

## 1. LOGIC FUNCTIONS

### 1. BASIC PROPERTIES OF SWITCHING ALGEBRA

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

$$(xy) \cdot z = x(yz)$$

Idempotency:  $x \cdot x = x$  |  $x+1 = 1$ 

$$x+x = x$$
 |  $x \cdot 0 = 0$

$$x \cdot 1 = x$$

$$x \cdot 0 = x$$

Commutativity:  $x+y = y+x$ 

$$xy = y \cdot x$$

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

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

+ = OR

• = AND

- = NOT

Distributive:  $x(y+z) = xy+xz$ 

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

### 2. SWITCHING EXPRESSIONS AND SIMPLIFICATIONS

Switching expression is a finite no. of combinations of switching variables and constants {0,1} by means of switching operations (+, •, NOT).

Ex:  $x + \bar{x}y^2 + x\bar{z}$ ,  $a+bc+b\bar{d}$ 

Properties for simplifying SE

$$\begin{aligned}
 \text{Absorption: } & x+xy=x = x \cdot 1 + xy & x+x'y = x+y & x(x'+y) = xy \\
 & = x(1+y) & \text{LHS} = (x+x')(x+y) & x \cdot x' + xy \\
 & = x(1) & = 1 \cdot (x+y) & = 0 + xy \\
 & = x, & \text{RHS} & = xy = \text{RHS}.
 \end{aligned}$$

\*\* Consensus theorem:  $xy + \bar{x}z + yz = xy + \bar{x}z$ 

$$\begin{aligned}
 &= xy + \bar{x}z + yz(1) \\
 &= xy + \bar{x}z + yz(x+\bar{x}) \\
 &= xy + \bar{x}z + yzx + yz\bar{x} \\
 &= xy(1+z) + xz(1+y) \\
 &= xy + \bar{x}z
 \end{aligned}$$

Minimize  $x'y'z + yz + xz$ 

$$\begin{aligned}
 &= z(x'y' + y + x) \\
 &= z((x'+y)(y'+y) + x) \\
 &= z(x'+y+x) = z(1+y) \\
 &= z
 \end{aligned}$$

If the value of an Exprn  
does not depend on any  
term then it is called  
Redundant Exprn

### 3. DEMORGAN'S LAW AND SIMPLIFICATION

(2)

$$\Rightarrow (\bar{x}y) = \bar{x} + \bar{y} \quad (2) (\bar{x+y}) = \bar{x}\bar{y}$$

$$f(a, b, c, \dots, z, 0, 1, \cdot, +) \\ \bar{f} = (\bar{a}, \bar{b}, \bar{c}, \dots, \bar{z}, 1, 0, +, \cdot) \\ f_d = \text{duality} = (a, b, c, \dots, z, 1, 0, +, \cdot)$$

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

$$\text{Simplify } (x+y)[x'(y'+z')]' + x'y' + x'z'$$

$$= (x+y) [x + (y+z)] + x'y' + x'z' \\ = x + \underline{xy^2 + yx + y^2 + x'y' + x'z'} \\ = x(1+y^2) + yx + y^2 + x'y' + x'z' \\ = x + xy + y^2 + x'y' + x'z' \\ = \underline{x(1) + y^2 + x'y' + x'z'} \\ = (x+x')(x+y') + y^2 + x'z' \\ = \underline{x+y' + y^2 + x'z'} \\ = x + z' + y' + y^2 \\ = \underline{x+z' + y' + z} = x+y'+1 = x+y'+1 \\ = 1$$

### 5. CANONICAL SUM OF PRODUCTS

① A product term which contains each of 'n' variables as factors either in complemented or uncomplemented form is called a minterm.

② A minterm given the value '1' for exactly one combination of the variables.

③ The sum of all minterms of 'f' for which 'f' assumes '1' is called canonical sum of products or disjunctive Normal form.

$f(a, b, c)$  then the no. of min terms = 8

$f(a, b, c, \dots, n)$  then the no. of min terms =  $2^n$

(2)

$$\text{i)} f(a, b, c) = \bar{a}\bar{b}c + \bar{a}b\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c} + abc \quad (3)$$

$= ab$  is not a minterm (it should contain all the variables  $a, b, c$  in complemented or

ii) 2nd point:  $\bar{a}\bar{b}\bar{c}$   
 $\Rightarrow 000 \Rightarrow \bar{a}\bar{b}\bar{c}=1$   $\left\{ \begin{array}{l} \bar{a}\bar{b}c \\ 001 \end{array} \right. \Rightarrow \bar{a}\bar{b}c=1 \}$   
 for the values of 001 the minterm  $\bar{a}\bar{b}c$  gives 1.

a	b	c	f
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	1
1	1	1	0

canonical sum of products

Sum of products  $\Rightarrow$  need not contain all the variables  
 $f(a, b, c) \Rightarrow a\bar{b}c + \bar{a}c$

Canonical sum of products  $\Rightarrow$  should contain all the variables

$$f = \underbrace{\bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + ab\bar{c}}_{\text{canonical sum of products.}} = \sum(1, 2, 4, 6) \rightarrow \text{compact representation.}$$

## 6. CANONICAL PRODUCT OF SUMS

$\rightarrow$  A sum term which contains each of 'n' variables as factors either in complemented or uncomplemented form is called a maxterm.

$\rightarrow$  A Maxterm gives the value '0' for exactly one combination of the variables.

$\rightarrow$  The product of all maxterms of 'f' for which 'f' assumes '0' is called Canonical product of sums or Conjunctive Normal form.

	a	b	c	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	0

$$f = [(a+b+c)(a+b+\bar{c})(a+\bar{b}+c)(\bar{a}+b+\bar{c})](\bar{a}+\bar{b}+\bar{c})$$

I = complemented form  
 O = uncomplemented form.

### 7. EXAMPLES ON CANONICAL FORMS

(4)

a	b	c	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>
1) 0	0	0	0	1	1
2) 0	0	1	0	0	1
3) 0	1	0	1	1	0
4) 0	1	1	1	0	0
5) 1	0	0	0	1	0
6) 1	0	1	1	0	0
7) 1	1	0	1	1	1
8) 1	1	1	1	0	1

$$f_1 = \text{sum of products} = \text{minterms} = \Sigma(3, 4, 6, 7, 8)$$

$$\text{product of sum} = \pi(1, 2, 5)$$

$$f_2 = \text{sum of products} = \Sigma(1, 3, 5, 7)$$

$$\text{product of sum} = \pi(2, 4, 6, 8)$$

$$f_3 = \Sigma(1, 2, 7, 8) = \pi(3, 4, 5, 6)$$

$$\triangleright f(x, y, z) = x'y + z' + xyz \quad \text{convert into canonical form?}$$

$$\begin{aligned}
 f(x, y, z) &= x'y + z' + xyz \quad (\text{sum of products}) \text{ but not (canonical sum of products)} \\
 &= x'y(z+z') + z'(x+x')(y+y') + xyz \\
 &= x'yz + x'yz' + z'(xy+xy'+x'y+x'y') + xyz \\
 &= x'yz + \underbrace{x'yz'}_{xy} + \underbrace{z'(xy)}_{xy} + \underbrace{x'yz'}_{xy'} + \underbrace{x'yz'}_{xy} + xyz \\
 &= x'yz + x'yz' + z'xy + xyz' + x'y'z' + xyz \\
 &= \Sigma(0, 2, 3, 4, 6, 7) = \pi(1, 5)
 \end{aligned}$$

### 8. FUNCTIONAL PROPERTIES

- The Canonical SOP or POS form a switching function is unique.
- Two switching functions  $f_1(x_1 \dots x_n)$  and  $f_2(x_1 \dots x_n)$  are said to be logically equivalent iff both functions have same value for each and every combination of  $(x_1, x_2, \dots, x_n)$ .
- Two switching functions are equivalent if their canonical POS or SOP are identical.

### 9. NO. OF FUNCTIONS

n - Boolean variables

How many boolean variables

1	2	3	4	...	n	f
$2 \times 2$	$2 \times 2$	$2 \times 2$	$2 \times 2$	...	$2 \times 2$	

combinations of  
 $\Rightarrow 2^n$  boolean variables are  
present  
 $\Rightarrow$  Now, each combination has  
2 choices (maybe present/may  
be absent)  
 $\therefore$  No. of Boolean functions possible are  $= 2^{2^n}$

(4)

⇒  $n$ -ternary variables

⇒ no. of combinations of  $n$ -ternary variables =  $3^n$

(3,4,6,7,8)

$$\Rightarrow \boxed{\text{No. of functions} = (2)^{(3^n)}}$$

(5)

$$\begin{matrix} 1 & 2 & 3 & \dots & n \\ \downarrow & \downarrow & \downarrow & \dots & \downarrow \\ 3 & 3 & 3 & \dots & 3 \end{matrix}$$

⇒  $n$  K-ary variables, How many m-ary functions possible?

1	2	3	...	$n$	$f$
K	K	K	...	K	

⇒  $K^n$  combinations

$$\therefore \boxed{\text{No. of functions} = m^{(K^n)}}$$

lucts)

## 10. COUNTING NO. OF FUNCTIONS AND NEUTRAL FUNCTIONS

1) How many boolean functions are possible with 3 variables such that there are exactly 3 minterms?

a	b	c	$f$
0	0	0	
0	0	1	
0	1	0	0
0	1	1	0
1	0	0	
1	0	1	
1	1	0	
1	1	1	0

⇒ Now, we should assign '3' ones to the 8 combinations of the boolean variables.

$$\boxed{8C_3 \text{ ways}}$$

HOOTING 23RD DAY - 1

HOOTING 23RD DAY - 1

2) Atmost 3 minterms. =  $\boxed{8c_0 + 8c_1 + 8c_2 + 8c_3}$

3) Atleast 3 minterms

$$\Rightarrow \boxed{8c_3 + 8c_4 + 8c_5 + \dots + 8c_8}$$

n

3) Assume that there are  $K$  variables then  $2^K$  combinations of boolean variables are possible, Now if they want to assign ' $m$ ' minterms exactly then the

$$\text{no. of ways} = (2^K)c_m$$

4) How many Neutral functions are possible with two boolean variables?

A Neutral func is a func in which no. of minterms = No. of maxterms.

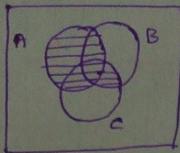
A	B	$f=A$	$f=\bar{A}$	$f=\bar{B}$	$f=B$	$f=1$	$f=0$
		EX-NOR	EX-OR	↑	↑	$f=1 \oplus B$	
0	0	1	1	0	1	0	
0	1	1	0	1	0	1	
1	0	0	1	0	0	1	
1	1	0	0	1	1	1	

(No. of 1's = No. of 0's)

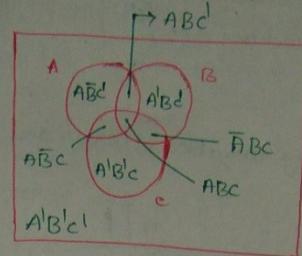
$$\left\{ \begin{array}{l} \text{"n' variables = } 2^n \text{ combinations} \\ = (2^n)c_m = (2^n)c_{2^n} \\ = (2^n)c_{2^n} = (2^n)c_{(2^n-1)} \end{array} \right.$$

## II. VENN DIAGRAM REPRESENTATION

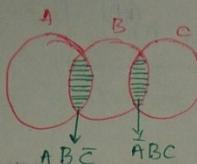
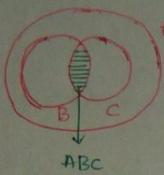
Find the Boolean function that the following Venn diagrams represents.



- a)  $A + AB'C$
- b)  $A + BC$
- c)  $A + A'BC$
- d)  $AB + C$



$$F = [A + \bar{A}BC] \rightarrow A + BC$$



## 13. NESTED

In the fall  
of  $w, x, y, z$

i)  $x+y+z$

ii)  $xy+yz$

iii)  $xw+xz$

②  $f(A, B) = A + B$

Now, f

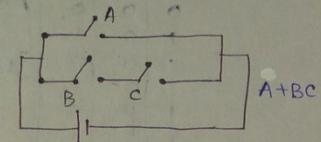
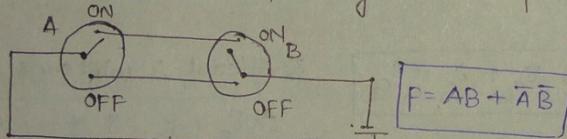
## 12. CONTACT REPRESENTATION

→ Every Boolean function can be represented with the help of serial and parallel contact

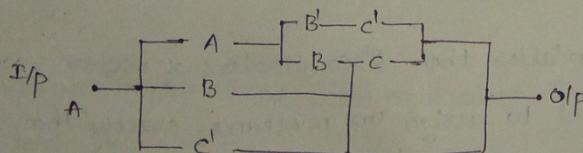
→ Serial contacts are performing "AND" operation

→ Parallel contacts are performing "OR" operation.

Which function does this following circuit represent.



I) Identify the boolean expression given by the following circuit



① Find the valid forward path

② perform OR among them.

Forward path:

Any path starting from I/P and ending at o/p without forming cycle

Validity: No path should contain a variable in both true and complemented form.

In above example core:  $AABC + AABC + AABA + ABC + ABA + A'c \rightarrow$  Invalid forward paths.

## 14. NAND GATE

NAND = N

A	B
0	0
0	1
1	0
1	1

## 15. NOR GATE

NOR = N

A	B
0	0
0	1
1	0
1	1

## 16. EX-OR

A	B
0	0
0	1
1	0
1	1

⑥

### 13. NESTED FUNCTION

In the following simultaneous Boolean Expressions what are the values of  $w, x, y, z$ .

$$i) x+y+z=1$$

$$\textcircled{1} \begin{array}{cccc} x & y & z \\ 0 & 0 & 0 & 1 \end{array} X$$

$$ii) xy+wz=0$$

$$b) \begin{array}{cccc} 1 & 1 & 0 & 1 \end{array} X$$

$$iii) xw+yz=1$$

$$c) \begin{array}{cccc} 0 & 1 & 0 & 1 \end{array} \checkmark$$

$$d) \begin{array}{cccc} 1 & 0 & 0 & 0 \end{array}$$

$$\textcircled{2} f(A, B) = A' + B \text{ then find } f(f(x+y, y), z)$$

$$\text{Now } f(x+y, y) = (x+y)' + y$$

$$= x'y' + y$$

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

$$\text{Now } f(x+y, z)$$

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

$$= xy' + z$$

### 14. NAND GATE AND PROPERTIES

1. a)

$$\text{NAND} = \text{NOT AND} = (\overline{A \cdot A})$$

$\Rightarrow$  NAND gate does not obey Identity

+BC

A	B	$A \uparrow B$
0	0	1
0	1	1
1	0	1
1	1	0

$$\Rightarrow A \uparrow A \neq A$$

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

$$\boxed{\overline{A \cdot B} = B \uparrow A}$$

$$\Rightarrow A \uparrow (B \uparrow C) \neq (A \uparrow B) \uparrow C$$

$\downarrow$  NAND is not Associative

### 15. NOR GATE AND PROPERTIES

$$\text{NOR} = \text{NOT OR}$$

A	B	$(A \downarrow B)$
0	0	1
0	1	0
1	0	0
1	1	0

$$\Rightarrow A \downarrow B = (\overline{A+B})$$

$\Rightarrow$  NOR is not Associative

$$\Rightarrow (\overline{A \downarrow A}) \neq (\overline{A}) = \overline{(\overline{A})}$$

$$\overline{A \downarrow (B \downarrow C)} = (\overline{A \downarrow B}) \downarrow C$$

$$\Rightarrow (\overline{A \downarrow B}) = (B \vee A)$$

### 16. EX-OR GATE AND PROPERTIES

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

$$A \oplus B = \overline{AB} + A\overline{B} \text{ (modulo 2 sum)}$$

$$1 \oplus 1 \oplus 1 = 3 \bmod 2 = 1 \checkmark$$

$$1 \oplus 0 \oplus 1 = 2 \bmod 2 = 0 \checkmark$$

$$\Rightarrow A \oplus A \neq A \quad (1 \oplus 1 = 0 \neq 1) \quad \left\{ A \oplus (B \oplus C) = (A \oplus B) \oplus C \right.$$

$$\Rightarrow A \oplus B = B \oplus A \quad \left. \left\{ \overline{B \oplus C} = \overline{B} \oplus C = C \oplus \overline{B} \right. \right.$$

## 17. EX-NOR GATE AND PROPERTIES

EX-NOR = Exclusive - NOR = Negation of EX-OR

A	B	$A \oplus B$
0	0	1
0	1	0
1	0	0
1	1	1

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

$$\Rightarrow A \oplus A = A$$

$$\Rightarrow A \oplus B = B \oplus A$$

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

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

$$\Rightarrow \overline{A \oplus B} = A \odot B = A \oplus B'$$

$$\Rightarrow \overline{A \oplus B} = A \oplus B = A' \oplus B = A \oplus B'$$

$$\Rightarrow \overline{A \oplus B} = A \odot B = A' \oplus B = A \oplus B$$

## 18. PROPERTIES OF EX-OR AND EX-NOR

Now we have checked the EX-OR and EX-NOR both are complements to each other.

XOR

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

X-NOR

A	B	$A \oplus B$
0	0	1
0	1	0
1	0	0
1	1	1

$$\oplus = 1 \text{ for odd no. of } 1's$$

$$\odot = 1 \text{ for even no. of } 0's$$

Now, for three variables both XOR and X-NOR are showing same? why?

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

Now, Let  $n = \text{no. of inputs}$

If  $n$  is even:

If 1's are odd  $\Rightarrow$  no. of 0's are odd  
for such combinations  $\oplus = 1 \quad \odot = 0$

If  $n$  is even:

If no. of 1's even  $\Rightarrow$  no. of 0's are even

$$\oplus = 0, \odot = 1$$

$\therefore$  If  $n = \text{even}$  both XOR, X-NOR are Complement

If  $n = \text{odd}$

$\Rightarrow$  If 1's odd  $\Rightarrow$  0's is even

$$\oplus = 1, \odot = 1$$

$\Rightarrow$  If 1's even  $\Rightarrow$  0's is odd

$$\oplus = 0, \odot = 0$$

$\therefore$  If  $n = \text{odd}$  both XOR, X-NOR are same

Now, draw

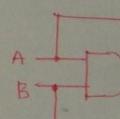
$A \oplus B$

AB
00
01
10
11

{ ones, an  
no. of  
we ca

## 19. MINIMUM

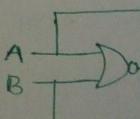
Ex-OR (U)



Min. no. o

build X

Ex- NOR (C)

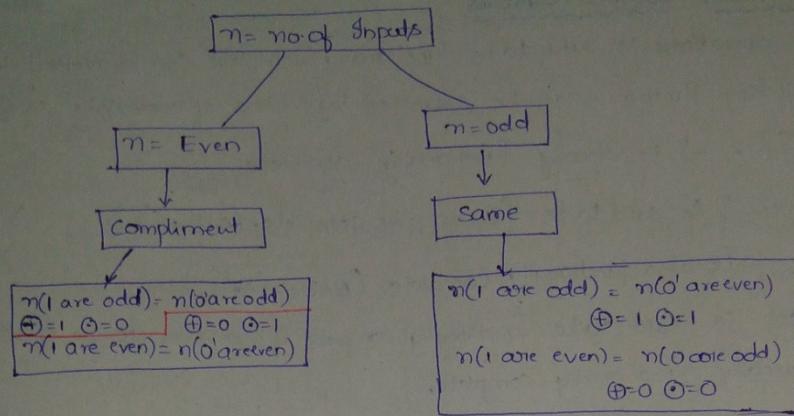


Min. no. of

construct X

⑧

$$\begin{aligned} \text{OB}' \\ \text{OB} = A \oplus B \\ (\oplus B) = A \oplus B \end{aligned}$$



both

Now, drawing Karnaugh maps

i)  $A \oplus B \oplus C \oplus D$ 

	AB	CD	00	01	11	10
CD	00	00	1	1	1	1
00	01	1	1	1	1	1
01	10	1	1	1	1	1
10	11	1	1	1	1	1

EX-OR

ones are filled at positions where  
 no. of 1s are odd  $\therefore$  it is EX-OR  
 we cannot minimise them further

ii)  $A \oplus B \oplus C \oplus D$ 

	AB	CD	00	01	11	10
CD	00	00	1	1	1	1
00	01	1	1	1	1	1
01	10	1	1	1	1	1
10	11	1	1	1	1	1

EX-NOR

ones are filled at positions where  
 no. of 1s are even  $\therefore$  it is EX-NOR.  
 we cannot minimise them further.

## 19. MINIMUM NUMBER OF GATES REQUIRED XOR AND XNOR

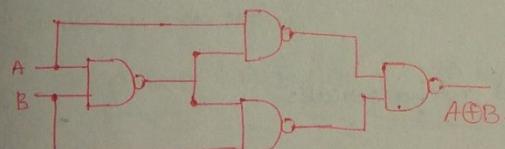
obtained

$\ominus=0$

one even

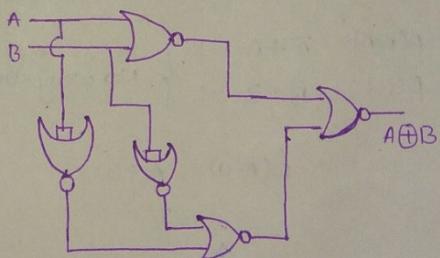
compliment

EX-OR (using NAND)



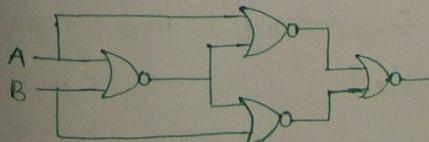
Min no. of NAND gates Req'd for  
build XOR gate is 4

EX-OR (using NOR)



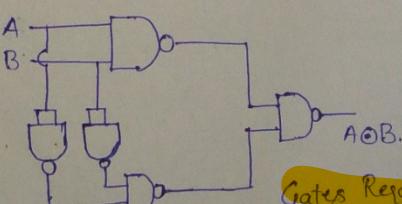
Min. no. of NOR gates Req'd to build  
XOR gate = 5.

EX-NOR (using NOR)



Min. no. of NOR gates required to  
construct XNOR is 4

EX-NOR (using NAND)



Gates Req'd = 5.

## 20. FUNCTIONALLY COMPLETENESS

(10)

- ⇒ A set of operations is said to be functionally complete (or) universal if and only if every switching function can be expressed by means of operations in it.
- ⇒ The set  $\{+, \cdot, -\}$  is clearly functionally complete
- ⇒ The set  $\{+, -\}$  is said to be functionally complete. ( $A \cdot B = (\overline{A} + \overline{B})$ )
- ⇒ The set  $\{\cdot, -\}$  is also functionally complete ( $A + B = (\overline{A} \cdot \overline{B})$ )
- ⇒ NOTE: A set is said to be functionally complete if we can derive a set which is already functionally complete.

23. E

1> F C

27 f(A)

f  
f

## 21. EXAMPLE 1 ON FUNCTIONAL COMPLETENESS

- 1>  $F(A, B, C) = A' + BC'$  is this functionally complete?

For it to be functionally complete it should definitely derive  $-$  (NOT) and  $(+ \text{ or } \cdot)$

$$\Rightarrow f(A, B, C) = A' + BC' \therefore (-, \cdot) (-, +)$$

⇒ Now check  $f(A, A, A) = A' + A \cdot A' = A' \therefore$  complement can be realised.

$$\Rightarrow \underbrace{f(f(A, A, A), B, f(B, B, B))}_{\begin{array}{c} A' \\ B \\ B' \end{array}} \Rightarrow f(A', B, B') = (A')' + B(B')' \\ \boxed{f(A', B, B') = A + B}$$

37 f

## 22. EXAMPLE 2 ON FUNCTIONAL COMPLETENESS

- 1>  $F(A, B) = (\overline{A} + B)$  is this functionally complete?

$$F(A, B) = \overline{A} + B$$

$$\left. \begin{array}{l} f(A, A) = \overline{A} + A = 1 \\ f(B, B) = \overline{B} + B = 1 \end{array} \right\} \text{No way we can get rid of a variable}$$

$$F(A, 0) = \overline{A} + 0 \\ 0 = \overline{A}$$

$$\text{Now, } F(f(A, 0), B)$$

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

} Using  $\vee$  we can make it complete so when we take support from '0' and '1' then we call it partially complete.

4> An

fol

(10)

## 3.4.5.6 ON FUNCTIONAL COMPLETENESS

(11)

if and only if  $f(A, B) = \bar{A}B$ 

$$f(A, A) = \bar{A} \cdot A = 0.$$

$$f(B, B) = \bar{B}B = 0.$$

$$f(A, 1) = \bar{A} \quad (\text{Complementation Achieved}).$$

$$f(f(A, 1), B) = (\bar{A})B = AB.$$

} partially functionally complete.

which

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

$$f(AAA) = A + A + A$$

$$f(AAA) = A$$

} complementation cannot be achieved (Not functionally complete)

$\Rightarrow$  If a function does not contain Complement then it cannot be functionally complete.

(v.)

(+)

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

$$\begin{aligned} f(x, x) &= \bar{x}x + x\bar{x} \\ &= 0 + 0 = 0. \end{aligned}$$

$$f(x, 1) = \bar{x}1 + x(0)$$

$$f(x, 1) = \bar{x}$$

I'm getting complement with '1'

$$\begin{aligned} f(y, y) &= \bar{y}y + y\bar{y} \\ &= 0 + 0 = 0. \end{aligned}$$

⇒ partially functionally complete

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

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

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

$\{\cdot, -\} \quad \{+, -\}$

cannot be achieved.

∴ Not functionally complete.

4) Any Boolean function can be defined with which of the following operations

a)  $\oplus$ , NOTb)  $\oplus, 1$ , ORc)  $\oplus, 1$ , NOTd)  $\odot, 1$ , NOT

$\oplus$  does not lead  
to  $\{+, \cdot\}$  check

above example

partially complete

↓  
Functionally

complete.

## 27. SELF DUAL FUNCTIONS

$$1) f(A, B, C) = AB + BC + CA$$

$$2) f_d(A, B, C) = (A+B)(B+C)(C+A)$$

$$= AB + BC + CA$$

$$2) f(A, B, C) = AB(C+C') + (A+A')BC + C(B+B')A$$

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

\* Boolean function is self dual if

1) It is Neutral (No of minterms = No. of maxterms)

2) The function does not contain two mutually exclusive terms.

$$\begin{aligned} (ABC) &\rightarrow (A'B'C') \\ (A'B'C') &\Rightarrow (A'BC) \end{aligned} \quad \left. \begin{array}{l} \text{mutually Exclusive} \end{array} \right\}$$

## 28. NO. OF SELF DUAL FUNCTIONS

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

(0,7) (1,6) (2,5) (3,4)  $\Rightarrow$  ME pairs

$$F = 2 \times 2 \times 2 \times 2 = 2^4 = 16$$

(0 1 2 3) = self dual function.

$$\therefore \text{No. of self dual functions} = 2^{(2^n)}$$

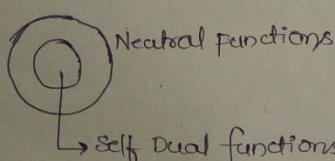
$\therefore n$  variables  $\rightarrow 2^n$  terms

$\downarrow$  divide into pairs (having two in each set)

$$2^n/2 = 2^{n-1} \text{ pairs}$$

$\downarrow$  Each pair has two choices (include 1st ele or 2nd ele)

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



$$\therefore \text{No. of self dual functions} = \left[ 2^{(2^{n-1})} \right]$$

## 29. SELF DUAL FUNCTIONS ARE CLOSED UNDER COMPLEMENTATION

Which of the following functions are closed under complementation?

$$1) f(A, B, C) = \sum(0, 1, 2, 3) = \text{No. of minterms} \neq \text{No. of maxterms} \Rightarrow \text{Not self dual}$$

$$2) f(A, B, C) = \sum(0, 1, 6, 7) = (0, 7) \cup (1, 6) \cup (2, 5) \Rightarrow \text{Neutral but not self dual (contains both elec. mutual exclusive pair)}$$

(12)

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

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

$\Rightarrow$  self dual

30. INTRO

1) Electro output

2) The pa

of log

3) The h

high

4) Thus

5) If w

called

6) If w

then

$$3) f(A_1, B_1, C) = \sum(0, 1, 2, 4) = (0, 1)(1, 6)(2, 4) \Rightarrow \text{No. of minterms} = \text{N. of maxterms} & \text{ME pairs.}$$

one unique (SD)

$$4) f(A_1, B_1, C) = \sum(3, 5, 6, 7) = (3, 4)(5, 2)(6, 1)(0, 7) \Rightarrow MEC(v) N(min) = N(max) = SD$$

$\Rightarrow$  self dual is closed under Complementation (Complement of selfdual function is selfdual)

### 30. INTRODUCTION TO ELECTRONIC GATES

(13)

- 1) Electronic gates generally receive voltages as inputs and produce voltages as outputs.
- 2) The precise values of these voltages are not significant towards determination of logical operation of gates.
- 3) The significant point is that voltages are restricted to two ranges of values high and low.
- 4) Thus two valued variables may be used to represent these voltages
- 5) If we associate constant 1 with high voltage and 0 with low voltage, it is called positive logic system.
- 6) If we associate the constant 1 with low voltage and 0 with high voltage then it is called Negative logic system.

