|| Jai Gajanan Maharaj ||

Rahul D. Khurade. (08019768502)

PM4 - CSE. ZOOM Internet.

Digital System

1. Boolean expressions, laws, k-maps, Hazards in digital system
2. Multiplexer, Decoder, Encoder
3. flip-flop, interconnection, counters, shift registers
4. Numbersystems, representation, fixed point & floating point arithmatic.

- Combinational circuits

$y = f(x)$

In this current output depends on current input
Its ckt of sequential elements.

- Sequential circuits.

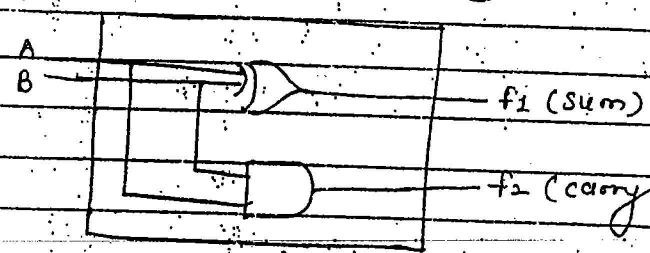
$y = f(x, s)$; $s \rightarrow$ state (previous output)

In this current output depends on current input & previous output i.e state. ex: $\rightarrow$ flipflop

formula for number of flip-flop required $= \log_2 n$

- Combinational function

→ Multiinput & single o/p system ex: → XOR gate
→ Multiinput & multi o/p system ex: → HalfAdder



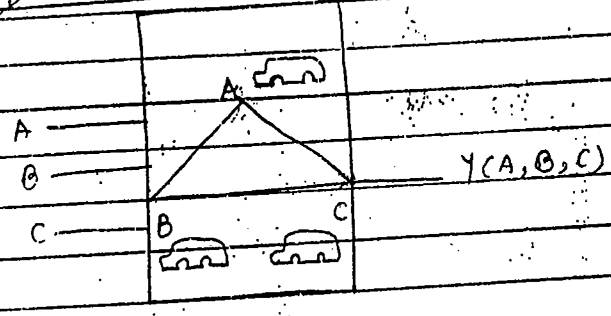
Half Adder

Truth Table

- Physical Problem → Boolean expressn → Computing machine → Boolean Results → ~~Actuator~~ (provide soln) (Electromechanical Device)

~~Track Simulation~~

ex:→



- ~~Track laid in form of equilateral triangle.~~
- ~~Three vehicles moving with some velocity are present at three vertices.~~
- ~~System should generate output 1 as when any two vehicle are meet together.~~

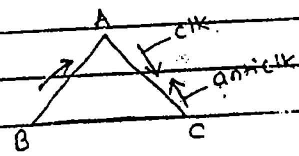
0 → clockwise movement

1 → anticlockwise movement

→ minterm :- product of i/p such that product is 1 i.e. $\boxed{A'B'C}$

$A'B'C$

	A	B	C	Y
Minterm 0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
Min term 3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0



Maxterm

→ max term → sum of i/p such that total value of sum is 0 → $A'+B'+C'$

0 0 0 → 0

When all vehicle moving in same direction then they cannot meet together.

* Sum of Products * (SOP)
* Product of sum * (POS)

Input combination for which output is '1' called min term.

Sum of Products (SOP) → All the Minterm

$Y(A,B,C) = A'B'C + A'BC' + A'BC + AB'C' + AB'C + ABC'$

$= \Sigma(1,2,3,4,5,6)$

** Product of sum (POS) **

$Y(A,B,C) = (A+B+C)\cdot(A'+B'+C') = \pi(0,7)$

for sum of products (SOP) ⇒

6 ... (3 i/p) AND gate

1 ... (6 i/p) OR gate

3 ... NOT gate

for product of sum (POS) ⇒

2 ... (3-i/p) OR gate

1 ... (2-i/p) AND

} 3 - NOT gate

If more min term i.e. output is '1' then use POS because we have to reduce h/w required.

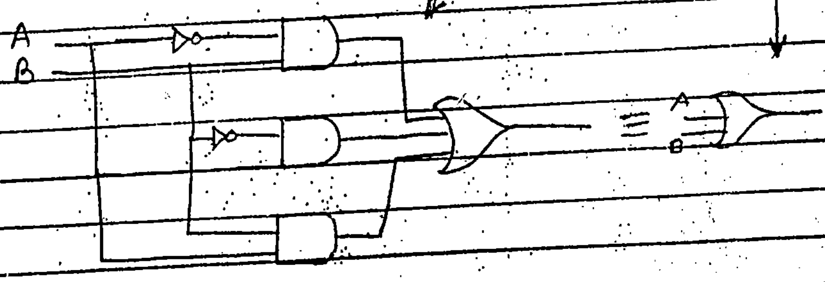
min A+B A'

A	B	OR
0	0	0
0	1	1
1	0	1
1	1	1

minterm go for pos

POS ⇒ $f(A,B) = A+B$

SOP ⇒ $f(A,B) = \underbrace{A'B + AB' + AB}$



A	B	AND
0	0	0
0	1	0
1	0	0
1	1	1

maxterm sop

SOP

POS ⇒ $(A+B)\cdot(A+B')\cdot(A'+B)$

SOP ⇒ $(A'+B')$

~~Identify~~ the max terms for following expressions

$f(A,B,C) = A'+BC$

(a) 3, 5, 6

(b) 4, 5, 6

(c) 3, 4, 5, 6

(d) None

	A	B	C	A'+BC
	0	0	0	1
	0	0	1	1
	0	1	0	1
	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
	1	1	1	1

Maxterm

⇒ $(A'+BC) = \underbrace{(A'+B)}\cdot\underbrace{(A'+C)}$ { pos form

	A	B	C	A	B	C	
	1	0	?	1	?	0	
4	1	0	0	1	0	0	4
5	1	0	1	1	1	0	6

{4, 5, 6}

remaining 0, 1, 2, 3, 7 are minterms

↳ min soc
max 0 - pos

2) Consider following 4 i/p and 4 o/p system.

I/ps are a b c d
o/ps are w, x, y, z.

The system having the following behaviour

$$\boxed{p_o = (p_i + 3) \text{ MOD } 16}$$

Where,

$p_o, p_i \rightarrow$ Are decimal value of concerned binary o/ps and i/ps resp.

A	B	C	D	
1	1	0	1	$\rightarrow p_i$ (13)

then

$(13+3) \text{ Mod } 16 = 0 = p_o$

$\Downarrow$

w	x	y	z
0	0	0	0

↳ Maxterm

See page No - (4) ⇒

to be continue.....

+ → OR 16/6/2011
. → AND

Note: See page No - 11 directly

* Boolean Laws *

1) ~~Identity~~ law ⇒ Identical operation on identical variable gives unique result.

ex:→ $A+A+A=A$
$A \cdot A \cdot A = A$ } Identical Law.

⊙ AND, OR follows this law.

NAND (↑) ⇒ $(\cdots((\underbrace{A\uparrow A}) \uparrow A) \uparrow A) \cdots)$

$(\bar{A}) \uparrow \bar{A} = A + \bar{A} = 1, 0$

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

⊙ NAND, NOR $\boxed{\text{doesnot}}$ follow identity law.

⊙ XOR ⇒ $(\cdots((\underbrace{A \oplus A}) \oplus A) \oplus A) \cdots) = \begin{matrix} 0 \\ A \end{matrix}$, } alternat 0, A

$\underbrace{0 \oplus A}$

$A \oplus A = 0$ ✓

∴ XOR and XNOR $\boxed{\text{doesnot}}$ follow identity law.

⊙ $\boxed{A \$ B = A + B' = \$(A,B)}$ ≠ $A' + B$

$(\cdots((\underbrace{X \$ X}) \$ X) \$ X) \cdots) = 1$

$X \$ X' = 1$

$\boxed{1 \$ X' = 1}$

∴ \$ follows identity law.

Note: AND, OR and \$ follows Identity law.

4

Continue....

A	B	C	D	
1	1	0	1	$\Rightarrow P_i$ 13

Then $(13+3) \text{ MOD } 16 = 0 = P_o$

w	x	y	z
0	0	0	0

maxterm

→ which of the following gives eqn for

$w(A, B, C, D)$

(a) $\Sigma(3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$

(b) $\Sigma(5, 6, 7, 8, 9, 10, 11, 12)$

(c) $\Sigma(6, 7, 8, 9, 10, 11, 12, 13)$

(d) $\Sigma(5, 6, 7, 8, 9, 10, 11, 12, 13)$

$\Sigma \rightarrow$ minterm (1) only

Soln: 13 can not be minterm — because $(13+3) \text{ MOD } 16 = 0$

so Eliminate option (c) and (d)

$P_i = 3$ $\quad P_o = (P_i+3) \text{ MOD } 16$

$= (3+3) \text{ MOD } 16$

$\boxed{P_o = 6}$

	A	B	C	D	P_o	w	x	y	z
$\boxed{P_i = 3}$ →	0	0	1	1	6	$\boxed{0}$	1	1	0

for 3 binary is

3 cannot be Minterm

Eliminate option (a)

∴ The correct option is (b) Ans.

This is for 3 variable ex. $f(A,B,C)$ we get 8 input 000 to 111, P_i is from 0 to 7 of those are 3-variable

$$\boxed{P_o = (P_i + K) \text{ MOD } 8}$$

Where K is any no. from 0 to ...

Observation: If $P_o = (P_i + n) \text{ MOD } 16$ then

n → is any no. from 0 to ∞

P_i → is from 0 to 15 for 4-variable

~~then option (c) is alway correct option~~

(3) With resp. to Above problem identify the correct statement

(a) Except the function 'w' all the output function having equal no. of minterm and maxterm.

(b) Except for the fn 'z' all output having equal no. of minterm ~~and~~ maxterm.

(c) All the outputs having equal no. of Minterms and maxterm. ✓

(d) only fn 'w' has 8 minterm and 8 maxterm.

$$\boxed{P_o = (P_i + 3) \text{ MOD } 16}$$

A	B	C	D	w	x	y	z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	1	1	0	1
1	0	1	1	1	1	1	0
1	1	0	0	1	1	1	1
1	1	0	1	0	0	0	0
1	1	1	0	0	0	0	1
1	1	1	1	0	0	1	0

That is there are equal no. of minterm and maxterm for all outputs

Characteristics of Boolean expression

- Duality
- Complementation

1. Duality :-

$F(\underbrace{x_1, x_2, x_3 --- x_n}_{\text{Boolean variable}}, \underbrace{+, \cdot, 1, 0}_{\text{operator}})$

dual

$F_d(x_1, x_2, x_3, --- x_n, \cdot, +, 0, 1)$

AND $\rightleftharpoons$ OR

$0 \rightleftharpoons 1$ (0 changes to 1 and viceversa)

- providing function properties indirectly
- Inter-conversion realization across the Logic system.

$\boxed{+ve \rightleftharpoons -ve}$

Categories of Boolean exp. on basis of Behaviour :

- Orthogonal $\Rightarrow f_d \to f'$
- Selfdual $\Rightarrow f_d \to f$
- Non-orthogonal
- Non-self dual

$\Rightarrow f_d \neq f'$, $f_d \neq f$

AND Replace with OR and vice-versa

$f_1(A, B) = A'B + A \cdot B'$ (EX-OR)

dual

~~$f_1 d(A, B) = (A' + B) \cdot (A + B')$~~ EX NOR

$= A'B' + AB$ (EX-NOR)

$f_1' = f_{1d} = (A'+B)\cdot(A+B')$

$\boxed{f_{1d} = f_1'}$ orthogonal

2, 4, 6 variable EX-NOR

For even no. of variable, $\boxed{\text{EX-OR}}$, EX-NOR follows complementary relation.

For odd no. of variable, EX-OR EX-NOR follows $\boxed{\text{Equality}}$ relations.

(1, 0, +, ·)

example :- $A \oplus B \oplus C = A \odot B \odot C$

$A \oplus B = (A \odot B)'$

$f_2(A,B,C) = AB + BC + CA$

dual

$f_{2d}(A,B,C) = (A+B)\cdot(B+C)\cdot(C+A)$

**

$\Rightarrow (B+AC)(C+A)$

$AC + AC + AB = AB + BC + CA$

$\boxed{f_{2d} = f_2}$ self dual

check

dual $\boxed{\begin{matrix} x'+1 = 1 \\ x\cdot 0 = 0 \end{matrix}}$

replace 0 by 1, + by ·

$f_3(A,B) = A'+B$

dual

$f_{3d}(A,B) = A\cdot B$

$f_3' = A\cdot B'$

$f_{3d} \neq f_3$, $f_{3d} \neq f_3'$

$\boxed{f}$ property (P) ; If f has property P then its dual f_d also follows property P.

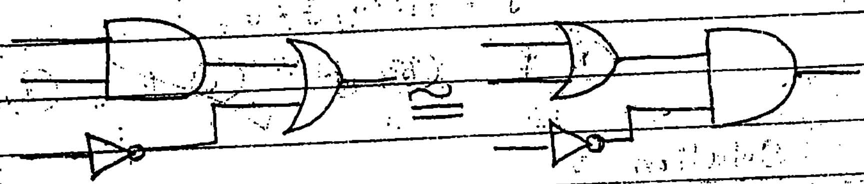
$\boxed{f_d}$

5V → 1
-5V → 0

+ve logic and -ve Logic

+ve logic : In +ve logic most voltage is '1'

-ve logic : In -ve logic most voltage is '0'



if it is +ve logic then it is -ve logic

Vice versa

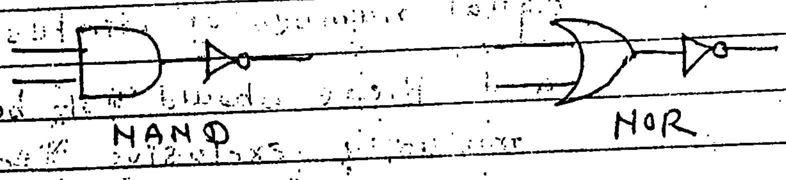
~~If it is -ve logic then it is +ve logic.~~

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

+ve logic -ve logic system

+ve logic AND become OR in -ve logic

+ve logic NAND becomes NOR in -ve logic



$\boxed{f = f_d}$ $\boxed{f = f_1}$

Self Duality

Que) $f(A, B, C) = \Sigma(0, 1, 2, 3)$ it is self dual fn.

the dual function of f is f_1

$f' \cdot f_1' + f \cdot f_1$

(a) f (b) f' (c) 1 (d) 0

Solution :

$$f' \cdot f_1' + f \cdot f_1 = f' \cdot f' + f \cdot f$$
$$= f' + f$$
$$= 1 \quad \text{Ans}$$

$f(A, B, C) = (A'B'C) + A$

$f_d(A, B, C) = (A' + B' + C) \cdot A$

no. minterm = no. of Maxterm

Note: A boolean function becomes $\boxed{\text{Self dual}}$ if and only if it contents equal number of minterm and maxterm and there should not be any mutually exclusive term.

The presentation of mutually exclusive term is given by condition.

$\Sigma = min = 1$ SOP
$\Pi = max = 0$ POS

$f(x_1, x_2, x_3, \ldots x_n) = f(x_1', x_2', x_3', \ldots x_n')$

$f_1(A, B, C) = \Sigma(0, 1, 3)$ Not a self dual functi
Minterm → 1 o/p – SOP

$f_2(A, B, C) = \Sigma(0, 1, 2, 6)$ → Not self dual f^n.

Minterm → output 1
Maxterm → output 0

$f_2(0) = 1 \quad f_2(0, 0, 0)$

$f_2(1) = 1 \quad f_2(0, 0, 1)$

$f_2(2) = 1 \quad f_2(0, 1, 0)$

Minterm $f_2(6) = 1 \quad f_2(1, 1, 0)$

$f_2(7) = 0 \quad f_2(1, 1, 1)$
maxterm

The function f_2 is not self dual.

1 and 6 are mutually exclusive
Hence, it can not be self dual.

If (0, 7), (1, 6), (2, 5), (3, 4)
then they are Mutual exclusive.

if 0 is present then there is no '7' then it is called self dual.

$f_3(A, B, C) = \Sigma(0, 1, 2, 3)$
→ self dual.

$(0,7)$ $(1,6)$, $(2,5)$, $(3,4)$

$2^3 = 8$ combination

Q) What will be number of self dual function are possible with three boolean variables?

$n=3$

(a) 3 (b) 8 (c) 16 (d) 256

Soln: $\Sigma(0,1,2,3)$

$\Sigma(0,1,2,4)$

$\Sigma(0,1,5,3)$

4 group

$2^{\text{group}} = 16$ self dual fn

$\Sigma(0,1,5,4)$

no. of group $= 2^{n-1}$

no. of self $= 2^{\text{no. of group}}$

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

for $n=3$

$(0,7), (2,5), (1,6), (3,4) = 2^4 = 16 = 2^{(\text{no. of groups})}$

Q) How many self dual possible with n boolean variable

2^{n-1}

no. self dual $= 2^{2^{n-1}}$

Ans: (a) 2^{n-1} (b) $2^n - 1$ (c) 2^{2^n} (d) $2^{2^{n-1}}$

Soln: n variable 2^n combination are possibles.

$3 = 2^3 = 8$

$\frac{2^n}{2} = \boxed{2^{n-1}}$ groups.

$\frac{2^3}{2} = 4$

$2^{(\text{group})} = \left(2^{2^{n-1}}\right)$ Ans

for $\boxed{n=3}$ $2^{2^{3-1}} = 2^4 = 16$

Complementation

The complementation obtain similar to that of dual with only exception is even $\boxed{\text{variable}}$ also to be replace by their complements.

$$f = F(x_1, x_2, x_3, \ldots x_n, +, \cdot, 0, 1)$$

$$f' = F^D(x_1', x_2', x_3', \ldots x_n', \cdot, +, 1, 0)$$

Variable $\rightleftharpoons$ complemented

AND $\rightleftharpoons$ OR

$0 \rightleftharpoons 1$

Complementation in term of Duality

$$f'(x_1, x_2, x_3, \ldots x_n) = f_d(x_1', x_2', x_3', \ldots x_n')$$

Duality in term of Complementation

$$f_d(x_1, x_2, x_3, \ldots x_n) = f'(x_1', x_2', x_3', \ldots x_n')$$

If the variable of complementation function are complemented then it becomes dual.

