

# Boolean Algebra - I

Comprehensive Course on Digital Logic Design 2023/2024



# DIGITAL LOGIC DISIGN

(CS IT)



# Syllabus

# DIGITAL LOGIC DESIGN

## 1. Basics ✓

- Boolean Algebra ✓
- Logic Gates ✓
- K-Map ✓
- Number Systems ✓

(4 - 6 M)

## 2. Combinational Circuits

- Arithmetic Circuits ✓
- Multiplexer and De-multiplexer ✓
- Decoder and Encoder ✓
- Comparator ✓
- Code Converter ✓
- Parity Generator and Checkers ✓

(1 + 1 + 2 + 2)

## 3. Sequential Circuits

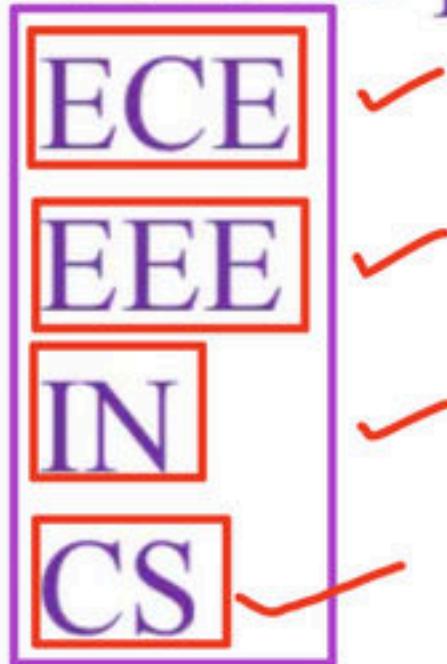
- Flip Flops ✓
- Registers ✓
- Counter's ✓
- State Machines ✓

## **UNIQUE WAY OF TEACHING**

- **BUILIDING THE STRONG CONCEPT**
- **SOLVING BASIC PROBLEM TO MAKE MORE STRONG IN CONCEPTS**
- **SOLVING PRVIOUS GATE and ESE PROBLEMS**

# Preparation Strategy

1. Class notes ✓
2. Previous paper of GATE



3. Previous Papers of ESE



# Preparation Strategy

1. Class notes
2. Previous paper of GATE

ECE

EEE

IN

CS

15

3. Previous Papers of ESE

ECE

EEE

## Things I will provide

1. Complete Notes ✓
2. Short Notes ✓
3. DPPs with all PYQs
4. My Contact No :

93980 21419

**SOLVE ALL THE DPPs**

## Digital Short Notes

Follow me @uncademy/bvreddy  
for the full length course

93980 21419

Use the Code: **BVREDDY**, to get maximum discount ,  
complete notes ,DDPs and Short Notes

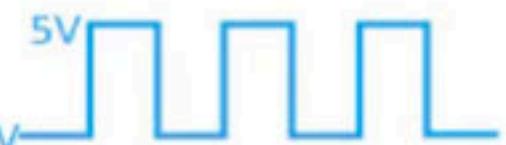
## Positive logic system

High voltage corresponds to logic “1”

Maximum positive value is taken as logic ‘1’

+5V ----> logic “1”

0V ----> logic “0”



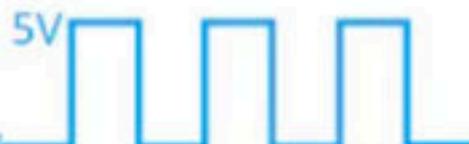
## Negative logic system

High voltage corresponds to logic “0”

Maximum positive value is taken as logic ‘0’

+5V ----> logic “0”

0V ----> logic “1”



A positive logic system is converted into negative logic system by using the concept of duality

### Finding the dual of a given Boolean expression

1.  $* \leftrightarrow +$

2.  $0 \leftrightarrow 1$

3. Keep the variables as it is

**BVREDDY**

### OR -Operation

$$A + 0 = A$$

$$1 + A = 1$$

$$A + A = A$$

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

$$A * 1 = A$$

$$A * 0 = 0$$

$$A * A = A$$

$$A * \bar{A} = 0$$

BVREDDY

### Transposition theorem ( T- 2 )

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

### Consensus theorem ( Rajinikanth wala )

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

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

**AND-Operation**

$$A + B = B + A$$

$$A * B = B * A$$

### Distribution Law

(Mingle wala)

$$A(B+C) = AB+AC$$

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

$$A + B = B + A$$

$$A * B = B * A$$

### Associative Law

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

$$A * B * C = (A * B) * C = (B * C) * A = (C * A) * B$$

### D- Morgan's Law

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

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

### Transposition theorem ( T- 1 )

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

**Literal** : A Boolean variable either in normal form (or ) complimented form is known as literal

**Minterm** : Each term in canonical SOP representation is known as minterm

**Maxterm**: Each term in canonical POS representation is known as maxterm

**BVREDDY**

**Canonical form** : Each minterm ( maxterms ) contains all the Boolean variables

$$F(A, B, C) = ABC + \bar{A}BC + AB\bar{C} \rightarrow \text{SOP}$$

$$F(A, B, C) = (A+B+C)(A+\bar{B}+C)(\bar{A}+B+\bar{C}) \rightarrow \text{POS}$$

**Minimal Form** : The minimized form of Boolean expression

$$F(A, B, C) = BC + AB$$

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

1. Maximum possible minterms =  $2^n$

BVREDDY

2. Maximum possible maxterms =  $2^n$

3. Number of minterm's + number of maxterm's =  $2^n$

4. The sum of all the minterms = **ONE**

5. The product of all maxterms = **ZERO**

6. Minterm's and maxterm's of same index are **compliment** to each other

7. By using 2- Boolean variables total number of possible Boolean functions = 16

8. By using n- Boolean variables total number of possible Boolean functions =  $2^{2^n}$

9. By using 2- Boolean variables total number of possible Boolean functions having at most 3- minterms =  $4c_0 + 4c_1 + 4c_2 + 4c_3 = 15$

10. By using 2- Boolean variables total number of possible Boolean functions having at most 3- maxterms = 15

11. By using 2- Boolean variables total number of possible Boolean functions having 3- minterms =  $4c_3 = 4$

12. By using n- Boolean variables total number of possible Boolean functions having 2- minterms =  $2^n c_2$

13. By using 5- Boolean variables total number of possible Boolean functions having at most 3- minterms =  $32c_0 + 32c_1 + 32c_2 + 32c_3$

BVREDDY

**Neutral Function :**

The number of minterms = number of maxterms

**Mutually exclusive terms**

The mutually exclusive term of  $m_i$  is  $m_{2^n-i-1}$

**Self Dual Expression**

If one time dual of the Boolean expression result the same expression , then it is called as self dual expression

Eg :  $f = AB+BC+AC$

**Conditions for the given expression is self dual**

1. The number of minterms = number of maxterms (Neutral Function)  
number of minterms+ number of maxterms =  $2^n$   
number of minterms = number of maxterms =  $2^{n-1}$

2. If  $m_i$  belongs to f , then  $m_{2^n-i-1}$  should belongs to  $\bar{f}$

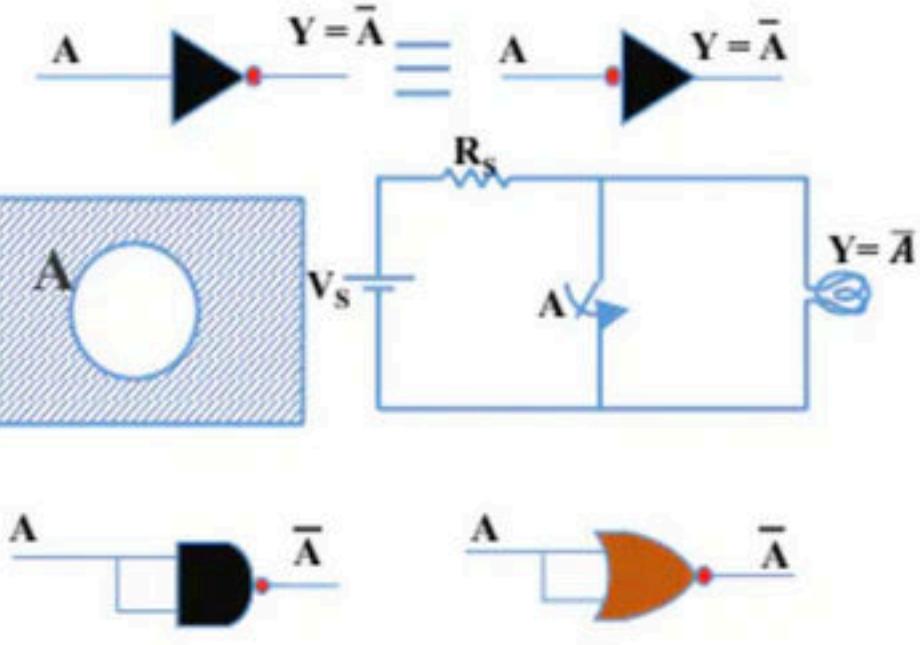
3. The number of self dual functions =  $2^{2^{n-1}}$

