

1. LOGIC FUNCTIONS

1. BASIC PROPERTIES OF SWITCHING ALGEBRA

Associativity: $(x+y)+z = x+(y+z)$
 $(xy) \cdot z = x(yz)$

Idempotency: $x \cdot x = x \mid x+1 = 1$
 $x+x = x \mid x \cdot 0 = 0$
 $x+0 = x$
 $x \cdot 1 = x$

Commutativity: $x+y = y+x$
 $xy = yx$

Complementation: $x+\bar{x} = 1$
 $x \cdot \bar{x} = 0$

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

$+ = \text{OR}$
 $\cdot = \text{AND}$
 $- = \text{NOT}$

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

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 $(+, \cdot, \text{NOT})$.

Ex: $x + \bar{x}yz + x\bar{z}$, $a+bc+\bar{b}d$

Properties for simplifying SE

Absorption: $x+xy = x = x \cdot 1 + xy$
 $= x(1+y)$
 $= x(1)$
 $= x$

$x+x'y = x+y$

LHS $= (x+x')(x+y)$
 $= 1(x+y)$

$x(x'+y) = xy$

$x \cdot x' + xy$
 $= 0 + xy$

$x+y = \text{RHS}$
 $= xy = \text{RHS}$

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

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

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

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

$= z(x'y' + y + x)$
 $= z((x'+y)(y'+y) + x)$
 $= z(x' + y + x) = z(1+y)$
 $= z$

DE MORGANS LAW AND SIMPLIFICATION

(2)

$$1) (\overline{xy}) = \overline{x} + \overline{y} \quad 2) (\overline{x+y}) = \overline{x} \cdot \overline{y}$$

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

$$f = \overline{x} + \overline{y}$$

$$\overline{f} = x \cdot y$$

$$f_d = \overline{x} \cdot \overline{y}$$

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

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

$$= x + xy + yx + yz + x'y' + x'z'$$

$$= x(1+y) + yx + yz + x'y' + x'z'$$

$$= x + xy + yz + x'y' + x'z'$$

$$= \underbrace{x(1)} + yz + x'y' + x'z'$$

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

$$= \underbrace{(x+y') + yz + x'z'}_1$$

$$= x + z' + y' + yz$$

$$= \underbrace{x + z' + y' + z}_1 = x + y' + 1 = x + y' + 1 = 1$$

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

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

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

ii) 2nd point: $\bar{a}\bar{b}\bar{c}$

$\Rightarrow 000 \Rightarrow \bar{a}\bar{b}\bar{c} = 1$ $\left\{ \begin{array}{l} \bar{a}\bar{b}c \\ 001 \Rightarrow \bar{a}\bar{b}c = 1 \end{array} \right\}$

for the values of 001 the minterm $\bar{a}\bar{b}c$ gives '1'.

iii)

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

$f(a, b, c) \Rightarrow a + bc + ac$

Canonical sum of products should contain all the variables

$f = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc = \sum(1, 2, 4, 6)$ $\rightarrow$ Compact Representation.

Canonical sum of products.

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

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

1 = Complemented form
0 = uncomplemented form.

7. EXAMPLES ON CANONICAL FORMS

(4)

	a	b	c	f ₁	f ₂	f ₃
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

$f_1 = \text{sum of products} = \text{minterms} = \Sigma(3, 4, 6, 7, 8)$
 $\text{product of sum} = \Pi(1, 2, 5)$

$f_2 = \text{sum of products} = \Sigma(1, 3, 5, 7)$
 $\text{product of sum} = \Pi(2, 4, 6, 8)$

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

1) $f(x, y, z) = x'y + z' + xyz$ Convert into Canonical form?

$f(x, y, z) = x'y + z' + xyz$ (sum of products) but not (Canonical sum of products)

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

$$= x'yz + x'y'z' + z'(xy + xy' + x'y + x'y') + xyz$$

$$= x'yz + x'y'z' + z'(xy + xy' + x'y + x'y') + xyz$$

$$= x'yz + x'y'z' + z'xy + xy'z' + x'y'z' + xyz$$

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

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	2	2	2	...	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 $\left\{ \begin{matrix} 1, 2, 3, \dots, n \\ \downarrow \downarrow \downarrow \dots \downarrow \\ 3 \ 3 \ 3 \dots 3 \end{matrix} \right\}$

(3, 4, 6, 7, 8)

⇒ No. of functions = $(2)^{(3^n)}$

(5)

⇒ 'n' k-ary variables, how many m-ary functions possible?

1	2	3	...	n	f
k	k	k	...	k	

⇒ k^n combinations

∴ No. of functions = $m^{(k^n)}$

(ucks)

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

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

$8C_3$ ways

2) Atmost 3 minterms =

$$8C_0 + 8C_1 + 8C_2 + 8C_3$$

3) Atleast 3 minterms

$$\Rightarrow 8C_3 + 8C_4 + 8C_5 + \dots + 8C_8$$

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 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=A	f=B	f=B	f=A⊕B	f=A⊕B
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	1	0	0	1	0	1
1	1	1	0	1	0	1	0

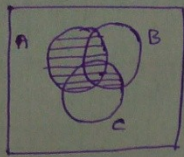
EX-NOR → EX-OR

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

$$\left. \begin{matrix} \text{Neutral functions} = 4 \\ = (2^n)C_{(2^n/2)} = (2^n)C_{(2^{n-1})} \end{matrix} \right\} \begin{matrix} n \text{ variables} = 2^n \text{ combinations} \end{matrix}$$

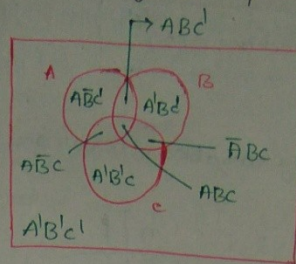
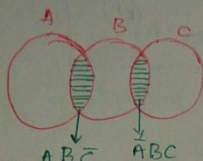
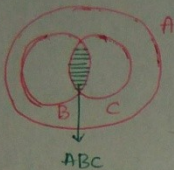
11. VENN DIAGRAM REPRESENTATION

Find the Boolean function that the following Venn diagram represents.



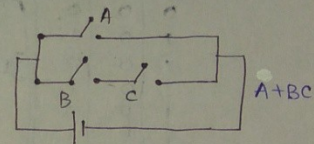
- a) $A + A'B'C$
- b) $A + BC$
- c) $A + A'BC$
- d) $AB + C$

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

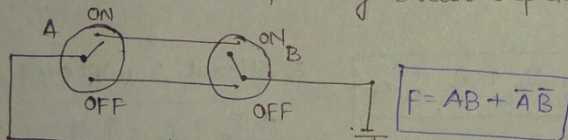


12. CONTACT REPRESENTATION

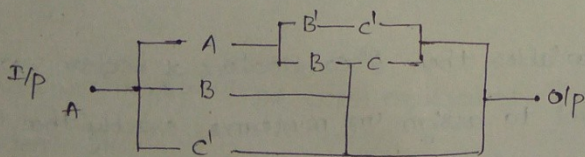
- Every Boolean function can be represented with the help of serial and parallel contact
- Serial contacts are performing "AND" operation
- parallel contacts are performing "OR" operation.



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 case: $AABC + A'ABC + AABA + ABC + ABA + A'AC \rightarrow$ Invalid forward paths.

13. NESTED

In the fol
of w, x, y,

i> $x + y + z$

ii> $xy + w$

iii> $xw +$

$$\textcircled{2} f(A, B) = A$$

Now, f

14. NAND

NAND =

A	B
0	0
0	1
1	0
1	1

15. NOR

NOR =

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) $x+y+z=1$

ii) $xy+wz=0$

iii) $xw+yx=1$

a) $\begin{matrix} w & x & y & z \\ 0 & 0 & 0 & 1 \end{matrix}$ X

b) 1101 X

c) 0101 ✓

d) 1000

② $f(A,B) = A+B$ then find $f(f(x+y,y),z)$

Now $f(x+y,y) = (x+y)' + y$

$= x'y' + y$

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

Now $f(x'+y,z)$

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

$= xy' + z$

14. NAND GATE AND PROPERTIES

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

⇒ NAND gate does not obey Identity

⇒ $A \uparrow A \neq A$

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

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

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

↓
NAND is not Associative

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

15. NOR GATE AND PROPERTIES

NOR = NOT OR

A	B	$A \downarrow B$
0	0	1
0	1	0
1	0	0
1	1	0

⇒ $A \downarrow B = \overline{A+B}$

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

⇒ $(A \downarrow B) = (B \downarrow A)$

⇒ NOR is not Associative

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

16. EX-OR GATE AND PROPERTIES

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

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

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

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

⇒ $A \oplus A = 0$ ($1 \oplus 1 = 0 \neq 1$)

⇒ $A \oplus B = B \oplus A$

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

$\overline{B \oplus C} = \overline{B} \oplus \overline{C} = C \oplus \overline{B}$

17. EXNOR GATE AND PROPERTIES

EXNOR = Exclusive - NOR = Negation of EX OR

A	B	A ⊙ B
0	0	1
0	1	0
1	0	0
1	1	1

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

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

$$\Rightarrow A \odot B = B \odot A$$

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

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

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

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

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

18. PROPERTIES OF EX-OR AND EX-NOR

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

XOR

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

X-NOR

A	B	A ⊙ 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 ⊕ B ⊕ C	A ⊙ B ⊙ 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 $\oplus = 1$ $\odot = 0$

If n is odd:

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

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

$\therefore$ If $n = \text{even}$ both XOR, XNOR are complement

If $n = \text{odd}$

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

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

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

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

$\therefore$ If $n = \text{odd}$ both XOR, XNOR are same

$\therefore$ From the above proofs

i) $A \oplus B = \overline{A \odot B}$

ii) $A \oplus B \oplus C = A \odot B \odot C$

iii) $A \oplus B \oplus C \oplus D = A \odot B \odot C \odot D$

iv) $A \oplus B \oplus C \oplus D \oplus E = A \odot B \odot C \odot D \odot E$

Now, draw

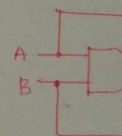
i) $A \oplus B \oplus C$

CD	AB
00	00
01	01
10	10
11	11

ones and
no. of 0's
we can

19. MINIMIZATION

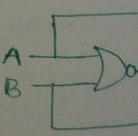
Ex-OR (XOR)



Min no. of

build XNOR

Ex-NOR (XNOR)

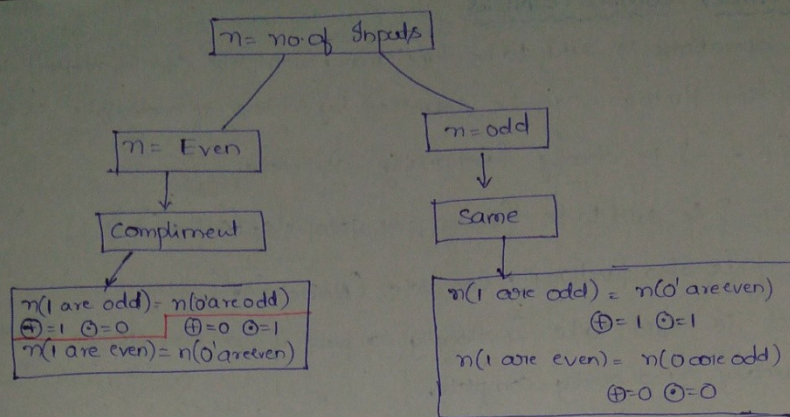


Min. no. of

construct XNOR

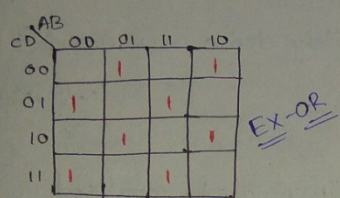
⑧

$\oplus B'$
 $\oplus B = A \oplus B'$
 $\oplus B = A \oplus B'$



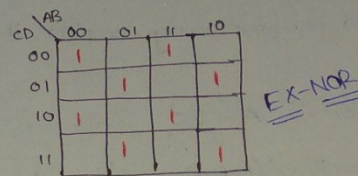
Now, drawing Karnaugh maps.

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



Ones are filled at positions where
 no. of 0's are odd $\therefore$ it is EX-OR
 we cannot minimise them further

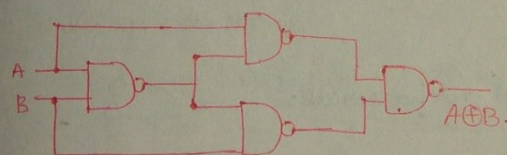
ii) $A \oplus B \oplus C \oplus D$



Ones are filled at positions where
 no. of 0's are even $\therefore$ it is EX-NOR.
 we cannot minimise them further.

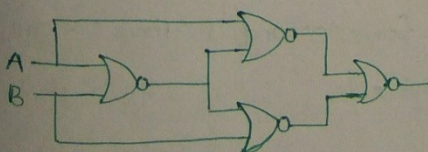
19. MINIMUM NUMBER OF GATES REQUIRED XOR AND XNOR

EX-OR (using NAND)



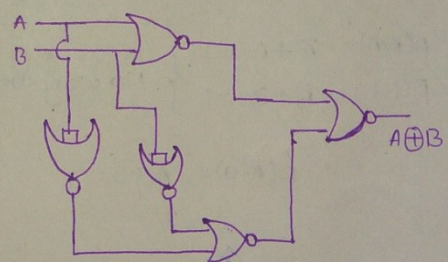
Min no. of NAND gates Reqd for build XOR gate is = 4

EX-NOR (using NOR)



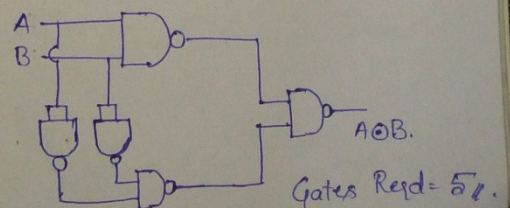
Min. no. of NOR gates required to construct XNOR is 4

EX-OR (using NOR)



Min. no. of NOR gates Reqd to build XOR gate = 5

EX-NOR (using NAND)



Gates Reqd = 5

20. FUNCTIONALLY COMPLETENESS

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

→ The set $\{\cdot, -\}$ is also functionally complete ($A + B = \overline{(\overline{A} \cdot \overline{B})}$)

→ NOTE: A set is said to be functionally complete if we can derive a set which is already functionally complete.

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.

$$\Rightarrow f(\underbrace{f(A, A, A)}_{A'}, \underbrace{B, f(B, B, B)}_{B'}) \Rightarrow f(A', B, B') = (A')' + B(B')'$$

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

22. EXAMPLE 2 ON FUNCTIONAL COMPLETENESS

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

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

$$F(A, A) = \overline{A} + A = 1$$

$$F(B, B) = \overline{B} + B = 1$$

} No way we can get rid of a variable

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

$$= \overline{A}$$

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

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

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

(10)

23. EXAMPLE 3, 4, 5, 6 ON FUNCTIONAL COMPLETENESS

(11)

if and only if

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

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

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

$$f(A, 1) = \bar{A} \quad (\text{complementation achieved}).$$

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

} partially functionally complete.

which

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

$$f(A, A, A) = A + A + A$$

$$f(A, A, A) = A$$

} complementation cannot be achieved (Not functionally complete)

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

r.)

-, +)

$$3) f(x, y) = \bar{x}y + x\bar{y}$$

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

$$= 0 + 0 = 0.$$

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

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

I'm getting complement with '1'

$\Rightarrow$ partially functionally complete

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

$$= 0 + 0 = 0.$$

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

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

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

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

cannot be achieved.

$\therefore$ Not functionally complete.

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

a) $\oplus$, NOT

b) $\oplus$, 1, OR

c) $\oplus$, 1, NOT

d) $\oplus$, 1, NOT

$\oplus$ does not lead to { $+, \cdot$ }. Check above example

partially complete
 $\downarrow$
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) $f(A, B, C) = AB(C+C') + (A+A')BC + (B+B')A$
 $= ABC + ABC' + A'BC + AB'C$

A 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.

$(ABC) \rightarrow (A'B'C')$
 $(A'B'C') \Rightarrow (ABC)$ } mutually Exclusive

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.

$\therefore$ No. of self dual functions = $2^{(2^n-1)}$

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

↓ divide into pairs (having two in each set)

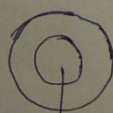
$2^n/2 = 2^{n-1}$ pairs

↓

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

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

$\therefore$ No. of self dual functions = $2^{(2^n-1)}$



Neutral functions

Self Dual functions

29. SELF DUAL FUNCTIONS ARE CLOSED UNDER COMPLEMENTATION

Which of the following functions are closed under complementation?

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

2) $f(A, B, C) = \Sigma(0, 1, 6, 7) = (0, 7)(1, 6)(2, 5) \Rightarrow$ Neutral but not self dual (contains both ele of mutual exclusive pair)

3) $f(A, B, C) = \Sigma(0, 1, 2, 4) = (0, 7)(1, 6)(2, 5)(3, 4) \Rightarrow \text{No. of minterms} = \text{No. of maxterms} \& \text{ME pairs.}$
 also unique (SD) ✓

4) $f(A, B, C) = \Sigma(3, 5, 6, 7) = (3, 4)(5, 2)(6, 1)(0, 7) \Rightarrow \text{ME (✓)} \quad N(\min) = N(\max) = 3 \text{ ✓}$

$\Rightarrow$ self dual is closed under complementation (Compliment of self dual function is self dual)

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

$$\begin{aligned}\text{Ex: } f(x, y, z) &= xy + yz + zx \\ &= x'yz + xy'z + xyz' + xyz\end{aligned}$$

→ 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

$$\begin{aligned}F(x, y, z) &= x'yz' + x'y'z' + xy'z' + x'yz + xyz + xy'z \\ &= x'z'(y+y') + xy'z' + yz(x+x') + xy'z \\ &= x'z' + xy'z' + yz + xy'z \\ &= (x' + xy')z' + (y + xy')z \\ &= ((x' + x)(x' + y))z' + ((y + x)(y + y'))z \\ \boxed{F(x, y, z) = x'z' + y'z' + yz + xz}\end{aligned}$$

⇒ 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-map is a modified form of the truth table in which the arrangement of combinations is particularly convenient for minimising (15)

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

$x \backslash y$	0	1
0	0	1
1	2	3

2V-K-map.

$xy \backslash z$	00	01	11	10
0	0	2	6	4
1	1	3	7	5

3V-Kmap

$wxyz \backslash yz$	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 subcube is as long as possible.

5. EXAMPLES ON K-MAP

Minimise $f(w,x,y,z) = \Sigma(0,4,6,7,8,9,15)$

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

$wxyz \backslash yz$	00	01	11	10
00	1	1		1
01				1
11	1	1		
10	1	1		

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

$wxyz \backslash yz$	00	01	11	10
00			1	
01	1	1	1	
11		1	1	
10	1			

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

Not Required.

6. COVERING FUNCTIONS

(16)

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.

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

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

$\Rightarrow$ NO. of min terms possible = $2^4 = 16$ minterms

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

$\Rightarrow$ NO. of covering functions = $2^9 = 2^9 = 512$

wx

00
01
10
11

 $+$

00
01
10
11

 yz

= NO. of minterms = 7

NO. of remaining terms = $16 - 7 = 9$.

