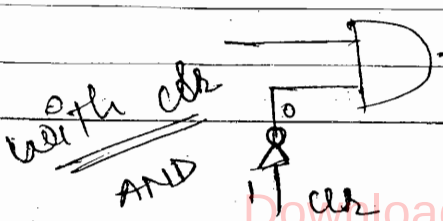


Still J.C.



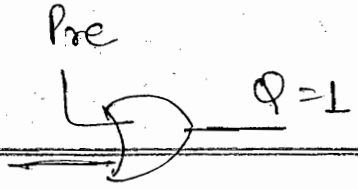
Still JoC

Asynchronous :- O/p Q → taken with comb of → clr + Q
F.F. O/P
= Q final.



when clr → active
O/p Q ⇒ 0 always.

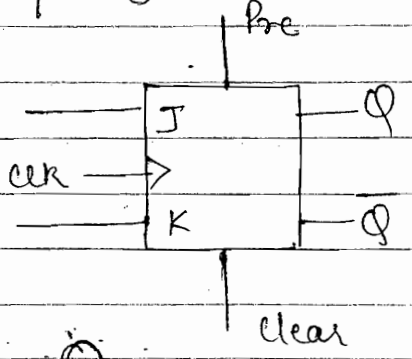
with Pre or

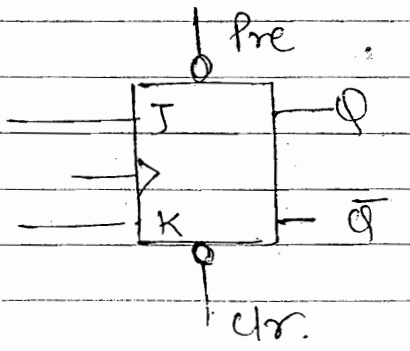


If pre is active Q=1 always.

- If clr or Pre active → J & K i/p are Not considered
- If both are 0 and 0 → J & K i/p → o/p comes in o/p.
- If Both are 1 and 1 → o/p comes → which having fast speed.

Asynchronous Counter :- (Preset AND Clear)



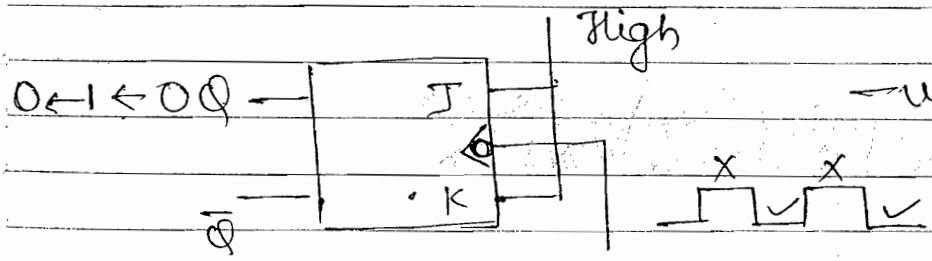


Consider (1)

Pre	clr	Q+
0	0	flip flop
0	1	0
1	0	1
1	1	X Don't care

Consider (2)

Pre	clr	Q+
1	1	f.f
1	0	0
0	1	1
0	0	X don't car



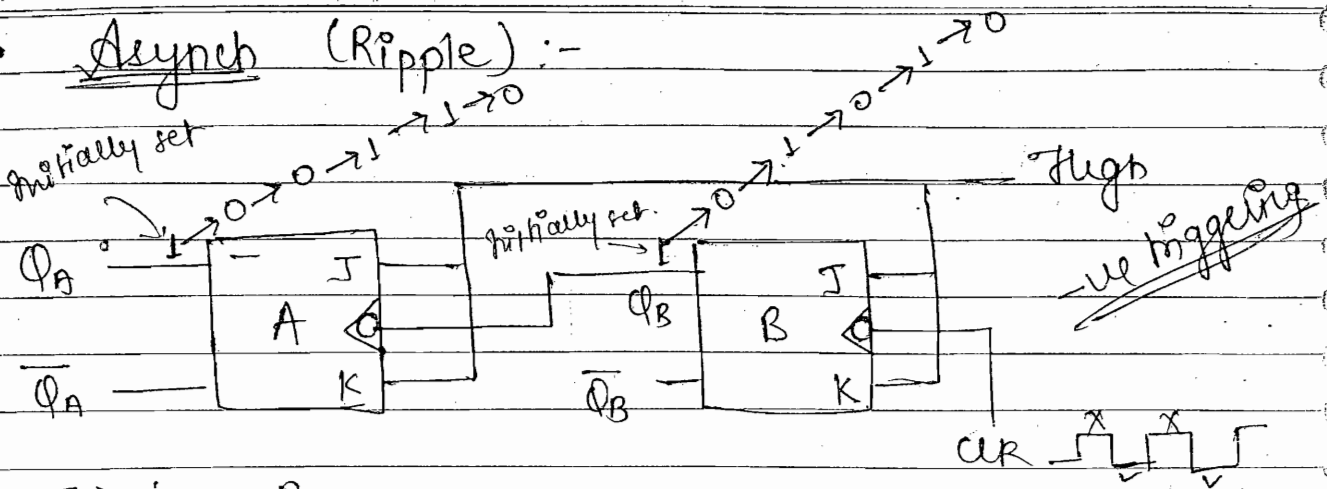
-ve triggering

⇒ Give o/p in -ve pulse. T flip flop.

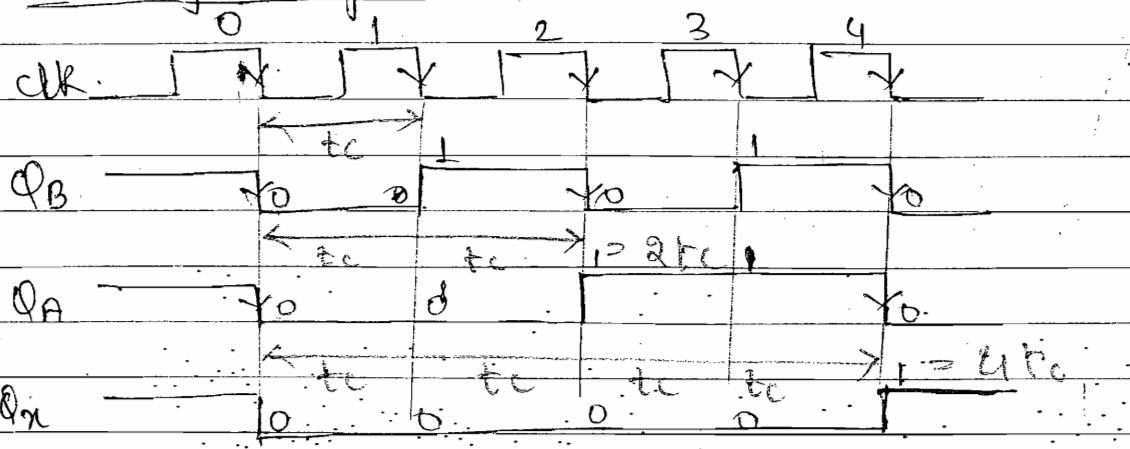
Conventional + Objective

Asynch → F.F. are NOT triggers simultaneously F.F. triggers by previous F.F. o/p.

- **Asynch (Ripple) :-**



Timing diagram



Table

CLK	Q_A	Q_B	
0	0	0	Mod 4
1	0	1	
2	1	0	
3	1	1	
4	0	0	

formula

$n \rightarrow 2^n$

n → No. of F.F.

2^n → Mod value → possible state

→ CLR.

- If their are n No. of flip-flop. first draw the timing diagram than Truth Table
- In 3 F.F. when 3 F.F. triggers 1st flip-flop get next state pulse so consider 1 FF previous state.
- Consider all 1st state 0 all F.F.

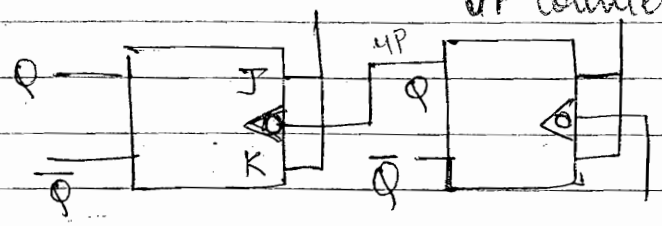
Ripple → frequency Divider

Q_B → trigger by clk (-ve edge) Q_A → trigger by -ve edge of Q_B → Q_C trigger by -ve edge of Q_A

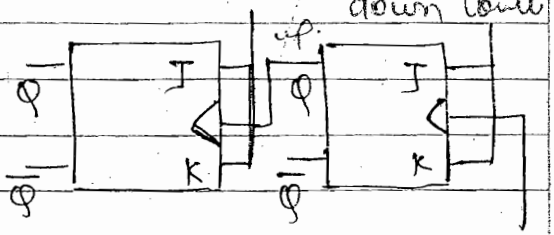
- Also called **frequency Divider**

Because Time period multiple by 2
&
frequency. Divided by 2.

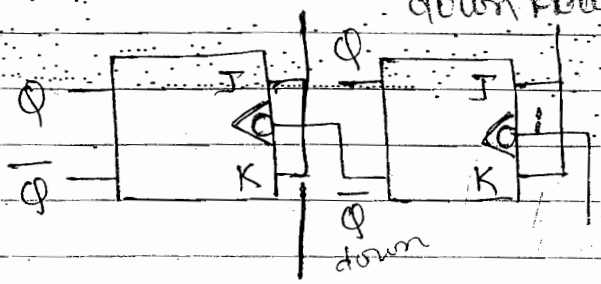
- (-ve) triggering Q — UP counter



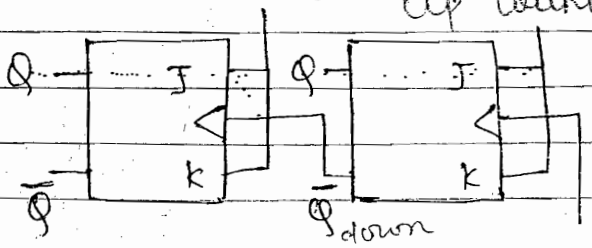
(+ve) triggering Q. — down counter



- (-ve) trigger $\bar{Q}$ — down counter



(+ve) trigger Q — up counter



Shortcut (−)ve triggering → up connecting up counter
→ down connected down counter

+ve triggering → up connecting down counter
down connected up counter.

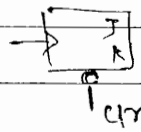
11000
2^4 2^3 2^2 2^1 2^0
16 8 (24)

- **Designing of Asynchronous MOD Counter :-**

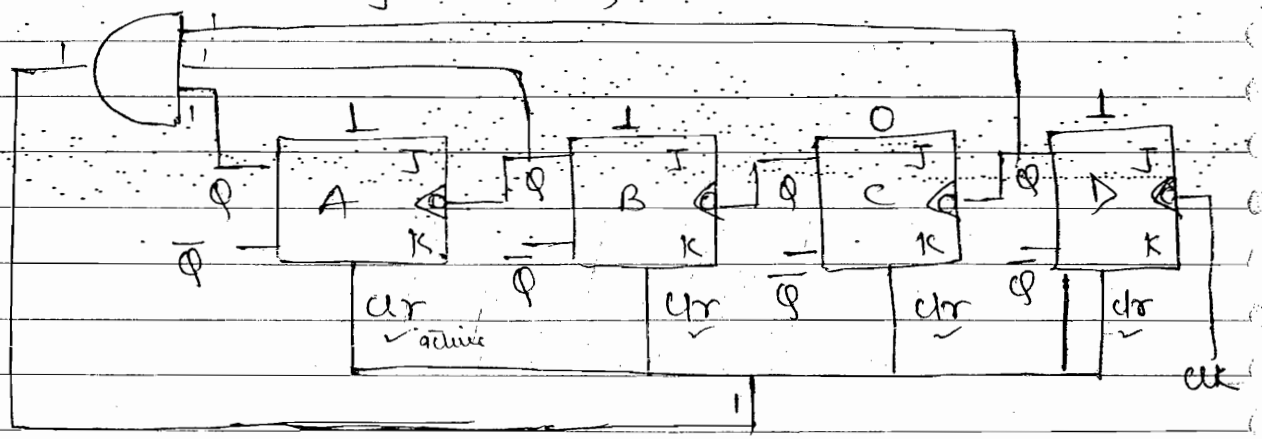
shortcut Right Trigger → Right Code
Left Trigger → opposite code.

ex:- 1101 → 1011

- If clr is trigger by → 1 (active) — AND Gate Used.
- If clr is active by → 0 — NAND Gate Used

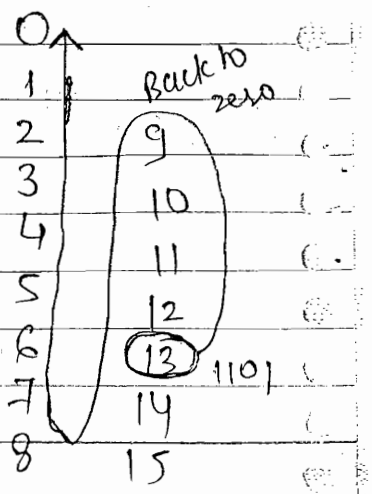


- AND OR NAND gate are connected by 1 [those (F.F.) having 1 value]



Right Trigger

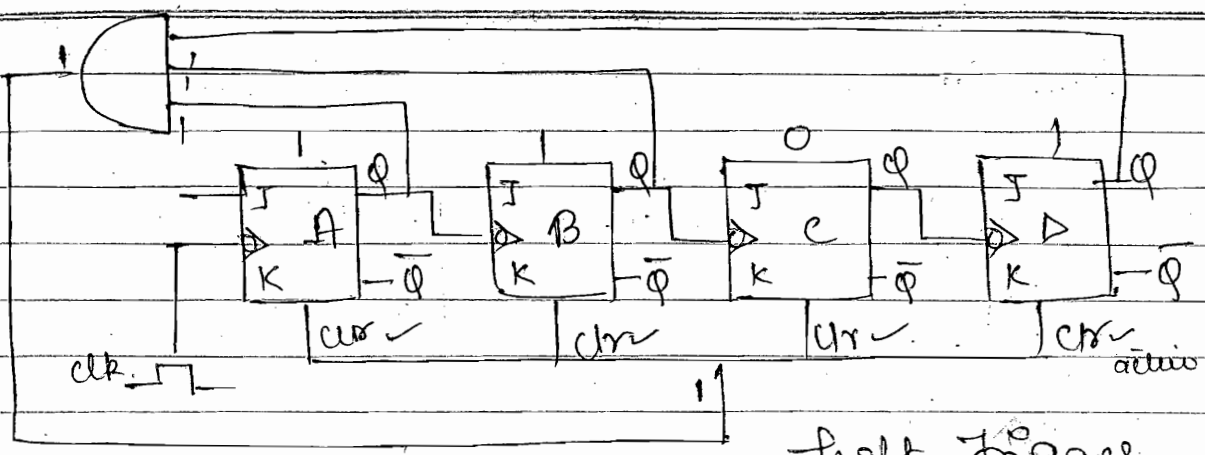
For MOD 13 ⇒ (1101) (code)



Steps

- Given design MOD 13 first write the code of 13. 1101, this code write in F.F, those flip flop having 1 at o/p. → connect those F.F. with AND get to get 1 at o/p AND gate so that clr will get active and counter o/p become 0000 (reset)

for MOD 13 1101



left trigger

(code will reverse)

MOD 13 ⇒ 1101

⇒ 1011

MOD 11

Right trigger

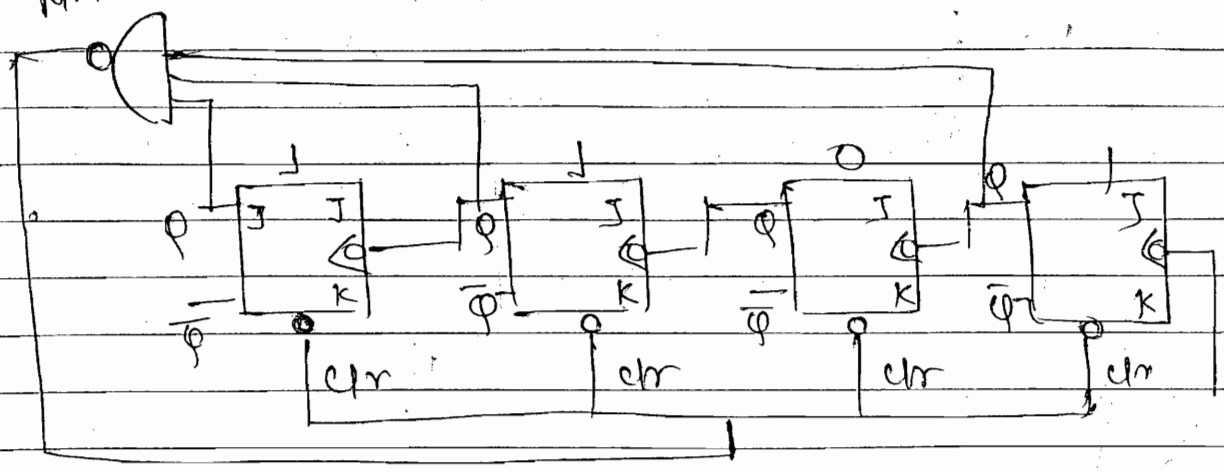
	A	B	C	D	A	B	C	D
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	0	0
3	0	0	1	1	1	1	0	0
4	0	1	0	0	0	0	1	0
5	0	1	0	1	1	0	1	0
6	0	1	1	0	0	1	1	0
7	0	1	1	1	1	1	1	0
8	1	0	0	0	0	0	0	1
9	1	0	0	1	1	0	0	1
10	1	0	1	0	0	1	0	1
11	1	0	1	1	1	1	0	1
12	1	1	0	0	0	0	1	1
13	1	1	0	1	1	0	1	1
14	1	1	1	0	0	1	1	1
15	1	1	1	1	1	1	1	1

left trigger

The unstable state from where F.F. Reset to 0000 donot count is counted and called Quasi State

If clr having Bubble.

NAND Gate

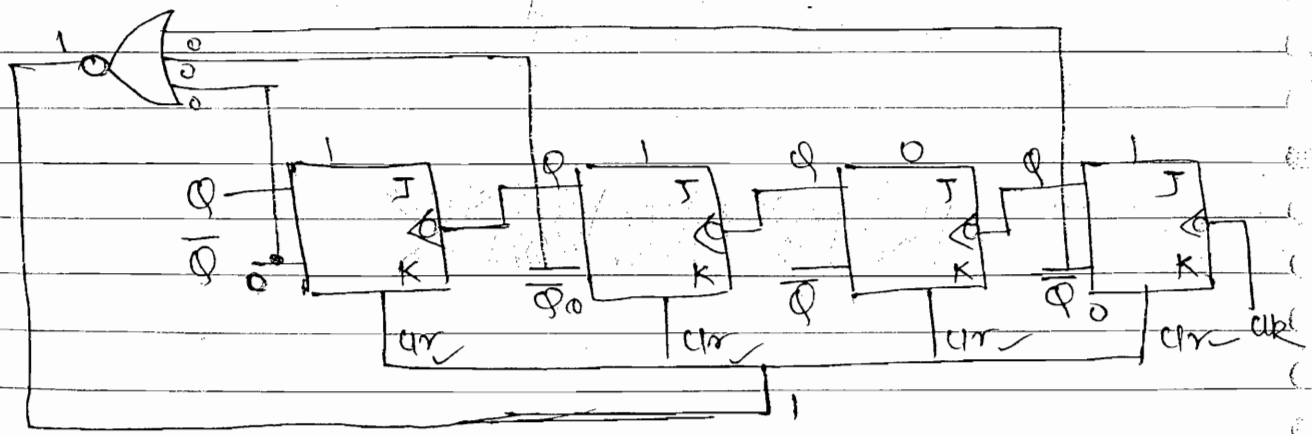


- As we are using AND gate, NO Need of clk pulse is required to make F.F. in Reset position after the undesirable state comes ex:- Mod 13 → 1101 when this comes. F.F. O/p. reset to 0000 automatically.

- But if we don't use AND gate counter O/p will Reset only when clk pulse will applied. automatic action will not occur.

24/9/2014

By using NOR Gate :- MOD 13 - 1101



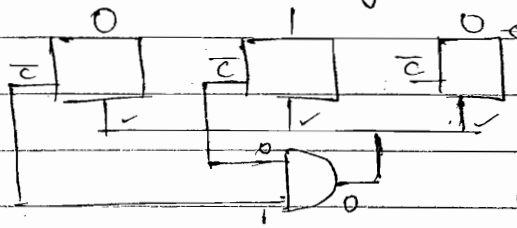
- Connect with $\overline{Q}$. don't connect with Q O/p F.F. because before coming of 13, 0 will comes in some Before state, so it will return from that state Not comes upto 13.

Shortcut:-

NAND	Q	$\overline{clr}$
NOR	$\overline{Q}$	clr
AND	Q	clr
OR	$\overline{Q}$	$\overline{clr}$

Q.39 $\overline{clr}$, $110 \rightarrow \overline{Q}$ So OR gate MOD 6

Don't use AND gate because if we use



000
001
010
011
100
101
110 ✗

Return Back from 010 and became MOD 2

- Down Counters :-

Connect the the gate i/p with 1 with the 1 in that position and take the code, that code ~~this~~ substracted by Maximum value.

No. of F.F

Ex:- 101 ⟶ 7 - 5 = 2 MOD 2 (3 flip-flop)

(2^n)

101 ⟶ 15 - 5 = 10 MOD 10 (4 flip-flop)

- Right Trigger ⟶ Right code (value)
  Left Trigger ⟶ Reverse value & substract By Maximum value (state).

- Procedure To find Mod value for A Syn. Up Counters.

Step 1:- Keep the one's (1's) at those flip-flops which are having connection to the logic gate & other F.F. should be kept at zero]

Step 2:- Check wheather it is right trigger or left trigger.

Right Trigger ⟶ Right code
Left Trigger ⟶ Reverse code is the mod value

- Procedure for Down Counter:-

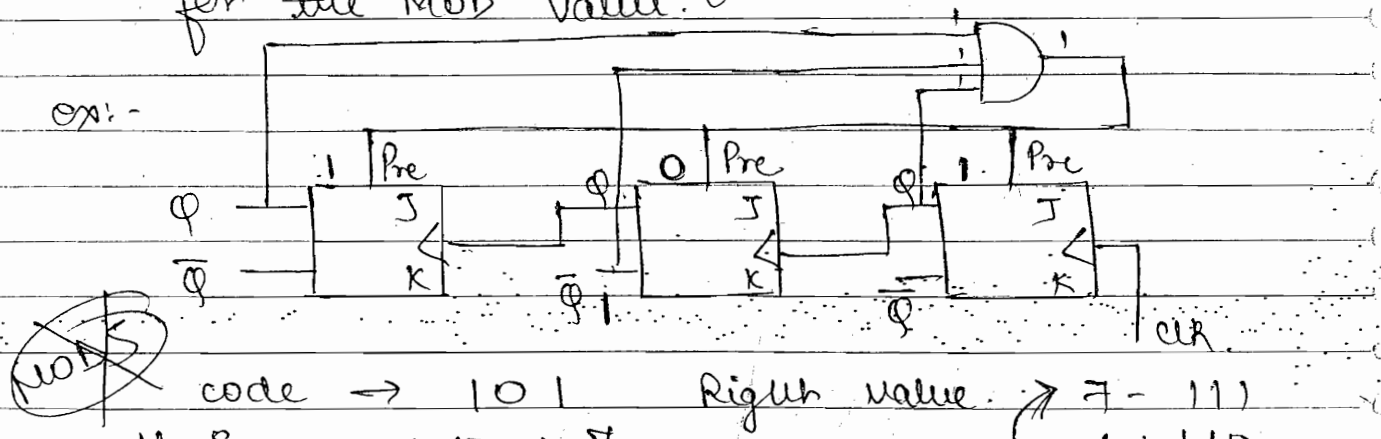
Step1: Keeps the 1's at those position from where logic gate is having connection. Other position should be kept at zero.

Step2 check wheather it is right Trigger or left Trigger.

Right Trigger ⟶ Right value
Left Trigger ⟶ Reverse value

Should be substract from the maximum state for the MOD value.

ex:-

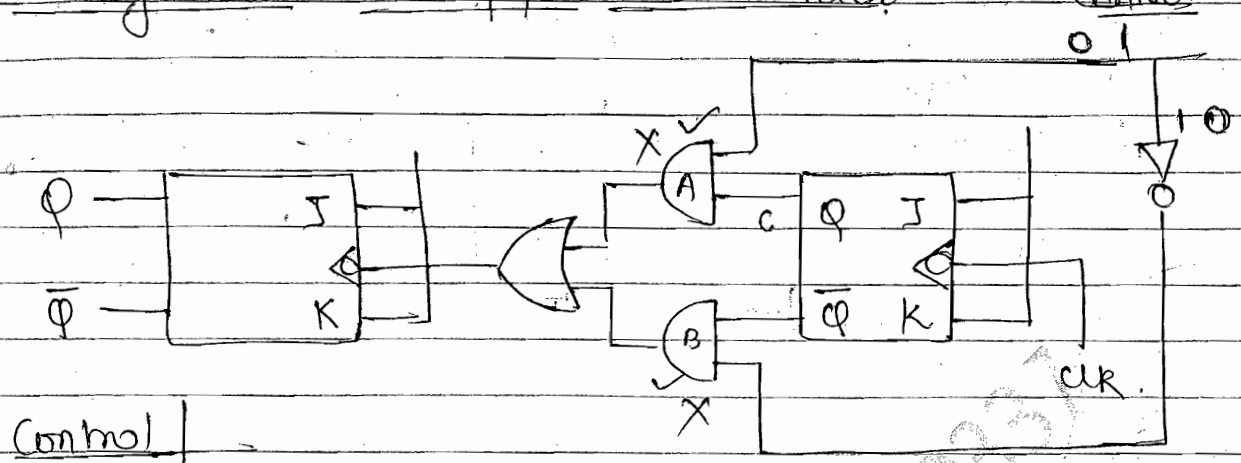


code ⟶ 101 Right value ⟶ 7 - 111
5 - 101
5 101 X

Maximum state ⇒ 7
⇒ 7 - 5 = 2
= MOD 2

ex:- Left Trigger. Suppose code is 100
Reverse value is 001
Maximum state = 7 ⇒ 7 - 1 = 6
= MOD 6

• Asynchronous up/down Counter :-



Control	
0	B operate → down counter (pass the value present in $\bar{Q}$)
1	A operate → up counter (pass the value present in Q)

Conventional

• Synchronous Counter :- [Orderly sequence]

V.V.Imp

(NO check for Right or left Trigger). Simultaneously Trig.

Synchronous Up Co.

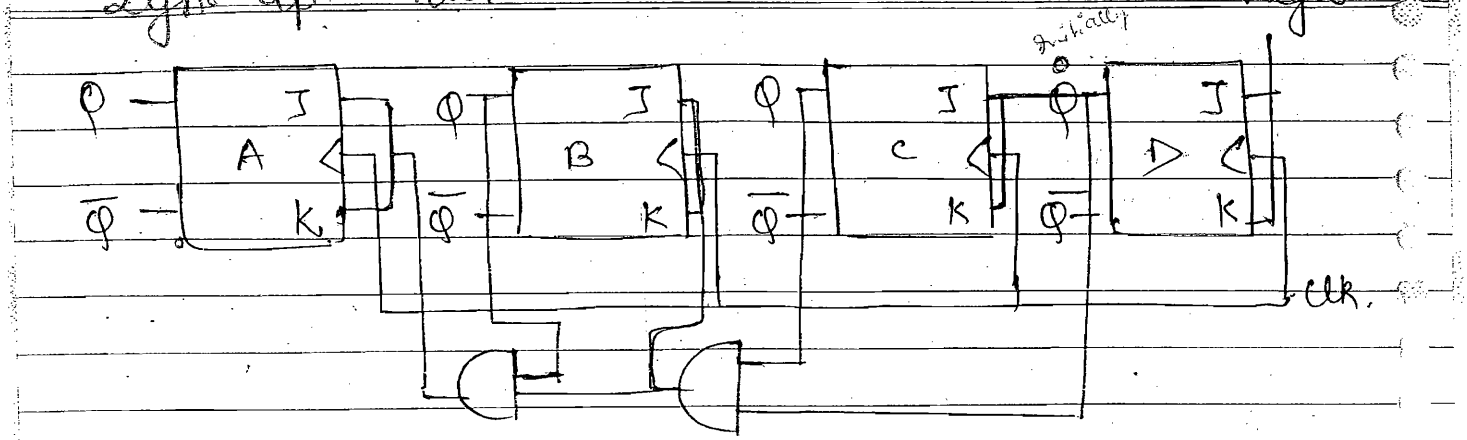
A	B	C	▷
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Syn. Down Counter

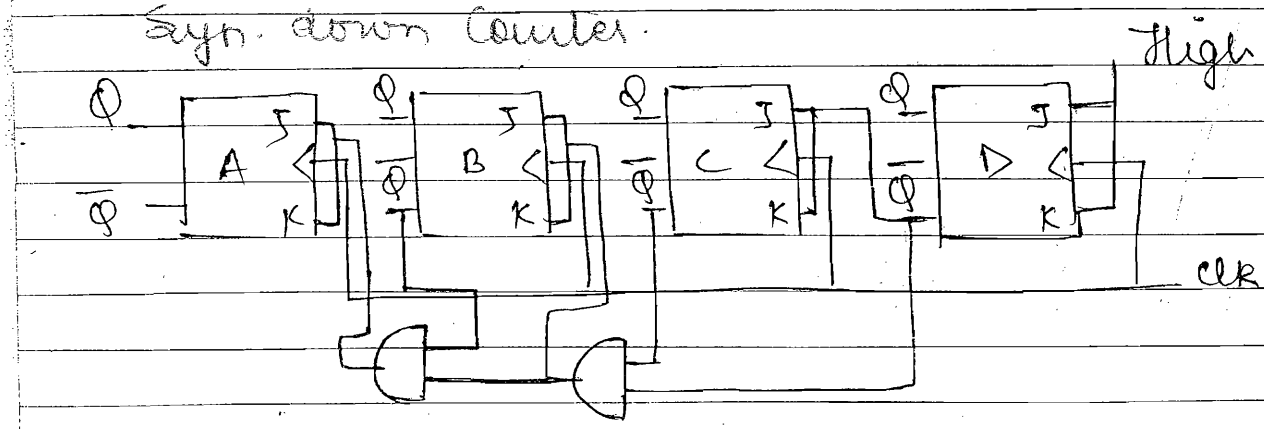
A	B	C	▷
1	1	1	1
1	1	1	0
1	1	0	1
1	1	0	0
1	0	1	1
1	0	1	0
1	0	0	1
1	0	0	0
0	1	1	1
0	1	1	0
0	1	0	1
0	1	0	0
0	0	1	1
0	0	1	0
0	0	0	1
0	0	0	0

Called Synchronous Series Carry

Syn. Up Counter



Syn. down Counter.