**Use the Code: BVREDDY, to  
get maximum discount ,  
complete notes ,DDPs and  
Short Notes**

### NOT GATE

$$Y = \bar{A}$$

The output is the complement of the input



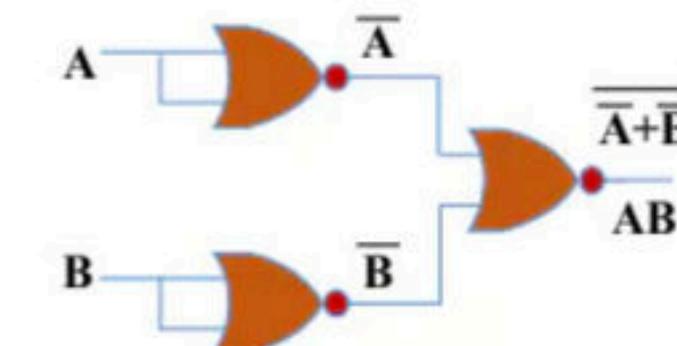
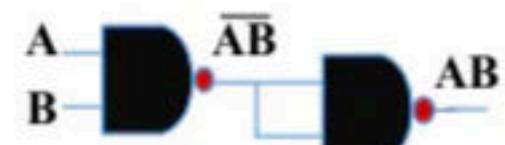
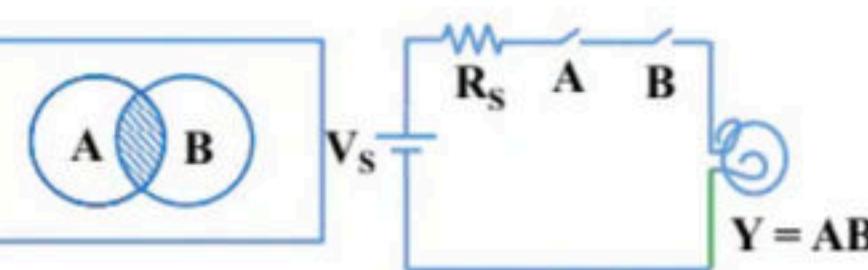
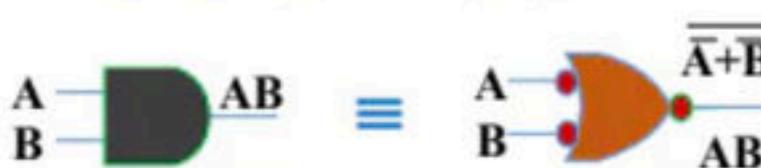
**Use the Code:  
BVREDDY, to get  
maximum discount,  
complete notes ,DDPs  
and Short Notes**

### AND GATE

$$Y = AB$$

**BVREDDY**

- Output is '0' if any one input '0'
- $Y = AB = \Sigma(3) = \Pi(0, 1, 2)$
- Enable input  $\Rightarrow 1$
- Disable input  $\Rightarrow 0$
- Commutative law  $\Rightarrow$  Obeys
- Associative law  $\Rightarrow$  Obeys

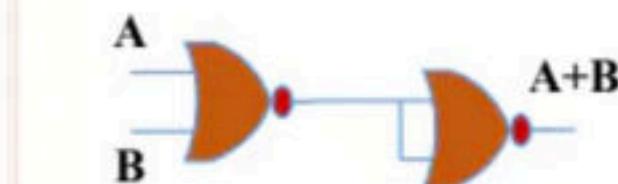
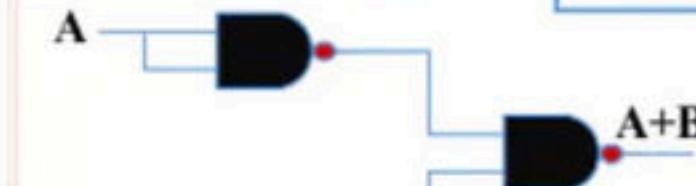
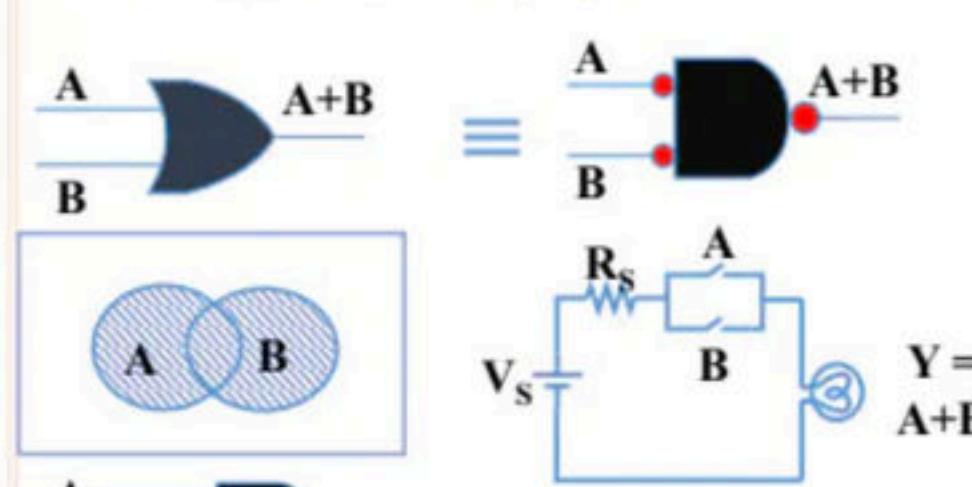


### OR GATE

$$Y = A+B$$

**BVREDDY**

- Output is '1' if anyone of the inputs are '1'
- $Y = A+B = \Sigma(1, 2, 3) = \Pi(0)$
- Enable input  $\Rightarrow 0$
- Disable input  $\Rightarrow 1$
- Commutative law  $\Rightarrow$  Obeys
- Associative law  $\Rightarrow$  Obeys

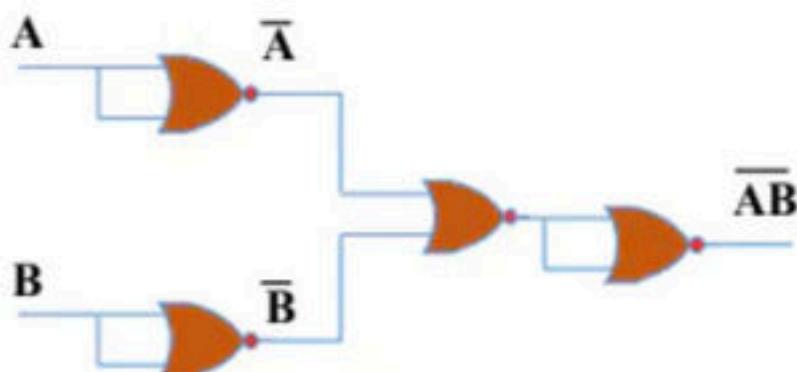
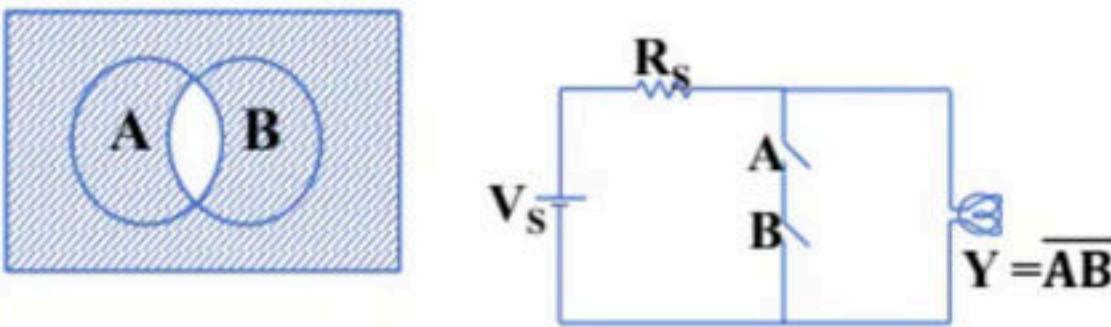
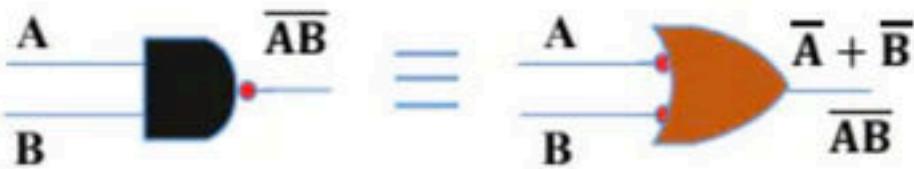