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

$$f_d(A, B, C) = A \cdot (B + C)$$

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

$$f'(A, B, C) = f_d(A', B', C')$$

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

only variable of f_d are complement

$\boxed{\text{Vice - Versa}}$

$f_d(A, B, C) = f'(A', B', C')$

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

$= A(B + C)$

Gate Que

How many boolean function possible with 'n' boolean such that

$n \rightarrow$ no of variable

$2^n \rightarrow$ Natural var

$2^{2^n} \rightarrow$ Natural of fn

$f(x_1, x_2, x_3 \ldots x_n) = f'(x_1', x_2', \ldots x_n')$

It is nothing but both are same

$= f(x_1, x_2, \ldots x_n)$

Q) If the following exp. solve then

$A' + BC = 0$

$AB + A'C = 1$

$B'D' + BD = 0$

what will be the values of A, B, C, D

Soln :- $A' = 0$ & $BC = 0$ ⟸ then only $A' + BC = 0$

$\therefore (A = 1)$

$1 \cdot B + 0 \cdot C = 1$

$\boxed{\therefore B = 1}$

$AB + A'C = 1$

$B'D' + BD = 0$

$0 + 1 \cdot D = 0$

$\boxed{D = 0}$

$A' + BC = 0$ $B = 1$

But $A = 1$, $\boxed{\therefore C = 0}$

Mixed Logic

How many $\boxed{\text{feingey}}$ function are possible with ~~two~~ boolean variables

$p \to$ n-g variable
$m^{n} \to f^n$

$n \to$ no. of variable
$2 \to$ Nature of Variable (binary)
$2 \to$ Natural f^n (

Ans :- How many boolean function possible with n-boolean variable

$\Rightarrow 2^{2^{n}}$
$n \to$ Numbers of variable
$2 \to$ Nature of variable
$2 \to$ 2 is Natural function

Ternary $\begin{cases} 0 \\ 1 \\ 2 \end{cases}$

for 3 function

$3^{2^2} = 3^4 = 81$

$m^{n^{p}}$
$p \to$ Numbers of variable
$n \to$ Nature of variable
$m \to$ Nature of f^n

Q) How many boolean function possible with 4 ternary variable

boolean = 2 $\to$

$2^{3^{4}} =$ ______ $\qquad 2^{3^{4}}$

$m^{n^{p}} \to$ Numbers of (m-ary) function formed by p variable of n-valued Algebra.

Q) How many boolean function possible with 3 boolean variables such that the output contents Atmost 3 minterms ***

$\to$

$^8C_0 + {}^8C_1 + {}^8C_2 + {}^8C_3 + \cancel{^8C_4} + \cancel{^8C_5}$

$= 1 + 8 + 28 + 56 = 93$

Ans:- with 8 minterm

$^8C_0 + {}^8C_1 + {}^8C_2 + {}^8C_3 + {}^8C_4 + {}^8C_5 + {}^8C_6 + {}^8C_7 + {}^8C_8$

$2^8 = 256$

no. of boolean exp. with 3 variable such that it contain

$= 256 - ({}^8C_0 + {}^8C_1 + {}^8C_2)$

$= 256 - 37$

$= \underline{\underline{219}}$

Q) How many nuted function are possible with 3 variables.

$\rightarrow$ $^{2^n}C_{2^n/2} = {}^{2^3}C_{2^{3-1}} = {}^8C_4$

How many nuetral function with n variables

(a) $^{2^n}C_{2^n/2}$ (b) $^{2^{2^n}}C_{2^n/2}$ (c) $^{2^n}C_n$ (d) None

*

Continue...

(2) **Commutative Law** ⟹ Order of inputs doesnot cha[nge] the result.

ex: → $A \cdot B = B \cdot A$ } Commutative
$A + B = B + A$

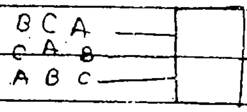
⊙ All symmetric funcTions follows commutative law
NAND, NOR, XOR & XNOR are symmetric funcn.

⊙ A symmetric funcn is not going to be chan[ged] permuting the inputs.

$f(A,B,C) = AB + BC + CA$

** $\boxed{f(B,A,C) = f(C,A,B) = f(A,B,C)}$ **

above funcn follows commutative law.



⊙ A boolean funcn is said to symmetric if contains all possible terms for each a-number (^nC_r). (The a-number of a term is total nu[mber] of 1's in its binary pattern.)

ex: → a-number of $8\ (1000)_2$ is '1'
a-number of $7\ (111)_2$ is '3'.

ex: → $f(A,B,C) = \sum(3,5,6,7) \equiv S_{2,3}(A,B,C)$ ($2,3$ → a-number)

a-number 2, 2, 2, 3 (1's in 3, 5, 6, 7)

Total possible terms with a-number '2' $= {}^3C_2 = 3$

Total possible terms with a-number '3' $= {}^3C_3 = 1$

∴ the given funcn satisfy req. for a-numb[er 2] & a-number 3 & hence it is symmetric.

S → symmetric funcn

The given funcn is symmetric as it satisfied ^nC_r requirement for each a-number & hence this funcn follows commutative law.

ex:→ $S_{0,3}(A,B,C) = f(A,B,C) = \sum(0,7)$ $\left.\right\}$ $^3C_0 = 1$, $^3C_3 = 1$

$= A'B'C' + ABC$

Que) Check whether following funcn is symmetric or not and if its not symmetric what are the minimum no. of terms to be added to make the funcn as symmetric.

$\underbrace{f(A,B,C,D)}_{n=4} = \sum(1, 2, 3, 4, 6, 8, 10, 12, 15)$

a-num ↓ 1, 1, 2, 1, 2, 1, 2, 2, 4

$^4C_1 + {}^4C_2 + {}^4C_4$ → total term

$4 + 6 + 1 = 11$

$^4C_1$ satifies i.e. 4 1's present & $^4C_4$ also satisfied

But $^4C_2$ not satisfy as $^4C_2 = 6$ & term are only 4

∴ not symmetric

The funcn is symmetric if thee term 5 & 9 are added.

$f(A,B,C,D) + \sum(5,9)$ } as minimum term to be added ∴ 5, 9

$0 \oplus x = 0'x + 0x'$
$= 1 \cdot x = x$
$1 \oplus x = 1'x + 1 \cdot x'$
$= 0 + x' = x'$

(3) Associative law ⟹ Order of operation doesnot change the outcome or result.

ex:→ $A+(B+C) = (A+B)+C$ } Associative law.
$A \cdot (B \cdot C) = (A \cdot B) \cdot C$

(i) NAND ⟹ Prove $A \uparrow (B \uparrow C) = (A \uparrow B) \uparrow C$

$B' + C'$ | $A' + B' + C' = AB + C'$

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

$A' + BC$

$A' + BC \neq AB + C'$ } Not follow Associative law

∴ NAND, NOR doesnot follow associative law.
XOR follow Associative property.

Q. Which of the following operation is commutative associative, But not distributive.
(a) AND (b) NAND (c) OR (d) XOR

(i) XOR ⟹ $A \oplus (B \oplus C) = (A \oplus B) \oplus C$

fixed A variable on both sides. (any variable can fixe

	A = 0	A = 1
LHS	$B \oplus C$	$B \odot C$
RHS	$B \oplus C$	$(B \oplus C)' = B' \oplus C = (B')'C + B'.C'$

$= BC + B'C'$
$= B \odot C$

$\boxed{0 \oplus x = x}$
$\boxed{1 \oplus x = x'}$

$\boxed{0 \odot x = x'}$
$\boxed{1 \odot x = x}$

∴ LHS = RHS

XOR, XNOR follows ~~commutative~~ associative law.

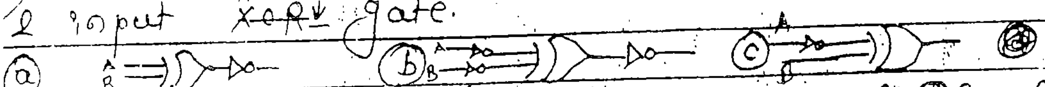
$\boxed{A \oplus B = A' \odot B = A \odot B' = A' \oplus}$

$\boxed{A \odot B = A' \oplus B = A \oplus B' = A' \odot}$

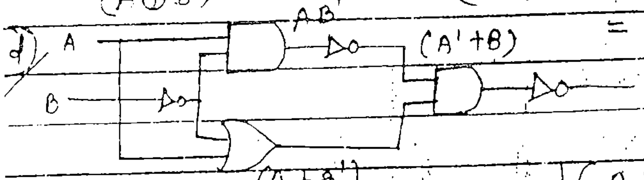
$A' \oplus B' = (A')' \cdot B' + A' \cdot (B')'$
$= AB' + A'B$
$= A \oplus B$

$\boxed{A \oplus A = 0}$

1) Which of the following is not equivalent to 2 input ~~XOR~~ XNOR gate.



(a) $(A \oplus B)' =$ (b) $(A' \oplus B')' = (A \oplus B)'$ $= A \odot B$ (c) $A' \oplus B = A \odot B$ (d)



$[(A'+B)(A+B')]'$
$= (A'+B'+AB)'$
$= A \oplus B$

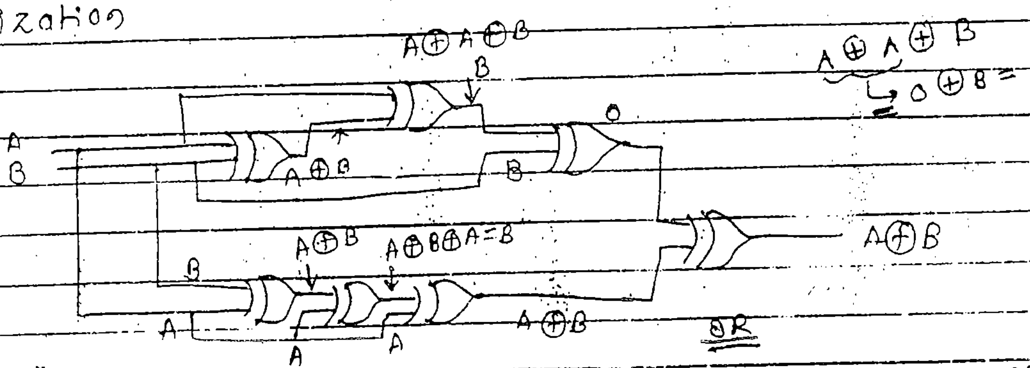
2) Let P denotes $P = A \oplus B$, what's the value of the expression? $P \oplus A \oplus P \oplus A \oplus P \oplus B \oplus A$

(a) 0 (b) 1 (c) A (d) B

⇒ XOR is commutative & associative.

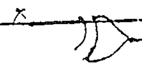
$P \oplus P \oplus P \oplus A \oplus A \oplus A \oplus B$ $\boxed{\because A \oplus A = 0}$
$0 \oplus P \oplus 0 \oplus A \oplus B$
$0 \oplus A \oplus B \oplus 0 \oplus A \oplus B = 0 \oplus A \oplus A \oplus B \oplus B \oplus 0$
$0 \oplus 0 \oplus 0 \oplus 0$

3) What was the value given by the following realization



$\underbrace{A \oplus A}_{} \oplus B$
$\hookrightarrow 0 \oplus B = B$

OR



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

Distributive Law ⇒ This law is excluively in the minimisation.

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

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

The distributive law is available in two forms:-

(i) Combining law.
(ii) Absorption law

(i) Combining law ⇒ $AB + AB' = A$

$(A+B)\cdot(A+B') = A$

(ii) Absorption law ⇒ $A + AB = A$

$A\cdot(A+B) = A$.

ex:→ $(\underbrace{xy}_{P} + \underbrace{A'B+AB}_{Q})$ $(A'B+AB)' = A'B + AB'$

$(\underbrace{xy}_{P} + \underbrace{A'B+AB'}_{Q'})$

que.) The simplified SOP expression of the follo

$(P+\overline{Q}+\overline{R})\ (P+\overline{Q}+R)\ (P+Q+\overline{R})$

(a) $\overline{P}Q+\overline{R}$ (b) $P+\overline{Q}\overline{R}$ (c) $\overline{P}Q+R$ (d) $PQ+R$

⇒ $(\underbrace{P}_{A}+\underbrace{\overline{Q}+\overline{R}}_{B})\ (\underbrace{P+\overline{Q}}_{A}+\underbrace{R}_{C})$

$(P+(\overline{Q}+\overline{R})(\overline{Q}+R))$

$(P+\overline{Q})\cdot(P+\overline{Q}R) = (\underbrace{P}_{A}+\underbrace{\overline{Q}}_{B})\ (\underbrace{P}_{A}+\underbrace{\overline{Q}R}_{C}) =$

OR

$P + ((\bar{Q}+\bar{R})(\bar{Q}+R)(Q+\bar{R}))$

The behaviour of XOR on distributive property

1. XOR is not distributive on any other operation including itself.

2. No other operation is distributive on XOR except AND operation

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

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

(iii) $\boxed{A \cdot (B \oplus C) = AB \oplus AC}$

If in above case $A = 1$

$$\left\{\begin{array}{l} \text{LHS } (B \oplus C)' = B \odot C \\ B' \oplus C' = B \oplus C \\ A = 1 \end{array}\right.$$

RHS = 1

RHS = $1 \oplus 1 = 0$

$A = 0$	$A = 1$
LHS = 0	LHS = $B \oplus C$
RHS = $0 \oplus 0 = 0$	RHS = $B \oplus C$

SOP gives NAND-NAND & it represented by A

De-Morgans Law

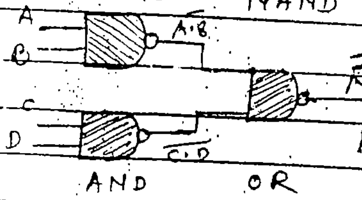
Special form of complimentation

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

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

- NAND $\rightleftharpoons$ Bubbled OR
- Any AND-OR realization is repeated with NAND-NAND (SOP)
- Basic realization $\rightleftharpoons$ Universal realization
- Bubbled NAND $\rightleftharpoons$ OR gate

(1) NAND - NAND



$\overline{\overline{A\cdot B}\cdot\overline{C\cdot D}}$

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

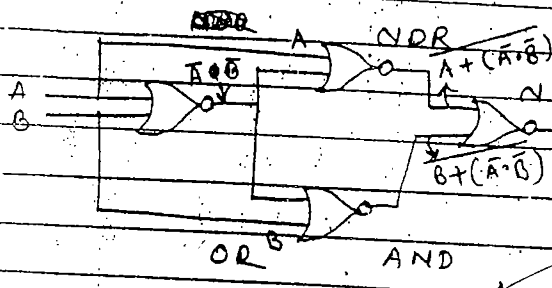
(a) $\overline{A\cdot B}+\overline{CD}$ $\Rightarrow$

(b) $AB + CD$

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

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

(2) funcn represented by



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

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

2 i/p NOR

XNOR

$$(\overline{A}\cdot\overline{B}+A)\cdot(\overline{A}\cdot\overline{B}+B)$$

$$(A+\overline{A})\cdot(A+\overline{B})\cdot(\overline{B}+B)\cdot(\overline{A}+B)$$

$$1\cdot(A+\overline{B})\cdot 1\cdot(\overline{A}+B)$$

$$(A+\overline{B})\cdot(\overline{A}+B)$$

$$(AB+\overline{A}\cdot\overline{B})$$

~~Note~~ → Minimum number of 2 input NOR gates required to realize XNOR = 4

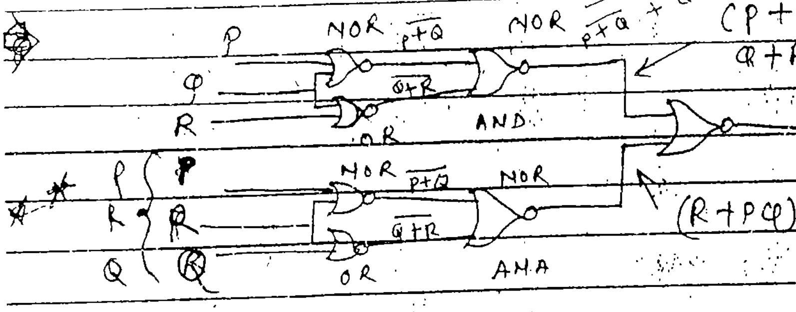
$A(A+B)$

$A + AB = A(1+B)$

$= A \cdot 1 = A$

Minimum number of 2 input NAND gates required to realize XOR ⇒ 4

Q) What will be the funcn represented by the following realization?