- When $Q_A = 1$ C → toggle so J,K (C) → Q (connected) $Q_C = 1$ B → toggle. So, $Q_D = Q_C = 1$ So use AND gate, together give o/p to J-K (B), A → toggle when $Q_D = Q_C = Q_B = 1$ So use AND gate, together give o/p. to J-K (A),

Objective

	Time delay.
• Asynchronous	N.Tc
• Synchronous	Tc
• Synchronous Series Carry	Tc + (N-2)TG

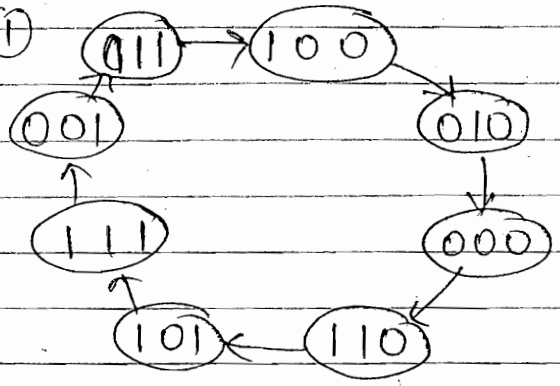
TG ⇒ Gate delay.

- If Mention Syn. Series Carry OR ORderly Manner go for this Method otherwise Next coming Method Any ORder

• Synchronous Design (NO Need Order)

S	N	E	K	B	D
State Diagram	Next state	Excitation Table	K-Map	Boolean equation	Design

Client Requirement:- 4 2 0 6 5 7 1 3. (By JK)

① 

② JK Table

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

③

	Present state $Q_2\ Q_1\ Q_0$	Next state $Q_2^+\ Q_1^+\ Q_0^+$	$J_0\ K_0$	$J_1\ K_1$	$J_2\ K_2$
4	1 0 0	0 1 0	0 X	1 X	X 1
2	0 1 0	0 0 0	0 X	X 1	0 X
0	0 0 0	1 1 0	0 X	1 X	1 X
6	1 1 0	1 0 1	1 X	X 1	X 0
5	1 0 1	1 1 1	X 0	1 X	X 0
7	1 1 1	0 0 1	X 0	X 1	X 1
1	0 0 1	0 1 1	X 0	1 X	0 X
3	0 1 1	1 0 0	X 1	X 1	1 X

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	0	X 1	X 3	2
Q_2	1 4	X 5	X 7	1 6

$$J_0 = Q_2Q_1$$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	X 0	1	X 3	X 2
Q_2	X 4	5	7	X 6

$$K_0 = \bar{Q}_2Q_1$$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	0	1	X	X
Q_2	4	5	X	X

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	X	X	3	2
Q_2	X	X	7	6

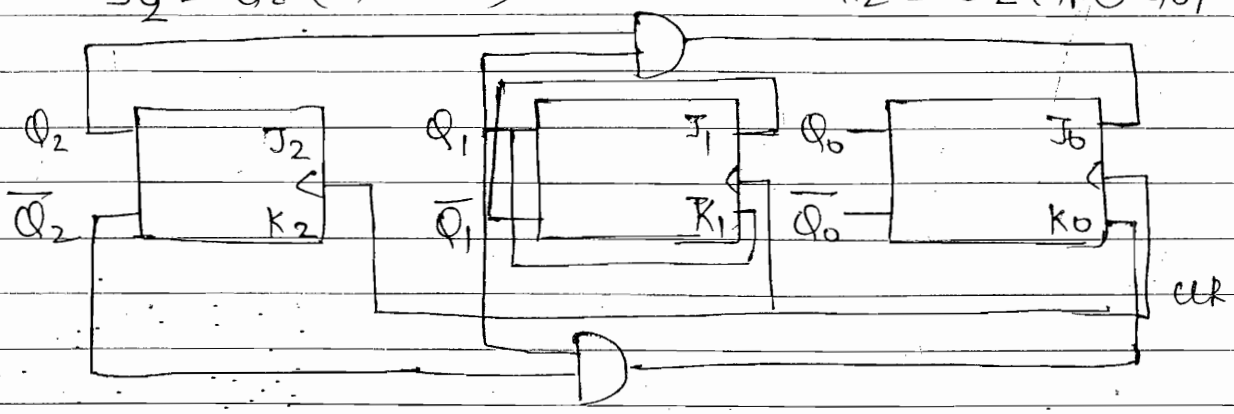
$J_1 = \bar{Q}_1$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	1	0	1	0
Q_2	X	X	X	X

$K_1 = Q_1$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	X	X	X	X
Q_2	1	0	1	0

$J_2 = \bar{Q}_2 (Q_1 \odot Q_0)$ $\quad$ $K_2 = Q_2 (Q_1 \odot Q_0)$



Q T flip flop. 0 2 3 1 0

Q	Q^+	T
0	0	0
0	1	1
1	0	1
1	1	1

	Present State		Next State		T_1	T_2
	Q_2	Q_1	Q_2^+	Q_1^+		
0	0	0	1	0	0	1
2	1	0	1	1	1	0
3	1	1	0	1	0	1
1	0	1	0	0	1	0
0	0	0				

	$\bar{Q}_1$	Q_1
$\bar{Q}_2$		1
Q_2	1	

$T_1 = Q_1 \oplus Q_2$

	$\bar{Q}_1$	Q_1
$\bar{Q}_2$	1	
Q_2		1

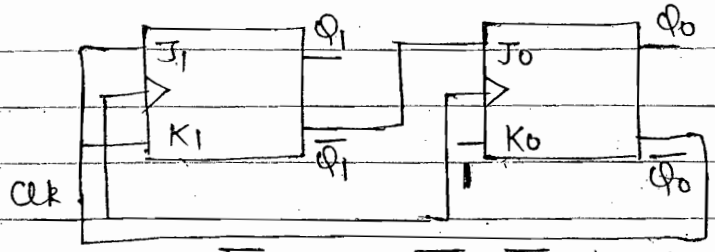
$T_2 = Q_1 \odot Q_2$

- **Procedure To find MOD Value: For Syn. Counter:-**

Step 1:- Write I/p and O/p in form of Table

Step 2:- Write the I/p connection on their head.

Step 3:- Keep some default state and make a horizontal line.



$K_0 \rightarrow$ Keeps 1

$\overline{Q_1}$ 1 $\overline{Q_0}$ $\overline{Q_0}$ step ②

CLK	J_0	K_0	J_1	K_1	Q_0	Q_1	
					0	0	step ③
↓	1	1	1	1	1	1	
↓	0	1	0	0	0	1	
↓	0	1	1	1	0	0	✗ Repeated

MOD 3.

Ans will Be

$Q_0 Q_1 \Rightarrow 00, 11, 01, 00 \ldots$

or $11, 01, 00, 11 \ldots$

$Q_1 Q_0 \Rightarrow 00, 11, 10, 00 \ldots$

OR $4(Q_1) + 2(Q_0) = ?$ $\Rightarrow 4(0) + 2(0) = 0$

$4(1) + 2(1) = 6$

Ans $\rightarrow$ 0, 6, 4, ... $4(1) + 2(0) = 4$

Q8. Ans No 63.

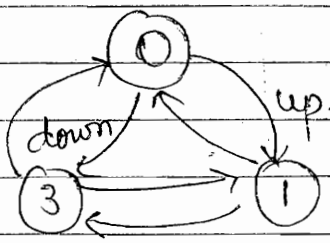
CLK	$\downarrow J_A$	$\downarrow K_A$	$Q_C \downarrow J_B$	K_B	$\downarrow J_C$	$\overline{Q_A} \downarrow K_C$	Q_A	Q_B	Q_C	
							1	1	0	
↓	1	1	0	1	1	0	0	0	1	
↓	1	1	1	1	1	1	0	1	0	

Repeat

Q9 110 will Repeat after 2 pulse MOD 2

- Synchronous (up / down) → Externally control must
- K-Map for up counter → Take Both value of J_0 (up & down)

Given.



up	down
0	3
1	1
3	0

Up // Additional o/p for up.

y	Q_1	Q_0	Q_1^+	Q_0^+	J_0	K_0	J_1	K_1
0	0	0	0	1	1	X	0	X
0	0	1	1	1	X	0	1	X
1	0	1	1	0	0	X	X	1
1	1	0	0	0	X	1	X	1

down // Additional o/p for down

y	$P.S$		$N.S$		J_0	K_0	J_1	K_1
1	1	1	0	1	X	0	X	1
1	0	1	0	0	X	1	0	X
1	0	0	1	1	1	X	1	X

Q_0	Q^+	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{y}$	1	X	X	X
y	1	X	X	X

$J_0 = 1$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{y}$	X	0	1	X
y	X	1	0	X

$K_0 = y \oplus Q_1$

- When the empty space left keep their don't care
- Both J_0 of up / down taken in K-Map.

Both K_0 ——— ———

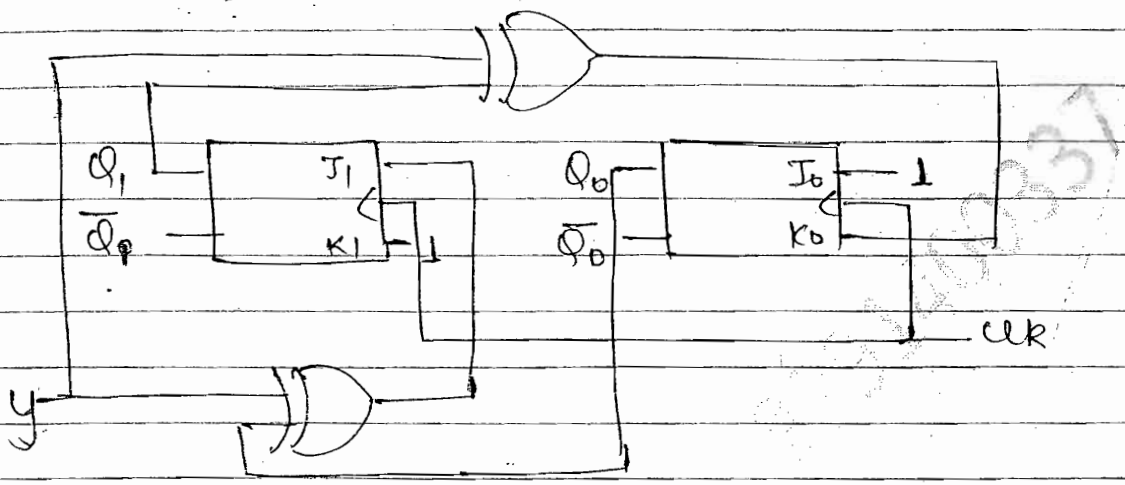
	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{y}$	0	1	X	X
y	1	0	X	X

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{y}$	X	X	1	X
y	X	X	1	X

$J_1 = \bar{y}Q_0 + y\bar{Q}_0$

$= y \oplus Q_0$

$K_1 = 1$



Q. Client Requirement :- 0 3 5 6 (up).

P.S.			N.S.			T_0	T_1	T_2
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0	J_0 K_0	J_1 K_1	J_2 K_2
0	0	0	0	1	1	1	1	0
0	1	1	1	0	1	0	1	1
1	0	1	1	1	0	1	1	0
1	1	0	0	0	0	0	1	1

Q	Q+	T
0	0	0
0	1	1
1	0	1
1	1	0

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	1	X	0	X
Q_2	X	1	X	0

$T_0 = \bar{Q}_1$

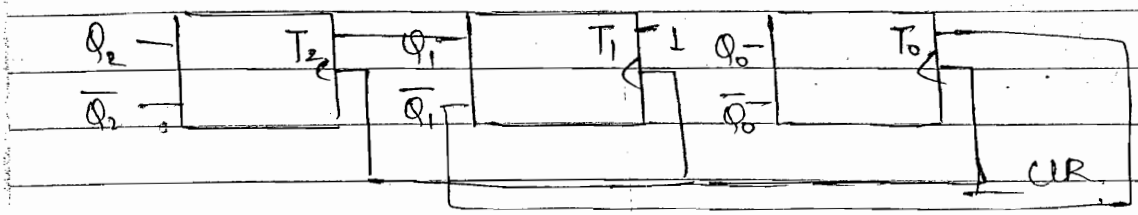
	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	1	X	1	X
Q_2	X	1	X	1

$T_1 = 1$

	$\bar{Q}_1\bar{Q}_0$	$\bar{Q}_1Q_0$	Q_1Q_0	$Q_1\bar{Q}_0$
$\bar{Q}_2$	0	X	1	X
Q_2	X	0	X	1

$T_2 = Q_1$

Wanted ⇒ 0 3 5 6 Unwanted ⇒ 1 2 4 7



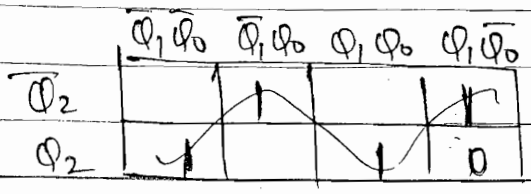
Q_1 1 $\overline{Q_1}$

CLK	T_2	T_1	T_0	Q_2	Q_1	Q_0	Suppose unwanted state → 4
				1	0	0 → 4	
1	0	1	1	1	1	1 → 7	
2	1	1	0	0	0	1 → 1	
3	0	1	1	0	1	0 → 2	
4	1	1	0	1	0	0 → 4	

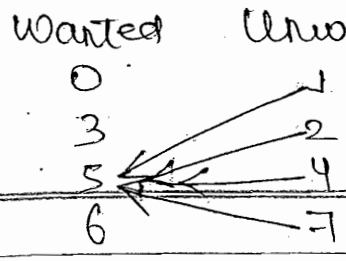
Lock Out

Self Corrected Counter :- Because of voltage fluctuation the counter go to unwanted state and if it is coming back to original state after one or more pulse it is known as self corrected counter if it is not coming back known as lock out condition

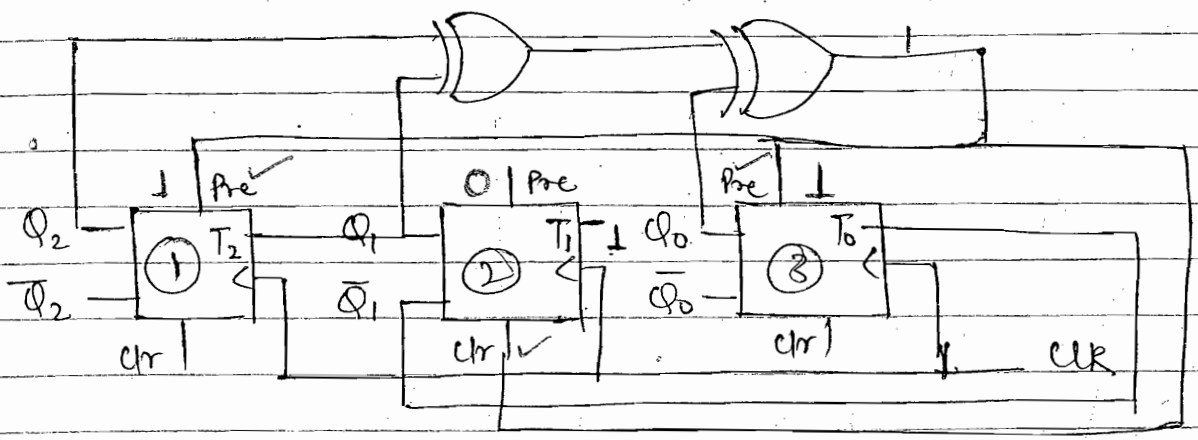
Methods of Avoid lock out Condition. Design an additional combitional ckt with the help of unwanted state whose o/p should be connected to desire preset and clear i/p. as per the required wanted state.



$f = Q_2 \oplus Q_1 \oplus Q_0$



Suppose want to come back original state (5) after any unwanted state.



5 → 1 0 1 Connect 1 with preset value and Connect 0 " clear value.

Suppose 3 → 0 1 1 1 → clear
2, 3 → preset.

25/9/2014

- State Reduction Method :-

State Diagram :- Graphical representation of a ckt performance consist of a parameters present state, I/p, Next state, O/p.

Step 1:- Draw the state diagram for the given ckt
Step 2:- Draw the state table for the state diagram.
Step 3:- Identify the equivalent state

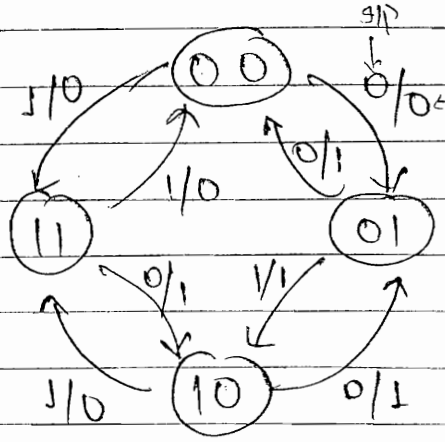
Equivalent state :- Two state are said to be equivalent if their Next state and o/p are equal.

Step 4:- If two state are equivalent one of the state should be eliminated

Steps :- Repeat the above process until we get all different state.

steps :-

Given



present state	Next state		output	
	x=0	x=1	x=0	x=1
0 0	0 1	1 1	0	0
0 1	0 0	1 0	1	1
1 0	0 1	1 1	1	0
1 1	1 0	0 0	1	0

x | y x → i/p y → o/p. 00 → i/p applied get. 01 which gives o/p → 0.

Pg No. 62

Q. 4

	Present state	Next state x=0	Next state x=1	Output x=0	Output x=1	
	a	c	b	0	0	5 state Ans.
	b	d	c	0	0	
② equivalent state	c	g e	d	1	1	
	d	e	f d	1	0	
equivalent state	e	f d	a	0	1	
①	f eliminate	g e	f	1	0 x	
	g	f	q	0	1 x	

eliminate.

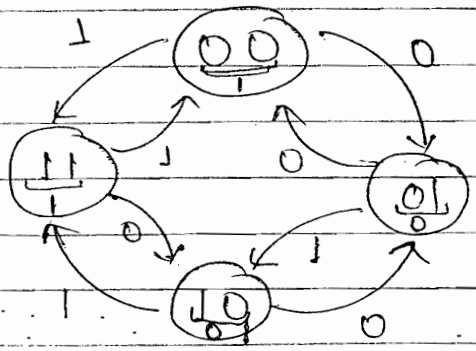
- Types (1) Mile Method
  (2) Moore Method

→ Present o/p depends upon present i/p and present state.

Mile Method:- The present o/p not only depends on the present state it also depend upon the present i/p.

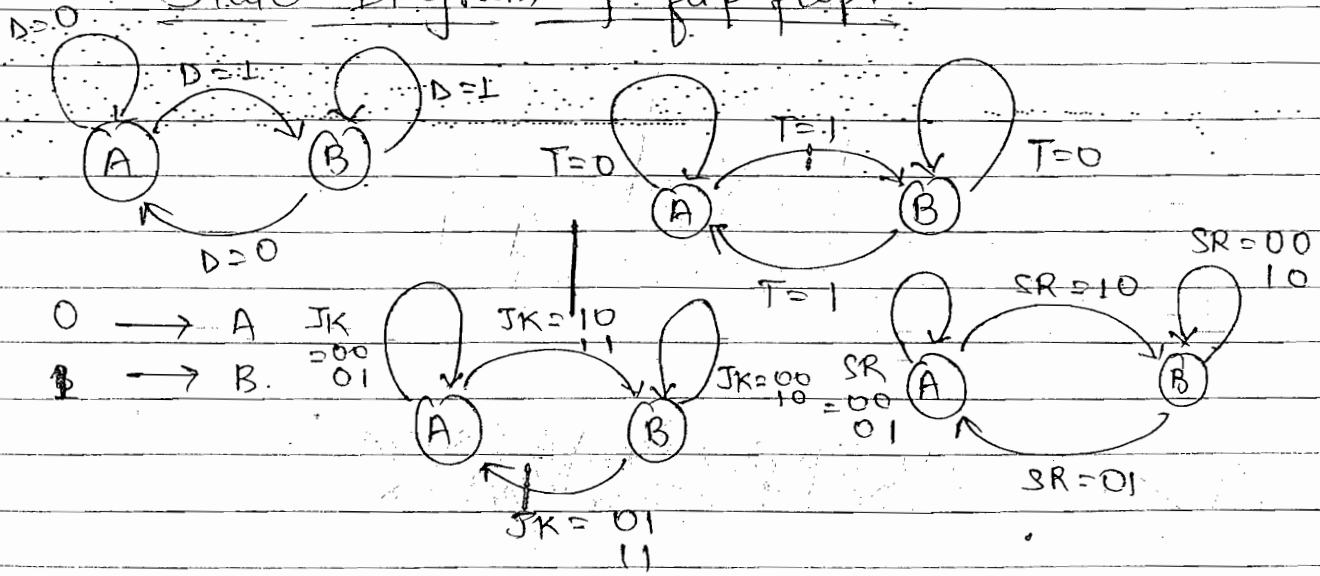
NOTE:- The above example is Mile Method only.

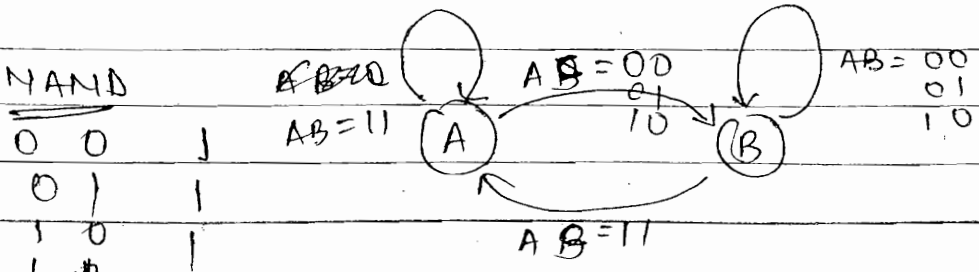
Moore Method:- In this mathod o/p the present o/p depend on only present state.



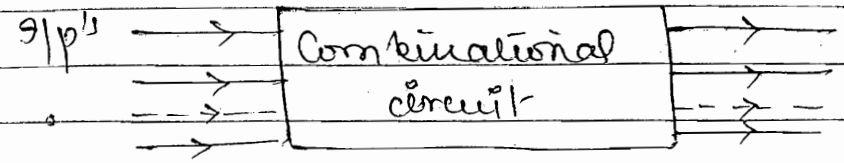
whatever will be the i/p, 0,1 o/p will depend only present state.

State Diagram of flip-flop:-





Combinational Circuit

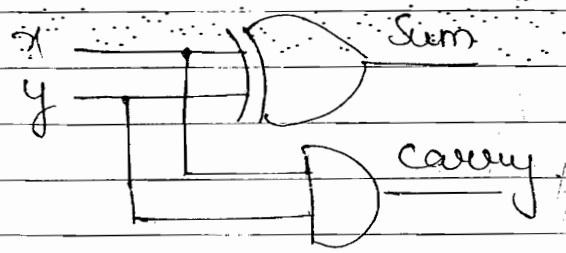


Half Adder

x	y	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Sum = $\bar{x}y + x\bar{y}$
= $x \oplus y$

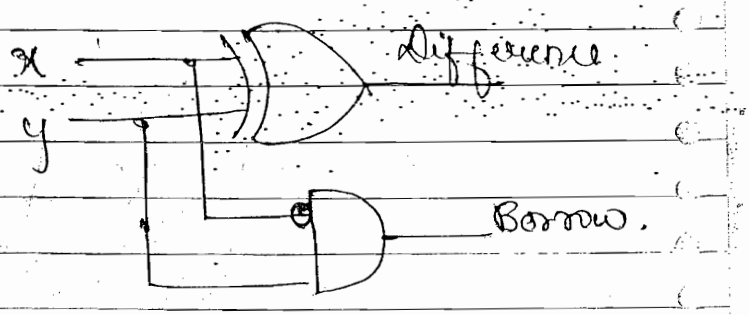
Carry = $x \cdot y$.



Half Subtractor

x	y	Borrow	Diff.
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Difference = $x \oplus y$
Borrow's = $\bar{x} \cdot y$



Full Adder

x	y	z	Carry	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Full Subtractor

x	y	z	Borrow	Differe.
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

✓ Sum and difference have same eq = $x \oplus y \oplus z$

✓ Carry and Borrow having also same eq. carry = $xy + xz + yz$

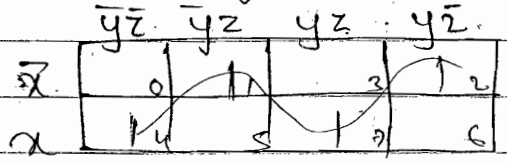
only Borrow having $\bar{x}$; Borrow = $\bar{x}y + z\bar{x} + yz$

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

$= x \oplus y \oplus z$.

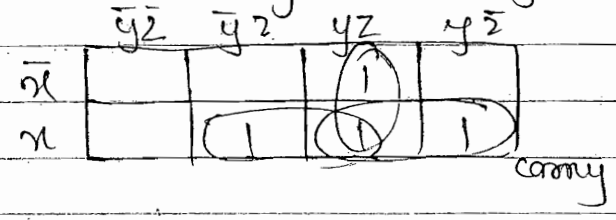
Difference = $x \oplus y \oplus z$.

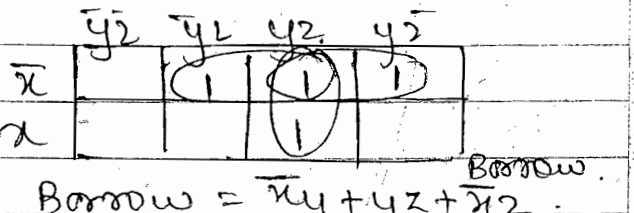
K-Map of Both sum & Diff.



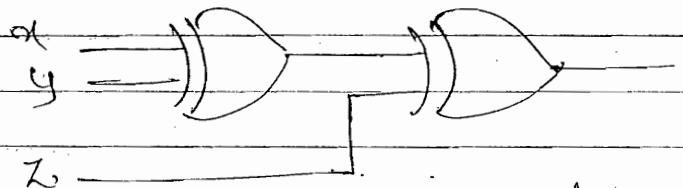
Carry = $\bar{x}yz + x\bar{y}z + xyz + xy\bar{z}$

$= xy + xz + yz$

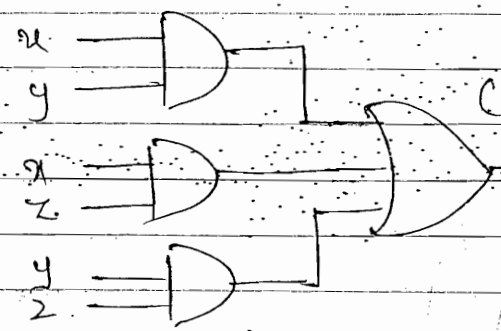


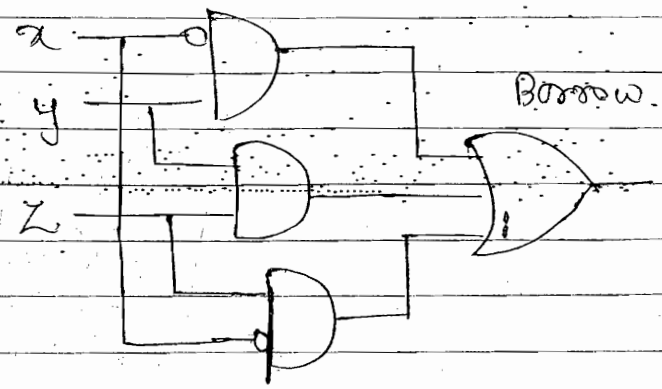


Borrow = $\bar{x}y + yz + \bar{x}z$.



Sum OR Difference.

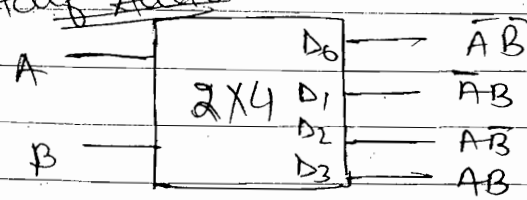




- **Decoder**:- These are used in memory system.

For n- select lines 2^n location activated

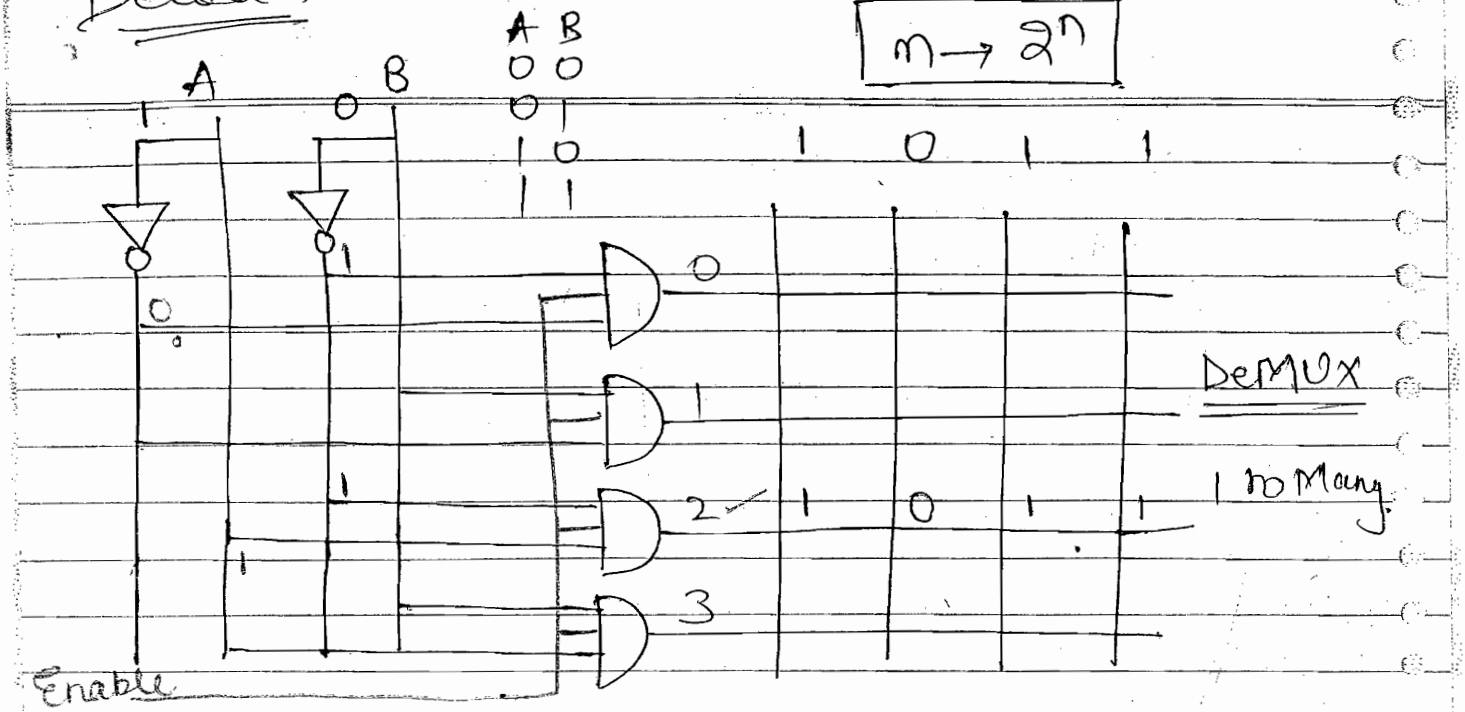
Half Adder



	A	B	Sum	Carry
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1

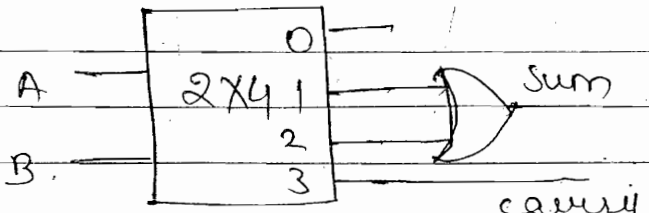
Decoders

$n \rightarrow 2^n$



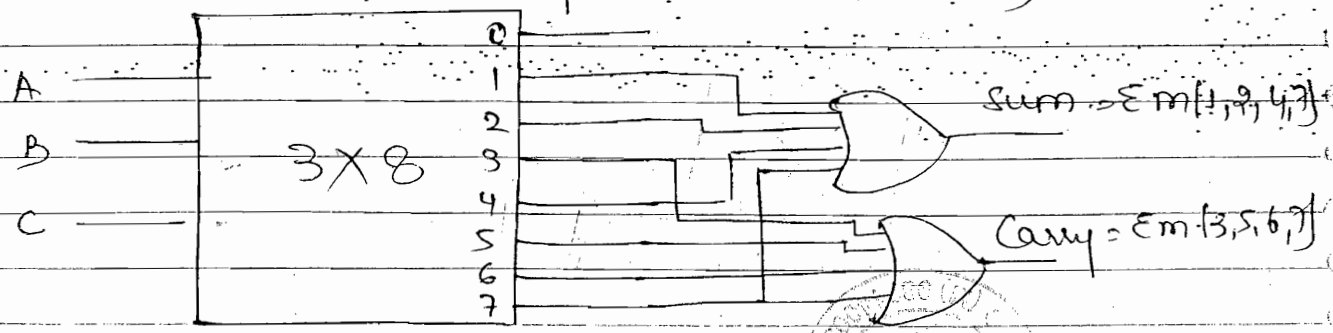
H.A Sum = $\Sigma m(1,2)$

Carry = $\Sigma m(3)$



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

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

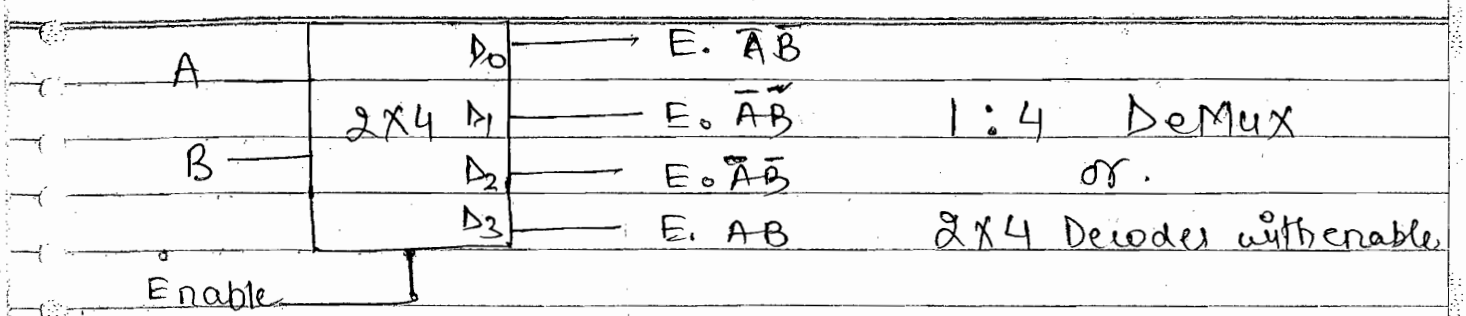


De Mux:- Decoder with enable i/p is DEMUX
It is 1 to Many circuit.

called as.