## 2. MINIMISATION

(14)

### 1. INTRODUCTION TO MINIMISATION OF BOOLEAN EXPRESSIONS

→ A switching function can usually be represented by using a no. of expressions.

$$\begin{aligned} \text{Ex: } f(x_1, y_1, z) &= x_1 y_1 + y_1 z + z x_1 \\ &= x_1 y_1 z + x_1 y_1 z' + x_1 y_1' z + x_1 y_1' z' \end{aligned}$$

→ While simplifying a switching function  $f(x_1, x_2, \dots, x_n)$  our aim is to find an expression  $g(x_1, x_2, \dots, x_n)$  which is equivalent to  $f$ , which minimises some cost criteria.

Criteria to determine minimal cost:

- 1) Minimum no. of appearances of literals
- 2) Min no. of literals in SOP or POS expression
- 3) Minimum no. of terms in SOP expression, provided there is no other such expression with the same no. of terms and fewer literals.

### 2. IRREDUNDANT OR IRREDUCIBLE EXPRESSIONS

$$F(x_1, y_1, z) = x_1' y_1 z + x_1' y_1 z' + x_1 y_1 z + x_1 y_1 z' + x_1 y_1' z + x_1 y_1' z'$$

$$= x_1' z (y_1 + y_1') + x_1 y_1 z + y_1 z (x_1 + x_1') + x_1 y_1' z + x_1' z (y_1 + y_1')$$

$$= x_1' z + x_1 y_1 z + y_1 z + x_1' z$$

$$= (x_1' + x_1 y_1) z + (y_1 + x_1) z$$

$$= ((x_1' + x_1)(z + y_1)) z + ((y_1 + x_1)(z + y_1')) z$$

$$F(x_1, y_1, z) = x_1' z + y_1 z + y_1' z$$

$$F = x_1' z + y_1 z + y_1' z$$

⇒ Minimal is not always unique.

→ An SOP expression from which no term or literal can be deleted without altering its logical value is called an irredundant or irreducible expression.

→ Note: An irredundant expression is not necessarily be minimal, nor the minimal expression always unique.

### 3. K-MAP INTRODUCTION

→ The algebraic procedure of combining various terms and applying to them the rules becomes very tedious as the no. of various variables increases.

→ The map method provides a systematic method for combining the terms and derive minimal expression.

→ A K-

of co-

→ Every

of va-

x

o

4. K-M

1) A co-  
and -

2) Each

'n' =

3) Any

4) A fun-

to 'su-

5) The  
while

6) Theref-

smar-

5. EXAM

Minimize

y2

∴ F

→ A K-map is a modified form of the truth table in which the arrangement of combinations is particularly convenient for minimizing (15)

→ Every  $m$ -variable map consists of  $2^m$  "cells", representing all possible combinations of variables.

$x$	$y$	0	1
0	0	0	1
1	0	2	3

2v - K-map.

$x$	$y$	00	01	11	10
0	0	0	2	6	4
1	1	1	3	7	5

3v - Kmap

$wx$	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

4v - K-map

#### 4. K-map Simplification

1) A collection of  $2^m$  cells, each adjacent to 'm' cells of collection is called a subcube, and the subcube is said to cover these cells.

2) Each subcube can be expressed by a product containing  $n-m$  literals where 'm' = no. of variables on which function depends.

3) Any cell may be included in as many subcubes as desired.

4) A function 'f' can be expressed as sum of those product terms which corresponds to subcubes necessary to cover all its '1' cells.

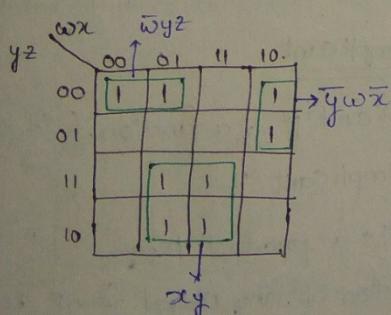
5) The no. of product terms in the expression for 'f' is equal to the no. of subcubes while the no. of literals in each term is determined by size of corresponding subcubes.

6) Therefore to obtain a minimal expression we must cover all '1' cells with smallest no. of subcubes as long as possible.

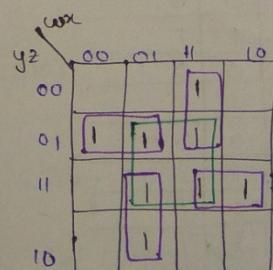
#### 5. EXAMPLES ON K-MAP

$$\text{Minimize } f(w, x, y, z) = \sum(0, 4, 6, 7, 8, 9, 15)$$

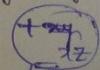
$$② F(w, x, y, z) = \sum(1, 5, 6, 7, 11, 12, 13, 15)$$



$$\therefore F = \bar{w}yz + \bar{y}w\bar{x} + xy$$



$$F = \bar{w}\bar{y}z + w\bar{x}\bar{y} + \bar{w}xy + wyz$$



Not Required.

## 6. COVERING FUNCTIONS

(6)

If the minterms present in 'g' are also present in 'f' then 'f' is said to cover 'g'.

Implicants:

A switching function  $f(x_1, x_2, \dots, x_n)$  is said to be cover of  $g(x_1, x_2, \dots, x_n)$  denoted by "f is superset of g" if f assumes true value whenever g does.

Note: If 'f' covers 'g' and 'g' covers 'f' then both 'f' and 'g' are equivalent.

→ If 'g' has  $x$  min terms and 'g' is a function of 'n' variables, then no. of covering functions of g =  $\frac{2^{n-x}}{2}$

Ex:  $f(w,x,y,z) = wx + yz$  how many functions are covering f?

$$\Rightarrow \text{No. of min terms possible} = 2^4 = 16 \text{ minterms}$$

$\Rightarrow$  7 minterms are chosen Remaining = 9 minterms

$$\Rightarrow \text{No. of covering functions} = 2^9 - 2 = 512$$

$$\begin{array}{c} \rightarrow \\ \begin{array}{ccc} wx & - & - \\ \begin{array}{cc} 00 \\ 01 \\ 10 \end{array} & + & \begin{array}{cc} - & - \\ 00 \\ 01 \\ 10 \end{array} \\ \boxed{11} & & \boxed{11} \end{array} \end{array}$$

$$= \text{No. of minterms} = 7$$

$$\text{No. of remaining terms} = 16 - 7 = 9.$$

## 7. IMPLICANTS AND PRIME IMPLICANTS

→ If 'f' covers 'g' then 'g' is said to imply 'f'. This denoted by  $g \rightarrow f$

Ex:  $f(a,b,c) = ab + c$

a	b	c	$f = ab + c$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Both ab, c are implicants of 'f'.

Prime Implicant

→ An implicant 'P' of a function 'f' is said to be prime implicant if

i) 'P' is a product term

ii) Deletion of any literal from P results in a new product which is not covered by 'f' (i.e. A subcube cannot be part of any other subcube)

## 8. ESSENTIAL IMPlicants

A prime implicant

if it covers

implicants

of

1

## 9. PROCESS

1) Delete

2) Remove

prime

3) If the

minim

the

prime

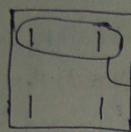
Ex

w	1	0	0
y	1	1	0
z	0	1	1
wz	1	0	0
wy	0	1	0
yz	0	0	1
wyz	1	1	1

## 8. ESSENTIAL PRIME IMPLICANTS

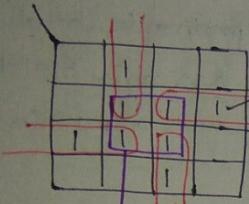
(17)

A prime implicant 'P' of a function 'f' is said to be an essential prime implicant if it covers atleast one minterm of 'f' which is not covered by any other prime implicants.



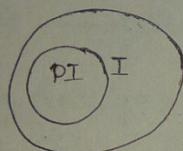
Implicant (prime Implicant)

Implicant (Not prime Implicant because it is completely covered by another implicant.)



This one cannot be covered by any other any other Implicant  
Essential prime Implicants (It cannot be covered by any other Implicant).

prime Implicant



I = Implicants

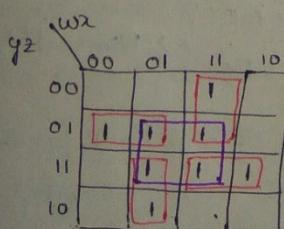
PI = prime Implicant

## 9. PROCEDURE FOR OBTAINING MINIMAL SOP

- 1) Determine all essential prime implicants and include them in minimal sop.
- 2) Remove from list of prime implicants all those which are covered by essential prime implicants.
- 3) If the set determined in step 1 covers all the minterms of 'f', then it is unique minimal expression, otherwise select the additional prime implicants so that the function 'f' is covered completely and the total number and size of prime implicants added are minimal.

Ex

be



$$F = \Sigma(1, 5, 6, 7, 11, 12, 13, 15)$$

$$f = \bar{w}yz + wz\bar{y} + w\bar{y}z + \bar{w}\bar{y}z \quad (\text{Unique minimal expression})$$

Now, to check whether the above minimal expression is minimal or not then we go for Tabulation method.

	1	5	6	7	11	12	13	16
* $wyz$	1	1						
* $w\bar{y}$					1	1		
* $wyz$				1				
* $\bar{w}xy$		1	1					
$xz$	1	1				1	1	

Now,  $wyz$  covers the boxes 1,5 in K-map. So put 1,1 in 1,5 column and  $wyz$  row, and mark 1,5 (marked by  $*$ ) are covered.

→ Similarly do for all minterms. Implicants (Rows). Column numbered

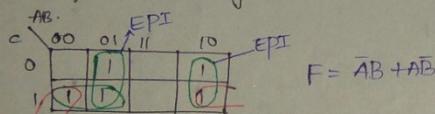
→ Now check 1 minterm (1 column) it is covered only with  $wyz$  so it is a prime implicant that is essential so put \* on  $wyz$  and it also covers 5

→ Similarly the min term '6' is covered by  $wxy$  so  $wxy$  is the essential prime implicant.

→ Similarly analyze each column and mark the essential prime implicants and also mark the minterms that are also covered by the prime implicants. If all the minterms are covered then check \*'s on the Rows, and they will form minimal expression.

### 10. MINIMAL SOP EXAMPLE

Which of the following prime implicants are essential?



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

a)  $Bc, A'B$ .

Now, if they have asked how many minimal expressions

b)  $A'C, A'B$ .

are there then how to find is.

c)  $A'B, AB'$  ✓

Equally probable candidates,

d)  $A'B, AB', B'C$

	1	2	3	4	5
* $A'B$	1				
* $AB'$			1		
$A'C$	1				
$B'C$	1				

$$\therefore F = \bar{A}\bar{B} + \bar{A}\bar{B} + \bar{A}C \quad \left. \begin{array}{l} \text{2 Minimal Expression} \\ F = \bar{A}\bar{B} + A\bar{B} + BC \end{array} \right.$$

### 11. MINIMAL POS

Continued in Nextpage

①  $F(w, x, y, z) = :$

wz	00	01
00	0	0
01	0	1
11	0	0
10	0	1

( $w+x$ )

②

wx	00	01
00	1	1
01	1	0
11	1	1
10	1	0

### 12. EXAMPLES

Find the mo

$$f(w, x)$$

wx	00	01
00	1	1
01	0	0
11	1	0
10	1	0

### 13. INTRODUCTION

→ A function is of variables.

→ Combinations

→ The value of

→ Since each specified function of  $2^k$  dis

→ our task is to

→ we could as

in a minute

the boxes  
in 1,5 column  
is means they  
are covered

row numbered  
↑  
column)

$y_2 \otimes z$

is essential

also covers 5

prime implicant

00

the

a)

$$\textcircled{1} F(w, x, y, z) = \Sigma(5, 6, 9, 10)$$

$wx$	00	01	11	10
$y_2$	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	0	0	0	0
10	0	1	0	1

(1) Examples on Don't Cares -

$$\therefore f = (y+z)(w+x)(\bar{y}+\bar{z})(\bar{w}+\bar{x}) = pos.$$

$$= \bar{w}x\bar{y}z + w\bar{x}\bar{y}z + \bar{w}x\bar{y}\bar{z} + w\bar{x}y\bar{z} = sop$$

\textcircled{2}

$wx$	00	01	11	10
$y_2$	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	1	1	1	1
10	1	0	0	0

$$Sop = \bar{y}\bar{z} + \bar{w}\bar{x} + w\bar{x} + yz$$

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

## 12. EXAMPLES ON MINIMAL POS.

Find the no. of literals in minimum pos and sop for

$$f(w, x, y, z) = \Sigma(1, 5, 6, 7, 11, 12, 13, 15)$$

essions

$wx$	00	01	11	10
$y_2$	00	01	10	11
00	1	1	0	0
01	0	0	0	1
11	1	0	0	0
10	1	0	1	1

$$pos = (\bar{w}+\bar{x})+y * (w+y+\bar{z})(\bar{w}+\bar{y}+\bar{z})(w+\bar{x}+\bar{y}+z)$$

$$Sop = w\bar{x}\bar{y} + w\bar{y}\bar{z} + \bar{w}\bar{x}y + \bar{w}\bar{y}\bar{z}$$

## 13. INTRODUCTION TO DON'T CARES

→ A function is said to be completely specified if it is given by '1' for every combination of variables. There exist some functions which are not completely specified.

→ Combinations for which the value of a function is not specified are called don't care combinations.

→ The value of a function for such combination is denoted by ' $\phi$ ' or 'd'.

→ Since each don't care combination represents 2 values {0, 1}, an incompletely specified function containing  $k$ -don't care combinations corresponds to a class of  $2^k$  distinct functions.

→ Our task is to choose a function having minimal representation out of these  $2^k$  functions.

→ We could assign a '0' or '1' to a '0' or '1' to a don't care combination in such a way to increase or decrease size of a subcube.

#### 14. EXAMPLES ON DONT CARES - 1

$$W = \sum(5, 6, 7, 8) + \phi(10, 11, 12, 13, 14, 15)$$

wz	00	01	11	10
yz	00	01	11	10
AB	00	01	11	10
	0	0	1	0
	X	X	1	X
	0	1	1	0
	0	1	1	0

Now, SOP =  $w'z'x' w'y + xy + z'w'x$

$$\boxed{\text{SOP} = w' + xy + z'w'}$$

#### 15. EXAMPLES ON DONT CARE SET - 2

AB	00	01	11	10	CD
yz	00	01	11	10	
wz	00	01	11	10	
	0	0	1	0	
	X	X	1	X	
	0	1	1	0	
	0	1	1	0	

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

$\downarrow$   
BD.

yz	00	01	11	10	wz
AB	00	01	11	10	
wz	00	01	11	10	
	0	X	0	X	
	X	1	X	1	
	0	X	1	0	
	0	1	X	0	

$$= \bar{y}z + xy.$$

yz	00	01	11	10
wz	00	01	11	10
AB	00	01	11	10
	0	1	1	0
	X	0	0	1
	X	0	0	1
	0	1	1	X

$$x\bar{z} + \bar{x}z$$

#### 16. EXAMPLES ON DONT CARE SET 3

yz	00	01	11	10	
wz	00	01	11	10	
AB	00	01	11	10	
	0	X	0	0	
	0	X	1	1	
	1	1	1	1	
	0	X	0	0	

$$\Rightarrow A(w+x)y$$

$$\Rightarrow B xy + yw$$

$$\Rightarrow C (w+x)(\bar{w}+y)(\bar{x}+y)$$

None

$$(\bar{x}+y)(x+y)(\bar{x}+w)$$

#### 17. FINDING MINIMAL EXPRESSIONS

AB	00	01	11	10	CD
yz	00	01	11	10	
wz	00	01	11	10	
	1	0	4	11	
	1	1	5	7	
	3	4	9	10	
	2	6	14	10	
	1				

How many minimal expressions are possible?

*	A	1	1	1	1	1	1	1	1	1	1
*	B										
c		1	1								
D	1	1			1	1	1	1	1	1	

$\therefore$  No. of minimal expressions =  $9 \times 2 = 4$

18. BR

A = w  
B = w  
C = w  
D = xy  
E = w  
F = w  
G = w  
H = x

$\Rightarrow$  No

$\Rightarrow$  No

$\Rightarrow$  No

$\Rightarrow$  (

II

$\Rightarrow$  .

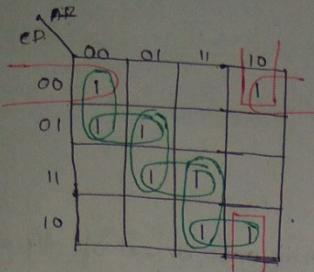
(2)

z	xy	00
		1

$\Rightarrow$

## 18 BRANCHING TECHNIQUE FOR MINIMISING CYCLIC FUNCTIONS

(2)



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$A = \bar{w}x'y'$	x	x														
$B = \bar{w}y'z'$			x													
$C = \bar{w}xz$			x	x												
$D = xyz$				x												x
$E = wxy$											x					x
$F = wyz$									x	x						
$G = wzx$				x	x											
$H = x'y'z'$	x			x												

COVERING A FUNCTION

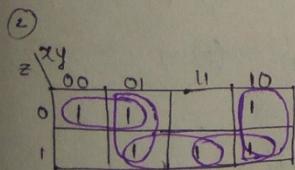
⇒ Now, if I choose 'A' and include it in minimal expression then  $A'$  (included in ME)

⇒ Now, 0,1 minterms are covered by 'A' so delete 'A' row and 0,1 columns

⇒ Now, we can observe  $B \rightarrow C$  ( $B$  is covered by  $C \Rightarrow B$  is irredundant) and  $G \rightarrow H$

⇒ ( $G$  is <sup>covering</sup> covered by  $H \Rightarrow H$  is irredundant) so delete, B, H rows and continue the procedure and mark the essential prime implicants.

⇒ ∴ The minimal ans will be  $A + C + G + E$



	0	1	2	3	4	5	6	7
$A = \bar{w}z$	x	x						
$B = \bar{x}y$			x	x				
$C = yz$				x				x
$D = xz$					x	x		
$E = xy$					x	x		
$F = y'z'$	x				x			

⇒ Let me choose  $A + C + E$

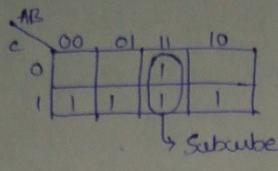
$\downarrow$

$B \rightarrow C \Rightarrow B$  waste       $F \rightarrow E \Rightarrow F$  is not needed.

### 19. IMPLICANT AND PRIME IMPLICANT DIFFERENCE

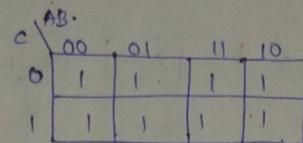
Which of the following can be a prime implicant of a function  $f(ab|bc)$

- a)  $ab+c$
- b)  $bc+a$
- c)  $ac+b$
- d)  $ab$ .



② Which of the following cannot represent prime Implicants. onset of  $f(A|B|c)$

- a)  $\{ab, bc, ca\}$
- b)  $\{abc, ba, bc\}$
- c)  $\{a, b\bar{c}\}$
- d)  $\{ab, ab\}$ .  
 $\cancel{ba}, \cancel{ba}$



$\{ab, bc, ca\}$   
↓  
If  $ab$  is PI then  
anything containing  $ab$   
should not be present

### 20. CONVERTING A FUNCTION INTO SELF DUAL

What minterms have to be added to make the following function a self dual

$$f(A|B|C|D) = A'B'C + (AC+B)D.$$

$$= A'B'C + ACD + BD. \rightarrow \text{Now the Mutually exclusive terms for this are}$$

$A'BCD + A'CD' + B'D'$  these minterms should not be added.

$$\begin{aligned} f(A|B|C|D) &= A'B'C(D+D') + ACD(B+B') + BD(A+A') * (C+C') \\ &= A'BCD + A'BCD' + ABCD + ABCD' + [ABD + A'BD](C+C') \\ &= A'BCD + A'BCD' + ABCD + ABCD' + A'BCD + A'BCD' + ABC'D + ABC'D' \\ &= A'BCD + \cancel{A'BCD'} + ABCD + ABCD' + \cancel{ABC'D} + \cancel{ABC'D'} \\ &= 7 \quad 6 \quad 15 \quad 11 \quad 5 \quad 13 \\ &\quad (7,8) \quad (6,9) \quad (0,15) \quad (4,11) \quad (5,10) \quad (2,13) \end{aligned}$$

(0,15) (1,14) (2,13) (4,11)  
(5,10) (6,9) (7,8) (3,12)

The left out pairs are: (1,14) (3,12)

↓ 1 minterm from  
① minterm from (1,14) ↓ + this  
= 2 choices 2 choices

= 4 minterms can be added

### 21. COMB

How many

$$f_1(C|B|C) =$$

$$f_2(A|B|C) =$$

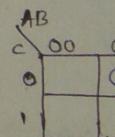
Ans

A	B	C	$f_1$
000	0	0	1
001	0	1	0
010	1	0	1
011	1	1	0
100	0	1	1
101	1	0	0
110	1	1	0
111	1	1	0

### 22. PRIM

What are

prime s



A function

### 23. NUMBER

In a K-

Covering all  
by 3- nor



## COMBINING FUNCTIONS HAVING DONT CARE

How many functions does  $f_1 \cdot f_2$  and  $f_1 + f_2$  represent?

$$f_1(a|bc) = \sum(0, 2, 4) + \phi(3, 5, \#) = \sum(0, 2, 4) + \sum_{\phi}(3, 5, \#).$$

$$f_2(a,b,c) = \sum (2,3) + \sum_{\phi} (1,6,7)$$

ADS

D.	A	f <sub>1</sub>	f <sub>2</sub>	f <sub>1</sub> ·f <sub>2</sub>	f <sub>1</sub> +f <sub>2</sub>
000 = 0	1	0	0	0	1
001 = 1	0	0	0	0	0
010 = 2	1	1	1	1	1
011 = 3	0	1	0	1	1
100 = 4	1	0	0	0	1
101 = 5	0	0	0	0	0
110 = 6	0	0	0	0	0
111 = 7	0	0	0	0	0

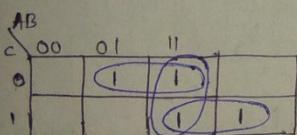
2 Dont cares ( $\emptyset$ ,  $\emptyset$ )

$\Rightarrow$  can take 2 values  $(0,1)$  can take two values  $(0,1)$

$$= 2 \times 2 \\ = 4 \text{ functions}$$

## 22. PRIME IMPLIANTS AND DON'T CARES.

What are the no. of prime Implicants, Essential prime Implicants and Redundant prime Implicants for the function  $f(A,B,C) = \Sigma(2,5,6,7)$ .

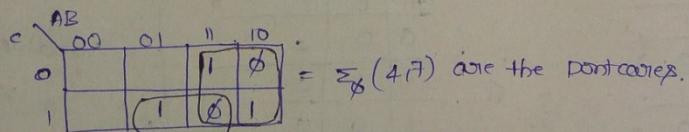


No of Implicants = 3.

Essential = 2

Redundant = 1

A function  $f(A|B,C) = \Sigma(3,5,6)$  is minimised to  $(A+B\bar{C})$  then what are the don't cares?



## 23. NUMBER OF MINIMAL EXPRESSIONS

In a K-map it was found out that all the essential prime implicants are covering all terms except 2 min terms. Those 2 minterms are in turn covered by 3- non-essential prime implicants each. what is the no. of minimal SOP expressions?

The soln is EPI +  $\downarrow$  +  $\overline{\square}$   
 ↓  
 3 choices       $\overline{\square}$  2 choices

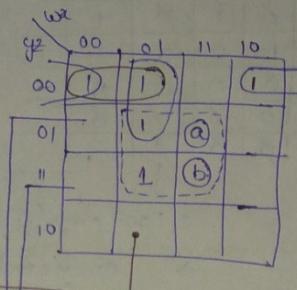
∴ 9 minimal expressions are possible

#### 24. BEAUTIFUL QUESTION ON PRIME IMPLICANT CHART

→ Show prime implicant chart representing boolean expression  $f(wxyz)$   
 columns represent minterms and rows represent PIs. Identify P,Q,R,S  
 and a,b?

	0	4	5	7	8	a	b
$\bar{w}\bar{y}\bar{z} = P$	✓	✓					
$\bar{x}\bar{y}\bar{z} = Q$				✓			
$\bar{w}\bar{x}\bar{y} = R$		✓	✓				
$xz = S$			✓		✓	✓	✓

$$\therefore a=13, b=15$$



→ a,b cannot be in this position because, if they were present P would not become a prime implicant

→ a/b cannot be present here because, if they are present then they should cover the minterm ④ but if we check it does not cover the minterm 4.

#### 25. VARIABLE ENTRANT MAP

$$f(A,B,C) = \sum(1,3,5,7)$$

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

So for the given values of A/B the function 'f' behaves exactly as 'c'

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

= Variable Entrant map.

$$\begin{aligned} A=0, B=0 &\Rightarrow f=c \\ A=0, B=1 &\Rightarrow f=c' \\ A=1, B=0 &\Rightarrow f=c' \\ A=1, B=1 &\Rightarrow f=c \end{aligned}$$

Similarly for  $f = \sum(1,2,5,6)$  = behaves as  $(c, c', c, c')$

$$f = \sum(1,2,3,5,6) = \text{behaves as } (c, c', c, c')$$

$\begin{array}{c|cc}
\downarrow & A=0 & A=1 \\
\hline
B=0 & P & S \\
B=1 & R & Q
\end{array}$

$\begin{array}{c|cc}
\downarrow & A=0 & A=1 \\
\hline
B=0 & P & S \\
B=1 & R & Q
\end{array}$

#### 26. MINIMISATION

1) Set all the

2) (a) Make one

minterms (1)

(b) Multiply

3) Repeat the

4) Sop of VE

Ex:

w	AB	00	01	1
y	00	00	01	1
z	00	D	1	b
	01			
	1			

Step-1: Replace

w	AB	00	01	1
y	00	00	01	1
z	00	0	1	0
	01			
	1			

Step-2: Replace

minterms (ce)

w	AB	00	01	1
y	00	0	0	0
z	00	1	0	0
	01			
	1			

Step-3: Now

and 1's

Step-4:

F = A'B
---------

#### 27. EXAMPLE

$$f(A,B,C) = \sum$$

A	B	C	f
0	0	1	P
0	1	0	S
1	0	0	R
1	0	1	Q

Q1

## 26. MINIMISATION USING VEM

25

- 1> Set all the variables in the cell as '0' and obtain SOP expression
- 2> (a) Make one variable in the cell as '1' and obtain SOP by making minterms (1's) as dont cares.
- (b) Multiply the above 'SOP' with the concerned variable
- 3> Repeat the step(2) until all the variables in the cell are covered
- 4> SOP of VEM is obtained by ORing the previous SOP expressions.

Ex:

AB		CD	
00		01	11
C	0	D   1   0   1   0	
	1	D   1   0   Ø   Ø	

→ If a K-map is used then it contains  $4 \times 4 = 16$  entries

→ It is 4-variable map and hence the function behaves as one of the variable for a given set of other variables. Here 'f' behaves as  $D/D'$  for given set of inputs ABCD.

Step-1: Replace the variables with 0's and obtain SOP expression.

AB		CD	
00		01	11
C	0	0   1   0   0	
	1	0   1   0   Ø	

$$\text{SOP} = A'B.$$

Step-2: Replace any one of the variables D and  $D'$  with 1's and replace the minterms (cells having ones) with Dontcares ( $\emptyset$ );  $D$  replaced.