$\overline{P+Q} + \overline{Q+R}$ ⇒ $(P+Q)\cdot(Q+R)$

$Q + PR = x$

$\bar{x} \cdot \bar{y} = (Q+PR)' \cdot (R+PQ)'$

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

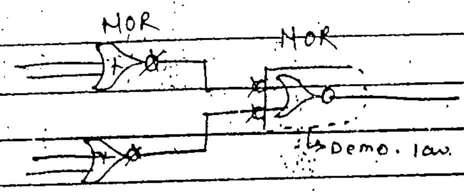
$Q'P' + Q'R'$

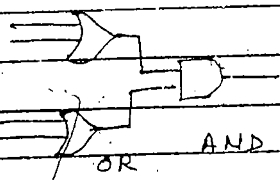
$= Q'(P'+Q') \, R'(P'+R')$

$= Q'R'$

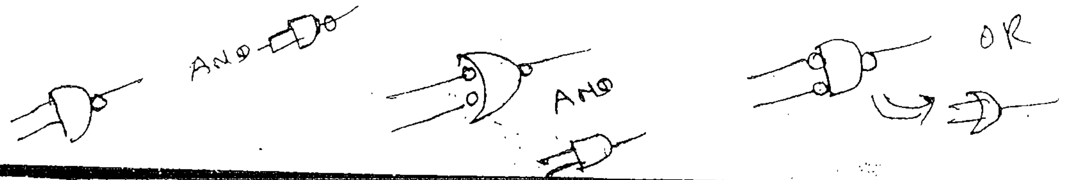
$(R+PQ) = y$

How NOR-NOR replaced with OR-AND? $A(A+B) = A$



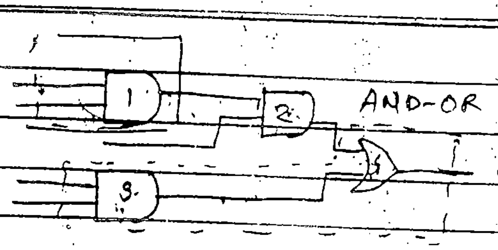


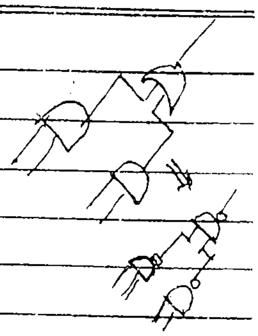
Q2) What will be the minimum no. of two input NAND gates required to realised the following one

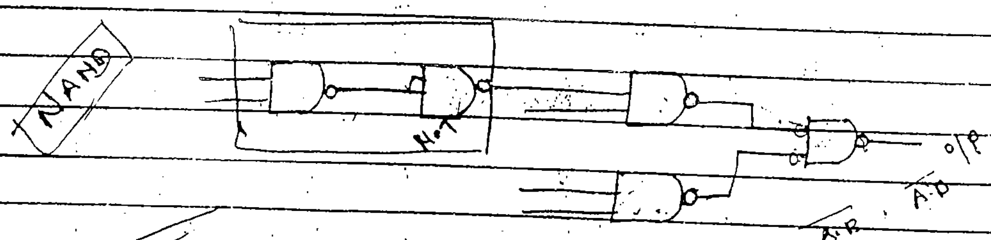


(AB)'

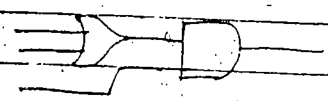
AND-OR replaced with NAN



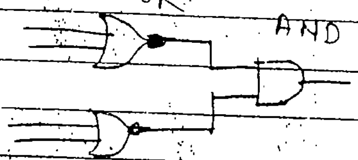




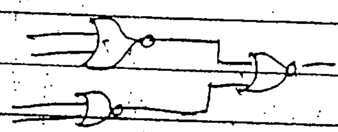
Q. Minimum no. of NOR gates required to realised following



Addnal OR gate is required



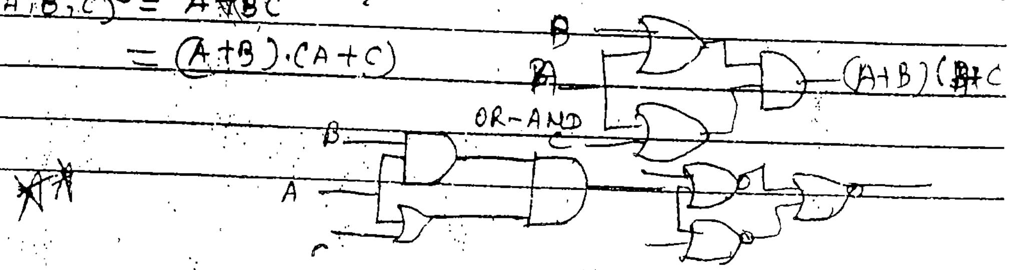
then convert to NOR NOR



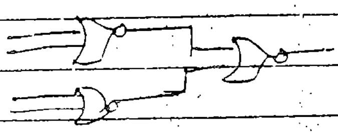
Q. Minimum of 2 i/p NOR gate req. to realise

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

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



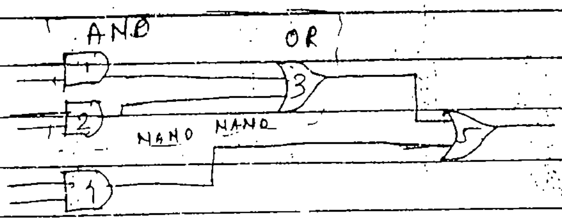
NOR - NOR



3 NOR gate required to realised it.

8) What will be the minimum no. of two i/p NAND gates required to realised the following funcn

(a) 4 (b) 5 (c) 6 (d) 7



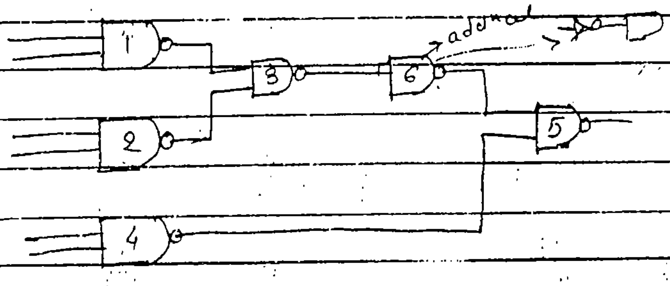
Soln.

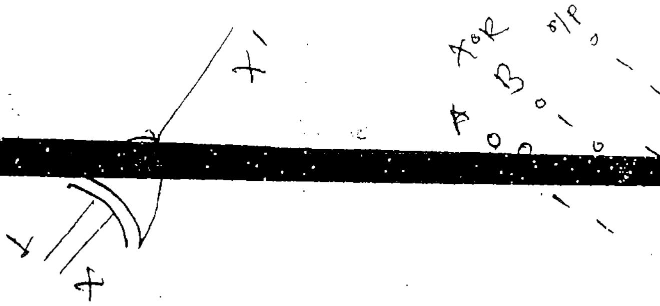
NAND = Bubble OR

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

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

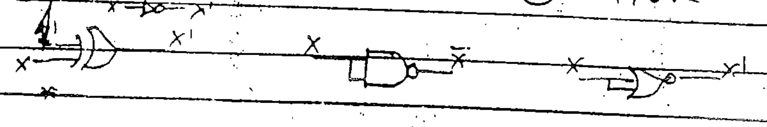
Bubbled NAND-OR





Q.) which of the following can't be used as an inv

(a) XOR (b) NAND (c) NOR (d) None of above



* Compensation Theorem *

To identify redundant term & remove them.

$$AP_1 + A'P_2 + \underbrace{P_1 P_2}_{\text{redundant}} = AP_1 + A'P_2$$

$$(A+P_1)\cdot(A'+P_2)\cdot\underbrace{(P_1+P_2)}_{\text{redundant}} = (A+P_1)\cdot(A'+P_2)$$

*

$P_1 \cdot P_2 = 0$ — LHS = RHS

$P_1 P_2 = 1 \Rightarrow P_1 = P_2 = 1$

LHS: $A + A' + 1 = 1$

RHS: $A + A' = 1$

As LHS = RHS in both case hence it's redundant.

Q.) $(A'+BC)(B'+CA)(C'+AB)$

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

$= (A+B')\,\cancel{(A'+C)}\,(B'+C)\,(A+C')\,(A'+B)\,(B+C')$ — redundant { from of com

$= (A+B')(A+C')\,(A'+C)\,(A'+B)$

$= (A+B'C')\cdot(A'+BC) = \underset{0}{AA'} + ABC + A'B'C' + \underset{0}{B'C'BA}$

$= A'B'C' + ABC$

Limitations of Boolean Laws

- The boolean laws ~~unable~~ are able to provide minimization but it is not possible to know which boolean law is to be applied when. If the laws are not substituted in proper order, the expression may endup in expansion form instead of minimization.

- We are not sure that the final expression is in redundant form or irredundant form. These problems are resolved using k-maps.

K-Maps

K-maps provides systematic way of minimization. It denotes the truth table in pictorial form. A k-map of n-variables contains 2^n cells. Each of which is addressed by gray code.

- The adjecent cells containing minterms or maxterms form prime implicants.

- A prime implicant can't be subset of another. And it becomes essential if it exclusively covers (not share by anybody) atleast one minterm.

- If a map contains all essential prime implicants then it will have only one minimal form.

- If none of the prime implicants are essential (cyclic implicants) the map will have more than one minimal expression.

- If the no. of variables are 6 ~~or~~ more construc of K-maps is difficult & hence minimizatn also difficult. ~~As~~ an alternative method tabulation method, variable entrant map are used
- The minimal expression also called as "Minimal Cover." It must be minimum & covers all minterms.

Minimal expression = Minimal cover = All Essential prime-implicant + Prime Implican required to cover remai minterm, if a

① $F(A,B,C) = \sum(1,2,3,4,6)$

C \ AB	00	01	11	10
0	[0]	1 [2]	1 [6]	*1 [4]
1	1* [1]	1 [3]	[7]	[5]

essential prime imp. P Q R S

Basic terms

① How many prime implicants? $\longrightarrow$ 4 i.e. P, Q, R, S (pairs) no!
② How many essential prime implicants? P, S
③ How many redundant prime implicant? Q, R
④ How many minimal expreo?
⑤ How many literals?

② ⇒ 2 are essential prime implicant P and S.
becoz it contain essential '1' not cover by other.

③ ⇒ Q, R are redundant prime implicant

$(1, \overbrace{②, 3}^{P}, \overbrace{4, 6}^{S})$

④ ⇒ 2 minimal expressions $\boxed{P+S}+Q$, $\boxed{P+S}+R$ to cover left.

prim implicant

$P+S+Q = A'C + AC' + A'B$

$P+S+R = A'C + AC' + BC'$

⑤ $P+S+Q = \boxed{\underset{2}{A'C} + \underset{2}{AC'} + \underset{2}{A'B}}$, $P+S+R = \underset{2}{A'C} + \underset{2}{AC'} + \underset{2}{BC'}$

No. of literals in SOP is $2+2+2 = 6$.
though it simplify $\boxed{A'(B+C) + AC'}$ & give literal $\boxed{5}$
~~then~~ but we have asked SOP so its 6.

for above quesns maximization is ⇒

C \ AB	00	01	11	10
0	0 [0]	[2]	[6]	[4]
1	[1]	[3]	0 [7]	0 [5]

P, Q

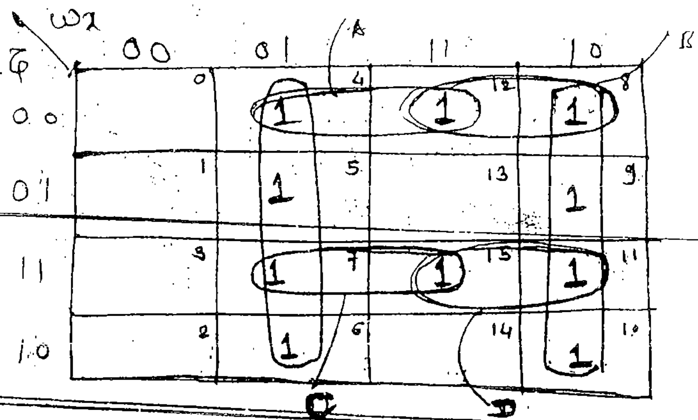
① ⇒ 2 prime implicants for POS. P & Q

② ⇒ 2 prime imp. which is essential

③ ⇒ No redundant.

④ ⇒ POS have one min. exp.

P.T.O



(5) ⇒ pos have 5 literals $\underbrace{(A+B+C)}_{3} \cdot \underbrace{(A'+C')}_{2}$

expansion of pos ⇒ $(A+B+C)\cdot(A'+C')$

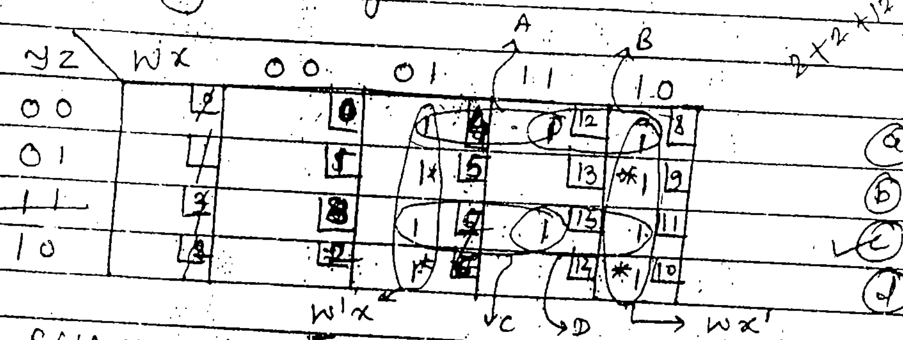
$= AC' + \underbrace{A'B + BC'}_{\text{compensati}} + A'C$ or

$= \cancel{AC' + A'B + A'C}$

$= AC' + A'B + \underbrace{BC' + A'C}_{\text{compensatn}}$

$POS = AC' + BC' + A'C = SOP$

(7) How many minimal expressions possible for following k-map?



(a) 3
(b) 4
(c) 6
(d) 7

z) $f(w,x,y,z) = \sum(4,5,6,7,8,9,10,11,12,15)$

prime implicants are 6

essential Prime Implicants 2 (Quads)

redundant Prime implicants 4 (pairs)

minimal expression ⇒ All essential prime implicant + 12 + 15

to cover 12 A or B

covers C

All Esential prime Implicants +A+C

+A+D

+B+C

+B+D

∴ 4 minimal exp.

Literals are ⇒ (2+2+3+3) = 10

from quad element ~~2 variables~~ so 4-2=2

from other 1 ∴ 4-1=3

$w'x + w'x + xy'z' + xyz$ — A+C

$w'x + w'x + xy'z' + wyz$ — +A+B

$w'x + w'x + wy'z' + xyz$ — +B+C

$w'x + w'x + wy'z' + wyz$ — +B+D

$f(w,x,y,z) = \Sigma(4,5,6,7,8,9,10,11,12,15)$

$\therefore f(w,x,y,z) = \Pi(0,1,2,3,13,14)$

yz \ wx	00	01	11	10
00	0* [0]	[4]	[12]	[8]
01	0 [1]	[5]	0* [13]	[9]
11	0 [3]	[7]	[15]	[11]
10	0 [2]	[6]	0* [14]	[10]

$w'x'$ $wxy'z$ $wxyz'$

1) ⇒ primer implicants are 3.

2) ⇒ essential prime implicants are also 3

3 ⇒ redundant are 0:

4 ⇒ minimal expression is 1

5 ⇒ Literals are 2+4+4=10

Q2 Identify which of the following is not prime implicant?

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

(a) BD *
(b) A'C'D
(c) A' BC *
(d) ABD ✓

⇒ prime implicant can't be subset of another & essi cover minterm.

As ABD is a subset of BD i.e. pair within the quad.

∴ ABD not prime implicant.

dual of

3) ~~With~~ respect to the number of literals of minimal SOP & POS identify the correct statement.

(a) The number of literals in minimal SOP is 1 more than POS.

(b) The number of literals in minimal SOP is 1 less than POS.

(c) The difference between minimum number of literals in SOP & POS is 2.

(d) The number of literals are equals in both SOP & POS.

⇒ from que. 2

SOP literals 4 * 3 = 12

prime I. 5 → (4-P, 1-Q)

** E.P.I 4 → (4-P)

Redunt. ~~(10)~~ 1 ⇒ P.I − E.P.I = 5−4 = 1

Exp. literals 1

∴ 3 * 3 * 3 * 3 = 12

POS literals 3 * 4 = 12

SOP expression of que 2 ⇒ A'C'D + A'BC + ABC' + ACD

(A+C'+D)(A'+B+C)(A+B+C')(A+C+D)

POS expression ⇒ AC'D + A'BC + AB+C + ACD

POS dual of SOP

Essential P.I. check whether term present in k-map

Essential P

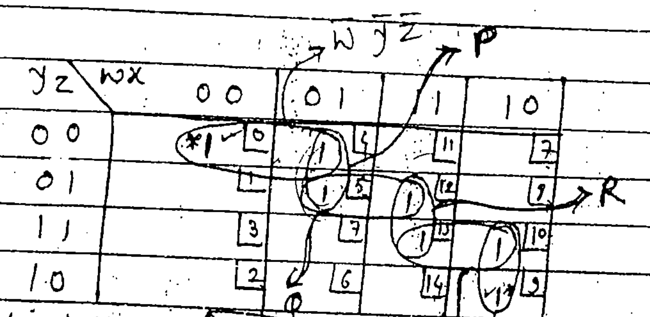
(4) Consider hypothetical map in which E.P.M. are covering all the minterms except 2. Each of the leftover minterm is covered by three different redundant prime implicants. What will be the number of minimal expressions denoted by this map?

(a) 3 (b) 6 (c) 9 (d) 12

Ans. $\Rightarrow$ minimal cover = All E.P.M. + $\underbrace{P \text{ or } Q \text{ or } R}_{m_1}$ + $\underbrace{S \text{ or } T \text{ or } U}_{m_2}$

P+S Q+S R+S
P+T Q+T R+T
P+U Q+U R+U

(5) Which of the following is not represented by k-map?



(a) $\boxed{w'y'z'} + xy'z + wyz + \boxed{wx'y}$ → E.P.I.
(b) $w'y'z' + wx'y + w'xy' + wxz$
(c) $wx'y + xy'z + wyz + wxz$ → as essential prime implicant is missing. hence not in k-map
(d) $wx'y + w'y'z' + xy'z + wxz$

5) What will be the no. of minimal exp. represented by above map?

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

⇒ Ess + P + R
Ess + Q + R
Ess + Q + S

Minimality should be protected & not violated.

6) A three variable function f(A, B, C) is minimised as f(A, B, C) = A + BC. The minterms of f are 3, 5, 6. Which of following don't cares (X) are used in the minimisation?

(a) $\sum_{\phi}(2,7)$ (b) $\sum_{\phi}(4,7)$ (c) $\sum_{\phi}(2,4,7)$ (d) $\sum_{\phi}(2,4)$

⇒

C \ AB	00	01	11	10
0	0	2	1 (6)	φ (4)
1	1	1 (3)	φ (7)	1 (5)

BC, A

7) What will be the values of don't cares P, Q, R if the k-map is minimised?

⇒

	00	01	11	10
00	0 (0)	0 (4)	P (12)	0 (8)
01	0 (1)	0 (5)	1 (13)	1 (9)
11	1 (3)	1 (7)	1 (15)	1 (11)
10	0 (2)	0 (6)	R (14)	0 (10)

	P	Q	R
(a)	0	0	0
(b)	1	0	0
(c)	1	1	0
(d)	1	1	1

As octate is happens here POS ∴ all are 0.