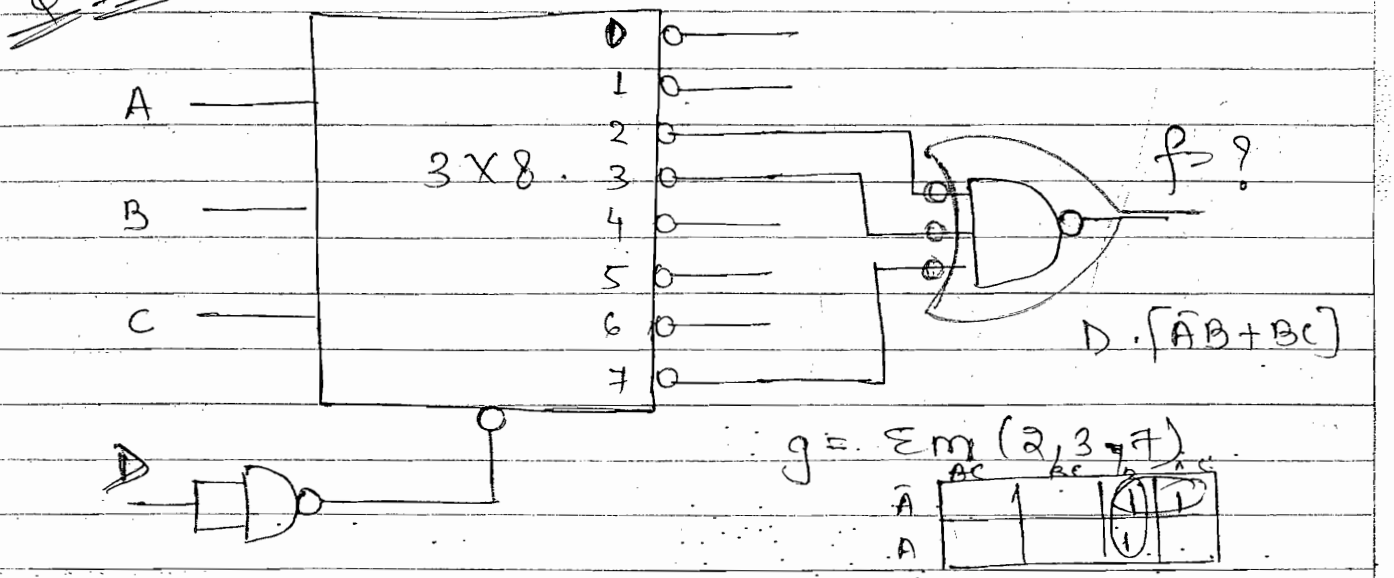
Decoder with Enable = DeMux

- If Enable (1) activated then only o/p comes.
  If Enable (0) deactivated o/p will always 0



1 : 4 DeMux
or.
2 X 4 Decoder with enable

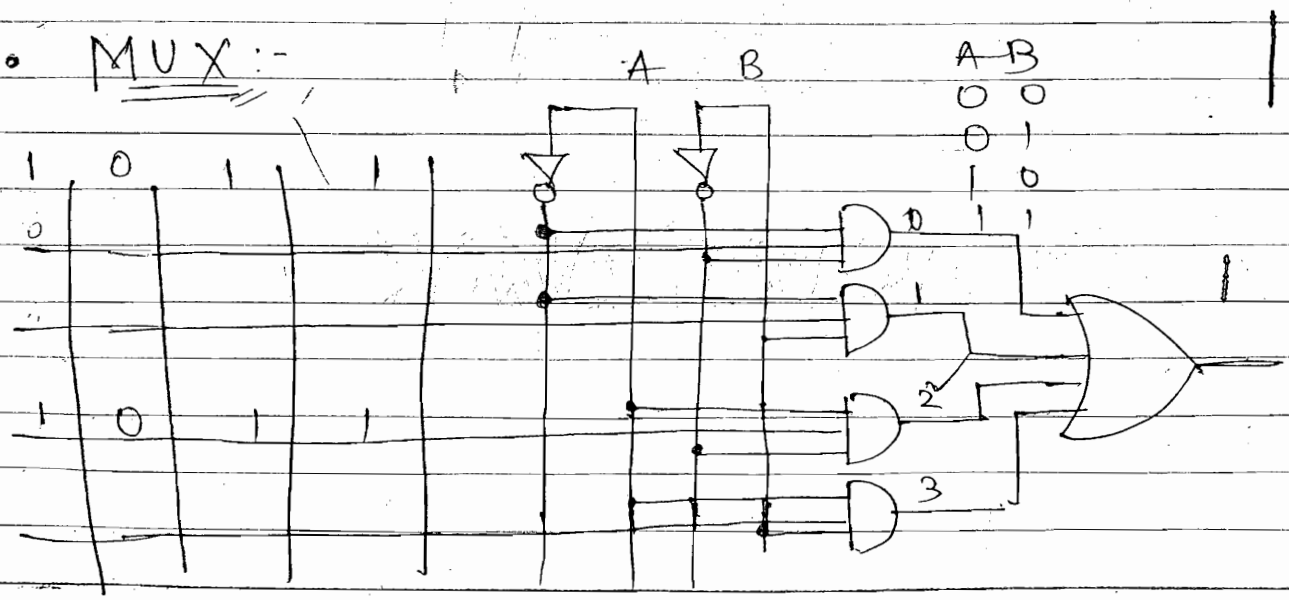
Q IES



g = Σm (2, 3, 7)

- DeMux or Decoder MUST have OR & NOR gate only otherwise change into OR & NOR ex. NAND given with equivalent to bubbled ~~NOR~~ OR.

- MUX :-

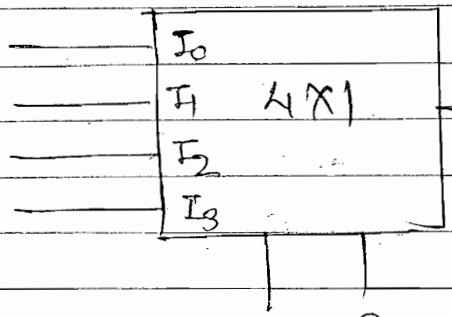


Names

→ Many to 1 circuit, waveform generator or parallel to serial Converter, Data Selector

- Enables can also be used if E=1 Operate
  E=0 O/p is 0.
- If given like that:

always find I_0 and write with $\bar{A}\bar{B}$



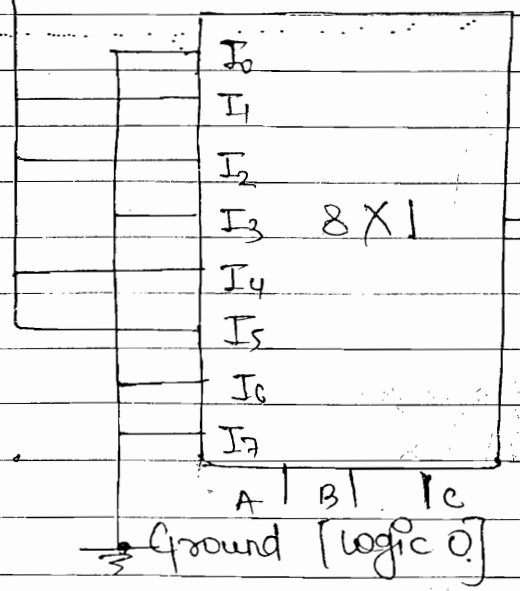
$f = (I_0)\bar{A}\bar{B} + (I_1)\bar{A}B + (I_2)A\bar{B} + (I_3)AB.$

V. Vo imp.

- Designing By MUX

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

Vcc [logic 1]



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

Maximum value 7. (3 bit)

So 8X1

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

Design By using 4X1 MUX! - $f = \Sigma m(1,2,4,5)$

	A	B	C	f	
0	0	0	0	0	I_0
1	0	0	1	1	
2	0	1	0	1	I_1
3	0	1	1	0	
4	1	0	0	1	I_2
5	1	0	1	1	
6	1	1	0	0	I_3
7	1	1	1	0	

$A = I_0$

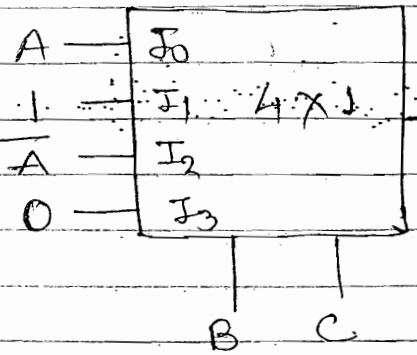
$1 = I_1$

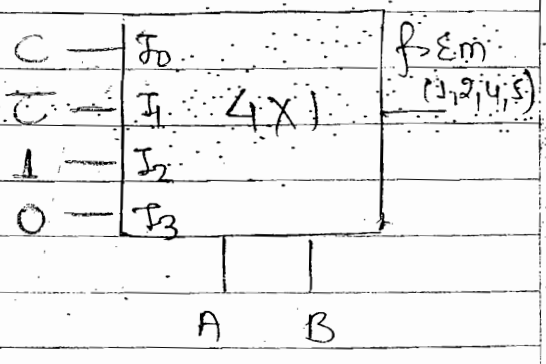
$\bar{A} = I_2$

$0 = I_3$

MSB (missing)
A used in Input
- Writing O/p f in form of A

LSB (missing)
C used in Input.
- Writing O/p f in terms of C





Shortcut लो। लो। ला। ला

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

ला
ला

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

By 2X1 MUX

- MSB in Select line
- LSB in Select line.

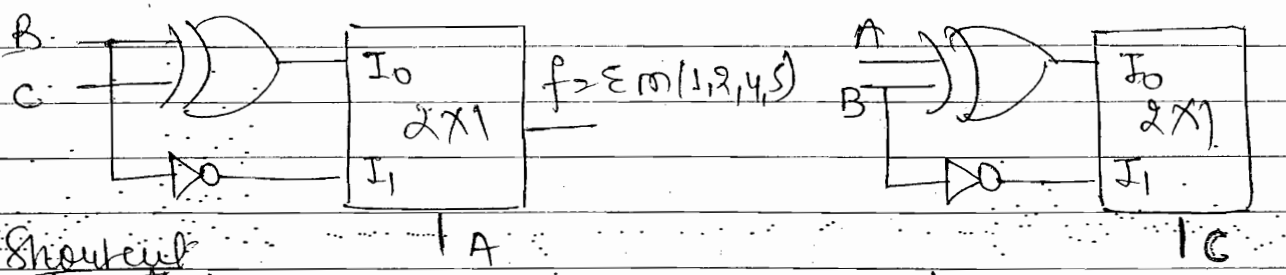
A in Selection

- C in Select line.

		A	B	C	f
I_0	0	0	0	0	0
	1	0	0	1	1
	2	0	1	0	1
	3	0	1	1	0
I_1	4	1	0	0	1
	5	1	0	1	1
	6	1	1	0	0
	7	1	1	1	0

$I_0 = B \oplus C$

$I_1 = \bar{B}$



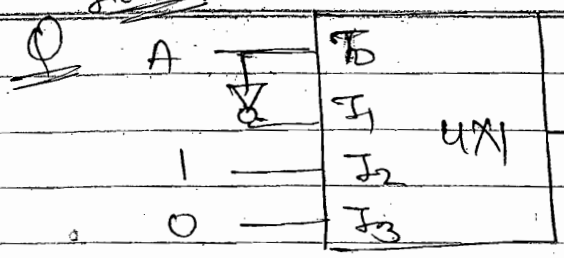
Shortcut

	I_0	I_1
$\bar{B}\bar{C}$	0	(4)
$\bar{B}C$	(1)	(5)
$B\bar{C}$	(2)	6
BC	3	7
	$\bar{B}C + B\bar{C}$	$\bar{B}\bar{C} + \bar{B}C$
	$B \oplus C$	$\bar{B}$

	I_0	I_1
$\bar{A}\bar{B}$	0	(1)
$\bar{A}B$	(2)	3
$A\bar{B}$	(4)	(5)
AB	6	7
	$A \oplus B$	$\bar{A}\bar{B} + AB$
		$= \bar{B}$

F.o.R.S

Q



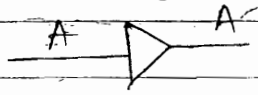
$f = ?$ $\quad f = \Sigma m(1,2,4,6)$

Solution

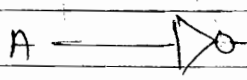
	I_0	I_1	I_2	I_3
$\bar{A}$	0	(1)	(2)	3
A	(4)	5	(6)	7
	A	$\bar{A}$	1	0

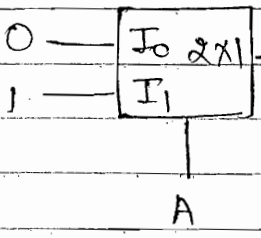
write this colum then circle the NO. and write the fuction.

- **Buffer**

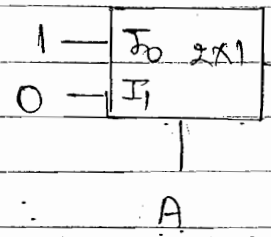


NOT





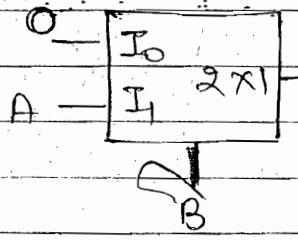
$f = \bar{A}(I_0) + A(I_1)$
$= \bar{A}(0) + A(1)$
$= A$



$f = \bar{A}(I_0) + A(I_1)$
$= \bar{A}(1) + A(0)$
$= \bar{A}$

AND

	A	B	f
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	(1)



$f = \bar{B}(I_0) + B(I_1)$
$= \bar{B}(0) + BA$
$= AB$

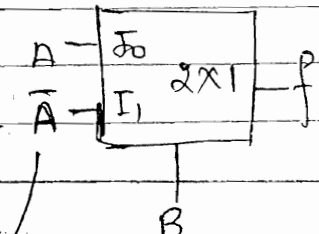
	I_0	I_1
$\bar{A}$	0	1
A	2	(3)
	0	A

लो लो लो लो

EXOR

	A	B	f
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

	I_0	I_1
$\bar{A}$	0	(1)
A	(2)	3
	A	$\bar{A}$



This gate I/p comes from A NOT.

Shortcut

2×1 MUX

NOT	1
AND	1
OR	1
EX-OR	2
EX-NOR	2
NAND	2
NOR	2

Q H.A. No. of MUX. 2×1.

Sum → Ex-OR → 2

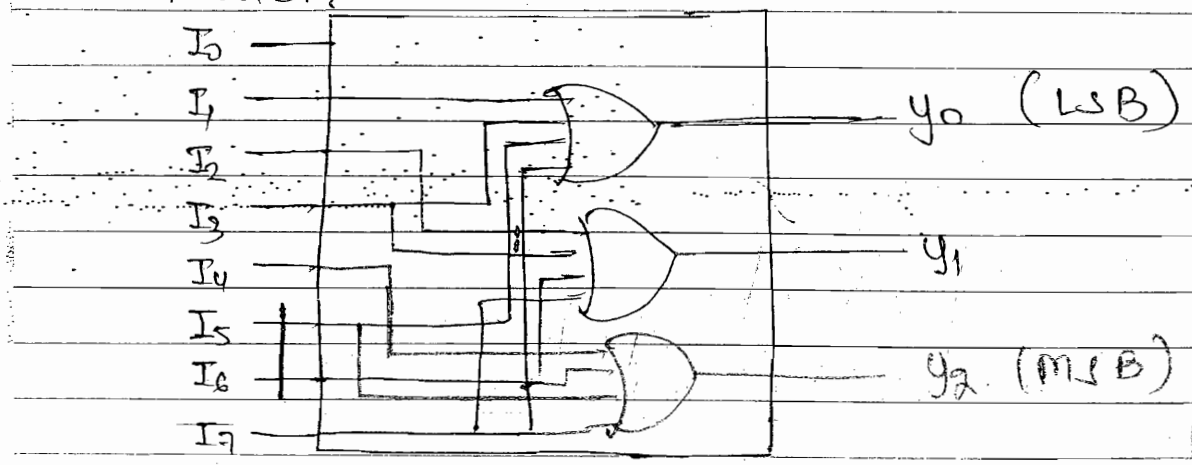
Carry → AND → 1

Total 3 Ag

26/9/2014

• Design A Circuit :- I/p → jis corresponding code will generate in O/p.

• Encoder :-



8 × 3 Encoder.

Octal to Binary converter.

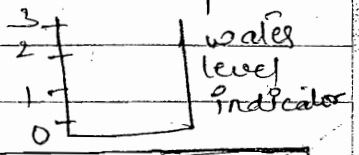
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	y_2	y_1	y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

For Hexa to Binary conversion 16X4

$y_0 = I_1 + I_3 + I_5 + I_7$ $\quad y_1 = I_2 + I_3 + I_6 + I_7$

$y_2 = I_4 + I_5 + I_6 + I_7$

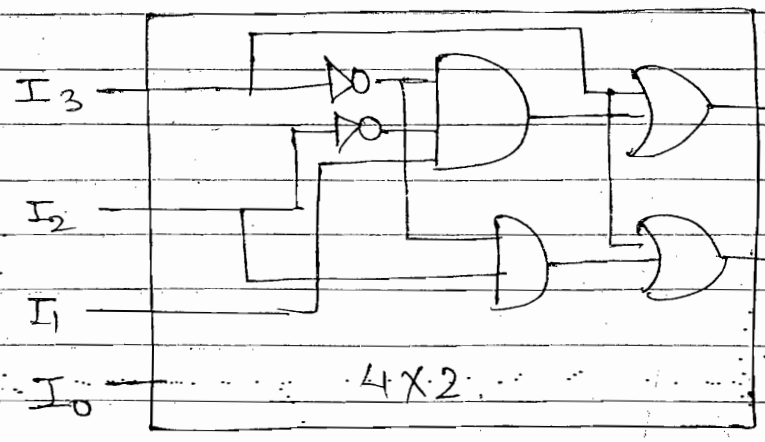
- Priority Encoder : Code will generate only when its higher NO. having 0 value.



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

$y_0 = I_1 I_2' I_3' + I_3$

$y_1 = I_2 I_3' + I_3$



code for 1 only generate when I_2, I_3 are 0, I_0 is whatever X.

- Higher Order Circuits By using lower Order ckt

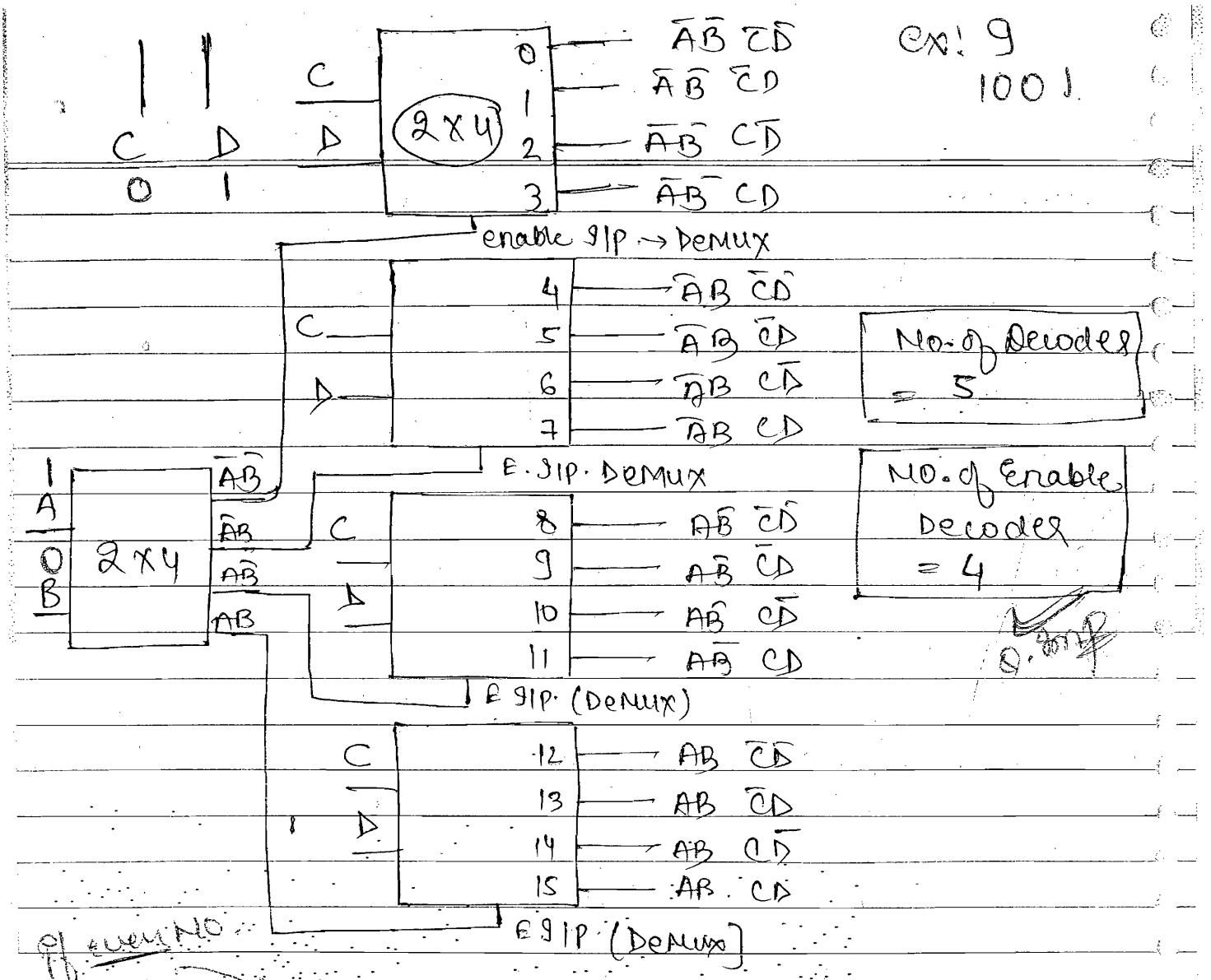
4X(16) concentrate in O/P ND.

By using 2X(4)

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

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

$4 + 1$ Total Required (5)



If even No.

Ex:- 16 x 256

H x 16

No external ckt Required.

256 ÷ 16

16 / 16 = 1

(17) required.

ex:- (3) x (8) By using 2 x (4)

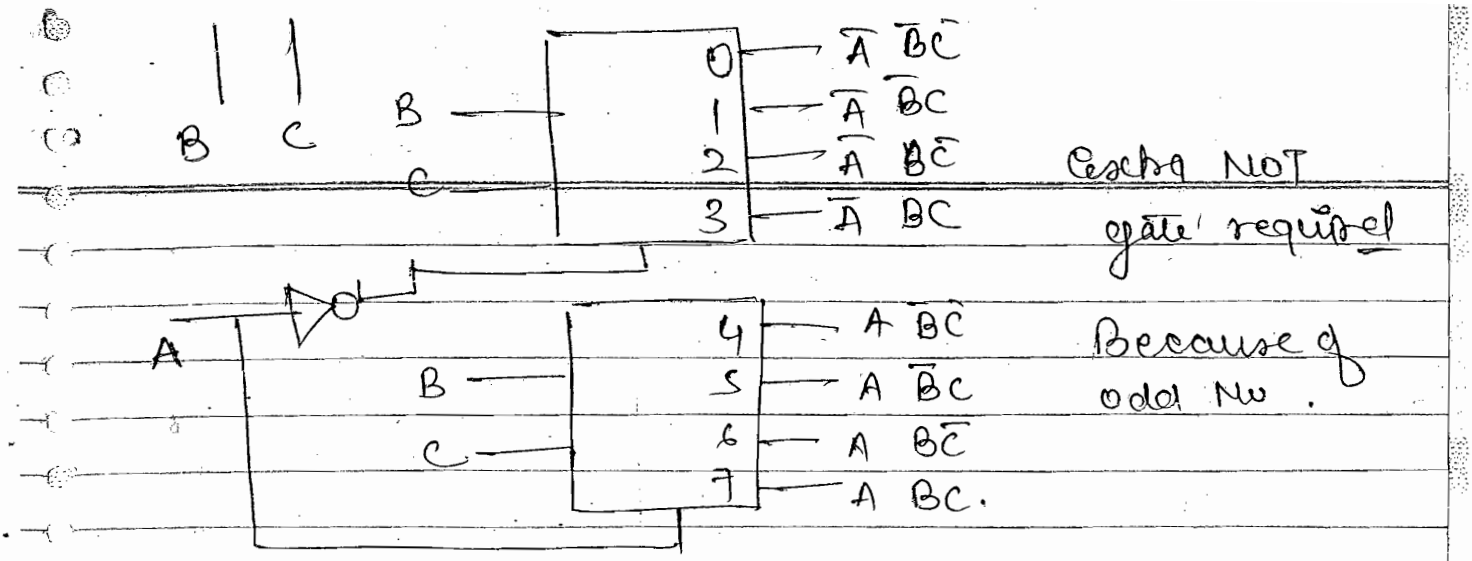
If odd No.

8/4 = 2

2/4 = —

Not possible

External ckt Circuit Required "NOT gate".



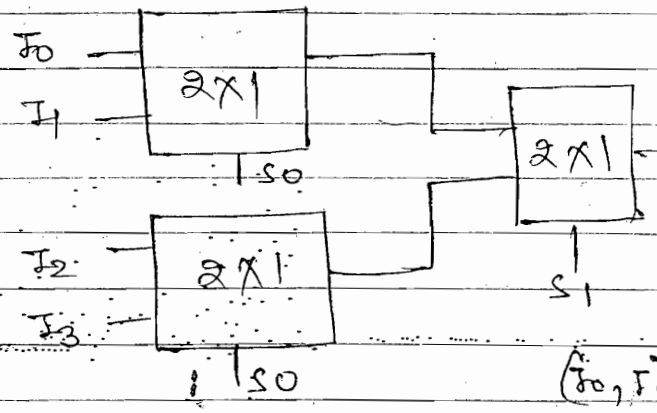
Extra NOT gate required

Because of odd No.

Total ckt required. 2 with NOT gate.

If NOT gate not given in options ⇒ Then 3 ckt Required

MUX :- (4)×1 using (2)×1 → Concentrate on I/p No.



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

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

(3) Total

So will select from both (I_0, I_1) (I_2, I_3) But By S_1 only 1 will select.

- Codes & Code Converters :-

BCD :- Valid only upto 4 digit "8 4 2 1" upto 9. only BCD written same as Binary.

Valid

10 → 1010
11 → 1011
12 → 1100
13 → 1101
14 → 1110
15 → 1111 → Binary code

Invalid 1/0

BCD 0001 0000

2/4

0001 0100

BCD → Not a self complimentary code

"2421" Code.