**BVREDDY**

## NAND GATE

$$Y = \overline{AB}$$

- Output is '1' if any one input is '0'
- $Y = \overline{AB} = \sum(0, 1, 2) = \prod(3)$
- Enable input --1
- Disable input-- 0
- Commutative law ---> Obeys
- Associative law ----> not Obeys

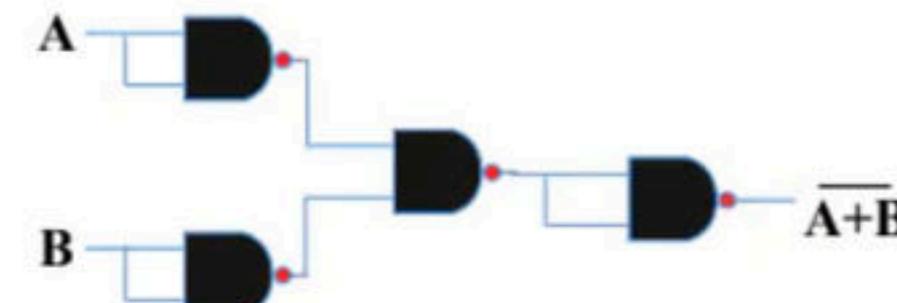
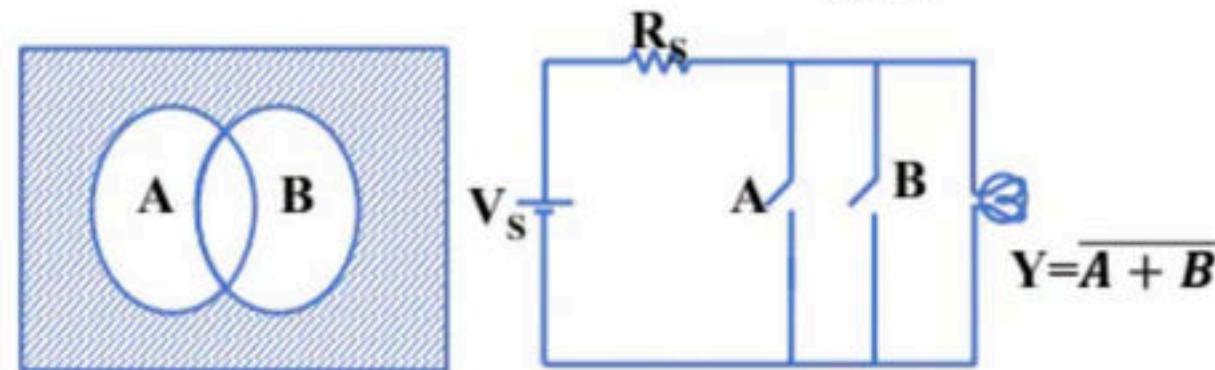
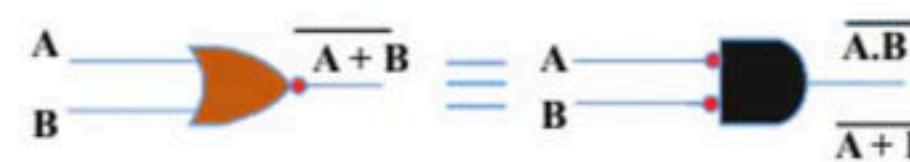


BVREDDY

## NOR- GATE

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

- Output is '0' if any one of the input is '1'
- $Y = \overline{A + B} = \sum(0) = \prod(1, 2, 3)$
- Enable input --0
- Disable input- 1
- Commutative law ---> Obeys
- Associative law ----> not Obeys



BVREDDY

## EX-OR GATE

- Output is '1' for odd number of '1's in the input
- $Y = A \oplus B = \sum(1, 2) = \Pi(0, 3)$
- $Y = A \oplus B \oplus C = \sum(1, 2, 4, 7)$
- $Y = A \oplus B \oplus C \oplus D = \sum(1, 2, 4, 7, 8, 11, 13, 14)$