8) The funcⁿ represented by following k-map is free from :- (a) 1 variable (b) 2 variables (c) 3 variables (d) dependent on all

yz \ wx	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

$x'z$ $\boxed{x'z + xz'}$

xz'

Application on k-maps

Let, $A \$ B = A' + B$ what will ~~be~~ the expression denoted by $f_1 \$ f_2$.

f_1

R \ PQ	00	01	11	10
0	0 ⁰	1 ²	0 ⁶	0 ⁴
1	1 ¹	0 ³	1 ⁷	0 ⁵

f_2

R \ PQ	00	01	11	10
0	0 ⁰	1 ²	1 ⁶	0 ⁴
1	1 ¹	0 ³	0 ⁷	1 ⁵

$f_1(A,B,C) = \Sigma(0,1,7) + \sum_{\phi}(2,4)$

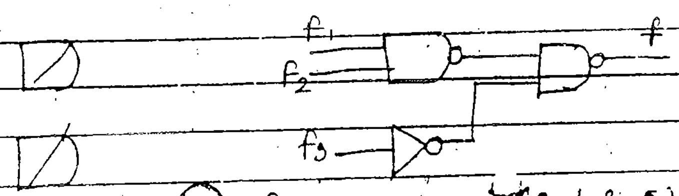
$f_2(A,B,C) = \Sigma(1,2,3,6) + \sum_{\phi}(0)$

1) $f_1 + f_2 = ?$

2) $f_1 f_2 = ?$

~~$\$B = A'+B$~~

AND OR



(B) $f_1(x,y,z) = \Sigma(0,1,3,5)$

$f_2(x,y,z) = \Sigma(1,6,7)$

$f(x,y,z) = \Sigma(1,4,5)$

$f_3 = ?$

Sol^n $A \$ B = A'+B$

$\Rightarrow$ $A \$ B = 0$

iff $A' = 0$ & $B = 0$
($\because A = 1$)

else $A \$ B = 1$

where 1st var is 1 & 2nd var is 0 then \$ obtain

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

else 1

R \ PQ	00	01	11	10
0	1	1	1	1
1	1	1	0	1

$\therefore$ $f_1 \$ f_2 = (P'+Q'+R')$

A	B	f_0	f_1	f_2	... f_6 ...	f_{14}	f_{15}
0	0	0	0	0	0	1	1
0	1	0	0	0	1	1	1
1	0	0	0	1	1	1	1
1	1	0	1	0	0	0	1
		NULL	AND	OR		NAND	Tautology

$f_1 = \Sigma(0,1,7) + \Sigma_{\phi}(2,4$

$f_2 = \Sigma(1,2,3,6) + \Sigma_{\phi}(0)$

~~1~~ ⇒ $\phi + \overset{\text{maxterm}}{0} = \phi$ —— (a)

The term must be in ϕ list of one funcn & ma of another function for OR operation th it's in don't care list.

$\phi \cdot \overset{\text{minterm}}{\phi} = \phi$ —— (b)

The ~~trem~~ term must be in ϕ list of one funcn & minter of another funcntn for AND.

∴ from case (a)

$$f_1 + f_2 = \Sigma(0,1,2,3,6,7) + \Sigma_{\phi}(4)$$

from case (b)

$$f_1 f_2 = \Sigma(1) + \Sigma_{\phi}(0,2)$$

$\phi \cdot 1 = c$

0 1 2 3 4

dont cure

$f_1 . f_2 = BA'$

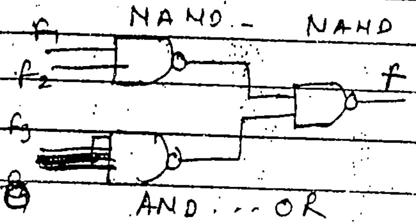
ex: →

C \ AB	0 0	01	11	10
0	1	1	1	ϕ
1	1	1	1	0

$f_1 + f_2 = A' + B$

(8) ⇒

NAND - NAND



AND ... OR

$$\boxed{f = f_1 . f_2 + f_3}$$

$f_1 . f_2 = \Sigma(1)$, $f = \Sigma(1,4,5)$

$f_3 <$ $\Sigma(1,4,5)$ / $\Sigma(4,5)$

?

by formula $f = \Sigma(1) + \Sigma(1,4,5) = \Sigma(1,4,5)$

$f = \Sigma(1) + \Sigma(4,5) = \Sigma(1,4,5)$

Q) It is required to realised given within Basic gates (AND, OR) the only constraints is the realisation used only one inverter with respect to the following k-map identify the correct statement.

yz \ wx	00	01	11	10
00	0	1	1	0
01	1	0	0	1
11	1	0	0	1
10	0	1	1	0

$f_1 = xz' + x'z \Rightarrow$

z \ xy	00	01	11	10
0	1	1	0	1
1	1	1	0	0

$f_2 = x' + y' \Rightarrow xy$

$(x'+y')' = xy$

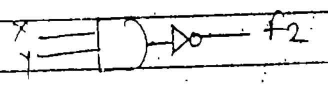
(a) only f_1 is realisable.
(b) only f_2 is realisable.
(c) Both f_1 & f_2 are realisable
(d) Neither f_1 nor f_2 are realisable.

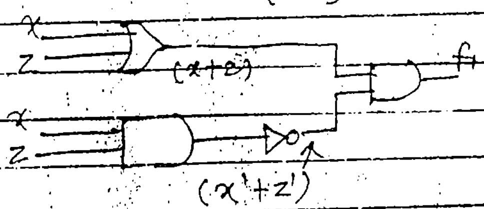
$\Rightarrow$ $f_2 = xy$

$f_1 = xz' + x'z$

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

$(xz)'$





00 01 11 10

00
01
11
10

Q. Consider the following three 3-variable fun.

S, P $f_1(A,B,C) = \sum(0,1,3)$

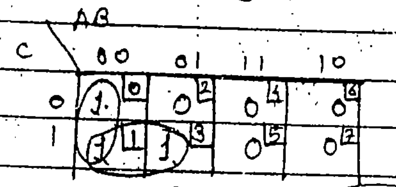
R, Q $f_2(A,B,C) = \sum(3,5,7)$

Q, P $f_3(A,B,C) = \pi(1,3,7)$

In the simplification it was found there are 4 prime implicants P, Q, R, S (all are pairs). The prime implicant P is common to f_1 & f_3 and Q is common to f_2 & f_3. What will be the expressions for prime implicants Q, R, S?

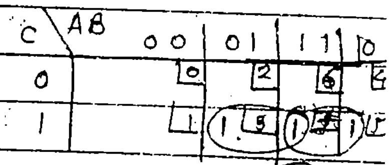
R is the prime implicant of f_2 while S is prime implicant of f_1

⇒

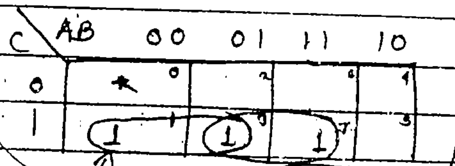


$f_1 = A'B' + A'C$

S



$f_2 = BC + A$



Q

$f_3 = A'C + BC$

P

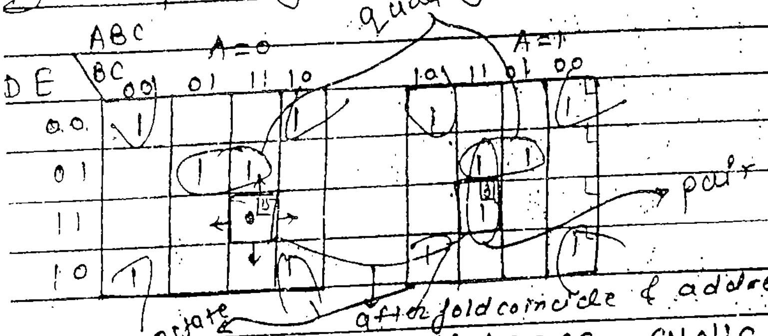
P	Q	R	S
A'C	BC	AC	A'B'

(*) The more the number of variables then it more difficult to construct prime implicant.

one bit bin & gray are same

(A)

Simplify following $\boxed{\text{5-variable k-map}}$



Gray code :→ Unit distance, cyclic, self reflection

0	0	0
1	0	1
	1	1
	1	0

2-bit gray code

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

3 bit

$f(A,B,C,D,E) = \Sigma(0, 2, 8, 10, 13, 16, 18, 21, 24, 26, 29, 31)$

octet = $C'E'$

Quad = $CD'E$

Pair = $A\bar{B}CE$

VEM

VEM (Variable Entrant Map)

- It's capable of representing higher variable funct in a lesser variable map.

C \ AB	00	01	11	10
0	1	1	φ	D'
1	D	D	0	D'

f (A, B, C, D)

- Here the contains of cells can be variables.

* Procedure for Minimisation *

~~Step~~ 1 ⇒ Make all the variables in the cell as 0 (Zero) construct SOP.

~~Step~~ 2 ⇒

(a) Make one variable (Minterm (1)) high at a time & obtain SOP treating earlier ~~mean~~ Minterms as don't care.

(b) Multiply above SOP with the concerned variable.

~~Step~~ 3 ⇒ Repeat the step (2) untill all the variables in cell are covered.

~~Step~~ 4 ⇒ SOP of VEM is obtained by ORing previous 3

Ex: (1)

C \ AB	00	01	11	10
0	D	1	0	0
1	D	1	ϕ	$\bar{D}$

Step 1 ⇒

C \ AB	00	01	11	10
0	0	1	0	0
1	0	1	ϕ	0

SOP = $\bar{A}B$

Step 2 (a) ⇒ consider D as high

C \ AB	00	01	11	10
0	1	ϕ	0	0
1	1	ϕ	ϕ	$\bar{D}$

SOP = $\bar{A}$

Step 2 (b) ⇒ As we considered D ∴ SOP = $\bar{A} \cdot D$

Step 2 (a) for $\bar{D}$

C \ AB	00	01	11	10
0	0	ϕ	0	0
1	0	ϕ	ϕ	1

SOP = AC

Step 2 (b) for $\bar{D}$ ⇒ SOP = $A \cdot C \cdot \bar{D}$

∴ $f(A,B,C,D) = \bar{A} \cdot D + AC\bar{D} + \bar{A}B$

$= \Sigma(1,3,4,5,6,7,10,14)$

	A	B	C	D	
$\bar{A}D$ →	0	ϕ	ϕ	1	→ 1,3,5,7 main terms
$AC\bar{D}$ →	1	ϕ	1	0	10,14
$\bar{A}B$ →	0	1	ϕ	ϕ	4,5,6,7

② Ans :→ $f(A,B,C,D) = \bar{B}C + \bar{C}D + C$

Right Correct

Ex ②

B\AC	00	01	11	10
0	D	1	1	D
1	ϕ	$\bar{D}$	$\bar{D}$	0

⇒ Step 1 ⇒

B\AC	00	01	11	10
0	0	1	1	0
1	ϕ	0	0	0

$\boxed{f = \bar{B}C}$

Step 2 ⓐ ⇒ cosider D as high

B\AC	00	01	11	10
0	1	ϕ	ϕ	1
1	ϕ	$\bar{D}$	$\bar{D}$	0

$f = \bar{B}$

Step 2 ⓑ $\boxed{f = \bar{B}\cdot D}$

Step 2 ⓐ → consider $\bar{D}$ as high

B\AC	00	01	11	10
0	D	ϕ	ϕ	D
1	ϕ	1	1	0

$f = C$

Step 2 ⓑ → $\boxed{f = C\cdot\bar{D}}$

∴ $f(A,B,C,D) = B'C + B'D + CD'$ Correct

8 A	4 B	2 C	1 D		
ϕ	0	1	ϕ	B'C	(6,11)
ϕ	0	ϕ	1	B'D	(1,11)
ϕ	ϕ	1	0	CD'	(2,6,

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

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

P.3)

D\C	0	1
0	$A \oplus B$	0
1	0	$A \odot B$

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

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

$\Rightarrow \bar{A}B\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}CD + ABCD$

$\Sigma(8, 4, 3, 15)$

P.4) Consider the following 3-variable funcn $f(A,B,C) = \Sigma(0,1,2,4,5,7)$ is realised by the following VEM. What could be the values of PQRS.

$\Rightarrow f(A,B,C) = \Sigma(0,1,2,4,5,7)$

$= A'B'C' + A'B'C + A'BC' + AB'C' + AB'C + ABC$

Given

B\A	0	1
0	P	R
1	Q	S

Given

$= \bar{A}\bar{B}\cdot P + \bar{A}BQ + A\bar{B}R + AB\cdot S$

$\boxed{P = C' + C = 1}$

$\boxed{P = 1}$

$\boxed{Q = C'}$

$\boxed{S = C}$

$\boxed{R = 1}$

B\A	0	1
0	1	R
1	Q	S

$\boxed{Q = C'}$ $\boxed{R = C' + C = 1}$ $\boxed{S = C}$

B\A	0	1
0	1	1
1	C'	C

Apply VEM

step 1 ⇒

B\A	0	1
0	ϕ	1
1	0	0

$\boxed{f = B'}$

Step 2 (a) → make C as high

B\A	0	1
0	ϕ	ϕ
1	C'	1

B\A	0	1
0	ϕ	ϕ
1	C'	1

$f = A$

(b) → $\boxed{f = AC}$

Step 3 (a) → make C' as high

B\A	0	1
0	ϕ	ϕ
1	1	C

$f = A'$

(b) → $\boxed{f = A'C'}$.

$f = B' + AC + A'C'$

	A	B	C
$\bar{B}$	ϕ	0	ϕ

{0, 1, 4, 5}

	A	B	C
$\bar{A}\bar{C}$	0	ϕ	0

{0, 2}

	A	B	C
AC	1	ϕ	1

{5, 7}

Time Complexity
$O(2^n)$

* Tabulation Method *

This method is used to generate all possible minimal expressions & prime implicants. The time complaxity of tabulation process $O(2^n)$ (exponential time complexity).

Procedure for minimisation →

Step 1 → Rearrange the minterms according to the number of 1's in its binary pattern.
ex:→ the min-term 8 is placed in group 1. while 7 is placed in group 3.

$8(1000)_2 \rightarrow G_1$ G_1

$7(111)_2 \rightarrow G_3$ G_3

(2) Match the minterms of adjecent groups which are at unit distance & form the new group.

(3) Repeat the step (2) exhaustively until no further new groups are possible.

(4) Use the implication chart & extract all minimal expressions.

Distance = no. 0-1 pair

① f (A,B,C,D) = Σ (0, 1, 2, 4, 8, 12, 14, 15)

⇒ Step 1 →

0 0 0 0 (0) → G_0

0 0 0 1 (1)

0 0 1 0 (2)

0 1 0 0 (4) → G_1

1 0 0 0 (8)

1 1 0 0 (12) → G_2

1 1 1 0 (14) → G_3

1 1 1 1 (15) → G_4

0 0 0 _ (0, 1) P G_{01} P

0 0 _ 0 (0, 2) Q G_{01} Q

0 _ 0 0 (0, 4) G_{01} G_{01}

_ 0 0 0 (0, 8) G_{01}

_ 1 0 0 (4, 12) G_{12}

1 _ 0 0 (8, 12) G_{12}

1 1 _ 0 (12, 14) R G_{23} R

1 1 1 _ (14, 15) S G_{34} S

_ _ 0 0 (0, 4, 8, 12) T G_{012} T

Nothing (φ) G_{0123}

φ G_{234}

The (✓) are not valid prime implicant. P, Q, R, S are prime implicant & they form pair.

T form the quad.

~~Implicant chart~~

Prime Implicants	0	1	2	4	8	12	14	15
* P	✓	✓						
* Q	✓		✓					
R						✓	✓	
* S							✓	✓
* T	✓			✓	✓	✓		

when one tick (✓) in column its essential P. Implic.

minimal exp. = All essent + other

$\Sigma(0, 1, 3, 4, 8, 12, 14, 15)$

∴ one ~~minterm~~ minimal exp only P+Q+S+T

$P+Q+S+T = A'B'C' + A'B'D' + ABC + C'D'$

(2) Using tabulation method minimize the following

$F(A, B, C, D, E) = \Sigma(0, 2, 5, 8, 10, 13, 16, 18, 24, 26)$

0 0 0 0 0	(0)	G_0	0 0 0 – 0	(0, 2) ✓	G_{01}	(0, 2)
0 0 0 1 0	(2)		0 – 0 0 0	(0, 8) ✓		(0, 8)
0 1 0 0 0	(8)	G_1	– 0 0 0 0	(0, 16)		(0, 16)
1 0 0 0 0	(16)		0 – 0 1 0	(2, 10) ✓	G_{12}	
0 0 1 0 1	(5)		– 0 0 1 0	(2, 18)		
0 1 0 1 0	(10)	G_2	0 1 0 – 0	(8, 10) ✓		
1 0 0 1 0	(18)		– 1 0 0 0	(8, 24) ✓		
1 1 0 0 0	(24)		1 0 0 – 0	(16, 18) ✓		
1 1 0 1 0	(26)	G_3	1 – 0 0 0	(16, 24) ✓		
0 1 1 0 1	(13)		0 – 1 0 1	(5, 13)	P	
			– 1 0 1 0	(10, 26) ✓	G_{23}	
			1 – 0 1 0	(18, 26) ✓		
			1 1 0 – 0	(24, 26) ✓		

Higher value in grp ≠ lower val in high grp

∴ $G_1 8 \neq 5\ G_2$

0 - 0 - 0 (0, 2, 8, 10) ✓
- 0 0 - 0 (0, 2, 16, 18)
- - 0 0 0 (0, 8, 16, 24) G_{1012}
- 1 0 - 0 (8, 10, 24, 26)
1 - 0 - 0 (16, 18, 24, 26) ✓ G_{1123}
- - 0 1 0 (2, 10, 18, 26)

- - 0 - 0 (0, 2, 8, 10, 16, 18, 24, 26) ⟶ φ

	0	2	5	8	10	13	16	18	24	26
*P			✓			✓				
*φ	✓	✓		✓	✓		✓	✓	✓	✓

Both prime implicant are essential & hence single mint

A'CD'E + C'E'

$P + \varphi \Rightarrow A'CD'E + C'E'$

Reduction of Implication Charts

- The implication charts is reduced by ~~rego~~ retaining the dominant rows & removing the dominant column.
- A row P is said to be dominating φ if it covers all the minterms of φ in addition it having some more minterms. ($\varphi \subset P$)
- The tickmark (✓) concept in tabulation process is to care about row dominancy.

The reduction process has reduced two prime implicants and 5 minterms. further between m6 & m7 only 1 of them can be retein as both of them covered by same prime implicant.

PIM	m1	m2	m3	m4	m5	m6	m7	m8	m9
PCS ~~P~~	✓		✓						
Q		✓	✓					✓	✓
R	✓		✓	✓	✓				
S					✓	✓	✓	✓	
T	✓	✓							✓
UCQ ~~U~~		✓							

~~P~~ + Q + R + S + T + ~~U~~

	m4	m6	m9
*Q			✓
*R	✓		
*S		✓	
T			

∴ all min terms are canceled by above Q, R, S

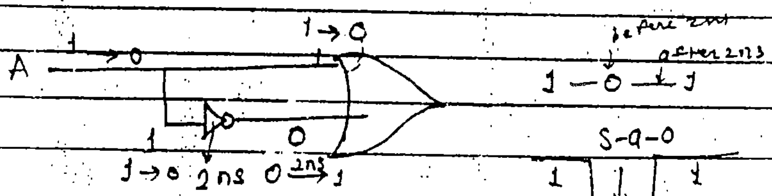
∴ contain 1 min term as Q + R + S

Hazards

Hazards

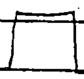
- Hazards - malfunction of digital circuits
- Temprory hazards are resulted due to uneven delay of input signals.
- Permenant Hazards results due to open circuit & short circuit of connecting leads.
- Test vector :- Subset of combinations that influence the path.
- ~~Family~~ Fault path :- If the digital circuit given wrong output for each element in the Test vector it is declared fault path.
- Path Sensitization Technique :- It's used to construct Test Vector for the chosen path.