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

(2 fixed, 7 fixed)

"Self complimentary code"

- "5421" same as "2421" ~~upto two, [illegible]~~ half MSB is 0 and half LSB is 1
- Excess-3 (Ex-3 code) → Addition of 3 in BCD

0 → 0 0 0 0 → BCD
+ 0 0 1 1
0 0 1 1 → Ex-3

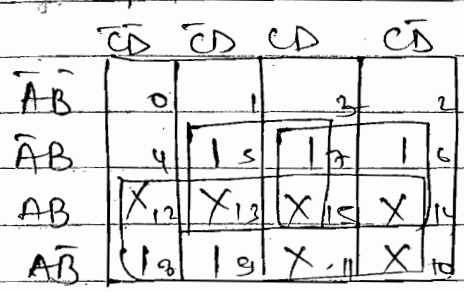
1 → 0 0 0 1 → BCD
+ 0 0 1 1
0 1 0 0 → Ex-3

- BCD to Ex-3 Code Conversion:

	I/p BCD				O/p Ex-3			
	A	B	C	D	w	x	y	z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0

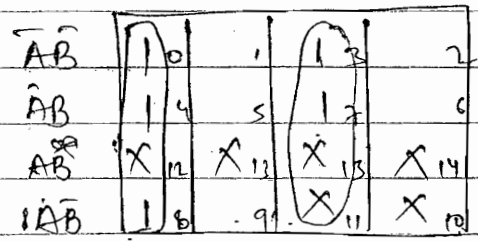
EX-OR → Arithmetic logic gate
Is used in Add & subtractor

6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

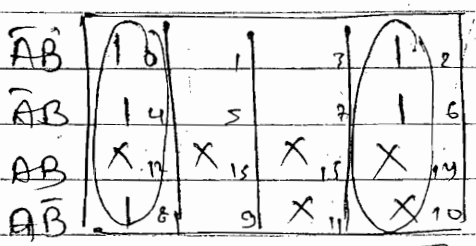


$W = A + BD + BC$

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

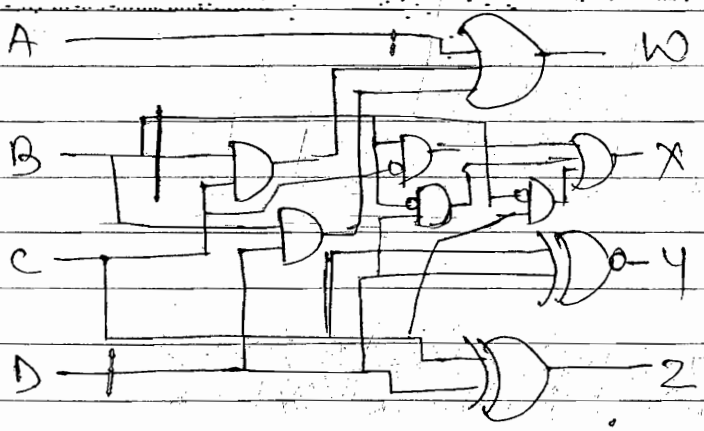


$Y = \bar{C}D + C\bar{D}$



$Z = \bar{C}D + C\bar{D}$

I/p. BCD O/p. Ex-3



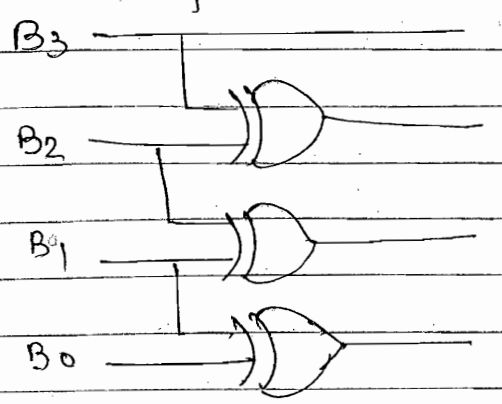
- Binary To Gray:- MSB will not change.

B_3 B_2 B_1 B_0

Binary 1 ⊕ 1 ⊕ 0 ⊕ 1

Gray ⇒ 1 (MSB g_3) 0 (g_2) 1 (g_1) 1 (g_0) = 1011

- Binary → Gray



$G_3 = B_3$

$G_2 = B_3 \oplus B_2$

$G_1 = B_2 \oplus B_1$

$G_0 = B_1 \oplus B_0$

- Gray To Binary Code:-

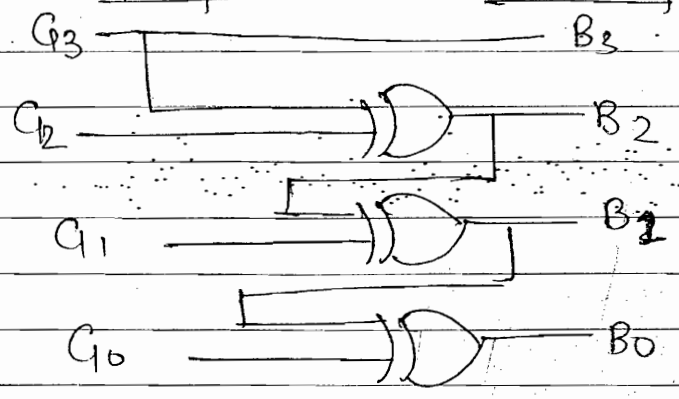
Gray ⇒ G_3 G_2 G_1 G_0 = 1 0 1 1

Binary ⇒ 1 1 0 1

MSB(B_3) B_2 B_1 B_0

Gray code:-
- ✓ unit distance code
- ✓ Hamming distance

Gray → Binary



$B_3 = G_3$

$B_2 = B_3 \oplus G_2$

$B_1 = B_2 \oplus G_1$

$B_0 = B_1 \oplus G_0$

	Binary code		Gray code		
0	0	0	0	0	
1	0	1	0	1	1 bit difference
2	1	0	1	1	1 "
3	1	1	1	0	1 "

2 bit difference

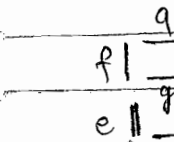
for Gray code

Hamming Distance is '1' in Gray Code

⇒ Gray code is "unit distance code"

⇒ Gray ⇒ K-Maps.

- B.C.D to Seven Segment Code Conversion:-



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

Draw K-map. A B C D → I/p.
a, b, c, d, e, f, g → O/p. (individually).
And write the eqn.

- Number Representation:-

Binary → 4 digit 8 4 2 1
5 digit 16 8 4 2 1
6 digit 32 16 8 4 2 1

1) Sign Magnitude
2) 1's Complement
3) 2's Complement

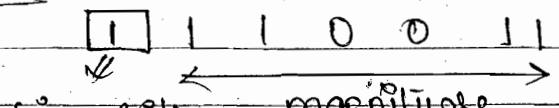
Case (i) +ve Number (sign, 1's, 2's complement)

same

+51 ⇒ 0 1 1 0 0 1 1

sign Bit. magnitude

Case(ii) −ve Number ex:- −51



1 0 0 1 1 0 0 → 1's compliment

0 0 1 1 0 0
+ 1

1 0 0 1 1 0 1 → 2's compliment.

Range for sign magnitude and 1's compliment.

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

ex:- n = 3 n = No. of I/p Bits.

$-(2^{3-1}-1)$ to $+(2^{3-1}-1)$

− 3 to +3

	Sign. Magnitude			1's Compliment			2's Compliment		
	x	y	z	x	y	z	x	y	z
(+ve) sign Bit	0	0	0 +0	0	0	0 +0	0	0	0 +0
	0	0	1 +1	0	0	1 +1	0	0	1 +1
	0	1	0 +2	0	1	0 +2	0	1	0 +2
	0	1	1 +3	0	1	1 +3	0	1	1 +3
(These value remain in 1's & 2's comp)	1	0	0 −0	1	1	1 −0	1	0	0 −4
	1	0	1 −1	1	1	0 −1	1	1	1 −1
	1	1	0 −2	1	0	1 −2	1	1	0 −2
	1	1	1 −3	1	0	0 −3	1	0	1 −3.

(−ve) Sign Bit

Changes comes only in −ve No

Range for 2's compliment

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

Special case for 2's compliment :- Carry enter into sign Bit.

Ex:- ① 1 0 0 $\Rightarrow -(2^{n-1})$

$\Rightarrow -4$

② 1 0 0 0 $\Rightarrow -8$

③ 1 0 0 0 0 $\Rightarrow -16$.

1 0 0	
1 1	1's compl.
+1	
1 0 0	2's compl.

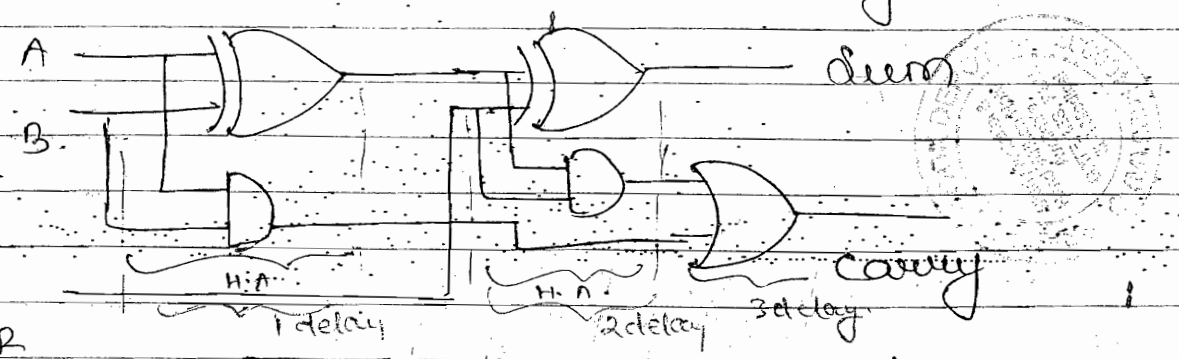
NO Representation for -0 in 2's compliment

Ex:-

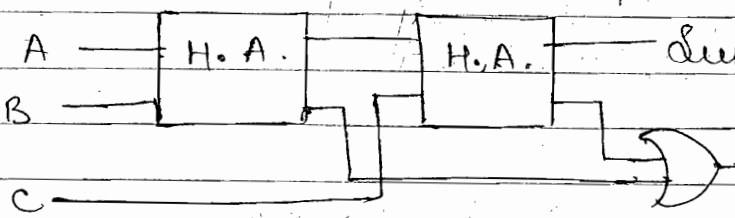
	Sign. Mg.	1's comp.	2's compl.
⓪ 0 1 0 1 (+ve)	+5	+5	+5
① 1 1 0 1 (-ve)	-13	-2	-3

28/9/2014

- Full Adder :- & Parallel Carry Adder



OR

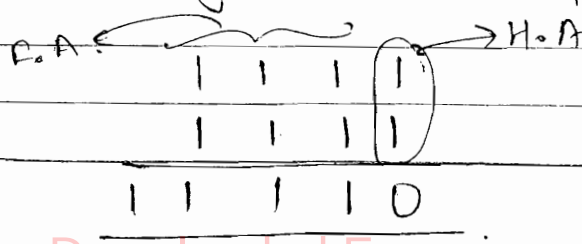


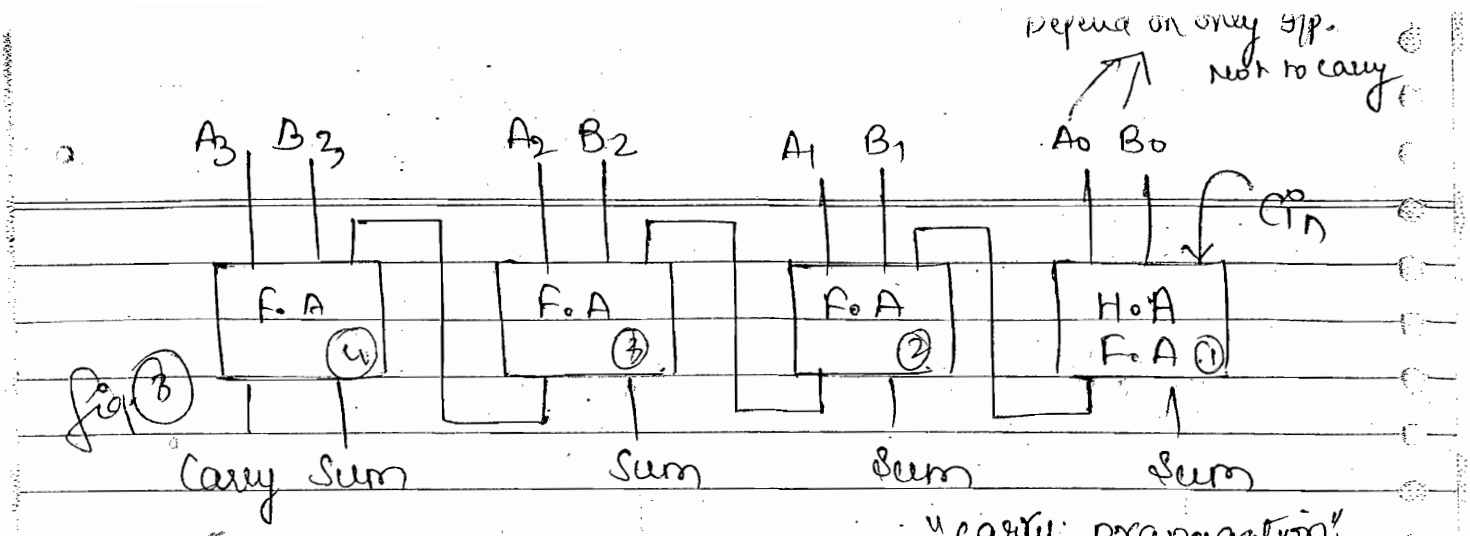
2 H.A + 1 OR gate

fig ①

$A \rightarrow A_3 A_2 A_1 A_0$ $B \rightarrow B_3 B_2 B_1 B_0$

- Ripple Carry Adder OR Parallel Carry Adder:





"carry prapagation"

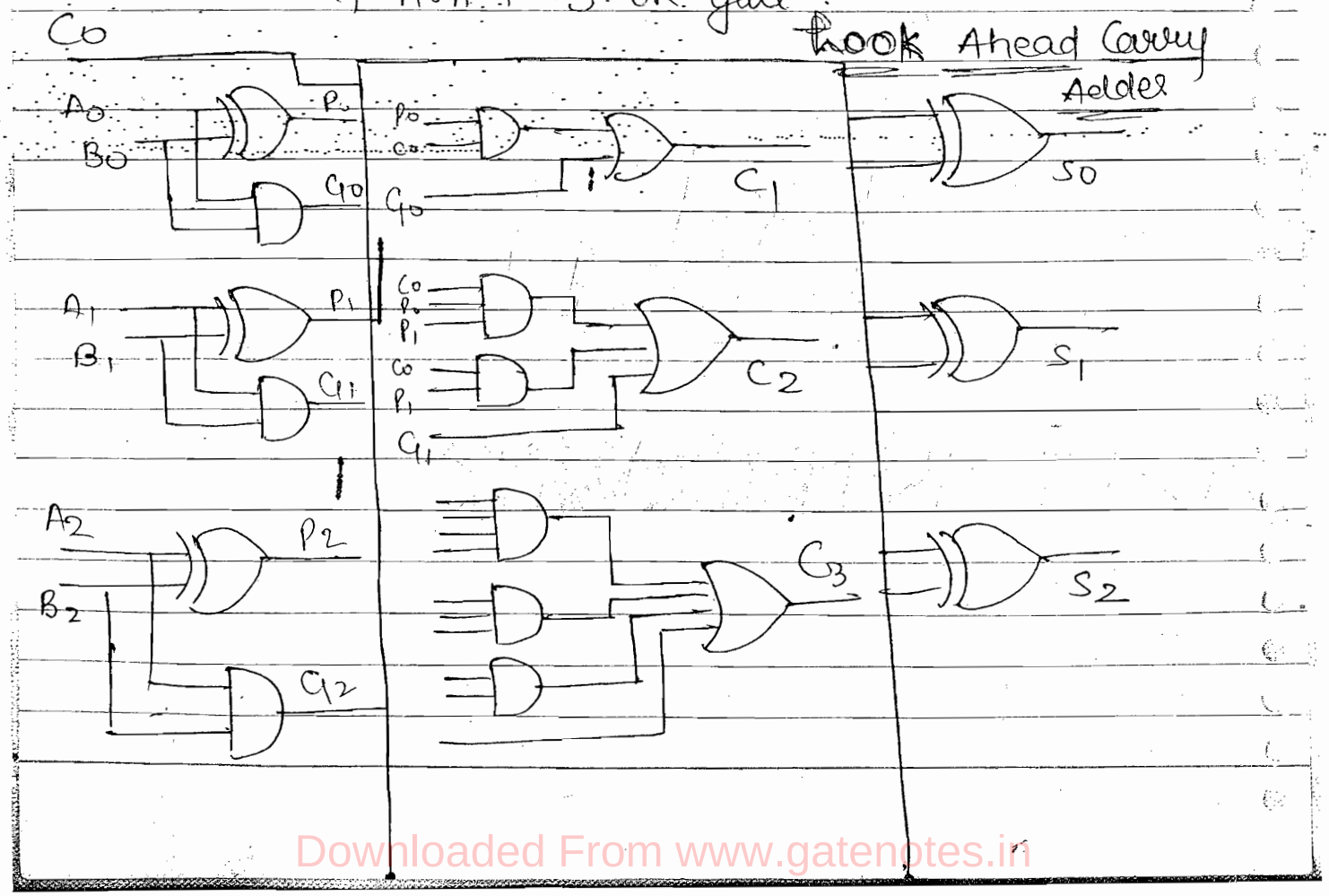
- (N-1) F.A + 1 H.A.
- If Cin present :— N F.A.
  (added with other ckt)

alternate: $(N-1)\left[2\text{ H.A} + 1\text{ OR gate}\right] + 1\text{ H.A.}$ from fig (1)

$\Rightarrow (2N-1)$ H.A. $+ (N-1)$ OR gate

Ex:- N = 4.

7 H.A + 3 OR gate.



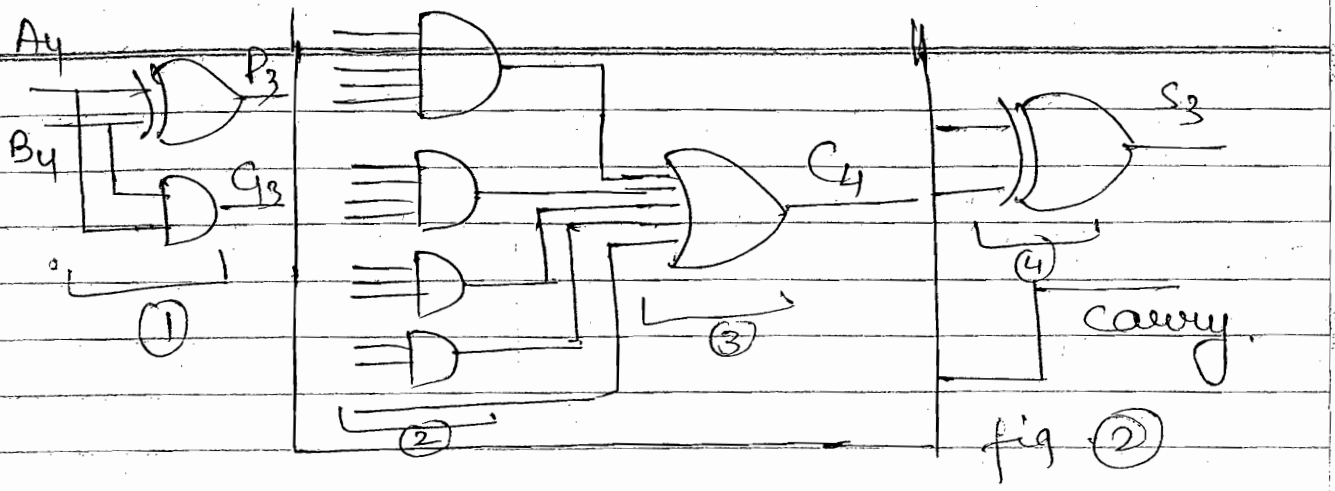


fig (2)

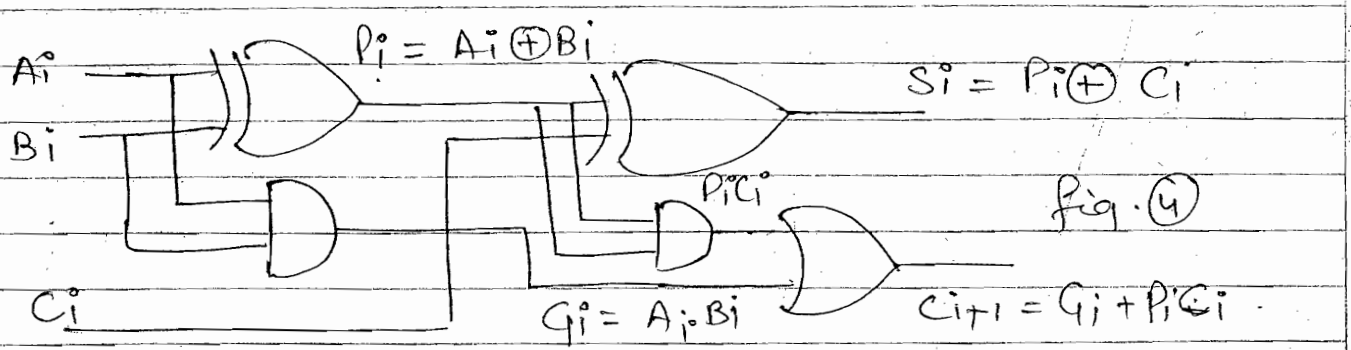


fig. (4)

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

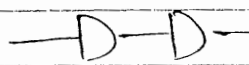
$C_{1+1} = C_2 = G_1 + P_1 C_1 = G_1 + P_1 [G_0 + P_0 C_0] = G_1 + P_1 G_0 + P_1 P_0 C_0$

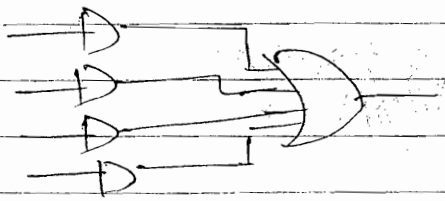
$C_{2+1} = C_3 = G_2 + P_2 C_2 = G_2 + P_2 [G_1 + P_1 G_0 + P_1 P_0 C_0]$

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

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

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

-  = 5 tp delay.

-  = 2 tp. delay.

- fig. (3) HoA having 3 delay. But all three FoA having 2 delay Ai, Bi are already operated in case of FoA.

So total Time delay ⇒ 3+2+2+2 = 9 td delay.

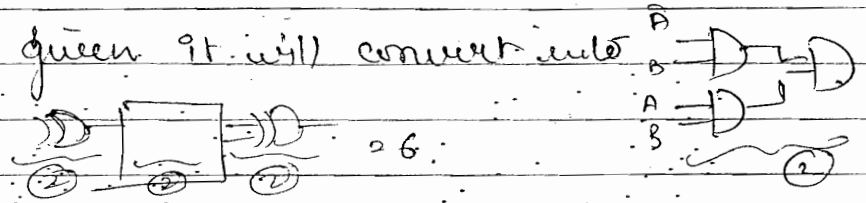
But in fig ② only 4 td delay ① ② ③ ④

✓ ex: fig ⑧ for 10 bit = 2×20 (F.A) = 20
= 20+1 = 21 td delay.
↳ H.A.

✓ But in fig ② only vertical ckt, horizonal remain same even in 10 bit ⇒ 4 td delay.

✓ ckt are individually having carry, all ckt carry individually and add finally By OR gate.

Q.14 X-OR NOR given it will convert into



= 6 = 6.

Q(17) only Box. Total OR gate ⇒ 4 and AND = 10 gate

Shortcut

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

- Compliment Methods :-

	Dec'	Bin'	Oct'	Hexa'
r's ⟶	10's	2's	8's	16's
(r-1)'s ⟶	9's	1's	7's	15's

r → Radix (or) Base

shortcut for 2's compliment:

ex:- 10100 → 01100 2's.

Comes upto Non-zero. write same as it and remaining we get inverted

ex:- 10101 → 01011 → 2's.

ex:- 1000 → 1000 → 2's.

ex:- 9's compliment.

① 9 − 7 = 2 (9's)

② 99 − 17 = 82

③ 99.9 − 17.3 = 82.6

ex:- 10's compliment.

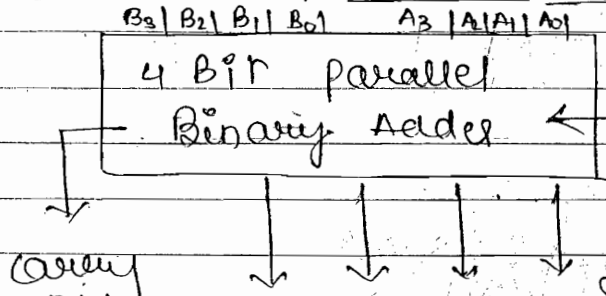
① 2 + 1 = 3

② 82 + 1 = 83

③ 82.6 + 1 = 82.7

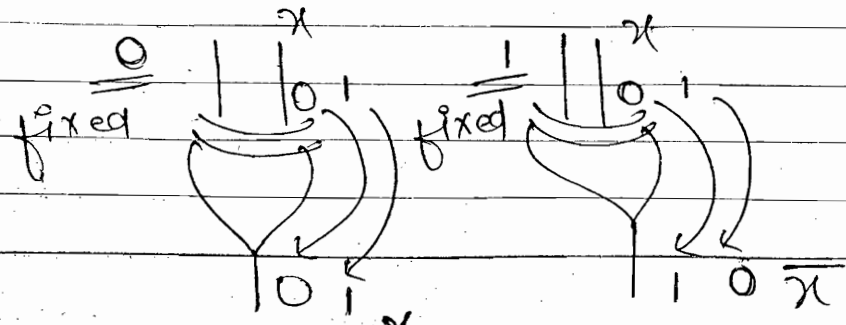
added 1 Must be to LSB

- **2's Compliment Adder/Subtractor**



A → A3 A2 A1 A0

B → B3 B2 B1 B0

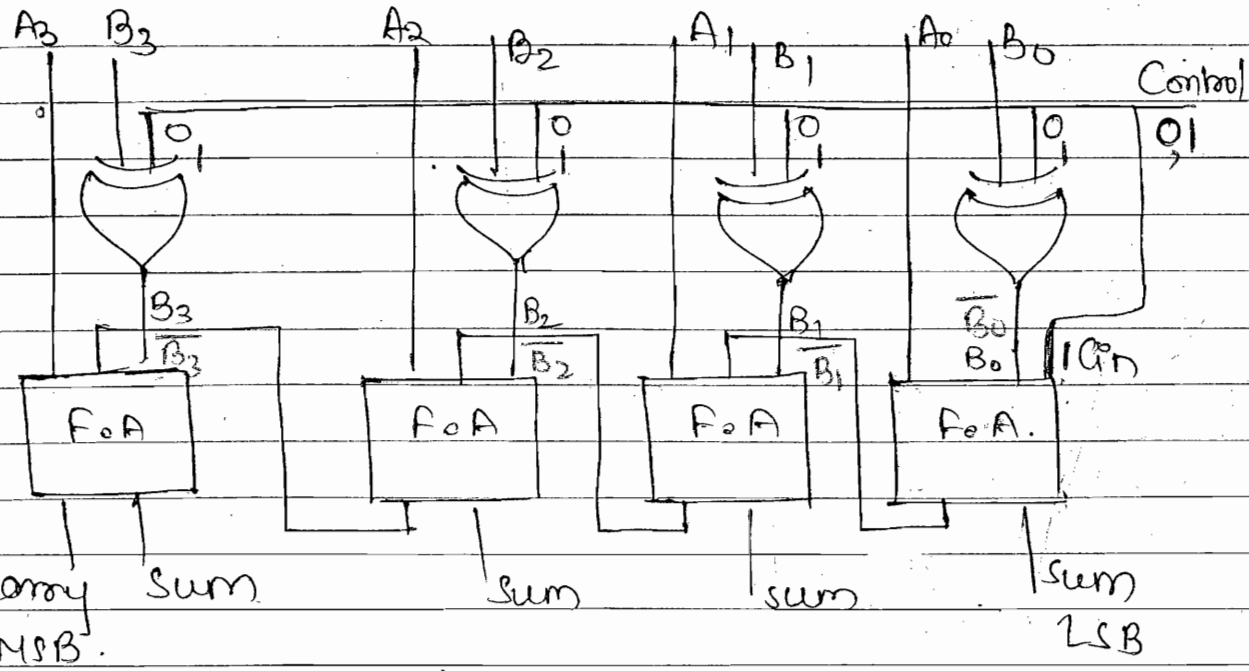


x − y

x ⇒ minuend

y ⇒ subtrahend

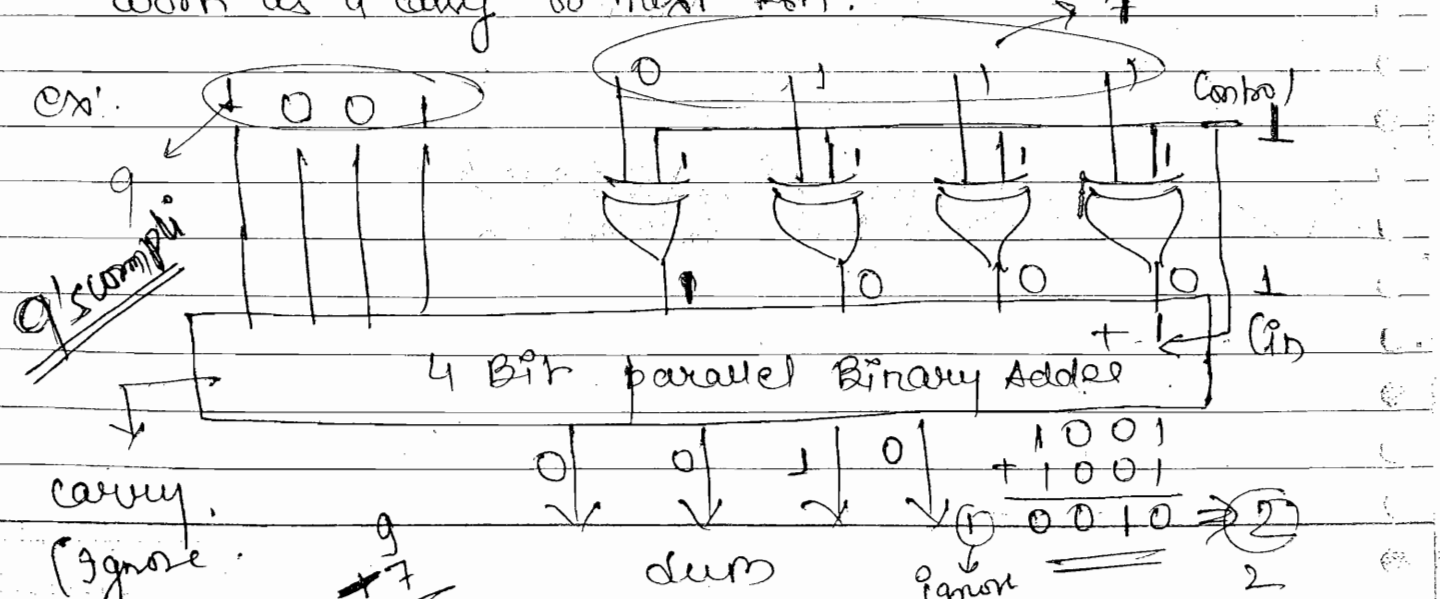
$x - y \Rightarrow x +$ 2's complement of y



Control ⇒ 0 ⇒ A + B
⇒ 1 ⇒ A − B

$B_0 \rightarrow$ 1's complement
$+1 \rightarrow$ 2's complement

When control = 1, $B_0 \Rightarrow \bar{B}_0$ (1's complement) By adding $C_{in} = 1$ $B_0 \Rightarrow$ 2's complement which will added with $A_0 \Rightarrow A +$ 2's complement. This result will work as a carry to next FA.