➤ Commutative law  $\Rightarrow$  Obeys

➤ Associative law  $\Rightarrow$  Obeys

➤  $A \oplus 0 = A$

➤  $A \oplus 1 = \bar{A}$

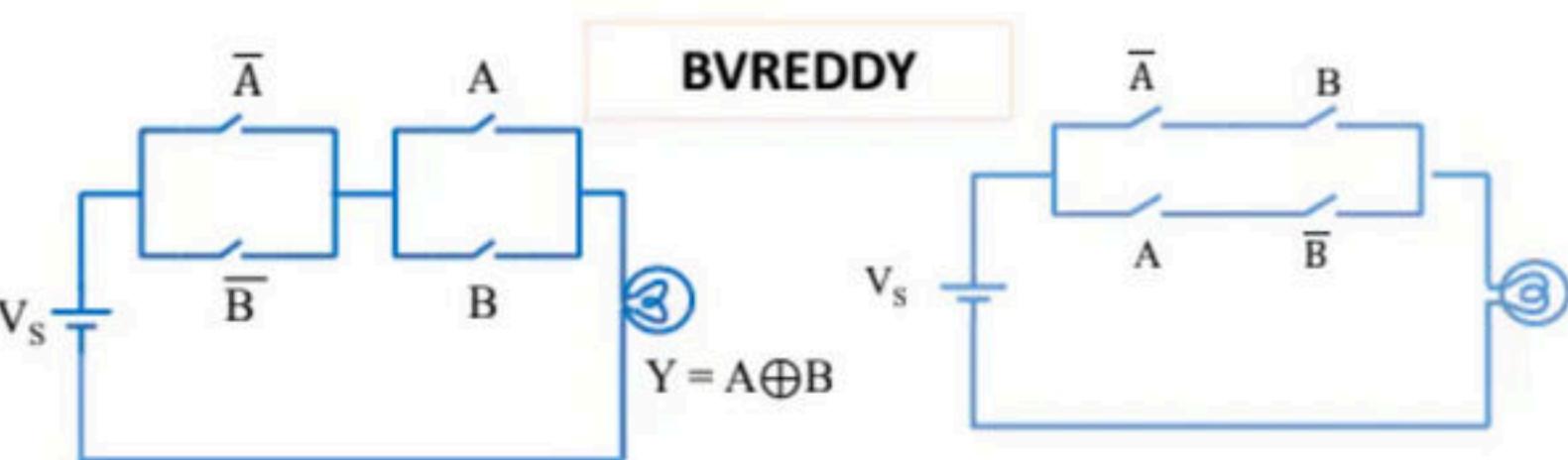
➤  $A \oplus A = 0$

➤  $A \oplus \bar{A} = 1$

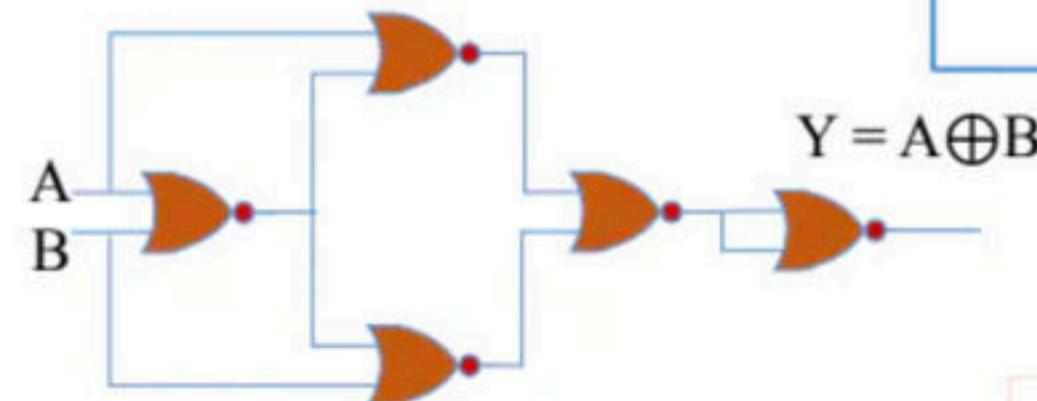
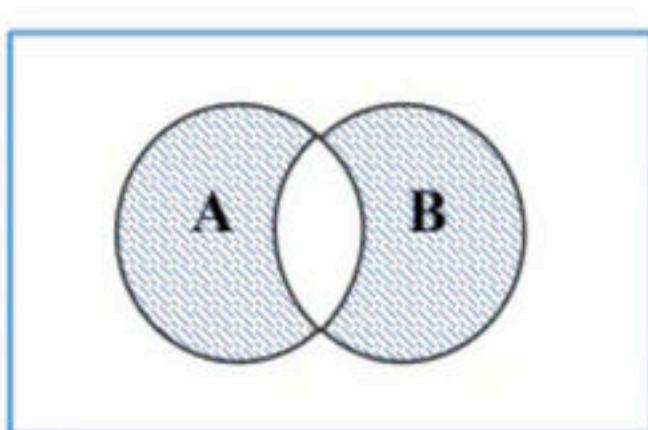
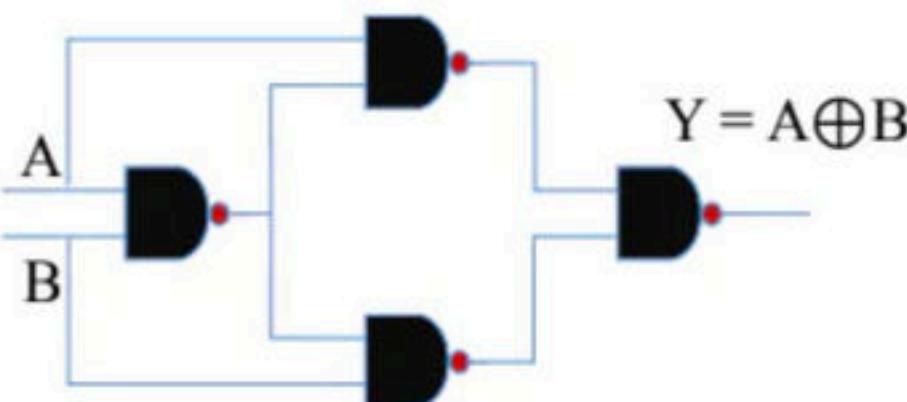
➤  $A \oplus A \oplus A \oplus \dots \text{ n times} = \begin{cases} A, & n \text{ is odd} \\ 0, & n \text{ is even} \end{cases}$

➤  $A \oplus \bar{A}B = A + B$

➤  $AB \oplus BC = B(A \oplus C)$



**BVREDDY**



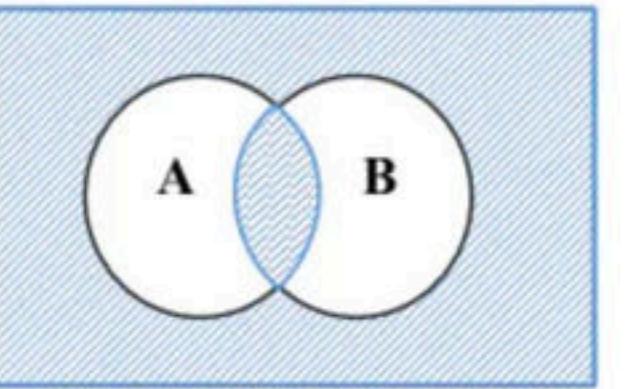
**BVREDDY**

Use the Code: **BVREDDY**, to get  
maximum discount,  
complete notes ,DDPs and Short Notes

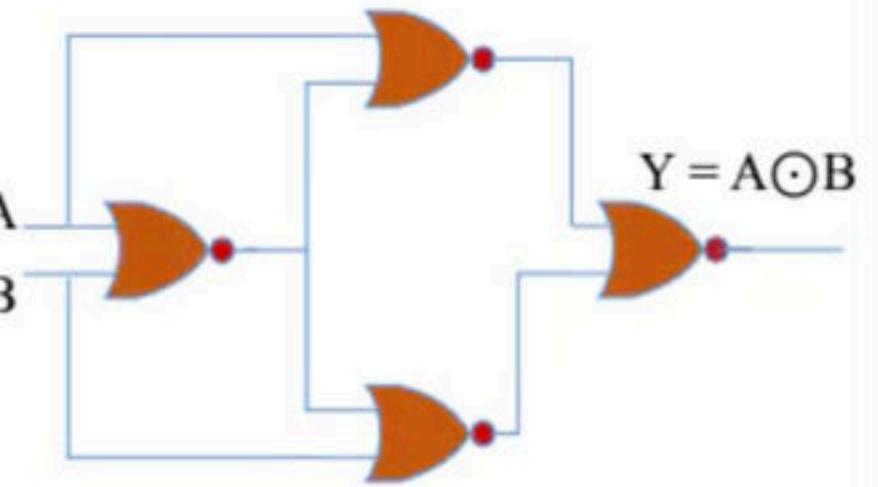
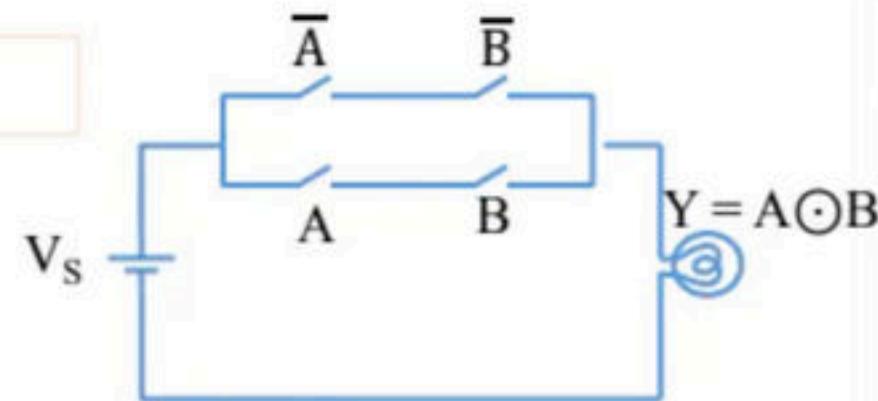
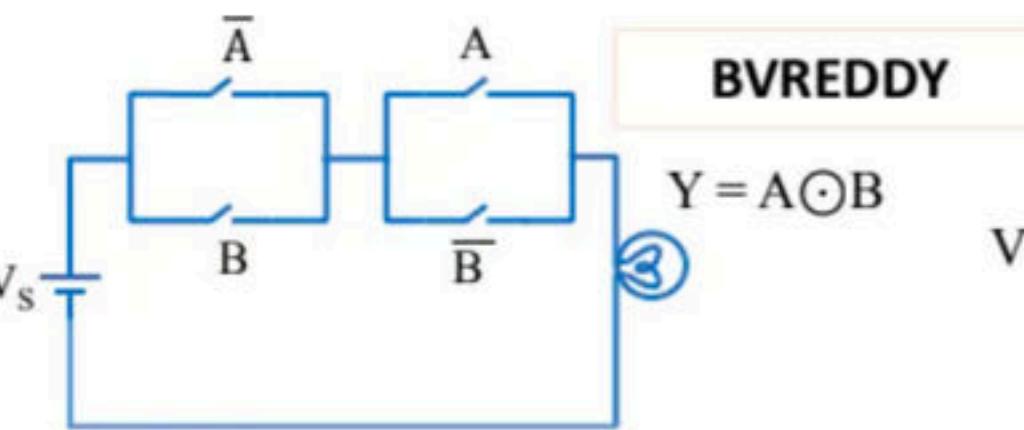
## EX-NOR GATE