7. IMPLICANTS AND PRIME IMPLICANTS

$\rightarrow$ 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

$\Rightarrow$ 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. ESSEN

A prime

if it cover

Implicants

9. PROC

1) Dete

2) Remo

prime

3) If the

minim

the

prime

Ex

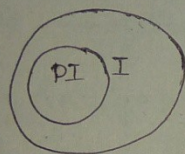
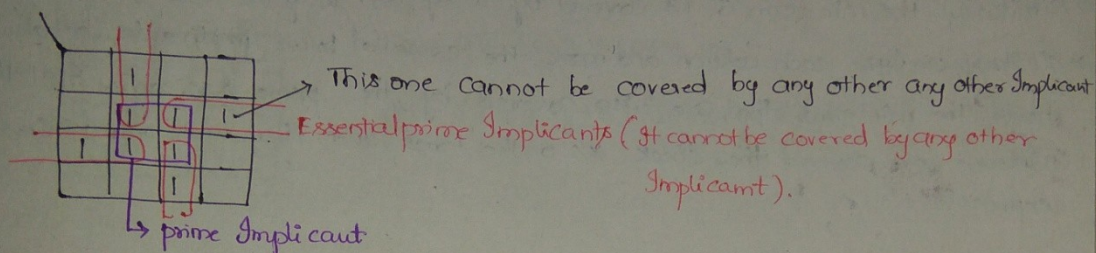
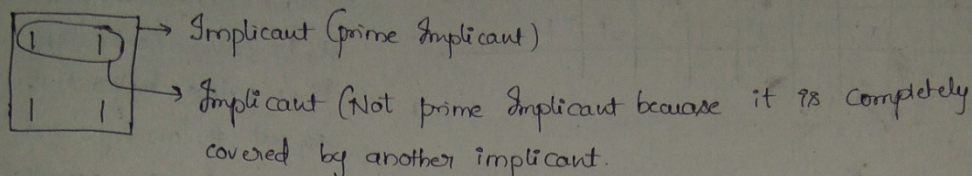
yz

w
00
01
11
10

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.



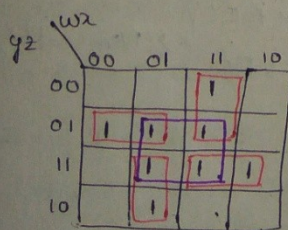
I = Implicants

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



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

$$f = \bar{w}yz + wx\bar{y} + wyz + \bar{w}xy \text{ (Unique minimal expression)}$$

⇒ 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	15
* $\bar{w}yz$	1	1						
* wxy						1	1	
* wyz					1			1
* $\bar{w}xy$			1	1				
xz		1		1			1	1

2.11.18 ⇒ Now, $\bar{w}yz$ covers the boxes 1, 5 in K-map. So put 1, 1 in 1, 5 column and $\bar{w}yz$ row, and mark 1, 5 (means they are covered) ⇒ similarly do for all minterms. Implicants (Rows). (column number) 1. ⇒ Now check 1 minterm (1 column) it is covered only with $\bar{w}yz$ so it is a prime implicant that is essential so put * on $\bar{w}yz$ and it also covers 5

⇒ Similarly the min term '5' is covered by $\bar{w}xy$ so $\bar{w}xy$ 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?

$\bar{A}\bar{B}$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

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

- $B\bar{C}, A'\bar{B}$
- $A'\bar{C}, A'\bar{B}$
- $A'\bar{B}, AB'$ ✓
- $A'\bar{B}, AB', B\bar{C}$

Now, if they have asked how many minimal expressions are there then how to find is.

	1	2	3	4	5
* $A'\bar{B}$		1	1		
* AB'				1	1
$A'\bar{C}$	1		1		
$B\bar{C}$	1				1

Equally probable candidates.

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

11. MINIMAL POS

Continued in Nextpage

12. Examples

① $F(w, x, y, z)$

yz	00	01
00	1	1
01	1	1
11	1	1
10	1	1

(w+x)

②

yz	00	01
00	1	1
01	1	0
11	1	1
10	1	0

13. INTRODUCT

Find the mo

$f(w, x, y, z)$

yz	00	01
00	1	1
01	0	0
11	1	0
10	1	0

13. INTRODUCT

→ A function is of variables.

→ Combinations

Core combi

→ The value of

→ Since each d

specified fu

of 2^k dis

→ our task is t

→ we could as

a muto in

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

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

Annotations: $(y+z)$ for columns 01 and 11; $(\bar{y}+\bar{z})$ for columns 00 and 10; $(w+x)$ for rows 01 and 11; $(\bar{w}+\bar{x})$ for rows 00 and 10.

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

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

②

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

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

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

12. EXAMPLES ON MINIMAL POS.

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

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

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

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

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

13. INTRODUCTION TO DONT CARES

→ A function is said to be completely specified if it is given '0' or '1' for every combination of variables. There exist some function 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 'φ' or 'd'.

→ Since each don't care combination represents 2 values for 1, an incompletely specified function containing k-dont 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

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

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

$$\text{Now, Sop} = \bar{w}x wy + xy + z\bar{w}x$$

$$\text{Sop} = w + xy + z\bar{w}x$$

15. EXAMPLES ON DONT CARE SET-2

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

$= CD + \bar{B}D$

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

$= \bar{y}z + xy$

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

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

16. EXAMPLES ON DONT CARE SET3

xy \ wz	00	01	11	10
00	0	X	0	0
01	0	X	1	1
11	1	1	1	1
10	0	X	0	0

$\Rightarrow wy + xy = y(x+w)$

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

Options:

- (A) $(w+x)y$
- (B) $xy + yw$
- (C) $(w+x)(\bar{w}+y)(\bar{x}+y)$
- (D) None

17. FINDING MINIMAL EXPRESSIONS

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

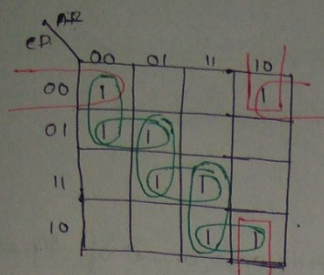
How many minimal expressions are possible?

	0	1	2	5	7	8	9	10	13	15
*A	1		1			1		1		
*B				1	1					1
C		1		1				1		1
D	1	1				1	1			

$\therefore$ NO. of minimal expressions = 4

18. BRANCHING TECHNIQUE FOR MINIMISING CYCLIC FUNCTIONS

(21)



	0	1	5	7	8	10	14	15
$A = wx'y'$	X	X						
$B = wy'z'$			X					
$C = w'xz$			X	X				
$D = xyz$				X				X
$E = wx'y$						X	X	X
$F = wy'z'$						X	X	
$G = wx'z'$				X		X		
$H = x'y'z'$	X			X				

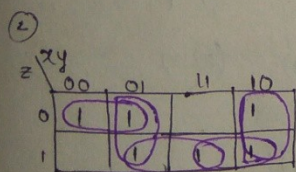
⇒ Now, if i choose 'A' and include it in minimal expression then A' (included in ME)

⇒ 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 ⇒ B is irredundant) and $G \rightarrow H$

⇒ (G is covered by H ⇒ G 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	2	3	4	5	7
$A = x'z'$	X	X				
$B = x'y$		X	X			
$C = yz$			X			X
$D = xz$				X	X	
$E = xy'$				X	X	
$F = y'z'$	X			X		

⇒ Let me choose

$$A + C + E$$

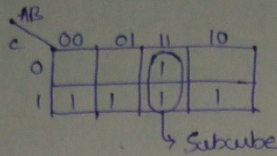
$B \rightarrow C \Rightarrow B$ waste $F \rightarrow E \Rightarrow F$ is not needed.

19. IMPLICANT AND PRIME IMPLICANT DIFFERENCE

(22)

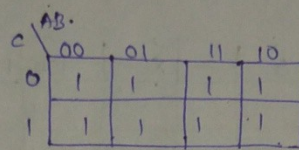
Which of the following can be a prime implicant of a function $f(a,b,c)$

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



Which of the following cannot represent prime implicants, onset of $f(A,B,C)$

- a) $\{ab, bc, ca\}$
- b) $\{bcd, ba, bc\}$
- c) $\{a', b'ac\}$
- d) $\{a'b', ab\}$



$\{ab', bc', ca'\}$
 $\downarrow$
 $\$fab$ 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

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

$= A'BC + ACD + BD$. $\Rightarrow$ Now the Mutually exclusive terms for this are
 $A'B'C' + A'C'D' + B'D'$ these minterms should not be added.

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

The left out pairs are: $(1,14)$ $(3,12)$
 $\downarrow$ $\downarrow$
 1 min term from this $\downarrow$ + this
 $\Rightarrow$ 2 choices $\downarrow$ 2 choices
 $=$ 4 minterms can be added

21. COMB

How many

$$f_1(a,b,c)$$

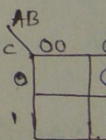
$$f_2(a,b,c)$$

Ans

A	f ₁
000 = 0	1
001 = 1	0
010 = 2	1
011 = 3	0
100 = 4	1
101 = 5	0
110 = 6	0
111 = 7	0

22. PRIME

What are
 prime



A function

23. NUMBER

In a k-n
 covering a
 by 3- n

21. COMBINING FUNCTIONS HAVING DONT CARE

(23)

How many functions does f_1, f_2 and $f_1 + f_2$ Represent?

$$f_1(A, B, C) = \sum(0, 2, 4) + \phi(3, 5, 7) = \sum(0, 2, 4) + \sum_{\phi}(3, 5, 7)$$

$$f_2(A, B, C) = \sum(2, 3) + \sum_{\phi}(1, 6, 7)$$

Ans

A	f_1	f_2	$f_1 \cdot f_2$	$f_1 + f_2$
000 = 0	1	0	0	1
001 = 1	0	ϕ	0	ϕ
010 = 2	1	1	1	1
011 = 3	ϕ	1	ϕ	1
100 = 4	1	0	0	1
101 = 5	ϕ	0	0	ϕ
110 = 6	0	ϕ	0	ϕ
111 = 7	ϕ	ϕ	ϕ	ϕ

4 Dont cares $\Rightarrow f_1 + f_2 = 2^4$ functions

2 Dont cares (ϕ, ϕ)

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

$$= 2 \times 2$$

$$= 4$$

$$\therefore f_1 \cdot f_2 = 4 \text{ functions}$$

22. PRIME IMPLICANTS AND DONT CARES

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

AB	00	01	11	10
0		1	1	
1			1	1

No. of Implicants = 3.

Essential = 2

Redundant = 1

A function $f(A, B, C) = \sum(3, 5, 6)$ is minimised to $(A + Bc)$ then what are the dont cares?

AB	00	01	11	10
0			1	ϕ
1		1	ϕ	1

$$= \sum_{\phi}(4, 7) \text{ are the dont cares.}$$

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 min terms are in turn covered by 3 non-essential prime implicants each. what is the no. of minimal sop expressions?

The soln is $EPI + \downarrow + \rightarrow$
3 choices 2 choices

$\therefore$ 9 minimal expressions are possible

24. BEAUTIFUL QUESTION ON PRIME IMPLICANT CHART

$\rightarrow$ In a prime implicant chart representing boolean expression $f(w,x,y,z)$ columns represent minterms and rows represent P.I.s. Identify P, Q, R, S and a, b?

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

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

wz	00	01	11	10
yz	00 ①	01 ①	11 ①	10 ①
01		①	②	
11		①	③	
10				

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 'S' 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.

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

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

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

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

26. MINIMIS

- Get all the
- (a) Make one minterm (1)
- (b) Multiply
- Repeat the
- Sop of VE

Ex:

AB	00	01	11	10
C	0 D	0 1	1 D	1 0

Step-1: Replace

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

Step-2: Replace

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

Step-3: Now

and 1/2

Step-4: A/B

$$f = A/B$$

27. EXAMPLE

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

AB	00	01	11	10
C	0 P	0 S	1 Q	1 Q

26

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 earlier minterms (1's) as dont cares
- (b) Multiply the above 'sop' with the concerned variable
- 3) Repeat the step 2) untill all the variables in the cell are covered
- 4) Sop of VEM is obtained by OR ing the previous sop expressions.

Ex:

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

→ 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	00	01	11	10
C	0	0	1	0	0
	1	0	1	0	ϕ

Sop = $A'B$.

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

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

Sop = $\bar{A}1$.

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

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

= AC'

Step-3: $A'B + D + AC'D'$

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

27. Example ON VEM

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

	P	Q
R	P	S
S	R	Q

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 $f(A,B,C) = \Sigma(3,5,6,7)$

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,

A \ B	0	1
0	0	0
1	1	1

$$P=0$$

$$R=C$$

$$S=C$$

$$Q=1$$

28. PROBLEM ON K-MAP

RS PQ

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

10- prime implicants.

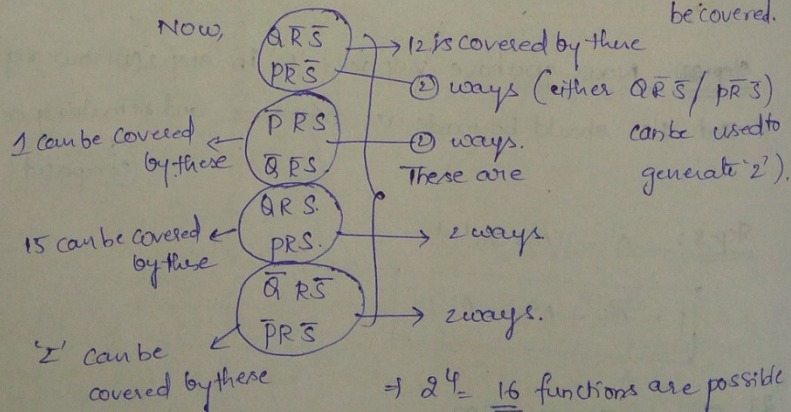
12- Minterms.

How many Minimal functions are possible?

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

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

if $\Rightarrow PQ$ included then (8,9,10,11) will be covered.



29. FIND

How many

K-map

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

30. RELAT

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

31. RELAT

Let $t = (0$

minimal

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

$$F = \Sigma$$

$$F = \Sigma$$

$\therefore F$ and

32. Comp

Minimal

$$a) + \infty$$

29. FINDING FREE VARIABLES

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

$yz \backslash wx$	00	01	11	10
00		1	1	
01	1			1
11	1			1
10		1	1	

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

$$F = \bar{x}z + x\bar{z} = \text{sop} \therefore (w, y) \text{ are not Required.}$$

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

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

--1

$yz \backslash wx$	00	01	11	10
00		$\emptyset$	$\emptyset$	
01			1	1
11	1	1	1	1
10				

$$F = (yz + wz)$$

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.

$yz \backslash wx$	00	01	11	10
00				$\emptyset$
01	1	1	1	$\emptyset$
11	1	$\emptyset$	$\emptyset$	$\emptyset$
10	1	$\emptyset$		

$$F = z + yw$$

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

$yz \backslash wx$	00	01	11	10
00	0	0	0	$\emptyset$
01	1	1	1	$\emptyset$
11	1	$\emptyset$	$\emptyset$	$\emptyset$
10	1	$\emptyset$	0	0

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

There are 2⁶ functions are possible (6 Dont care and each Dont care has 2 choices $\therefore 2^6$ functions are possible).

$\therefore F$ and G are not identical always.

32. COMPARING INDEPENDENT VARIABLES IN MINIMAL SOP AND POS.

Minimal expression represented by K-map is free from.

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

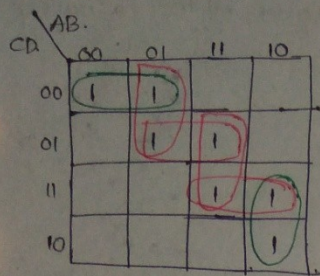
$cd \backslash ab$	00	01	11	10
00	1	0	0	0
01	1	0	0	0
11	1	1	1	1
10	1	1	1	1

$$c + A'B' + A'C'D$$

$\therefore$ Dependent on all.

33. NO-OF IRREDUNDANT AND MINIMAL EXPRESSION

28



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
* $P = \bar{A}\bar{C}\bar{D}$	✓	✓					
$Q = \bar{A}B\bar{C}$		✓	✓				
$R = \bar{B}\bar{C}D$			✓	✓			
$S = ABD$				✓	✓		
$T = ACD$					✓	✓	
* $V = A\bar{B}C$						✓	✓

Q is dominated by R

T is dominated by S

$(Q \rightarrow R)$
Waste
 $(T \rightarrow S)$

$P, V + R + S$

∴ Minimal pos = $P + V + R + S$

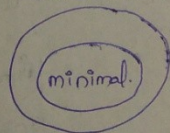
5) No. of Minimal sop's possible are,

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

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

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

Irredundant



$$\therefore (Q+T)(R+S)(S+T)$$

$$= QRS + QRT + QSS + QST + RRS + RRT + RSS + RST$$

$$= \underline{QRS} + QRT + \underline{QS} + \underline{QST} + \underline{RT} + \underline{RS} + RST$$

$$= QS + RT + RS$$

$$\therefore \left. \begin{array}{l} P + V + Q + S \\ P + V + R + T \\ P + V + R + S \end{array} \right\}$$

No. of mE = No. of Irredundant mE = 3 (In this case)

⇒ 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 $f_1 \$ f_2$

	AB	00	01	11	10
0		0	1	0	1
1		0	1	1	0

	AB	00	01	11	10
0		1	0	0	1
1		0	1	1	1

Now, $f_1 \$ f_2 = f_1' + f_2$

$f_1 = 0 \Rightarrow f_1' = 1 + f_2 = 1 + 1 = 1$

	AB	00	01	11	10
0		1	0	1	1
1		1	1	1	1

$f_1 = 0 \Rightarrow f_1' = 1 + 0$
 $f_1 \$ f_2 = 1$

(Red)
 e that
 waste
 (T → S.)

38. FUNCTIONS INVOLVING FUNCTIONS - Example 3

Consider three three-variable functions $f_1(p, q, r) = \Sigma(0, 1, 3)$, $f_2(p, q, r) = \Sigma(3, 5, 7)$, $f_3(p, q, r) = \Sigma(1, 3, 7)$. These functions are sharing 4 prime implicants AB, CD (all of them are pairs) as shown.

A	*	*	*
B	*	*	*
C	*	*	*
D	*	*	*
	f_1	f_2	f_3

find ABCD?

Now, f_1

	pq	00	01	11	10
0		1	1	0	0
1		1	0	1	0

$f_1 = p'q' + R\bar{p}$

f_2

	pq	00	01	11	10
0		0	0	1	0
1		0	1	1	1

$f_2 = QR + PR$

f_3

	pq	00	01	11	10
0		0	1	1	0
1		0	1	0	1

$f_3 = p'R + QR$

$\bar{Q}\bar{P} = A$	*	*	*
$QR = B$	*	*	*
$p'R = C$	*	*	*
$PR = D$	*	*	*
	f_1	f_2	f_3

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

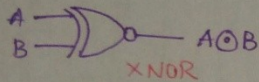
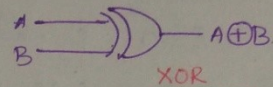
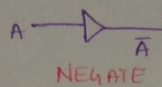
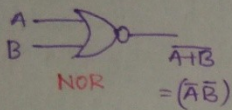
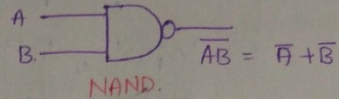
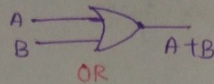
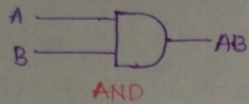
3. DESIGN AND SYNTHESIS OF COMBINATIONAL CIRCUITS

30

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.