AB		CD	
00		01	11
C	0	1   Ø   0   0	
	1	1   Ø   0   Ø	

$$\text{SOP} = \bar{A}I.$$

Step-3: Now you have replaced 'D' in step 2(a) now replace  $D'$  with 1's and and 1's should be made as Dontcares and remaining cells as '0's.

D replaced

AB		CD	
00		01	11
C	0	0   Ø   1   1	
	1	0   Ø   0   Ø	

$$= AC'$$

Step-3:  $A'B + D\bar{A} + AC'D'$

$$F = A'B + A'D + AC'D'$$

## 27. EXAMPLE ON VEM

$f(A,B,C) = \sum(3,5,6,7)$  is realised by following VEM, then find P.Q.R.S.

P		Q	
0	0	1	1
R	0	P   S	R   Q
	1		

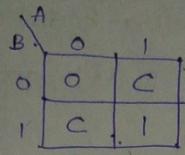
- a)  $P=0, R=S=C, Q=1$   
 b)  $P=Q=0, R=C, S=1$   
 c)  $P=0, Q=R=C, S=1$

Given  $f(A+B+C) = \sum(3, 5, 6, 7)$

Now,

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

Now,



$$P=0$$

$$R=C$$

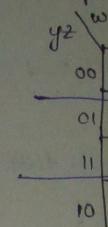
$$S=C$$

$$Q=1$$

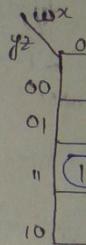
## 29. FIND

How many

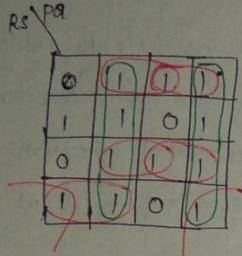
K-map?



## 30. RELA



## 28. PROBLEM ON K-MAP



10 - prime implicants.

12 - Minterms.

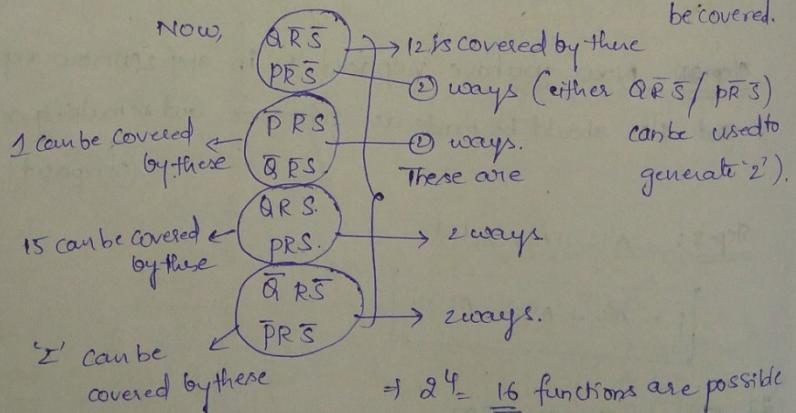
How many minimal functions are possible?

	1	2	4	5	6	7	8	9	10	11	12	15
$\bar{P}Q$			✓	✓	✓	✓						
$P\bar{Q}$							✓	✓	✓	✓		
$Q\bar{R}\bar{S}$				✓								
$P\bar{R}\bar{S}$								✓				
$\bar{P}RS$	✓					✓						
$QPS$	✓											
$QRS$								✓				
$PRS$							✓					
$Q\bar{R}\bar{S}$				✓					✓			
$\bar{P}RS$	✓									✓		

Now, if we include  $\bar{P}Q$  then (4, 5, 6, 7) will be covered

if  $\rightarrow \bar{P}Q$  included then (8, 9, 10, 11) will be covered.

NOW,



$\Rightarrow 2^4 = 16$  functions are possible

## 32. COMP

Minimal

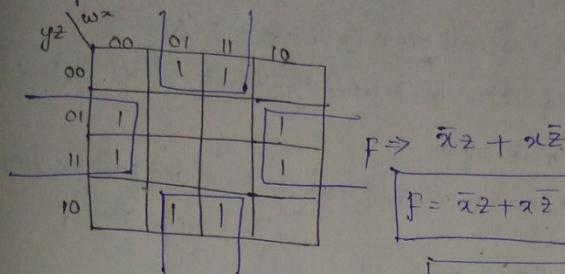
a) F =  $\bar{w}x$

b) F =  $w\bar{x}$

### 29. FINDING FREE VARIABLES

How many variables are free in the expression denoted by the following K-map?

K-map?

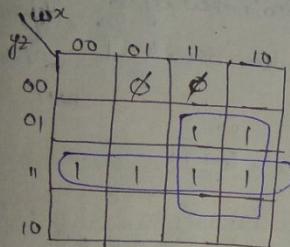


$$f \Rightarrow \bar{x}z + x\bar{z}$$

$f = \bar{x}z + x\bar{z}$  SOP  $\therefore (w,y)$  are not required.

$$\text{POS} = (x+z)(\bar{x}+z)$$

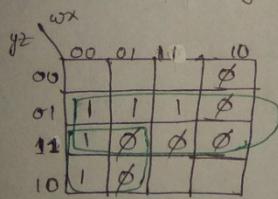
### 30. RELATIONSHIP IN BETWEEN MINIMAL POS, SOP IN CASE OF DONT CARES - 1



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

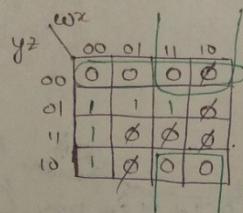
### 31. RELATIONSHIP IN BETWEEN MINIMAL POS, SOP IN CASE OF DONT CARES - 2

Let  $t = (w,x,y,z) = \sum(1,2,3,5,13) + \sum_{\phi}(6,7,8,9,11,15)$  and let  $f(w,x,y,z)$  be minimal sop and  $g(w,x,y,z)$  be the minimal pos. Are  $f$  and  $g$  identical.



$$F = z + yw$$

$$F = \sum(1,2,3,5,6,7,9,11,13,15)$$



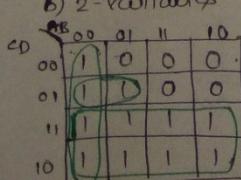
$$g = \sum(1,2,3,5,13,19)$$

There are 2<sup>6</sup> functions are possible (6 dont care and each dont care has 2 choices i.e. 2 functions are possible).

### 32. COMPARING INDEPENDENT VARIABLES IN MINIMAL SOP AND POS.

Minimal expression represented by K-map is free form.

- a) 1-variable    b) 2-variables    c) 3 variables    d) Dependent on all.



$$c + AB' + A'C'D.$$

$\therefore$  Dependent on all.

### 33. NO. OF IRREDUNDANT AND MINIMAL EXPRESSION

28

AB.		CD	00	01	11	10
		00	1	1		
		01	1	1		
		11		1	1	
		10			1	1

1) = No. of prime Implicants = 6 (Both Red + Green)

2) = No. of essential prime Implicants = 2 (Green)

3) = No. of Redundant prime Implicants =  $(6-2) = 4$  (Red)

4) = Minimal sop?

5) How many minimal expressions are possible that are irredundant?

Now, To answer 4,5 construct prime Implicant chart

	0	4	5	13	15	11	10
* P = $\bar{A}C\bar{D}$	✓	X					
Q = $\bar{A}B\bar{C}$		✓	✓				
R = $B\bar{C}D$		✓	✓				
S = $A\bar{B}D$			✓	✓			
T = $A\bar{C}D$				✓	✓		
* V = $A\bar{B}C$					✓	✓	✓

P+V+R+S

$\therefore$  Minimal pos = P+V+R+S.

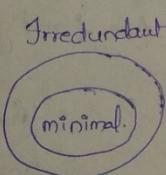
Q is dominated by R  $(Q \rightarrow R)$   
T is dominated by S  $(T \rightarrow S)$

6) No. of Minimal sop's possible are,

5  $\rightarrow$  2 ways (Q+T)

13  $\rightarrow$  2 ways (R+S)

15  $\rightarrow$  2 ways (S+T)



$$\begin{aligned}
 & \therefore (Q+T)(R+S)(S+T) \\
 &= QRS + QRT + QST + QST + \underbrace{RRST}_{RS \text{ is covered in } R} + \underbrace{RRT}_{RS \text{ is covered in } R} + \underbrace{RSS}_{RS \text{ is covered in } S} + RST \\
 &= \underbrace{QRS + QRT + QST}_{QST} + \underbrace{RT}_{RS} + \underbrace{RS}_{RS} + RST \\
 &= QST + RT + RS.
 \end{aligned}$$

$$\begin{aligned}
 & \therefore P+V+Q+S \\
 & \quad P+V+R+T \\
 & \quad P+V+R+S.
 \end{aligned}
 \left. \right\} \text{No. of mE} = \text{No. of Irredundant mE} = 3.$$

(In this case)

$\Rightarrow$  Don't Cares are not included in prime implicant chart.

37. FUNCTIONS INVOLVING FUNCTIONS EXAMPLE 2

Consider a new boolean operation '\$' defined as  $A \$ B = A' + B$ , then find  $f_1 \$ f_2$

(Red)

		AB	
		0 1	
A	0	0	1
	1	1	0

		AB	
		0 1	
B	0	1	0
	1	0	1

e that

$$\text{Now, } f_1 \$ f_2 = f_1' + f_2$$

		AB	
		0 1	
f_1	0	1	1
	1	1	1

$$f_1 = 0 \Rightarrow f_1' = 1 + 0$$

$$f_1 \$ f_2 = 1, 1$$

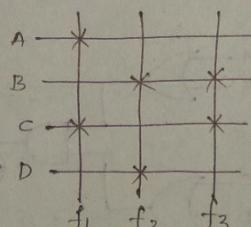
$\begin{cases} Q \rightarrow R \\ \text{waste} \\ T \rightarrow S \end{cases}$

38. FUNCTIONS INVOLVING FUNCTIONS - EXAMPLE - 3

Consider three - three variable functions  $f_1(P, Q, R) = \sum(0, 1, 3)$ ,  $f_2(P, Q, R) = \sum(3, 5, 7)$

$f_3(P, Q, R) = \sum(1, 3, 7)$ . These functions are showing 4 prime implicants  $A'B'C'D'$

(all of the minterm pairs) as shown.

find  $A'B'C'D'$ ?

		PQ	
		00 01	
R	0	0	1
	1	1	0

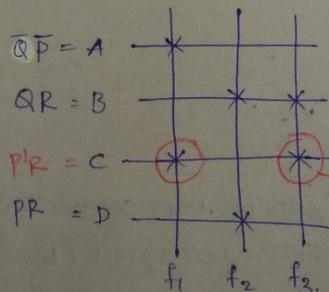
$$f_1 = P'Q + R\bar{P}$$

		PQ	
		00 01	
R	0	1	1
	1	1	0

$$f_2 = QR + PR$$

		PQ	
		00 01	
R	0	1	1
	1	1	0

$$f_3 = P'R + QR$$



common part in  $f_1$  and  $f_3 = P'R$

### 3. DESIGN AND SYNTHESIS OF COMBINATIONAL CIRCUITS

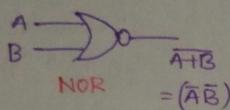
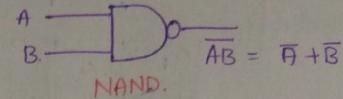
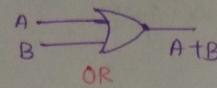
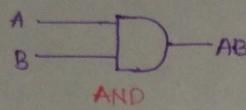
30

NOW,

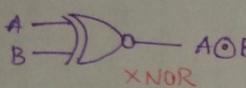
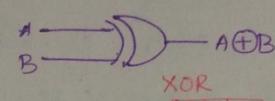
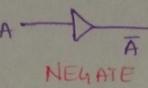
#### 1. INTRODUCTION TO LOGIC DESIGN

→ The main application of switching theory is in the design of digital circuits. (called logic design)

→ The circuits are designed using the basic elements called gates.



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



$$A \oplus B$$

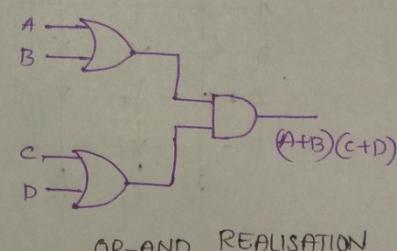
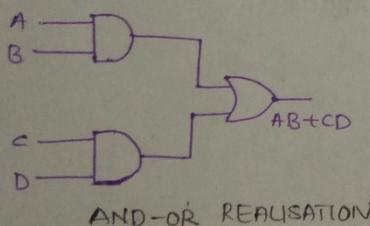
A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

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

#### 2. AND-OR OR-AND REALISATION

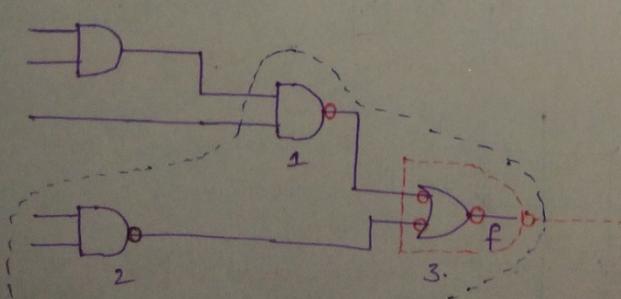
SOP:  $f(A,B,C,D) = AB + CD$ .

f(A,B,C,D) = (A+B)(C+D)



#### 3. MINIMUM NO OF NAND GATES EXAMPLE

Identify min no. of two i/p NAND gates required to represent the following



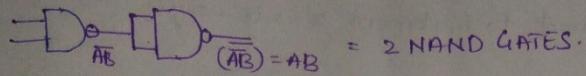
AND-OR REALISATION. TO MAKE IT NAND GATE JUST PLACE BUBBLE AS SHOWN

#### 5. MINI

Find the

$$\text{NOR, } AB = \overline{\overline{AB}}$$

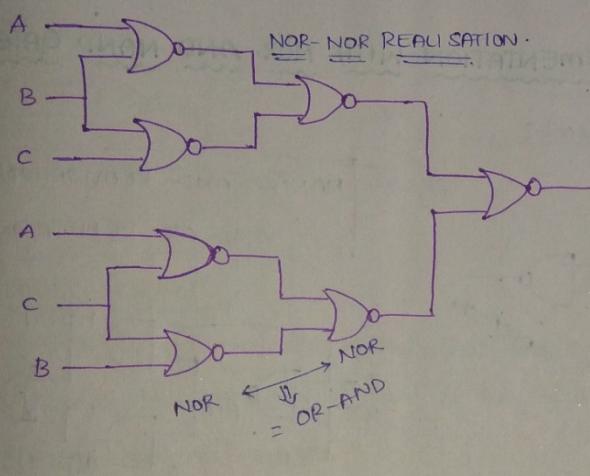
called



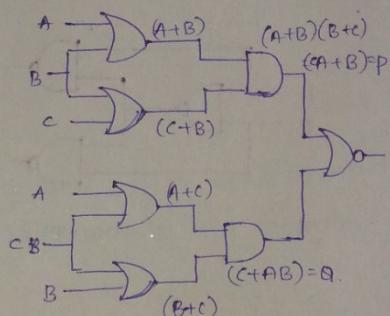
$\therefore$  Total no. of NAND GATES Required = 5.

#### 4. NOR-NOR Example

What does the following function Represent?



NOR-NOR REALISATION =  
OR-AND REALISATION



$\Rightarrow$  Now, output will be  $\overline{PQ}$   
 $= \overline{P} \overline{Q}$

$$\begin{aligned} \therefore \overline{PQ} &= (\overline{B+AC})(\overline{C+AB}) \\ &= \overline{B}(\overline{A+C})(\overline{C})(\overline{A+B}) \end{aligned}$$

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

$$\begin{aligned} \overline{PQ} &= \overline{B}[\overline{A+C}] \overline{C}(\overline{A+B}) = (\overline{A}\overline{C} + \overline{C})(\overline{A}\overline{B} + \overline{B}) \\ &\boxed{\overline{PQ} = \overline{C} + \overline{B}} \end{aligned}$$

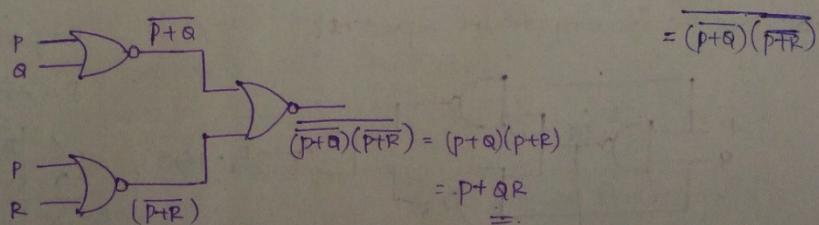
ing

#### 5. MINIMUM NO. OF NOR GATES EXAMPLE

Find the min no. of 2 i/p NOR gates required to represent  $F(P_1Q_1R_1) = P_1 + Q_1R_1$   
 $F(P_1Q_1R_1) = (P_1 + Q_1)(P_1R_1)$

AND

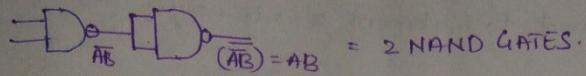
)





$$\text{NOR, } AB = \overline{\overline{AB}}$$

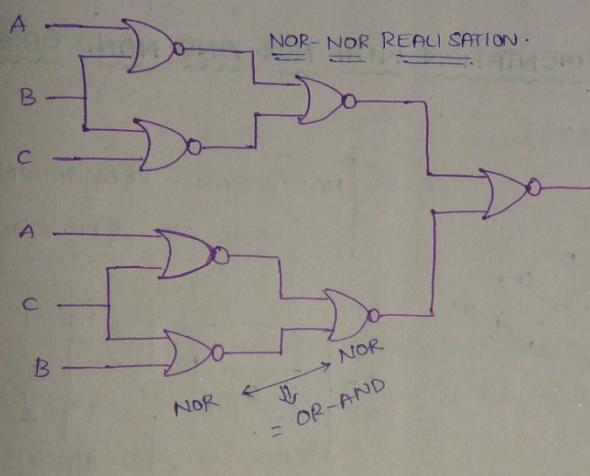
called



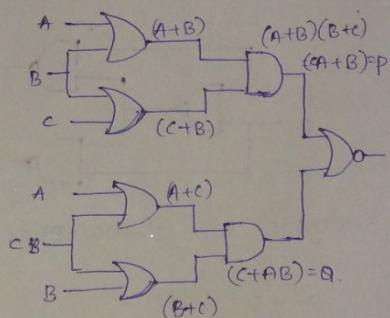
$\therefore$  Total no. of NAND GATES Required = 5.

#### 4. NOR-NOR Example

What does the following function Represent?



NOR-NOR REALISATION =  
OR-AND REALISATION



$\Rightarrow$  Now, output will be  $\overline{PQ}$   
 $= \overline{P} \overline{Q}$

$$\begin{aligned} \therefore \overline{PQ} &= (\overline{B+AC})(\overline{C+AB}) \\ &= \overline{B}(\overline{A+C})(\overline{C})(\overline{A+B}) \end{aligned}$$

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

$$\begin{aligned} \overline{PQ} &= \overline{B}[\overline{A+C}] \overline{C}(\overline{A+B}) = (\overline{A}\overline{C} + \overline{C})(\overline{A}\overline{B} + \overline{B}) \\ &\boxed{\overline{PQ} = \overline{C} + \overline{B}} \end{aligned}$$

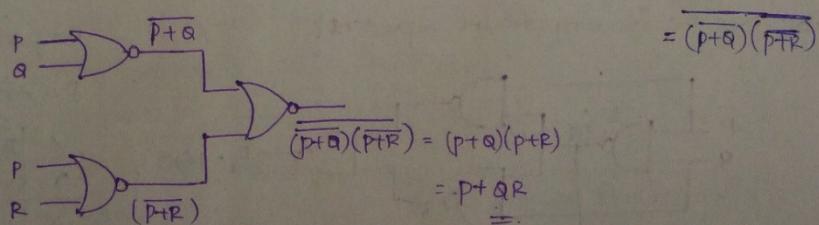
ing

#### 5. MINIMUM NO. OF NOR GATES EXAMPLE

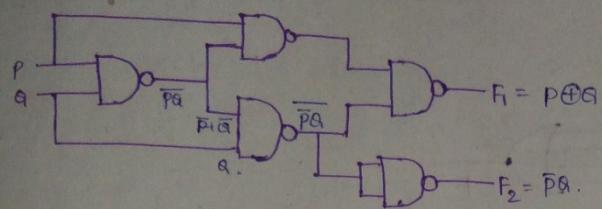
Find the min no. of 2 i/p NOR gates required to represent  $F(P_1Q_1R_1) = P_1 + Q_1R_1$   
 $F(P_1Q_1R_1) = (P_1 + Q_1)(P_1R_1)$

AND

)



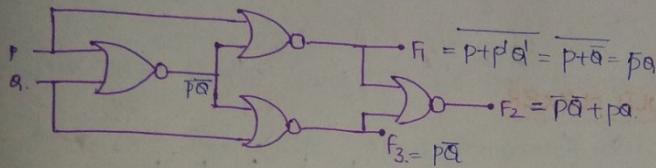
32

9. HALF SUBTRACTOR

33

		Difference	
P	Q	F1	F2
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Borrow.

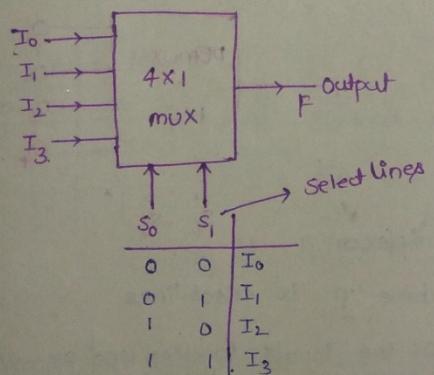
10. COMPARATOR

		P < Q	
P	Q	F2	F1
0	0	1	0
0	1	0	1
1	0	0	1
1	1	1	0

P > Q.

11. MULTIPLEXER

- A Mux is an electronic switch that can connect one out of inputs to output
- It cannot change the logical level of the input, it only provides the connection between input and output.
- It is functionally complete, i.e all boolean func can be realized using only multiplexers without any other gates.



$$F = \underbrace{\bar{S}_0 \bar{S}_1 I_0 + S_0 \bar{S}_1 I_1}_{\text{if } S_0, S_1 \text{ are } 0, 0 \text{ then } \text{ofp will be } I_0} + \underbrace{S_1 \bar{S}_0 I_2 + S_1 S_0 I_3}_{\text{if } S_0, S_1 \text{ are } 0, 1 \text{ then } \text{ofp will be } I_1}$$

$S_0, S_1$  are (1,0) then ofp will be  $I_2$

$S_0, S_1$  are (1,1) then ofp will be  $I_3$

⇒ The select lines are used to pick one among the inputs and show it as output

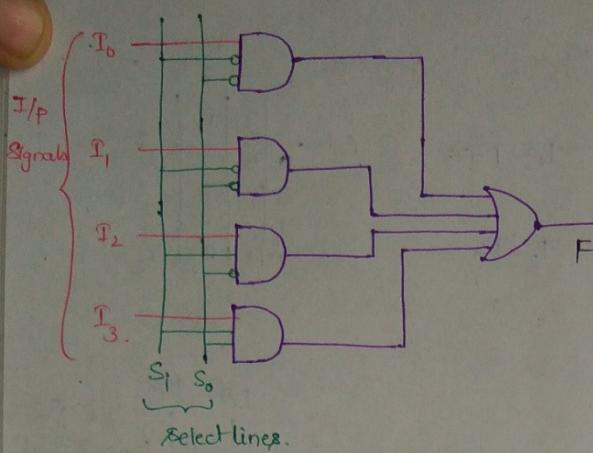
⇒ If we have  $2^D \times 1$  Mux then we have ' $n$ ' select lines, and ' $n$ ' bits are uniquely required to represent each input signal

$$\therefore F = \underbrace{\bar{S}_0 \bar{S}_1 I_0 + S_0 \bar{S}_1 I_1 + S_1 \bar{S}_0 I_2 + S_1 S_0 I_3}_{\text{AND-OR REALISATION}}.$$

AND-OR REALISATION.

To say

Now,



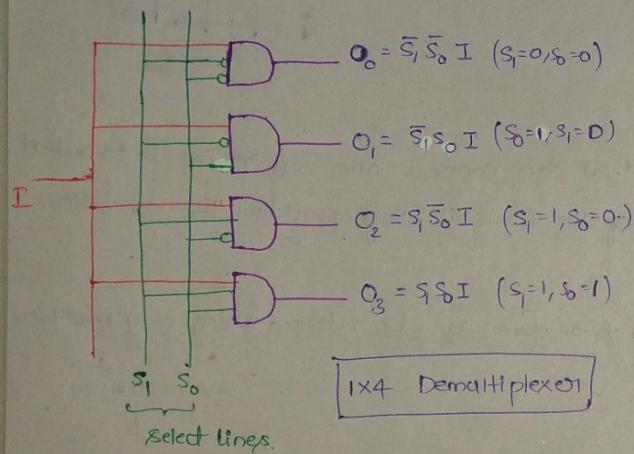
BOTH SELECTED

X	$\bar{X}$
0	1
1	0

Now, +

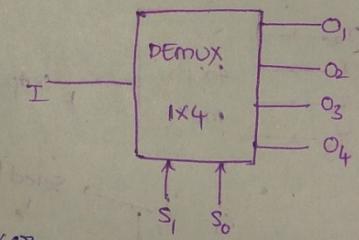
NO

→ Now

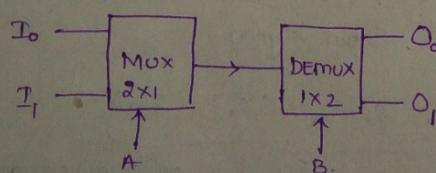
22. INTRODUCTION TO DE MUX

S <sub>1</sub>	S <sub>0</sub>	O <sub>0</sub>	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

⇒ The AND GATE works if all the other inputs (other than that you are giving) are ones)



- It performs opposite operation of Multiplexer
- It has one input and  $2^n$  outputs where ' $n$ ' is select lines
- It is derived by Mux by joining all the inputs together and removing OR GATE
- Mainly used in construction of switches
- DEMUX is used at Receiving end and MUX is used at transmitting end



A	B	
0	0	(I <sub>0</sub> -O <sub>0</sub> )
1	1	(I <sub>1</sub> -O <sub>1</sub> )
0	1	(I <sub>0</sub> -O <sub>1</sub> )
1	0	(I <sub>1</sub> -O <sub>0</sub> )

MSB ↘

Straight connection  
cross connection

T<sub>0</sub> —  
T<sub>1</sub> —  
T<sub>2</sub> —  
T<sub>3</sub> —

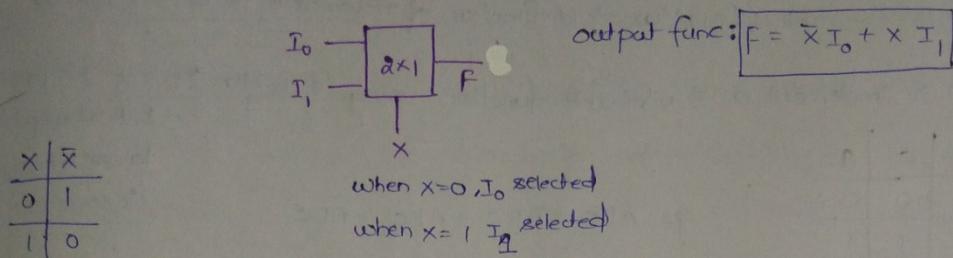
## 13. IMPL



## 12. PROVING MUX IS FUNCTIONALLY COMPLETE

To say functionally complete we should derive  $(+, \cdot, -)$  or  $(+, -)$   $(\cdot, -)$

Now, let us consider 2x1 Multiplexer.