- Output is '1' for even number of '1's in the input
- $Y = A \odot B = \sum(0,3) = \Pi(1,2)$
- Commutative law  $\Rightarrow$  Obeys
- Associative law  $\Rightarrow$  not Obeys
- $A \odot 0 = \bar{A}$
- $A \odot 1 = A$
- $A \odot A = 1$
- $A \odot \bar{A} = 0$
- $A \odot A \odot A \odot \dots \text{n times} = \begin{cases} \bar{A}, & n \text{ is odd} \\ 1, & n \text{ is even} \end{cases}$
- $\overline{A \odot B} = A \oplus B$
- $A \oplus \bar{B} = A \odot B$
- $\bar{A} \oplus B = A \odot B$
- $\bar{A} \oplus \bar{B} = A \oplus B$
- $A \odot B \odot C = \sum(0,3,5,6)$
- $A \oplus B \oplus C = \sum(1,2,4,7)$
- $(A \odot B) \odot C = \sum(1,2,4,7)$
- $(A \odot C) \odot B = \sum(1,2,4,7)$
- $A \oplus B \oplus C = (A \odot B) \odot C = (A \odot C) \odot B$
- $A \odot B = \bar{A} \oplus B = A \oplus \bar{B} = \bar{A} \odot \bar{B}$
- $A \oplus B = A \odot \bar{B} = \bar{A} \odot B = \bar{A} \oplus \bar{B}$
- $\overline{A \oplus B \oplus C} = A \odot B \odot C = [A \oplus B] \odot C = A \odot [B \oplus C]$

BVREDDY



BVREDDY



## EX-OR GATE

Output is '1' for odd number of '1's in the input

Odd number of 1's detector

Inequality detector

Anti-coincident gate

## EX-NOR GATE

Output is '1' for even number of '1's in the input

Even number of 1's detector

Equality detector

Coincident gate

	No. of NAND GATES	No. of NOR GATES
NOT	1	1
AND	2	3
OR	3	2
EX-OR	4	5
EX-NOR	5	4
NAND	1	4
NOR	4	1

- For a n-variable Boolean expression , the maximum number of literals = n
- For a n-variable K-Map if group is done by considering  $2^m$  number of cells , then the resulting term from that group contains ( n-m ) number of literals .
- 8 cells –  $2^3$  cells → Octet --> 3 variables eliminated
- 4 cells –  $2^2$  cells → Quad --> 2 variables eliminated
- 2 cells –  $2^1$  cells → Pair --> 1 variables eliminated
- Minimal expression may not be unique .
- The minimal expression = ( All EPI's ) + ( Optional PI's )
- If all PI's are EPI's , then the minimal expression is unique
- The sufficient condition for a K-map to have unique solution is number of PI's = number of EPI's

Use the Code :  
BVREDDY

## K- Map

**Implicant** : Each minterm in canonical SOP expression is known as Implicant .

**Prime Implicant** is a product term , obtained by combining maximum possible cells in the K-Map. While doing so make sure that a smaller group is not completely inside a bigger group .

**Essential Prime Implicant** : A prime Implicant is an EPI , if and only if it contains at least one minterm which is not covered by multiple groups

All EPI's are PI's , but vice versa not true

$EPI \leq PI$

		Minterm mode				Maxterm mode			
		$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$	$C+D$	$C+\bar{D}$	$\bar{C}+\bar{D}$	$\bar{C}+D$
AB	CD	$\bar{A}\bar{B}\bar{C}\bar{D}$	$\bar{A}\bar{B}\bar{C}D$	$\bar{A}\bar{B}CD$	$\bar{A}\bar{B}C\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$	$A+B+C+\bar{D}$	$A+B+\bar{C}+\bar{D}$	$A+\bar{B}+\bar{C}+D$
		0	1	3	2	4	5	7	6
AB	CD	$\bar{A}B\bar{C}\bar{D}$	$\bar{A}B\bar{C}D$	$\bar{A}BC\bar{D}$	$\bar{A}B\bar{C}\bar{D}$	$A+\bar{B}+\bar{C}+D$	$A+\bar{B}+C+\bar{D}$	$A+\bar{B}+\bar{C}+\bar{D}$	$A+\bar{B}+\bar{C}+D$
		12	13	15	14	8	9	11	10
AB	CD	$AB\bar{C}\bar{D}$	$AB\bar{C}D$	$ABC\bar{D}$	$ABC\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+D$	$\bar{A}+\bar{B}+C+\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$	$\bar{A}+\bar{B}+C+D$
		1	2	4	3	5	6	7	8
AB	CD	$AB\bar{C}D$	$AB\bar{C}\bar{D}$	$ABC\bar{D}$	$ABC\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$	$\bar{A}+\bar{B}+C+\bar{D}$	$\bar{A}+\bar{B}+\bar{C}+\bar{D}$	$\bar{A}+\bar{B}+C+D$
		10	11	13	12	15	14	16	17

Use the Code : BVREDDY ,to get the maximum discount

## Number systems

- Base (b) is always a positive integer .
- In general  $b \geq 0$

Base	Different digits
2 ( Binary )	0 , 1
8( Octal )	0,1,2,3,4,5,6,7
10 ( Decimal )	0,1,2,3,4,5,6,7 ,8,9
16 (Hexadecimal)	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

## r's Complement

BVREDDY

r's Complement of the number (N) =  $r^n - N$

r -----> Radix

n -----> number of integer digits

N -----> given number

## (r-1)'s Complement

Use the Code :  
BVREDDY

(r-1)'s Complement of the number (N) =  $r^n - r^{-m} - N$

r -----> Radix

n -----> number of integer digits

m -----> number of decimal digits

N -----> given number

BVREDDY

(r-1)'s Complement of the number (N) =  $r^n - r^{-m} - N$

r's Complement of the number (N) = (r-1)'s complement +  $r^{-m}$   
if m= 0

r's Complement of the number (N) = (r-1)'s complement + 1

## Unsigned Number Representation

BVREDDY

- Strictly applicable for positive numbers
- There is no sign bit concept
- + 5 -----> 101
- 5 -----> not allowed
- Range = 0 to  $2^n - 1$

## Signed Magnitude representation

- Valid for both positive and negative numbers .
- Sign bit concept is used .



Sign bit = 0 , for  $\oplus$ Ve number

= 1, for  $\ominus$ ve number

Range = -  $(2^{n-1} - 1)$  to  $+(2^{n-1} - 1)$

Use the Code :  
BVREDDY

## 1's Compliment representation

In this  $\oplus$ Ve numbers are represented as normal binary number with MSB '0'

**BVREDDY**

### Representation of $\ominus$ ve number

1. Write the binary equivalent of magnitude
2. Take its 1's compliment
- Range = -  $(2^{n-1} - 1)$  to +  $(2^{n-1} - 1)$

### Overflow

Over flow occurs in signed arithmetic operations if two same sign numbers are added and result exceeds with given number of bits . Overflow can be avoided by taking extra bits

#### 1.By using carry bits

$C_{in}$  -----> carry into MSB

$C_{out}$  -----> carry out from MSB

if  $C_{in} \oplus C_{out} = 0$  , no overflow occurs

$C_{in} \oplus C_{out} = 1$  , over flow occurs

#### 2. By using Sign Bits

X -----> Sign bit of 1<sup>st</sup> number

Y -----> Sign bit of 2<sup>nd</sup> number

Z-----> Sign bit of Resultant

$$\text{Over flow} = XYZ + \bar{X}\bar{Y}\bar{Z}$$

## 2's Compliment representation

In this  $\oplus$ Ve numbers are represented as normal binary number with MSB '0'

### Representation of $\ominus$ ve number

1. Write the binary equivalent of magnitude
2. Take its 2's compliment
- Range = -  $(2^{n-1})$  to +  $(2^{n-1} - 1)$

## BCD (Binary Coded Decimal)Code

In this code each decimal number is represented by a separate group of 4- bits

- It uses only 0 to 9
- 0 to 9 are valid BCD Code
- 10, 11, 12 , 13 , 14,15 are invalid BCD Code
- Coding method is very simple but it requires more number of bits .