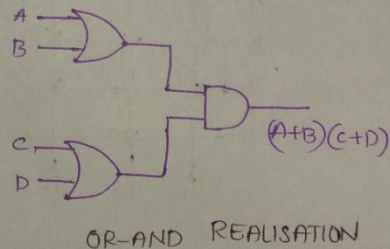
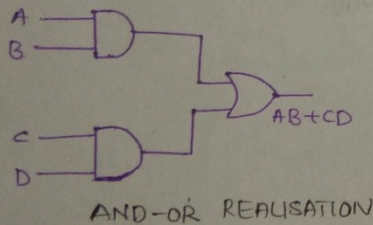
A	B	A ⊙ B
0	0	1
0	1	0
1	0	0
1	1	1

A	B	A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

2. AND-OR OR-AND REALISATION

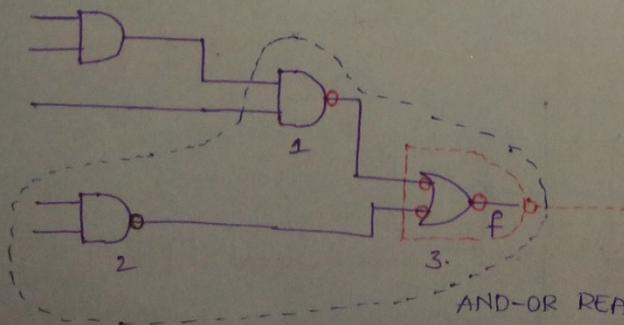
SOP: $f(A, B, C, D) = AB + CD$

POS: $f(A, B, C, D) = (A+B)(C+D)$



3. MINIMUM NO OF NAND GATES EXAMPLE

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

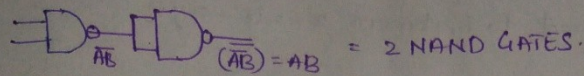


AND-OR REALISATION. TO MAKE IT NAND GATE JUST PLACE BUBBLE AS SHOWN

5. MINIMUM

Find the

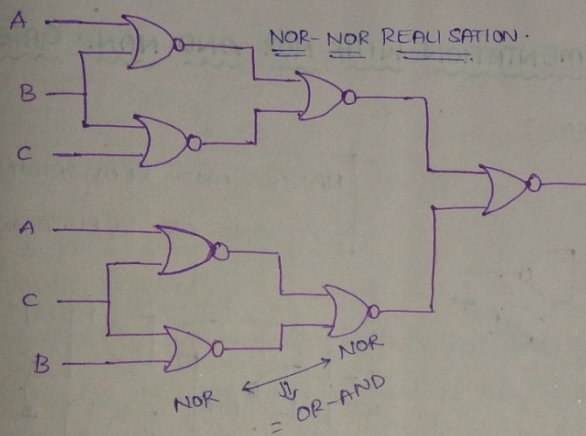
$$\text{NAND, } AB = \overline{(\overline{AB})}$$



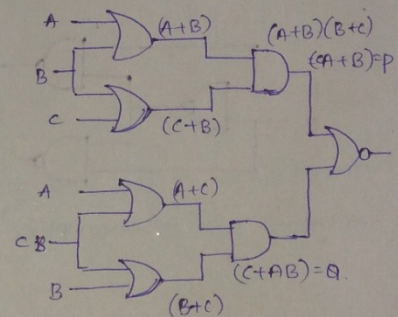
∴ Total no. of NAND GATES Required = 5.

4. NOR-NOR Example

What does the following function Represent?



NOR-NOR REALISATION =
OR-AND REALISATION



⇒ Now, output will be $\overline{P+Q}$
 $= \overline{P} \overline{Q}$

$$\therefore \overline{PQ} = \overline{(B+AC)} \overline{(C+AB)}$$

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

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

$$\overline{PQ} = \overline{B} \underbrace{[\overline{A+C}]}_{\overline{A} + \overline{C}} \overline{C} (\overline{A+B}) = (\overline{A} \overline{C} + \overline{C}) (\overline{A} \overline{B} + \overline{B})$$

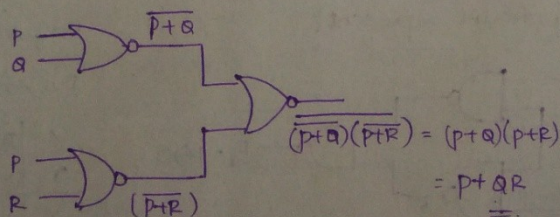
$$\boxed{\overline{PQ} = \overline{C} + \overline{B}}$$

5. MINIMUM NO-OF NOR GATES EXAMPLE

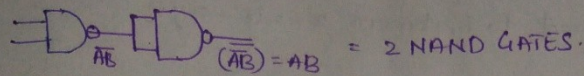
Find the min no of 2 i/p NOR gates required to represent $F(p,q,r) = p+q+r$

$$F(p,q,r) = (p+q)(p+r)$$

$$= \overline{(\overline{p+q})} \overline{(\overline{p+r})}$$



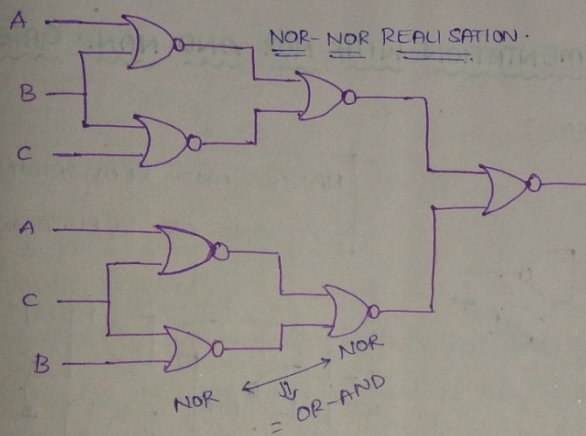
$$\text{NAND, } AB = \overline{(\overline{AB})}$$



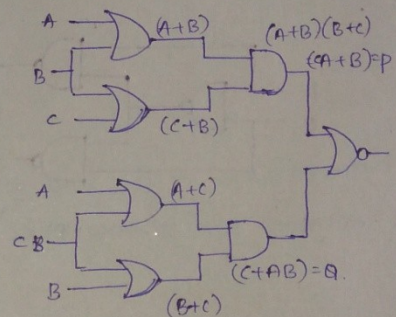
∴ Total no. of NAND GATES Required = 5.

4. NOR-NOR Example

What does the following function Represent?



NOR-NOR REALISATION =
OR-AND REALISATION



⇒ Now, output will be $\overline{P+Q}$
 $= \overline{P} \overline{Q}$

$$\therefore \overline{PQ} = \overline{(B+AC)} \overline{(C+AB)}$$

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

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

$$\overline{PQ} = \overline{B} \underbrace{[\overline{A+C}]}_{\overline{A} + \overline{C}} \underbrace{[\overline{C+A+B}]]}_{\overline{A} + \overline{B}} = (\overline{A} \overline{C} + \overline{C})(\overline{A} \overline{B} + \overline{B})$$

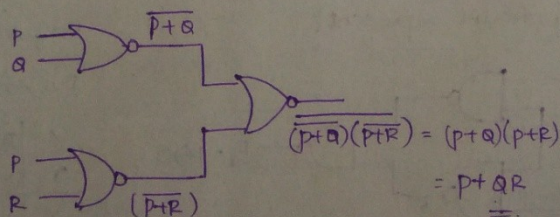
$$\boxed{\overline{PQ} = \overline{C} + \overline{B}}$$

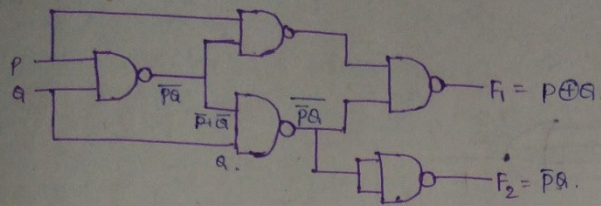
5. MINIMUM NO-OF NOR GATES EXAMPLE

Find the min no of 2 i/p NOR gates required to represent $F(p, q, r) = p + q + r$

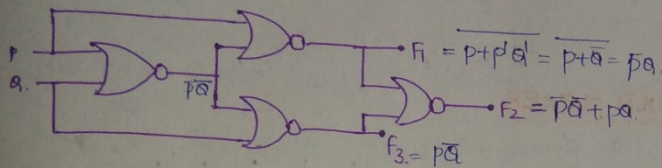
$$F(p, q, r) = (p+q)(p+r)$$

$$= \overline{(\overline{p+q})} \overline{(\overline{p+r})}$$



9. HALF SUBTRACTOR

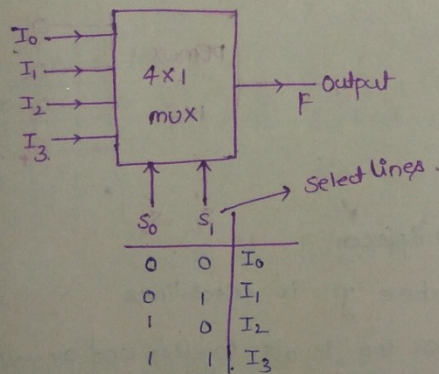
P	Q	F ₁	F ₂
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

10. COMPARATOR

P	Q	F ₂	F ₁	F ₃
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

11. MULTIPLEXER

- A MUX is an electronic switch that can connect one out of 'n' 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.



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

If S_0, S_1 are 0,0 then o/p will be I_0

S_0, S_1 are (0,1) then o/p will be I_1

S_0, S_1 are (1,0) then o/p will be I_2

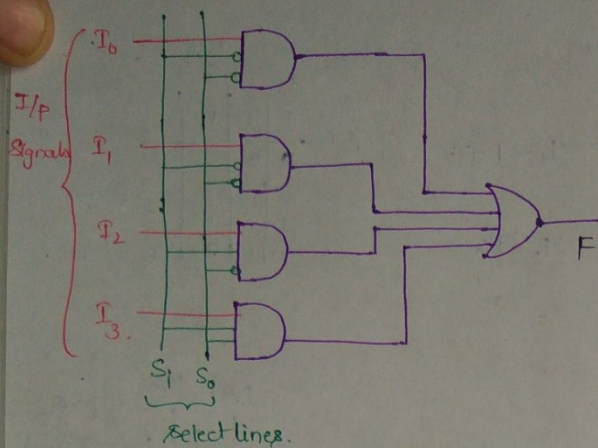
S_0, S_1 are (1,1) then o/p will be I_3

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

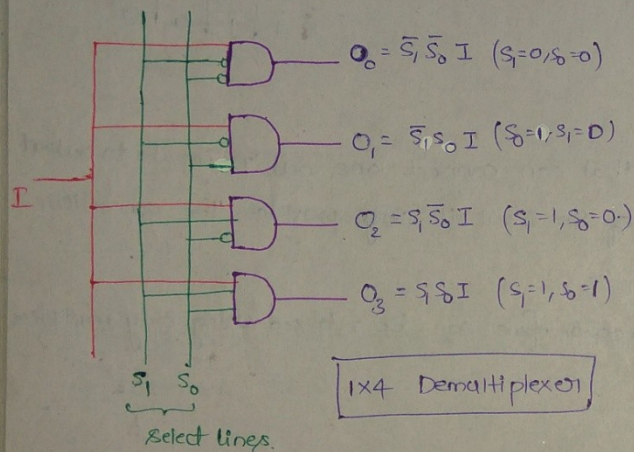
⇒ If we have $2^n \times 1$ Mux then we have 'n' select lines, and 'n' bits are uniquely required to represent each input signal

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

AND-OR REALISATION.

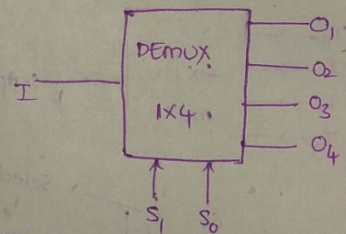


22. INTRODUCTION TO DE MULTIPLEXER

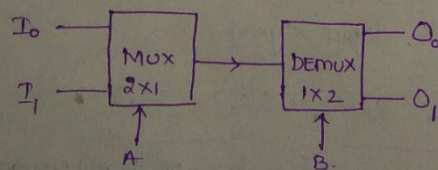


S_1	S_0	O_0	O_1	O_2	O_3
0	0	I	0	0	0
0	1	0	I	0	0
1	0	0	0	I	0
1	1	0	0	0	I

⇒ 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	$(I_0 - O_0)$
1	1	$(I_1 - O_1)$
0	1	$(I_0 - O_1)$
1	0	$(I_1 - O_0)$

Straight connection: $(I_0 - O_0)$ and $(I_1 - O_1)$
 Cross connection: $(I_0 - O_1)$ and $(I_1 - O_0)$

12. PRO

To say

Now,

Now, +

No

⇒ Now

$F = AB$

$F =$

⇒ Now,

13. Imple

I_0

I_1

I_2

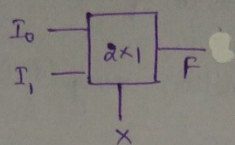
I_3

MSB

12. PROVING MUX IS FUNCTIONALLY COMPLETE

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

Now, let us consider 2×1 Multiplexer.



output func: $F = \bar{X} I_0 + X I_1$

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

when $X=0$, I_0 selected
when $X=1$, I_1 selected

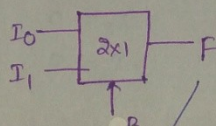
Now, the select line 'X' can be '0' or '1' then if $X=0$, then I_0 will be selected

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 Complement manner)

⇒ Now,

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



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

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

$I_0=0$, $I_1=A$ then AND can be realised

⇒ Now,

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

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

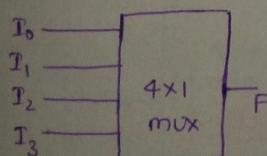
$A+B = \bar{B}A + B(A+A)$

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

∴ OR can be Realised.

OR GATE

13. IMPLEMENTING FUNCTIONS WITH MUX EXAMPLE 1



$F = \bar{A}\bar{B} I_0 + \bar{A}B I_1 + A\bar{B} I_2 + AB I_3$

Now, I want to Implement function 'g' say

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

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

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

$\downarrow$

$\downarrow$

$\downarrow$

$\downarrow$

I_0

I_1

I_2

I_3

∴ Mux can implement 'g'

Now, using 4×1 multiplexer i can implement a function of 2 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 (4×1) 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

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

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

Now, (4×1) Mux equation $F = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + A\bar{B}I_2 + ABI_3$

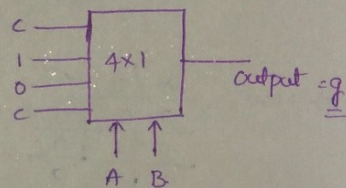
comparing f & g

$$I_0 = C$$

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

$$I_2 = 0$$

$$I_3 = C$$



$\therefore$ 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 (4×1) 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

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

Select lines

function requires AND GATE

14. IMPLEMENTING FUNCTIONS WITH MUX - EXAMPLE-2

Implement the following function $f(A, B, C) = \sum(1, 2, 4, 6, 7)$ using 4×1 Mux?

$f(A, B, C) = \sum (1, 2, 4, 6, 7) \Rightarrow$ for 4x1 Mux the characteristic Equation is 37

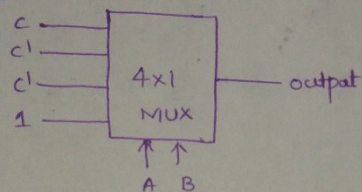
	A	B	C	f
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$$G_1 = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + AB\bar{I}_2 + ABI_3$$

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

Comparing with G_1 $I_0 = C$ $I_2 = C'$
 $I_1 = C'$ $I_3 = 1$

The Mux will be



$$A=0, B=0 \Rightarrow \text{Op} = C$$

$$A=0, B=1 \Rightarrow \text{Op} = C'$$

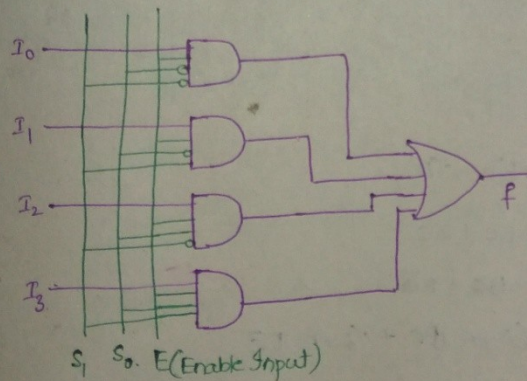
$$A=1, B=0 \Rightarrow \text{Op} = C'$$

$$A=1, B=1 \Rightarrow \text{Op} = 1$$

15. IMPLEMENTING WITH MUX-EXAMPLE

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

15. MULTIPLEXER WITH ENABLE INPUT



$$f = E(I_0 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_1 S_0 + I_2 S_1 \bar{S}_0 + I_3 S_1 S_0)$$

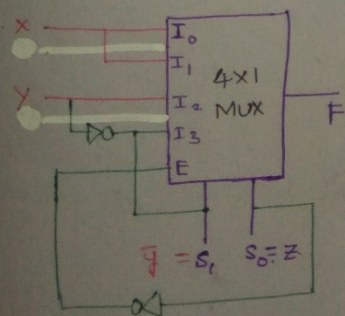
$\Rightarrow$ If the value of $E=1$ then mux works and gives op 1

$\Rightarrow$ If $E=0 \Rightarrow$ Mux stops and gives 0.

Minimize the function represented by following Mux?

The characteristic Equation of 4x1 Mux is

$$\begin{aligned} F &= E(I_0 \bar{S}_1 \bar{S}_0 + I_1 \bar{S}_1 S_0 + I_2 S_1 \bar{S}_0 + I_3 S_1 S_0) \\ &= \bar{Z}(x\bar{z}\bar{y} + x\bar{y}z + y\bar{y}\bar{z} + \bar{y}yz) \\ &= \bar{Z}(x\bar{z}\bar{y} + x\bar{y}z + 0 + \bar{y}yz) \\ &= \bar{Z}(x\bar{y} + \bar{y}z) \\ &= x\bar{y}\bar{z} + \bar{y}z\bar{z} = x\bar{y}\bar{z} \end{aligned}$$



Mux?

16. RELATION BETWEEN SELECT LINES AND INPUTS OF A MUX

$F(A, B, C) = \sum(2, 3, 5, 6, 7)$ Implement using 4×1 Mux such that,

a) $S_1 = B, S_0 = C$

b) $S_1 = C, S_0 = B$

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

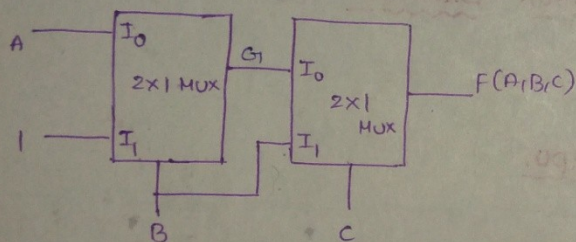
Now, the select lines are $BC \Rightarrow$

$$\begin{array}{cccc} (0) \bar{B}\bar{C} & + & (1) \bar{B}C & + & (2) B\bar{C} & + & (3) BC \\ \downarrow & & \downarrow & & \downarrow & & \downarrow \\ I_0 & & I_1 & & I_2 & & I_3 \end{array}$$

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

What is the function in canonical SOP?



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

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

$$F = \bar{C}I_0 + CI_1$$

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

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

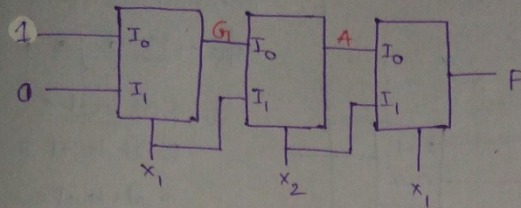
$$F = A\bar{B}\bar{C} + B\bar{C} + BC$$

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

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

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

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

18. CASCADING MUX EXAMPLE-2

$$G_1 = \bar{X}_1(1) + X_1(0) = \bar{X}_1$$

$$A = \bar{X}_2(G_1) + X_2(I_1)$$

$$= \bar{X}_2(G_1) + X_2(X_1)$$

$$= \bar{X}_2 \bar{X}_1 + \bar{X}_2 X_1$$

$$A = (X_1 \odot X_2)$$

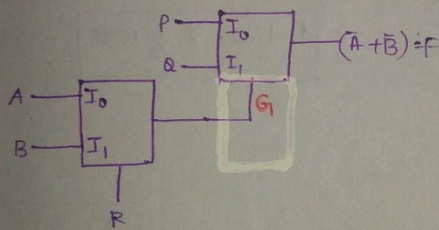
$$F = \bar{X}_1 A + X_1(I_1)$$

$$= \bar{X}_1 (X_1 \odot X_2) + X_1(X_2)$$

$$= \bar{X}_1 (X_2 X_1 + \bar{X}_2 \bar{X}_1) + X_1 X_2$$

$$= \bar{X}_2 \bar{X}_1 + X_1 X_2$$

$$F = X_1 \odot X_2$$

19. CASCADING MULTIPLEXERS - EXAMPLE-3

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

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

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

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

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

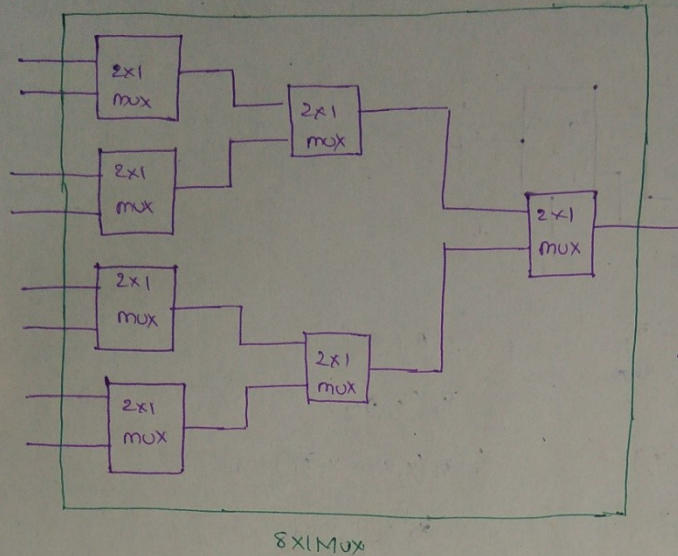
$$\bar{A} + \bar{B} = R\bar{B}P + \bar{A}\bar{R}P + \bar{A}\bar{B}P + \bar{R}AQ + RBQ$$

= Now, do cpm.