Now, the Select line 'X' can be '0' or 1 then if  $X=0$ , then  $I_0$  will be selected

Now,  $\bar{X} = \bar{X} \cdot 1 + X \cdot 0$ . Now comparing with F

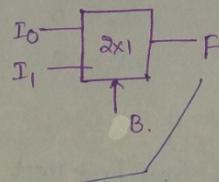
when  $I_0=1, I_1=0$ , then it generates NOT (Behaves in Compliment manner)

⇒ Now,

A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

$$F = AB = B(A) + \bar{B}(0)$$

$$F = \bar{B}I_0 + BI_1, \quad \begin{matrix} I_0=0, I_1=A \text{ then AND can be realised} \\ \downarrow \end{matrix}$$



⇒ Now,

A	B	$A+B$
0	0	0
0	1	1
1	0	1
1	1	1

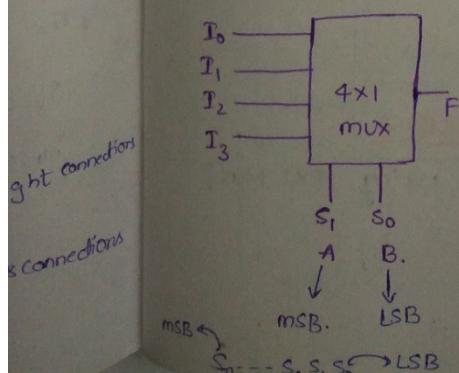
$$A+B = \bar{A}\bar{B} + A\bar{B} + \bar{A}B + AB$$

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

$$F = \bar{B}I_0 + BI_1 \quad \therefore \text{OR can be Realised.}$$

OR GATE

## 13. IMPLEMENTING FUNCTIONS WITH MUX EXAMPLE 1



$$F = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + A\bar{B}I_2 + ABI_3$$

Now, I want to Implement function 'g' say

A	B	g
0	0	1
0	1	0
1	0	0
1	1	1

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

$$g(A,B) = \bar{A}\bar{B}(1) + \bar{A}B(0) + A\bar{B}(0) + AB(1)$$

$\therefore$  Mux can implement 'g'

Now, using  $4 \times 1$  multiplexer i can implement a function of  $n$  variables

$\Rightarrow (2^2 \times 1)$  MUX  $\rightarrow$  2 variable fun can be implemented

$\Rightarrow (2^3 \times 1)$  MUX  $\rightarrow$  3 variable fun can be implemented

$\Rightarrow (2^n \times 1)$  MUX  $\rightarrow$  A function of  $n$ -variable can be implemented

Now, can i implement a 3 variable function using  $(4 \times 1)$  MUX ?? Yes it is possible but we might have to use more gates (in some cases)

A	B	C	g
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

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

$$g = \bar{A}\bar{B}(C) + \bar{A}B(C+\bar{C}) + A\bar{B}(0) + AB(C)$$

Now,  $(4 \times 1)$  MUX equation  $f = \bar{A}BI_0 + \bar{A}BI_1 + A\bar{B}I_2 + AB\bar{I}_3$

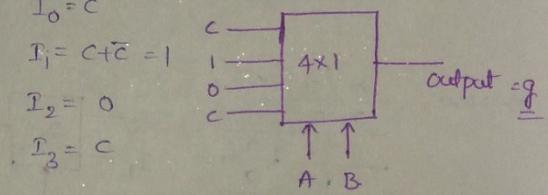
comparing f/g

$$I_0 = C$$

$$I_1 = C + \bar{C} = 1$$

$$I_2 = 0$$

$$I_3 = C$$



∴ Now, just by using  $(2^n \times 1)$  MUX we may implement a function having more than  $n$  variables. but i may require some more gates.

$\Rightarrow$  Now, we cannot implement a function of 4 variables using  $(4 \times 1)$  MUX because out of the 4 variables 2 variables must be given to select lines and the remaining two variables are not independent they are function of two variables and we require more gates.

The function that is formed with 4 variables is

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

↓      ↓      ↓      ↓  
Selecting      function requires AND GATE

#### 14. IMPLEMENTING FUNCTIONS WITH MUX - EXAMPLE - 2

Implement the following function  $f(A_1B_1C) = \sum(1, 2, 4, 6, 7)$  using  $4 \times 1$  Mux?

36

$$f(A, B, C) = \sum(1, 2, 4, 6, 7) \rightarrow \text{for } 4 \times 1 \text{ Mux the characteristic Equation is}$$

37

$$G_1 = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + A\bar{B}I_2 + AB\bar{I}_3.$$

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

anted

it is possible  
we might have  
more gates  
(some cases)

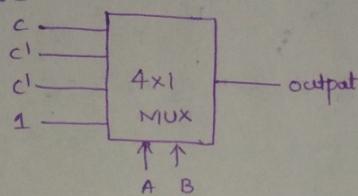
$$I_2 + AB\bar{I}_3.$$

$$\Rightarrow f = A'B'C + A'BC' + ABC' + ABC + ABC \\ = A'B'(C) + AB(C') + AB'(C') + ABC(C+C')$$

$$f = A'B'(C) + A'BC(C') + AB'C(C') + AB.$$

$$\text{Comparing with } G_1 \quad I_0 = C \quad I_2 = C' \\ I_1 = C' \quad I_3 = 1$$

The MUX will be



$$A=0, B=0 \Rightarrow Q_P = C$$

$$A=0, B=1 \Rightarrow Q_P = C'$$

$$A=1, B=0 \Rightarrow Q_P = C'$$

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

$$\text{Output} = g$$

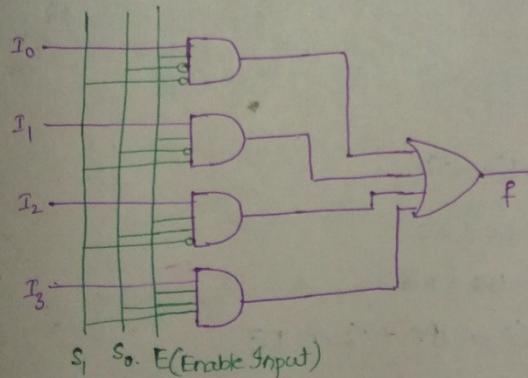
### 15. IMPLEMENTING WITH MUX - EXAMPLE - 2

$$f(A, B, C) = \sum$$

### 15. MULTIPLEXER USING EXCLUSIVE INPUT

more than

MUX because  
and the  
variables



$$f = E(I_0 \bar{s}_0 \bar{s}_1 + I_1 s_0 \bar{s}_1 + I_2 \bar{s}_1 \bar{s}_0 + I_3 \bar{s}_0 s_1) \\ + I_2 \bar{s}_1 s_0 + I_3 s_1 s_0$$

→ If the value of E=1 then  
mux works and gives Q\_P=1

→ If E=0 → MUX stops and  
gives 0.

Mimise the function represented by following MUX?

The characteristic Equation of 4x1 MUX is

$$F = E(I_0 \bar{s}_0 \bar{s}_1 + I_1 \bar{s}_1 \bar{s}_0 + I_2 s_0 \bar{s}_1 + I_3 s_1 s_0)$$

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

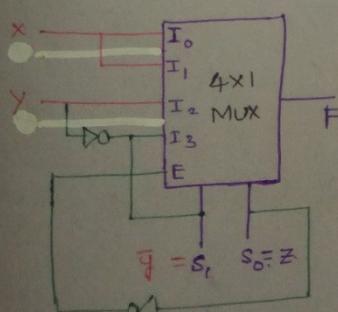
$$= \bar{z}(\underbrace{x\bar{z}y}_{xy\bar{z}} + \underbrace{x\bar{y}z}_{\bar{y}z} + 0 + \bar{y}z)$$

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

$$\therefore F = xy\bar{z}$$

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

MUX?





## 16. RELATION BETWEEN SELECT LINES AND INPUTS OF A MUX

18

$F(A_1B_1C) = \sum(2, 3, 5, 6, 7)$  Implement using 4x1 Mux such that,

a)  $S_1 = B, S_0 = C$

b)  $S_1 = C, S_0 = B$

Sol:  $F = \sum(2, 3, 5, 6, 7) = \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + ABC + A\bar{B}\bar{C}$

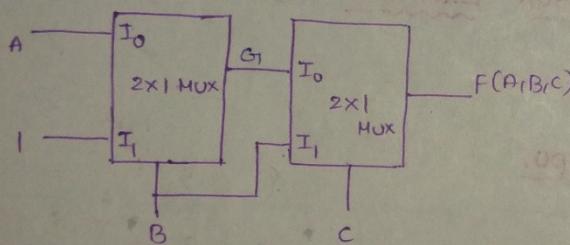
Now, the select lines are  $B, C \Rightarrow (0) \bar{B}\bar{C} + (1)\bar{B}C + (2)B\bar{C} + (3)BC$   
 $\downarrow \quad \downarrow \quad \downarrow \quad \downarrow$   
 $I_0 \quad I_1 \quad I_2 \quad I_3$

Now, if  $S_1 = C$  and  $S_0 = B$  then  $I_0 = 0, I_1 = 1, I_2 = A, I_3 = A$

## 17. CASCADING MUX EXAMPLE - 1

19.

What is the function in canonical SOP?



$$G_1 = \bar{B}A + B(1)$$

$$F = \bar{C}I_0 + C I_1$$

$$G_1 = B + \bar{B}A$$

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

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

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

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

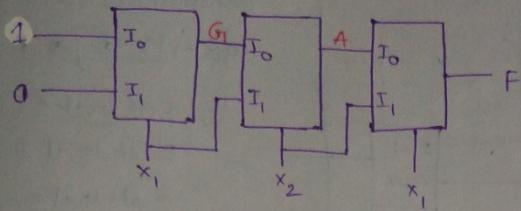
$$= \underline{ABC} + \underline{ABC} + \underline{A'BC} + \underline{ABC} + \underline{A'B\bar{C}}$$

$$= \cancel{ABC} + \cancel{ABC} + \cancel{A'BC} + \cancel{ABC}$$

$$= \cancel{ABC} + \cancel{ABC} + \cancel{A'B}$$

20.

How

18. CASCADING MUX EXAMPLE - 2

$$G_1 = \bar{x}_1(I_0) + x_1(I_1) = \bar{x}_1$$

$$F = \bar{x}_1 A + x_1(I_1)$$

$$A = \bar{x}_2(I_0) + x_2(I_1)$$

$$= \bar{x}_2(x_1 \odot x_2) + x_1(x_2)$$

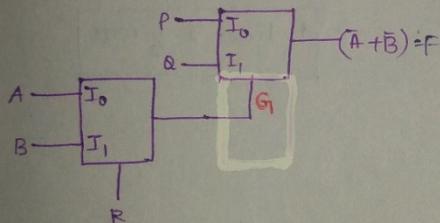
$$= \bar{x}_2(G_1) + x_2(x_1)$$

$$= \bar{x}_2(\bar{x}_1 x_1 + \bar{x}_2 \bar{x}_1) + x_1 x_2$$

$$A = (x_1 \odot x_2)$$

$$= \bar{x}_2 \bar{x}_1 + x_1 x_2$$

$$F = x_1 \odot x_2$$

19. CASCADING MULTIPLEXERS - EXAMPLE - 3

$$G_1 = \bar{R}(I_0) + R(I_1) = \bar{R}(A) + R(B)$$

$$F = \bar{G}_1(P) + G_1(Q)$$

$$\bar{A} + \bar{B} = \overline{[\bar{R}(A) + R(B)]} P + [\bar{R}(A) + R(B)] Q$$

$$\bar{A} + \bar{B} = [(\bar{R} + \bar{A})(\bar{R} + \bar{B})] P + [\bar{R}A + RB] Q.$$

$$\bar{A} + \bar{B} = [\bar{R}\bar{R} + \bar{R}\bar{B} + \bar{A}\bar{R} + \bar{A}\bar{B}] P + [\bar{R}AQ + RBQ]$$

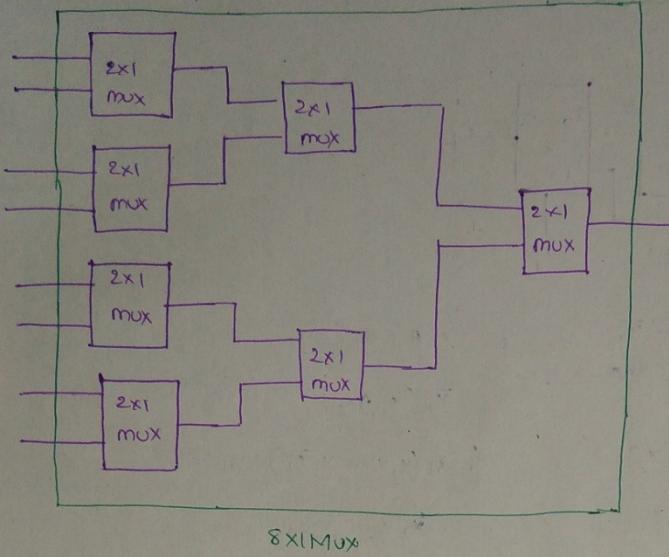
$$\bar{A} + \bar{B} = R\bar{B}P + \bar{A}\bar{R}P + \bar{A}\bar{B}P + \bar{R}AQ + RBQ.$$

= Now do opm.

==

20. EXPANSION OF MUX'S

How can you construct a 8x1 MUX using 2x1 MUX?



Now, it is clear that to construct 8x1 MUX we need 7  $(2 \times 1)$  MUX and at 3 levels  
 .4 at level 1  
 2 at level 2  
 1 at level 3.

$\therefore$  Totally 7 MUX are needed  
 3 levels are formed

Given,  
 $(M \times 1)$   
 m lines

$\therefore$  One  
 $\Rightarrow$  MIP

No.

Now, Initially I have 8 lines to cover and now my  $(2 \times 1)$  MUX will cover two lines

$\Rightarrow$  1 MUX will cover 2 lines

$\therefore$  8 lines are covered by  $8/2$  MUX = 4

$\Rightarrow$  2 lines  $\rightarrow$  1 MUX

Now, 4 lines are covered by  $4/2$  MUX = 2 MUX

$\Rightarrow$  1 line  $\rightarrow$   $1/2$  MUX

Now 2 lines are covered by = 1 MUX

i.e. 8 lines  $\rightarrow$   $8/2$  MUX  $\rightarrow$  4 MUX

4 lines  $\rightarrow$   $4/2$  MUX  $\rightarrow$  2 MUX

2 lines  $\rightarrow$   $2/2$  MUX  $\rightarrow$  1 MUX.

$\Rightarrow$  Now

Ex: Construct 32x1 MUX with 4x1 MUX

Initially 32 lines and my small 4x1 MUX covers 4 lines  $\Rightarrow$  1 MUX  $\rightarrow$  4 lines

$1$  line  $\rightarrow$   $1/4$  MUX

32 lines  $\rightarrow$   $32/4$  MUX = 8

$\therefore$  8 MUX are needed to cover 32x1 MUX

8 lines  $\rightarrow$   $8/4$  MUX = 2

2 lines  $\rightarrow$   $2/4$  MUX =  $1/2$  = 1

11 MUX

Ex: No

Devices

$(8 \times 1)$   
 $\downarrow$   
 m

Ex:  $(32 \times 1)$   
 $\downarrow$   
 m

$\Rightarrow$  levels

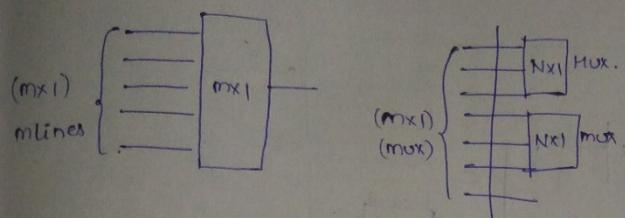
So, when you are trying to manufacture a  $(M \times 1)$  MUX using  $(N \times 1)$  MUX then how many levels are required and how many MUX are required.

that to

(40) Given,  $(M \times 1)$  MUX should be covered by  $(N \times 1)$  MUX

(41)

we need  
at 3 levels



needed  
needed

$\therefore$  One  $(Nx1)$  MUX is going to cover  $N$  lines  $\Rightarrow$  1 Device covers  $N$  lines

$\Rightarrow$   $M$  lines will be covered by  $M/N$  Devices.

$$1 \text{ line} \rightarrow \frac{1}{N} \text{ Devices}$$

Note,  $M$  lines  $\rightarrow \left(\frac{M}{N}\right)$  devices

$\frac{M}{N}$  lines  $\rightarrow \left(\frac{M}{N^2}\right)$  devices

$\left(\frac{M}{N^2}\right)$  lines  $\rightarrow \frac{M}{N^3}$  devices

(1) we continue this procedure until  $\frac{M}{N^K} \leq 1$

$$\Rightarrow N^K \geq M \Rightarrow K \geq \lceil \log_N M \rceil$$

No of levels

$\Rightarrow$  Now, the No of devices used are  $\frac{M}{N} + \frac{M}{N^2} + \frac{M}{N^3} + \dots + \frac{M}{N^K}$

$$\boxed{\text{Devices} = \sum_{K=1}^{\lceil \log_N M \rceil} \left( \frac{M}{N^K} \right)}$$

$\Rightarrow 4$  lines

$\frac{1}{4}$  MUX

Ex: Now, I want to implement  $8 \times 1$  MUX using  $(2 \times 1)$  MUX, how many levels and devices are needed?

1 MUX

$$(8 \times 1) \rightarrow (2 \times 1) \text{ mux} \quad \Rightarrow \text{No. of levels} = \lceil \log_2 8 \rceil = \lceil 3 \rceil = 3 \text{ levels}$$

$$\Rightarrow \text{No. of devices} = \sum_{K=1}^3 \left( \frac{8}{2^K} \right) = 8/2 + 8/4 + 8/8$$

then

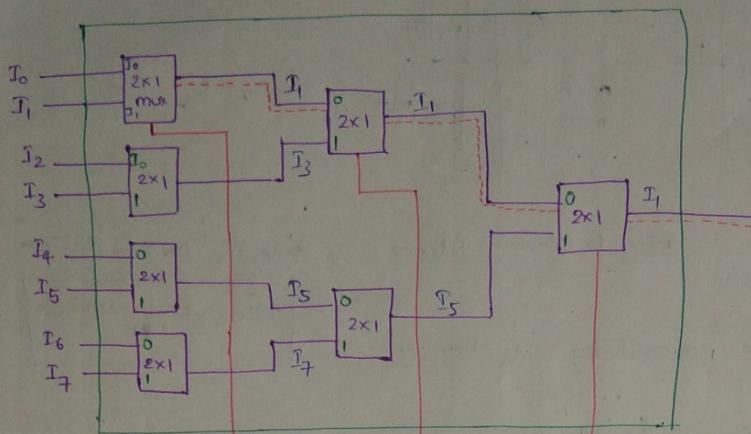
$$\underline{\underline{\text{Ex: } (32 \times 1) \text{ MUX} \rightarrow (4 \times 1) \text{ mux.}}} \quad = 4 + 2 + 1 = 7 \text{ mux.}$$

$$\Rightarrow \text{levels} = \lceil \log_4 32 \rceil = \lceil \log_4 32 \rceil = \lceil 5/2 \rceil = \lceil 2.5 \rceil = 3,$$

$$\text{Devices} = \frac{32}{4^1} + \frac{32}{4^2} + \frac{32}{4^3} = 32/4 + 32/16 + 32/64 = 8 + 2 + \frac{1}{2} = 11 \text{ mux.}$$

## ASSIGNING SELECT LINES WHILE EXPANDING THE MUX.

(42)

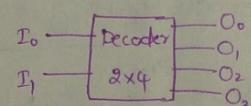
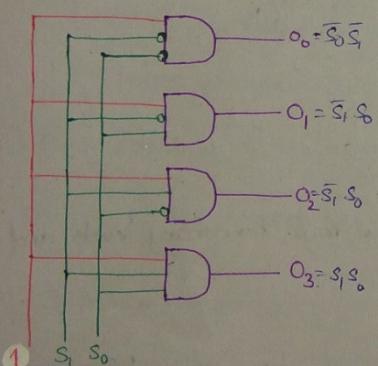


$$S_0 = 1, S_1 = 0, S_2 = 0$$

The inputs that  
get selected at level one are  $(I_1, I_3, I_5, I_7)$

$\Rightarrow$  This assignment of select lines  
 $S_2 = 0$  is wrong because if we give  
 $S_2, S_1, S_0 = 001$ , then the o/p should  
be  $I_1$ , but if  $S_2 = 0$  then  
it will select ' $I_0$ ' not  $I_1$ .

## 23. INTRODUCTION TO DECODER



$$\text{MUX} = 2^n \times 1$$

$$\text{DEMUX} = 1 \times 2^n$$

$$\text{DECODER} = n \times 2^n$$

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

$$O_0 = S_1 S_0$$

$$O_1 = \overline{S}_1 S_0$$

$$O_2 = S_1 \overline{S}_0$$

$$O_3 = \overline{S}_1 \overline{S}_0$$

→ A demux can be converted to a decoder by always setting  $I=1$  and making select lines as inputs.

→ since a decoder provides all the minterms, we could implement a func in canonical SOP using "OR" Gate.

→ If all the 'AND' gates are replaced with NAND gates, the decoder becomes active low.

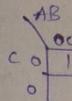
→ Implementing func with decoder is AND-OR REALISATION / NAND-NAND REALISATION.

$$\overline{D_1 D_2} = \overline{D_1} \overline{D_2}$$

## 24. IMPLEMENTATION

A  
B  
C

$$F = \sum (0, 1, 2, 3)$$



## 25. IMPLEMENTATION

A  
B  
C

Now, A  
B  
C

Now, the above

$$\begin{aligned} \Sigma &= (0, 1, 2, 3) \\ &= \overline{A} \overline{B} \overline{C} + \overline{A} \overline{B} C + \overline{A} B \overline{C} + A \overline{B} \overline{C} \end{aligned}$$

## 26. DECODER

Same Question

⇒ Now, when a

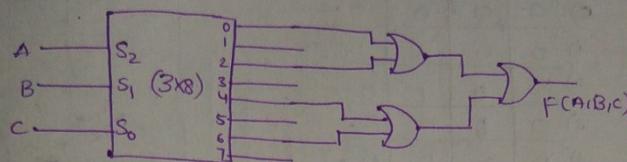
$$O_0 = \overline{(ABC)}$$

(42)

## 24. IMPLEMENTING FUNCTIONS WITH EXAMPLE - 1

(43)

DECODER



$$F = \sum(0, 2, 4, 6)$$

		AB	00	01	11	10
		C	0	1	1	1
S0	S1 (3x8)	0	1	1	1	1
		1	1	1	1	1

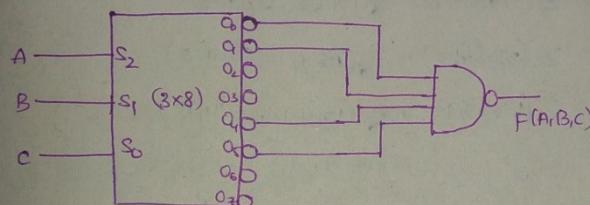
F is free from

- a) 1 variable    b) 3 variables  
b) 2 variables    d) None

 $\therefore [F = c^1] \Rightarrow$  Independent of 2 variables.

If select lines  
if we give  
q/p should  
= 0 then  
'I<sub>o</sub>' not I<sub>1</sub>

## 25. IMPLEMENTING FUNCTIONS WITH DECODER EXAMPLE - 2



F is free from.

- a) 1 variable    b) 2 variables  
c) 3 variables    d) None.

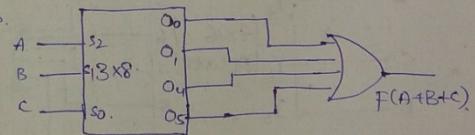
Now,  $A \quad B \quad C$      $\overline{ABC} = \overline{A} + \overline{B} + \overline{C}$     ( $\cong$ )     $A \quad B \quad C$      $\overline{A} + \overline{B} + \overline{C}$

Now, the above diagram can be modified as.

$$\begin{aligned} O_3 & \\ 0 & O_0 = \overline{S_2} \overline{S_0} \\ 0 & O_1 = \overline{S_1} S_0 \\ 0 & O_2 = \overline{S_1} \overline{S_0} \\ 1 & O_3 = S_1 S_0 \end{aligned}$$

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

		AB	00	01	11	10
		C	0	1	1	1
S0	S1 (3x8)	0	1	1	1	1
		1	1	1	1	1

 $\therefore [F = \overline{B}] \Rightarrow$  Independent of two variables.

## 26. DECODER FOR FUNCTION IMPLEMENTATION - Ex-3

Same Question above but the NAND gate is replaced by AND Gate.

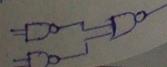
⇒ Now, when a minterm is bubbled (complemented) then it becomes maxterm.

$$O_0 = \overline{(\overline{a} \overline{b} \overline{c})} = (a + b + c) \therefore \pi(0, 1, 4, 5) = \sum(2, 3, 6, 7)$$

		AB	00	01	11	10
		C	0	1	1	1
S0	S1 (3x8)	0	1	1	1	1
		1	1	1	1	1

 $\therefore [F = B] \Rightarrow$  Independent of '2' variables

REALISATION:



## 27. CONVERTING ONE CODE TO ANOTHER USING DECODER

44

A	B	C	D
8	4	2	1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	0	1	0
6	0	1	0
7	0	1	1
8	1	0	0
9	1	0	1

w	x	y	z
2	4	2	1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	1	1	0
9	1	1	1

complements of each other (self compliment representation)

In 2421 code the

$$9's \text{ complement of } 5 = 9 - 5 = 4$$

So take 4 &

complement it

$$W = \sum(5, 6, 7, 8) + \phi(10, 11, 12, 13, 14, 15)$$

$$X = \sum(4, 6, 7, 8, 9) + \phi(10, 11, 12, 13, 14, 15)$$

$$Y = \sum(2, 3, 5, 8, 9) + \phi(10, 11, 12, 13, 14, 15)$$

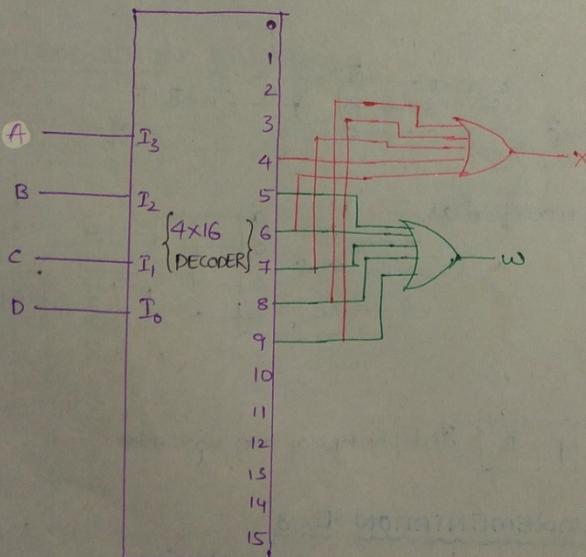
$$Z = \sum(1, 3, 5, 7, 9) + \phi(10, 11, 12, 13, 14, 15)$$

29. 11

→ To

30. 10

Now,



⇒ similarly Y, Z can also be drawn.

Now,

multi

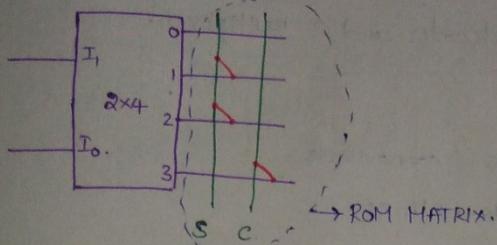
## 28. ROM IMPLEMENTATION USING DECODER

→ ROM can be used to realize combinational functions by storing appropriate values at appropriate locations.

→ Every ROM is expressed in terms of ROM Matrix and decoder

→ ROM matrix contains set of links and connections. The lines entering the matrix and leaving it are called the intersection of rows and columns are called links.

(44)



	A	B	Sum(s)	Carry(c)
0)	0	0	0	0
1)	0	1	1	0
2)	1	0	1	0
3)	1	1	0	1

ts of each  
if compliment  
validity)

code the  
rento)  $S = 9 - S$   
 $= 4$   
state by  $S$   
complement it

### 29. IMPLEMENTING FUNCTIONS USING ONLY DECODER

→ To implement a function of  $m$  variables we need  $(m \times 2^n)$  decoder

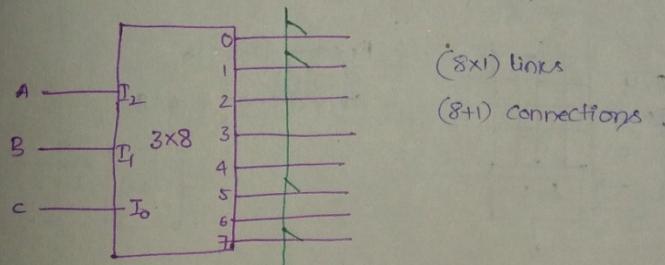
∴ The no. of lines coming/entering into rom matrix is  $2^n$

The no. of connections to implement  $m$ -func =  $(2^n + m)$  connections

" " " links ( $x^n$  points) to implement  $m$ -func =  $(2^n * m)$  links

### 30. IMPLEMENTING FUNCTIONS USING DECODER + MUX EXAMPLE 1

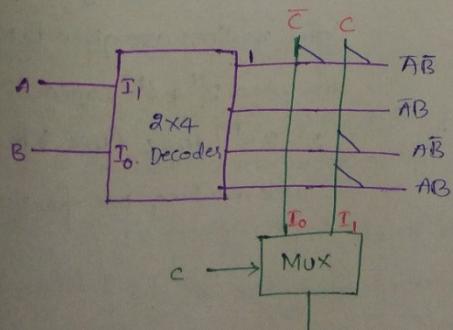
Now, let us consider  $3 \times 8$  decoder



(8+1) connections

$$f(A, B, C) = \sum(0, 1, 5, 7) = \bar{A}B\bar{C} + \bar{A}BC + AB + C + ABC$$

Now, I need to implement the same function using  $2 \times 4$  decoder. Then we need a multiplexer also.