### EX-3 Code

The EX-3 code can be derived from the natural BCD code by adding 3 to each coded number

Valid EX -3 : 3 ,4,5,6,7,8,9,10,11,12

Invalid EX-3 : 0,1,2,13,14,15

### Gray Code

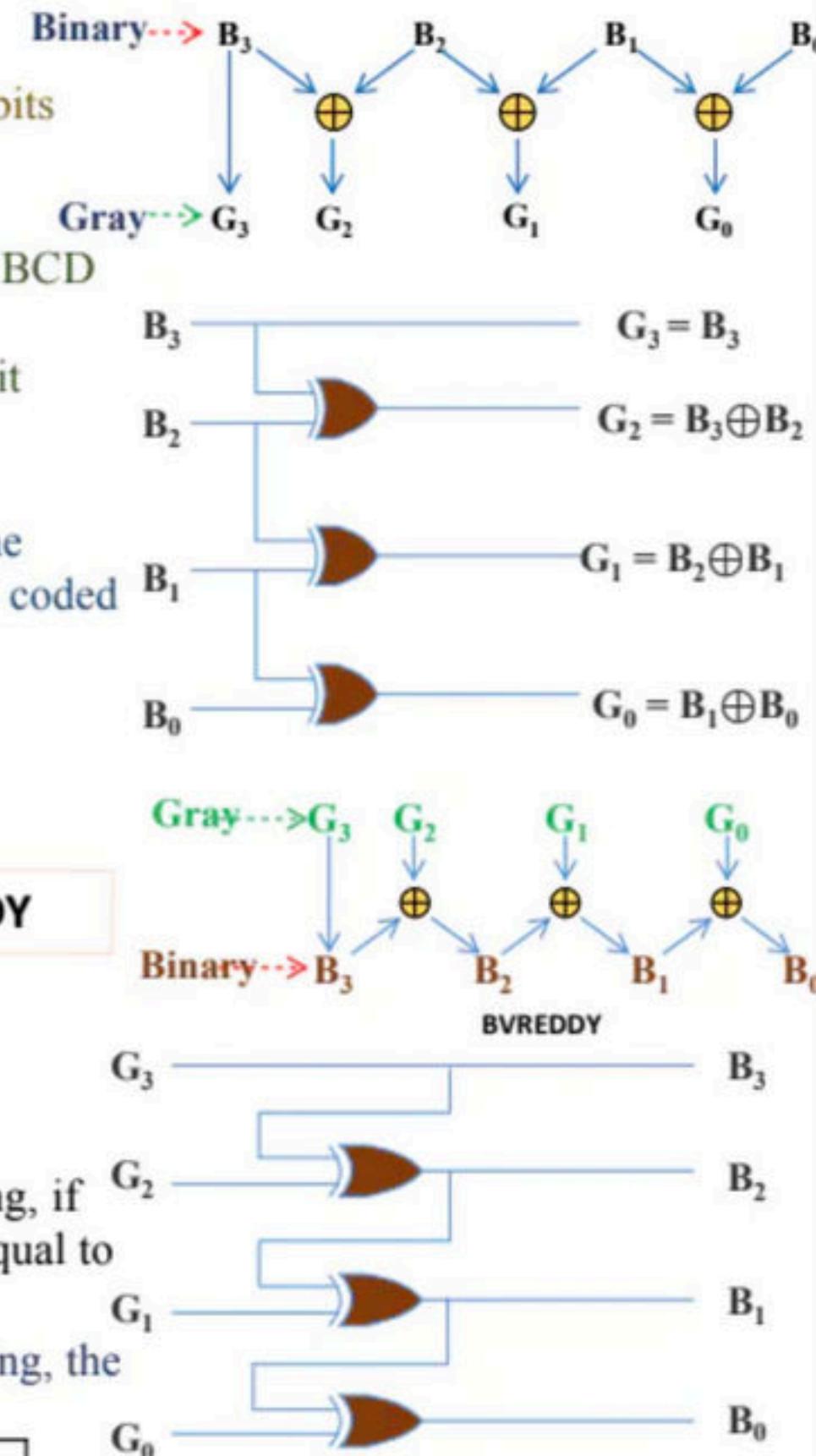
- Non weighted code
- Unit distance code
- Cyclic code
- Reflective code
- Minimum error code

**BVREDDY**

## SELF COMPLEMENTING CODE

A code is said to be self complementing, if the 1' complement of a number N is equal to the 9's complement of the number.

- For a code to be self complementing, the sum of all its weights must be 9 .



**HA****BVREDDY**

- Logical expression for Sum =  $A \oplus B$
- Logical expression for Carry =  $AB$
- Minimum number of NAND Gates = 5
- Minimum number of NOR Gates = 5

**FA**

- Logical expression for Sum =  $A \oplus B \oplus C$
- Logical expression for Carry =  $AB + (A \oplus B)C$
- Minimum number of NAND Gates = 9
- Minimum number of NOR Gates = 9

**HS**

- Logical expression for Difference =  $A \oplus B$
- Logical expression for Barrow =  $\bar{A}B$
- Minimum number of NAND Gates = 5
- Minimum number of NOR Gates = 5

**FS**

- Logical expression for Difference =  $A \oplus B \oplus C$
- Logical expression for Barrow =  $\bar{A}B + (\bar{A} \oplus B)C$
- Minimum number of NAND Gates = 9
- Minimum number of NOR Gates = 9

**Half Adder**

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

**Full Adder**

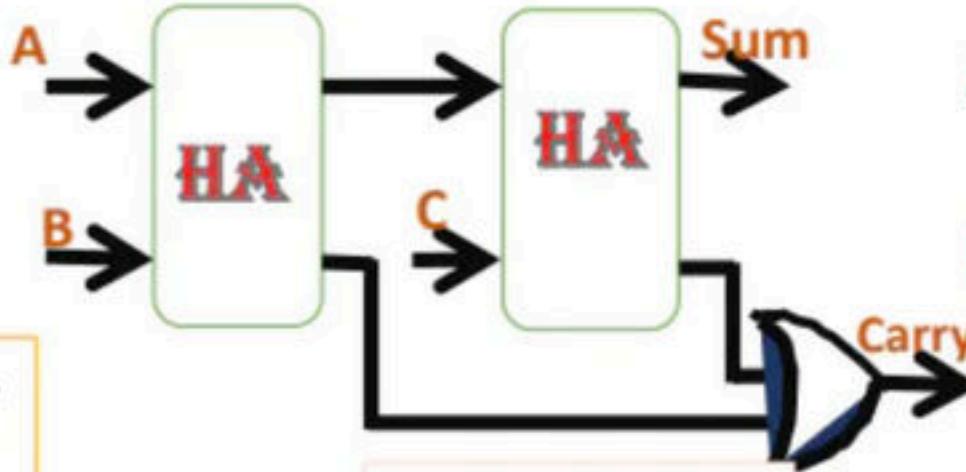
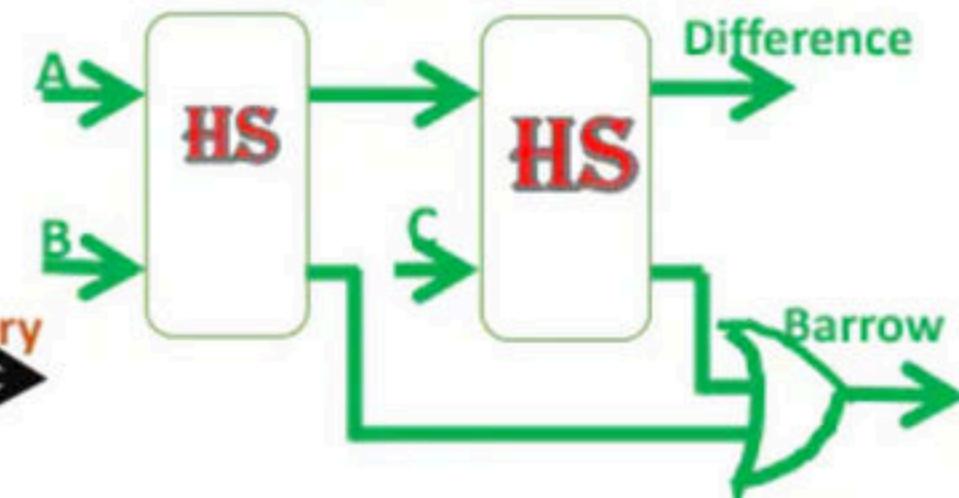
A	B	C	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