=

20. EXPANSION OF MUX'S

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



Now, it is clear that to construct 8×1 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

Now, initially I have 8 lines to cover and now my (2×1) Mux will cover two lines
 $\Rightarrow 1$ Mux will cover 2 lines

$\Rightarrow 2$ lines $\rightarrow 1$ mux

$\Rightarrow 1$ line $\rightarrow \frac{1}{2}$ mux

$\therefore$ 8 lines are covered by $8/\frac{1}{2}$ mux = 4

Now, 4 lines are covered by $4/\frac{1}{2}$ mux = 2 mux

Now 2 lines are covered by = 1 mux

i.e. $8 \text{ lines} \rightarrow 8/\frac{1}{2} \text{ mux} \rightarrow 4$ 7 mux

$4 \text{ lines} \rightarrow 4/\frac{1}{2} \text{ mux} \rightarrow 2 \text{ mux}$

$2 \text{ lines} \rightarrow 2/\frac{1}{2} \text{ mux} \rightarrow 1 \text{ mux}$

② Ex: Construct 32×1 Mux with 4×1 Mux

Initially 32 lines and my small 4×1 mux covers 4 lines $\Rightarrow 1 \text{ mux} \rightarrow 4 \text{ lines}$

$32 \text{ lines} \rightarrow 32/4 \text{ mux} = 8$

$8 \text{ lines} \rightarrow 8/4 \text{ mux} = 2$

$2 \text{ lines} \rightarrow 2/4 \text{ mux} = \frac{1}{2}$

11 Mux

$\therefore 11$ mux are needed to cover 32×1 Mux

1 line $\rightarrow \frac{1}{4}$ mux

Ex: No

Devices

(8×1)
 $\downarrow$
m

Ex: (32×1)
 $\downarrow$
m

$\Rightarrow 1 \text{ level}$

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)

we need
at 3 levels

2

needed
ned

two lines

2 lines

ux

ux

4 lines

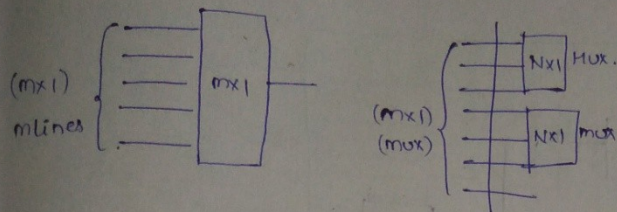
$\frac{1}{4}$ mux

1 mux

then

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

(41)



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

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

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

Now, M lines $\rightarrow (M/N)$ devices

M/N lines $\rightarrow (M/N^2)$ devices

(M/N^2) lines $\rightarrow M/N^3$ devices

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

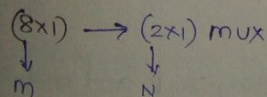
$$\Rightarrow N^k \geq M \Rightarrow k \geq \lceil \log_N M \rceil$$

No of levels

$\Rightarrow$ Now, the No of devices used are $M/N + M/N^2 + M/N^3 + \dots + M/N^k$

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

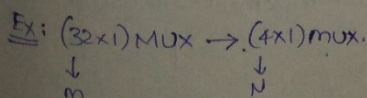
Ex: Now, I want to implement 8×1 mux using (2×1) mux, how many levels and Devices are needed?



$$\Rightarrow \text{No. of levels} = \lceil \log_N M \rceil = \lceil \log_2 8 \rceil = \lceil 3 \rceil = 3 \text{ levels}$$

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

$$= 4 + 2 + 1 = 7 \text{ mux.}$$

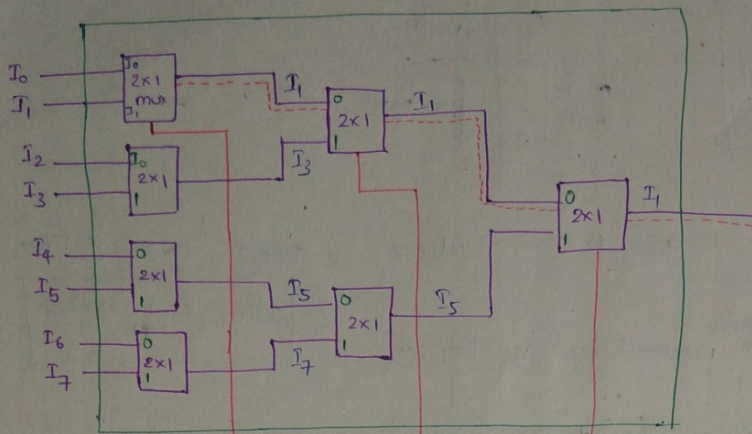


$$\Rightarrow \text{levels} = \lceil \log_N M \rceil = \lceil \log_4 32 \rceil = \lceil 5/2 \rceil = \lceil 2.5 \rceil = 3$$

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

ASSIGNING SELECT LINES WHILE EXPANDING THE MUX. 21

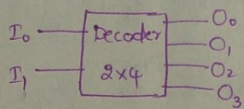
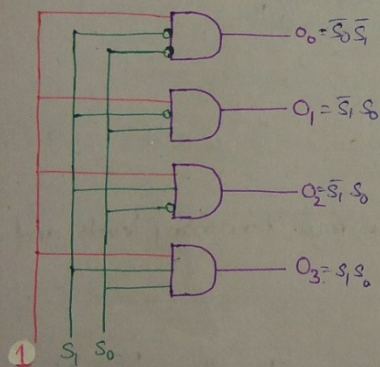
(42)



$S_0 = 1, S_1 = 0, S_2 = 0$
 The inputs that gets selected at level one are (I_1, I_3, I_5, I_7)

$S_2 = 0$ 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



MUX = $2^n \times 1$
 DEMUX = 1×2^n
 DECODER = $n \times 2^n$

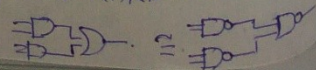
S_1	S_0	O_0	O_1	O_2	O_3	
0	0	1	0	0	0	$O_0 = \bar{S}_1 \bar{S}_0$
0	1	0	1	0	0	$O_1 = \bar{S}_1 S_0$
1	0	0	0	1	0	$O_2 = S_1 \bar{S}_0$
1	1	0	0	0	1	$O_3 = S_1 S_0$

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

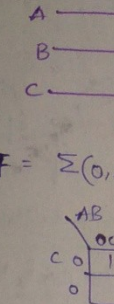
→ Since a decoder provides all the minterms, we could implement, we could implement a func in canonical sop using "OR" Gate.

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

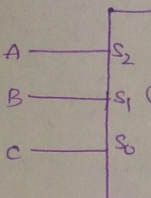
→ Implementing func with decoder is AND-OR REALISATION / NAND-NAND REALISATION.



24. IMPLEMENTATION



25. IMPLEMENTATION



Now, A — S_2
 B — S_1
 C — S_0

Now, the above

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

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

26. DECODER

Same question

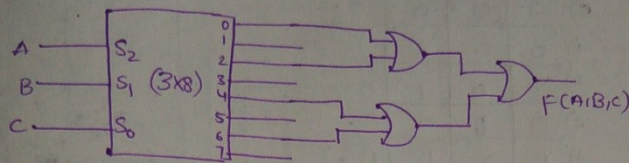
⇒ Now, when a

$$O_0 = (\bar{A} \bar{B} \bar{C})$$

(42)

24. IMPLEMENTING FUNCTIONS WITH EXAMPLE - 1 DECODER

(43)



F is free form

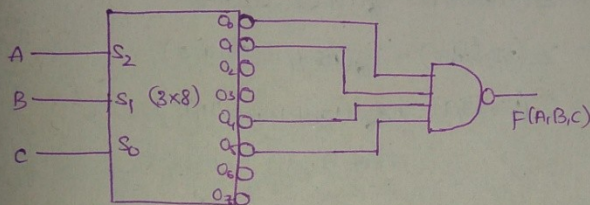
- a) 1 variable c) 3 variable
b) 2 variables d) None

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

AB	00	01	11	10
0	1	1	1	1
1	0	0	0	0

$$\therefore F = 1 \Rightarrow \text{Independent of 2 variables}$$

25. IMPLEMENTING FUNCTIONS WITH DECODER EXAMPLE - 2



F is free form.

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

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

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

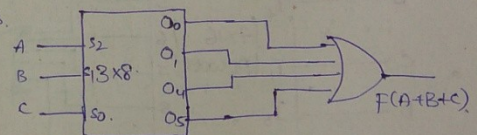
Now, the above diagram can be modified as.

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

AB	00	01	11	10
0	1	1	0	0
1	0	0	1	1

$$= \overline{B}$$

$$\therefore F = \overline{B} \text{ Independent of two variables.}$$



26. DECODER FOR FUNCTION IMPLEMENTATION - Ex-3

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

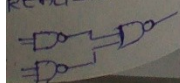
$\Rightarrow$ Now, when a minterm is bubbled (complemented) then it becomes max term.

$$0_0 = (\overline{a}\overline{b}\overline{c}) = (a+b+c) \therefore \pi(0, 1, 4, 5) = \Sigma(2, 3, 6, 7)$$

AB	00	01	11	10
0	0	1	1	0
1	0	1	1	0

$$F = B = \text{Independent of 2 variables}$$

REALISATION.



27. CONVERTING ONE CODE TO ANOTHER USING DECODER

44

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

	W 2	X 4	Y 2	Z 1
0)	0	0	0	0
1)	0	0	0	1
2)	0	0	1	0
3)	0	0	1	1
4)	0	1	0	0
5)	1	0	1	1
6)	1	1	0	0
7)	1	1	0	1
8)	1	1	1	0
9)	1	1	1	1

complements of each other (self complement Representation)

In 2421 code the

9's complement of 5 = 9 - 5 = 4

So take 4 &

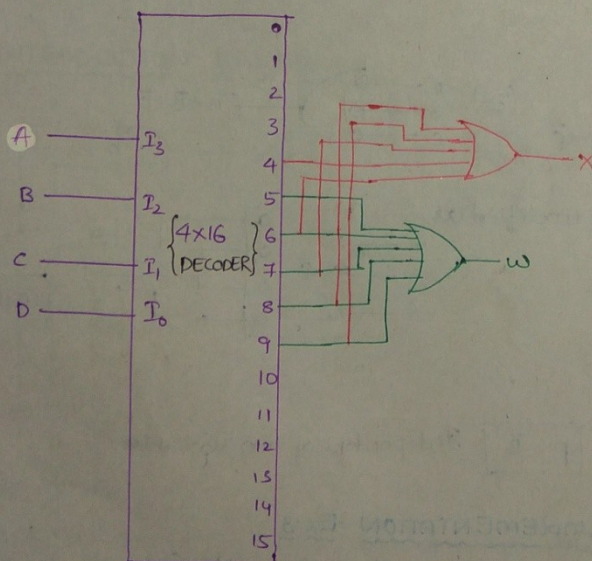
complement it

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

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

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

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



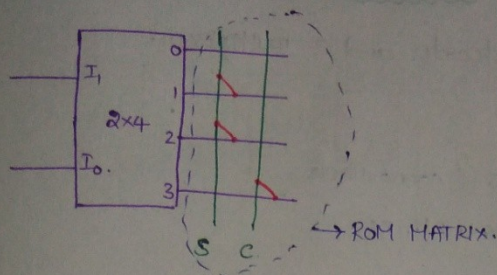
→ similarly Y, Z can also be drawn.

28. Rom IMPLEMENTATION USING DECODER

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

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

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



	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

29. IMPLEMENTING FUNCTIONS USING ONLY DECODER

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

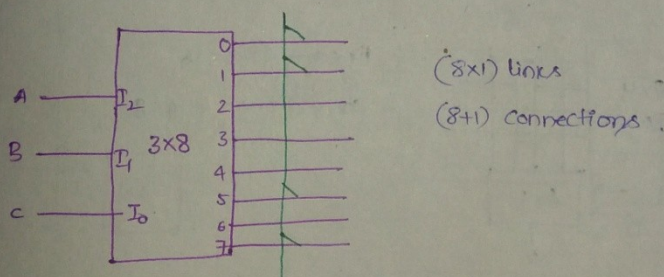
∴ The no. of lines coming/entering into rom matrix is 2^n

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

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

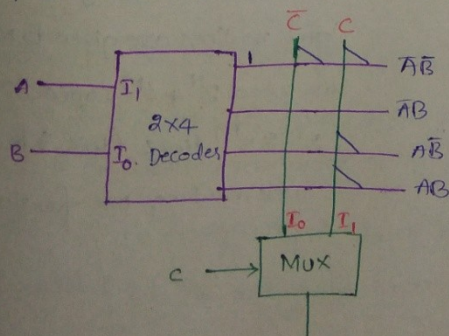
30. IMPLEMENTING FUNCTIONS USING DECODER + MUX EXAMPLE 1

Now, let us consider 3x8 decoder



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

Now, I need to implement the same function using 2x4 decoder. then we need a multiplexer also.

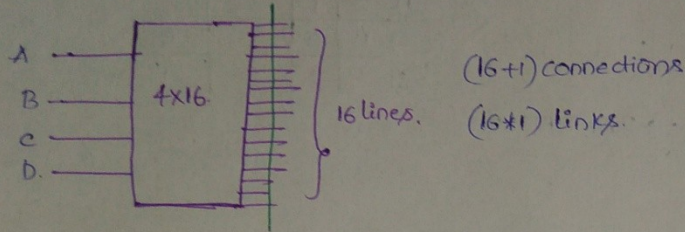


$$f(A,B,C) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}C + ABC$$

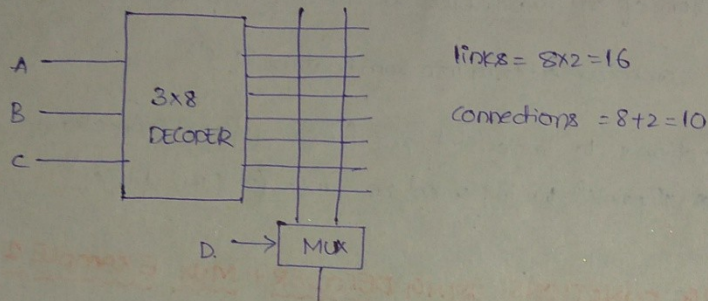
for all - this inputs output should be one and when we give inputs other than these it should be 0.

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

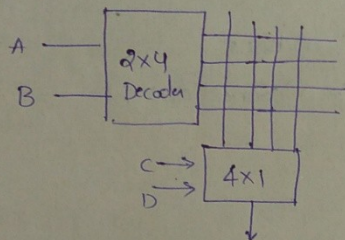
Implement 4x16 Decoder using 3x8 decoder and a multiplexer?



Now, 3x8 decoder



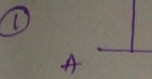
Now, 2x4 Decoder



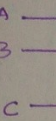
Decoder (n)	Mux	Connections	Links
10×2^{10}	0	$1024 + 1$	10×2^4
5	5	$2^5 + 2^5 = 64$	$2^5 \times 2^5 = 1024$
4	6	$2^4 + 2^6 = 80$	$2^4 \times 2^6 = 1024$
3	7	$2^3 + 2^7 = 126$	$2^3 \times 2^7 = 1024$

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 $2^n \times 2^m$ links. (This is true for implementing one function)

32. DECO



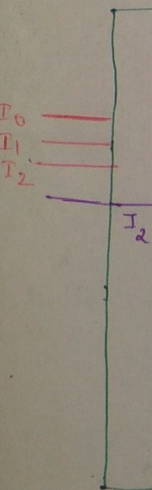
②



33. CONST

Constant

For Deco



⇒ we have

So at k
and 4.

⇒ Now let

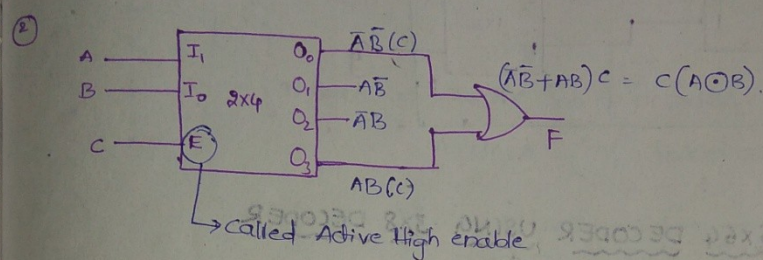
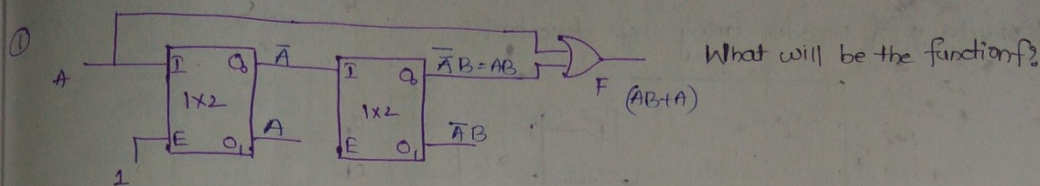
s_2 at lev

⇒ Now, if

(76)

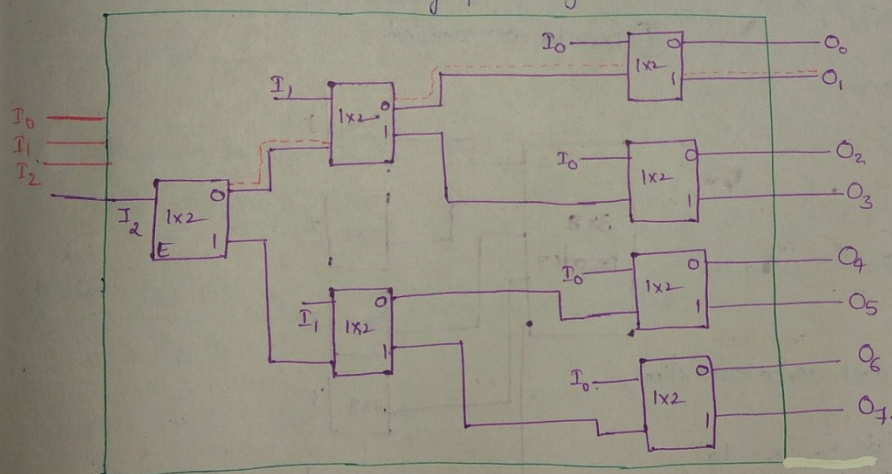
32. DECODER WITH ENABLE INPUT

(47)



Construct 3x8 Decoder using 1x2 Decoder.