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

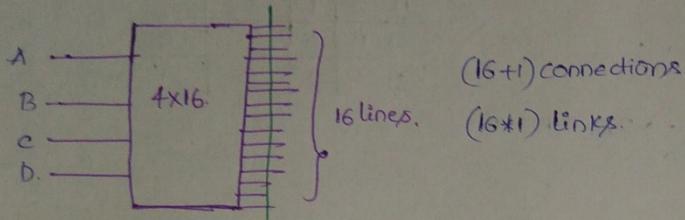
for all this inputs output should  
be one! and when we give  
inputs other than these it should be

appropriate

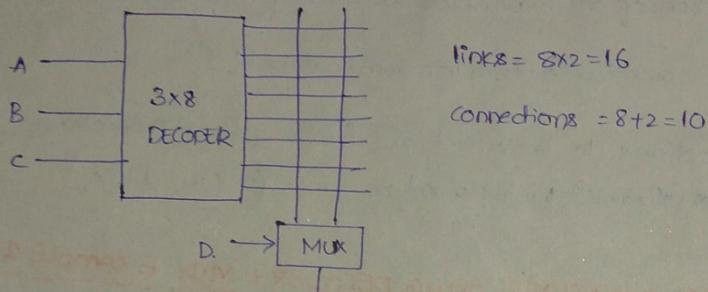
of the  
called links.

### 31. IMPLEMENTING FUNCTIONS USING DECODER + MUX - EX AMPLE-2

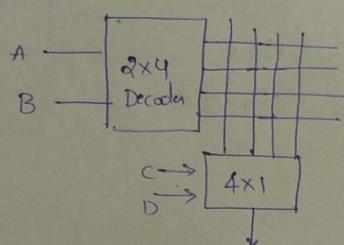
Implement 4x16 Decoder using 3x8 decoder and a multiplexer?



Now, 3x8 Decoder



Now, 2x4 Decoder



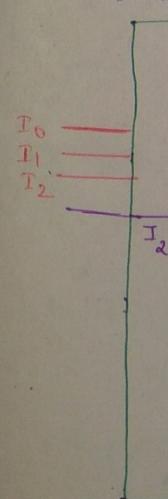
Decoder(n)	Mux	Connections	Links
$10 \times 2^{10}$	0	$1024 + 1$	$1024$
5	5	$2^5 + 2^5 = 64$	$2^5 * 2^5 = 1024$
4	6	$2^4 + 2^6 = 80$	$2^4 * 2^6 = 1024$
3	7	$2^3 + 2^7 = 128$	$2^3 * 2^7 = 1024$

$$\text{links} = 4 \times 4 = 16$$

$$\text{connections} = 4 + 4 = 8$$

Constant

For Deco



Now, if I want to implement a function of  $(n+m)$  variables where  $n$  lines are given to Decoder and  $m$  lines are given to Mux then we need  $2^n + 2^m$  connections and  $2^n * 2^m$  links. (This is true for implementing one function)

⇒ we have

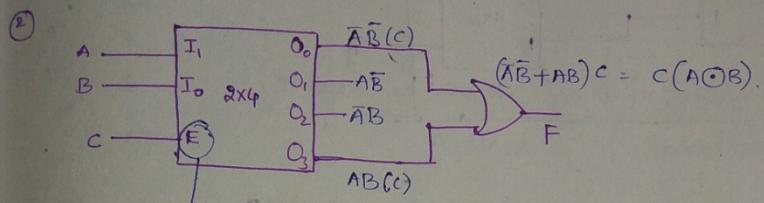
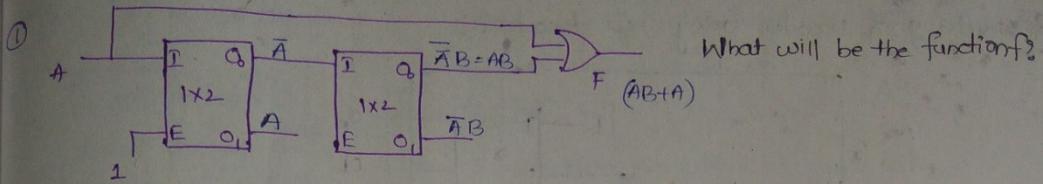
80 at level  
and 4 at level

⇒ Now, let's

82 at level

⇒ Now, 81

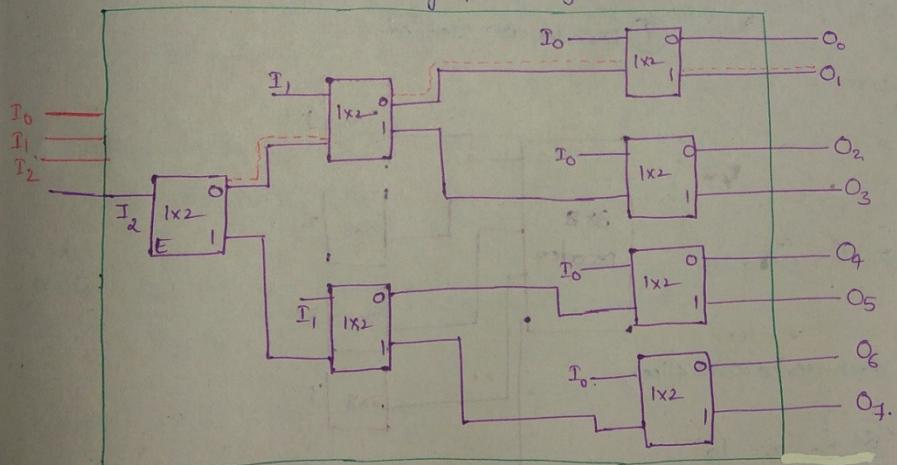
### 32. DECODER WITH ENABLE INPUT



### 33. CONSTRUCTING 3x8 DECODER USING 1x2 DECODER

Construct 3x8 Decoder using 1x2 Decoder.

For Decoders start building from Right side



⇒ we have to cover 8 lines ( $O_0, O_1, \dots, O_7$ ) and each 1x2 Decoder covers 2 lines

so at level 1 we will have 4 decoders now, join all the enables of 4 decoders and 4 enables need two decoders at level 2 and one decoder at level 1.

⇒ Now let me assign input  $I_0$  at level 1 for all decoders and  $I_1$  at level 2 and  $I_2$  at level 3 (I'm not sure about it just assigned randomly to check correctness)

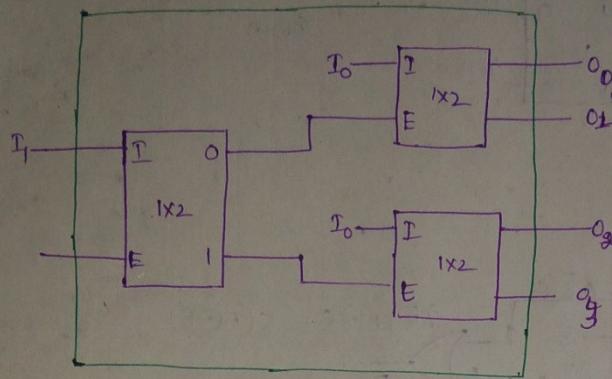
⇒ Now, if it is say  $I_2 I_1 I_0 = (0, 1, 1) \Rightarrow I_2 = 0 \Rightarrow 0$  is selected and passed  
then 0 should be enabled

⇒  $I_1 = 0 \Rightarrow 0$  is selected and passed

⇒  $I_0 = 1 \Rightarrow 1$  is selected and  $O_1$  is enabled

### 34. CONSTRUCTING 4x2 DECODER USING 1x2 DECODER

Construct  $2 \times 4$  decoder using  $1 \times 2$  decoders?



### 35. CONSTRUCTING 6x64 DECODER USING 3x8 DECODER

64 lines  $\rightarrow$  one present, Each device covers 8 lines

$$\Rightarrow 64 \text{ lines} \rightarrow 64/8 \text{ devices} = 8 \text{ decoders}$$

$$8 \text{ lines} \rightarrow 8/8 \text{ devices} = 1 \text{ decoder}$$

(9) Decoders are needed.

No. of devices

EX:  $6 \times 6$

$4 \times 16$

levels =

No. of de

Now,  $6 \times 64$  decoder using  $2 \times 4$  decoders

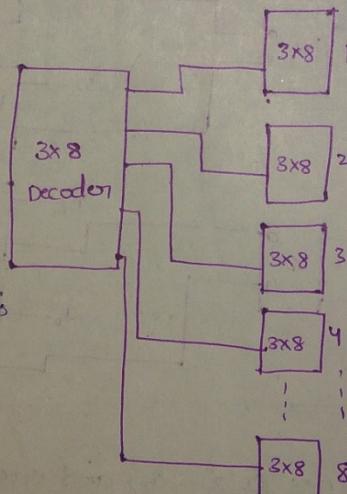
64 lines and each device covers 4 lines

$$= 64 \text{ lines} - \frac{64}{4} \text{ Devices} = 16$$

$$16 \text{ lines} - \frac{16}{4} \text{ devices} = 4$$

$$4 \text{ lines} - \frac{4}{4} \text{ D} = 1 \text{ device}$$

21 decoders are needed ( 16 at level 1  
4 at level 2  
1 at level 3 )



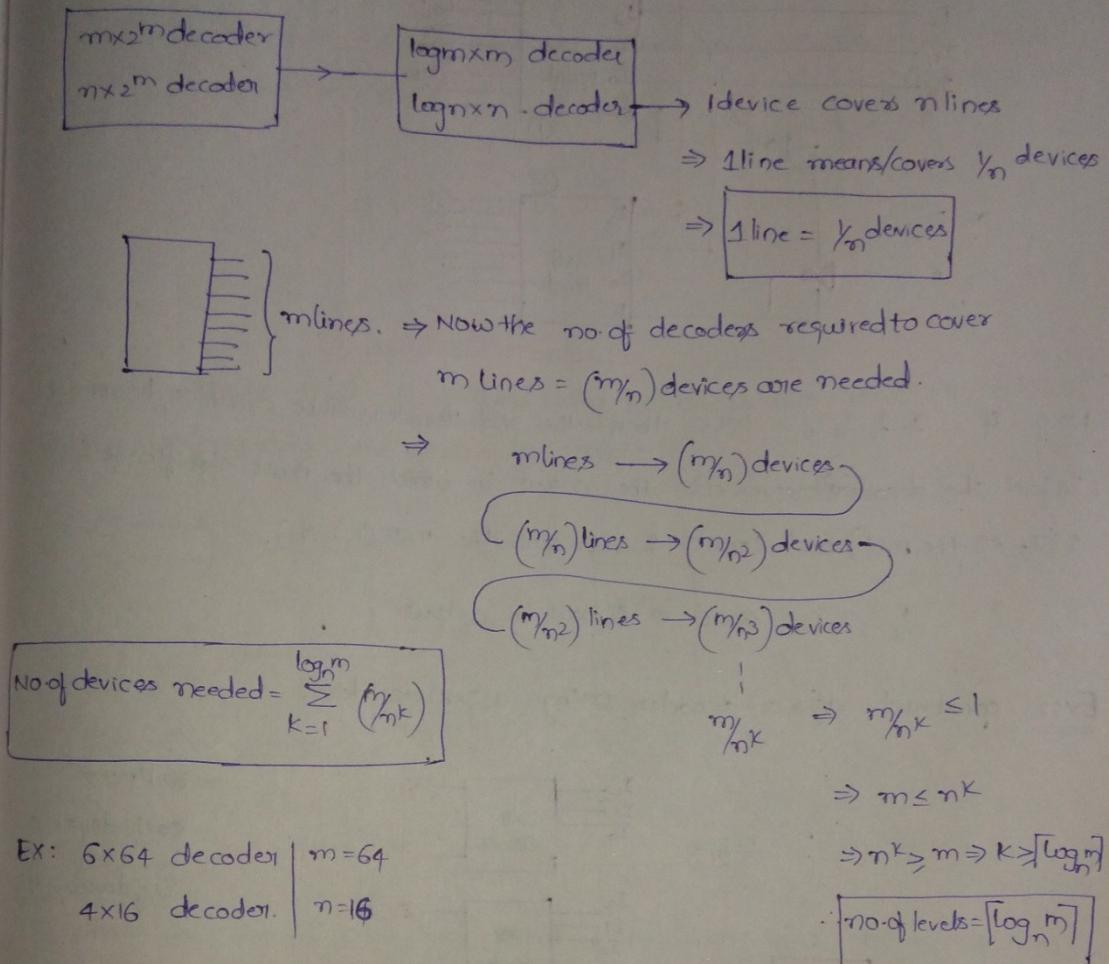
### 36. EXPANSION

Construct

We assume  
like wise).

### 36. EXPANSION OF DECODER IN GENERAL

Now I want to construct  $(m \times 2^m)$  decoder by using  $(n \times 2^n)$  decoder.



$$\text{Ex: } 6 \times 64 \text{ decoder. } m = 64$$

$$4 \times 16 \text{ decoder. } n = 16$$

$$\Rightarrow n^k > m \Rightarrow k > \lceil \log_n m \rceil$$

$$\therefore \text{no. of levels} = \lceil \log_n m \rceil$$

$$\text{levels} = \frac{\log 64}{\log 16} = \lceil \frac{64}{16} \rceil = \lceil \frac{64}{2} \rceil = 2 \text{ levels are needed}$$

$$\text{No. of devices} = \frac{64}{16} + \frac{64}{16^2}$$

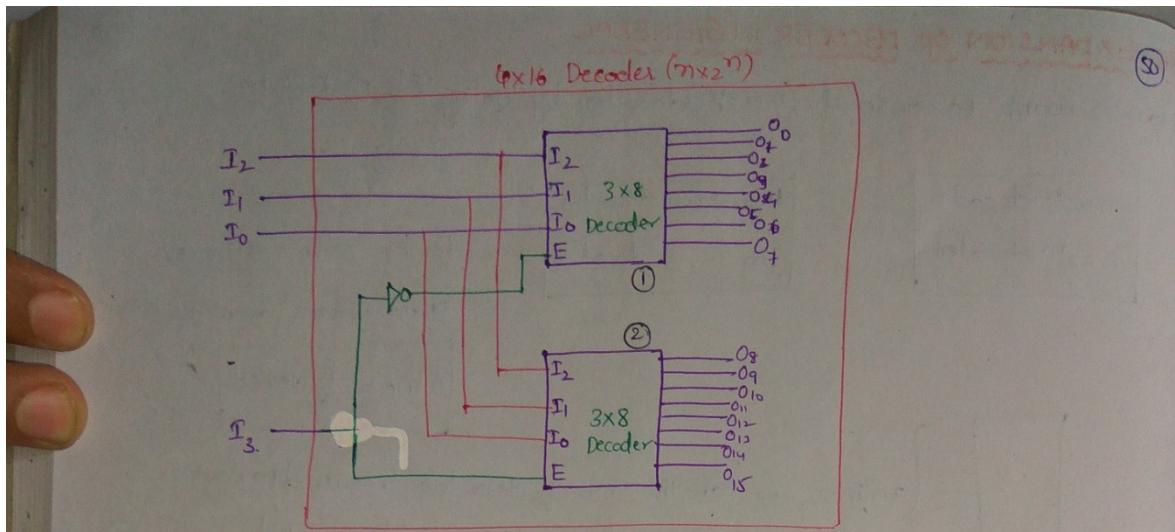
$$= 6 + \frac{64}{16 \times 16}$$

$$= 6 + \lceil \frac{1}{4} \rceil = 6 + 1 = 7 \text{ decoders are needed}$$

### 38. EXPANSION OF DECODERS IN ANOTHER WAY

Construct  $4 \times 16$  Decoder using two  $(3 \times 8)$  decoders.

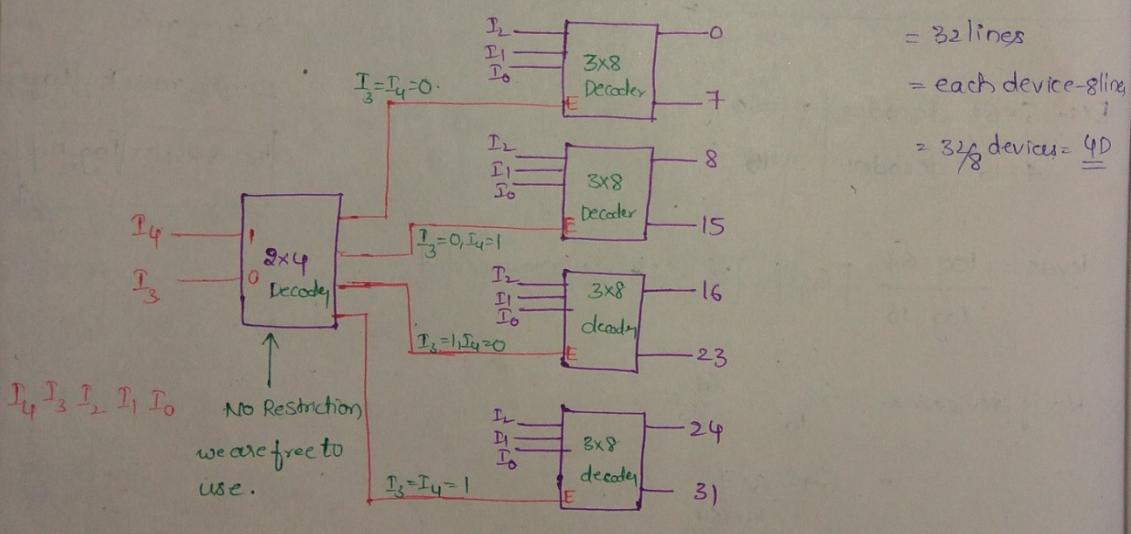
We assume that there are no restrictions (only  $8 \times 8$  and  $16 \times 16$  decoders are available like wise).



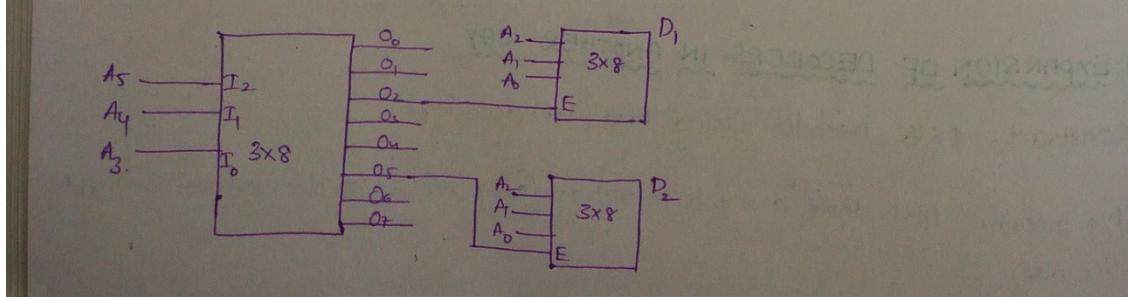
Now, if  $I_3 I_2 I_1 I_0 = 0000$  then the 2nd decoder will be disabled because  $E=0$ .  
 (In fact the decoder ② represents the output in which the most significant bit is 1 ( $I_3$  is the msb) and Decoder 1 represents the msb ( $I_3=0$ )).

$$I_3 I_2 I_1 I_0 = 0000 \Rightarrow O_8 \text{ will be output}$$

Ex:2 construct a  $(5 \times 32)$  decoder using  $4(3 \times 8)$  decoders.



#### 41. FINDING THE ADDRESS RANGES OF DEVICES



#### 42. Example

Consider a  
for how ma

A7  
A6

A5  
A4

A3  
A2  
A1  
A0

The device  
GATE is one

makes  $F=1$  { 0, 1 }

#### 43. FINDING

Consider the  
the Nature

Q

Now, the given Address lines are  $A_5 A_4 A_3 A_2 A_1 A_0$

51

Now, for  $D_1$  to be enabled the  $O_2$  should be enabled  $\Rightarrow A_5 A_4 A_3$  should be = (010)

$\therefore A_5 A_4 A_3 A_2 A_1 A_0$  similarly for  $D_2$ ,  $O_5$  should be enabled  $\Rightarrow A_5 A_4 A_3 = 101$

$$\begin{matrix} 0 & 1 & 0 \\ \left\{ \begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ \vdots & & \vdots \\ 1 & 1 & 1 \end{matrix} \right\} \end{matrix}$$

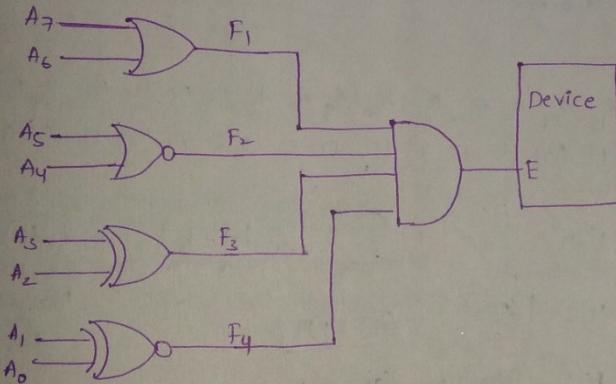
$$\begin{matrix} A_5 A_4 A_3 A_2 A_1 A_0 \\ \left\{ \begin{matrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ \vdots & & \vdots \\ 1 & 1 & 1 \end{matrix} \right\} \end{matrix}$$

## 42. EXAMPLE ON ENABLING A DEVICE

are E=0

bit is

Consider a device that is enabled using 8 address bits as shown below, for how many address the device is enabled



vice-8 lines  
at = 4D

The device is Enabled when the output of AND GATE is '1', the op of AND GATE is one iff all of its inputs should definitely be '1'.

Ways to get 1 output to enable

$$\therefore F_1 = 1, F_2 = 1, F_3 = 1, F_4 = 1$$

3 Combinations

$$\text{makes } F = \{0, 1\}$$

1 Combination

$$F_2 = \{1\} = \{0, 0\}$$

2 Combinations

$$\{0, 1\}$$

2 Combinations

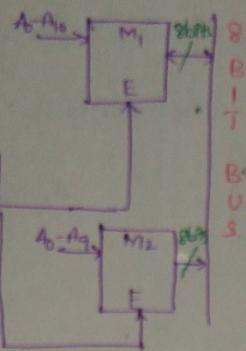
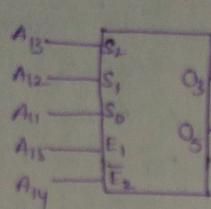
$$\{1, 1\}$$

$\therefore$  Total Addresses =  $3 \times 1 \times 2 \times 2 = 12$  Addresses.

## 43. FINDING THE ADDRESS RANGES OF MEMORY DEVICES

Consider the following interface of two memory devices with 3x8 decoder. Compute the Native and Address range of each memory device

(52)



Clearly  $M_1$  is Bidirectional Device  
 $M_2$  is unidirectional Device  
 $\rightarrow M_1 = \text{Ram}, M_2 = \text{Rom}$

$M_1$ : No. of address bits = 7  $\Rightarrow 2^7$  address  
 $\Rightarrow 2^7$  words are stored  
 $\Rightarrow$  size of each word = 8 bits  
 $\Rightarrow$  size of  $M_1 = 2^7 \times 8 = 2^{14}$  bits  
 $= 2^{11} B$

$M_2$ : No. of address bits =  $A_7 A_8 A_9 \dots A_0 = 10$

$\Rightarrow 2^{10}$  Address core possible  $\Rightarrow 2^{10}$  words can be uniquely identified

$\Rightarrow$  size of each word =  $2^{10} \times 8$  bits =  $2^{13}$  bits =  $2^{10}$  Bytes.

$M_1 = (A_0 - A_{15})$  core present

To enable  $M_1$ ,  $\Rightarrow Q_5$  should be enabled  $\Rightarrow S_2, S_1, S_0 = (101)$  & the decoder should also be enabled so that  $S_2, S_1, S_0 = (101)$  will be enabled.

$A_{15} A_{14} A_3 A_{12} A_{11} A_{10} A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0$

$(1)(0)(1)(0)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)$

$M_1: (A800H) - (AFFFH)$

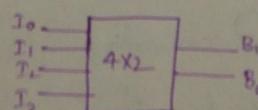
$M_2: (9800H) - (9FFFH)$  (enumerate as you did above)

Now combine 4x4 together

#### 44. INTRODUCTION TO ENCODERS

Ex: 4x2 Encoder (Non-priority)

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



Mux:  $2^n \times 1$   
DEMux:  $1 \times 2^n$   
DECODER:  $n \times 2^n$   
ENCODER:  $2^n \times n$

$$B_1(I_3, I_2, I_1, I_0) = I_0^1 I_1^1 I_2^1 I_3 + I_0^1 I_1^1 I_2^1 I_3^1$$

$$\boxed{B_1 = I_0^1 I_1^1 (I_2 \oplus I_3)}$$

$$B_2(I_3, I_2, I_1, I_0) = I_0^1 I_1^1 I_2^1 I_3 + I_0^1 I_1^1 I_2^1 I_3^1$$

$$= I_0^1 I_2^1 (I_1^1 I_3 + I_2^1 I_1)$$

$$\boxed{B_2 = I_0^1 I_2^1 (I_1 \oplus I_2)}$$

→ They  
→ They  
→ They

→ Due to  
45. PRIORITY

priority

Highest

$I_0$	3
$\emptyset$	2
$\emptyset$	1
$\emptyset$	0
1	1

46. INTRO

→ Hazards

47. HAZAR

→ The mal  
and per

→ The per  
of the c

→ The ha

→ The s

This te

Procedure f

→ Inactive c

the logic t

→ Apply the

→ The input

- They convert one code to another
- They perform lossless compression
- They are of two types
  - Non priority (Doesn't support simultaneous i/p activation)
  - Priority (supports simultaneous i/p activation) and used for interrupt servicing
- Due to static priorities, the lower priority i/p is exposed to starvation

(53)

#### 45. PRIORITY ENCODER

Priority of Inputs:  $I_3 > I_2 > I_1 > I_0$ .

Highest suffix i/p is given highest priority.

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

$$B_1 = I_3 + I_2 \bar{I}_3$$

$$B_1 = I_2 + I_3$$

$$B_0 = I_3 + I_1 \bar{I}_2 \bar{I}_3$$

$$B_0 = \bar{I}_3 + \bar{I}_2 I_1$$

#### 46. INTRODUCTION TO HAZARDS

- Hazards
  - Temporary Hazards
  - permanent Hazards → mainly occurs because of open circuiting / short circuiting.

#### 47. HAZARDS IN DIGITAL CIRCUIT

→ The mal function of a digital circuit is called hazard. They can be temporary and permanent. The temporary hazards are due to uneven delays of inputs.

→ The permanent hazards are resulted due to open circuit or short circuit of the connecting leads (terminals).

→ The hazards can be stuck at 0 (s-a-0) stuck at 1 (s-a-1)

→ The single fault analysis is made using "path sensitization" technique  
This technique provides test vector.

#### Procedure for finding Test vector

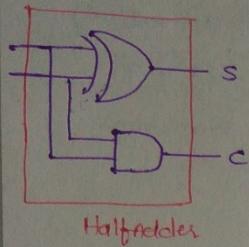
→ Inactive all the paths except the tested one, for AND and NAND apply logic 1 for inactivation, for OR, NOR apply logic 0 for inactivation.

→ Apply the opposite logic value at tested place

→ The input combinations satisfying the above requirements form test vector.