- **BCD Addition:-**

6 invalid BCD → 10 – 1010, 11 – 1011, 12 – 1100, 13 – 1101, 14 – 1110, 15 – 1111 → add (6)

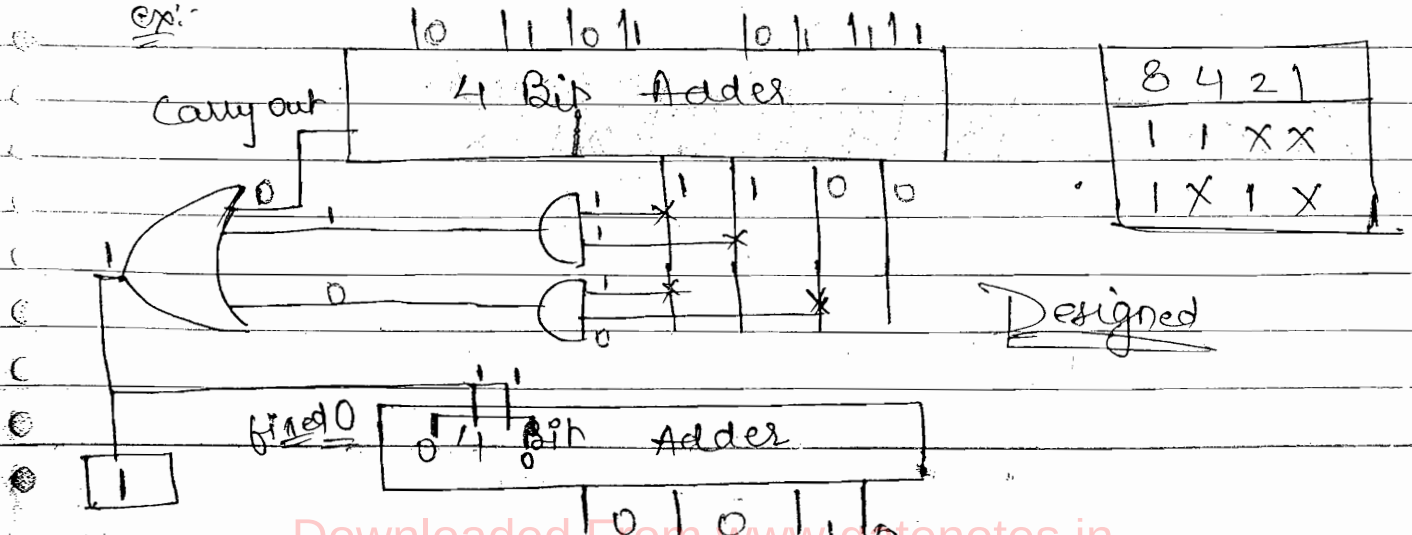
case(i) 5 → 0101
4 → 0100
1001 valid (NO carry).

- When two valid BCD No. are added the result in invalid or carry is generated. The fix should be added to get valid BCD No.

case(ii) 5 → 0101 (Invalid)
7 → 0111
1100 invalid
+ 0110
10001 / 0010
1 2 ⇒ 12.

case (iii) 8 → 1000 (Valid But carry generate)
9 → 1001
carry ← 1 0001 → valid
↓ + 0110
0001 / 0 111
1 7 ⇒ 17.

Ex:



• BCD Subtraction:-

• Case(i) +ve Result

ex 6 → 0 1 1 0

−4 → 0 1 0 1 [9's compli]

1 0 1 1 invalid

+ 1 1 0

(1) 0 0 0 1

end around carry +1

0 0 1 0

+2

⇒ 9's compliment of subtrahend should be added to the minuend If the result is invalid or carry is generated then 'six' should be added with "end around carry".

• caseii) −ve Result

⇒ 9's compliment of subtrahend should be added to minuend if we get valid o/p and No carry which indicated result is −ve and 9's compliment should be taken for final answer.

ex: → 8 → 1

ex:- −ve Result −6 → 0 0 1 1 (9's compliment)

+ 4 → 0 1 0 0

0 1 1 1 valid

"No carry"

1> valid
2> No carry } Result −ve

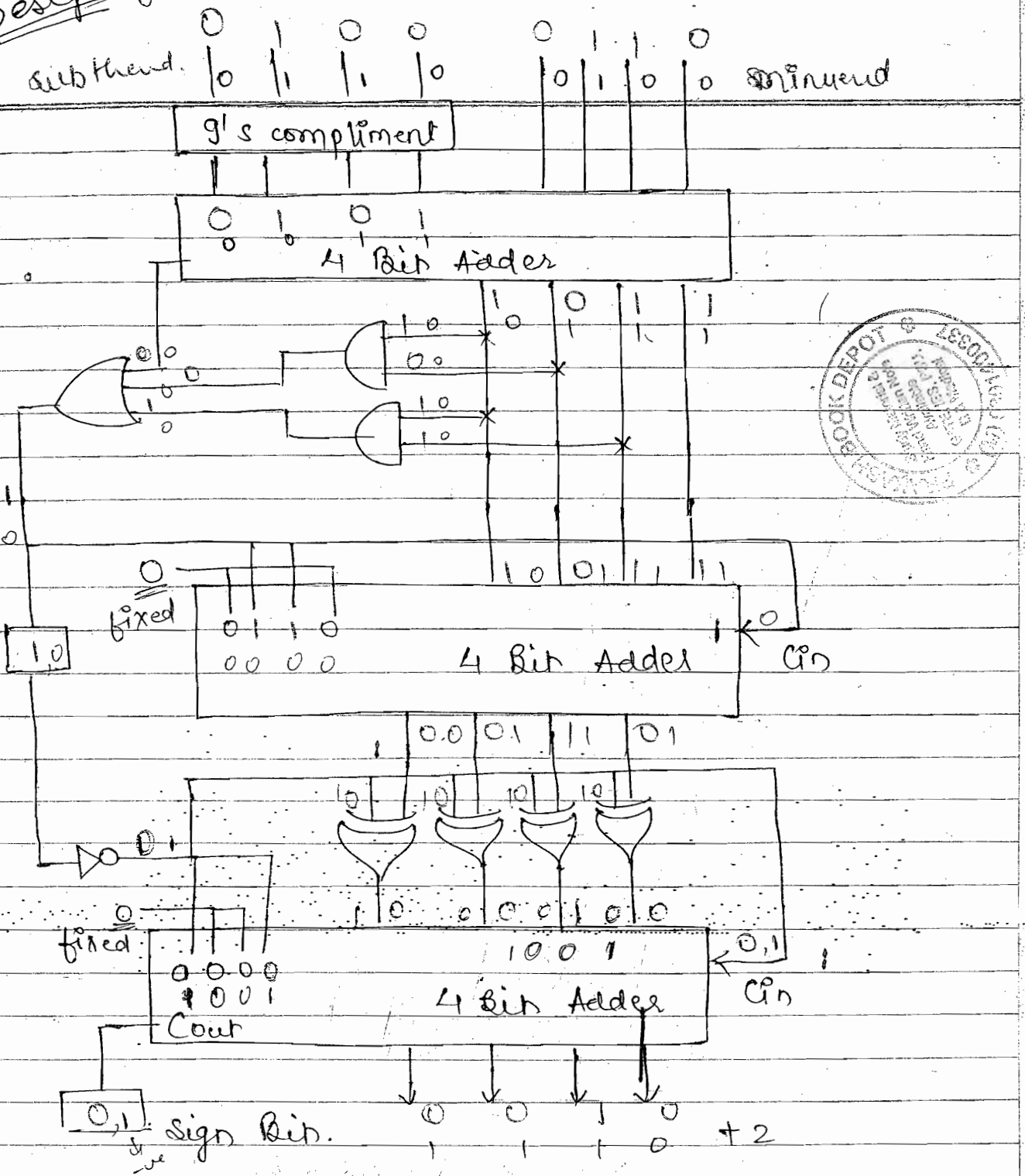
Ans ⇒ Take 9's

9

−7

−2

Designing



◉ 2's Complement Arithmetic :-

Step 1 :- 2's compliment of subtrahend is add to the minuend.

Step 2 :- Check the MSB Bit. (Sign Bit)

Step 3 :- If the sign Bit is one take 2's compliment

for the final answer. If the sign Bit is zero Answer is already in the true Binary.

Step 4:- Ignore the carry.

case(i) $+9 \rightarrow 0\,1\,0\,0\,1$
$+4 \rightarrow 0\,0\,1\,0\,0$
$0\,1\,1\,0\,1 \Rightarrow +13$.
sign Bit

case(ii) $+9 \rightarrow 0\,1\,0\,0\,1$
$-4 \rightarrow 1\,1\,1\,0\,0$ (2's compliment of 4)
$1\;\; 0\,0\,1\,0\,1 \Rightarrow +5$
carry (Ignore)

case(iii) $-9 \rightarrow 1\,0\,1\,1\,1$ [2's compli of 9]
$+4 \rightarrow 0\,0\,1\,0\,0$
$1\,1\,0\,1\,1$
sign Bit
2's compliment.
$1\,0\,1\,0\,1 \Rightarrow -5$

case(iv) $-9 \rightarrow 1\,0\,1\,1\,1$
$-4 \rightarrow 1\,1\,1\,0\,0$
$1\;\; 1\,0\,0\,1\,1$
carry
sign Bit
$1\,1\,1\,0\,1 \Rightarrow -13$

case (v) :- $-9 \rightarrow 1\,0\,1\,1\,1$
$+9 \rightarrow 0\,1\,0\,0\,1$
$1\;\; 0\,0\,0\,0\,0 \Rightarrow +0$
carry Ignore
sign Bit

- **1's compliment : Arithmetic.**

Step1 :- Add the 1's compliment subtrahend to the minuend. If the carry occures make it end around carry.

Step2 :- Check the MSB Bit (Sign Bit). If it is 0 Answer is True Binary. If it is 1 take 1's compliment for the final answer.

- **Overflow Condition :-** The carry enter to sign Bit it is known as overflow condition. Then sys. introduces one more bit for no. representation.

5 → x0 1 0 1
4 → y0 1 0 0
Cin ↓
Cout ← 1 0 0 1

5 → 0 0 1 0 1
4 → 0 0 1 0 0
1 0 1 0 0 1
+9

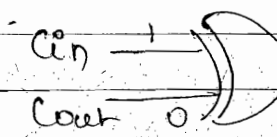
overflow condition : $f = \bar{x}\bar{y}z + xy\bar{z}$

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

$= 1$ (overflow)

if $f = 0$ No overflow
$f = 1$ overflow.

2nd Method :-



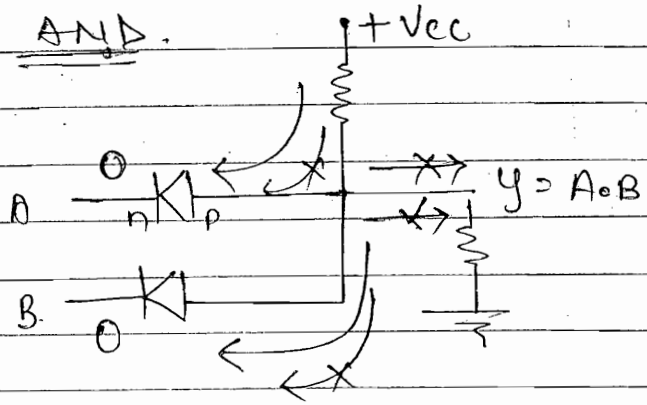
if $f = 0 \rightarrow$ No overflow
$f = 1 \rightarrow$ overflow.

Cin → Carry 1+1 ⇒ 10
Cout → Carry coming out By 1 ⇒ 0.

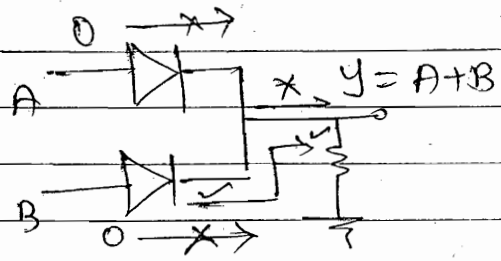
Logic Family (Theory Book Parameter Table Imp)

- Diode has logic gate.

AND.



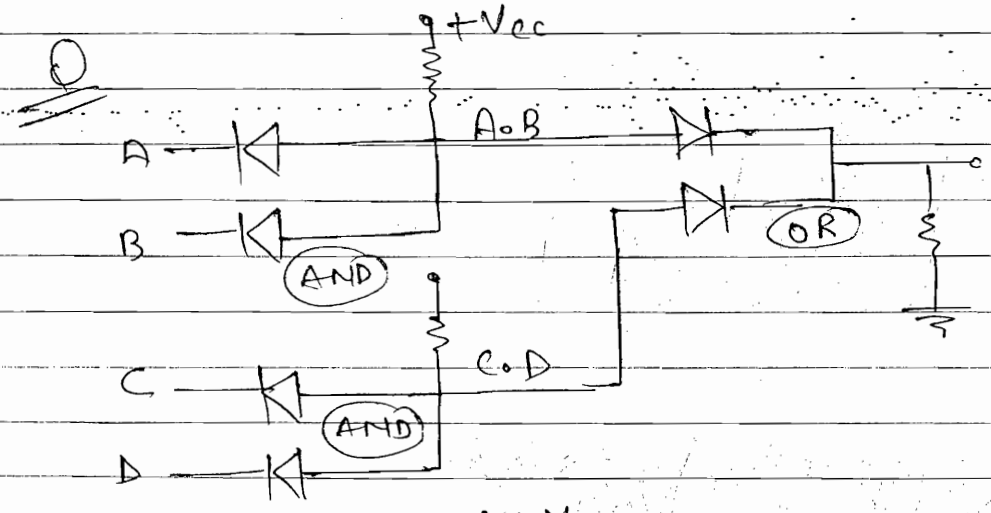
OR



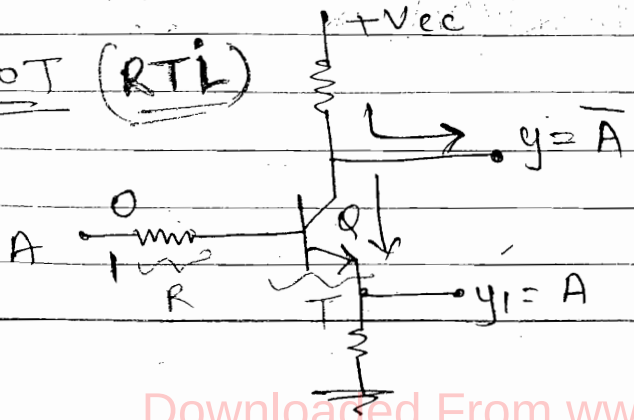
A	B	D_1	D_2	y
0	0	ON	✓	0
0	1	✓	X	0
1	0	X OFF	✓	0
1	1	X	X	1

A	B	D_1	D_2	y
0	0	0	0	0 GND
0	1	0	1	1
1	0	1	0	1
1	1	1	1	1

Q

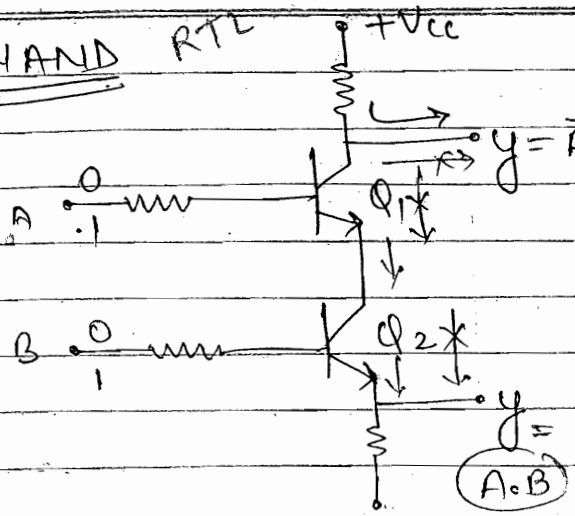


- NOT (RTL)



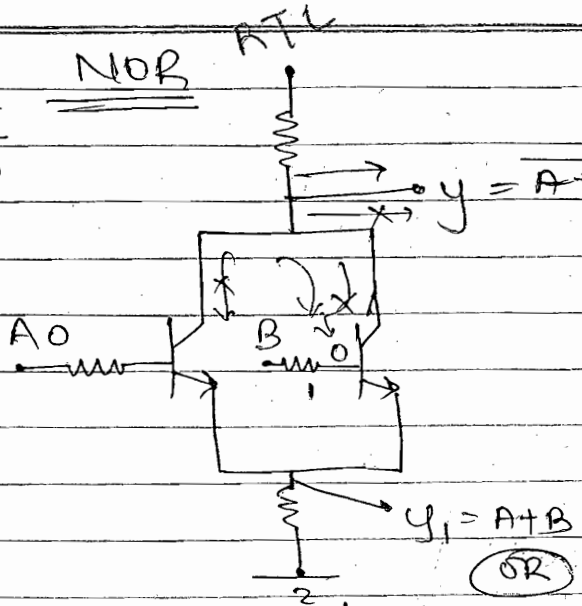
A	φ	y
0	X	1
1	✓	0

NAND RTL



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

NOR RTL



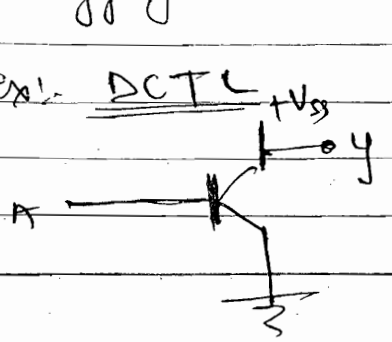
A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

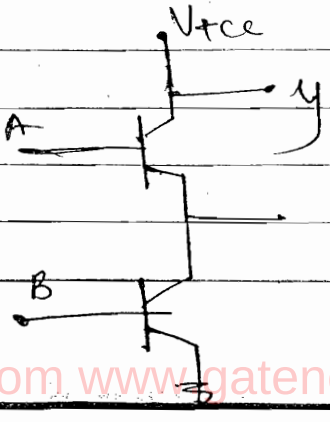
- DCTL :- Direct Coupled Transistor logic
  It is same as that of RTL except the I/p resistor are removed

NOTE :- DCTL suffers with current Hogging problem

Current Hogging :- Because of different saturation level of loading gates current will be hogged in certain loads only, and remaining loads are starvation of current, known as current hogging

Ex :- DCTL

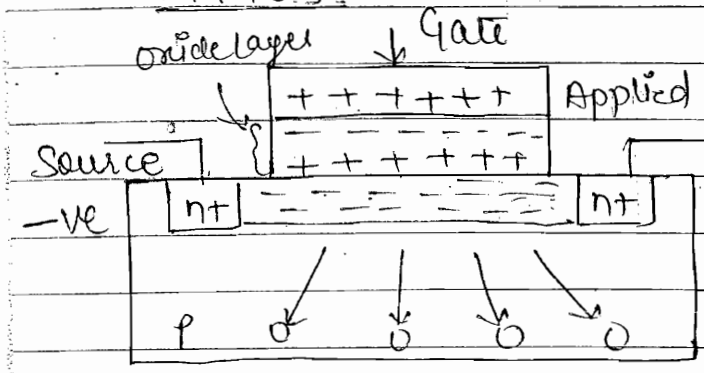




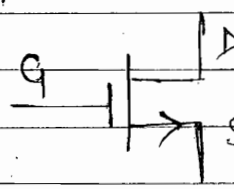
Conventional Diagr. Truth Table (15) Marks

MOS Technology

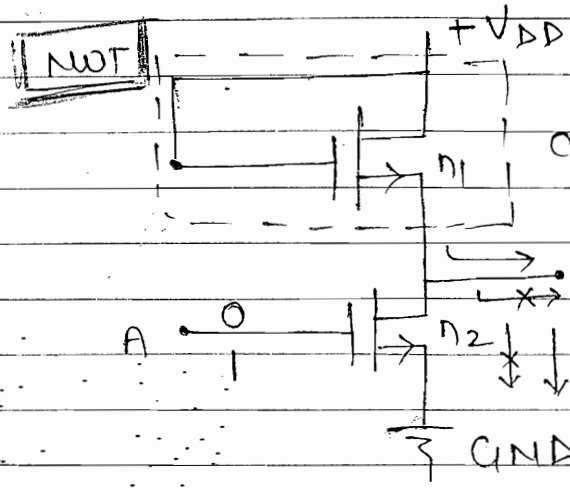
- **NMOS :-**



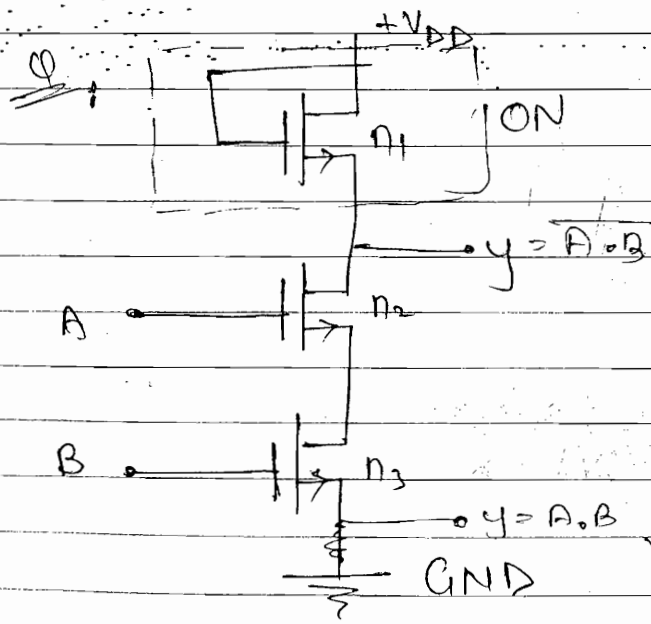
NMOS :- ⇒ Gate
+ve ⇒ ON
-ve ⇒ OFF



Symbol



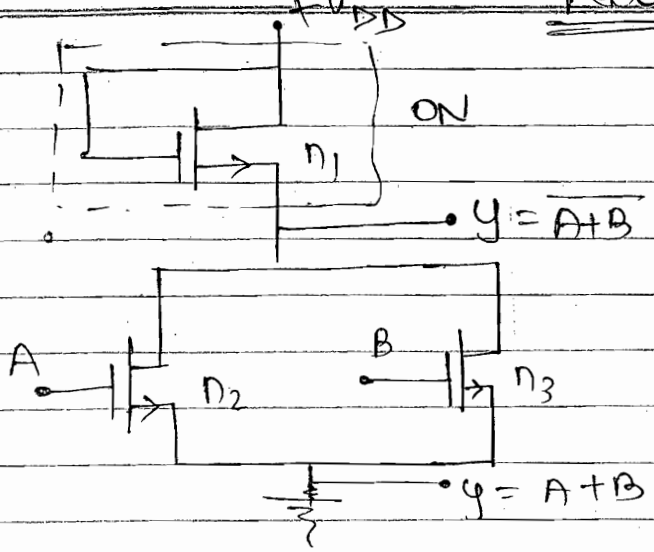
A	n_1	n_2	y
0	✓	X	1
1	✓	✓	0



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

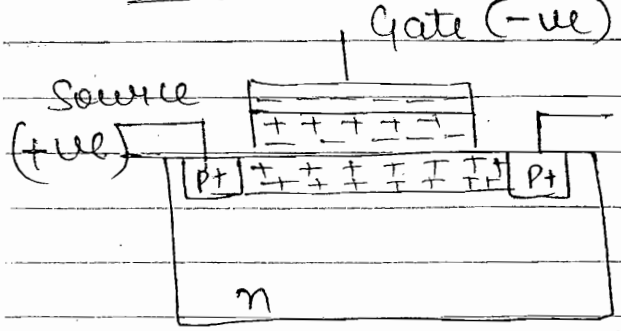
NAND Gate

"NOR" Gate



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

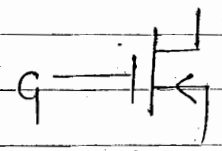
- **PMOS :-**



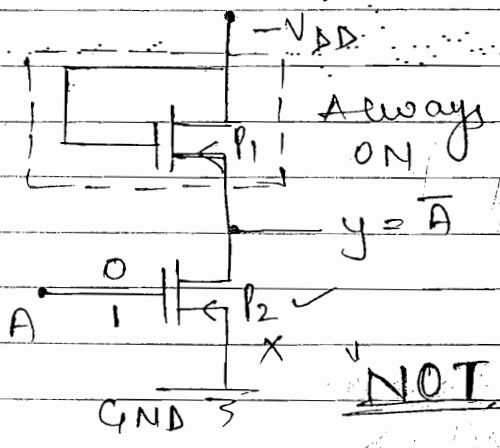
PMOS :- Gate.

+ve ⇒ OFF

-ve ⇒ ON

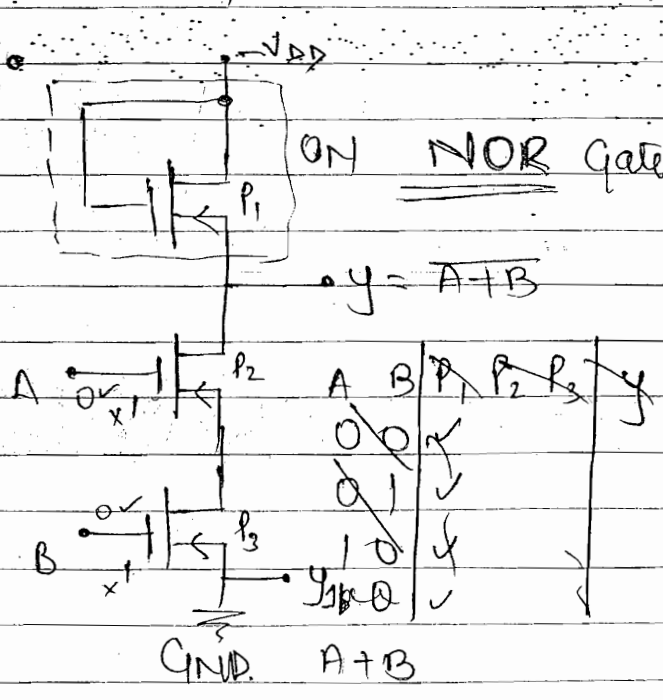


Symbol



"NOT"

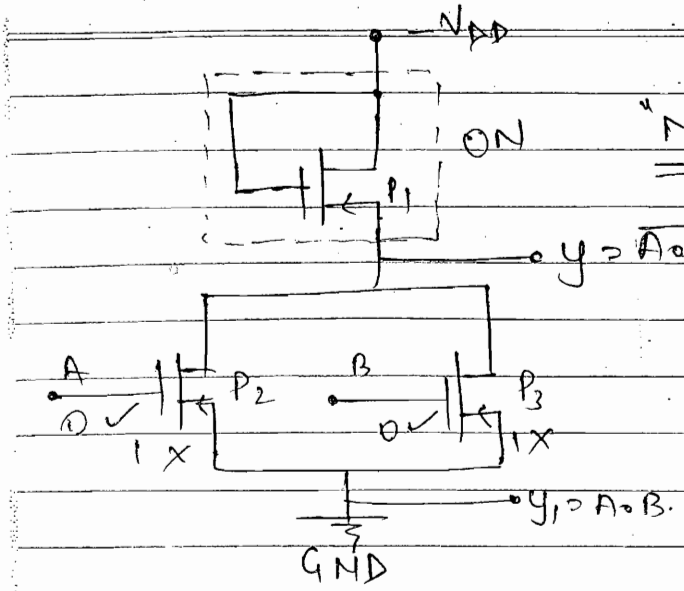
A	P_1	P_2	y	
0	✓	✓	1	GND
1	✓	X	0	-VDD



A	B	P_1	P_2	P_3	y
0	0	✓	✓	✓	1
0	1	✓	✓	X	0
1	0	✓	X	✓	0
1	1	X	X	X	0

NOR Gate Table

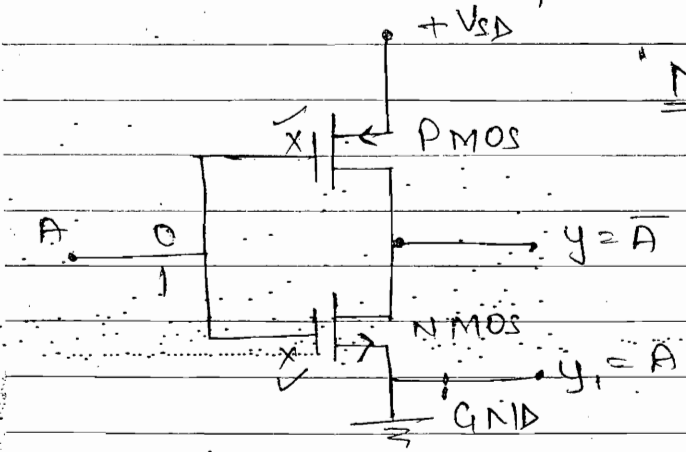
A	B	P_1	P_2	P_3	y	
0	0	✓	✓	✓	1	GND
0	1	✓	✓	X	0	$-V_{DD}$
1	0	✓	X	✓	0	$-V_{DD}$
1	1	✓	X	X	0	$-V_{DD}$



"NAND" Gate

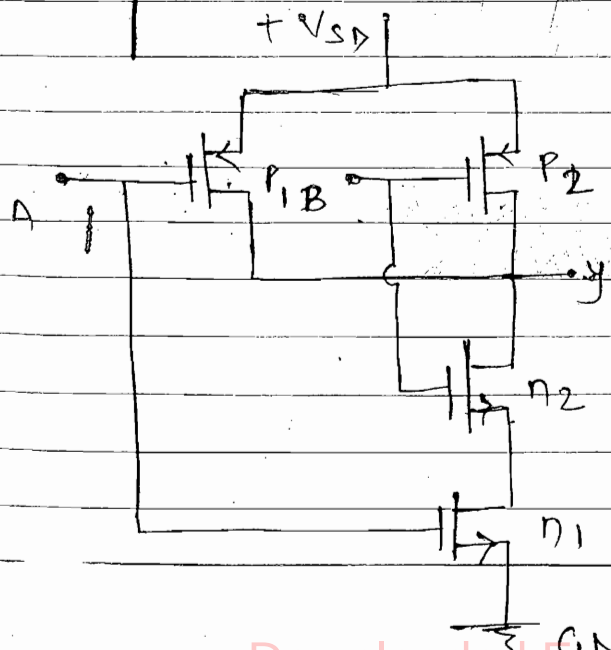
A	B	P_1	P_2	P_3	y	
0	0	✓	✓	✓	1	GND
0	1	✓	✓	X	1	GND
1	0	✓	X	✓	1	GND
1	1	✓	X	X	0	$-V_{DD}$

- **CMOS:- Complimentary MOS:-**



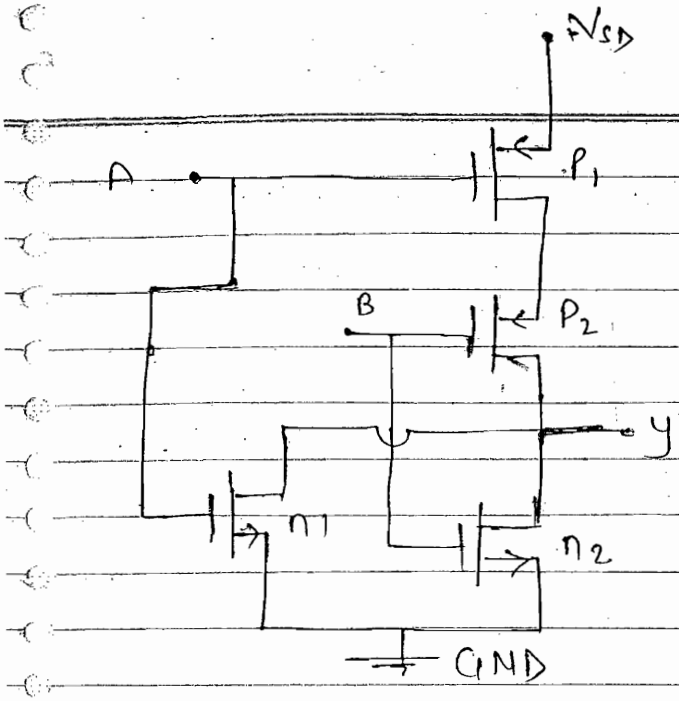
"NOT" Gate

A	P	N	y	
0	✓	X	1	$+V_{SD}$
1	X	✓	0	GND.