For decoders start building from Right side



$\Rightarrow$ we have to cover lines $(0, 0, \dots, 0)$ and each 1×2 Decoder covers 2 lines

So at level 1 we will have 4 decoders. Now, join all the Enables of 4 decodes 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 is say $I_2, I_1, I_0 = (0, 0, 1)$
then q should be enabled

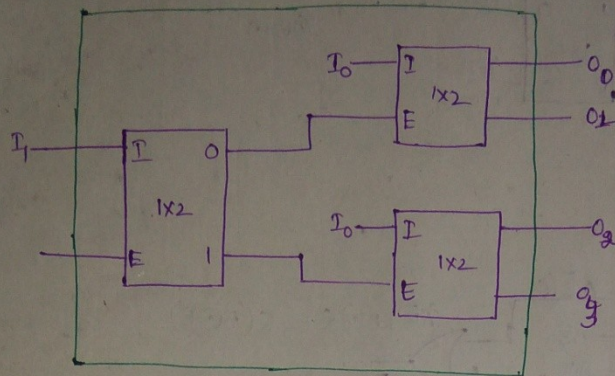
$\Rightarrow T_2 = 0 \Rightarrow 0$ is selected and passed

$\Rightarrow g_1 = 0 \rightarrow 0$ is selected and passed

$\Rightarrow g_0 = 1 \Rightarrow 1$ is selected and Q_1 is enabled

34. CONSTRUCTING 4x2 DECODER USING 1x2 DECODER

Construct 4x2 decoder using 1x2 decoder?



35. CONSTRUCTING 64x64 DECODER USING 3x8 DECODER

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

$$\Rightarrow 64 \text{ lines} \rightarrow \frac{64}{8} \text{ devices} = 8 \text{ Decoders}$$

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

(9) Decoders are needed.

Now, 64x64 decoder using 3x8 decoders

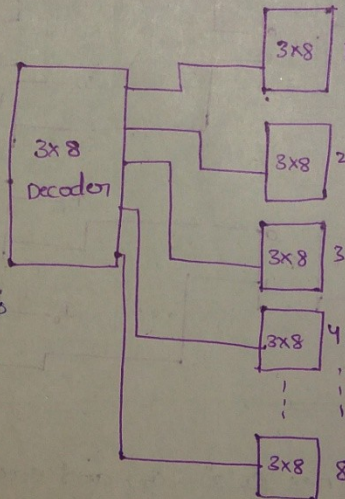
64 lines and each device covers 4 lines

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

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

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

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



36. EXPANSION

Now 3

m x 2
n x 2

No. of devices

EX: 6x6

4x16

levels =

No. of devices

38. EXPANSION

Construct

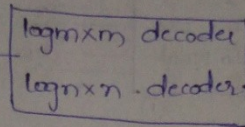
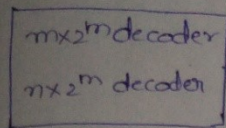
We assume
line wise).

(48)

36. EXPANSION OF DECODER IN GENERAL

(49)

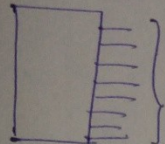
Now I want to construct $(m \times 2^m)$ decoder by using $(n \times 2^n)$ decoder.



1 device covers n lines

$\Rightarrow$ 1 line means/covers $\frac{1}{n}$ devices

$\Rightarrow$ 1 line = $\frac{1}{n}$ devices



m lines. $\Rightarrow$ Now the no. of decoders required to cover

m lines = $\left(\frac{m}{n}\right)$ devices are needed.

$\Rightarrow$ m lines $\rightarrow \left(\frac{m}{n}\right)$ devices

$\left(\frac{m}{n}\right)$ lines $\rightarrow \left(\frac{m}{n^2}\right)$ devices

$\left(\frac{m}{n^2}\right)$ lines $\rightarrow \left(\frac{m}{n^3}\right)$ devices

$$\text{No. of devices needed} = \sum_{k=1}^{\log_n m} \left(\frac{m}{n^k}\right)$$

$$\frac{m}{n^k} \Rightarrow \frac{m}{n^k} \leq 1$$

$$\Rightarrow m \leq n^k$$

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

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

EX: 6×64 decoder $|$ $m=64$
 4×16 decoder $|$ $n=16$

$$\text{levels} = \frac{\log 64}{\log 16} = \left\lceil \frac{6}{4} \right\rceil = \left\lceil \frac{3}{2} \right\rceil = 2 \text{ levels are needed}$$

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

$$= 6 + \frac{64 \cdot 1}{16 \times 16}$$

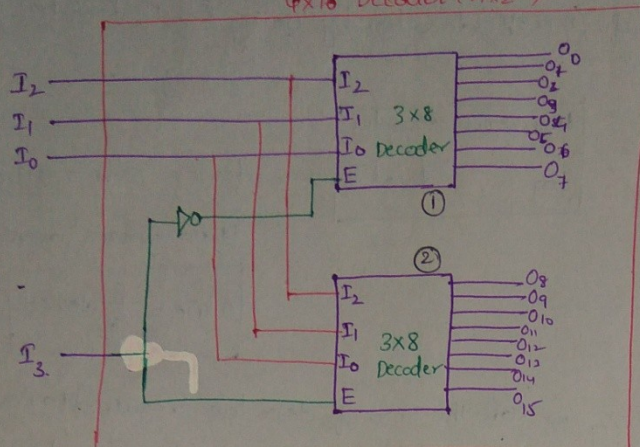
$$= 6 + \left\lceil \frac{1}{4} \right\rceil = 6 + 1 = 7 \text{ decoders are needed}$$

38. EXPANSION OF DECODERS IN ANOTHER WAY

Construct 4×16 decoder using two (3×8) decoders.

We assume that there are no restrictions (only 80 and 80 decoders are available like wise).

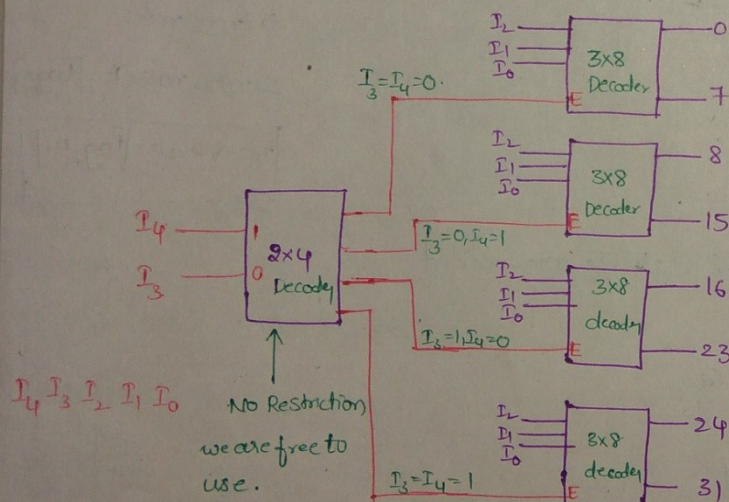
(4x16 Decoder (7x2^n))



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 ($I_3=0$)).

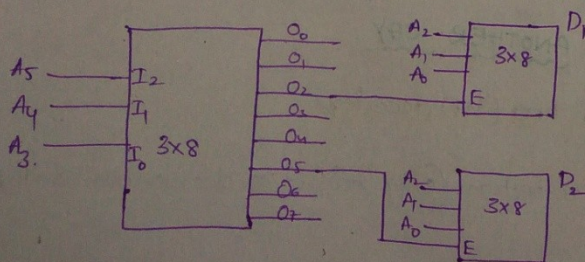
$I_3 I_2 I_1 I_0 = 0000 \Rightarrow O_0$ will be output

Ex:2 construct a (5x32) decoder using 4 (3x8) decoders.



= 32 lines
= each device - 8 lines
= $32/8$ devices = 4D

41. FINDING THE ADDRESS RANGES OF DEVICES



43. FINDING

Consider the
the Nature of

(50)

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)

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

Similarly for D_2 , O_5 should be enabled $\Rightarrow A_5 A_4 A_3 = 101$

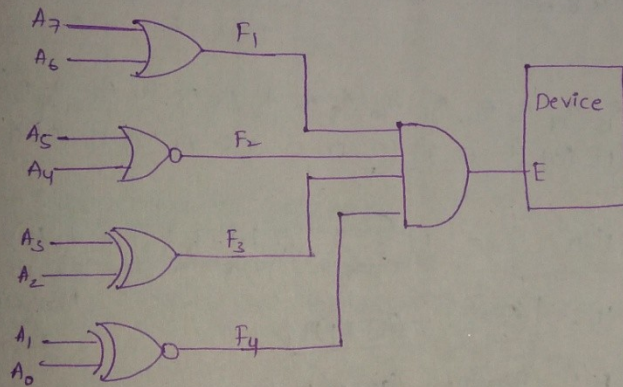
$$\begin{matrix} A_5 & A_4 & A_3 & A_2 & A_1 & A_0 \\ 1 & 0 & 1 & \begin{Bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{Bmatrix} \end{matrix}$$

42. EXAMPLE ON ENABLING A DEVICE

use $E=0$

bit is

Consider a device that is enabled using 8 address bits as shown below. For how many addresses the device is enabled



vice-glines

ans = 4D

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

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

3 Combination

1 Combination

2 combinations

2 combinations

makes $F_1=1$ $\begin{Bmatrix} 0,1 \\ 1,0 \\ 1,1 \end{Bmatrix}$

$F_2=1$ $\begin{Bmatrix} 0,0 \\ 1,1 \end{Bmatrix}$

$\begin{Bmatrix} 0,1 \\ 1,0 \end{Bmatrix}$

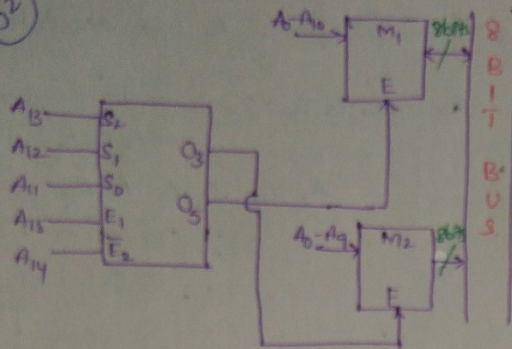
$\begin{Bmatrix} 1,1 \\ 0,0 \end{Bmatrix}$

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

43. FINDING THE ADDRESS RANGES OF MEMORY DEVICES

Consider the following interface of two memory devices with 3x8 decoder. Compute the Nature 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 = 16 $\Rightarrow 2^{16}$ addresses are possible

$\Rightarrow 2^8$ words are stored

$\Rightarrow$ size of each word = 8 bits

$\Rightarrow$ size of $m_1 = 2^{16} \times 8 = 2^{14}$ bits
 $= 2^{11}$ B

M_2 : No. of Address bits = $A_7 A_6 A_5 \dots A_0 = 10$

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

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

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

To enable $m_1 \Rightarrow q_3$ 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.

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

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

$M_1: (A800H) - (AFFFH)$

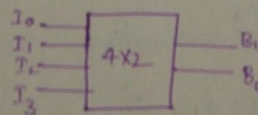
Now Combine 4, 4 together

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

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: 1×2^n
 DECODER: $n \times 2^n$
 ENCODER: $2^n \times n$

$$B_1(I_3, I_2, I_1, I_0) = I_0' I_1' I_2' I_3 + I_0' I_1' I_2 I_3'$$

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

$$B_2(I_3, I_2, I_1, I_0) = I_0' I_1' I_2' I_3 + I_0' I_1 I_2' I_3'$$

$$= I_0' I_2' (I_1' I_3 + I_1 I_3')$$

$$B_2 = I_0' I_2' (I_1 \oplus I_3)$$

$\rightarrow$ They are
 $\rightarrow$ They are
 $\rightarrow$ They are
 $\rightarrow$ Due to
45. PRIORITY
 priority of
 Highest

I_0	1
I_1	0
I_2	0
I_3	1
	0

46. INTRODUCTION
 $\rightarrow$ Hazards
47. HAZARD
 $\rightarrow$ The mal
 and per
 $\rightarrow$ The per
 of the
 $\rightarrow$ The haz
 $\rightarrow$ The sig
 This te
 Procedure for
 $\rightarrow$ Inactive
 the logic
 $\rightarrow$ Apply the
 $\rightarrow$ 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

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

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

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

$$B_1 = I_2 + I_3$$

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

46. INTRODUCTION TO HAZARDS

- Hazards
 - Temporary Hazards
 - Permanent Hazards. — mainly occur because of open circuiting/short circuiting.

47. HAZARDS IN DIGITAL CIRCUIT

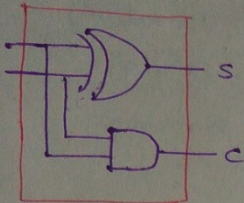
- The mal function of a digital circuit is called hazard. They can be Temporary and permanent. The temporary hazards are due to uneven delays of inputs.
- The permanent hazards are resulted due to open circuit or short circuit of the connecting leads (Terminals).
- The hazards can be stuck at 0 (s-a-0) 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 the 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 →



Half Adder

$$\text{sum} = S = \bar{x}y + x\bar{y} = x \oplus y$$

$$\text{carry} = xy$$

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

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

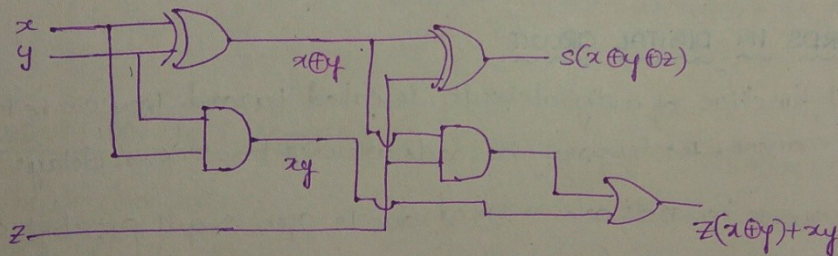
xy	00	01	11	10
0		1		1
1	1		1	

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

xy	00	01	11	10
0			1	
1		1	1	1

$$c = xy + yz + zx$$

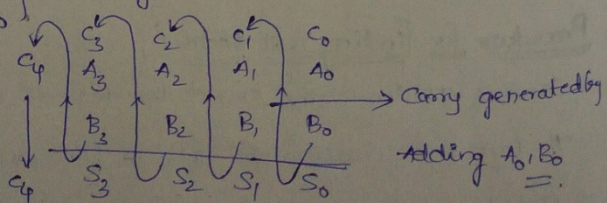
$$c = z(x \oplus y) + xy$$



51. RIPPLE CARRY ADDER

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

1st Binary number = $A_3 A_2 A_1 A_0$
2nd Binary number = $B_3 B_2 B_1 B_0$



Carry generated by

Adding A_0, B_0

Now, we can

⇒ The disad

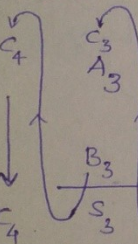
and C_2 is

in parallel

are desig

look ahead

52. CARRY



Now, let us

Now, $C_1 =$

$C_2 =$

$C_2 =$

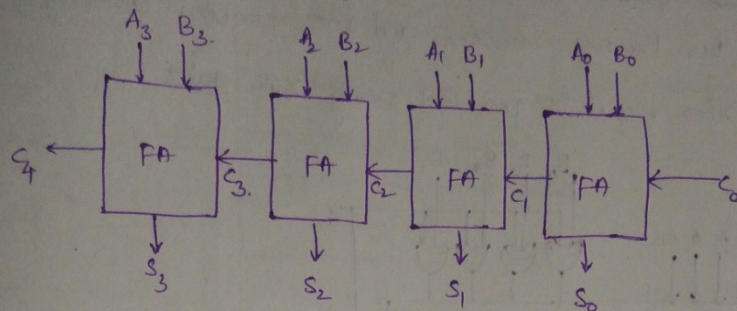
$C_4 = [C_0 P_0$

$C_4 = C_0 P_0$

=

(54)

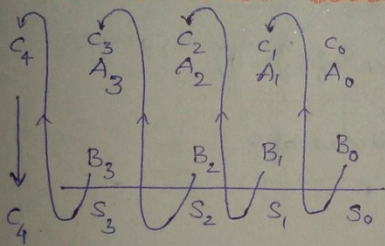
Now we can implement this using Full Adder.



Ripple carry Adder.

⇒ 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



$$C_1 = C_0 (A_0 \oplus B_0) + A_0 B_0$$

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

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

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

Now, Apply
Back substitution
method.

Now, let us assume $G_i = A_i B_i$, $P_i = A_i \oplus B_i$ Now, $C_1 = C_0 P_0 + G_0$

Generating func

propagating func

$$C_2 = C_1 P_1 + G_1$$

$$C_3 = C_2 P_2 + G_2$$

$$C_4 = C_3 P_3 + G_3$$

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

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

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

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

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

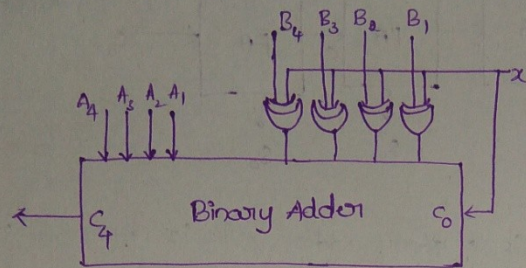
$$C_4 = [C_0 P_0 P_1 P_2 + G_0 P_1 P_2 + G_1 P_2 + G_2] P_3 + G_3$$

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

=

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

$$\begin{aligned} \text{If } x=1, \quad A + (\text{1st complement of } B) + 1 &\rightarrow \text{Carry's} \\ &= A + \underbrace{2\text{'s complement of } B} \\ &= A - B \end{aligned}$$

The Adder behaves as Adder when $x=0$

" " " " Subtractor " $x=1$.

The above circuit not only functions as Adder and subtractor

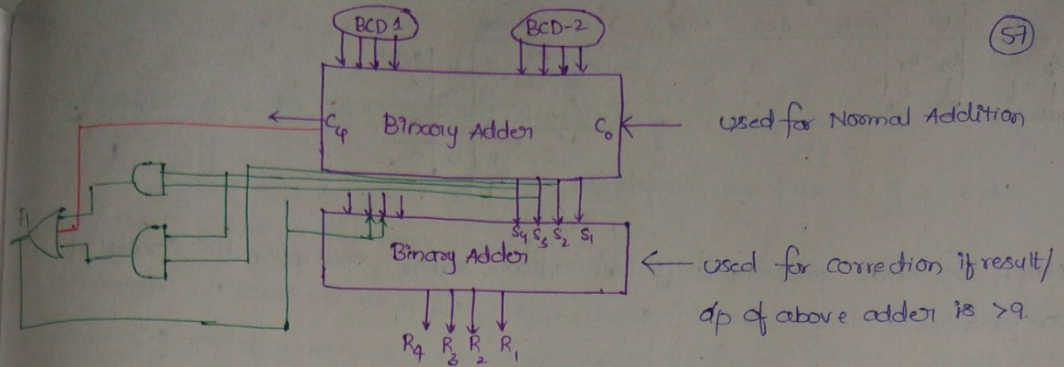
A	B	x	Function.
BCD	0011	0	BCD $\rightarrow$ Ex-3 Converter
Ex-3	0011	1	Ex-3 $\rightarrow$ BCD Converter
1001	BCD	1	$(9-B) = 9\text{'s complement of given no B in (BCD)}$
1010	BCD	1	$(10-B) = 10\text{'s complement of given no. B in (BCD)}$

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.

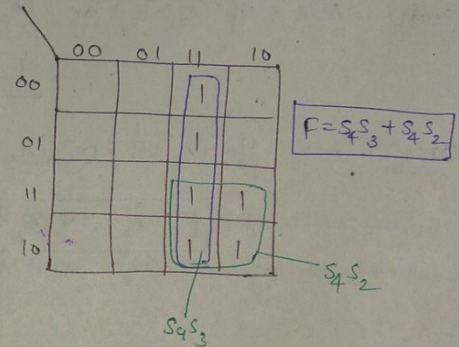
provided

(56)



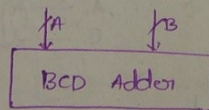
⇒ Correction is Required when $S_4S_3S_2S_1 > 9$ (i.e.) if $C_0 = 1$, Now let us form a func where the sum of $S_4S_3S_2S_1 > 9$, now the combinations for which the sum is > 9 are.