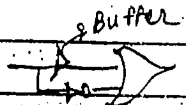
Procedure :- ① In activate all other paths except the tested one. (For AND/NAND are use '1' for activation)
② Apply opposite logic level at tested place.
③ All the input binary patterns satisfy above form test vector.



Struck-at-zero (S-a-0) for short time & recoverable.

Stuck-at-one (S-a-1) :





By introducing Buffer (appro. delay at appro. i/p) avoid the hazards.

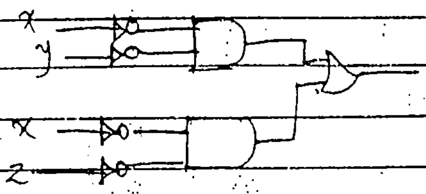
one part take i/p quickly & other after sometime then it having hazard.

~~which of~~ exp. having hazards & which not have?

$x'y' + xz'$, $x'y' + x'z' + xy + xz$, $xy' + yz'$

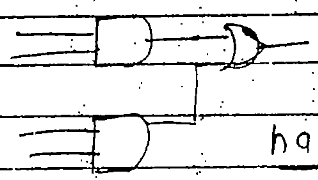
→ $x'y' + xz'$ → (S-a-0) Hazard

$x'y' + x'z'$



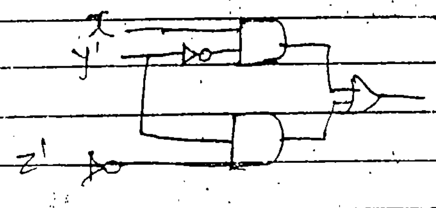
non-hazards.

$xy + xz$



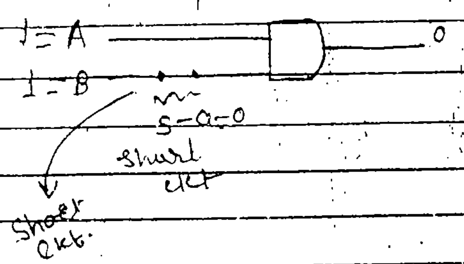
hazard free

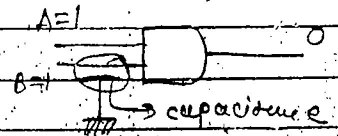
$xy' + yz'$



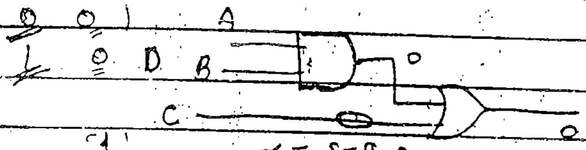
Hazard

~~Permanent~~ Hazards ⇒





Z-Kohavi book for Hazards



α = s-a-0

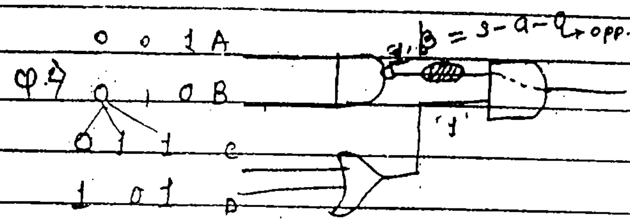
{011, 001, 101}
3, 1, 5

if we get '0' ans after given 1, 3, 5 i/p then it is s-a-0 hazards.

Step 1 → for AND/NAND are 1 for active

AB
A?
↓
'1'

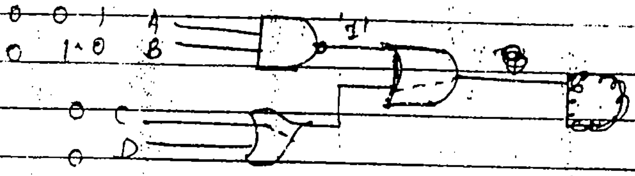
Q.4



β = s-a-1 → opp. value set w.r.t. hazard

{0001, 0010, 0011, 0101, 0110, 0111, 1001, 1010, 1011}
{1, 2, 3, 5, 6, 7, 9, 10, 11}

Q) for above construct s-a-1 test vector for input c

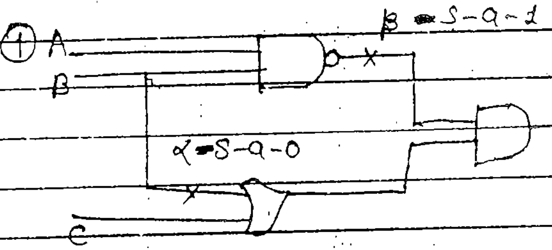


{0, 4, 8}

** Limitations of path sensitization technique :—

The path sensitization technique is not capable of providing the test vector if there is a conflicting requirement on any specific input.

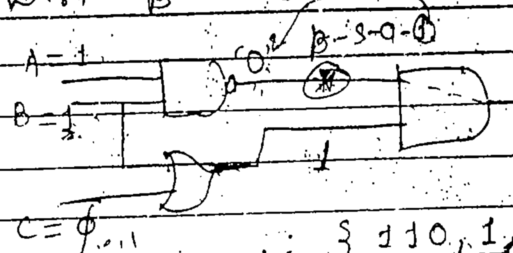
ex:→ ①



ⓐ Test vector is possible only for α-s-a-0
ⓑ Test vector is possible only for β-s-a-1
ⓒ Test vector is not possible for both
~~ⓓ~~ possible for both.

⇒ variefy w.r.t. β-s-a-1

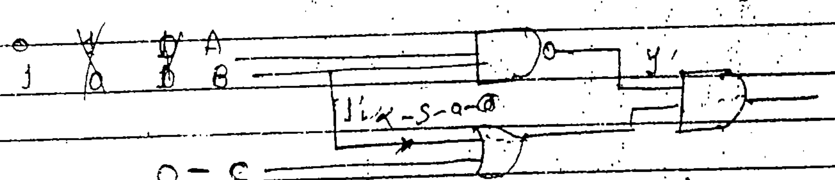
★



Test vector

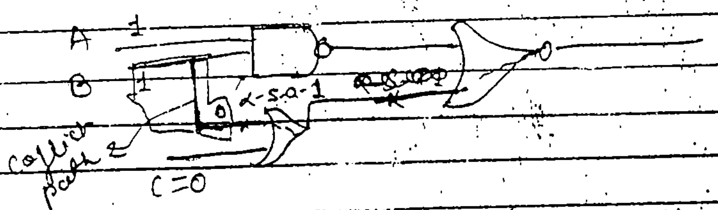
∴ {110, 111} = {6, 7}

variefy w.r.t. α-s-a-0



Test vector = {2}

②



∴ B requir both of 1 which are coflict req.

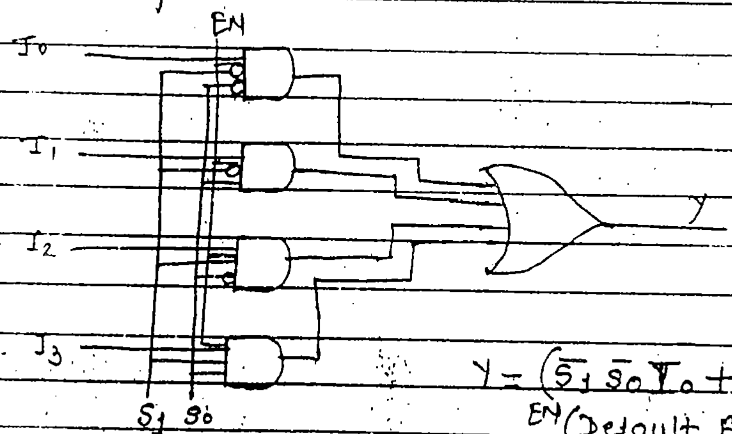
∴ Test vector is not possible with path sentization technique.

MSI Circuits (Medium Scale Integrated ckt)

Multiplexer	Data Selector	Many to one	8x1	8 i/p, 1
Demultiplexer	Data Distributor	one to many	1x16	1 i/p, 16
Decoder	Address selection	some to many	3x8	3 i/p, 8
Encoder	Interrupt Servicing (Decimal to Binary Conversion)	Many to some	4x2	4 i/p,

	gate Count
SSI	<10
MSI	>10 & <100
LSI	>100 & <1000
VLSI	>1000

4x1 Multiplexer

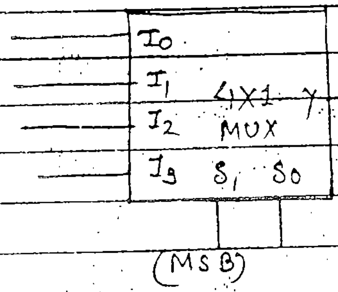


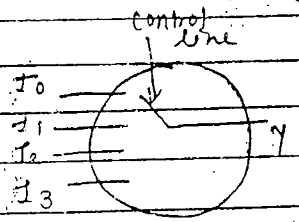
EN	S_1	S_0	Y
1	0	0	I_0
1	0	1	I
1	1	0	I
1	1	1	I
0	ϕ	ϕ	

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

EN (Default EN = 1)

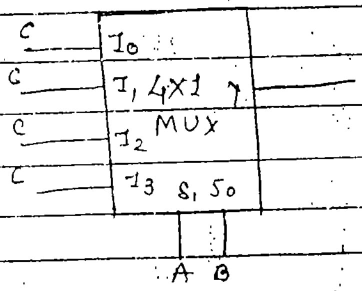
Mux is functionally complete operation.





(i) MUX represent any funcn in SOP form.

Q.1>



F (A,B,C) is free from?

(a) C (b) A,B

(c) A,C (d) A,B,C

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

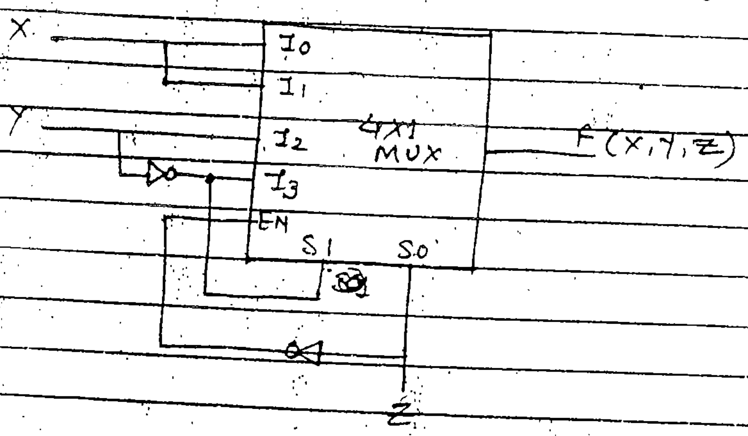
$= (\bar{A} + A)\cdot C$

$= C$

$\therefore$ free from $= A, B$

2005

(2) find the funcn represented by following multiplex



(a) xyz
(b) $\bar{x}+\bar{y}+\bar{z}$
(c) $xy\bar{z}$
(d) $xy+yz+$

$EN(\bar{S_0}\bar{S_1}I_0 + \bar{S_0}S_1I_1 + S_0\bar{S_1}I_2 + S_0S_1I_3)$

$S_1=\bar{Y}$, $S_0=Z$, $I_0=I_1=X$, $I_2=Y$, $I_3=\bar{Y}$

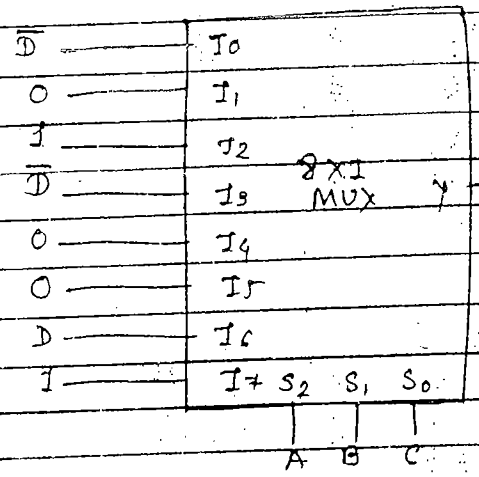
Z should be '0' only otherwise it will make EN & MUX can't be worked. i.e.

S_1	S_0	
ϕ	0	
0	0	I_0
1	0	I_1

or

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

Q What is the funcn represented by following MUX?



$Y(A,B,C) = ? = \Sigma(?)$

a) $\Sigma(1,4,5,7,12,13,14,15)$

b) $\Sigma(0,4,5,6,13,14,15)$

c) $\Sigma(0,4,5,13,14,15)$

d) $\Sigma(0,1,3,4,5,14,15)$

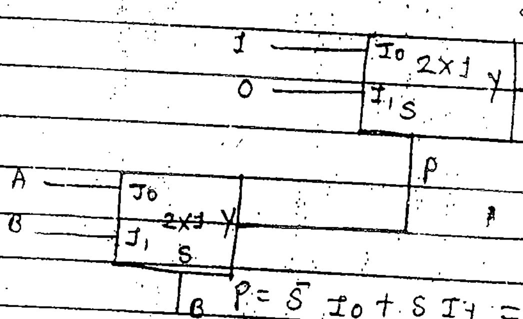
S_0	S_1	S_2		
0	0	0	I_0	$\bar{A}\bar{B}\bar{C}\bar{D} \rightarrow \{0\}$
0	0	1	I_1	$\bar{A}\bar{B}C \cdot 0$
0	1	0	I_2	$\bar{A}B\bar{C} \cdot 1 \rightarrow \{4,5\}$
0	1	1	I_3	$\bar{A}BC \cdot \bar{D} \rightarrow \{6\}$
1	0	0	I_4	$A\bar{B}\bar{C} \cdot 0$
1	0	1	I_5	$A\bar{B}C \cdot 0$
1	1	0	I_6	$AB\bar{C} \cdot D \rightarrow \{13\}$
1	1	1	I_7	$ABC \cdot 1 \rightarrow \{14,15\}$

A B C D
0 0 1 0 ϕ

$\bar{S_2}\bar{S_1}\bar{S_0}I_0 + \bar{S_2}\bar{S_1}S_0 I_1 + \bar{S_2}S_1\bar{S_0} \cdot I_2 + \bar{S_2} S_1 \cdot S_0 I_3 + S_2\bar{S_1}\bar{S_0} I_4 +$
$S_2\bar{S_1}S_0 I_5 + S_2 S_1 \bar{S_0} I_6 + S_2 S_1 S_0 I_7$

(4) Identify the funcn represented by following MUX combination.

$\bar{S} \cdot I_0 + S I_1 = \bar{P} \cdot 1 + P \cdot 0 = \bar{P}$



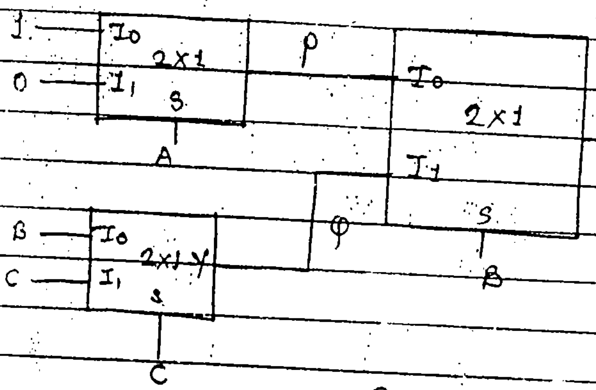
F(A, B, 0)

(a) $\bar{A} + \bar{B}$
(b) $\bar{A}\,\bar{B}$
(c) $A + B$
(d) $A \oplus B$

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

$\therefore \bar{P} = \bar{A} \cdot \bar{B}$

(5) What's the funcn represented by following M[illegible] block?



$\bar{S}\, I_0 + S I_1$

F (A, B, C) = ?

$P = \bar{A}$

$Q = B + C$

$\therefore \bar{B} P + B \cdot Q$

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

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

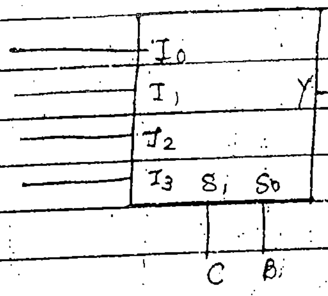
$= \bar{A} + B$

$= \Sigma(0, 1, 2, 3, 6, 7)$

A	B
0	ϕ

A	B	C
ϕ	1	ϕ

(4) Consider a 3 variable funcn $f(A,B,C) = \sum(0,2,3,4,7)$ is to be realised by the following 4x1 MUX. The select lines S_1 is given as C & S_0 given as B. What could be connections for inputs?



$F(A,B,C)$

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

$= \bar{C}\bar{B}I_0 + \boxed{\bar{C}B}I_1 +$
$C\bar{B}I_2 + CBI_3$

$F(A,B,C) = \sum(0,2,3,4,7)$

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

$= \boxed{\bar{C}\bar{B}}\bar{A} + \boxed{\bar{C}B}\bar{A} + CB\bar{A} + \bar{C}\bar{B}A + CBA$

$\therefore I_0 = \bar{A} + A = 1$

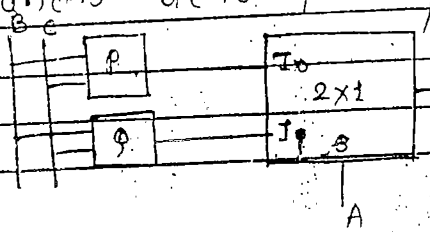
$I_1 = \bar{A}$

$I_2 = 0$

$I_3 = 1 = A + \bar{A}$

2017

(5) A majority funcn is the one in which number of 1 inputs are more than number of 0 inputs. If the majority funcn of 3 variables is represented by following realisation what will be the funcns denoted by P, Q?