"NAND" Gate

A	B	P_1	P_2	N_1	N_2	y	
0	0	✓	✓	X	X	1	$+V_{SD}$
0	1	✓	X	X	✓	1	$+V_{SD}$
1	0	X	✓	✓	X	1	$+V_{SD}$
1	1	X	X	✓	✓	0	GND



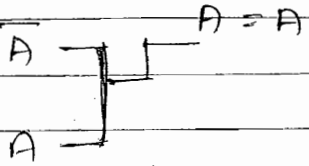
A	B	P_1	P_2	N_1	N_2	Y	
0	0	✓	✓	X	X	1	V_{SD}
0	1	✓	X	X	✓	0	GND
1	0	X	✓	✓	X	0	GND
1	1	X	X	✓	✓	0	GND

Shortcut:-

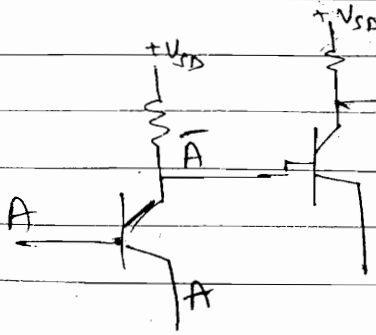
- (N-p-n) OR NMOS:-
  - (i) column → ~~AND~~ Multiplication → AND operation
  - (ii) Row → OR operation → Addition.

- (p-n-p) — OR PMOS:-
  - (i) Column → OR operation → Addition
  - (ii) Row → AND → Multiplication

For Both:- Bar → Non Bar → Bar Bar

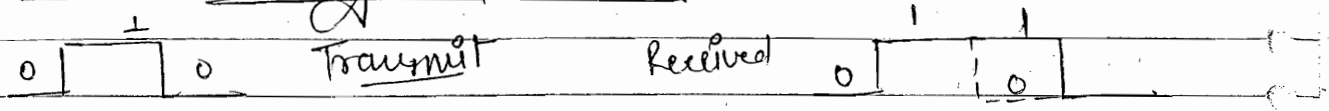


- CMOS:- Concentrate on N-MOS (and ans)



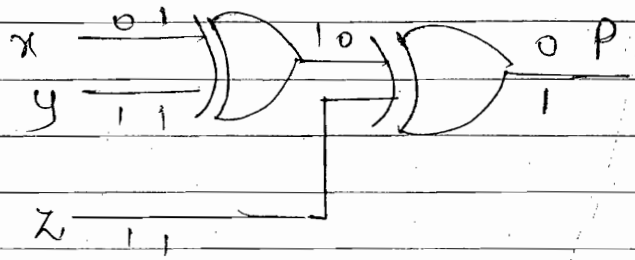
Q. logic family

• Even Parity Generator



Parity Even parity. 0 1 0 [1] (P)

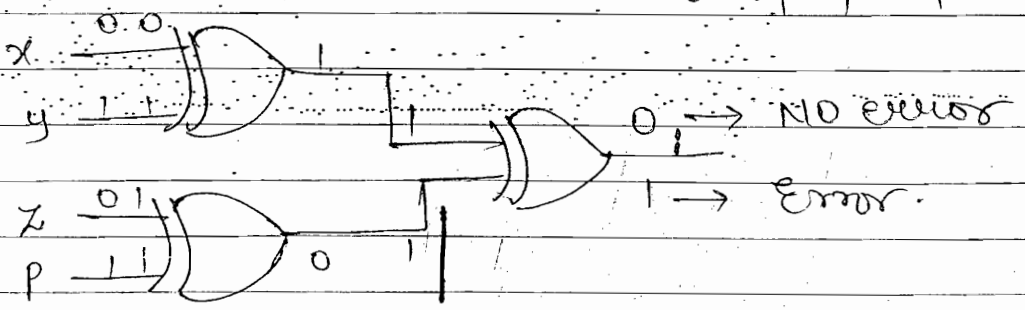
x	y	z	p
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



→ No. of 1's including parity Must be even No. ⇒ No error.

• Even Parity Checker

x y z p
0 1 1 1



• Hamming Code :- (7 Bit)

Data → 1 1 0 1

Tx →	P_1	P_2	P_3	P_4	P_5	P_6	P_7
	1	0	1	0	1	0	1

for 7 Bit → Parity Bits are ⇒ P_1 P_2 P_4
12 Bit → " " ⇒ P_1 P_2 P_4 P_8
15 Bit → " " ⇒ P_1 P_2 P_4 P_8
rest are data Bits.

	P_4	P_2	P_1
	2^2	2^1	2^0
	x	y	z
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

To make even parity ⇒ 1's No. even

P_1 (1,3,5,7) ⇒ (P_1 1 1 1) ⇒ $P_1 = 1$

P_2 (2,3,6,7) ⇒ (P_2 1 0 1) ⇒ $P_2 = 0$

P_3 (4,5,6,7) ⇒ (P_3 1 0 1) ⇒ $P_3 = 0$

Rx (Received) 1 0 0 0 1 0 1

P_1 P_2 P_3 P_4 P_5 P_6 P_7

P_1 (1,3,5,7) ⇒ (1 0 1 1) ⇒ $\gamma = 1$

P_2 (2,3,6,7) ⇒ (0 0 0 1) ⇒ $\beta = 1$

P_4 (4,5,6,7) ⇒ (0 1 0 1) ⇒ $\alpha = 0$

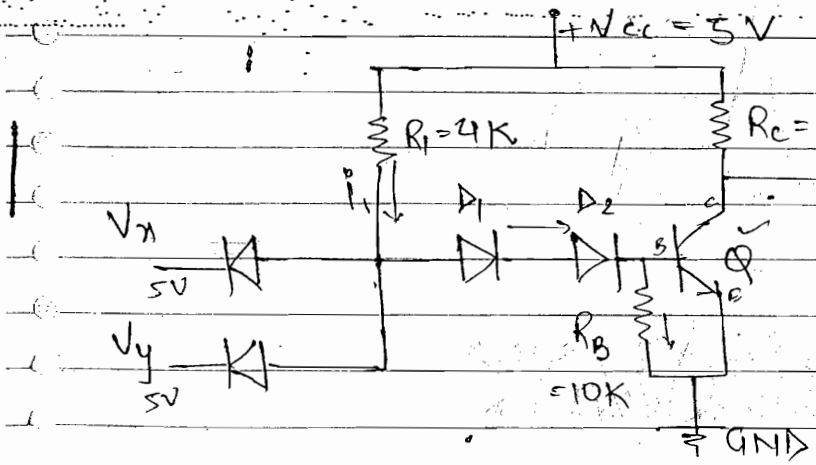
$\alpha\ \beta\ \gamma$

0 1 1 ⇒ (3) → 3 Bit having error

If No error $\alpha\ \beta\ \gamma$ = 0 0 0

29/9/2014

- **DTL**:- (Diode Transistor Logic):-



NAND Gate

Vx	Vy	V
0	0	1
0	1	1
1	0	1
1	1	0

Ckt Analysis:-

Give $V_\gamma = 0.7$ V Cut off voltage of Diode

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

$V_{BE(sat)} = 0.8$ V $V_{CE(sat)} = 0.1$ V

$\beta = 25$

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

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

$i_2 = I_B = I_R = 0 \Rightarrow Q$ is off

$\Rightarrow y =$ High (5V)

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

Same as the of case(i) $\Rightarrow y =$ High (5V)

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

Same as the case (ii) $\Rightarrow y =$ High (5V).

case(iv) $V_x = 5V \quad V_y = 5V$

$i_1 = \frac{5 - 2.2}{4K} = 0.7\,mA$

$i_2 = \frac{0.7\,mA}{1}$ $\quad i_R = \frac{V_{BE(sat)}}{10} = \frac{0.8}{10} = 0.08\,mA$

$i_R = 0.08\,mA$

$I_B = i_2 - i_R = 0.7 - 0.08 = 0.62\,mA$

$i_c = \frac{V_{cc} - V_{CE(sat)}}{R_C} = \frac{5 - 0.1}{4K} = 1.2\,mA$

$I_B = 0.62\,mA \quad I_c = 1.2\,mA$

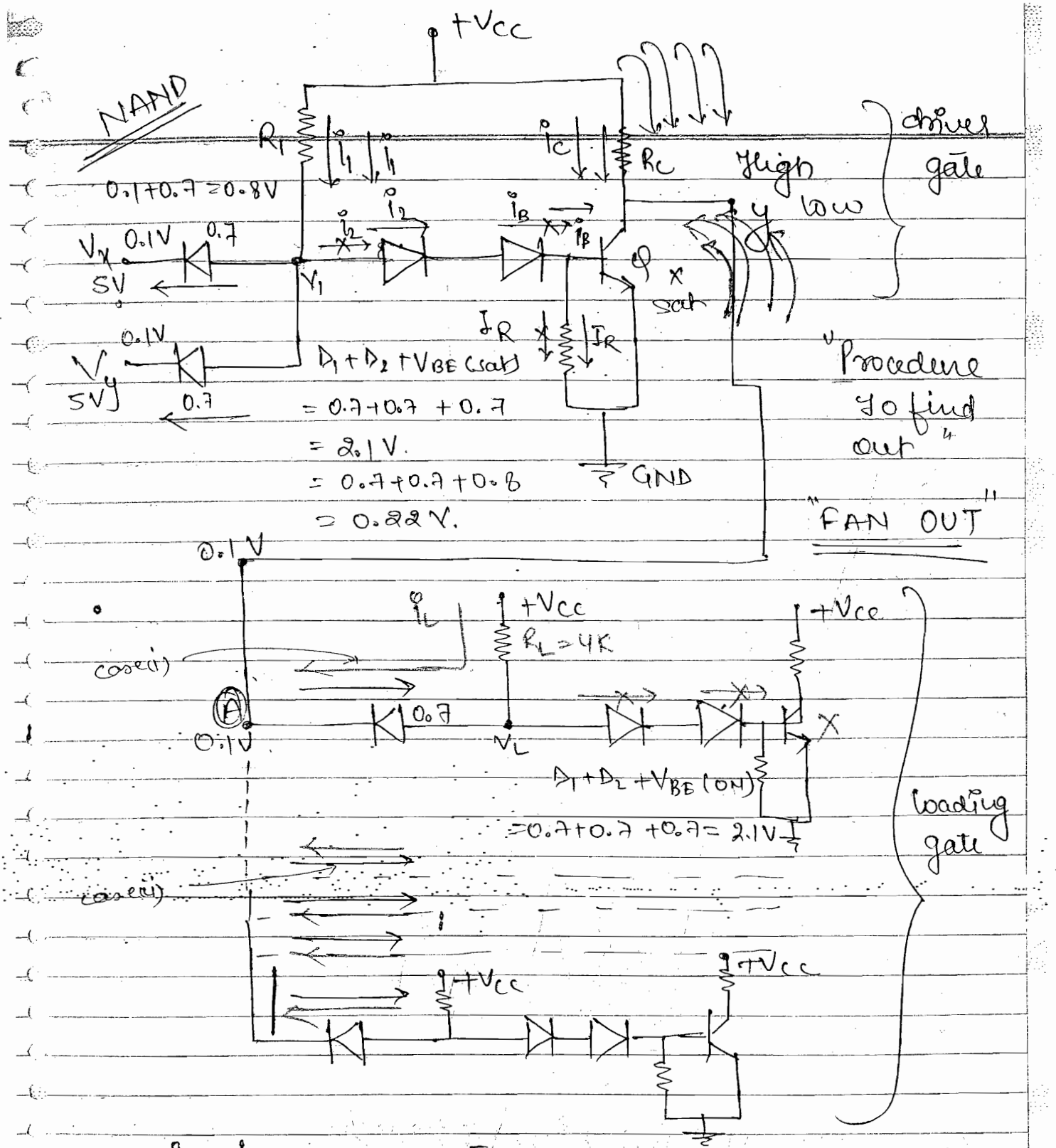
Saturation condition $\frac{i_c}{I_B} \leq \beta \quad \frac{1.2}{0.62} < 25, \; 1.9 < 25$

$\Rightarrow Q$ is in saturation region.

At the saturation condition is necessary because the more voltage will occur at $R_c = 4K$ by large value of i_c, so low voltage at y.

- Drive $\leftarrow$ connected and loaded ckt.

Find the no. of load ckt can drive



$$i_L = \frac{V_{cc} - V_L}{R_L} = \frac{5 - 0.8}{4K} = 1.05\,mA$$

$i_C' = i_C + N I_L$

Saturation condition $i_C' / i_B \leq \beta$.

$\Rightarrow N = 13$

Valid for this values only

O/p low case of driving is ⟶ Sinking
O/p high case of driving is ⟶ Sourcing.

$$\frac{I_c + N I_L}{I_B} = 25 \qquad (N \simeq 13)$$

$$\frac{1.2 + N(1.05)}{0.62} = 25$$

- When the o/p low → all load ckt gate get ON. So all the load gate draw the current which is coming from V_{cc} = 5V (driver ckt)

- Procedure :-

Case (i) O/p of driving gate is low :- In this case the voltage drop V_L for the loading gate when becomes as 0.8V then exist a load current which passes through the saturated Transistor of driving gate. Then for n No. of loads we get $I_c' = I_c + N I_L$ and the saturation condition will be $\frac{I_c'}{I_B} \leq \beta$ i.e. $\frac{I_c + N I_L}{I_B} \leq \beta$.

where N indicates the "FAN OUT" value.

Case (ii) O/p of driving gate is high :- In this o/p diode of loading gate is reversed Biased and then exist a reverse saturation current which passed through the Rc of driving gate. And by N No. of loaded it causes more voltage drop across Rc and o/p can alter from high voltage which leads to improper operation

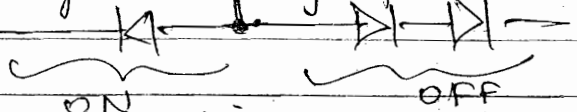
- So their is a limit on the fan out even in this case also.

NOTE:- From the among two cases the least value is preferrable for overall fan out.

So O/p low case of driving gate is preferrable.

- No. of load gate can be connected more in case high because minority current is very low compare to low case.

- At pt A voltage can vary upto 1.3V bcoz. 1.3+0.7 = 2.0V



This section will ON. (proper function)

⇒ This called Noise margin.

- Procedure to find Noise Margin:-

case(i) O/p of driving gate is low:- In this case the other section of loading gate to become ON the voltage drop required at V_i is $D_1 + D_2 + V_{BE(ON)} = 0.7 + 0.7 + 0.7 = 2.1V$

So the difference of voltage = 2.1 − 0.8 = 1.3 V

is the maximum acceptable floating voltage known as 'NOISE MARGIN'.

case(ii) O/p of driving gate is high:- In this case also ~~fan out~~ should be measured.
Noise Margin

NOTE:- Among above two case ~~Noise margin should be measured~~ as least value is preferrable of Noise Margin.

- Procedure to find Power dissipation:-

$$P_D = V_{cc} \cdot I_{cc} = \frac{P_D(\text{low}) + P_D(\text{high})}{2}$$

for high – OFF Transistor. $i_1 = i_{cc}$
for low – ON – transistor $i_{c1} + i_2 = i_{cc}$.

- D_1 & D_2 If only one diode are removed, fanout & Noise Margin effects.
FAN OUT value will ↑, Noise Margin ↓

- If add one More diode D_x, D_1 & D_2 fanout & Noise Margin value effect. Fanout value ↓
Noise Margin ↑

- As per client Requirement of fanout and Noise, we select the diode value and No. of diodes as per ckt value.

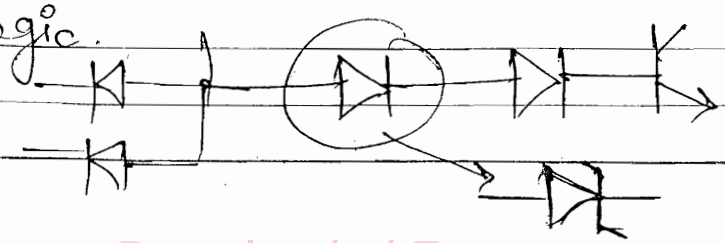
Q What happen when one of diode is Removed in DTL ckt:

Ans FAN OUT ↑, NOISE MARGIN ↓

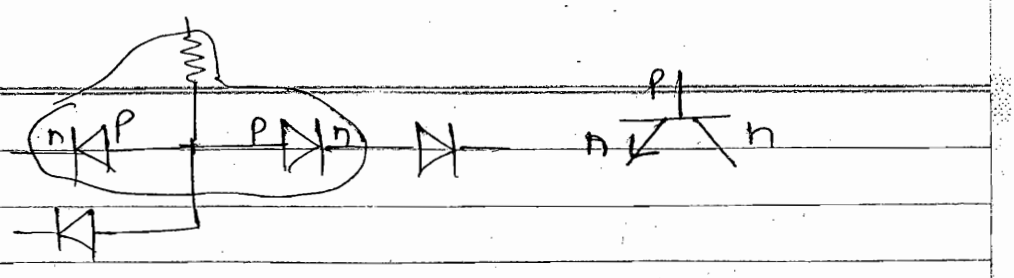
Q What happen when one more diode is Added.

Ans FAN OUT ↓, NOISE MARGIN ↑

- HTL: High Threshold Logic. When one of diode is replaced by zener diode, Noise Marginal levels can be ↑ known as high threshold logic.



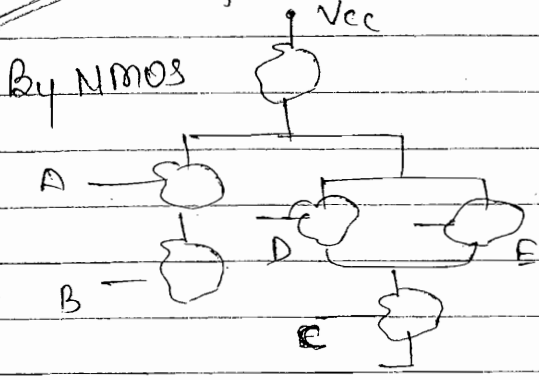
• TTL:-



procedure will remain same only values get change.

Q IES Q. Design the CKT NOSFET By using NMOS:- (5 Marks)

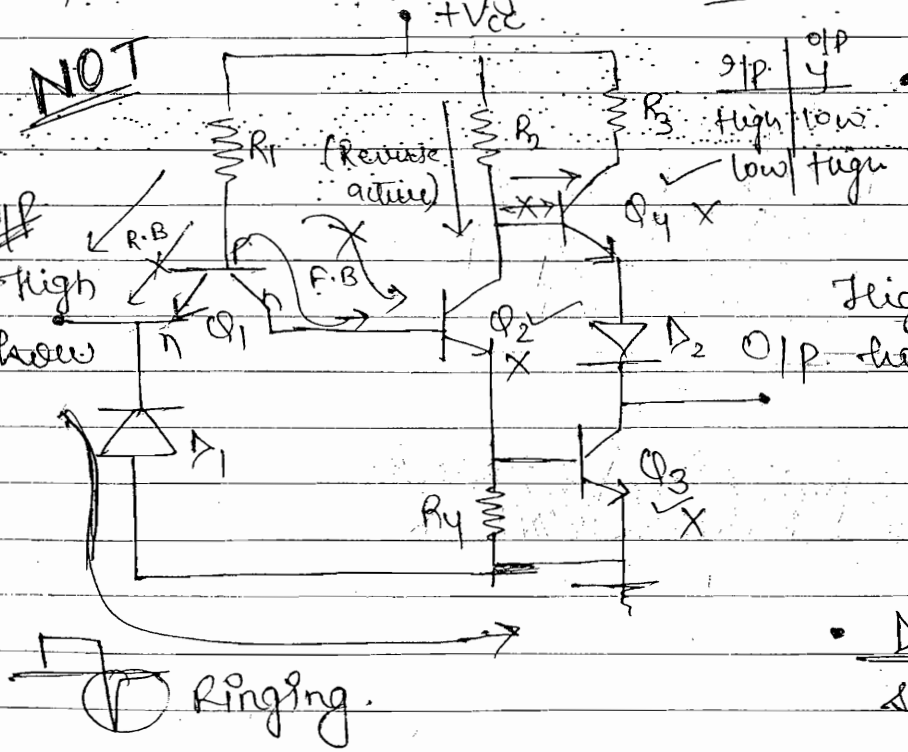
$f = \overline{AB + C(D+E)}$



By NMOS

By CMOS. NMOS design remain same, for PMOS again draw the ckt above NMOS by row ⇌ column

Q) Draw the diagram:- TTL



I/p	O/p Y
high	low
low	high

• D_1 is used for when the -ve voltage occure at the I/p. High | D_1 get ON and O/p low ground. So to ringing problem can be solved

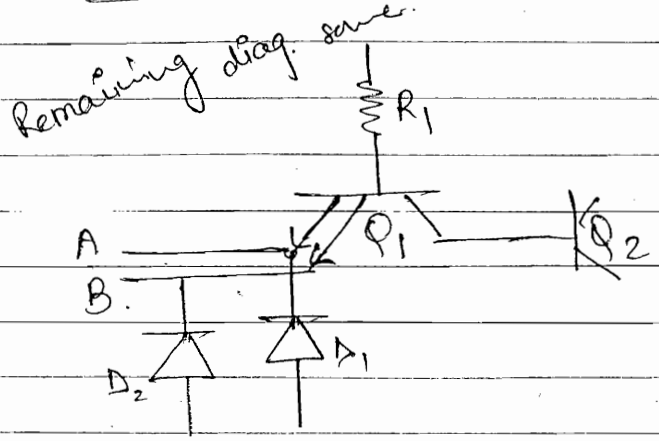
• D_2 → Used for switching

→ When the I/p is high n-ch is R.B, any fluctuation occure not problem.

- Purpose of D_1 :- It avoids the ringing i.e the -ve voltage fluctuation can be grounded by the i/p diode D_1 so that the i/p. transistor is saved

- Purpose of D_2 :- For proper switching.

- TTL NAND Gate.



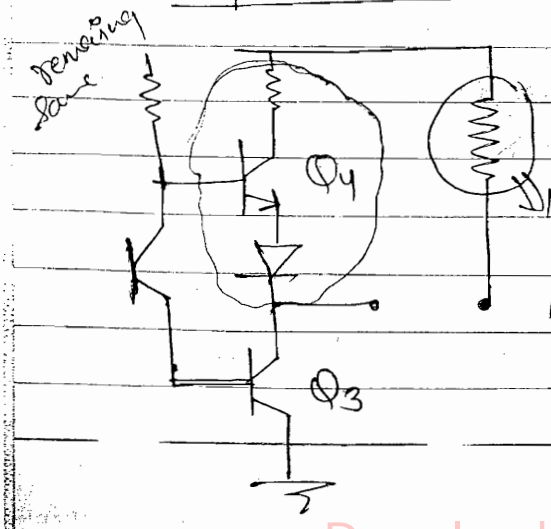
More i/p, it become NAND, when atleast i/p is low, o/p is high.

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

- TTL logic are three types :-

(i) Totempole TTL :- The above discuss TTL is known as totempole TTL. Bcoz Both o/p transistor cannot be ON simultaneously (Q_3, Q_4)

(ii) Open Collector TTL :- The o/p transistor collector end is open and resistor is connected for proper operation known as open collector TTL.

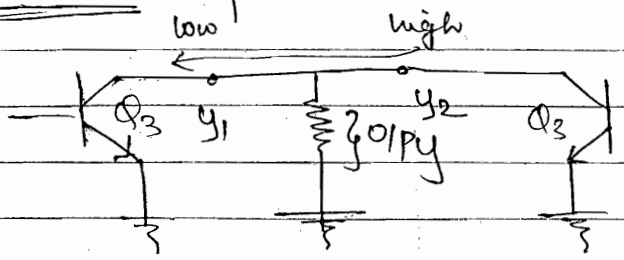


Purpose of Pull up Resistor :-

① for proper o/p switching voltage.

② It eliminates power supply fluctuation.

* NOTE Open collector TTL allows wired logic.



← This Q_3 is same (another Q_3 taken).

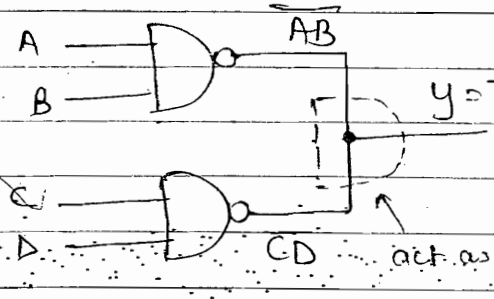
→ y_2 will give to y_1, Not left (current) for y. So $y \geq 0$.

→ When both $y_1 = y_2 = 1$ O/p $y > 1$

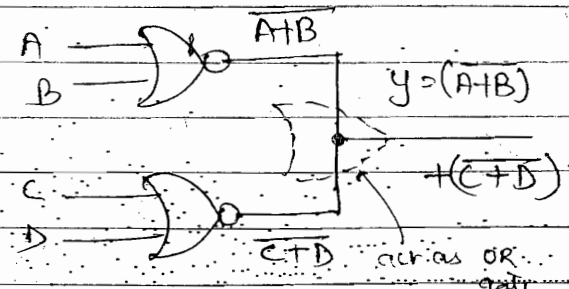
So called wired AND logic

y_1	y_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Wired AND logic



Wired OR logic

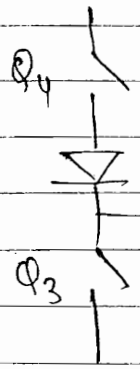


(iii) Tri state TTL: Having three state

(i) logic 0

(ii) logic 1

(iii) High Impedance state



Both transistor can be OFF Together By some external mean.

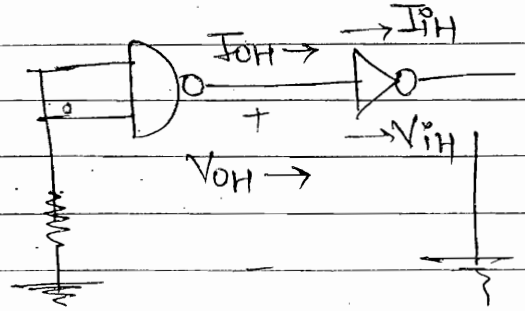
→ When current Driving → load ⇒ Sourcing

— " — load → Driving ⇒ Sinking

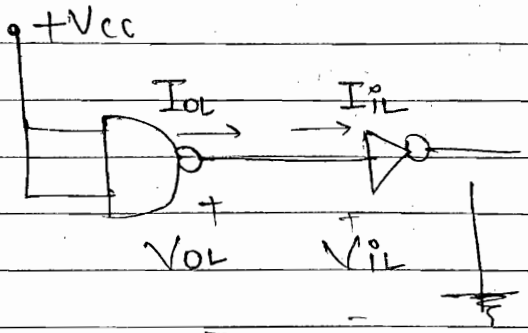
Question comes to find fanout value.

IES Voltage and current parameter :-

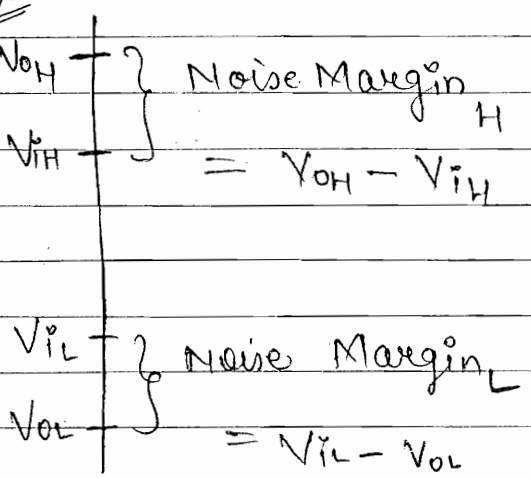
High



low.



graph



$$\text{Fan Out} = \frac{I_{OH}}{I_{iH}}$$

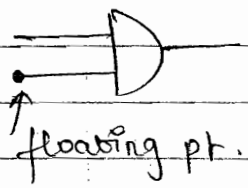
$$\text{Fan out} = \frac{I_{OL}}{I_{iL}}$$

} Take least value.

- V_{OH} → O/p high voltage I_{OH} → O/p high current
  I_{iH} → I/p high current V_{iH} → I/p high voltage.
  I_{OL} → O/p low current I_{iL} → I/p low current
  V_{OL} → O/p low voltage V_{iL} → I/p low voltage.

- Suppose V_{OH} ⇒ 5V But due to voltage fluctuation or By some other means $V_{iH} \neq 5V$, it is 4.9 V, 4.8 V, 4.7 V upto it acceptable But below these value high voltage 5V No any more high voltage become low voltage.
- V_{OL} = low voltage = 0V (ground voltage) V_{iL} can be accepted upto 0.1, 0.2, 0.3V But above this value V_{iL} is No any more low voltage it becomes high voltage

NOTE :- The floating I/P in TTL technique is taken as 1



I/p is Not connected to any logic

Q No. of NAND Gate = ?

• Implementation of Logic Gate By NAND

SOP (given) ⇒ AND-OR.

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

Shortcut :- Bar Bar, split inner Bar.

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

= 3 NAND Gate

NAND

• All terms should be product terms.

• To get this ⇒ wherever be the + make it double compliment ($\bar{\bar{+}}$)

For NOR Gate :

• All terms should be sum terms

• To get this ⇒ Where ever be the · make it double compliment ($\bar{\bar{\cdot}}$)

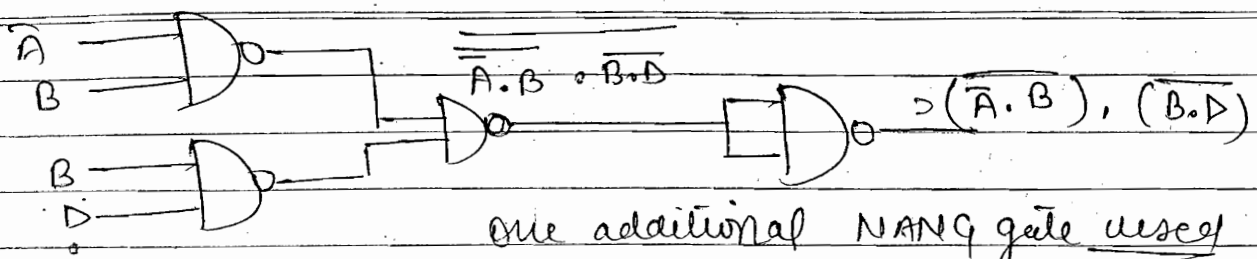
In POS form (NAND) :-

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

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

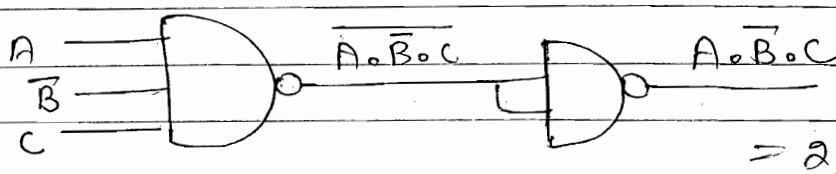
$$= \overline{(\overline{A} \cdot \overline{\bar{B}})} \cdot \overline{(\overline{\bar{B}} \cdot \overline{D})}$$

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



One additional NAND gate used.