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 $C_0 = 1$ 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 nois) are valid.

Invalid combinations = Total combinations - valid combinations

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

$$= 156$$

add 6 to

59. 2 BIT 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)

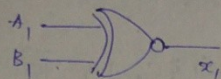
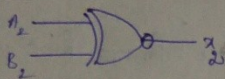
A	B	A=B	A<B	A>B
A ₂ A ₁	B ₂ B ₁			
↓ ↓	↓ ↓			
msB LSB	msB LSB			

XNOR

a	b	a ⊙ b
0	0	1
0	1	0
1	0	0
1	1	1

58

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



This diagrams represent $A=B$.

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 $(A_2 = B_2)$ 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 $(A_2 = B_2)$ and $A_1 < B_1$

$$F = \bar{A}_2 B_2 + x_2 \bar{A}_1 B_1 \quad \text{if } F=1 \text{ then } A<B$$

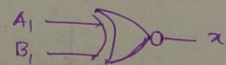
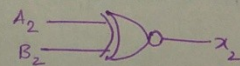
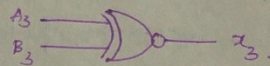
60. 3, 4 BIT COMPARATORS

$$A = A_3 A_2 A_1$$

$$B = B_3 B_2 B_1$$

Case (1): $A=B$

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



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

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

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

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

1. INTRO

→ The on pr

→ The pre

new c

presen

Inputs -

2. LATCH

→ A Flipp

→ A memo

stored

be step

→ The basi

→ It has

called

3. SR-FLIP

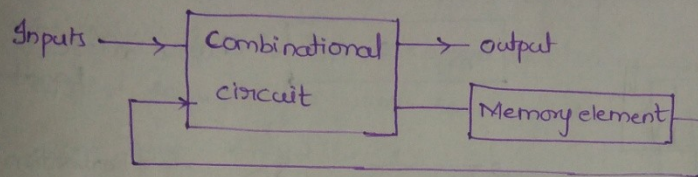
S — D

R — D

4. SEQUENTIAL CIRCUITS

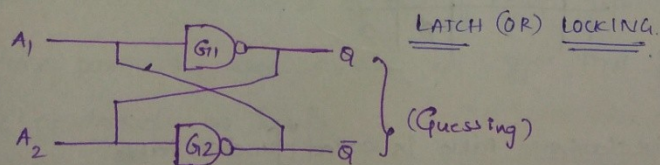
1. INTRODUCTION TO SEQUENTIAL CIRCUITS.

- The external op 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.

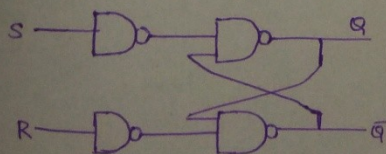


2. LATCH AND FLIP FLOP

- A flipflop 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 there after its contents can be replaced by New values.
- The basic binary or digital memory circuit is known as flipflop.
- It has 2 stable states which are known as '1' or '0', so it is called called bistable multi-vibrator.



3. SR- FLIPFLOP



- ⇒ 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 op is not just a func of the present inputs but also previous outputs.

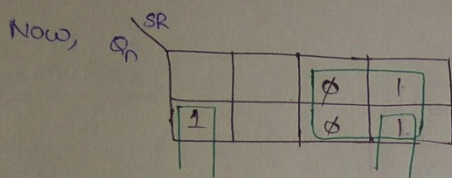
3 are 2^n
 $= \frac{2^{2n} - 2^n}{2}$
 $= \frac{2^{2n} - 2^n}{2}$

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	ϕ
1	1	1	ϕ

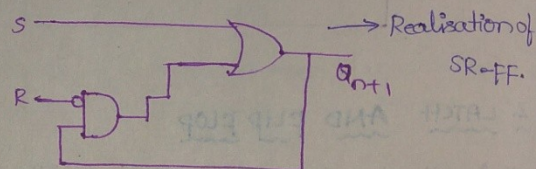
⇒ we can use the SR-flipflop in only 3 modes ⇒ latch, set, Reset modes.

characteristic table

invalid. (O/p is irrespective of the values of Q_n and Q_{n+1})



$$Q_{n+1} = S + Q_n R = \text{characteristic Equation}$$



→ 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 o/p ~~next~~ of next state that you have assumed.

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

→ R can be '0' or '1' so ' ϕ '

→ minimised characteristic table is called "function table"

S	R	Q_{n+1}
0	0	Q_n
1	0	1
0	1	0
1	1	ϕ

4. CLOCK

→ Now, some then the we should not be

→ clock

→ where

the value

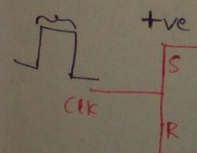
enabled

clock

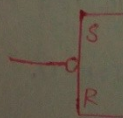
⇒ Edge trig

only on

the o/p



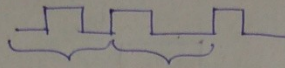
-ve



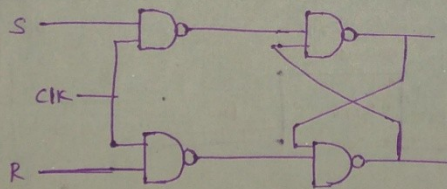
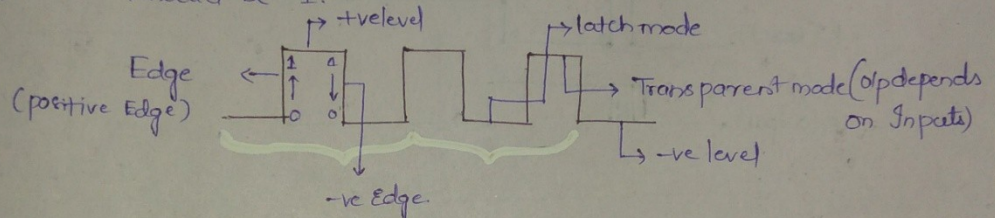
4. CLOCKED FLIPFLOPS

→ Now, what happens in general with SR-flipflop/flipflop? because of some external disturbances the $\bar{y}/p$ 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

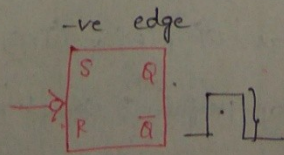
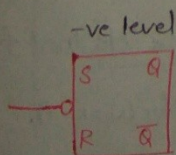
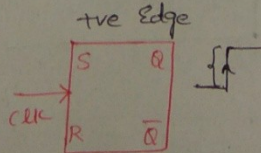
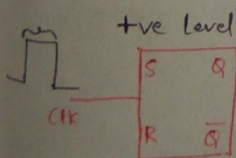


→ whenever clock is '0' then the o/p of NAND gates will be 1 and 1 irrespective of the values of S, R it is as good as having S, R as 0, 0. $\Rightarrow$ latch mode is enabled. so, when you want to change the o/p depending on S, R then the clock value should be '1'.



$\Rightarrow$ positive level triggered flipflop.

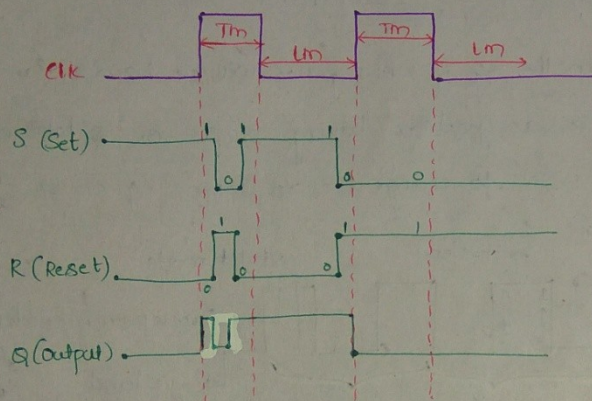
$\Rightarrow$ Edge triggered means the output of the flipflop depends on the input only on some particular edge so, when we trigger that particular edge then the o/p of FF depends on input.



5. POSITIVE LEVEL TRIGGERED

Now, let's see how the positive level triggered FF react to the inputs.

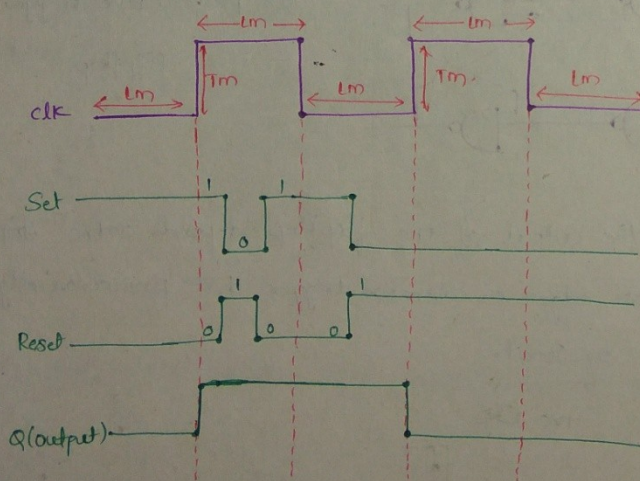
⇒ In SR-FF 'S' means Set and 'R' means Reset. The reason is whenever 'S' is '1' the o/p is always going to be one, and whenever $R=1$ the output is always going to be Zero. and when both $S,R=0$, then o/p will be in latch mode



T_m = Transparent mode
 L_m = Latch mode

⇒ Assume the o/p is present at '0' level.

6. EDGE TRIGGERED



T_m = Transparent mode

7. JK- FLIP FLOP

⇒ The J-K - flipflop is same as SR flipflop ($J=S$, $K=R$) It is defined for $J=K=1$ also and the flipflop performs complementation for ($J=K=1$).

J	K	Q_n
0	0	0
1	0	1
0	1	0
1	1	$\bar{Q}_n$

Characteristic

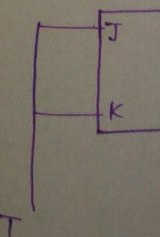
J	K	Q_n
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Excitation table

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

8. T- FLIP FLOP

T- flipflop = To



Q2

Q3

Whenever the output will be

output mode

o/p is level.

transparent mode

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

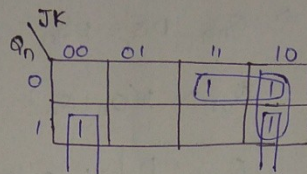
function table for J-K flipflop

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

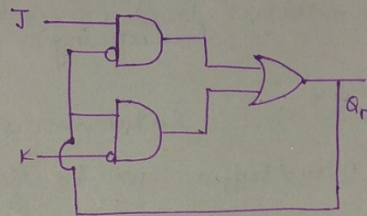
same as SR-FF

Complementation



$$F = J\bar{Q}_n + \bar{K}Q_n$$

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

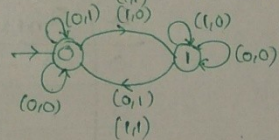


Excitation table

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

State diagram

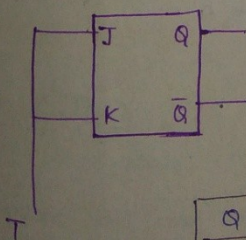
⇒ The 2 states of flipflop are {0,1}



look at function table and draw.

8. T-FLIP FLOP

T-Flipflop = Toggle Flipflop (This flipflop is either latching or complementing so it is called Toggling)



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

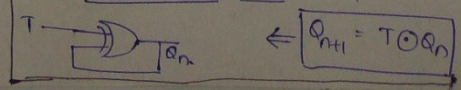
function table

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

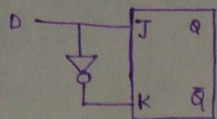
Excitation table

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



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

9. D-FLIP FLOP (Delay-Flipflop)



D	Q_{n+1}
0	0
1	1

= function table

characteristic table

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

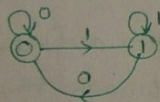
$$\Rightarrow Q_{n+1} = D\bar{Q}_n + DQ_n$$

$$Q_{n+1} = D(Q_n + \bar{Q}_n)$$

$Q_{n+1} = D$ $\therefore$ output is directly depending on input and there is no need of memory element at all (No feedback).

$\Rightarrow$ (what we 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

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

10. EXAMPLE ON FLIP FLOP 1

The characteristic Equation of a flipflop is $Q_{n+1} = \bar{x}_1\bar{Q} + \bar{x}_2Q$. Define the behaviour of this.

Function table

x_1	x_2	Q_n	
0	0	1	$\rightarrow$ Set state
0	1	$\bar{Q}$	$\rightarrow$ Toggle
1	0	Q	$\rightarrow$ Latch mode
1	1	0	$\rightarrow$ Complement Reset state

$$Q_{n+1} = \bar{x}_1\bar{Q} + \bar{x}_2Q$$

12. INTRODU

Conversion

1> Get the

2> Replace the

3> Obtain the

(64)

characteristic table

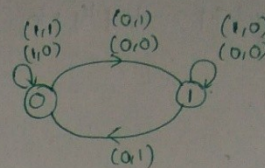
x_1	x_2	Q	Q_n
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Excitation table

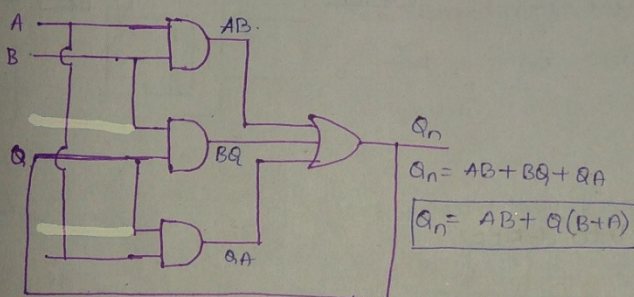
Q	Q_n	x_1	x_2
0	0	1	ϕ
0	1	0	ϕ
1	0	ϕ	1
1	1	ϕ	0

state diagram

(65)

II. EXAMPLE ON FLIPFLOP-2

consider the following realization, construct excitation table?

function table

A	B	Q_n	
0	0	0	Reset
0	1	Q	latch
1	0	Q	latch
1	1	1	setting

Excitation table

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

12. INTRODUCTION TO FLIP FLOP INTER CONVERSION

Conversion of given flipflop to another flipflop.

- 1) Get the characteristic table of target FF
- 2) Replace the next state using excitation of the given FF
- 3) Obtain the expressions for the i/p of given FF and realise them.

Ex1: Conversion of J-K flipflop to T-Flipflop.

66

1) 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

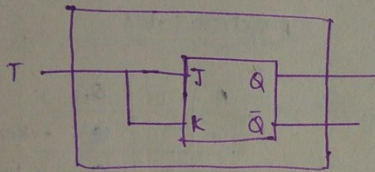
2) Replace the next state with excitation

T	Q	Q _{n+1}	J	K
0	0	0	0	ϕ
0	1	1	ϕ	0
1	0	1	1	ϕ
1	1	0	ϕ	1

Now, write the values of JK where the output changes to (0 $\rightarrow$ 0) (1 $\rightarrow$ 1) (0 $\rightarrow$ 1) (one $\rightarrow$ zero)

$\Rightarrow$ Now, characteristic eqn for $J=T$
 $K=T$

Q _n	0	1
J	0	1
K	0	1



Ex2: conversion of T-ff into J-K FF

1) Target FF = J-K flip flop $\Rightarrow$ 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

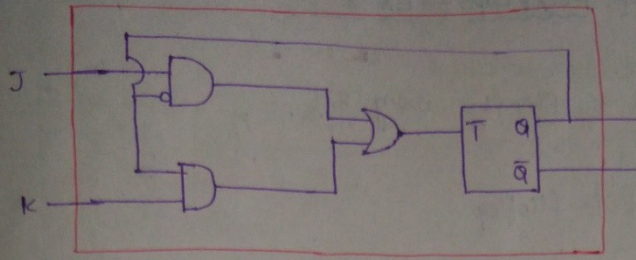
write the values of T for which output change from (Q_n $\rightarrow$ Q_{n+1}) looking at characteristic table of T.

characteristic equation for T =

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

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

66

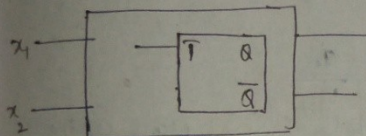


67

13. INTER CONVERSION OF FLIPFLOPS - Example-1

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.

⇒ Realise it using DT-flipflop means the given flipflop is 'T' and using T-flipflop you want to recognise new flip-flop.



∴ 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

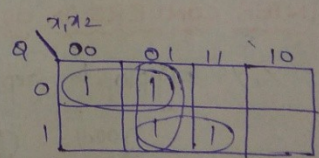
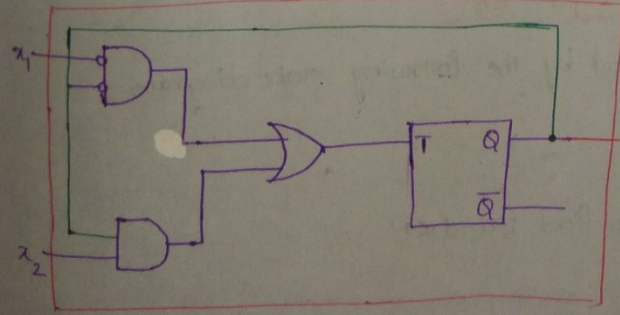
characteristic table of x_1, x_2, Q .

x_1	x_2	Q	Q_n
0	0	0	1
0	0	1	One(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 of 'T'.



ues of T
put change
(+1) looking
istic table



$$T = \bar{x}_1 \bar{Q} + \bar{x}_2 Q$$

14. INTER CONVERSION OF FLIPFLOPS - Ex-2

$x_1 x_2 \rightarrow T$
 $Q_n = \bar{x}_1 \bar{Q} + \bar{x}_2 Q$ } Construct T-flipflop using x_1, x_2 .

The characteristic table of T-flipflop.

T	Q	Q _n	x ₁	x ₂
0	0	0	1	0
0	1	1	0	0
1	0	1	0	0
1	1	0	0	1

Excitation table for x_1, x_2

$$\Rightarrow x_1 = \bar{T} \quad x_2 = T$$

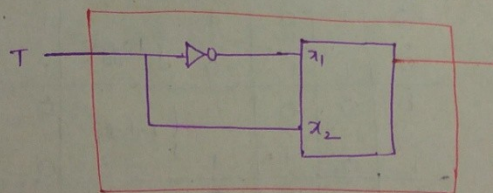
Now, in order to get excitation table of x_1, x_2, Q we should get function table of x_1, x_2
 $\rightarrow$ characteristic table of $x_1, x_2 \rightarrow$ Excitation table of x_1, x_2 .

x ₁	x ₂	Q _n
0	0	1
0	1	0
1	0	0
1	1	0

This are written by
 $Q_n = \bar{x}_1 \bar{Q} + \bar{x}_2 Q$

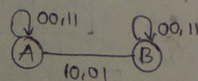
x ₁	x ₂	Q	Q _n
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Now, write excitation table beside the characteristic table of x_1, x_2 .

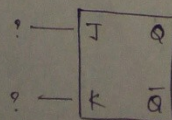


15. INTER CONVERSION OF FLIPFLOPS - Ex-3

An, X, Y flipflop is represented by the following state diagram.



To realize it using J,K flipflop find J and K.