(a) AND, OR
(b) OR, AND
(c) XOR, XNOR
(d) XNOR, XOR

is ⇒

1 1 1 → 7
~~1 1 0 → 6~~ i.e. $\Sigma(3,5,6,7)$
1 0 1 → 5
0 1 1 → 3

$\bar{S} I_0 + S I_1$ $\quad \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$

$3 \to \bar{A}BC$
$5 \to A\bar{B}C$
$6 \to AB\bar{C}$
$7 \to ABC$

$\bar{A}\boxed{P} + A\varphi \Rightarrow$

$\therefore P = BC \quad \varphi = \bar{B}C + B\bar{C} + BC = B + C$

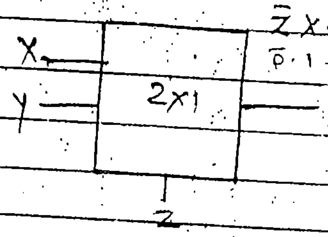
$\therefore$ P has to be AND operation

$\therefore$ AND, OR operatn

(6) Consider a 2x1 MUX which has inputs X, Y and the select line is Z. The MUX select X when Z=0 & select Y if Z=1. If it is required to represent a two variable funcn $F(P,\varphi) = \bar{P} + \varphi$. What will be X, Y, Z?

(a) P φ P (b) P 1 φ (c) P 1 P (d) ~~φ φ P~~ ~~φ P~~ φ 1 φ

⇒



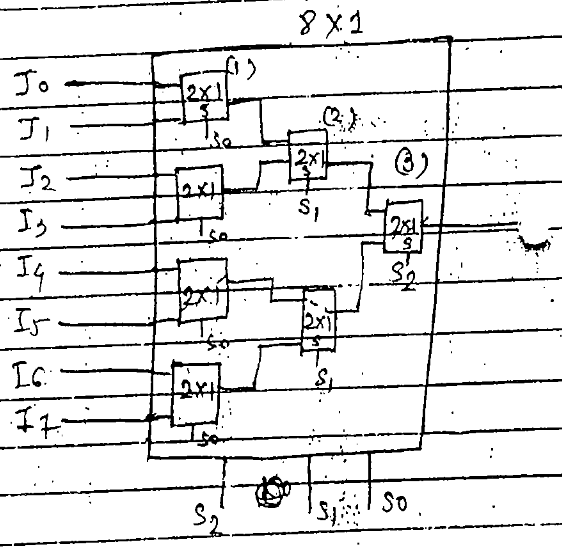
$\bar{Z}X + ZY$
$\bar{P} \cdot 1 + P \cdot q = \bar{P} + Pq$

	X	Y	Z	
(a)	P	φ	P	$= P\varphi$
(b)	P	1	φ	$= \bar{\varphi}P + \varphi = P +$
(c)	P	1	P	$= P$
(d)	1	φ	P	$= \bar{P} + P\varphi = \bar{P}$

Expansion of Multiplexers.

The higher capacity MUX is constructed with combination of lower capacity MUX arranged in multiple levels.

Target	$M \times 1$	8×1
Basic	$N \times 1 \quad N < M$	2×1
Number of levels K	$K = \log_N M$	$\log_2 8 = 3$
Number of MUX in i^{th} level	M / N^i	$\overset{1}{8/2} \quad \overset{2}{8/(2)^2} \quad \overset{3}{8/(2)^3}$
Total multiplexers	$\sum_{i=1}^{K} (M/N^i)$	$4 + 2 + 1 = 7$
Maximum capacity with K-levels	$N^K \times 1$	$2^3 \times 1 = 8 \times 1$



For ex (1 0 1) = 5

85

(1) How many 4x1 mux are required to construct 128 MUX?

⇒ Target 128x1
Basic 4x1

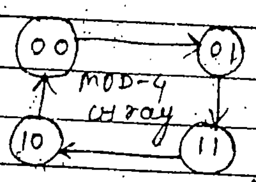
$\log_4 128 = \frac{7}{2} \simeq 4$ = No. of levels = k $\left\{ \log_{a^n} a^m = m/n \right.$

	1	2	3	4
Num of Mux in ith level	$\frac{128}{4}$	$\frac{128}{(4)^2}$	$\frac{128}{(4)^3}$	$\frac{128}{(4)^4}$
Total Mux	32 +	8 +	2 +	1 = 43

∴ 42, 4x1 a
1, 2x1 MU

Max capacity with k-level $(4)^4 \times 1 = 256 \times 1$

~~*~~ MOD-4 Gray counter *

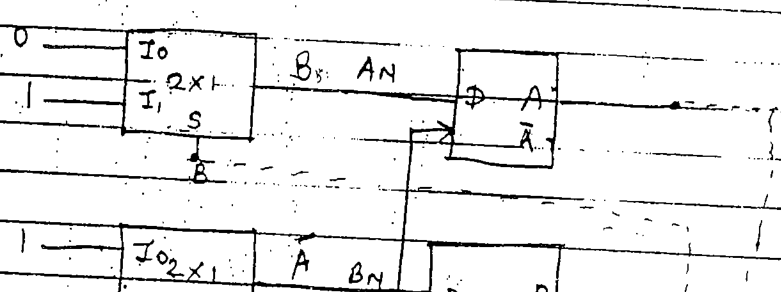


what is next state of state

A	B	AN	BN
0	0	0	1
0	1	1	1
1	0	0	0
1	1	1	0

AN: $\bar{A}B$, AB

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



every next ~~present~~ state become ~~next~~ present state after sometime

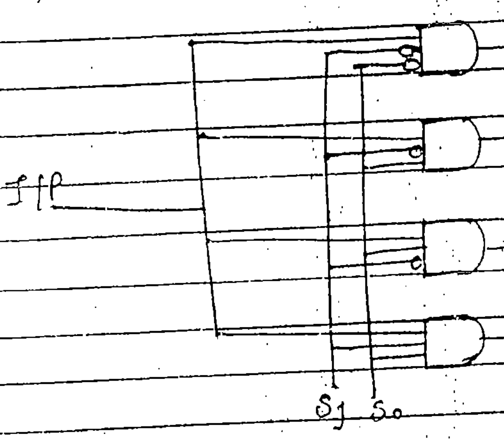
A_N — Delay — A B_N — Delay — B

De Multiplexer (one to many)

It is useful in the presence of multiplexed input. with the slight modification of the multiplexer h/w is converted into demultiplexer. The Dmux provides decoder after modifying its basic block. Decoders are useful for address selection & code conversion.

ex: → LS 74 138 (IC)

- LS → Lowspeed, Low power
- 74 → silicon
- 138 → 3x8 Decoder, 1x8 De-Mux



$O_0 = \bar{S_1}\bar{S_0}I$

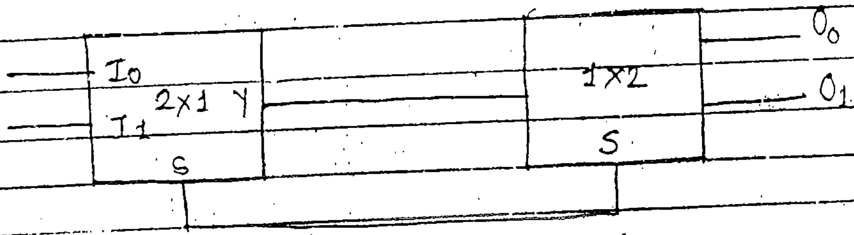
$O_1 = \bar{S_1}S_0I$

$O_2 = S_1\bar{S_0}I$

$O_3 = S_1S_0I$

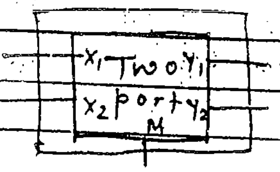
Mux → join inputs & detach last OR gate → De-Mux

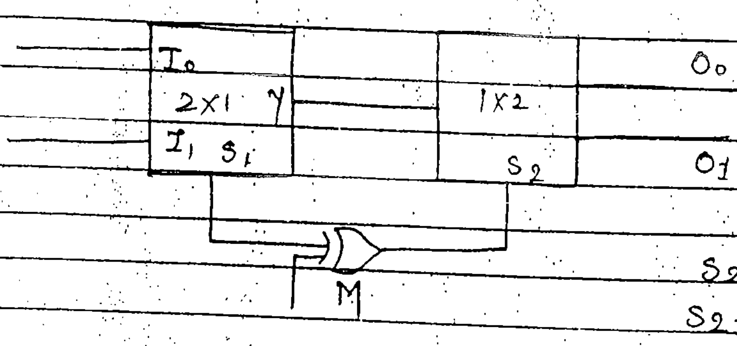
S_1	S_0	O_0	O_1	O_2	O_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I



$I_0 \rightarrow O_0$

$I_1 \rightarrow O_1$



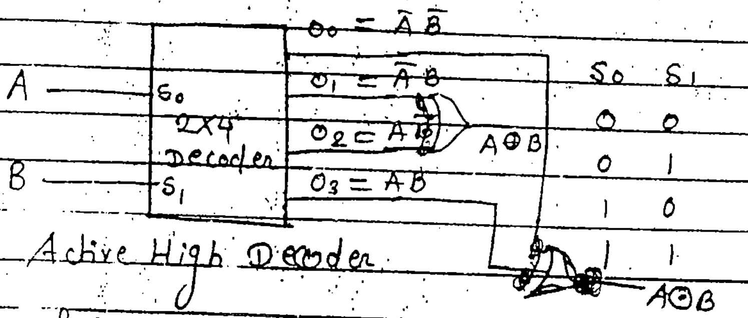


$S_2 = M \oplus S_1$

$S_2 = S_1$ {M = 0 ; straight conn

$S_2 = \overline{S_1}$ {M = 1 ; cross conn

The Decoder ⇒ If i/p of Demux is inactivated & select lines are converted into i/p then it funcn as decoder.



S_0	S_1	O_0	O_1	O_2	O_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

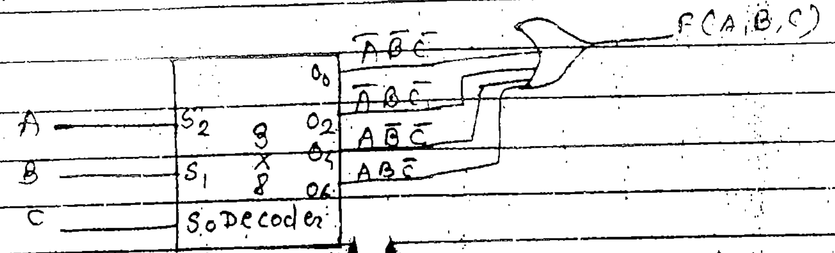
Active High Decoder

when O_1 & O_2 are XORd $O_1 + O_2 = A \oplus B$

when O_0 & O_3 are XNORd $O_0 + O_3 = A \odot B$

- one decoder can relalised multiple funcns.
- one Mux can relalised only one funcn.

① w.r.to following realisation identify the correct statement.



F is free from

ⓐ one variable

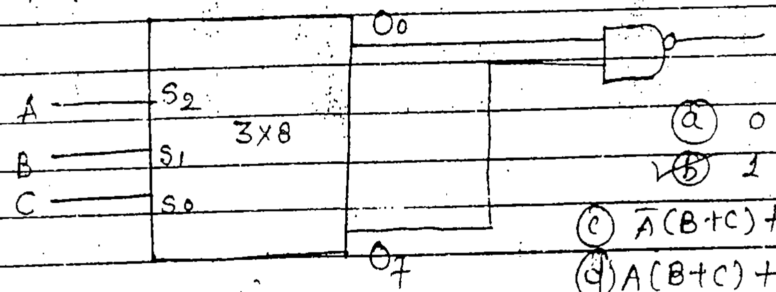
ⓑ 2 variables

ⓒ 3 variables

ⓓ not free from any variable

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

② What will be the funcn given by the following decoder setup?



ⓐ 0

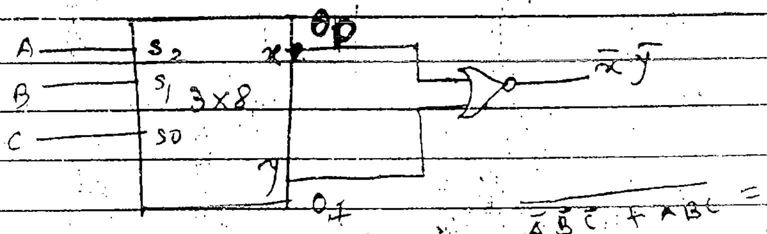
ⓑ 1

ⓒ $\bar{A}(B+C) + A(\bar{B}+\bar{C})$

ⓓ $A(B+C) + \bar{A}(\bar{B}+\bar{C})$

$\Rightarrow$ $O_0 = 0\ 0\ 0$
$O_7 = 1\ 1\ 1$ $\Big\}$ $O_0 . O_7 = 1$

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



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

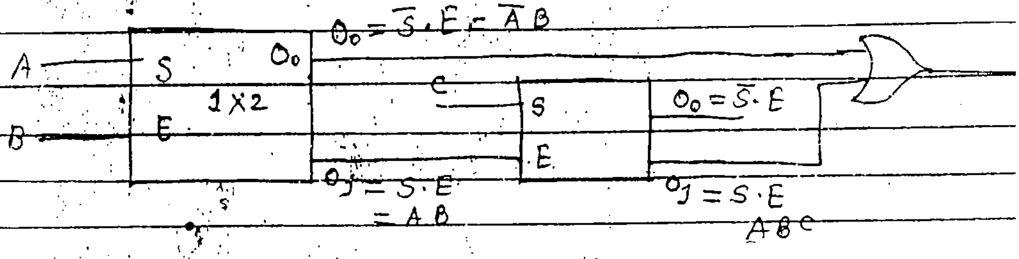
⇒ $= \overline{\overline{A}\,\overline{B}\,\overline{C}} \cdot \overline{ABC}$

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

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

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

(3) what is the funcn given by following decoder



$\overline{A}B + ABC$

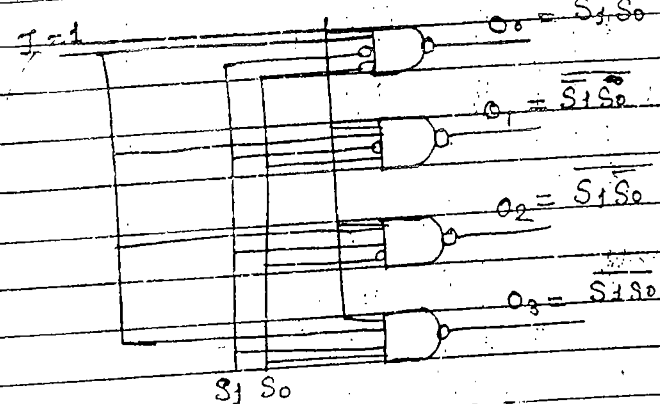
$= B(\overline{A}+AC) = B(\overline{A}+C)$

$= \overline{A}B + BC$

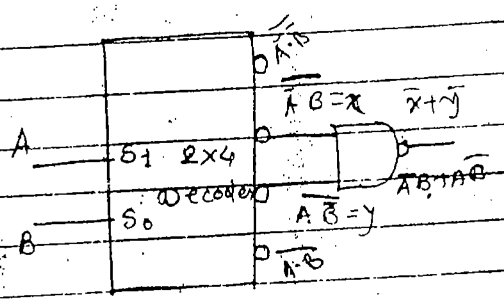
$\boxed{2\times4}$ $2+4=6$

$2+2\times4 = 2+8=10$

* Active Low Decoder *



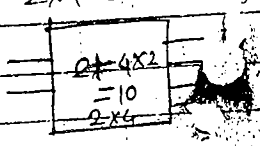
S_1	S_0	O_0	O_1	O_2	O_3
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0



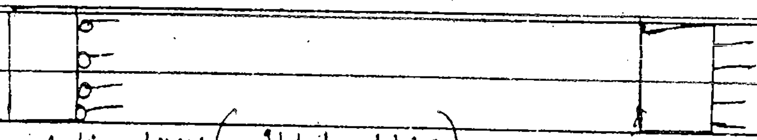
Active low decoders are prefered to Active high decoders as they require less number of universal gates.

2x4 $m\times n \Rightarrow n = 2^m$	
Decoder	NAND gates
Active High	m+2n
Active Low	m+n

ex:-> 2x4 active High Decoder

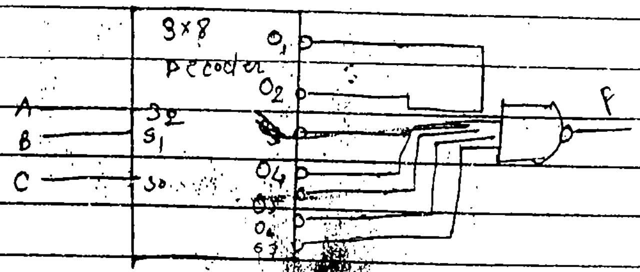


① As req. less no. gates active low prefer

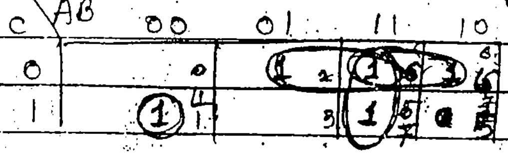


Active low (with bubble) Active High

Q1) func represented by following decoder.



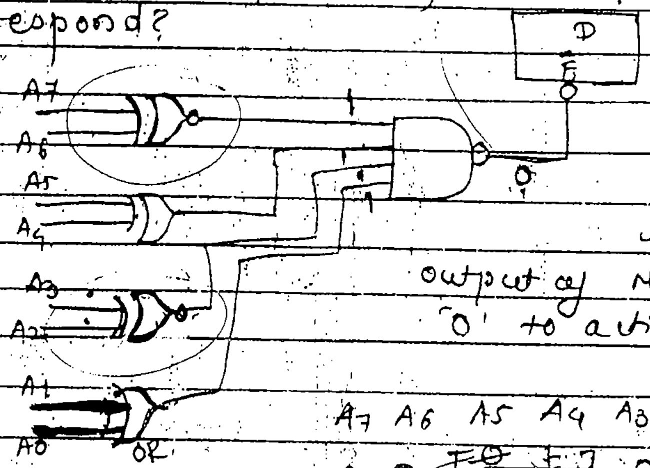
NAND
AND
$\Sigma(A,B,C)$
$= \Sigma(1,2,4$
AND



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

(2) consider an 8 bit address bus which was aim to activate device D. How many addresses does device D will respond?

⇒



$A_7 \equiv A_6$

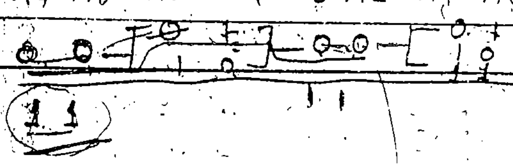
$A_5 \neq A_4$

$A_3 = A_2 = 0$

$A_1 = A_0 \neq 0$
(simultaneously can be zero)

(a) 3 (b) 8
(c) 12 (d) 16

output of NAND gate must '0' to active device.