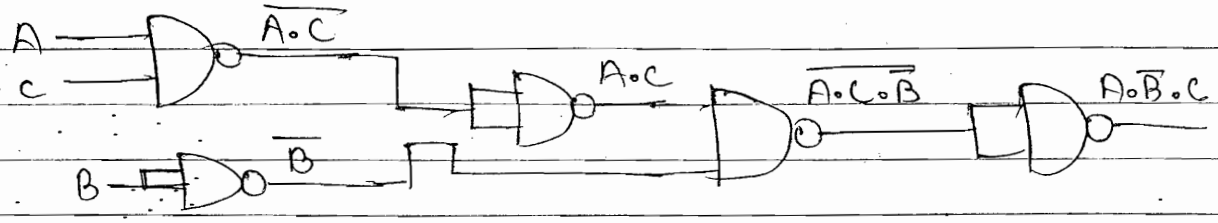
- If three I/p. $f = A \cdot \overline{B} \cdot C$



= 2 NAND Gate.

By using 2 I/p. terminal.

⇒ Take complimentary term separately



(NOR Gate) OR-AND Implimentation of logic gate By using NOR gate.

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

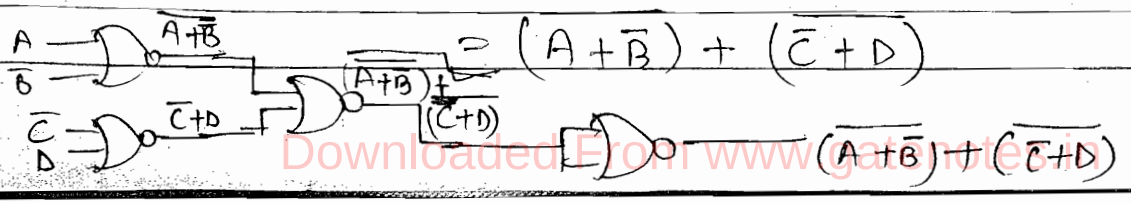
$= \overline{\overline{(\overline{A} + B) \cdot (C + \overline{D})}}$ $= \overline{(\overline{A} + B)_{①} + (C + \overline{D})_{②}}$ ③

3 NOR gate.

In SOR form:- $f = \overline{\overline{A} \cdot B} + \overline{C \cdot \overline{D}}$

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

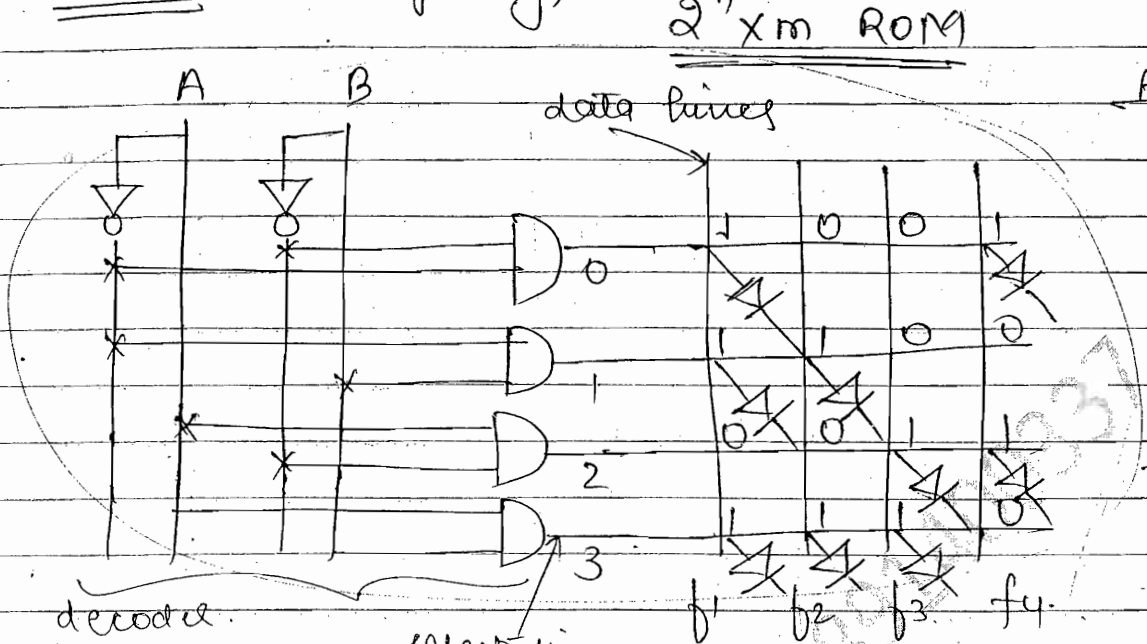
$= (A + \overline{B}) + (\overline{C} + D)$



Memory :-

30/9/2014

ROM :- (Designing):



- Diodes are putting there where '1' having.
- Blocks are Memory allocation.

If diodes are connected it call DIODE ROM

Transistor can also place in diodes.

$16 \rightarrow 4 \times 4$ (↓ select lines, → data line) $= 2^n \times 4$

$\rightarrow 2^n \times m$

where n = select lines m = data line

Given $f_1 = \Sigma m(0,1,3)$ $f_2 = \Sigma m(1,3)$

$f_3 = \Sigma m(2,3)$ $f_4 = (0,2)$

(above circuit)

- **For 3 Bit Squarer :-**

xyx →	
	0
	1
	2
	3
	4
	5
	6
	7 → $7^2 = 49$.

$7 \times 7 = 49$

32	16	8	4	2	1
1	1	0	0	0	1
~~f_6~~	~~f_5~~	f_4	f_3	f_2	f_1 → m = 6

No. of data lines required = 8

$\Rightarrow 2^3 \times 8 = 48$ → Memory allocation

No. of data lines are $m \rightarrow 6$

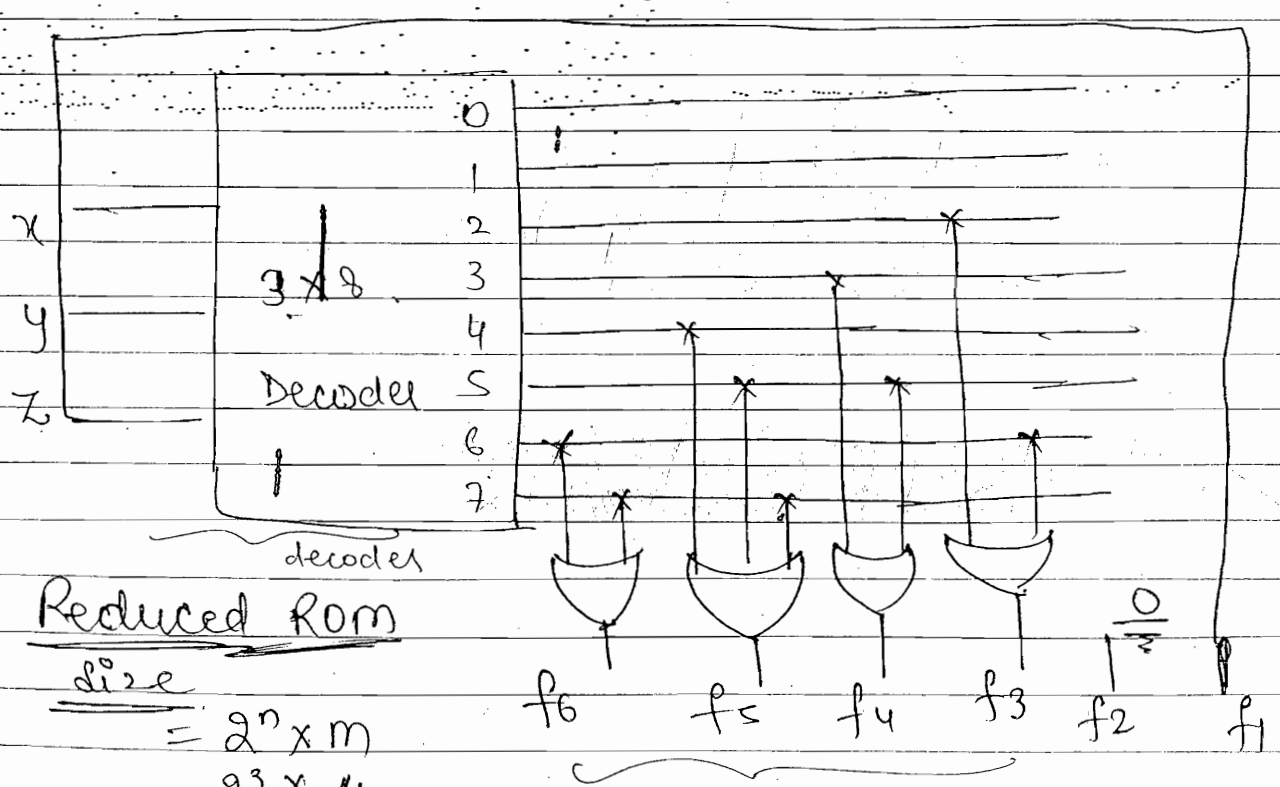
No — select — $n \rightarrow 3$.

$= 2^n \times m = 2^3 \times 6$

$= 48$.

	x	y	z	f6	f5	f4	f3	f2	f1	
0	0	0	0	0	0	0	0	0	0	'0'
1	0	0	1	0	0	0	0	0	1	'1'
2	0	1	0	0	0	0	1	0	0	'4'
3	0	1	1	0	0	1	0	0	1	'9'
4	1	0	0	0	1	0	0	0	0	'16'
5	1	0	1	0	1	1	0	0	1	'25'
6	1	1	0	1	0	0	1	0	0	'36'
7	1	1	1	1	1	0	0	0	1	'49'

Rom → Combination of Decoder and Encoder:



Reduced Rom size

$= 2^n \times m$

$= 2^3 \times 4$

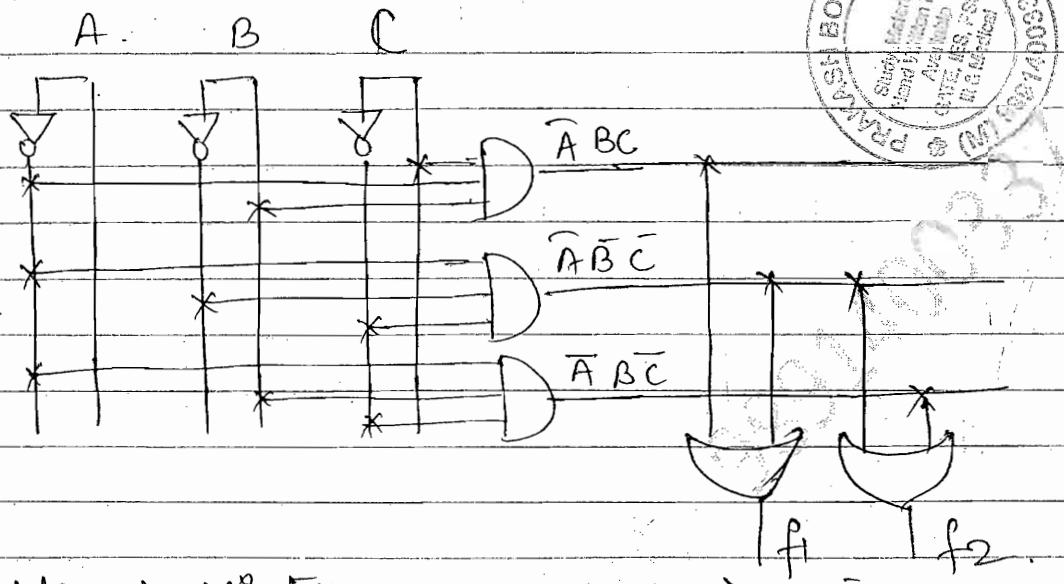
$= 32$.

• It will generate the No. square code so encoder.

• Given $f_1 = \bar{A}BC + \bar{A}\bar{B}\bar{C}$

$f_2 = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C}$ (same)

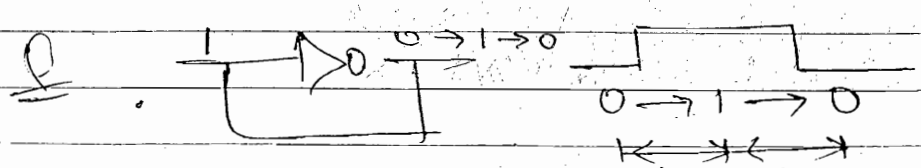
"PLA"



No. of Minterms = No. of AND gates

	AND Array	OR ARRay
ROM	fixed	programmable
PLA	programmable	programmable
PAL	programmable	fixed

• In PAL :- OR array is fixed and AND array is variable



$T = 2t_p$

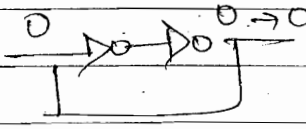
$\therefore f = \frac{1}{2t_p}$

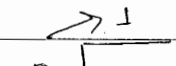
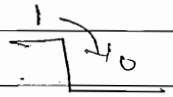
for n gates.

for N gates $\Rightarrow T = N(2t_p)$

$$f = \frac{1}{N(2t_p)}$$

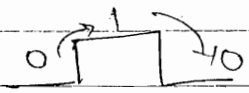
$\therefore n \rightarrow$ odd always

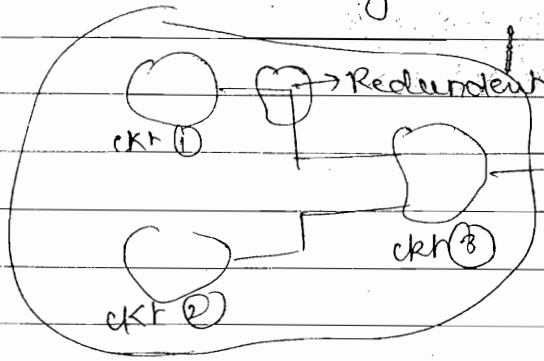
$n \rightarrow$ even ✗ $\longrightarrow$ otherwise 

- **Hazard :-**
- Static '0' Hazard :-  should come 0 but come 1
- Static '1' Hazard :-  should come 1 But comes 0

Because of different propagation delay the different part of the ckt, the o/p of the ckt is not accurate for temporary interval of time known hazard in the circuit.

For a temporary interval we are getting wrong result that result is called "Glitch".

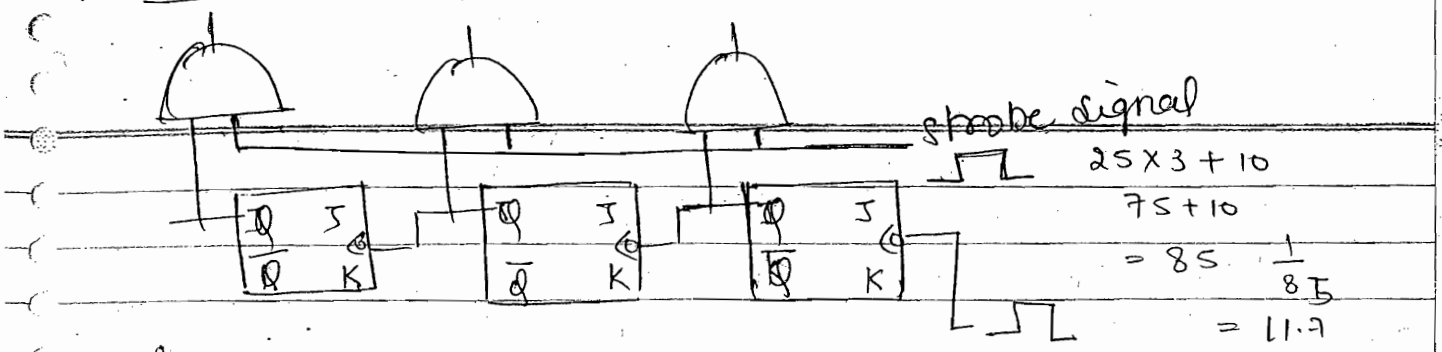
Types :- (1) Static '0' Hazard
(2) Static '1' Hazard
(3) Dynamic Hazard 



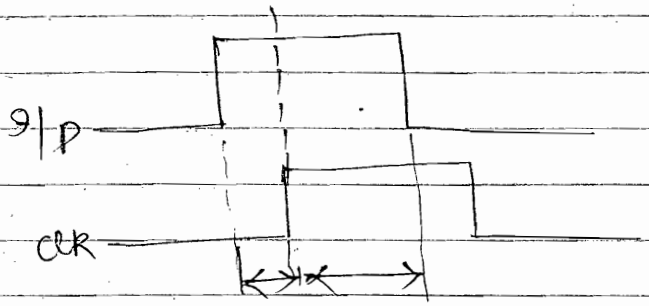
ckt ① o/p comes quickly
ckt ② o/p --- slowly
so o/p will generate by ckt ③ By ckt ① o/p and previous o/p of ckt ② for temporary period

So add Redundent ckt for delay the timing of ckt ①
Now o/p of ckt ① and ckt ② comes at same timing give accurate Result By ckt ③

Strobe signal : used to avoid error.



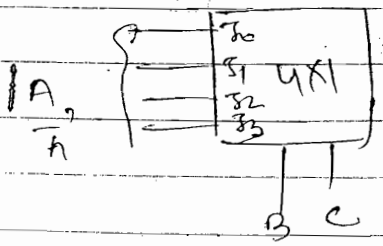
- To avoid decoding error strobe signal used.
- When the o/p is available then strobe signal is applied to get the display.



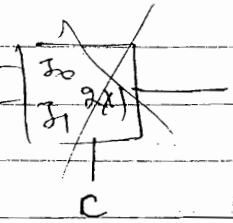
- Setup time should be added NOT Hold time for delay
- I/p should be applied first then clk pulse applied.
- MOD value divided to get frequency value.

Q.25 Page No. 65 → If initial value 101 not given
Ans: f/6

Pg No. 57
Q. 19 n = 3



$4\times1 = 2^2\times1 = 2^{3-1}\times1 = 2^{n-1}\times1$

If use 

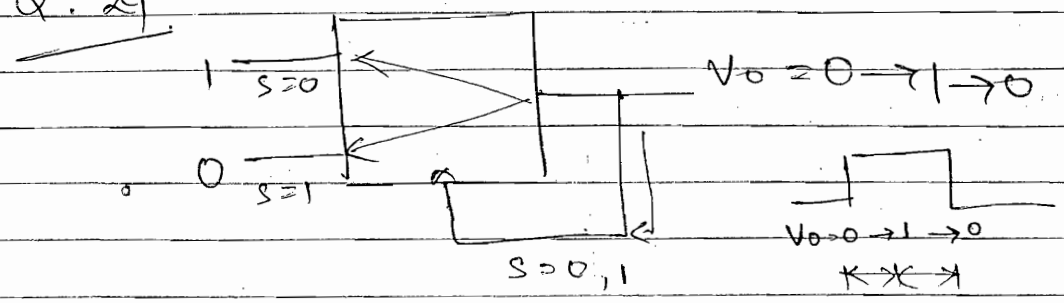
two multiplexer required

Q. 20 After 4 stage we get O/p = 1

Ans (B) 4 correction in answer

MUX → waveform generator

Q. 29



$f \rightarrow \frac{1}{2} = 0.5$ MHz,

✓ Logic families parameters [propagation delay, N.M., fan out etc) table in theory Book is imp (Q.). CMOS, TTL..

less delay → More Speed ✓

Number System :-

• Conversion

Decimal to any ⟹ Successive Division

(i) Decimal to Binary

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

→ Decimal pt

$73 = 7 \times 10^1 + 3 \times 10^0 = 70 + 3 = (73)_{10}$

2 | 12	→ 0	↑
2 | 6	→ 0	↑
2 | 3	→ 1	↑
1		

In fractional :-

$(0.75)_{10}$

$0.75 \times 2 = 1.50$

$0.50 \times 2 = 1.00$ ↓

$(12)_{10} = (1100)_2$

$(0.75)_{10} = (0.11)_2$

(ii) Decimal to Oct :- $(253)_{10}$ (iii) → To Hexa

8	253	
8	31	→ 5 ↑
	3	→ 7

$(253)_{10} = (375)_8$

$(351)_{10}$

16	351	
16	21	→ F ↑
	1	→ 5

$(351)_{10} = (15F)_{16}$

- Any to Decimal :-

ex :- (i) $(1101 \cdot 011)_2$

1	1	0	1 .	0	1	1
2^3	+ 2^2	+ 2^1	+ 2^0	2^{-1}	2^{-2}	2^{-3}
8	+ 4	+ 0	+ 1 .	0	+ 0.15	+ 0.15

$(13.375)_{10}$

100
8
0.95
10
0.1

(ii) Oct to Dec.

$(753)_8$

$= 7 \times 8^2 + 5 \times 8^1 + 3 \times 8^0$

$= (491)_{10}$

(iii) Hexa to Dec.

$(A3C)_{16}$

$= A \times 16^2 + 3 \times 16^1 + C \times 16^0$

$= (2620)_{10}$

- Binary to oct

$(\underline{11}\,\underline{101}\,\underline{011})_2$

$(75.34)_8$

Binary to Hexa

$(\underline{10}\,\underline{1101}\,.\,\underline{1101}\,\underline{11})_2$

$(2D.D8)_{16}$

- Any to Binary :-

(i) Oct to Binary

$(475.4)_8$

$(100\,111\,101.100)_2$

(ii) Hexa to Binary

$(DA9C.7)_{16}$

$(1101\,1010\,1001\,1100.0111)_2$

- Oct To Hexa :-

ex :- $(654.3)_8 = \underline{110}\,\underline{101}\,\underline{100}.\underline{011}$

$= (1AC.6)_{16}$ Ans

- Hexa to Octal :-

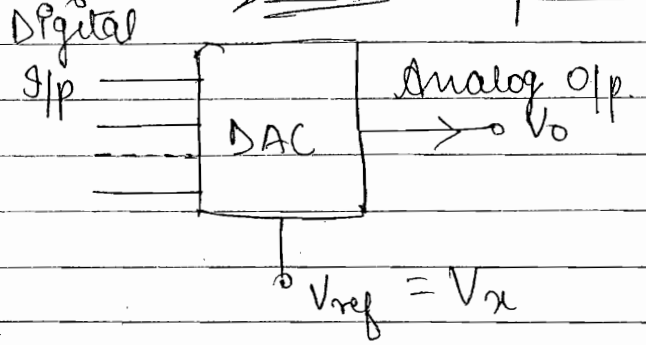
ex:- CA93.D

$\underbrace{1100}_{14}\underbrace{1010}_{5}\underbrace{1001}_{2}\underbrace{0011}_{2}.\underbrace{1101}_{3}$

$(14522\ 3\ .\ 64)_8$ Ans

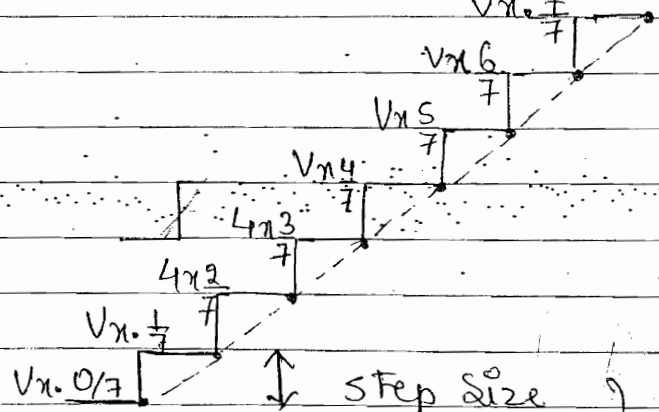
1|10|2014

- DAC :- parameter of DAC



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

Decimal eqⁿ. Binary data

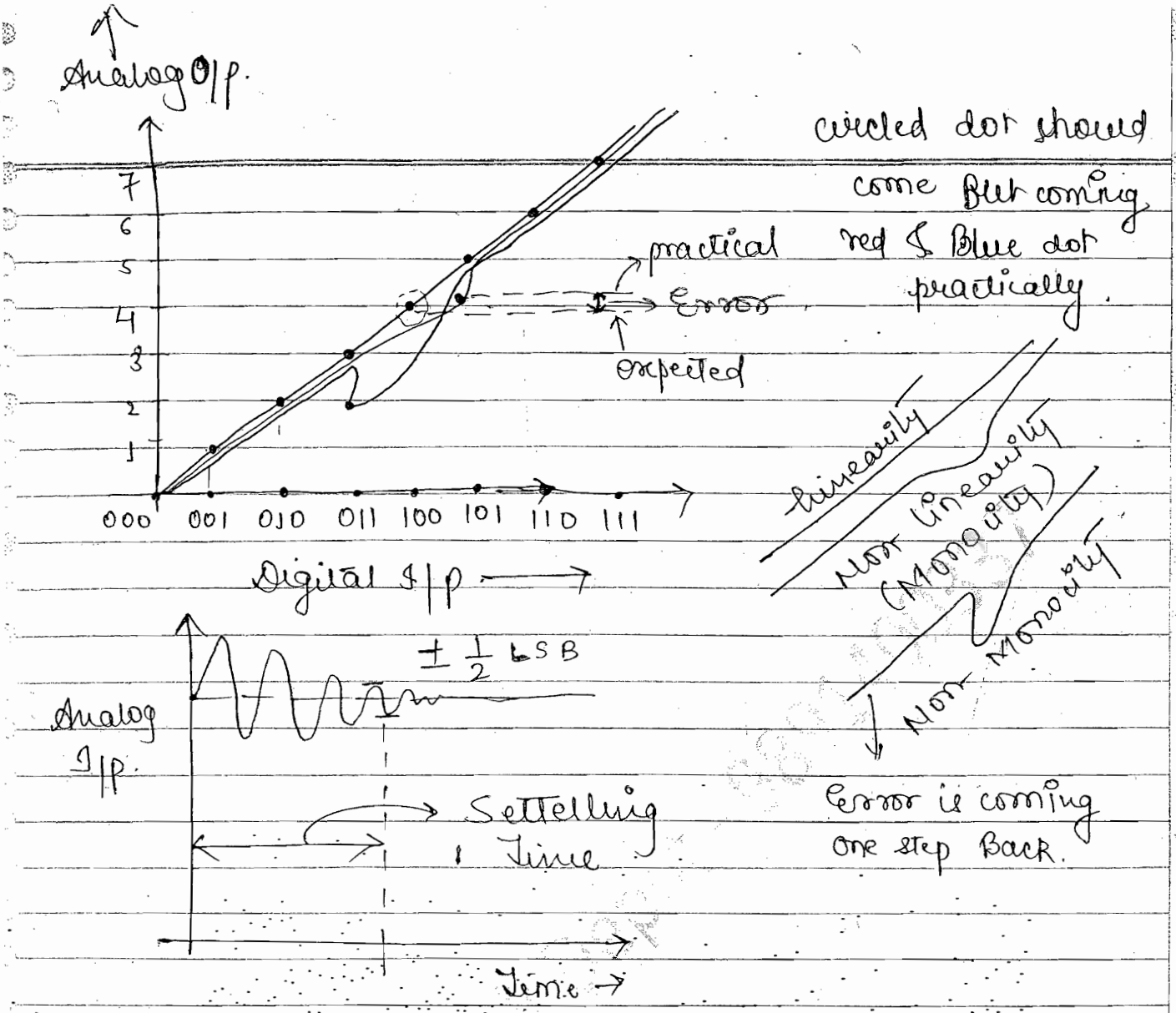


$$\text{No. of steps} = 2^n - 1$$

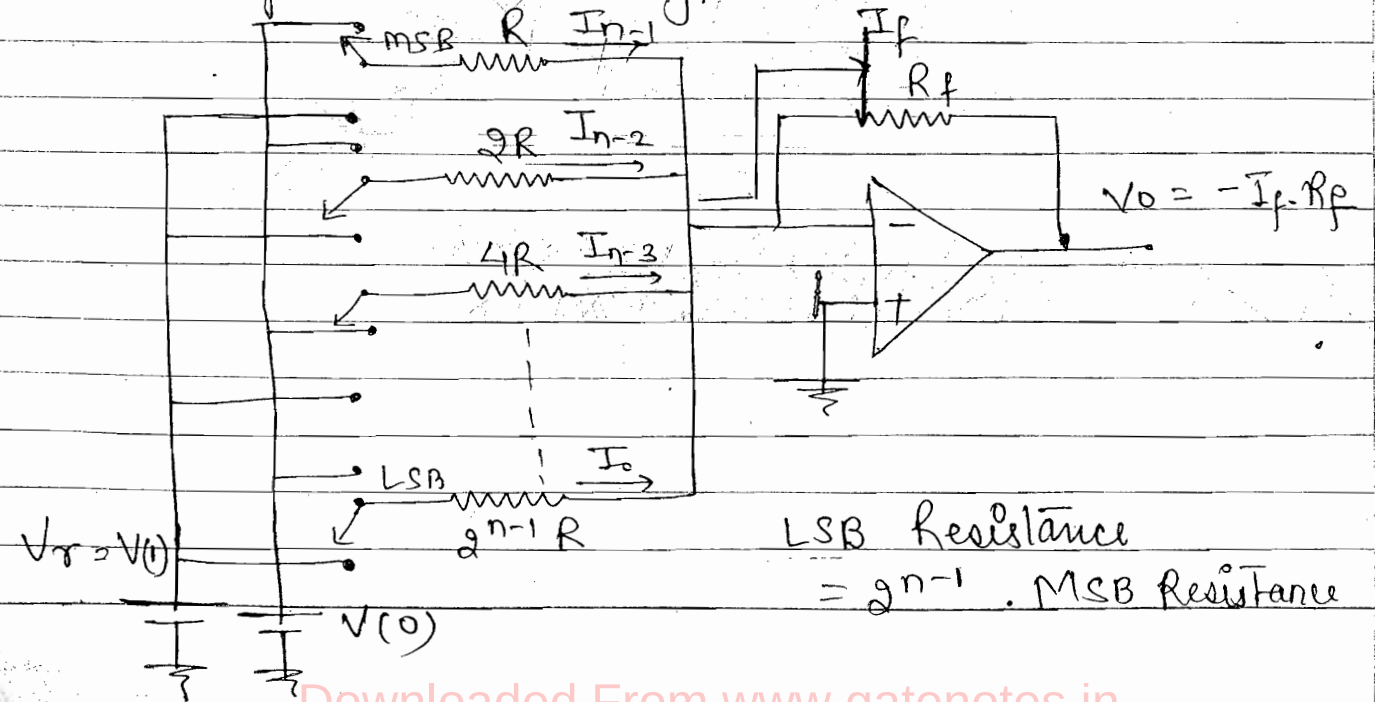
0 value don't require step so neglected

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

Resolution :- Smallest variation it can measure having high resolution



- Weighted Resistor Type DAC:



LSB Resistance $= 2^{n-1}$. MSB Resistance

(Assume 3 Bit DAC)

$$V_o = -I_f \cdot R_f$$

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

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

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

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

$$= \frac{-R_f}{R} \cdot \frac{V_r}{2^{n-1}} \left[1 + 2^1 + 2^2 + \cdots + 2^{n-2} + 2^{n-1} \right]$$

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

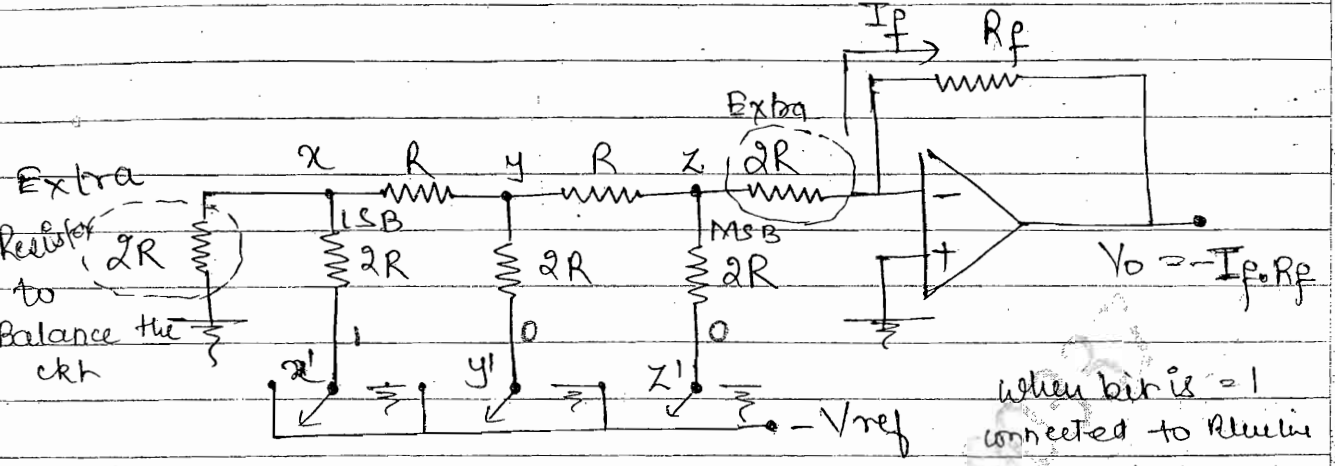
where $K = \left| \frac{-R_f \cdot V_r}{R\, 2^{n-1}} \right|$

- According to corresponding digital values, arrow will connect with V_r : V(1) line or V(0) line. Branches will give current. According to that analog voltage comes.
- V(0) connected arrow (Grounded) Branche current = 0
- $b_{n-1} \cdots b_0$ — If bit(1) is present, term is present otherwise bit (0) term absent.