### 49. HALF ADDER

> Half adder - (Two Input, Two output) circuit  $\rightarrow$



$$\text{Sum} = S = \overline{x}y + x\overline{y} = x \oplus y$$

$$\text{Carry} = xy.$$

x	y	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

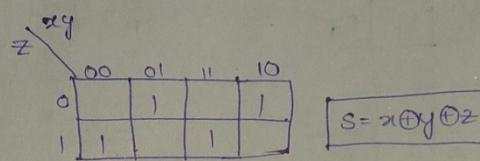
Now, we can

C<sub>4</sub> ←

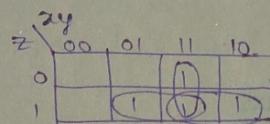
### 50. FULL ADDER

used for adding 3 bits.

x	y	z	c	s
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



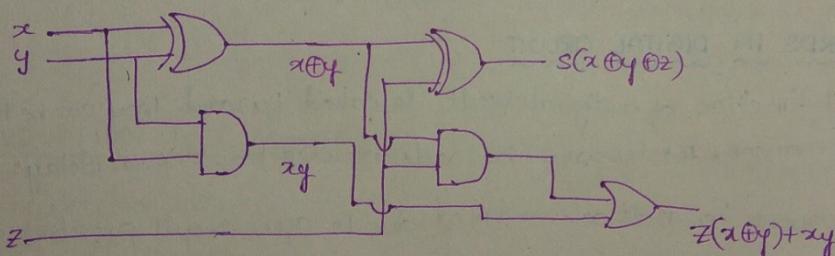
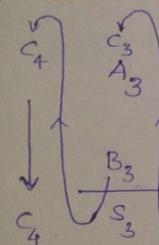
$$S = x \oplus y \oplus z$$



$$c = xy + yz + zx$$

$$c = z(x \oplus y) + xy$$

### 52. CARRY



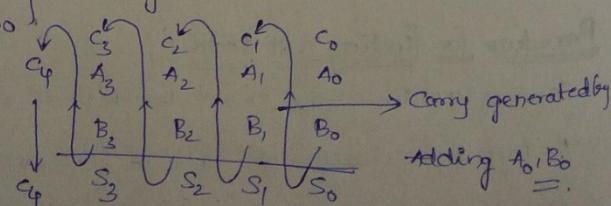
Now, let us

### 51. RIPPLE CARRY ADDER

The ripple carry adder is used to add two binary numbers.

1st Binary number = A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub> } Now, say carry 'c<sub>0</sub>' is provided, then

2nd Binary number = B<sub>3</sub> B<sub>2</sub> B<sub>1</sub> B<sub>0</sub>



Now, C<sub>1</sub> =

C<sub>2</sub> =

C<sub>3</sub> =

C<sub>4</sub> = [C<sub>0</sub> P<sub>0</sub>]

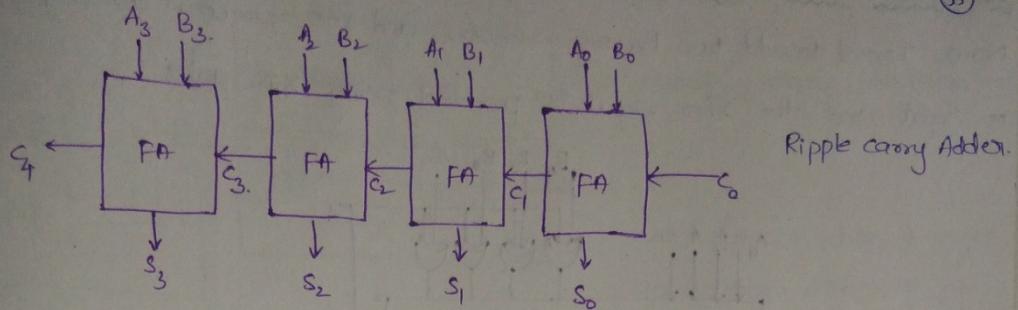
C<sub>4</sub> = C<sub>0</sub> P<sub>0</sub>

=

(54)

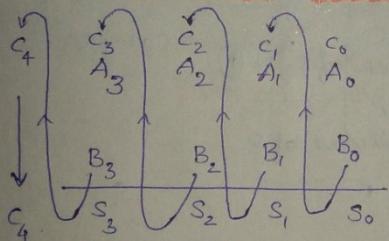
Now we can implement this using Full Adder.

(55)



⇒ The disadvantage of the Ripple carry adder is  $C_4$  is generated after  $C_0$  and  $C_2$  is generated only after  $C_1$ . ∴ The full adder cannot be executed in parallel and it takes more time to compute. So some automated circuits are designed and placed at each full adder, and it is called carry look ahead adder.

## 52. CARRY LOOK AHEAD ADDER



$$C_4 = C_0 (A_0 \oplus B_0) + A_0 B_0$$

$$C_2 = C_1 (A_1 \oplus B_1) + A_1 B_1$$

$$C_3 = C_2 (A_2 \oplus B_2) + A_2 B_2$$

$$C_4 = C_3 (A_3 \oplus B_3) + A_3 B_3$$

} Now, Apply  
Back Substitution  
method.

Now, let us assume  $G_i = A_i B_i$ ,  $P_i = A_i \oplus B_i$  Now,  $C_i = C_0 P_0 + G_i$

↓  
Generating func

↓  
Propagating func

$$C_2 = G_1 P_1 + G_1$$

$$C_3 = C_2 P_2 + G_2$$

$$C_4 = C_3 P_3 + G_3$$

$$\text{Now, } C_1 = C_0 P_0 + G_0$$

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

$$C_3 = (C_0 P_0 P_1 + G_0 P_1 + G_1) P_2 + G_2$$

$$\left| \begin{array}{l} C_3 = (C_0 P_0 P_1 + G_0 P_1 + G_1) P_2 + G_2 \\ C_3 = C_0 P_0 P_1 P_2 + G_0 P_1 P_2 + G_1 P_2 + G_2 \end{array} \right.$$

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

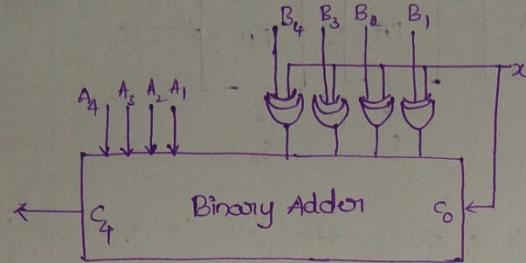
$$C_4 = C_0 P_0 P_1 P_2 P_3 + G_0 P_1 P_2 P_3 + G_1 P_2 P_3 + G_2 P_3 + G_3$$

=

### 56. BINARY ADDER - SUBTRACTOR

Now, I need to add two Binary numbers and one number is directly provided as Input and the other Input is provided using XOR gates.

56



Now  $B \oplus 0 = B$  } if  $x=0$  then the o/p of XOR GATES will be same  $B_4 B_3 B_2 B_1$   
 $B \oplus 1 = \bar{B}$  } and it behaves Binary adder.

If  $x=1$  then the o/p of XOR GATES will be Complimented

$(\bar{B}_4, \bar{B}_3, \bar{B}_2, \bar{B}_1)$  = 1st compliment

If  $x=1$ ,  $A + (\text{1st compliment of } B) + 1 \rightarrow \text{carry's}$

=  $A + 2\text{'s compliment of } B$

=  $A - B$

Now, if

$S_4 S_3$  sh  
(sum's)

The Adder behaves as Adder when  $x=0$

" " " " Substractor " "  $x=1$ .

The above circuit not only functions as Adder and subtractor.

A	B	x	Function
BCD	0011	0	BCD $\rightarrow$ Ex-3 Converter
Ex-3	0011	1	Ex3 $\rightarrow$ BCD Converter
1001	BCD	1	$(9-B) = 9\text{'s compliment of given no. B in (BCD)}$
1010	BCD	1	$(10-B) = 10\text{'s compliment of given no. B in (BCD)}$

58. INVA

Invali

### 57. BCD ADDER

The output is Invalid if o/p is  $> 9$ , and we do a small correction (we add 6 to the result) so we need two Adders.

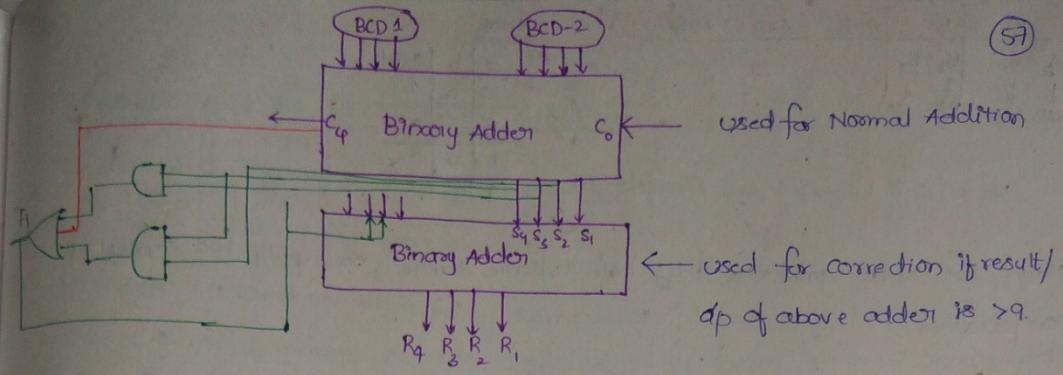
59. 2BIT

Now,

Exclusive

provided

56



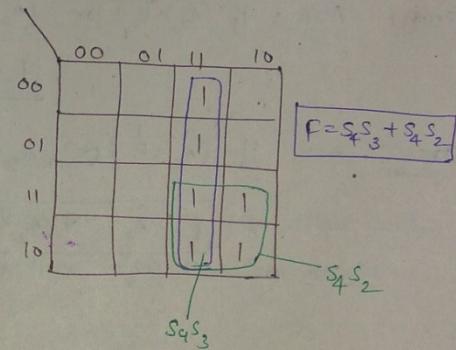
used for correction if result/

dp of above adder is >9.

⇒ Correction is Required when  $S_4S_3S_2S_1 > 9$  (or) if  $C_4 = 1$ , Now let us form a func

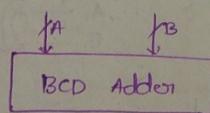
where the sum of  $S_4S_3S_2S_1 > 9$ , now the combinations for which the sum is >9 come.

$S_4$	$S_3$	$S_2$	$S_1$	F
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1



Now, if the sum have to be greater than 10 then the either  $S_4S_2$  should be 1 or  $S_4S_3$  should be 1, irrespective of other combinations. If  $S_4S_2$  or  $S_4S_3$  are 1 then (sum is >9) and when  $C_4=1$  the sum is >9.

### 58. INVALID COMBINATIONS OF BCD ADDER



A, B are each of 4 bits so 16 combinations are possible for A and 16 for B, out of which the sum (0-9) (10 mors) are valid.

$$\text{Invalid combinations} = \text{Total combinations} - \text{Valid combinations}$$

$$= 16 \times 16 - (10 \times 10)$$

$$= 156$$

dd 6 to

### 59. 2BIT COMPARATOR

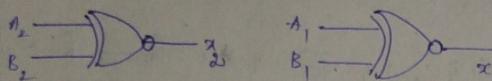
Now, say we have to compare unsigned numbers.  $[00 < 10] [10 > 01] (11 = 11)$

Exclusive-NOR is going to act as a Bit comparator. (for 2 bits)

A	B	$A=B$	$A < B$	$A > B$
$A_2 A_1$	$B_2 B_1$			
↓ msb ↓ lsb	↓ msb ↓ lsb			

XNOR		
a	b	$a \oplus b$
0	0	1
0	1	0
1	0	0
1	1	1

Now, the XNOR produces 1 iff both a, b are same, now apply the concept to comparators.



This diagram represents  $A = B$ .

Case(1):  $A = B$  : if  $x_1 = 1$  and  $x_2 = 1$  i.e.  $x_1 \cdot x_2 = 1$

Case(2):  $A > B$  : if  $(A_2 > B_2)$  or  $(A_2 = B_2 \text{ and } A_1 > B_1)$  then  $A > B$

$$F = A_2 \bar{B}_2 + x_2 A_1 \bar{B}_1 \quad \text{if } F = 1 \text{ then } A > B$$

Case(3):  $A < B$  : if  $(A_2 < B_2)$  or  $(A_2 = B_2 \text{ and } A_1 < B_1)$

$$F = \bar{A}_2 B_2 + x_2 \bar{A}_1 B_1 = \text{if } A < B$$

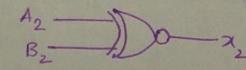
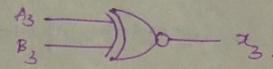
### Q. 3, 4 Bit comparators

$$A = A_3 A_2 A_1$$

$$B = B_3 B_2 B_1$$

case(1):  $A = B$

$$x_3 \cdot x_2 \cdot x_1 = 1$$



case(2):  $(A > B)$ :  $A_3 \bar{B}_3 + x_3 \cdot A_2 \bar{B}_2 + x_3 x_2 A_1 \bar{B}_1 = 1$



case(3):  $(A < B)$ :  $\bar{A}_3 B_3 + x_3 \bar{A}_2 B_2 + x_3 x_2 \bar{A}_1 B_1 = 1$

②

$$A = A_4 A_3 A_2 A_1$$

$$B = B_4 B_3 B_2 B_1$$

$$\downarrow \downarrow \downarrow \downarrow$$

$$x_4 x_3 x_2 x_1$$

$$(i) A = B : x_4 x_3 x_2 x_1 = 1$$

$$(ii) A > B : A_4 \bar{B}_4 + x_4 A_3 \bar{B}_3 + x_4 x_3 A_2 \bar{B}_2 + x_4 x_3 x_2 A_1 \bar{B}_1$$

$$(iii) A < B : \bar{A}_4 B_4 + x_4 \bar{A}_3 B_3 + x_4 x_3 \bar{A}_2 B_2 + x_4 x_3 x_2 \bar{A}_1 B_1$$

If there are  $n$  bit numbers, then the no. of combinations in which  $A = B$  are  $2^n$

$$A > B = \frac{2^n - 2^n}{2^n}$$

$$A < B = \frac{2^n - 2^n}{2^n}$$

### 3. SR-FLIP FLOP

S → D

R → D

### 1. INTRO

→ The e  
on pr

→ The pre  
new c  
presen

Inputs -

### 2. LATCH

→ A Flipp

→ A memo

stored  
be step

→ The basi

→ It has

called

## 4. SEQUENTIAL CIRCUITS

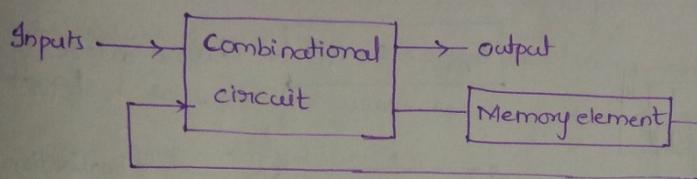
58

59

### 1. INTRODUCTION TO SEQUENTIAL CIRCUITS.

→ The external o/p of a sequential circuit depends on external i/p's and on present contents of memory element

→ The present contents of the memory elements are called present state and the new contents of memory elements are obtained by taking external inputs and present state. This is called Next state.



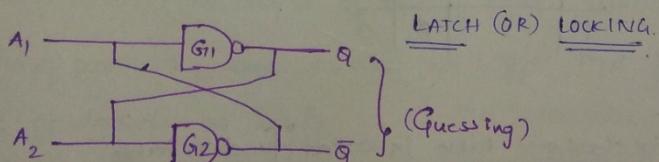
### 2. LATCH AND FLIP FLOP

→ A flip flop is going to store 1 bit of information

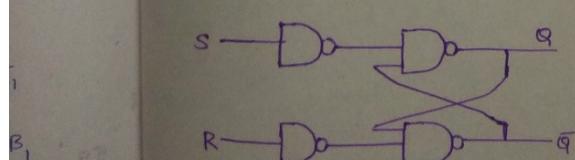
→ A memory element is some medium in which one bit of information can be stored or retained until necessary, and thereafter its contents can be replaced by new values.

→ The basic binary or digital memory circuit is known as flip flop.

→ It has 2 stable states which are known as '1' or '0'. So it is called called bistable multivibrator.



### 3. SR- FLIP FLOP



→ Let the present state be represented by  $Q_n$  and the next state be represented by  $Q_{n+1}$

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

The o/p is not just a func of the present i/p's but also previous outputs.

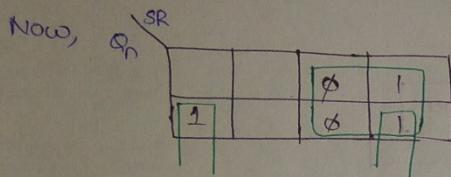
$$\begin{aligned}
 3 \text{ are } & 2^n \\
 &= \frac{2^n}{2} \cdot \frac{2^n}{2} \\
 &= \frac{2^n}{2} \cdot \frac{2^n}{2} \\
 &= \frac{2^n}{2} \cdot \frac{2^n}{2}
 \end{aligned}$$

S	R	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	$\emptyset$
1	1	1	$\emptyset$

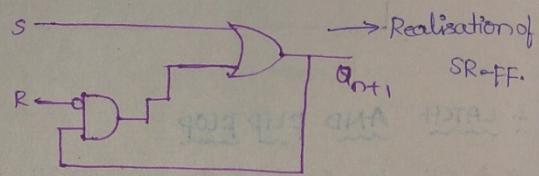
→ we can use the SR-flipflop in only 3 modes → latch, set, Reset modes.

Characteristic table

(6)



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



→ Excitation table represents the present output and what is the output that you are expecting in the next state and what are the combinations that you should provide to get the op of next state that you have assumed.

$Q_n$	$Q_{n+1}$	S	R
0	0	0	$\emptyset$
0	1	1	0
1	0	0	1
1	1	$\emptyset$	0

→ minimised characteristic table is called "function table"

S	R	$Q_{n+1}$
0	0	$Q_n$
1	0	1
0	1	0
1	1	$\emptyset$

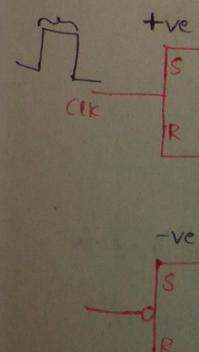
#### 4. CLOCK

→ Now, we some then we show not to

→ clock va

→ whenever the val enabled clock

⇒ Edge trip only on the o/p



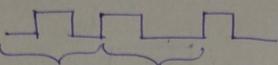
## 4. CLOCKED FLIPFLOPS

(6)

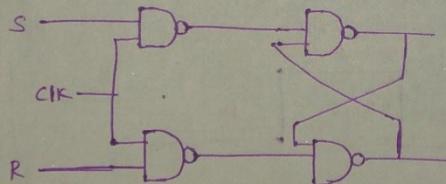
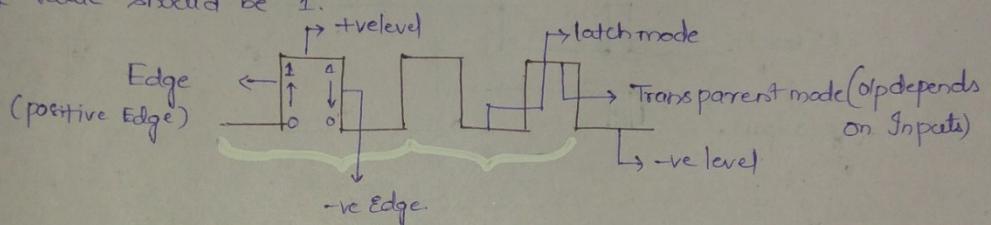
Now, what happens in general with SR-flipflop/flip flop is, because of some external disturbances the  $\bar{Q}$  symbol will suddenly fluctuate and then the output will change. And so what we are supposed to do is we should make the Inputs work only during sometime and we should not let it change during the other time.

)

→ clock varies b/w 0 and 1

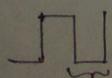
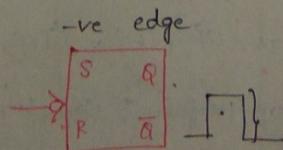
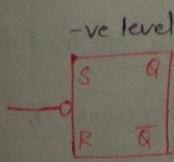
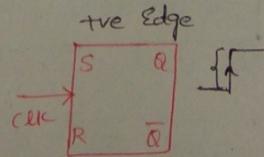
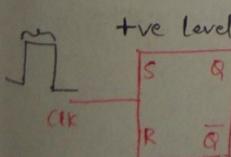


→ whenever clock is '0' then the o/p of NAND gates will be 1 and 1 irrespective of the values of S,R it is as good as having S,R as 0,0  $\Rightarrow$  latch mode is enabled. So, when you want to change the o/p depending on S,R then the clock value should be '1'.



⇒ positive level triggered  
flipflop.

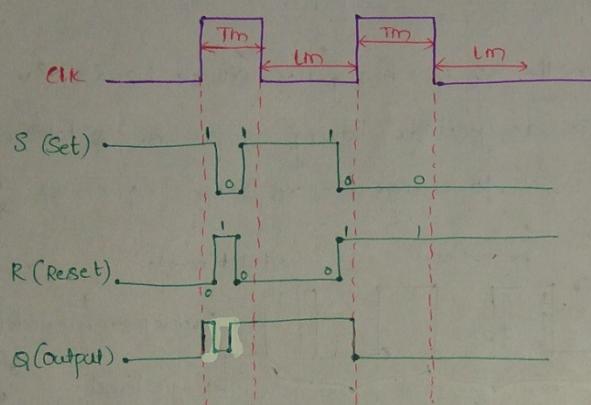
$\Rightarrow$  Edge triggered means the output of the flipflop depends on the Input only on some particular edge so, when we trigger that particular edge then the o/p of FF depends on Input.



## 5. POSITIVE LEVEL TRIGGERED

Now, let's see how the positive level triggered FF react to the inputs.

⇒ In SR-FF 'S' means Set and 'R' means Reset. The reason is whenever 'S' is '1' the output is always going to be one, and whenever R=1 the output is always going to be zero. and when both S=R=0, then o/p will be in latch mode



T<sub>m</sub> = Transparent mode  
L<sub>m</sub> = Latch mode

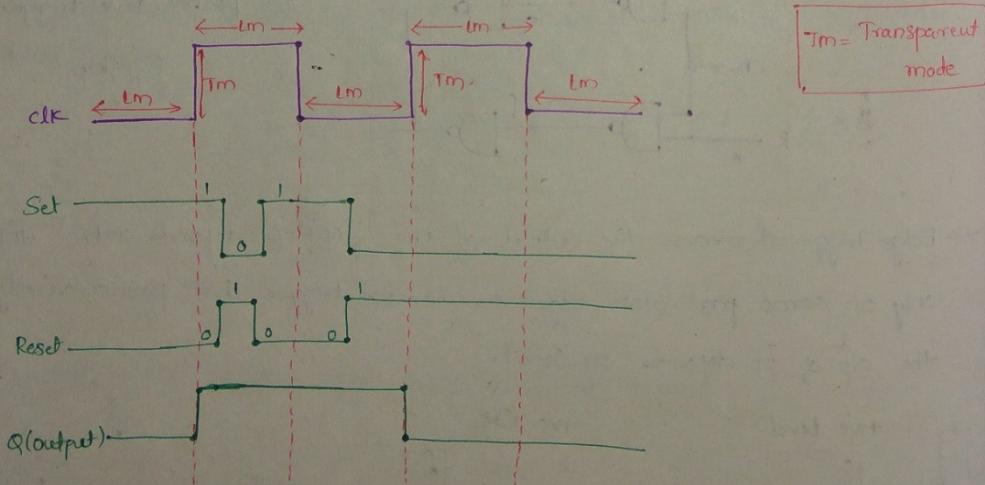
⇒ Assume the o/p is present at '0' level.

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

Characteristic

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

## 6. EDGE TRIGGERED



T<sub>m</sub> = Transparent mode

Excitation table

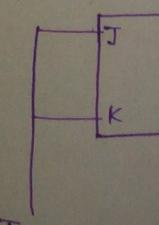
Q : Q <sub>n+1</sub>
0 → 0
0 → 1
1 → 0
1 → 1

## 8. T- FLIP FLOP

T-Flip flop = To

## 7. JK- FLIP FLOP

⇒ The J-K-flip flop is same as SR flip flop (J=S, K=R) It is defined for J=K=1 also and the flip flop performs complementation for (J=K=1).



(62)

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

(63)

Whenever  
the  
will be

out  
mode  
node

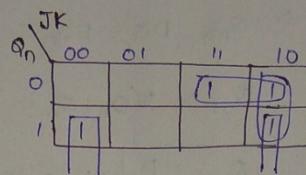
O/P is  
level.

Characteristic Table:

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

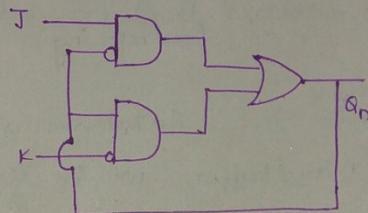
Same as  
SR-FF

Complementation.



$$F = J\bar{Q}_n + \bar{K}Q_n$$

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

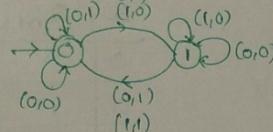


Excitation table

$Q'$	$Q_{n+1}$	J	K
0 → 0	0	0	∅
0 → 1	1	∅	
1 → 0	∅	1	
1 → 1	∅	0	0

State diagram

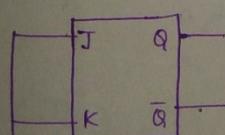
⇒ The 2 states of flip flop are for 0, 1



look at function  
table and draw.

### 8. T-FLIP FLOP

T-Flip flop = Toggle Flip flop (This flip flop is either latching or complementing so it is called Toggling)



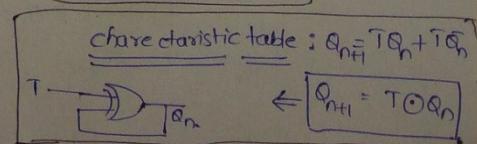
T	$Q_n$	$Q_{n+1}$
0	0	$Q_n$
1	1	$\bar{Q}_n$

function table

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

Q	$Q_{n+1}$	T
0	0	0
0	1	1
1	0	1
1	1	0

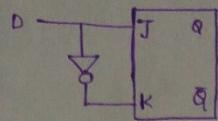
Excitation table



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

## 9. D-FLIP FLOP (Delay-FlipFlop)

(64)



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

= Functiontable

### Characteristic table

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

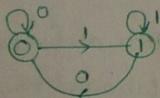
$$\Rightarrow Q_{n+1} = D\bar{Q}_n + DQ_n$$

$$Q_{n+1} = D(Q_n + \bar{Q}_n)$$

$Q_{n+1} = D$  ∴ output is directly depending on input  
and there is no need of memory element at all (No feedbacking).

$\Rightarrow D \rightarrow \text{Delay Buffer} \rightarrow D$ . (what are giving at the input the same input will be displayed as output after some delay so it is called delay flipflop)

### State diagram



### Excitation table

D.	$Q_{n+1}$	D.
0	0	0
0	1	1
1	0	0
1	1	1



### Excitation

Q
0
0
1
1
0

### 12. INTRODU

Conversion

1> Get the

2> Replace the

3> Obtain the

### 10. EXAMPLE ON FLIP FLOP

The characteristic equation of a flipflop is  $Q_n = \bar{x}_1\bar{Q} + \bar{x}_2Q$ . Define the behaviour of this.

#### Function table

$x_1$	$x_2$	$Q_n$	$Q_n = \bar{x}_1\bar{Q} + \bar{x}_2Q$
0	0	1	→ Set state
0	1	$\bar{Q}$	→ Toggle
1	0	Q	→ latch mode
1	1	0	→ Compl. Reset state

(64)

characteristic table

able

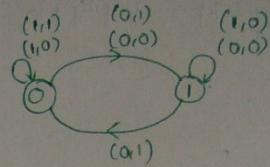
$x_1$	$x_2$	$Q$	$Q_n$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Excitation table

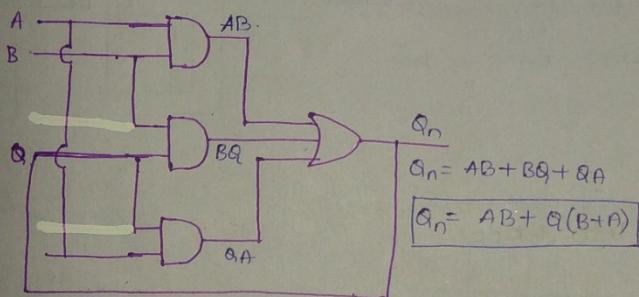
$Q$	$Q_n$	$x_1$	$x_2$
0	0	1	0
0	1	0	0
1	0	0	1
1	1	0	0

state diagram

(65)

m input  
(No)II. EXAMPLE ON FLIPFLOP-2

consider the following realization, construct excitation table?