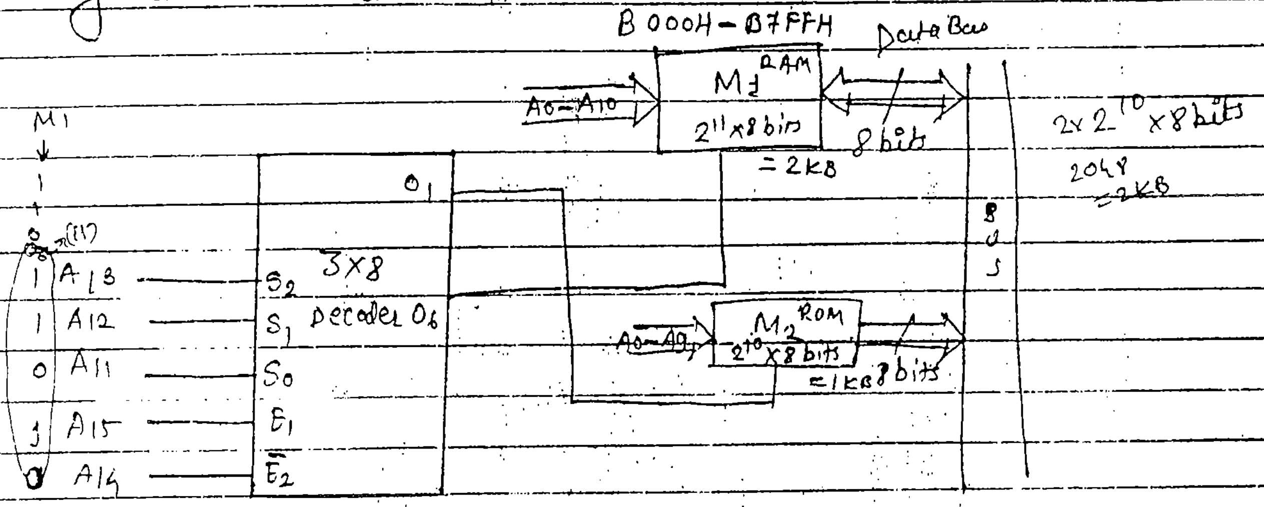


$2^{10} = 1KB$

0001 0001	0010 0001
0001 0010	0010 0010
0001 0011	0010 0011

11 H	21 H
12 H	22 H
13 H	23 H
D1 H	E1 H
D2 H	E2 H
D3 H	E3 H

Consider two memory devices M_1 & M_2 which are activated by the following decoder setup? A 16-bit Address bus is used for this purpose. What will be the type & capacity of device M_1 & M_2?



(i) Type & capacity of device

(ii) Address range of each device

(iii) First three location address of M_1 & last three locatn address of M_2

Sol^n ⇒ (i) To activate high decoder M_1 : 06

6 represent 1 1 0

(M_1) B000H - B7FFH

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	M_1
					1	1	1	1	1	1	1	1	1	1	1	
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	
					1	1	1	1	1	1	1	1	1	1	1	M_2

8800H to 8BFFH } M_2
8C00H to 8FFFH }

(iii) First three addresses of M_1

B000H, B001H, B002H

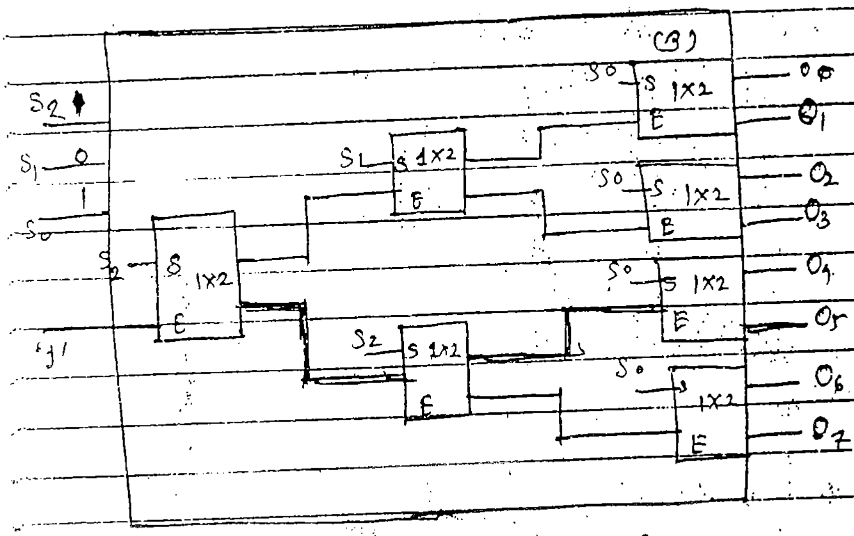
Last three addresses of M_2

8BFDH, 8BFEH, 8BFFH
8FFDH, 8FFEH, 8FFFH

Expansion of the Decoders

The higher capacity decoder is constructed using the smaller capacity decoders in multilevels.

Target	$M \times N$, $N = 2^M$	3×8
Basic	$p \times \varphi$, $\varphi = 2^p$ & $\varphi < N$	1×2
Number of levels	$k = \frac{M}{p} = \log_{\varphi}^{N}$	$k = \log_2^8 = 3$
Number of Decoders in ith level	$\frac{N}{\varphi^{k-i+1}} = x$	1: $\frac{8}{2^{3-1+1}} = 1$, 2: $\frac{8}{2^{3-2+1}} = 2$, 3: $\frac{8}{2} = 4$
Total decoders	$\sum_{i=1}^{k} x$	$1 + 2 + 4 = 7$
Maximum capability with k-levels	$(p * k) \times \varphi^k$	$(1 * 3) \times 2^3 = 3 \times 8$



$101 = O_5$

① How many 3x8 decoder are required ? to construct 8x256 decoder.

Target 8×256

Basic 3×8

Level $k = 8/3 \approx 3$

1 2 3

$\frac{4}{8}$ $\frac{32}{8}$ $\frac{256}{8}$ ~~= 32~~

1 4 32 = 37

$(5 * 3) * 8^3 = 9 \times 512$

36 → 3x8 decoder
1 → 2x4 decoder

② Consider a graphic system which was suppose to use the 3 primary colors with each color is having 256 levels depth. The screen that is used is having the resolution 512x 1024. What will the memory required for frame buffer?
(frame buffer = maintain info for every pixel)

→

	R	G	B
pixel	8 bit	8 bit	8 bit

$\underbrace{\qquad\qquad}_{24\text{ bit}}$

512x1024

$\frac{512 \times 1024 \times \cancel{24}^{3}}{\cancel{8}\text{ bit / Byte}}$

$2^9 \cdot 2^{10} \cdot 3\ B = 512\ KB \times 3$

$= 1.5\ MB$

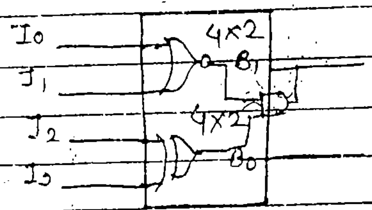
* Encoder *

The encoder used to convert decimal input to binary if they are provided with priorities they can support the interrupt servicing. In the design of encoder its needed to know the rule of encoding.

ex:→ Highest suffix input is encoded with highest Binary value.

The non-priority encoders doesnot support simultaneous activation of inputs. The priority encoders reduces the h/w & support simultaneous activation. The rule of assigning priorities can be taken as :- highest suffix i/p is given with highest priority.

The priority encoder suffer the starvation due to the static priorities.



Non-priority Encoder

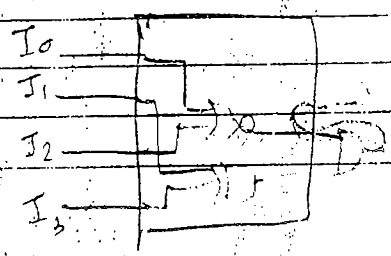
I_0	I_1	I_2	I_3	B_1	B_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

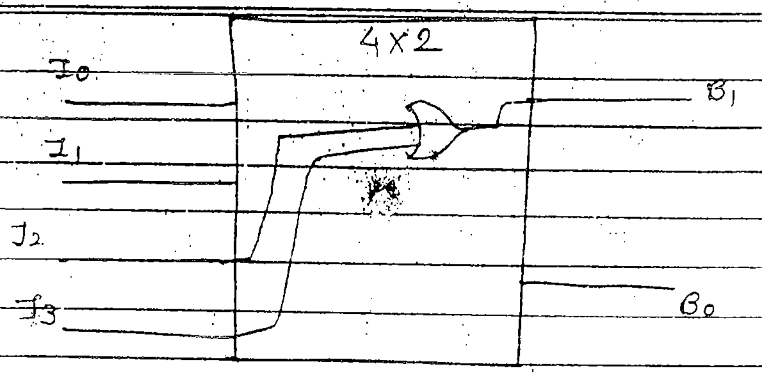
$B_1 = I_0' I_1' I_2 I_3' + I_0' I_1' I_2' I_3$

$B_0 = (I_0 + I_2)' \cdot (I_1 \oplus I_3)$

$B_1 = I_0' I_1' (I_2 \oplus I_3)$

$\boxed{B_1 = (I_0 + I_1)' \odot (I_2 \oplus I_3)}$





priority Encoder

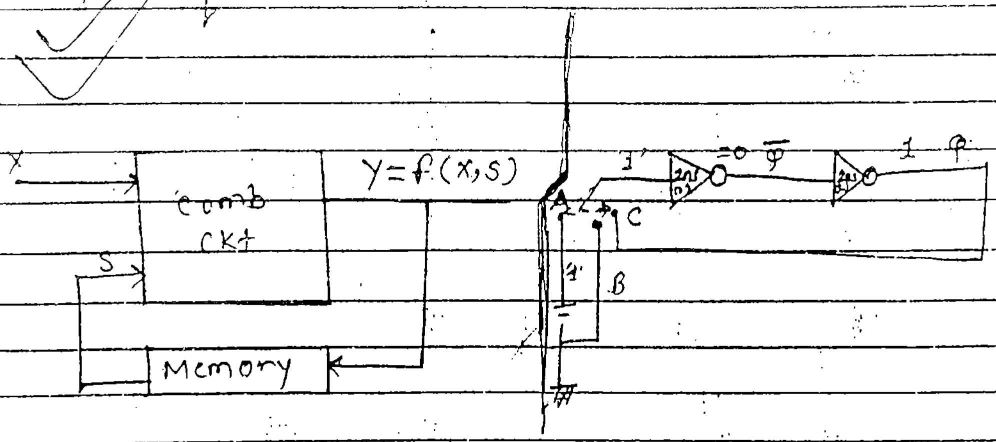
I_0	I_1	I_2	I_3	B_1	B_0
1	0	0	0	0	0
ϕ	1	0	0	0	1
ϕ	ϕ	1	0	1	0
ϕ	ϕ	ϕ	1	1	1

$B_1 = I_2 \cdot I_3' + I_3$ $\qquad$ $B_0 = I_1 I_2' I_3 + I_3$

$B_2 = (I_2 + I_3)$ $\qquad$ $B_0 = I_1 I_2' + I_3$

Approch ~~to~~ static priority & give starrato to lower priorit

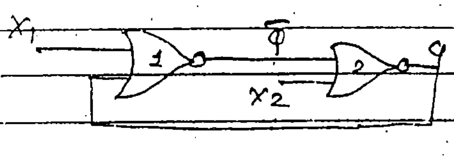
* Sequential Circuits

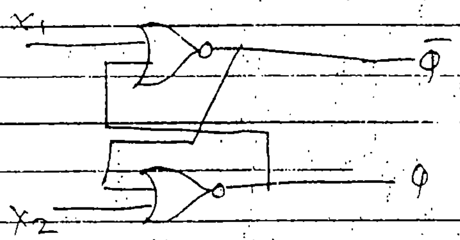


S --> state ---> previous output

- valid ~~output~~ operats ⇒ complimentory o/p

- Some control input used for entery of logic '1' & '0'.





flip-Flop

FlipFlop is basic memory element & its capable of 1-bit storage. It contains two stable states (stable states maintain complimentory relation for valid operation). The flip-flop also called as bistable multivibrator.

- They are classified based on construction, operating mode, nature of clock synchronization & nature of the funcn.

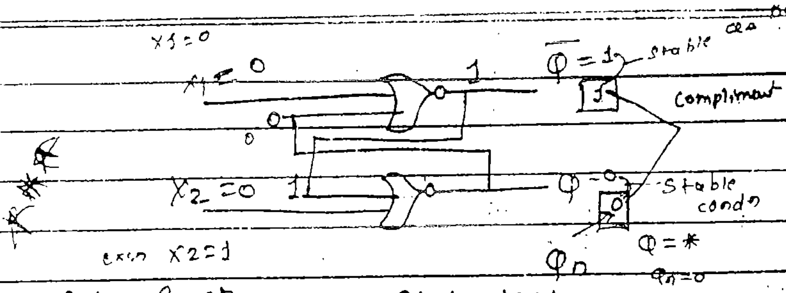
- construction
  - ~~NOR~~
  - NAND
- Operating mode
  - ~~Transperent (follow i/p)~~
  - Latched Mode ($Q_n = Q$)
- Nature of synchronizatn
  - ~~Level Triggered Flip-Flop } Latch~~
  - Edge Triggered Flip-Flop } synchronous flipflop
- Nature of funcn
  - ~~S-R (set-Reset) F.F.~~
  - J-K F.F.
  - D (Delay) F.F.
  - T (Toggle) F.F.

features are :- (1) characteristic expression

$$Q_n = f(X, Q)$$

(2) Excitation Tables

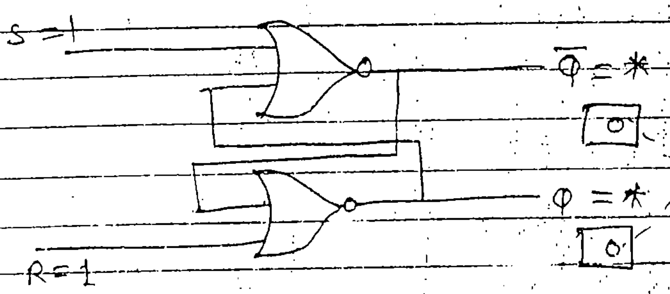
□ → stable state & maintain compliment



A	B	NOR
ϕ	ϕ	0
ϕ	1	0
0	0	1

State table

Set X_1	Reset X_2	Q	Q_n	
0	0	0 A	0	$Q_n = Q$
0	0	1 B	1	
0	1	0 A	0	$Q_n = 0$ whatever may be Q, Q_n always 0
0	1	1 B	0	
1	0	0 A	1	$Q_n = 1$ whatever may be Q, Q_n always 1
1	0	1 B	1	
1	1	0 A	ϕ	invalid
1	1	1 B	ϕ	



not compliment ∴ invalid

$$\therefore Q_n = f(S, R, Q) = \sum(1,4,5) + \sum_{\phi}(6,7) \quad ***$$

$Q \backslash SR$	00	01	11	10
0	^{0}	^{2}	$\phi\ {}^{6}$	$1\ {}^{4}$
1	$1\ {}^{1}$	^{3}	$\phi\ {}^{7}$	$1\ {}^{5}$

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

0	0	0	0
0	1		

S R
0 ɸ

Excitation table of S-R flip flop

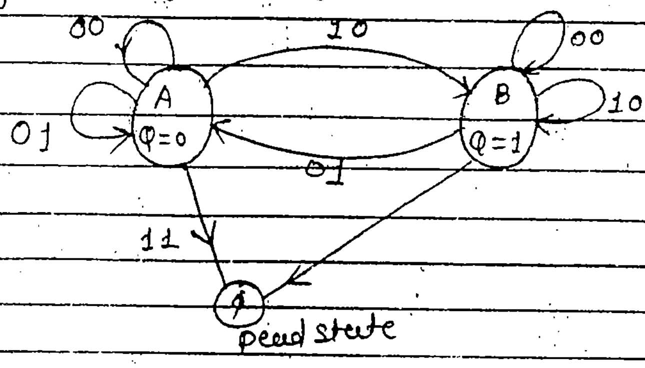
" for getting desired change at output what values are needed at input "

$Q \longrightarrow Q_n$		S	R
0	0	0	ɸ
0	1	1	0
1	0	0	1
1	1	ɸ	0

obtain above table from state table

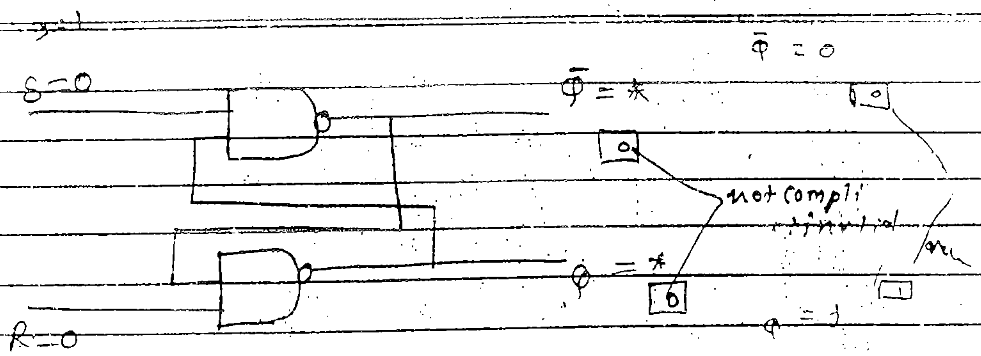
State diagram of S-R flip flip ⇒

(from state table)



S-R NAND flip flop ⇒ The SR-NAND behaviou is same as ~~NOR~~ $S \neq R$. It will perform oppositely when $S = R$.