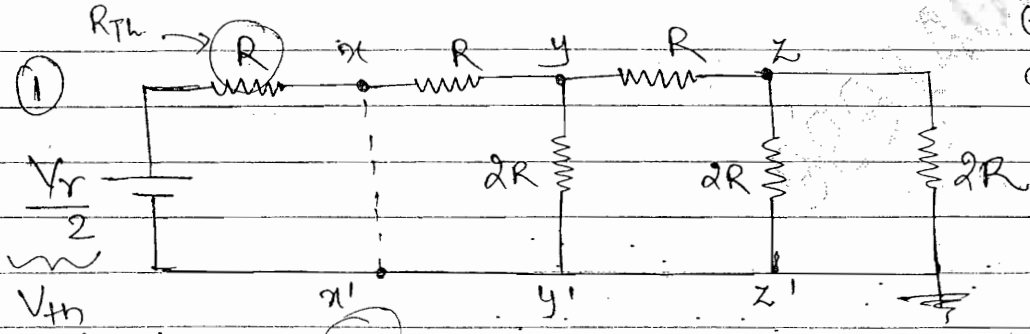
Q.

- LSB Resistance = 2^{n-1} . MSB Resistance
- $V_o = K \left[2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \cdots + 2^1 b_1 + 2^0 b_0 \right]$
- $K = \left| \frac{-R_f \cdot V_r}{R\, 2^{n-1}} \right|$

• R-2R ladder Type DAC [Inverting Type].

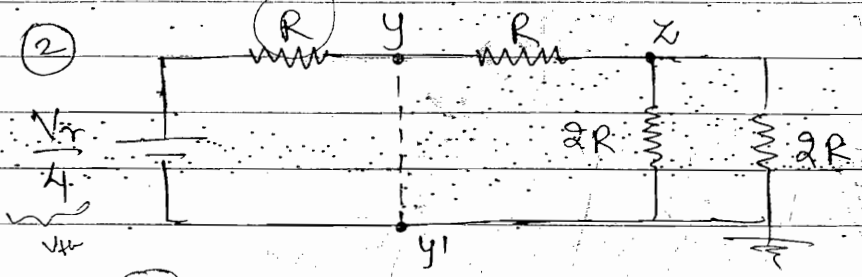


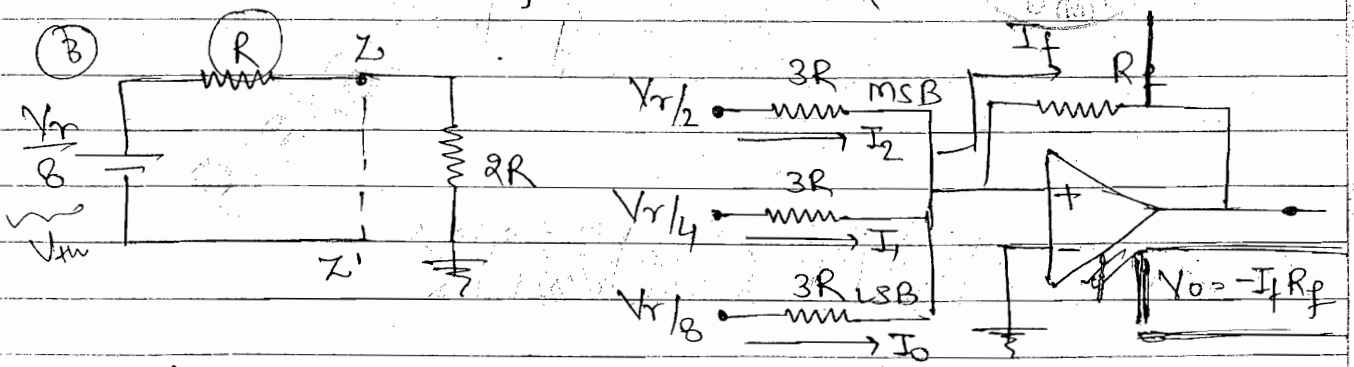
When bit is = 1 connected to Rleveln ($-V_{ref}$) otherwise connected to ground



V_{th} = Thevenins voltage

R_{th} = Thevenin's Resistor





$$V_o = -I_f R_f$$

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

$$= -R_f \left[\frac{V_r}{8(3R)} + \frac{V_r}{4(3R)} + \frac{V_r}{2(3R)} \right]$$

$$V_o = \frac{-R_f V_r}{3R}\left[\frac{1}{2^3} + \frac{1}{2^2} + \frac{1}{2^1}\right]$$

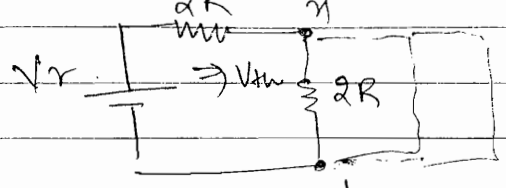
$$= \left[\frac{-R_f}{3R}\right] \cdot V_r \left[\frac{1}{2^n} + \frac{1}{2^{n-1}} + \cdots + \frac{1}{2^1}\right]$$

$$= \left[\frac{-R_f}{3R}\right]\left[\frac{V_r}{2^n}\right]\left[1 + 2^1 + \cdots + 2^{n-1}\right]$$

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

where $\boxed{K = \left(\frac{-R_f}{3R}\right)\left(\frac{V_r}{2^n}\right)}$

Suppose code is 100



$$V_{th} = V_r \cdot \frac{2R}{2R+2R} = \frac{V_r}{2}$$

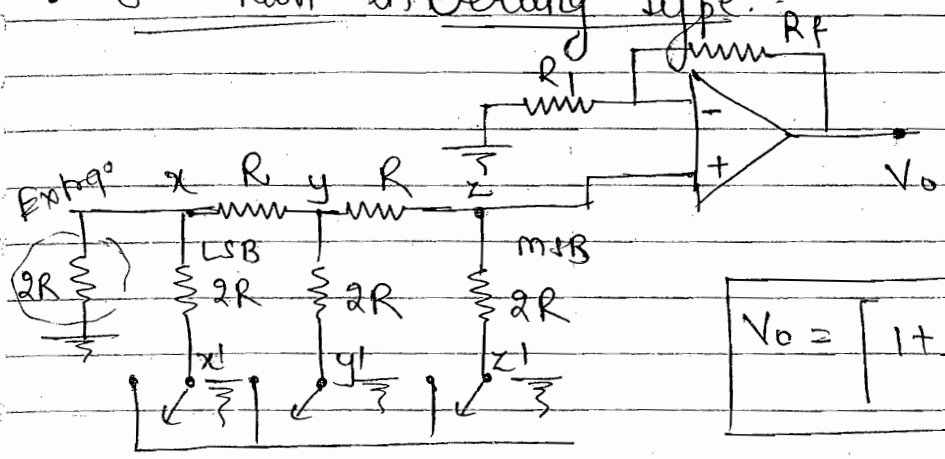
$$R_{th} = R$$

The apply to second we get $V_r/4$, R and again apply we get $V_r/8$, R.

- Suppose code 010 find upto $\frac{V_r}{4}$.
- Suppose code 001 find upto $\frac{V_r}{2}$

$$\boxed{V_o = \left|\frac{R_f}{3R}\right|\left(\frac{V_r}{2^n}\right)\left[\sum_{i=0}^{n-1} 2^i b_i\right]}$$

- For Non-inverting Type:



$$V_0 = \left[1 + \frac{R_f}{R_1}\right]\left[\frac{V_r}{2^n}\right]\left[\sum_{i=0}^{n-1} 2^i b_i\right]$$

Q.5 $V_0 = \left[1 + \frac{7}{1}\right]\left[\frac{1}{2^4}\right]\left[1010_{(2)} = 10\right]$ (LSB)

$\Rightarrow 8 \cdot \frac{1}{16_2} \cdot 10 = 5\ V$ Ans.

0101
1010

Q.2 $V_0 = K[\text{Dec. equivalent Binary data}]$

$V_0 = K\left[11011011_{(2)} = 219\right]$

$V_0 = K[219]$

$20 = K\left[11111111_{(2)} = 255\right)$ full scale value

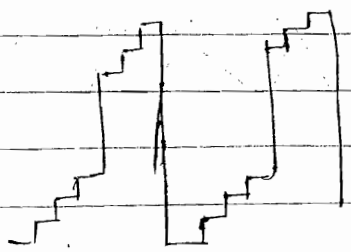
$20 = K[255]$

$K = 20/255$ Ans.

Q.8

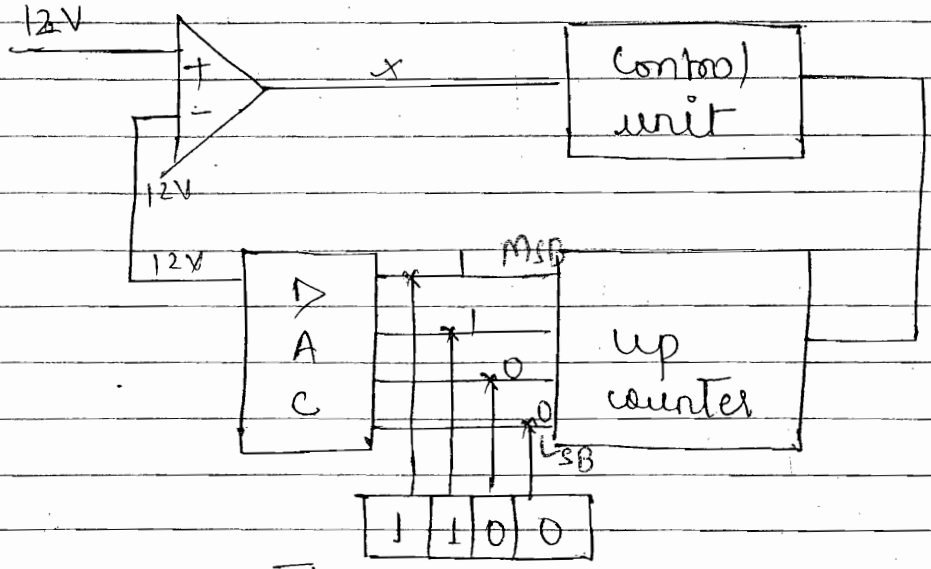
Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0	
0	0	0	0	0	0	0	→ 0
0	0	1	0	0	0	1	→ 1
0	1	0	0	0	1	0	→ 2
0	1	1	0	0	1	1	→ 3
1	0	0	1	0	0	0	→ 8
1	0	1	1	0	0	1	→ 9
1	1	0	1	0	1	0	→ 10
1	1	1	1	0	1	1	→ 11

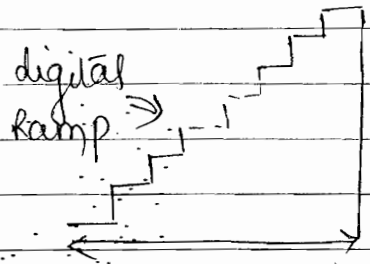
(B)



- ADC Digital Ramp Type ADC OR

Counter Type ADC





$t_c(max) = (2^n - 1)$ clk cycle

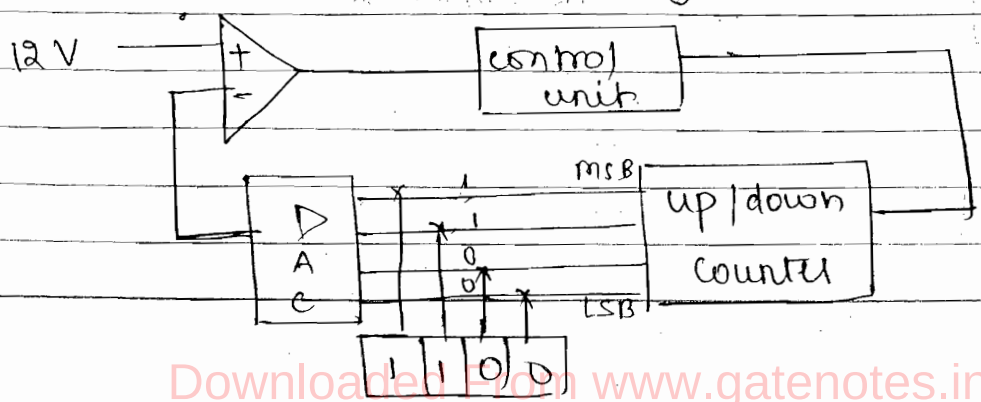
Conversion time (t_c)

When 12V is applied, it will ~~applied~~ display 1100,

ex: for 4 Bit → 15 clk cycle.

** Operation :- Theory Book

- Successive Approxiant type ADC



$t_c(max)$ = No. clk cycle

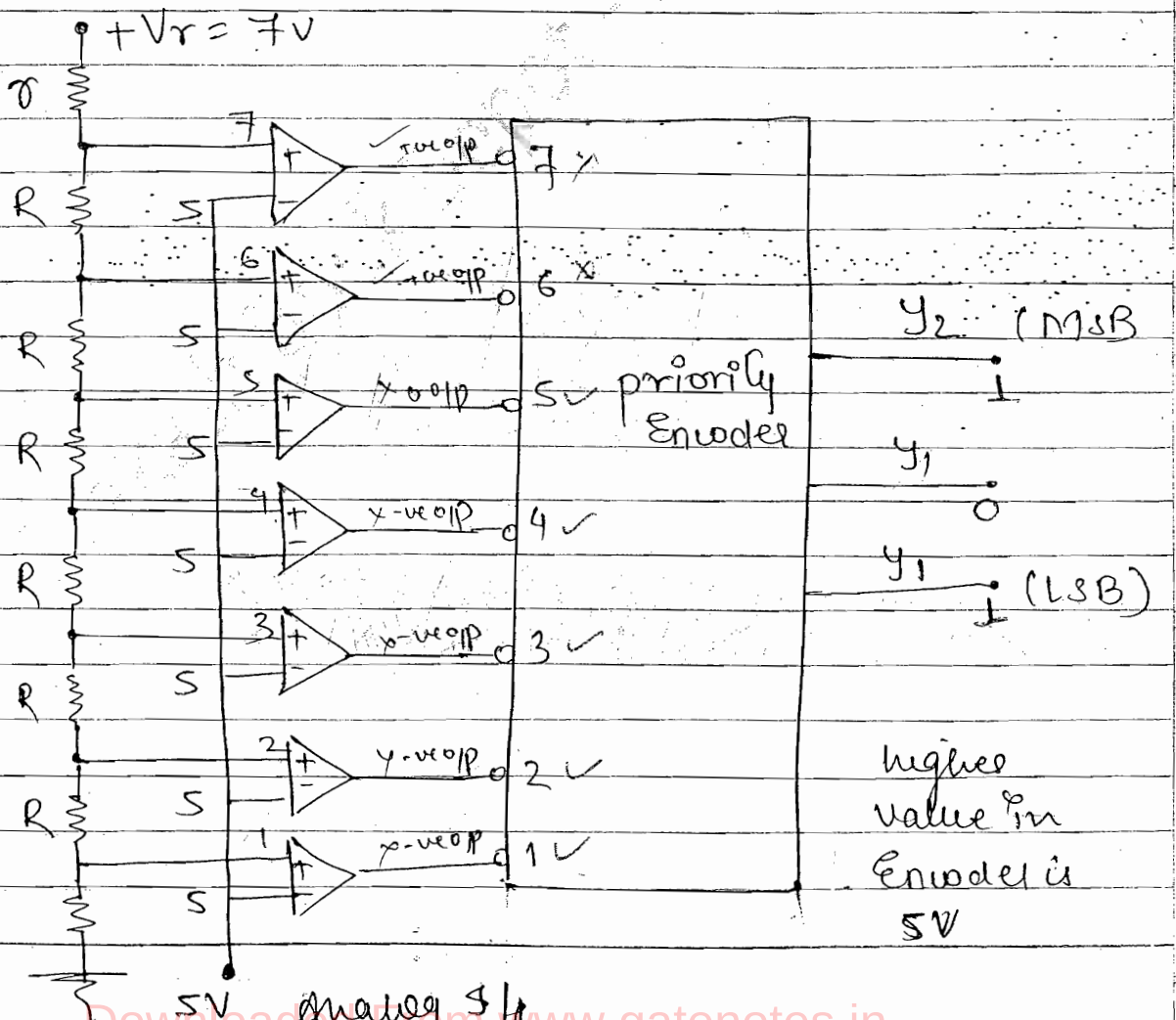
It will compare every Bit. when the voltage is -ve (greater than applied) o/p is 0 control unit convert again 1 → 0, when voltage is +ve (smaller than applied) o/p is 1 control unit allow remain 1 → 1

Initially 1 MSB
0
0
0 LSB.

4 Bit → 4 times comparision.

operation:- Theory Book.

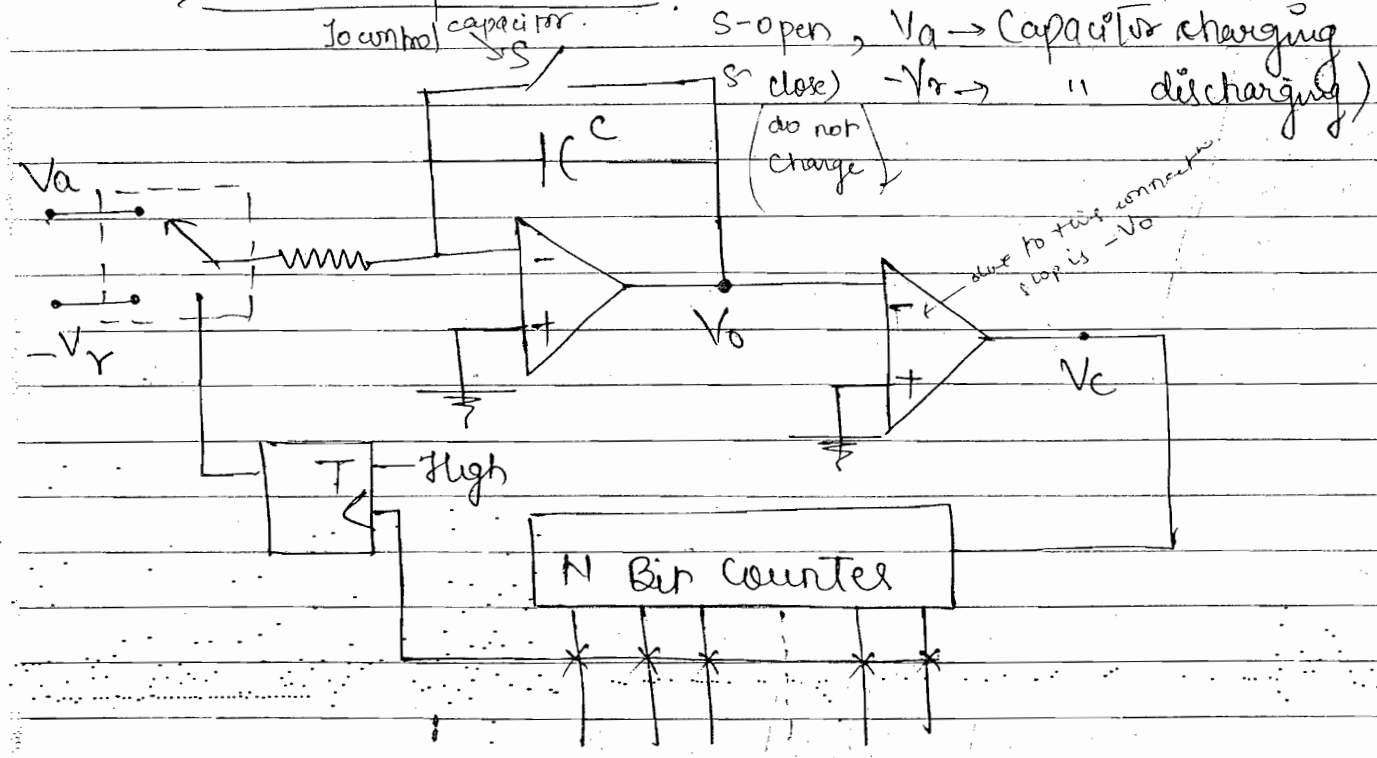
- **Flash Type ADC (OR) Simultaneously ADC (OR) Parallel Type ADC**

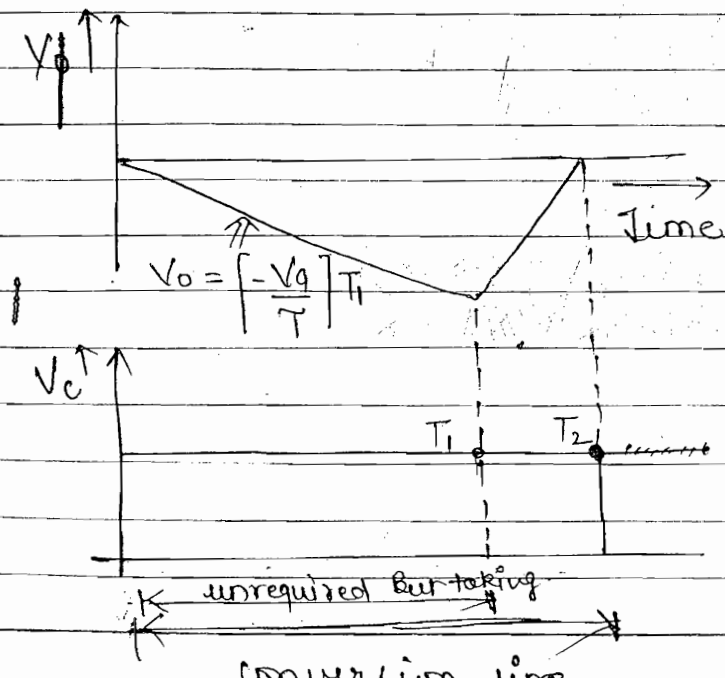


t_c (max) = 1 clk cycle

No. of comparators = $2^n - 1$

- **Dual Slope ADC**





$T_1 = 2^N \cdot T_c$

$T_2 - T_1 = n \cdot T_c$

N ⇒ No. of Bits

n ⇒ No. of counts

$$V_0 = -\frac{1}{T}\int_0^{T_1} (V_a)\,dt$$

$$= -\left[\frac{V_a}{T}\right]T_1 \quad \text{——— (1)}$$

$$V_0 = -\left[\frac{V_a}{T}\right]T_1 + \left[-\frac{1}{T}\int_{T_1}^{t} (-V_r)\,dt\right]$$

$$= -\left[\frac{V_a}{T}\right]T_1 + \left[\frac{V_r}{T}\right]\left[t - T_1\right]$$

$V_0 = 0$ at $t = T_2$

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

$$\left[\frac{V_a}{T}\right]T_1 = \left[\frac{V_r}{T}\right]\left[T_2 - T_1\right]$$

$V_a \cdot 2^N \cdot T_c = V_r \cdot n \cdot T_c$ $\qquad \boxed{V_a = \frac{V_r \cdot n}{2^N}}$

if $V_r = 2^N$ $\qquad \boxed{V_a = n}$

$t_c(max) = T_2$

$t_c(max) = T_1 + nT_c$

$= 2^N T_c + nT_c$

$t_c(max) = (2^N + n)T_c$

$t_c(max) = (2^N + 2^N)T_c$

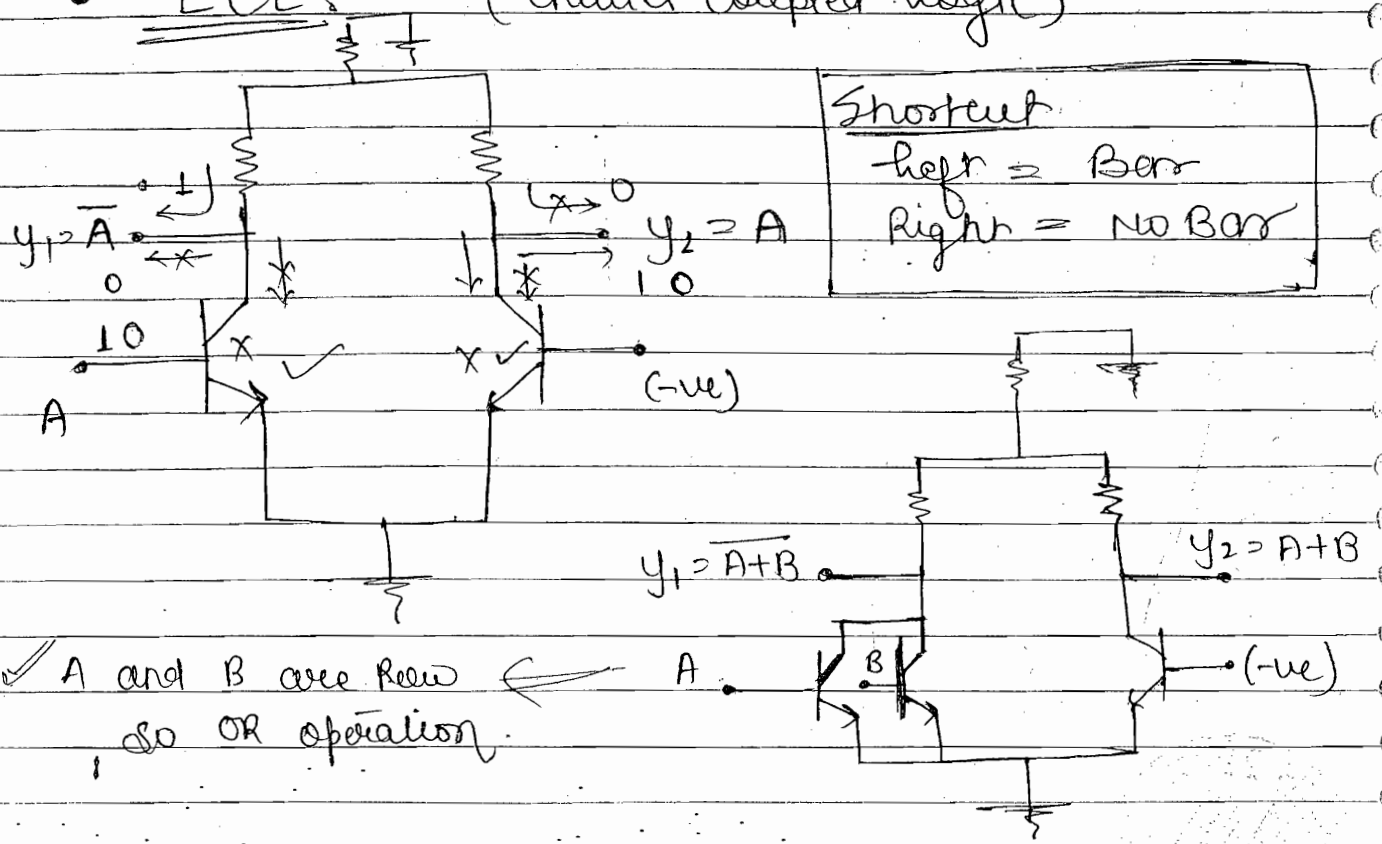
$t_c(max) = (2^{N+1})T_c$

$$\boxed{t_c(max) = (2^{N+1})T_c}$$

Disadvantage :- Even when the capacitor get charged counter still counts upto Max. value 111

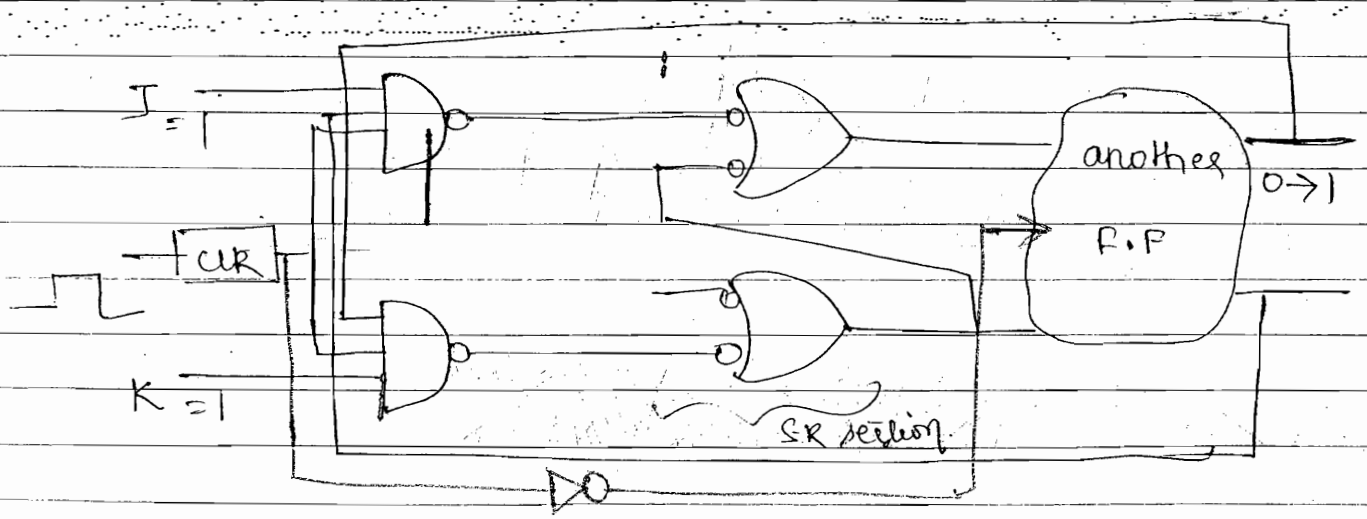
- After 7th FF trigger and Arrow get connected to V_r

- **ECL :-** (Emitter coupled logic)



✓ A and B are Row ⟸ so OR operation.

- **MASTER SLAVE :-**



It consist of two section MASTER and slave connected from same clock generator But the second section is operated By NOT GATE

So both the section cannot be trigger simultaneously and the final o/p occurance time time clk pulse is in OFF state So there is No toggle repetition and race round problem is eliminated external

NOTE :- Feedbacks are taken from the final o/p only.

End