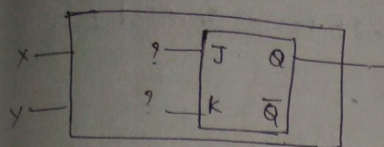


68

Given state diagram



69



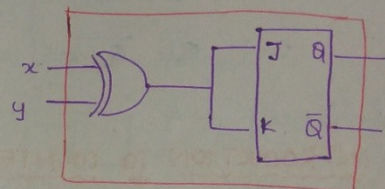
undirected edge which means $\begin{cases} A \xrightarrow{10} B \\ B \xrightarrow{10} A \end{cases}$
 $\begin{cases} A \xrightarrow{01} B \\ B \xrightarrow{01} A \end{cases}$

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	ϕ
0	0	1	1	ϕ	0
0	1	0	1	1	ϕ
0	1	1	0	ϕ	1
1	0	0	1	1	ϕ
1	0	1	0	ϕ	1
1	1	0	0	0	ϕ
1	1	1	1	ϕ	0

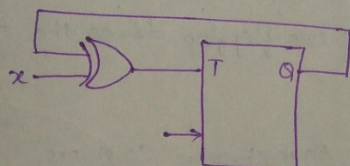
$$J = \bar{x}y + x\bar{y} = x \oplus y$$

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



16. INTER CONVERSION OF FLIPFLOPS - Ex-1

What is the behaviour of following one-input flipflop '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.

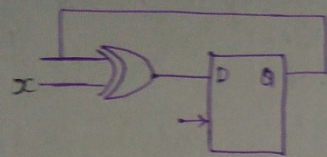
Characteristic table of X.

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}

∴ $Q_{n+1} = x$ ∴ what ever you give as input is displayed as output after some delay so it acts as Delayed flipflop.

17. INTER CONVERSION OF FLIPFLOPS - Ex-5



x	Q_n	Q_{n+1}	$D = Q_n \oplus x$
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

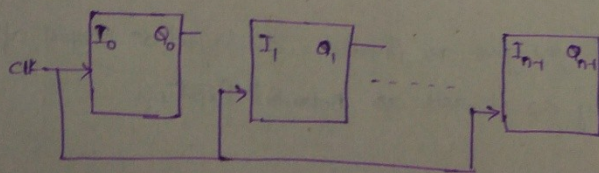
when $x=0$, $Q_n = Q_{n+1}$
 $x=1$, $Q_n = \overline{Q_{n+1}}$ } T flipflop.

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



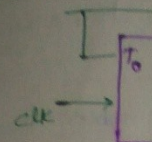
⇒ propagation delay in Synchronous counter is

$$T_{p\text{syn}} = T_{FF} + T_{\text{combinational}}$$

$$T_{clk} \geq T_{p\text{syn}}$$

⇒ T_{clk} means signal.

Asynchronous



20. SHIFT

Synchronous

⇒ $I_1 = f(Q_0, \dots)$
 ⇒ its design

$$I_1 = Q_0$$

$$I_2 = Q_1$$

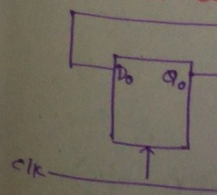
$$I_3 = Q_2$$

⇒ Here data

Counter (D)

$$I_0 = f(Q_n)$$

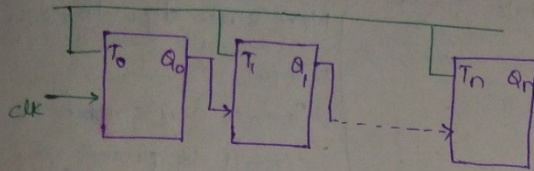
21. Mod-2 R



$\Rightarrow T_{clk}$ means the amount of time we should wait before sending the next input signal.

(71)

Asynchronous counter:



$\Rightarrow$ propagation delay in Asynchronous Counter

$$T_{pdasy} = N * T_{FF} + T_{combinational}$$

$$T_{clk} \geq T_{pdasy} \quad T_{FF} = \text{PD of each Flipflop}$$

20 SHIFT COUNTERS

Synchronous counter

$$\Rightarrow I_i = f(Q_0, Q_1, \dots, Q_{N-1}) \quad 0 \leq i \leq (N-1)$$

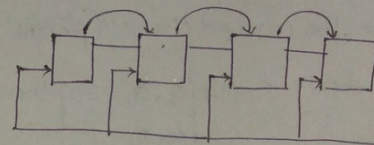
$\Rightarrow$ Its design is more complex, so simplifications are done as follows

$$1) I_i = Q_{i-1} \quad 1 \leq i \leq N-1$$

$$I_1 = Q_0$$

$$I_2 = Q_1$$

$$I_3 = Q_2$$



Each FF receives the output of previous FF as 'clk' input, then list about 1st FF?

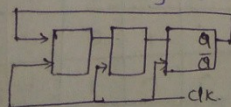
$\Rightarrow$ Here data is shifted from one flipflop to another FF so this is called shift counter (D-FF's are used)

$$2) I_0 = f(Q_{N-1})$$

$$I_0 = f(Q_{N-1})$$

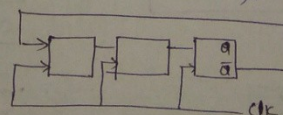
$$I_0 = (Q_{N-1})$$

(Ring counter)

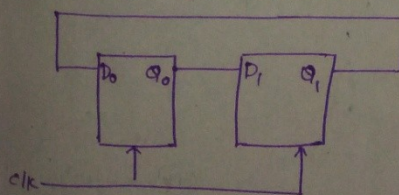


$$I_0 = (\bar{Q}_{N-1})$$

(Johnson Counter)



21. Mod-2 RING COUNTERS



$\Rightarrow$ 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-FFs) can be
 if we observe there are 2 FF and one FF is going to give Q_0 state and another
 FF is going to give Q_1 state (Here assume $Q_0 = \text{LSB}$, $Q_1 = \text{MSB}$)

Present state		Next state	
D_1	D_0	Q_1	Q_0
0	0	0	0
1	0	0	1
0	1	1	0
1	1	1	1

Next state depend on 2 things
 1) Input
 2) In case any combinational circuit is present what is the glp to the combinational circuit.

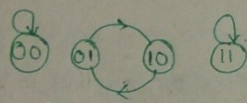
Here Q_{0N} depends on D_0 and
 D_0 depends on Q_1 (look at diagram)

$$\therefore Q_{0N} = Q_1 \quad (Q_{0N} = D_0 = Q_1)$$

$$\Rightarrow (Q_{0N} = Q_1)$$

⇒ Now, Q_{1N} is nothing but Input given at $D_1 \Rightarrow Q_{1N} = D_1$

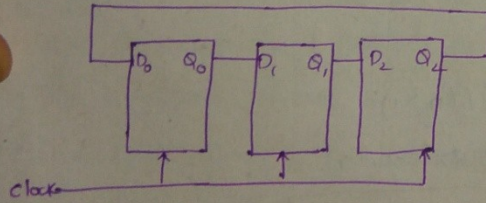
⇒ Now observe the marked circles (present state, Next state) and find how they behave, i.e. when Input $\{Q_0, Q_1 = 00\}$ then they remain in same state
 $\{Q_0, Q_1 = 11\}$



⇒ mod 2 counter (only 2 states are in counting)

⇒ using Ring counter we get "mod N" counters.

22. MOD 4 RING COUNTERS



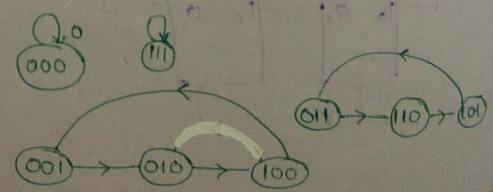
$$Q_{0N} = D_0 = Q_2 \Rightarrow Q_{0N} = Q_2$$

$$Q_{1N} = D_1 = Q_0 \Rightarrow Q_{1N} = Q_0$$

$$Q_{2N} = D_2 = Q_1 \Rightarrow Q_{2N} = Q_1$$

$$\begin{matrix} D_0 = Q_2 \\ D_1 = Q_0 \\ D_2 = Q_1 \end{matrix} \Rightarrow \begin{matrix} Q_{0N} = D_0 \\ Q_{1N} = D_1 \\ Q_{2N} = D_2 \end{matrix}$$

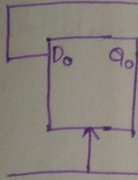
present			Next			D_0	D_1	D_2
Q_2	Q_1	Q_0	Q_{2N}	Q_{1N}	Q_{0N}			
0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	1	0
0	1	0	1	0	0	0	0	1
0	1	1	1	1	0	0	1	1
1	0	0	0	0	1	1	0	0
1	0	1	0	1	1	1	1	0
1	1	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	1



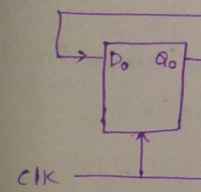
⇒ If a Ring counter
 ⇒ In above

23. MOD 4

⇒ Twisted



24. MOD 6

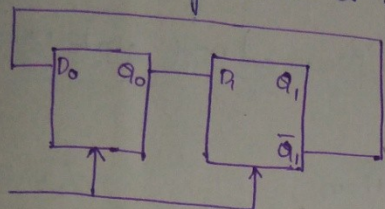


$$\begin{matrix} D_2 = Q_1 \\ D_1 = Q_0 \\ D_0 = Q_2 \end{matrix}$$

- 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 mod 3 operation.

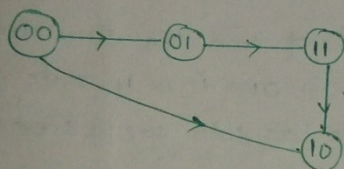
23. MOD 4 JOHNSON COUNTER

⇒ Twisted Ring counter or Johnson counter.



There are two states Q_0, Q_1 so.

Q_1	Q_0	Q_{1N}	Q_{0N}	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

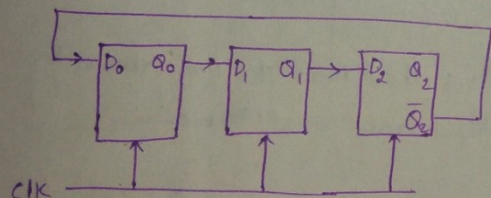


= mod 4 counter.

$$\begin{aligned} Q_{1N} &= D_1 \\ Q_{0N} &= D_0 \end{aligned}$$

$$\begin{aligned} D_0 &= \overline{Q_1} \\ D_1 &= Q_0 \end{aligned}$$

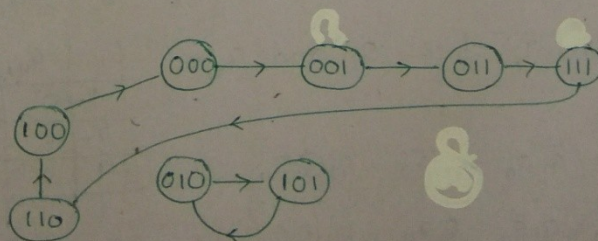
24. MOD 6 JOHNSON COUNTER



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

$$\begin{aligned} D_2 &= Q_1 \\ D_1 &= Q_0 \\ D_0 &= \overline{Q_2} \end{aligned}$$

$$\begin{aligned} Q_{2N} &= D_2 \\ Q_{1N} &= D_1 \\ Q_{0N} &= D_0 \end{aligned}$$



← check the loop, it's mod 6 counter.

25. MOD-4 GRAY COUNTER USING T-FF

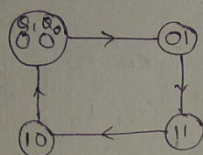
(94)

Now, let us say how to design a counter with specifications.

Designing a counter

- 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 (or) 01 (or) 10 (or) 11 state so each state has 2 bits so 2 FFs are enough

⇒ Now, these 2 FF should be connected in such a way that when a clock signal is applied it should make any of the above transitions (00 → 01, 01 → 11, 11 → 10, 10 → 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)	
Q_1	Q_0	Q_{1N}	Q_{0N}	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

→ for 2 flipflops, two inputs.

Now, to find T_1 , check Q_1 and Q_{1N}

$$T_1 = \bar{Q}_1 Q_0 + Q_1 \bar{Q}_0 = Q_1 \oplus Q_0$$

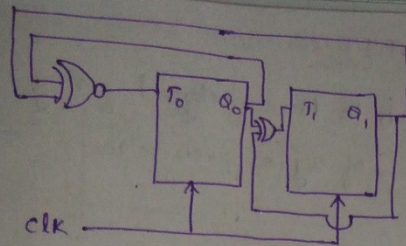
$$T_0 = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0 = Q_1 \odot Q_0$$

Q_1	Q_{1N}	T_1
0	0	0
0	1	1
1	0	0
1	1	1

→ latch

{ If Q is complemented then $T=1$, if Q is same then $T=0$ { look at function table of T-flipflop }

74

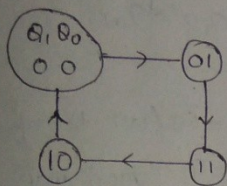


⇒ Mod 4 Gray-counter.

75

26. MOD-4 GRAY COUNTER USING D-FF

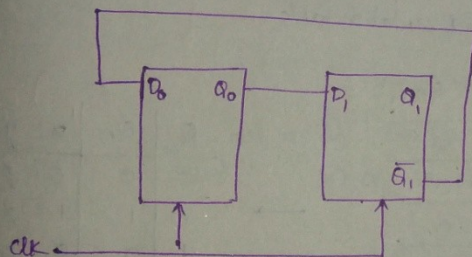
⇒ 2-flipflops are needed.



present state		Next state		Excitation (Inputs)	
Q_1	Q_0	Q_{1N}	Q_{0N}	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

$$D_1 = \bar{Q}_0 Q_1 + Q_1 Q_0 = Q_1$$

$$D_0 = \bar{Q}_1 \bar{Q}_0 + \bar{Q}_1 Q_0 = \bar{Q}_1 (\bar{Q}_0 + Q_0) = \bar{Q}_1$$



27. MOD 4 GRAY COUNTER USING 1D AND 1T FLIPFLOP.

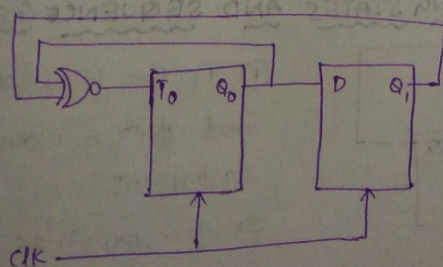
Same Question above

Q_1	Q_0	Q_{1N}	Q_{0N}	D_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

$$D_1 = Q_0$$

$$T_0 = \bar{Q}_1 \bar{Q}_0 + Q_1 Q_0$$

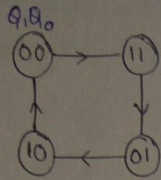
$$T_0 = Q_1 \odot Q_0$$



28. COUNTER USING TWO DIFFERENT FF'S

(76)

consider the following state diagram which is to be designed using T-Flipflops for MSB and xy for LSB. The behaviour of xy is given below:



x	y	Q _n
0	0	0
0	1	0
1	0	1
1	1	1

used for MSB so we consider Q₁ and Q₀

Q ₁	Q ₀	Q _{1n}	Q _{0n}	T ₁	x	y
0	0	1	1	1	1	0
1	1	0	1	1	0	1
0	1	1	0	1	0	0
1	0	0	0	1	0	0

Now $x = \bar{Q}_1$ (use K-map)

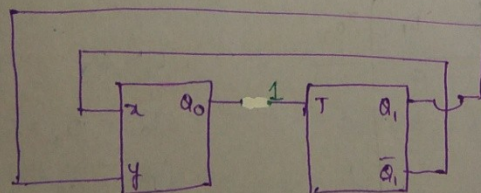
$y = Q_1$ (use K-map)

Given function table → characteristic table → Excitation table

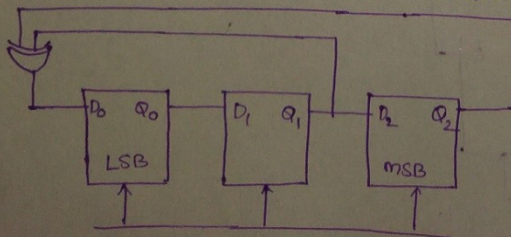
x	y	Q _n
0	0	0
0	1	0
1	0	1
1	1	1

x	y	Q ₁	Q ₀
0	0	0	0
0	1	0	0
1	0	1	1
1	1	1	1

Q ₁	Q ₀	x	y
0	0	1	0
0	1	1	0
1	0	0	1
1	1	0	1



29. MODEL ON ANALYSIS COUNTING STATES AND SEQUENCE GENERATIONS



① If the initial state is 000 and after 5 clocks what is going to happen?

② After 100 clocks what will happen?

This all depends on what initial state is.

76
flipflops

first let's analyse the state diagram.

77

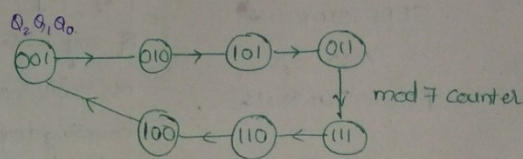
Q_2	Q_1	Q_0	Q_{2N}	Q_{1N}	Q_{0N}
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	0

map)
ap)

$$Q_{0N} = D_0 = Q_1 \oplus Q_2$$

$$Q_{1N} = D_1 = Q_0$$

$$Q_{2N} = D_2 = Q_1$$



① How many states are there is mod counter = 7.

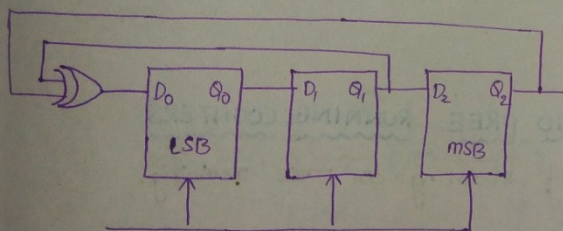
② Given Initial state = 011 then what happens after 4 clock cycles $\Rightarrow$ state after 4 clk cycles = 001

③ Initial state = 001 $\Rightarrow$ state after 10 clock cycles = 011

④ I tap the output at $Q_2 \Rightarrow$ what is the output = 0010111
write Q_2 values in all states

⑤ o/p is tapped at $Q_0 \Rightarrow$ 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) 40 MHz b) 60 MHz c) 90 MHz d) 300 MHz

$$T_{clk} \geq T_{FF} + T_{comb}$$

$$T_{clk} \geq 15 \text{ ns} + 5 \text{ ns}$$

$$T_{clk} \geq 20 \text{ ns}$$

$\Rightarrow$

$$f_{clk} \leq \frac{1}{20 \times 10^{-9}} \text{ (Time and freq are inversely proportional)}$$

$$f_{clk} \leq 50 \text{ MHz}$$

RATIONS

is so & so
not is going

will happen?
is.

76
flipflops

first let's analyse the state diagram.

77

Q_2	Q_1	Q_0	Q_{2N}	Q_{1N}	Q_{0N}
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

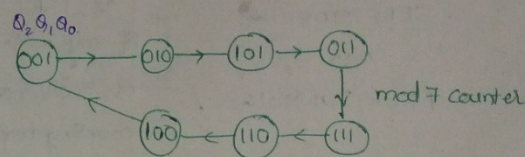
map)

ap)

$$Q_{0N} = D_0 = Q_1 \oplus Q_2$$

$$Q_{1N} = D_1 = Q_0$$

$$Q_{2N} = D_2 = Q_1$$



① How many states are there is mod counter = 7.

② Given Initial state = 011 then what happens after

4 clock cycles $\Rightarrow$ state after 4 clk cycles = 001

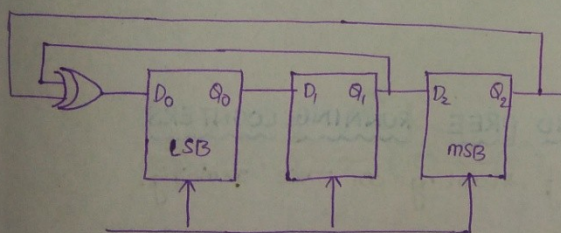
③ Initial state = 001 $\Rightarrow$ state after 10 clock cycles = 011

④ I tap the output at $Q_2 \Rightarrow$ what is the output = 0010111
write Q_2 values in all states

⑤ o/p is tapped at $Q_0 \Rightarrow$ 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) 40 MHz b) 60 MHz c) 90 MHz d) 300 MHz

$$T_{clk} \geq T_{FF} + T_{comb}$$

$$T_{clk} \geq 15 \text{ ns} + 5 \text{ ns}$$

$$T_{clk} \geq 20 \text{ ns}$$

$\Rightarrow$

$$f_{clk} \leq \frac{1}{20 \times 10^{-9}} \text{ (Time and freq are inversely proportional)}$$

$$f_{clk} \leq 50 \text{ MHz}$$

RATIONS

is so & so
not is going

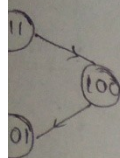
will happen?

is.