Input  
delay

function table

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

→ Reset  
→ latch  
→ latch  
→ setting

Excitation table

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

12. INTRODUCTION TO FLIP FLOP INTER CONVERSION

Conversion of given flip flop to another flip flop.

1&gt; Get the characteristic table of target FF

2&gt; Replace the next state using excitation of the given FF

3&gt; Obtain the expressions for the i/p of given FF and realise them.

Ex1: Conversion of J-K flipflop to T-flipflop.

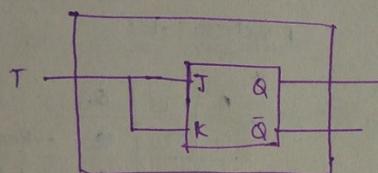
(Q)

i) Target FF = T-flipflop  $\Rightarrow$  The characteristic table =

T	Q	$Q_{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

2) Replace the next state with excitation

T	Q	$Q_{n+1}$	J	K
0	0 $\rightarrow 0$	0	0	$\emptyset$
0	1 $\rightarrow 1$	0	$\emptyset$	0
1	0 $\rightarrow 1$	1	$\emptyset$	$\emptyset$
1	1 $\rightarrow 0$	$\emptyset$	1	



Now, write the values of JK where the output changes to  $(0 \rightarrow 0), (1 \rightarrow 1), (0 \rightarrow 1), (\text{one} \rightarrow \text{zero})$

$\Rightarrow$  Now, characteristic eqn for  $J=T$   
 $K=T$

$$Q_n \begin{cases} 0 \\ 1 \end{cases} \Rightarrow J=T \quad Q_n \begin{cases} 0 \\ 1 \end{cases} \Rightarrow K=T$$

Ex2: conversion of T-FF into J-K FF

i) Target FF = J-K-flipflop  $\Rightarrow$  The characteristic table =

J	K	$Q_n$	$Q_{n+1}$	T
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	0	1
1	0	0	1	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	1

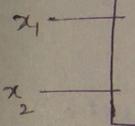
write the values of T for which output changes from  $(Q_n \rightarrow Q_{n+1})$  looking at characteristic table of T.

characteristic equation for T =  $Q_n \begin{matrix} JK \\ 00 \\ 01 \\ 11 \\ 10 \end{matrix} \begin{matrix} 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{matrix}$

$$T = J\bar{Q}_n + KQ_n$$

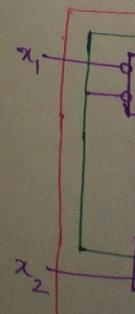
13. INT  
A nec  
it us

$\Rightarrow$  Reca  
: T -



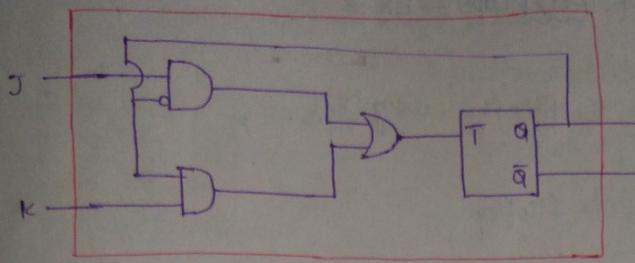
Function

$x_1$	$x_2$
0	0
0	1
1	0
1	1



(66)

(67)



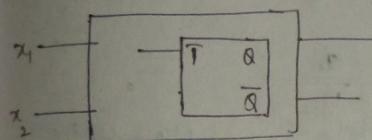
### 13. INTER CONVERSION OF FLIPFLOPS - EXAMPLE-1

JK  
pesto  
Zero)

A new FF  $x_1, x_2$  has characteristic expression  $Q_n = \bar{x}_1 \bar{Q} + \bar{x}_2 Q$ . Realise it using T-flipflop.

$\Rightarrow$  Realise it using D<sub>T</sub>-flipflop means the given flipflop is T, and using T-flipflop you want to recognise new flipflop.

K=T



$\therefore$  characteristic tables of  $x_1, x_2$  should be written first and then excitation tables of 'T' should be written.

Function table

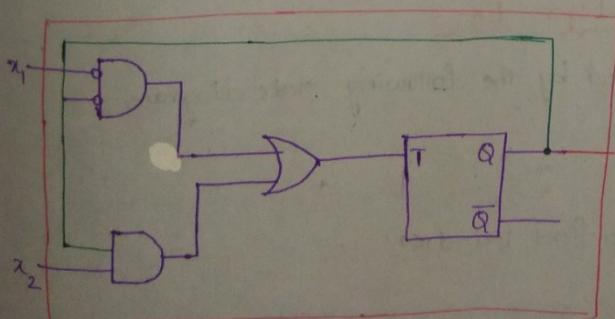
$x_1$	$x_2$	$Q_n$
0	0	1 → set
0	1	$\bar{Q}$ → Toggle
1	0	$Q$ → latch
1	1	0 → Reset

uses of T  
input change  
(+) looking  
onistic table

Characteristic table of  $x_1, x_2, Q$ .

$x_1$	$x_2$	$Q$	$Q_n$	$T$
0	0	0	1	1
0	0	1	One(1)	0
0	1	0	1	1
0	1	1	0	1
1	0	0	0	0
1	0	1	1	0
1	1	0	0	0
1	1	1	0	1

Excitation table of T?



$x_1, x_2$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$$T = \bar{x}_1 \bar{Q} + x_2 Q$$

### 14. INTER CONVERSION OF FLIPFLOPS - EX-2

$$x_1 x_2 \rightarrow T$$

$$Q_n = \bar{x}_1 \bar{Q} + \bar{x}_2 Q_1$$

} construct T-flipflop using  $x_1 x_2$ .

The characteristic table of T-flipflop.

T	Q	Q <sub>n</sub>	x <sub>1</sub>	x <sub>2</sub>
0	0	0	1	∅
0	1	1	∅	0
1	0	1	0	∅
1	1	0	∅	1

Excitation table for  $x_1 x_2$

$$\Rightarrow x_1 = \bar{T} \quad x_2 = T$$

Now, In order to get excitation table of  $x_1 x_2 / Q$  we should get function table of  $x_1 x_2$

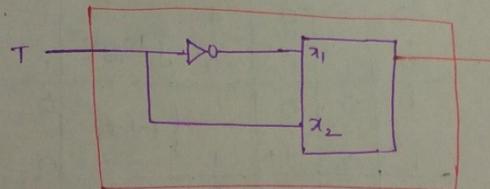
→ characteristic table of  $x_1 x_2 \rightarrow$  Excitation table of  $x_1 x_2$ .

x <sub>1</sub>	x <sub>2</sub>	Q <sub>n</sub>
0	0	1
0	1	0
1	0	0
1	1	0

This are written by  
Eqn.  $Q_n = \bar{x}_1 \bar{Q} + \bar{x}_2 Q$ .

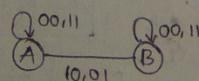
x <sub>1</sub>	x <sub>2</sub>	Q	Q <sub>n</sub>
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

→ Now, write  
excitation table  
beside the charac-  
teristic table of  
 $x_1 x_2$

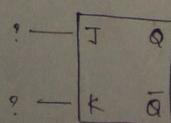


### 15. INTER CONVERSION OF FLIPFLOPS - EX-3

An, X,Y flipflop is represented by the following state diagram.

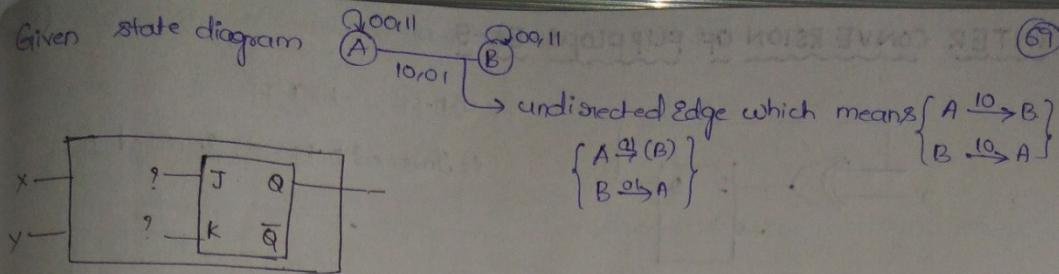


To realise it using J,K flipflop find J and K?



(8)

Given state diagram

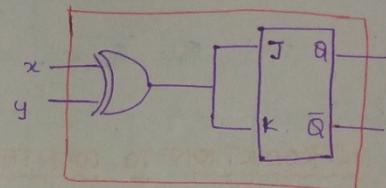


Now find the characteristic Equation of  $x, y$  and excitation table of  $J, K$

$x$	$y$	$q_n$	$q_{n+1}$	$J$	$K$
0	0	0	0	0	∅
0	0	1	1	∅	0
0	1	0	1	1	∅
0	1	1	0	∅	1
1	0	0	1	1	∅
1	0	1	0	∅	1
1	1	0	0	0	∅
1	1	1	1	∅	0

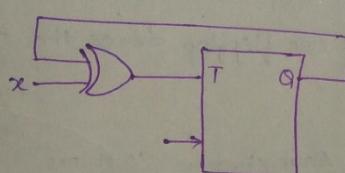
$$J = \bar{x}y + xy = x \oplus y$$

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



## 6. INTER CONVERSION OF FLIPFLOPS - 62-4

What is the behaviour of following one-input flip flop  $x$ ?



a) D-FF      b) T-FF

c) Inverted DFF    d) Inverted T-FF

⇒ This problem is nothing but implementation of  $x$ -flipflop using  $T$ -flipflop.

Characteristic table of  $x$ .

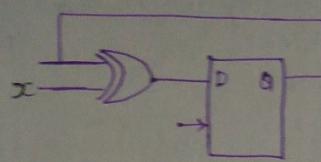
$x$	$q_n$	$q_{n+1}$	$T = x \oplus q_n$
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

⇒ The output  $q_{n+1}$  is a combination

of  $x, q_n, T$  so first find  $T = x \oplus q_n$   
and then find  $q_{n+1}$

$\therefore q_{n+1} = x$ . ∵ whatever you give as input is displayed as output after some delay so it acts as delayed flipflop.

### 17. INTER CONVERSION OF FLIPFLOPS - EX-5



- a) D-FF      b) T-FF  
c) Inverted D-FF    d) Inverted D-FF

x	$Q_n$	$Q_{n+1}$	$D = Q_n \oplus x$
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

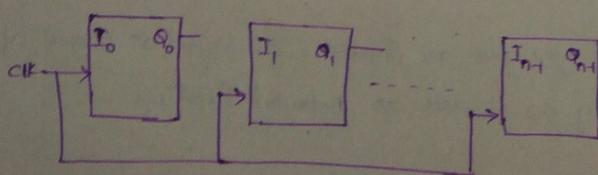
when  $x=0$ ,  $Q_n = Q_{n+1}$   
 $x=1 \cdot Q_n = \overline{Q_{n+1}}$  } T flipflop.

### 18. INTRODUCTION TO COUNTERS

- The counters are used to provide accurate timing and control signals.
- These are of 2 types
  - ↳ Synchronous
  - ↳ Asynchronous
- In Synchronous counters all the flipflops respond to the same clock instances.
- In Asynchronous counters, the output of one flipflop drives the clock of another flipflop.
- Synchronous counters are faster than Asynchronous counters.
- Due to simplicity of design, Asynchronous counters are used in IC fabrication.
- The simplified version of synchronous counter is called shift counters.
- The basic element in shift counter is D-flipflop.
- The Ring counter and Johnson counter are further simplified version of shift counter.

### 19. ASYNCHRONOUS AND SYNCHRONOUS COUNTERS

#### Synchronous Counter

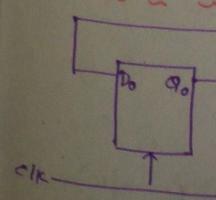


⇒ Propagation delay in Synchronous counter is

$$T_{pdsgn} = T_{ff} + T_{combinational}$$

$$T_{clk} \geq T_{pdsgn}$$

### 21. Mod-2 R

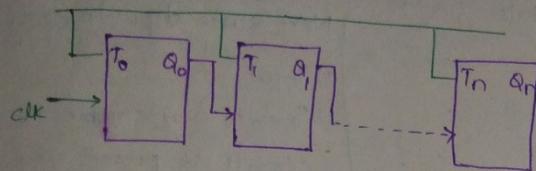


⇒  $T_{dk}$  means the amount of time we should wait before sending the next input signal.

FF

(7)

### Asynchronous counter:



⇒ Propagation delay in Asynchronous Counter

$$T_{pdasyn} = N \cdot T_{FF} + T_{combinational}$$

$$T_{dk} \geq T_{pdasyn} \quad T_{FF} = PD\text{ of each flip-flop.}$$

### 20 SHIFT COUNTERS

#### Synchronous counter

$$\Rightarrow I_i = f(Q_0, Q_1, \dots, Q_{N-1}) \quad 0 \leq i \leq (N-1)$$

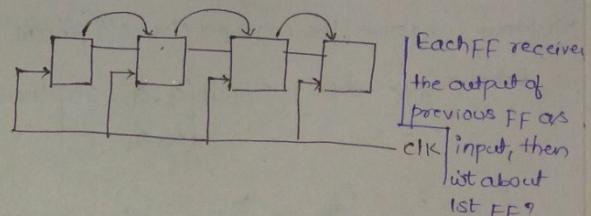
→ Its design is more complex, so simplifications are done as follows:

$$i) I_i = Q_{i-1} \quad [1 \leq i \leq N-1]$$

$$I_1 = Q_0$$

$$I_2 = Q_1$$

$$I_3 = Q_2$$

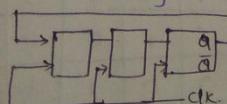


→ Here data is shifted from one flip-flop to another FF so this is called shift counter (D-FFs are used)

$$ii) I_0 = f(Q_{N-1})$$

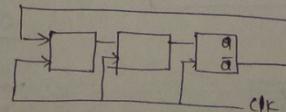
$$I_0 = f(Q_{N-1})$$

$I_0 = f(Q_{N-1})$   
(Ring counter)



$$I_0 = f(\bar{Q}_N)$$

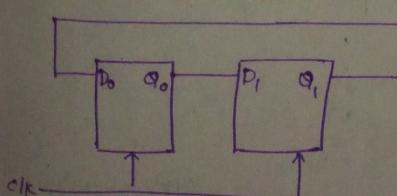
(Johnson Counter)



### 21. Mod-2 RING COUNTERS

Synchr

national



→ Generally there are 2 types of Questions are asked

i) They will give you the counter and ask you what it is counting

ii) They will tell you what you should count and ask you to design the counter.

→ First we should understand what are the states in which the FF (D-FF's) can be.

If we observe there are 2FF and one FF is going to give  $Q_0$  state and another

FF is going to give  $Q_1$  state (Here assume  $Q_0$  = LSB,  $Q_1$  = MSB)

(2)

		Present State		Next State	
D <sub>1</sub>	D <sub>0</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>1N</sub>	Q <sub>0N</sub>
0	0	0	0	0	0
1	0	0	1	1	1
0	1	1	0	0	0
1	1	1	1	1	1

(2)

(1)

(3)

Next state depends on 2 things,

1) Input

2) In case any combinational circuit is present what is the flip to the combinational circuit.

Here  $Q_{0N}$  depends on  $D_0$  and

$D_0$  depends on  $Q_1$  (look at diagram)

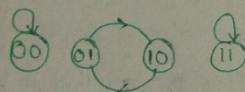
$$Q_{0N} = Q_1 \quad (Q_{0N} = D_0 = Q_1)$$

$$\Rightarrow (Q_{0N} = Q_1)$$

Now,  $Q_{1N}$  is nothing but Input given at  $D_1 \Rightarrow Q_{1N} = D_1$

Now observe the marked circles (present state, New state) and find how they

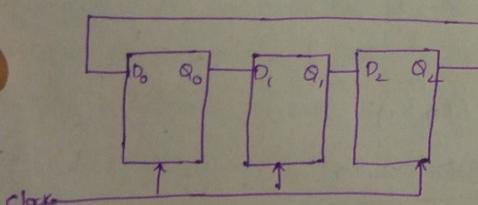
behave, i.e. when Input  $\{Q_0, Q_1 = 00\}$  then they remain in same state  
 $\{Q_0, Q_1 = 11\}$



→ mod 2 counter (only 2 states are in counting).

Using Ring counter we get "mod N" counters.

## 22. MOD 3 RING COUNTERS



$$Q_{2N} = D_0 = Q_0 \Rightarrow Q_{2N} = Q_2$$

$$Q_{1N} = D_1 = Q_0 \Rightarrow Q_{1N} = Q_1$$

$$Q_{0N} = Q_{IN} \Rightarrow Q_{0N} = Q_0$$

$$\begin{aligned} D_0 &= Q_2 \\ D_1 &= Q_0 \\ D_2 &= Q_1 \end{aligned}$$

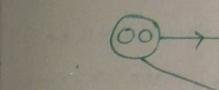
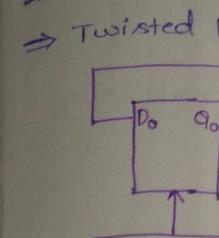
$$\begin{aligned} \Rightarrow Q_{0N} &= D_0 \\ Q_{1N} &= D_1 \\ Q_{2N} &= D_2 \end{aligned}$$

Now,  $Q_0, Q_1, Q_2$  are the states.

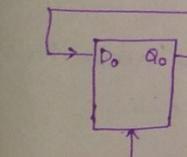
present      Next

Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>2N</sub>	Q <sub>1N</sub>	Q <sub>0N</sub>	D <sub>0</sub>	D <sub>1</sub>	D <sub>2</sub>
0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1	0
0	1	0	1	0	0	0	0	1
0	1	1	1	0	0	0	1	1
1	0	0	0	0	1	1	0	0
1	0	1	0	1	1	1	1	0
1	1	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	1

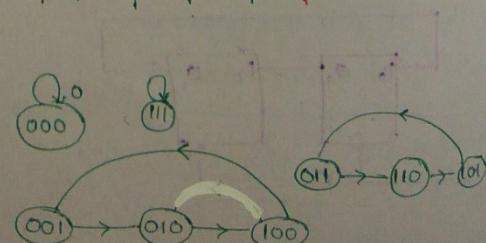
## 23. MOD 4



## 24. MOD 6



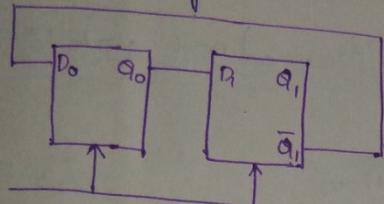
$$\begin{aligned} Q_2 &= Q_1 \\ D_1 &= Q_0 \\ D_0 &= \bar{Q}_2 \end{aligned}$$



- If a Ring counter has 'n' D-flipflops then it can perform "mod n" calculation  
 → In above Ring counter it has 3-ff so it can perform mod3 operation.

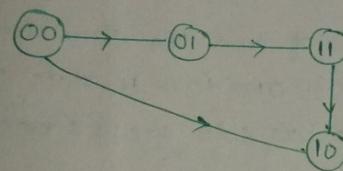
### 23. MOD 4 JOHNSON COUNTER

⇒ Twisted Ring counter or Johnson counter.



There are two states  $Q_0, Q_1, \& Q_0$ .

$Q_1$	$Q_0$	$Q_{IN}$	$Q_{ON}$	$D_0$	$D_1$
0	0	0	1	1	0
0	1	1	1	1	1
1	0	0	0	0	0
1	1	1	0	0	1

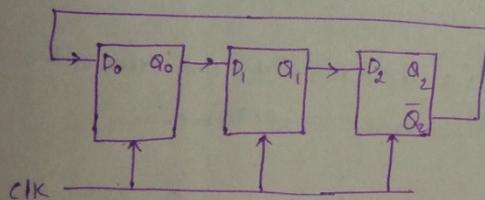


= mod 4 counter.

$$Q_N = D_1 \\ Q_{ON} = D_0$$

$$D_0 = \bar{Q}_1 \\ D_1 = Q_0$$

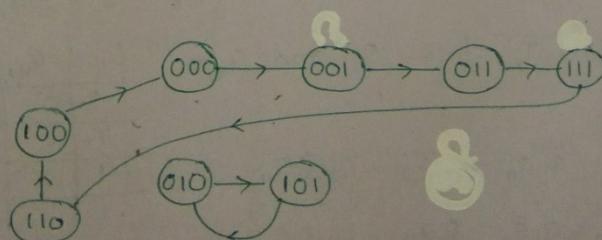
### 24. MOD 6 JOHNSON COUNTER



$$D_2 = Q_1 \\ D_1 = Q_0 \\ D_0 = \bar{Q}_2$$

$$Q_{2N} = D_2 \\ Q_{1N} = D_1 \\ Q_{0N} = D_0$$

$Q_2$	$Q_1$	$Q_0$	$Q_{2N}$	$Q_{1N}$	$Q_{0N}$	$D_2$	$D_1$	$D_0$
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	1
0	1	0	1	0	1	1	0	1
0	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0
1	0	1	0	1	0	0	1	0
1	1	0	1	0	0	1	0	0
1	1	1	1	1	0	1	1	0



← check the loop, it is mod6 counter.

## Q5. MOD-4 GRAY COUNTER USING T-FF

(7)

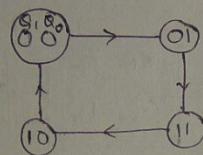
Now, let us say how to design a counter with specifications.

Designing a counter

REVIEW DESIGN OF COUNTERS

- 1) Get the state table from state diagram
- 2) Identify flipflops to be used and replace the concerned next state using excitation table of associated flipflop.
- 3) Get the expressions for inputs and realise them.

Design a synchronous counter for the following using T-flipflops.



⇒ Here two flips are enough, at any instant of time the circuit will be in 00(0), 01(0), 10(0), 11 state so each state has 2 bits so 2 FF's are enough.

⇒ Now, these 2FF should be connected in such a way that when a clock signal is applied it should make any of the above transitions ( $00 \rightarrow 01$ ,  $01 \rightarrow 11$ ,  $11 \rightarrow 10$ ,  $10 \rightarrow 00$ ).

⇒ Now, we need to find the input combinations that lead to the above transitions (This is also called Excitation table). Now, first draw state table.

Present State		Next State		Excitation (Inputs)		→ for 2 flipflops, two inputs.
$Q_1$	$Q_0$	$Q_{IN}$	$Q_{ON}$	$T_1$	$T_0$	
0	0	0	1	0	1	
0	1	1	1	1	0	
1	0	0	0	0	1	
1	1	1	0	1	0	

*written by seeing at diagram*

Now, to find  $T_1$  check  $Q_1$  and  $Q_{IN}$ .

$$T_1 = \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0 = Q_1 \oplus Q_0$$

$$T_0 = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0 = Q_1 \odot Q_0$$

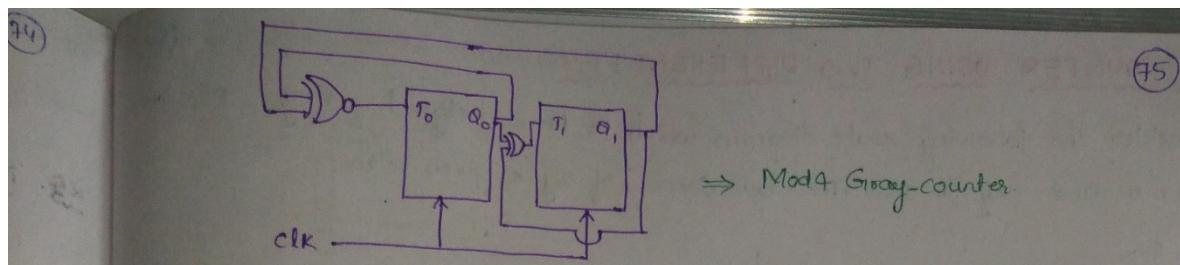
$Q_1$	$Q_{IN}$	$T_1$
0 → 0	0	0
0 → 1	1	1
1 → 0	0	0
1 → 1	1	1

→ latch

$\left\{ \begin{array}{l} \text{if } Q_1 \text{ is complemented} \\ \text{then } T=1, Q_1 \text{ is same} \\ \text{then } T=0 \text{ (look} \\ \text{at function table of} \\ \text{T-flipflop)} \end{array} \right.$

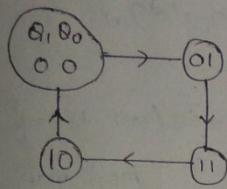
27. MOD

same Q

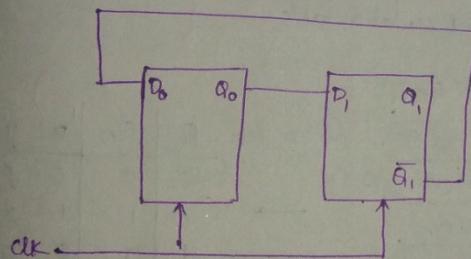


### 26. MOD-4 - GRAY COUNTER USING D-FF

⇒ 2-Flipflops are needed.



present state		Next state		Excitation (Inputs)	
$Q_1$	$Q_0$	$Q_{IN}$	$Q_{ON}$	$D_1$	$D_0$
0	0	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	1	1	0	1	0



$$D_1 = \overline{Q}_0 Q_1 + Q_1 \overline{Q}_0 = Q_1$$

$$D_0 = \overline{Q}_1 \overline{Q}_0 + \overline{Q}_1 Q_0 = \overline{Q}_1 (Q_0 + \overline{Q}_0) = \overline{Q}_1$$

### 27. MOD 4 GRAY COUNTER USING 1D AND 1T FLIPFLOP

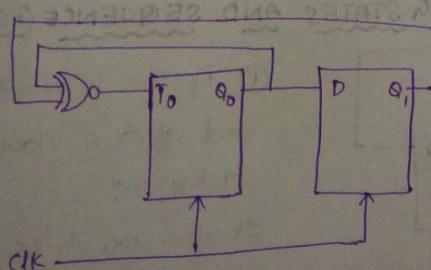
Same Question above

$Q_1$	$Q_0$	$Q_{IN}$	$Q_{ON}$	$D_1$	$T_0$
0	0	0	1	0	1
0	1	1	1	1	0
1	0	1	0	-1	1
1	1	0	0	0	0

$$D_1 = Q_0$$

$$T_0 = \overline{Q}_1 \overline{Q}_0 + Q_1 Q_0$$

$$T_0 = Q_1 \odot Q_0$$

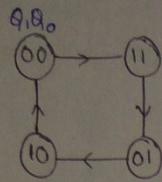


## 28. COUNTER USING TWO DIFFERENT FF's

(26)

First let's

Consider the following state diagram which is to be designed using T- flipflops for msb and x/y for lsb. The behaviour of x/y is given below:



x	y	Qn
0	0	0
0	1	Q
1	0	Q̄
1	1	1

Q <sub>1</sub>	Q <sub>0</sub>	Q <sub>IN</sub>	Q <sub>ON</sub>	T <sub>1</sub>	x	y
0	0	1	1	1	1	∅
1	1	0	1	1	∅	1
0	1	1	0	1	∅	0
1	0	0	0	1	0	∅

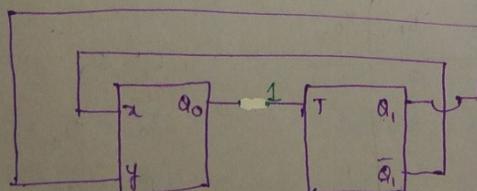
Now  $x = \bar{Q}_1$  (use K-map)  
 $y = Q_1$  (use K-map)

Given fundtable  $\rightarrow$  characteristic table  $\rightarrow$  Excitation table

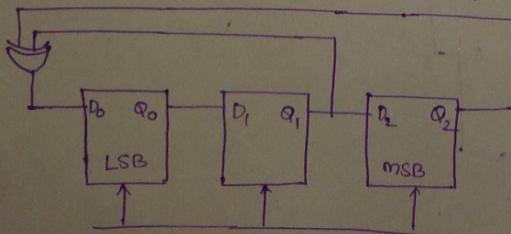
x	y	Qn
0	0	0
0	1	Q
1	0	Q̄
1	1	1

x	y	Q	Qn
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Q	Qn	x	y
0	0	0	∅
0	1	1	∅
1	0	∅	0
1	1	∅	1



## 29. MODEL ON ANALYSIS COUNTING STATES AND SEQUENCE GENERATIONS



① If the initial state is 10 & 10 and after 5 clocks what is going to happen?