**Full Subtractor**

A	B	C	Difference	Barrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

**Half Subtractor**

A	B	Difference	Barrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

**BVREDDY****Use the Code : BVREDDY****BVREDDY**

**FS : A- B- C**

$$\text{Difference} = A \oplus B \oplus C$$

$$\begin{aligned}\text{Barrow} &= \bar{A}B + (\bar{A} \oplus B)C \\ &= \bar{A}B + \bar{A}C + BC\end{aligned}$$

**FS : B- C- A**

$$\text{Difference} = A \oplus B \oplus C$$

$$\begin{aligned}\text{Barrow} &= \bar{B}C + (\bar{B} \oplus C)A \\ &= A\bar{B} + \bar{B}C + AC\end{aligned}$$

**FS : C- A- B**

$$\text{Difference} = A \oplus B \oplus C$$

$$\begin{aligned}\text{Barrow} &= \bar{C}A + (\bar{C} \oplus A)B \\ &= A\bar{C} + B\bar{C} + AB\end{aligned}$$

**Binary Multiplier**

Number of AND gates required =  $m \times n$

Number of Adders required =  $m+n-2$

$m \longrightarrow$  number of bits in A

$n \longrightarrow$  number of bits in B

In general for n-bit Parallel Adder

$$\text{Worst case Delay} = (n-1)(t_{pd})_{\text{carry}} + \text{Max(sum, carry)}$$

**Look Ahead Carry Adder**

- In this adder, the carry dependency of Ripple Carry Adder (RCA) is eliminated
- This is the fastest adder among all
- This adder have the maximum complexity

**Hardware Requirements**

$$L1 : n - \text{XOR} + n - \text{AND}$$

$$L2 : \frac{n(n+1)}{2} - \text{AND}$$

$$L3 : n - \text{OR}$$

$$L4 : n - \text{XOR}$$

$$\text{Total number of gates for carry} = 3n + \frac{n(n+1)}{2}$$

$$\text{Total number of gates for sum} = 4n + \frac{n(n+1)}{2}$$

$$\text{Worst delay for Carry} = \text{Max(xor, and)} + (t_{pd})_{\text{and}} + (t_{pd})_{\text{or}}$$

$$\text{Worst case delay for Sum} = \text{Max(xor, and)} + (t_{pd})_{\text{and}} + (t_{pd})_{\text{or}} + (t_{pd})_{\text{xor}}$$

**BVREDDY****For n-bit Magnitude Comparator**

Total number of input combinations =  $2^{2n}$

Lesser than combinations =  $\frac{2^{2n} - 2^n}{2}$

Greater than combinations =  $\frac{2^{2n} - 2^n}{2}$

Equal combinations =  $2^n$

**BVREDDY****For 3-bit magnitude comparator**

$$Y_1(A < B) = \bar{a}_2 b_2 + (a_2 \odot b_2) \bar{a}_1 b_1 + (a_2 \odot b_2) (a_1 \odot b_1) \bar{a}_0 b_0$$

$$Y_2(A = B) = (a_2 \odot b_2) (a_1 \odot b_1) (a_0 \odot b_0)$$

$$Y_3(A > B) = a_2 \bar{b}_2 + (a_2 \odot b_2) a_1 \bar{b}_1 + (a_2 \odot b_2) (a_1 \odot b_1) a_0 \bar{b}_0$$

**For 4-bit Magnitude Comparator**

$$Y_1(A < B) = \bar{a}_3 b_3 + (a_3 \odot b_3) (\bar{a}_2 b_2) + (a_3 \odot b_3) (a_2 \odot b_2) (\bar{a}_1 b_1) + (a_3 \odot b_3) (a_2 \odot b_2) (a_1 \odot b_1) (\bar{a}_0 b_0)$$

$$Y_2(A = B) = (a_3 \odot b_3) (a_2 \odot b_2) (a_1 \odot b_1) (a_0 \odot b_0)$$

$$Y_3(A > B) = a_3 \bar{b}_3 + (a_3 \odot b_3) (a_2 \bar{b}_2) + (a_3 \odot b_3) (a_2 \odot b_2) (a_1 \bar{b}_1) + (a_3 \odot b_3) (a_2 \odot b_2) (a_1 \odot b_1) (a_0 \bar{b}_0)$$

**BVREDDY**

## Multiplexer (MUX)

- Data selector
- Many to one
- Universal logic gate
- Parallel to serial converter  
 $2^n \times 1$

BVREDDY

$2^n$  -----> number of data inputs  
 $n$  -----> number of select inputs  
 $1$  -----> number of outputs

## Demultiplexer

- One input to many output
- Data distributor
- One to many circuit  
 $1 \times 2^n$

$n$  -----> number of select lines  
 $2^n$  -----> number of output lines  
 $1$  -----> number of inputs

BVREDDY

## Decoder

Decoder is a multi input ,multi output logic circuit which converts coded input into coded output , where the input and output codes are different

 $n \times 2^n$ 

$n$  -----> number of inputs  
 $2^{2n}$  -----> number of outputs

Decoder is a special case of Demultiplexer , in which the select lines or Demultiplexer are treated as input's to the decoder and input of Demultiplexer is treated as Enable input of the Decoder

**Inputs**  $\longleftrightarrow$  **Enable**  
**Select lines**  $\longleftrightarrow$  **Inputs**

Logic Gate	Number of MUX required
BUFFER	1
NOT	1
AND	1
OR	1
NAND	2
NOR	2
EX-OR	2
EX-NOR	2
HA	3
HS	2

## Encoder

Encoder is a combinational circuit , which is used to convert

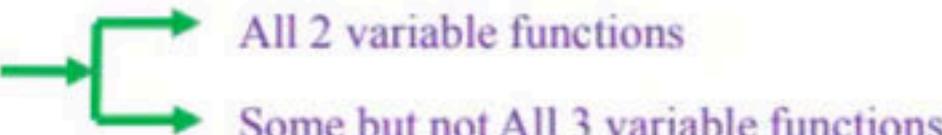
1. Octal to binary (  $8 \times 3$  encoder )
2. Decimal to Binary (  $10 \times 4$  encoder )
3. Hexadecimal to Binary (  $16 \times 4$  encoder )

$2^n X n$   
 $n$  -----> number of outputs  
 $2^n$  -----> number of inputs

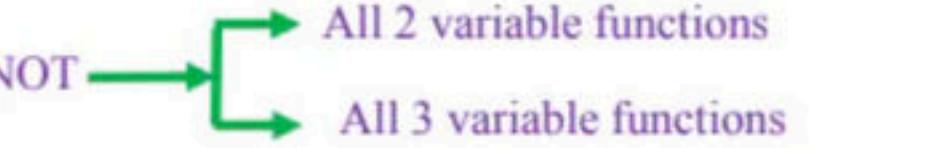
- For an Encoder at a time only one among the all inputs is high , remaining all inputs should be zero
- If multiple inputs are simultaneously high, then the output is not valid, to avoid this restriction we will go for priority encoder.