78

counting loop

steps in the



STARTING

are not entering counting loop

state you no. of clk enter the

running

S.

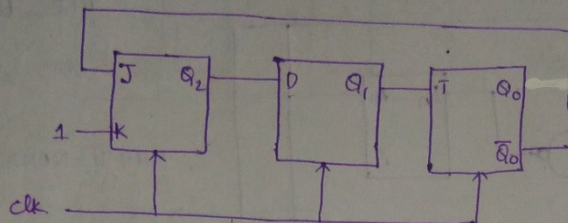
7.

mod 3 counter

33. COUNTER USING 3 DIFFERENT FLIPFLOPS

79

consider the following counter, Initially $Q_2, Q_1, Q_0 = 000$



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

$$D_1 = Q_2$$

$$Q_{2N} = ?? = JK$$

$$T_2 = Q_1$$

$$Q_{1N} = D_1 = Q_2$$

$$J_0 = Q_0$$

$$Q_{0N} = Q_0 (T=0) \Rightarrow Q_0 (Q_1=0)$$

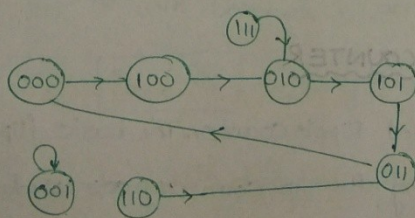
$$\bar{Q}_0 (T=1) \Rightarrow Q_0 (Q_1=1)$$

J	K	Q_N
0	1	0
1	1	$\bar{Q}_N$

$\bar{Q}_0$	K	Q_N
0	1	0
1	1	$\bar{Q}_N$

↓

Q_0	K	Q_{2N}
1	1	0
0	1	$\bar{Q}_N$



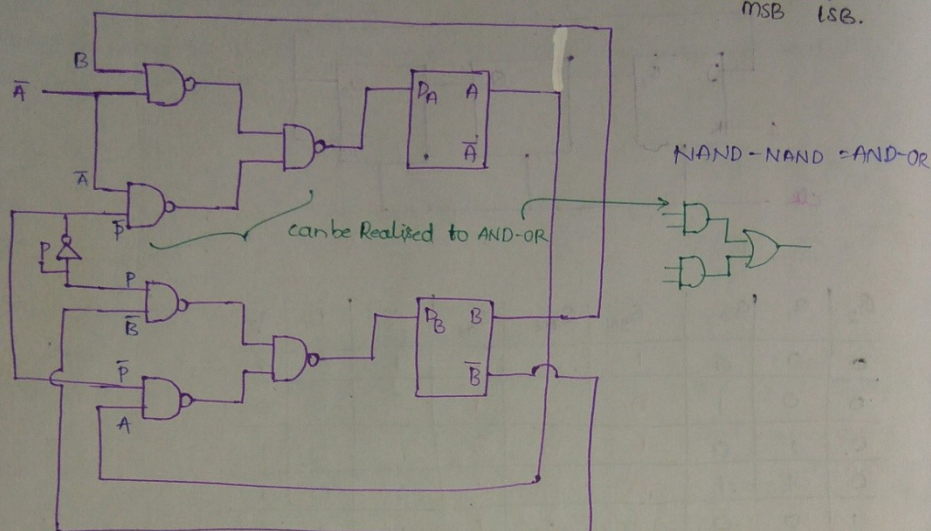
mod 5 counter.

① what will be the states after 4 clocks? a) 000 b) 010 c) 011 d) 101

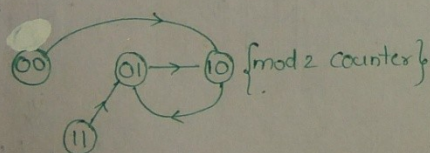
② Modulus of the Counter? a) 4 b) 5 c) 6 d) 7

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



$A_n = D_A$
 $B_n = D_B$ } If we observe the CC, they can be Realised

$$D_A = \overline{A}B + A\overline{P} \Rightarrow D_A = \overline{A}B + \overline{A} \quad \left. \begin{array}{l} \xrightarrow{\overline{A}(1+B)} \\ P=0 \end{array} \right\} = \overline{A}$$

$$D_B = P\overline{B} + \overline{P}A \Rightarrow D_B = A //$$

$$B_n = A \quad A_n = \bar{A} B$$

35. INTRODUCTION TO ASYNCHRONOUS COUNTER

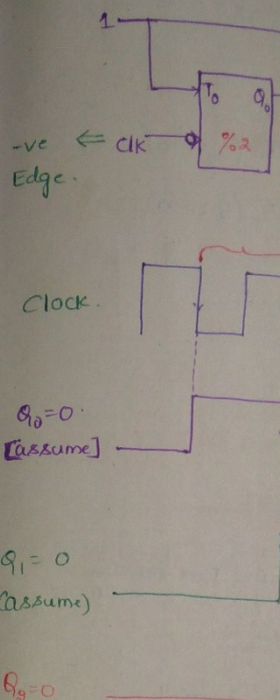
The asynchronous counters are also called Ripple Counter. The basic flipflop is T and basic counting is binary up counting. The up counters are used for implementing incrementation. Due to simpler design, Asynchronous counters are preferred in IC counter fabrication.

$\Rightarrow$ IC 7490 is a decade counter i.e. $(\%10)$ counter.

$\Rightarrow$ IC 7492 is a hexadecimal counter i.e. ($\%16$) counter.

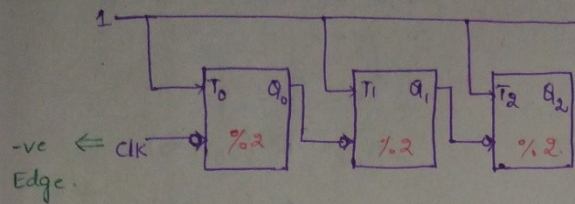
⇒ The d/p of one flipflop is going to be clock to the next FF.

36. MOD 8 UP COUNT

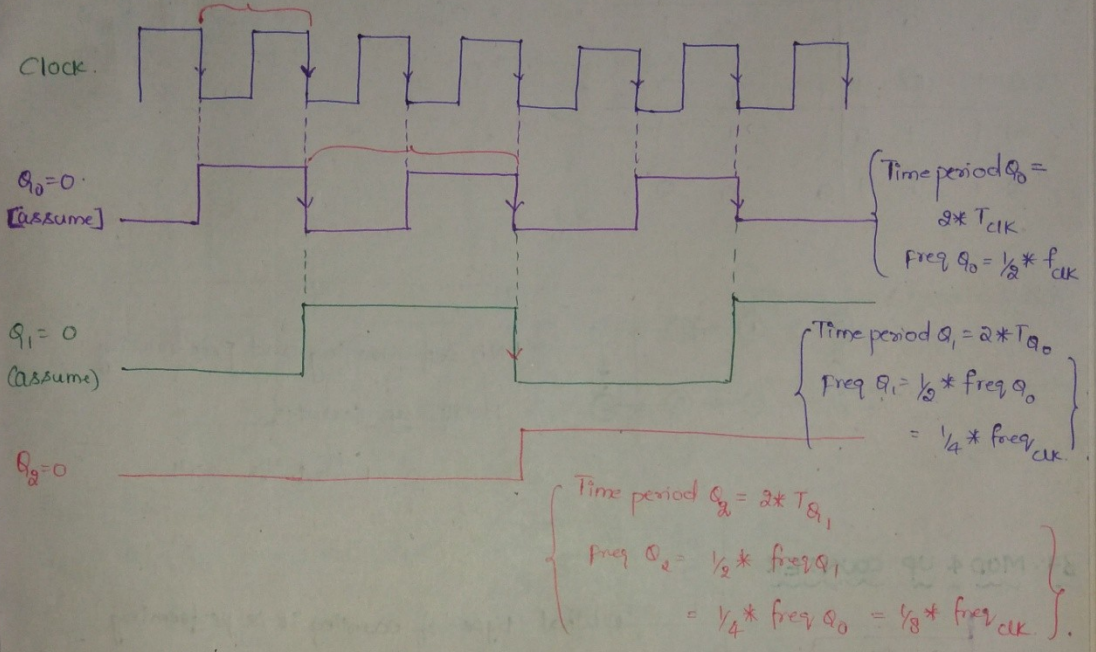


36. MOD 8 UP COUNTER

(81)



Assume $Q_0 = Q_1 = Q_2 = 0$. (Initially)



$\Rightarrow$ 3, %2 counters form %8 counter.

Now, consider 'Q0' 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 $\overline{Q_0}$.

$$\therefore Q_{0N} = \overline{Q_0} \text{ for every clock}$$

$\Rightarrow$ Now, Q_1 is changed / Q_1 depends on how Q_{0N} is providing the clock signal.

$$Q_{1N} = \overline{Q_1} \text{ (when there is -ve edge from } Q_0 \text{ (1} \rightarrow 0 \text{))}$$

$\rightarrow Q_0$ will toggle.

$\Rightarrow$ Now, $Q_{2N} = \overline{Q_2}$ ($Q_1: 1 \rightarrow 0$).

$$\therefore \begin{cases} Q_{0N} = \overline{Q_0} \\ Q_{1N} = \overline{Q_1} \text{ (} Q_0: 1 \rightarrow 0 \text{)} \\ Q_{2N} = \overline{Q_2} \text{ (} Q_1: 1 \rightarrow 0 \text{)} \end{cases}$$

$\rightarrow$ whenever a Que like above is given construct the equation and proceed.

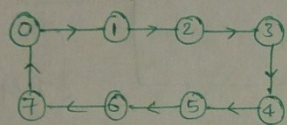
	Present state			Next State		
	Q_2	Q_1	Q_0	Q_{2N}	Q_{1N}	Q_{0N}
0)	0	0	0	0	0	1
1)	0	0	1	0	1	0
2)	0	1	0	0	1	1
3)	0	1	1	1	0	0
4)	1	0	0	1	0	1
5)	1	0	1	1	1	0
6)	1	1	0	1	1	1
7)	1	1	1	0	0	0

Above equations,

$$Q_{0N} = \bar{Q}_0$$

$$Q_{1N} = \bar{Q}_1 (Q_0: 1 \rightarrow 0)$$

$$Q_{2N} = \bar{Q}_2 (Q_1: 1 \rightarrow 0)$$

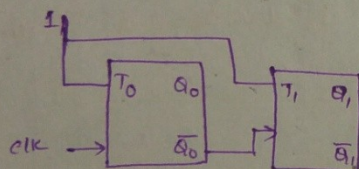


⇒ Both self starting and free running

⇒ Mod 8 up counter

$$0 \rightarrow 1 \xrightarrow{+1} 2 \xrightarrow{+1} 3 \xrightarrow{+1} 4 \dots$$

37. MOD 4 UP COUNTER



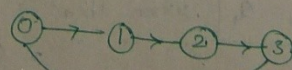
1) What type of counting it is performing and what is the type of the counter?

$$Q_{0N} = \bar{Q}_0 \text{ (for every clock)}$$

$$Q_{1N} = \bar{Q}_1 \text{ (} \bar{Q}_0: 0 \rightarrow 1 \text{)}$$

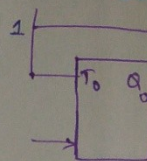
$$\text{(} \bar{Q}_0: 1 \rightarrow 0 \text{)}$$

Q_1	Q_0	Q_{1N}	Q_{0N}
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0



mod 4 up counter

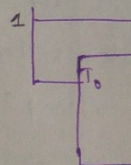
38. MOD 4 DOWN COUNTER



$$Q_{0N} = \bar{Q}_0$$

$$Q_{1N} = \bar{Q}_1 (Q_0: 1 \rightarrow 0)$$

39. MOD 8 DOWN COUNTER



$$Q_{0N} = \bar{Q}_0$$

$$Q_{1N} = \bar{Q}_1 (Q_0: 1 \rightarrow 0)$$

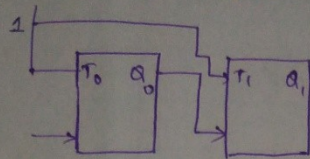
$$Q_{2N} = \bar{Q}_2 (Q_1: 1 \rightarrow 0)$$

40. APPLICATIONS

1) Shift Register

2) Counters →

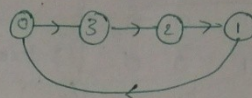
(82)

38. MOD 4 DOWN COUNTER

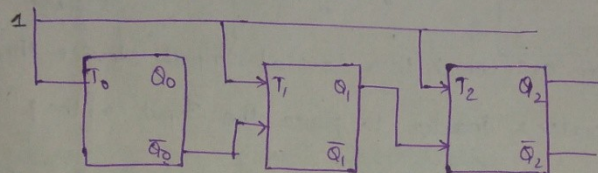
$$Q_{0N} = \overline{Q_0}$$

$$Q_{1N} = \overline{Q_1} \quad (Q_0: 0 \rightarrow 1)$$

Q_1	Q_0	Q_{1N}	Q_{0N}
0	0	1	1
0	1	0	0
1	0	0	1
1	1	1	0



mod 4 down counter

39. MOD 8 RANDOM COUNTER

If the initial state $Q_2 Q_1 Q_0$ is 101, what will be the state after 4 clock cycles?

$$= 001 = 1$$

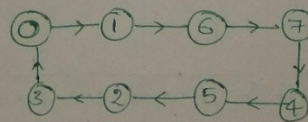
$$Q_{0N} = \overline{Q_0}$$

$$Q_{1N} = \overline{Q_1} \quad (Q_0: 0 \rightarrow 1) \Rightarrow (Q_0: 1 \rightarrow 0)$$

$$Q_{2N} = \overline{Q_2} \quad (Q_1: 0 \rightarrow 1)$$

In diagram it is positive edge so (0 → 1)
Symbol (→) = +ve edge

Q_2	Q_1	Q_0	Q_{2N}	Q_{1N}	Q_{0N}
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

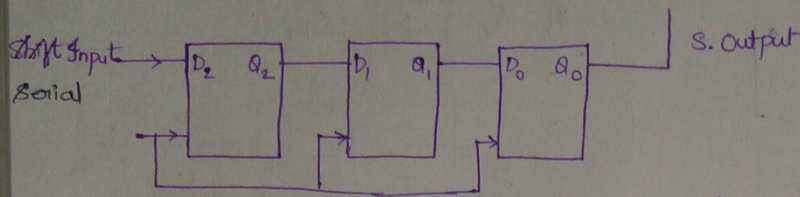


Self starting and Free Running.

40. APPLICATIONS OF FLIP FLOPS

- Shift Register → used as "Sequential memory". Ex: Accumulator in Micro processors.
- Counters → (1) used to count no. of pulses.
(2) used as frequency divider

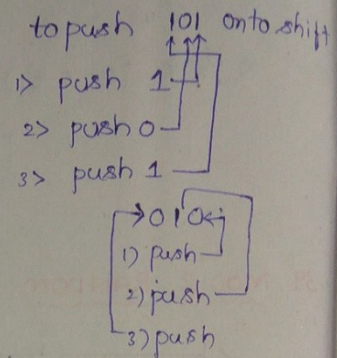
41. 3-BIT SHIFT REGISTERS (It requires 3 D-flipflops).



say I want to push 101 onto shift Register

101	clk	SI/P	Q ₂	Q ₁	Q ₀
	0	-	0	0	0
	1	1	1	0	0
	2	0	0	1	0
	3	1	1	0	1

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 ₂	Q ₁	Q ₀	o
0	1	0	1	1
1	0	1	0	0
2	0	0	1	1

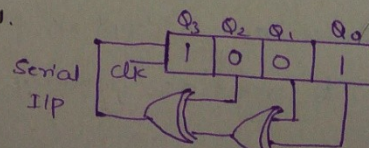
∴ To retrieve the O/p we need 2 clock pulses (1st one is counted in the pushing flip on Register).

∴ 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 flip, (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:

$$D_A = D$$

$$D_B = A$$

$$D_C = B$$

$$D_D = \bar{C}$$

44. BINA

Determining

1 onto shift

ok
sh
sh

then
e flipflop

n-bits

we need

one is
shifting flip on

Input and

clock pulses
Serial o/p.

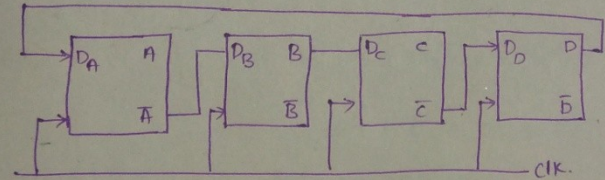
required to

CLK.	STP.	States $Q_3 \quad Q_2 \quad Q_1 \quad Q_0$	
0	-	1 0 0 1	
1	1	1 1 0 0	$\rightarrow SIP = (Q_2 \oplus Q_1 \oplus Q_0)_{\text{Previous clock}} = (0 \oplus 0 \oplus 1) = 1$
2	1	1 1 1 0	$\rightarrow SIP = (Q_2 \oplus Q_1 \oplus Q_0) = (1 \oplus 0 \oplus 0) = 1$
3	0	0 1 1 1	$\rightarrow SIP = 1 \oplus 1 \oplus 0 = 0$
4	1	1 0 1 1	$\rightarrow SIP = 1 \oplus 1 \oplus 1 = 1$
5	0	0 1 0 1	$\rightarrow SIP = 0 \oplus 1 \oplus 1 = 0$
6	0	0 0 1 0	$\rightarrow SIP = 1 \oplus 0 \oplus 1 = 0$
7	1	1 0 0 1	$\rightarrow SIP = 0 \oplus 1 \oplus 0 = 1$

7 clocks are needed.

43. EXAMPLE 2 ON SHIFT REGISTER

Initial value of ABCD = 0000, then what are the sequence of no's the following circuit is counting.
 → LSB
 → MSB.



Inputs:

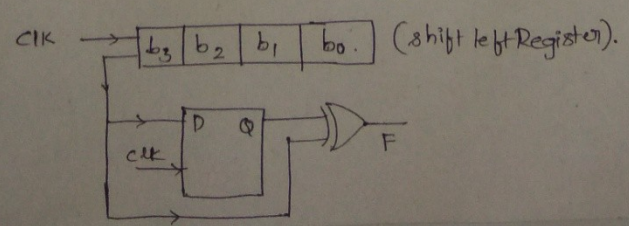
- $D_A = D$
- $D_B = \bar{A}$
- $D_C = B$
- $D_D = \bar{C}$

CLK	A	B	C	D	
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

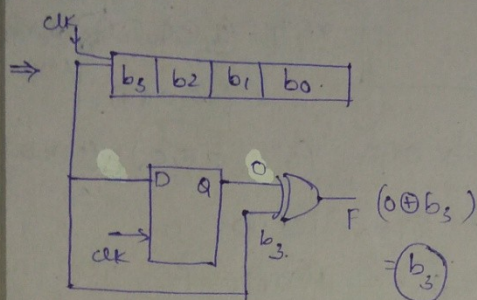
∴ 0, 5, 15, 10, 0, 5

44. BINARY TO GRAY CONVERTOR

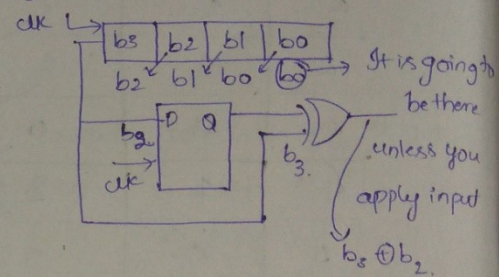
Determine the function of the following circuit



Initially assume the output of D-flipflop is '0'. Now, b_3 is given to the output of D-FF. $\therefore$ The output of XOR Gate will be $0 \oplus b_3 = b_3$ (86)



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