A	B	NAND
0	0	1
0	1	1
1	0	1
1	1	0



set X_1	Reset X_2	Q	Q_n	
0	0	0	ϕ	} invalid
0	0	1	ϕ	
0	1	0	0	} $Q_n = 0$
0	1	1	0	
1	0	0	1	} $Q_n = 1$
1	0	1	1	
1	1	0	0	$Q_n = 0$ } $Q_n = Q$
1	1	1	1	$Q_n = 1$

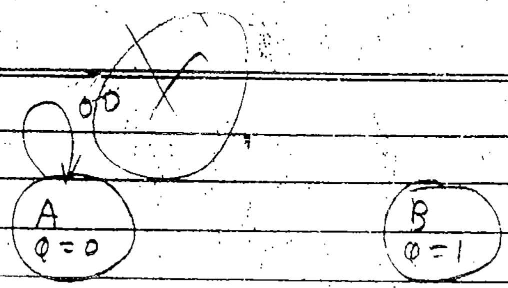
$$Q_n = (S, R, Q) = \sum (4, 5, 7) + \sum_{\phi} (0, 1)$$

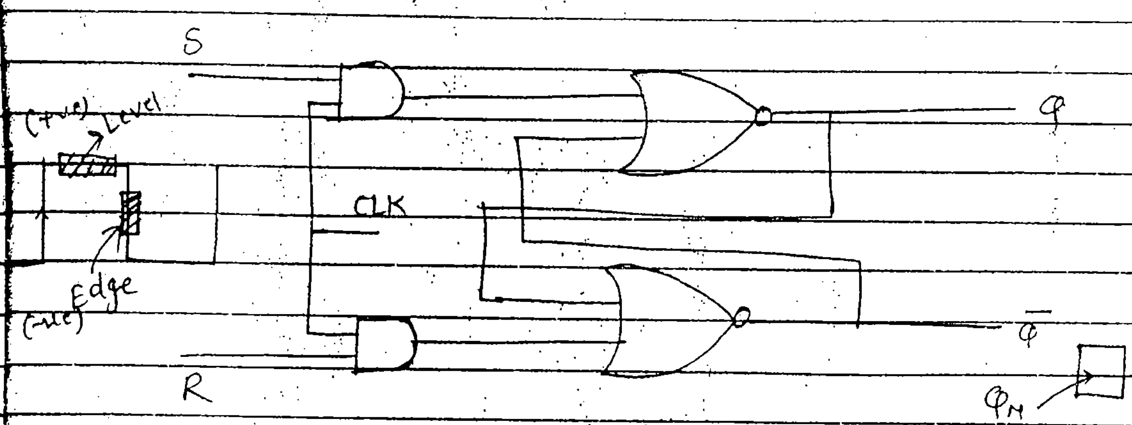
$$\boxed{Q_n = \bar{R} + SQ}$$

Excitation Table ⇒

$Q \rightarrow$	Q_n	S	R
0	0	ϕ	1
0	1	1	0
1	0	0	1
1	1	1	ϕ

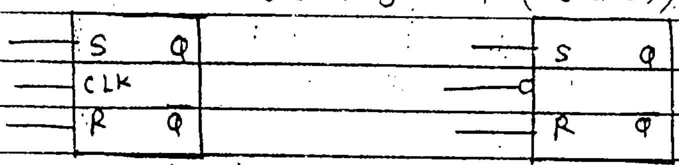
State Diagram ⇒





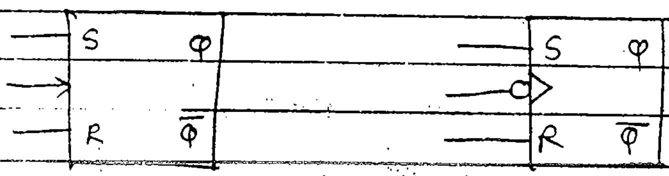
If clk is inactive in clocked FF the flip flop retain last state ~~input~~ independent of inputs.

Level Trigerred (Latch)



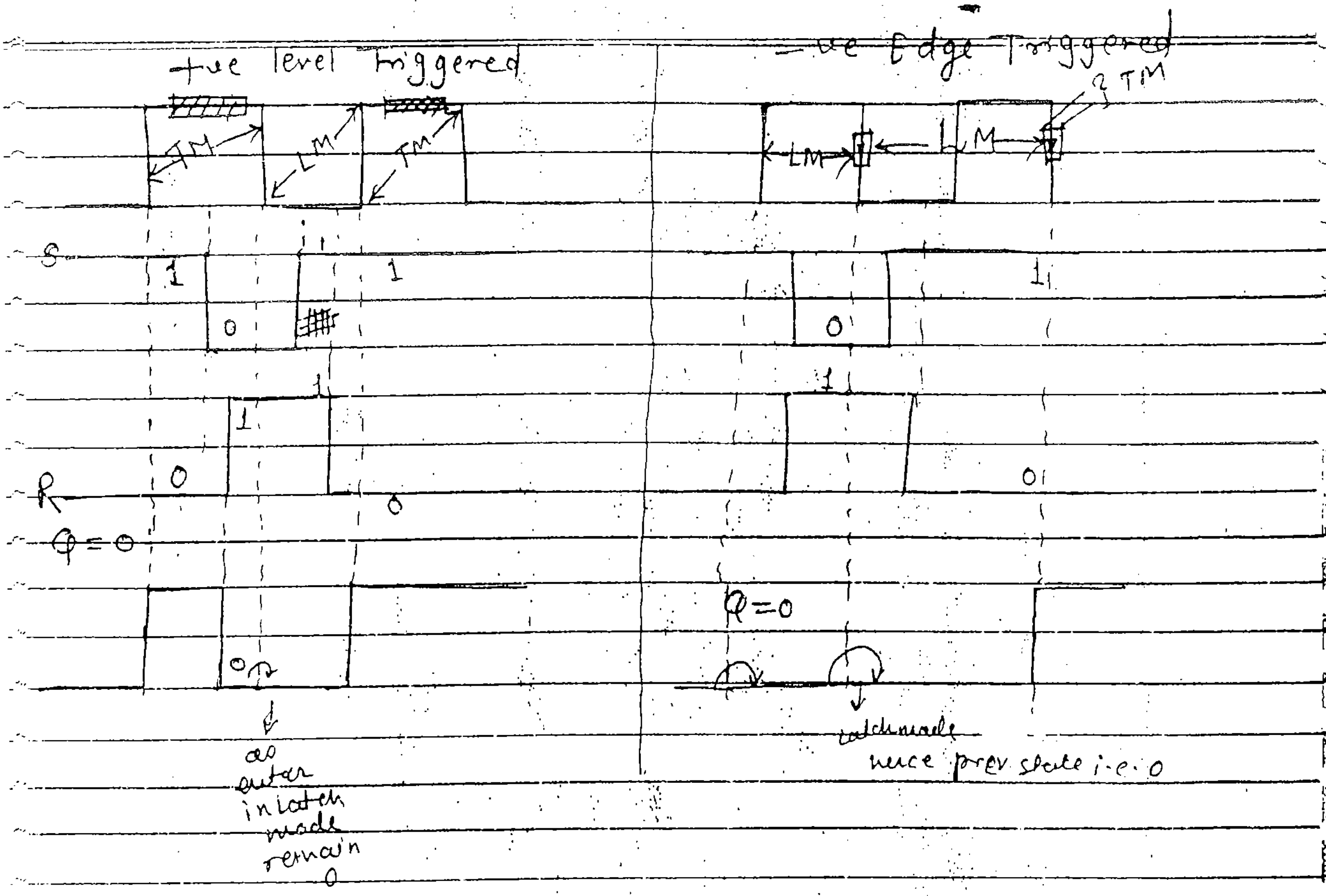
+ve level -ve level (bubble mean

Edge Triggered



→ invalid response in LM

Transperent Mode – TM
Latched Mode – LM



- The edge trigging is preferred ~~to~~ (not change) level triggering because it is more insensitive to the noise. The reason is shorter duration of transperent mode.
- In noisy environment edge triggred f.f. is preferred.
- The o/p is more smooth in edge triggring as compaired to level triggering.

* J-K flip-flop *

This f.f. is similar to SR-flipflop (J=S, K... It is defined when both inputs are 1. If J=K=... the f.f. compliments its previous output.

The J-K flip flop can be used as two indivi... one input f.f.

1. D-flipflop ($J \neq K$)
2. T- flipflop ($J = K$)

State Table of J-K Flip Flop

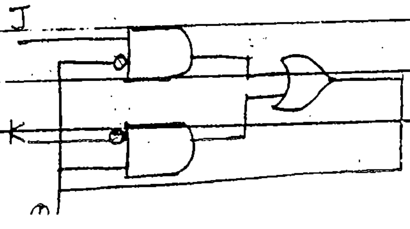
J	K	Q	Q_n	
0	0	0	0	$Q_n = Q$
0	0	1	1	
0	1	0	0	$Q_n = 0$
0	1	1	0	
1	0	0	1	$Q_n = 1$
1	0	1	1	
1	1	0	1	$Q_n = \overline{Q}$
1	1	1	0	

$$Q_n = f(J, K, Q) = \sum(1, 4, 5, 6)$$

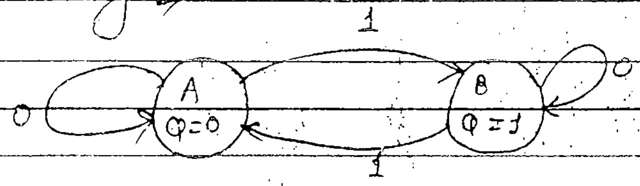
Q \ JK	00	01	11	10
0			1	1
1	1			1

Q \ JK	00	01	11	10
0			1	1
1	1			1

$$Q_n = \overline{J}\,\overline{Q} + \overline{K}Q$$

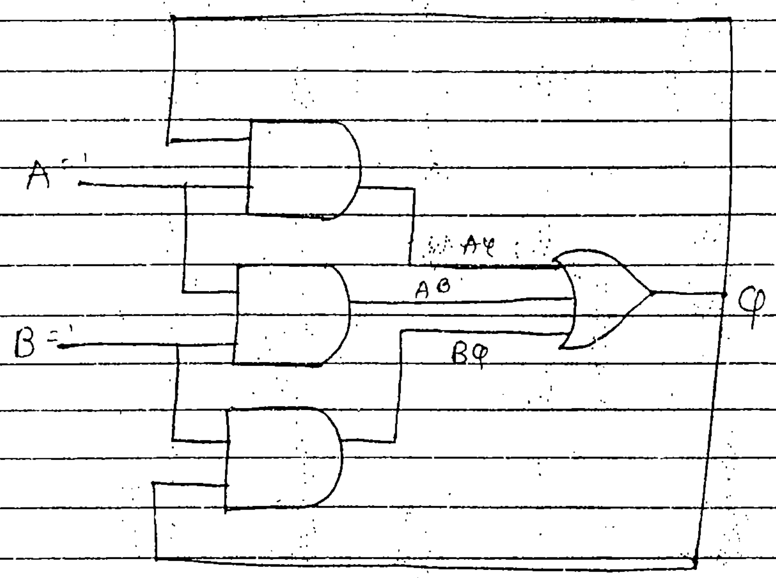


~~State diag~~ ⇒



Excitation statement ⇒ whenever there is a change at output T must be 1.

(1) Consider the following sequential ckt obtain the state table, char exp, state diag & excitation Table



$Q_n = AB + Q(A+B)$

$A = B = 1$

$Q_n = 1$

$A = 0, B = 1$ & $A = 1, B = 0$

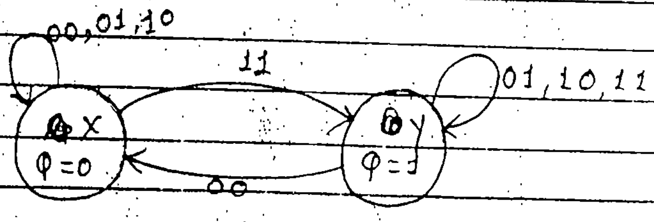
$Q_n = Q$

else

$A = B = 0$

$Q_n = 0$

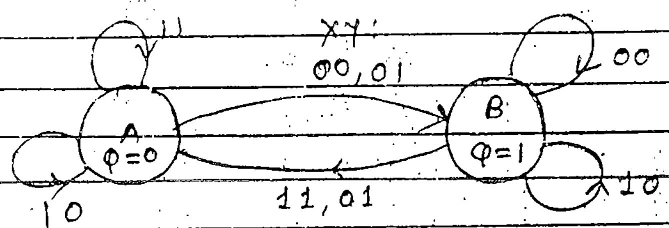
A	B	Q	Q_n
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



excitation Table ⇒

$Q \longrightarrow Q_n$		A	B
0	0	ϕ	ϕ
0	1	1	1
1	0	0	0
1	1	ϕ	ϕ

Q. Convert the following state machine using JK f.f. Obtain i/p J & K.

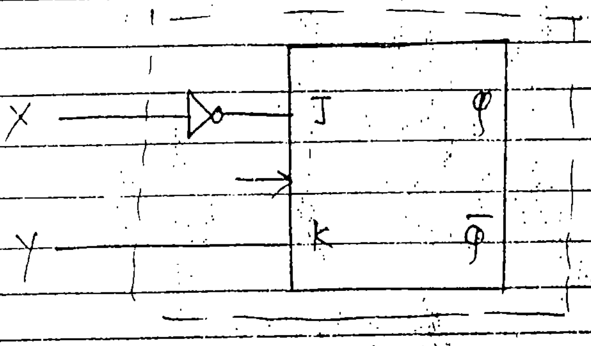


partial table

X	Y	Q_n	J	K
0	0	1	1	0
0	1	$\bar{Q}$	1	1
1	0	Q	0	0
1	1	0	0	1

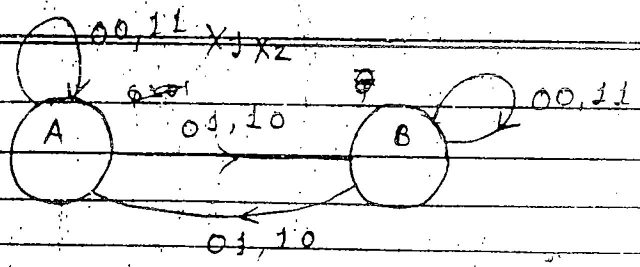
$J = \bar{X}Y + \bar{X}\bar{Y} = \bar{X}$

$K = \bar{X}Y + XY = Y$



Q.) Consider the following state diag which has two states A & B with two i/ps X_1 & X_2. It is to be represented by J-K f.f. What will be the expr for J & K i/p's.

9



(a) $\overline{x_1} x_2 + x_1 \overline{x_2}$, $\overline{x_1}\,\overline{x_2} + x_1 x_2$

(b) $\overline{x_1}\,\overline{x_2} + x_1 x_2$, $\overline{x_1} x_2 + x_1 \overline{x_2}$

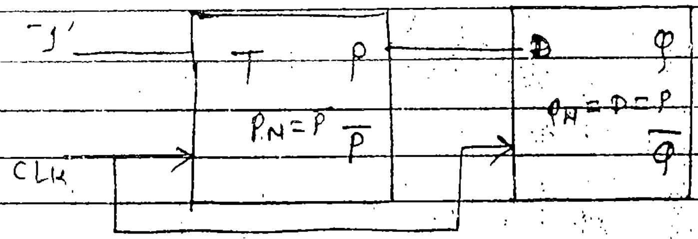
(c) $\overline{x_1}\,\overline{x_2} + x_1 x_2$, $\overline{x_1}\,\overline{x_2} + x_1 x_2$

(d) $\overline{x_1} x_2 + x_1 \overline{x_2}$, $x_1 \overline{x_2} + \overline{x_1} x_2$ ✓

X_1	X_2	φ_n	J	K
0	0	φ	0	0
0	1	$\overline{\varphi}$	1	1
1	0	$\overline{\varphi}$	1	1
1	1	φ	0	0

as
it is
the

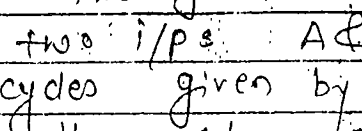
Q.> Consider the following sequential ckt which has initial value $P = 0$ & $Q = 1$. What will be the value of P & Q after 3 clock cycle?



$P = 0$
$Q = 1$

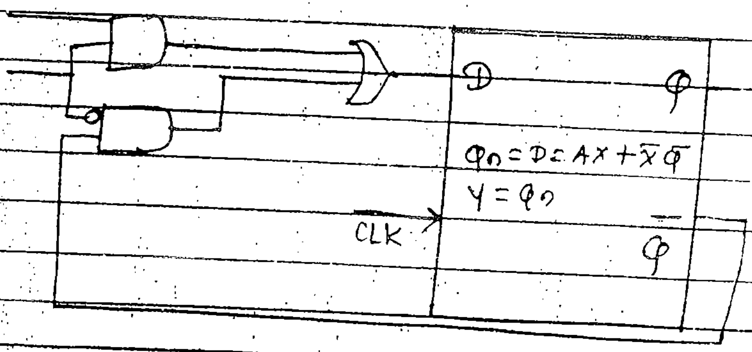
	present state		next	
	P	Q	$P_N = \bar{P}$	$Q_N = P$
	0	0	1	0
start →	0	1	1	0
	1	0	0	1
	1	1	0	1

(a) 00
(b) 01
(c) 10
(d) 11



after odd clk goto 10

Q.> Consider the following sequential machine which has two i/ps A & X. If their values for six clk cycles given by following table. What will be the o/p produced at Y?

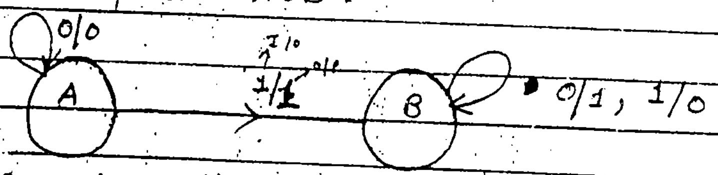


CLK	1	2	3	4	5	6
X	1	0	0	1	1	0
A	A_1	A_2	A_3	A_4	A_5	A_6
Y=?	A_1	A_1'	$(A_1')'$	A_4	A_5	A_5' → from cond^n s

Y=?

$$Q_n = AX + \bar{x}\bar{Q}$$
$$= \bar{Q} \text{ if } x = 0$$
$$= A \text{ if } x = 1$$

Q.) What is the func^n of following state machine if it has one i/p & one o/p? The first bit received is LSB.



(a) 1's complementor
✓(b) 2's complementor
(c) Bin to Gray Converter
(d) Increment i/p

Q.

Present State	Next state, Z x=1	x=0
A	B, 0	C, 1
B	C, 0	C, 1
C	D, 0	A, 0
D	C, 0	C, 1

First i/p
B : 1 0
Z(i/p) : 1 1

Bin	Gray
0 0	0 0
0 1	0 1
1 1	1 1
1 0	1 0

B B A A

2' compliment ⇒ x : 1 0 1 0 ← i/p A to Ra1

Z : f(x) : 0 1 1 0 ← o/p

Q.) If the initial state is not known what is min i/p string? Take the machine into state C.

(a) 01 (b) 10 (c) 101 (d) None