② After 100 clocks what will happen?

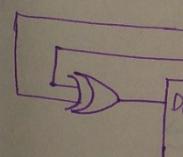
This all depends on what initial state is.

$T_{clk} \geq T_{FF}$

$T_{clk} \geq 20m$

write

## 30. DERIV



$T_{clk} \geq T_{FF}$

Flipflop<sub>2</sub>

(76)

First let's analyse the state diagram.

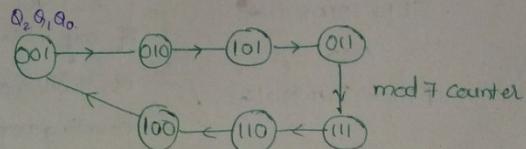
(77)

$Q_2$	$Q_1$	$Q_0$	$Q_{2N}$	$Q_{IN}$	$Q_{ON}$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	0

$$Q_{ON} = D_0 = Q_1 \oplus Q_2$$

$$Q_{IN} = D_1 = Q_0$$

$$Q_{2N} = D_2 = Q_1$$



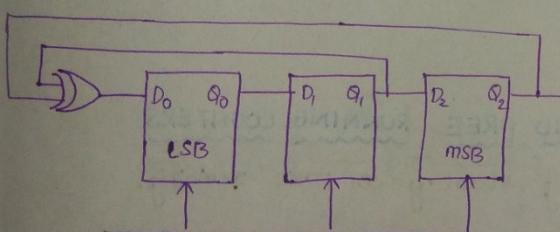
① How many states are there in mod counter = 7.

② Given Initial state = 011 then what happens after 4 clock cycles  $\Rightarrow$  state after 4 clk cycles = 001

③ Initial state = 001  $\Rightarrow$  state after 10 clock cycles = 011

write  $Q_2$  values in all states  $\leftarrow$  ④ Tap the output at  $Q_2 \Rightarrow$  what is the output = 0010111  
 ⑤ o/p is tapped at  $Q_2 \Rightarrow$  (1011100) (1st bit of all states)  
 Count Sequence.

### 30. DERIVING THE CLOCK FREQUENCY



$$T_{FF} = 15\text{ns}, T_{comb} = 5\text{ns}$$

which of the following clock frequency ensures proper counting?

- a) 40MHz b) 60MHz c) 90MHz d) 300MHz

$$T_{clk} \geq T_{FF} + T_{comb} \quad \Rightarrow \quad f_{clk} \leq \frac{1}{20 \times 10^{-9}} \quad (\text{Time and freq are inversely proportional})$$

$$T_{clk} \geq 15\text{ns} + 5\text{ns}$$

$$T_{clk} \geq 20\text{ns}$$

$$f_{clk} \leq 50\text{MHz}$$

RATIONS  
is 50 & 50  
what is going

It will happen!

FB

Flipflop<sub>2</sub>

(76)

first let's analyse the state diagram.

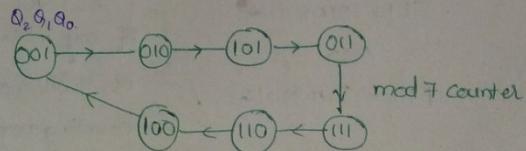
(77)

$Q_2$	$Q_1$	$Q_0$	$Q_{2N}$	$Q_{IN}$	$Q_{ON}$
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	0

$$Q_{ON} = D_0 = Q_1 \oplus Q_2$$

$$Q_{IN} = D_1 = Q_0$$

$$Q_{2N} = D_2 = Q_1$$



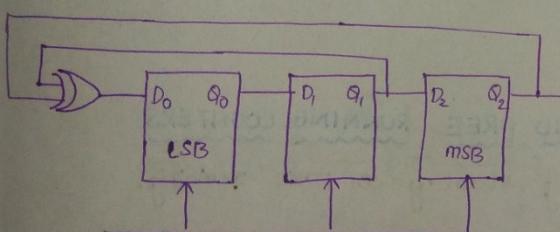
① How many states are there in mod counter = 7.

② Given Initial state = 011 then what happens after 4 clock cycles  $\Rightarrow$  state after 4 clk cycles = 001

③ Initial state = 001  $\Rightarrow$  state after 10 clock cycles = 011

write  $Q_2$  values in all states  $\leftarrow$  ④ Tap the output at  $Q_2 \Rightarrow$  what is the output = 0010111  
 ⑤ o/p is tapped at  $Q_2 \Rightarrow$  (1011100) (1st bit of all states)  
 Count Sequence.

### 30. DERIVING THE CLOCK FREQUENCY



$$T_{FF} = 15\text{ns}, T_{comb} = 5\text{ns}$$

which of the following clock frequency ensures proper counting?

- a) 40MHz b) 60MHz c) 90MHz d) 300MHz

$$T_{clk} \geq T_{FF} + T_{comb} \quad \Rightarrow \quad f_{clk} \leq \frac{1}{20 \times 10^{-9}} \quad (\text{Time and freq are inversely proportional})$$

$$T_{clk} \geq 15\text{ns} + 5\text{ns}$$

$$T_{clk} \geq 20\text{ns}$$

$$f_{clk} \leq 50\text{MHz}$$

RATIONS  
is 50 & 50  
what is going

It will happen!

FB

(78)

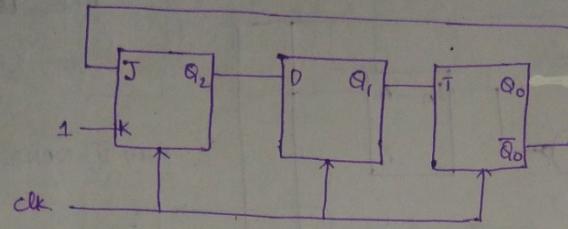
Counting loop

### 33. COUNTER USING 3 DIFFERENT FLIPFLOPS

(79)

Consider the following counter, Initially  $Q_2 Q_1 Q_0 = 000$ 

uses in the

01  
100

= STARTING  
ou are not  
entering Counting  
loop.

state you  
no. of clk  
enter the

running:

$Q_2$	$Q_1$	$Q_0$	$Q_{2N}$	$Q_{1N}$	$Q_{0N}$	$J_0$	$D_1$	$T_2$
0	0	0	1	0	0	1	0	0
0	0	1	0	0	1	0	0	0
0	1	0	1	0	1	1	0	1
0	1	1	0	0	0	0	0	1
1	0	0	0	1	0	1	1	0
1	0	1	0	1	1	0	1	0
1	1	0	0	1	1	1	1	1
1	1	1	0	1	0	0	1	1

$$D_1 = Q_2$$

$$Q_{2N} = ?? = JK$$

J	K	$Q_{2N}$
0	1	0
1	1	$\bar{Q}_N$

$$T_2 = Q_1$$

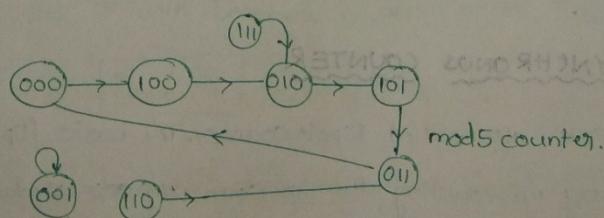
$$Q_{1N} = D_1 = Q_2$$

$$J_0 = Q_0$$

$$Q_{0N} = \bar{Q}_0 (T=0) \Rightarrow Q_0 (Q_1=0)$$

$$\bar{Q}_0 (T=1) \Rightarrow Q_0 (Q_1=1)$$

$\bar{Q}_0$	K	$Q_{2N}$
0	1	0
1	1	$\bar{Q}_N$



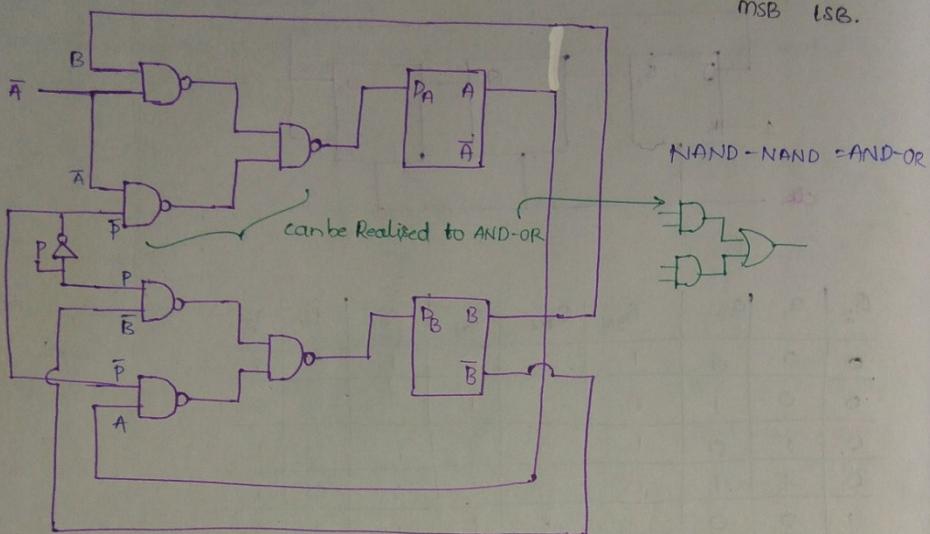
$Q_0$	K	$Q_{2N}$
1	1	0
0	1	$\bar{Q}_N$

- ① what will be the states after 4 clock? a) 000 b) 010 c) 011 d) 101
- ② Modulus of the Counter? a) 4 b) 5 c) 6 d) 7

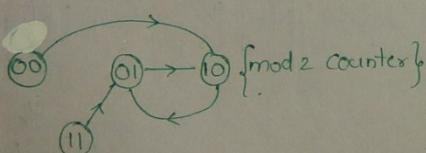
mod3 counter

### 34. EXAMPLES ON COMBINATIONAL CIRCUITS AND FLIPFLOPS

Consider the following counter, If  $p=0$ , the counting sequence of AB is



A	B	$A_n$	$B_n$
0	0	1	0
0	1	1	0
1	0	0	1
1	1	0	1



$A_n = D_A \quad \left\{ \begin{array}{l} \text{If we observe the CC,} \\ B_n = D_B \end{array} \right. \text{they can be Realised}$   
to AND-OR Realisation

$$\begin{aligned} D_A &= \bar{A}\bar{B} + \bar{A}\bar{P} \Rightarrow D_A = \bar{A}\bar{B}\bar{P} \quad \left\{ \begin{array}{l} P=0 \\ \bar{A}\bar{B}(1+P) \end{array} \right. \\ D_B &= P\bar{B} + \bar{P}A \Rightarrow D_B = \bar{A} \end{aligned}$$

$$\Rightarrow \boxed{B_n = \bar{A}} \quad \boxed{A_n = \bar{A}\bar{B}}$$

$Q_0 = 0$

$\Rightarrow 3 \frac{1}{2} \text{ mod } 2 \text{ counters form}$

Now, consider ' $Q_0$ ' it

it T-FF so 'T' will

$\therefore Q_{ON} =$

$\Rightarrow$  Now,  $Q_1$  is charged

$Q_{IN} = \bar{Q}_1$

$\Rightarrow$  Now,  $Q_{2N} = \bar{Q}_2$  ( $Q$ )

### 35. INTRODUCTION TO ASYNCHRONOUS COUNTER

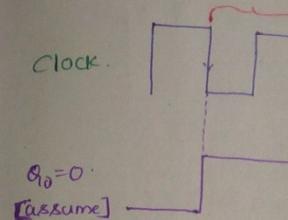
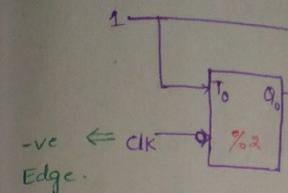
The asynchronous counters are also called Ripple Counter. The basic flip flop is T and basic counting is binary up counting. The up counters are used for implementing incrementation. Due to simpler design, Asynchronous counters are preferred in IC counter fabrication.

$\Rightarrow$  IC 7490 is a decade counter i.e. (%10) counter.

$\Rightarrow$  IC 7492 is a hexadecimal counter i.e. (%16) counter.

$\Rightarrow$  The q/p of one flip flop is going to be clock to the next FF.

### 36. MOD 8 UP COUNTER



$Q_1 = 0$   
(Assume)

$Q_0 = 0$

$\Rightarrow 3 \frac{1}{2} \text{ mod } 2 \text{ counters form}$

Now, consider ' $Q_0$ ' it

it T-FF so 'T' will

$\therefore Q_{ON} =$

$\Rightarrow$  Now,  $Q_1$  is charged

$Q_{IN} = \bar{Q}_1$

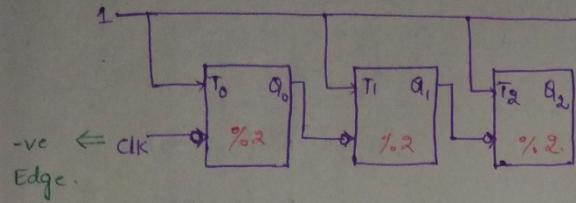
$\Rightarrow$  Now,  $Q_{2N} = \bar{Q}_2$  ( $Q$ )

### 36. MOD 8 UP COUNTER

is

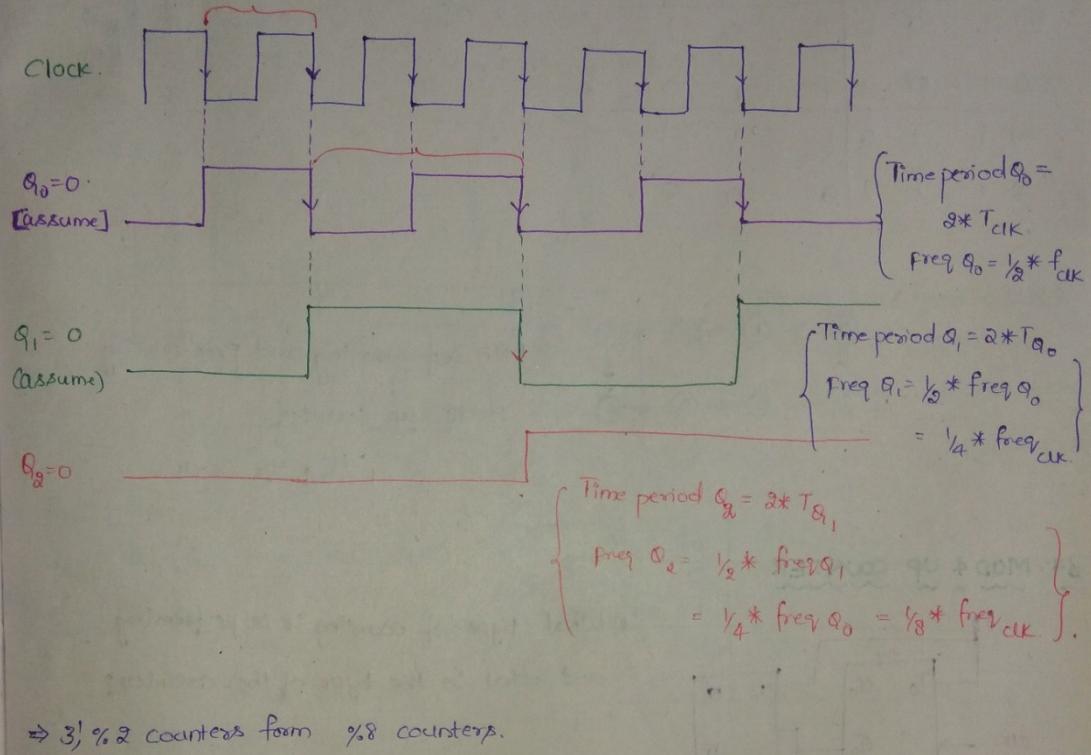
SB.

= AND-OR



Assume  $Q_0 = Q_1 = Q_2 = 0$ . (Initially)

(81)



ve freq,  
Recalised

isation

$$= \overline{A}B + \overline{B} \quad \begin{cases} P=0 \\ = \overline{P} \end{cases}$$

$$= A''$$

$\Rightarrow 3, \%2$  counters form  $\%8$  counters.

Now, consider ' $Q_0$ ' it is having Input 1 and it is -ve edge triggered and it is T-FF so 'T' will become 1 and it will Toggle  $\therefore$  The new dp  $Q_{0N}$  will be  $\overline{Q}_0$ .

$$\therefore Q_{0N} = \overline{Q}_0 \quad \text{for every clock}$$

for every clock

flip flop is T  
ed for  
us one

$\Rightarrow$  Now,  $Q_1$  is changed /  $Q_1$  depends on how  $Q_{0N}$  is providing the clock signal.

$$Q_{1N} = \overline{Q}_1 \quad \left( \text{when there is -ve edge from } Q_0 (1 \rightarrow 0) \right)$$

$\Rightarrow Q_0$  will toggle.

$$\Rightarrow \text{Now, } Q_{2N} = \overline{Q}_2 \quad (Q_1 : 1 \rightarrow 0).$$

$$\begin{aligned} Q_{0N} &= \overline{Q}_0 \\ Q_{1N} &= \overline{Q}_1 \quad (Q_0 : 1 \rightarrow 0) \\ Q_{2N} &= \overline{Q}_2 \quad (Q_1 : 1 \rightarrow 0) \end{aligned}$$

whenever a que like above is given construct the equation and proceed.

Present state	Next state
$Q_2 \ Q_1 \ Q_0$	$Q_{2N} \ Q_{IN} \ Q_{ON}$
0) 0 0 0	0 0 1
1) 0 0 1	0 1 0
2) 0 1 0	0 1 1
3) 0 1 1	1 0 0
4) 1 0 0	0 1 0
5) 1 0 1	1 1 0
6) 1 1 0	1 1 1
7) 1 1 1	0 0 0

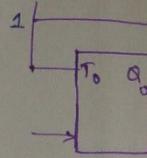
Above equations,

$$Q_{ON} = \overline{Q}_0$$

$$Q_{IN} = \overline{Q}_1 (Q_0 : 1 \rightarrow 0)$$

$$Q_{2N} = \overline{Q}_2 (Q_1 : 1 \rightarrow 0)$$

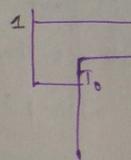
38. MOD 4 DO



$$Q_{ON} = \overline{Q}_0$$

$$Q_{IN} = \overline{Q}_1 (Q_0 : 1 \rightarrow 0)$$

39. MOD 8 RAM



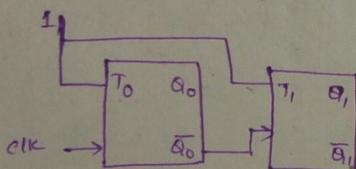
$$Q_{ON} = \overline{Q}_0$$

$$Q_{IN} = \overline{Q}_1 (Q_0 : 1 \rightarrow 0)$$

$$Q_{2N} = \overline{Q}_2 (Q_1 : 1 \rightarrow 0)$$

positive edge  
Symbol ( $\rightarrow$ )

### 37. MOD 4 UP COUNTER

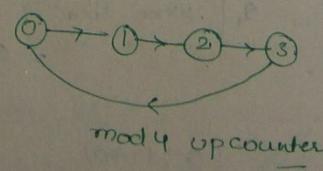


a) What type of counting it is performing and what is the type of the counter?

$$Q_{ON} = \overline{Q}_0 \text{ (for every clock)}$$

$$Q_{IN} = \overline{Q}_1 (\overline{Q}_0 : 0 \rightarrow 1) \\ (\overline{Q}_1 : 1 \rightarrow 0)$$

$Q_1 \ Q_0$	$Q_{IN}$	$Q_{ON}$
0 0	0	1
0 1	1	0
1 0	1	1
1 1	0	0

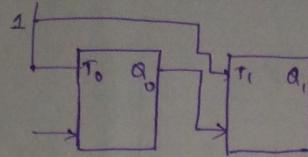


### 40. APPLICATION

1) Shift Registers

2) Counters →

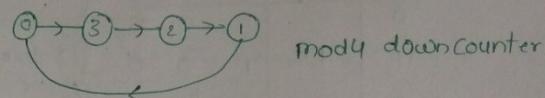
### 38. MOD 4 DOWN COUNTER



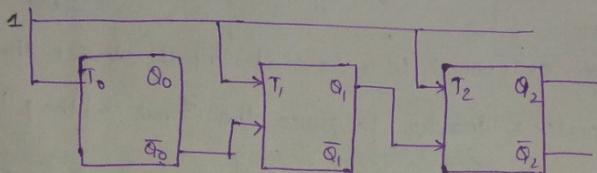
$$Q_{ON} = \overline{Q_0}$$

$$Q_{IN} = \overline{Q_1} \quad (Q_0: 0 \rightarrow 1)$$

$Q_1$	$Q_0$	$Q_{IN}$	$Q_{ON}$
0	0	1	1
0	1	0	0
1	0	0	1
1	1	1	0



### 39. MOD 8 RANDOM COUNTER



$$Q_{ON} = \overline{Q_0}$$

$$Q_{IN} = \overline{Q_1} \quad (Q_0: 0 \rightarrow 1) \Rightarrow (Q_0: 1 \rightarrow 0)$$

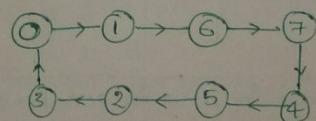
$$Q_{2N} = \overline{Q_2} \quad (Q_1: 0 \rightarrow 1)$$

In diagram it is  
positive edge so  $(0 \rightarrow 1)$   
Symbol ( $\rightarrow$ ) = true edge

$Q_2$	$Q_1$	$Q_0$	$Q_{2N}$	$Q_{IN}$	$Q_{ON}$
0	0	0	0	0	1
0	0	1	1	1	0
0	1	0	0	1	1
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	1	0	0

If the initial state  $Q_2 Q_1 Q_0 = 101$ , what will be the state after 4 clock cycles?

$$= 001 = 1$$



Self starting and  
Free Running.

### 40. APPLICATIONS OF FLIP FLOPS

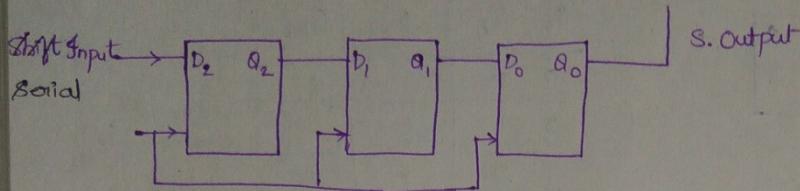
1) Shift Register  $\rightarrow$  used as "Sequential memory". Ex: Accumulator in Microprocessors.

2) Counter  $\rightarrow$  (1) used to count no. of pulses.

(2) used as frequency divider

## 41. 3-BIT SHIFT REGISTERS (It requires 3 D-flipFlops).

(8)



101	clk	SI/P	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>
	0	-	0	0	0
	1	1	1	0	0
	2	0	0	1	0
	3.	1	1	0	1

say I want to push 101 onto shift Register  
 1> push 1  
 2> push 0  
 3> push 1  
 4> 0101  
 1) push  
 2) push  
 3) push

3 clock pulses.  
 Entire for pushing Input on the shift Register.

- ⇒ we have n-bit shift Registers which is a serial Input device then we require 'n' clock pulses in order to place the Input on the FlipFlop
- ⇒ In case of Shift Registers 'n' clk pulses are reqd to place 'n'-bits onto the shift Register

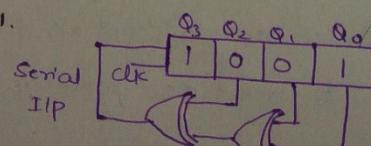
clk	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q
0	1	0	1	1
1	0	1	0	0
2	0	0	1	1

∴ To retrieve the op we need 2 clock pulses (1st one is counted in the pushing upon Register).

- ∴ Totally 3 clk + 2 clk pulses are needed for 3-bit serial Input and serial output.
- ∴ In general for serial Input and serial output we need "2n-1" clock pulses  
 n for serial S/I/P, (n-1) for serial O/P.

## 42. EXAMPLE 1 ON SHIFT RIGHT REGISTER

In the following Right shift Register, determine the no. of clocks required to bring it to the initial state of 1001.



43. EXA

Initial  
following

Inputs:  
 $D_A = 1$   
 $D_B = 1$   
 $D_C = 0$   
 $D_D = 0$

44. BINA

Determining

(84)

<u>CLK.</u>	<u>SIP</u>	States $Q_3 \ Q_2 \ Q_1 \ Q_0$
0	-	1 0 0 1
1	1	1 1 0 0
2	1	1 1 1 0
3	0	0 1 1 1
4	1	1 0 1 1
5	0	0 1 0 1
6	0	0 0 1 0
7	1	1 0 0 1

(85)

$\rightarrow SIP = (Q_2 \oplus Q_1 \oplus Q_0)_{\text{previous clock}} = (0 \oplus 0 \oplus 1) = 1$

$\rightarrow SIP = (Q_2 \oplus Q_1 \oplus Q_0) = (1 \oplus 0 \oplus 0) = 1$

$\rightarrow SIP = 1 \oplus 1 \oplus 0 = 0$

$\rightarrow SIP = 1 \oplus 1 \oplus 1 = 1$

$\rightarrow SIP = 0 \oplus 1 \oplus 1 = 0$

$\rightarrow SIP = 1 \oplus 0 \oplus 1 = 0$

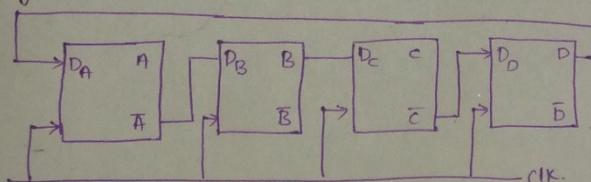
$\rightarrow SIP = 0 \oplus 1 \oplus 0 = 1$

$\therefore 7 \text{ clocks are needed.}$

n-bits

### 43. Example 2 on SHIFT REGISTER

Initial value of ABCD = 0000 then what are the sequence of no's the following circuit is counting.



we need

one is

shifting upon

Input and

clock pulses

Serial o/p.

Inputs:

$$D_A = D$$

$$D_B = \bar{A}$$

$$D_C = B$$

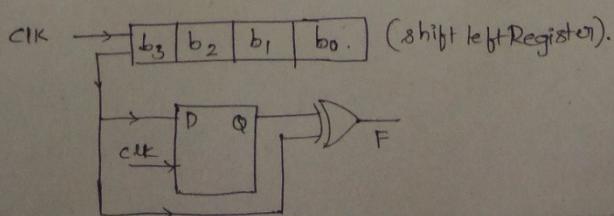
$$D_D = \bar{C}$$

<u>CLK</u>	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	
0	0	0	0	0	0
1	0	1	0	1	5
2	1	1	1	1	15
3	1	0	1	0	10
4	0	0	0	0	0
5	0	1	0	1	5

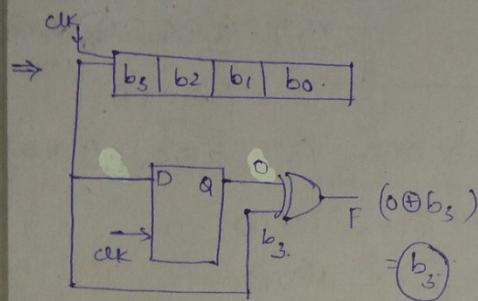
$\therefore 0, 5, 10, 15$

### 44. BINARY TO GRAY CONVERTOR

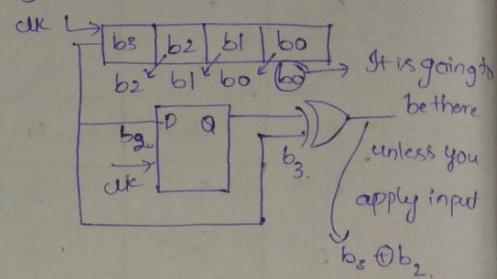
Determine the function of the following circuit



Initially assume the output of D-flipflop is '0'. Now,  $b_3$  is given to the output of D-FF. The output of XOR Gate will be  $0 \oplus b_3 = b_3$



After applying 1 clock:



continuing this procedure,  $b_3, b_3 \oplus b_2, b_2 \oplus b_1, b_1 \oplus b_0 \dots$

$\Rightarrow$  Graycode conversion

