1. By using one  $4 \times 1$  Mux

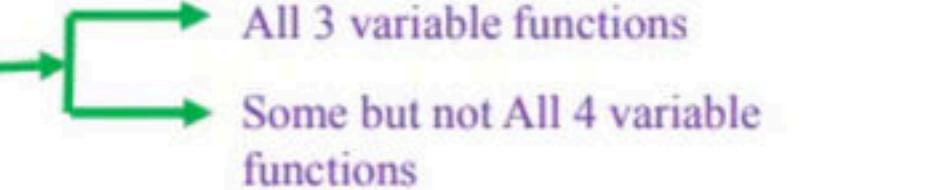
BVREDDY



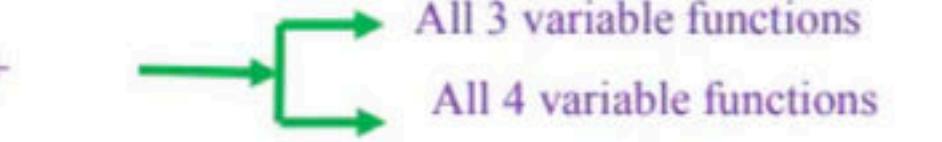
2. By using one  $4 \times 1$  Mux + NOT Gate



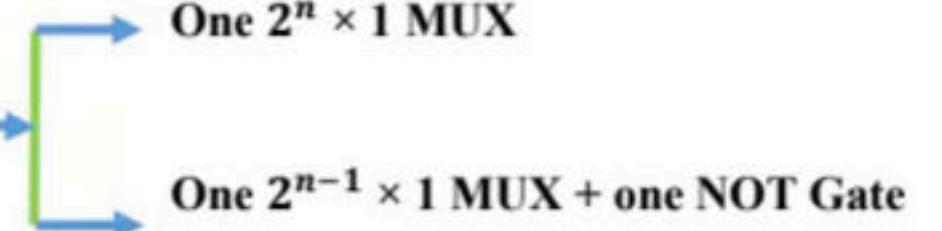
3. By using one  $8 \times 1$  Mux



4. By using one  $8 \times 1$  Mux + NOT Gate



5. n- variable function



BVREDDY

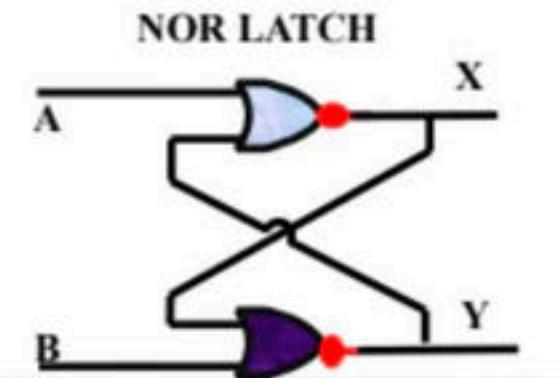
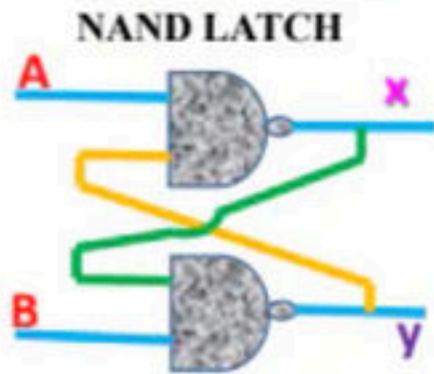
## Sequential Circuits

The logic circuit whose outputs at any instant of time depends on the present inputs as well as on the past outputs are called sequential circuits, in sequential circuits ,the output signals are fed back to the input side .

BVREDDY

BVREDDY

- Out put of combinational circuit depends on input combinations .
- Output of sequential circuits depends on input sequence.
- For unequal delay of gates also the operation is valid



A	B	X	Y
0	0	1	1
0	1	0	1
1	0	1	0
1	1	Memory	

A	B	X	Y
0	0	1	1
0	1	1	0
1	0	0	1
1	1	Memory	

For **SR NAND** latch , if the input sequence is **00 -----> 11** , then the following cases arises

- If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as critical race
- However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates

For **SR NOR** latch , if the input sequence is

**11 -----> 00** , then the following cases arises

- If the delay of both gates are same then we don't have any stable output , the output is oscillatory , this condition is known as critical race .
- However if the delay of both gates are not equal then there exist a stable output , but it depends on the individual delay of the gates .

## FLIP FLOP

In a latch the output changes immediately in response to external input , so to have an additional control , we are introducing a signal called “ **CLOCK** ” , whose purpose is same as Enable pin of Decoder.

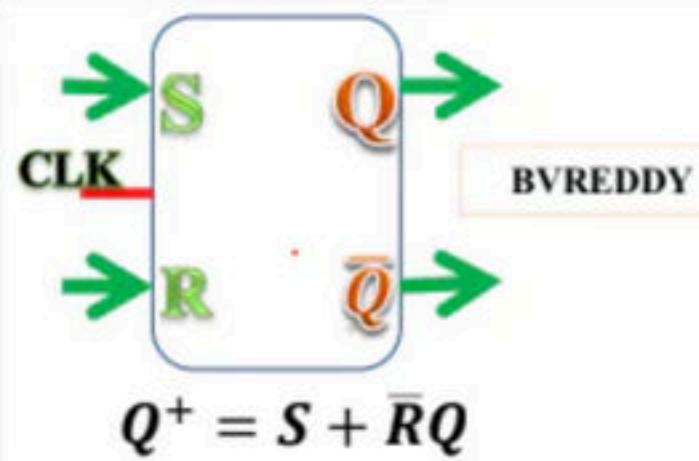
**Latch +Clock = Flip Flop**

Latches are universally not unique and hence their truth tables are not unique .

Flip Flops are universally unique , and their truth tables are unique .

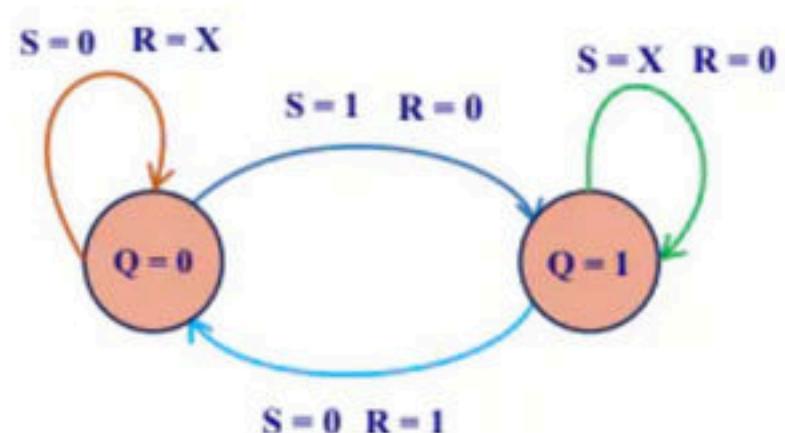
Use the Code : **BVREDDY** ,to get the maximum discount

BVREDDY



CLK	S	R	Q+	State
0	x	x	Q	Memory
	1	0	0	
	1	0	1	
	1	1	0	Memory
	1	1	1	
	1	0	1	Reset
1	1	0	1	Set
1	1	1	x	Invalid

Q	Q <sup>+</sup>	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

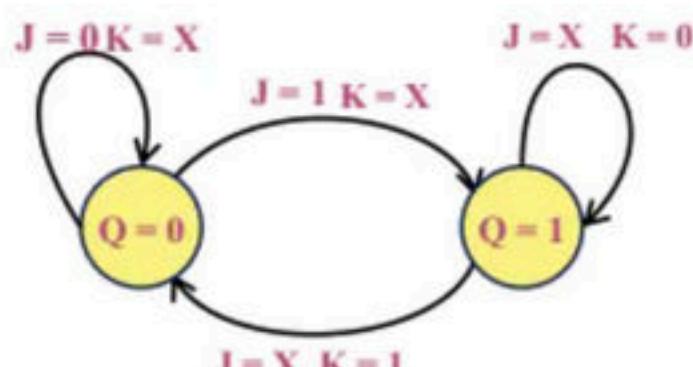


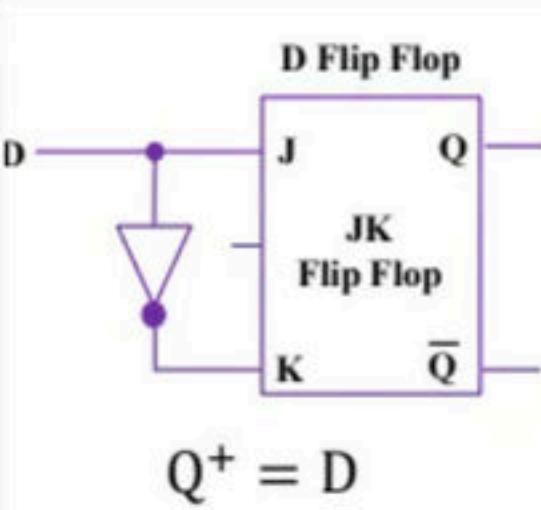
**JK flip-flop**

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

CLK	J	K	Q	Q <sup>+</sup>	State
0	x	x	Q	Q	Memory
	0	0	Q	Q	
1	0	1	0	0	Reset
	1	0	1	1	
1	1	0	1	1	Set
	1	1	1	0	
1	1	1	Q	Q	Toggle
	1	1	1	1	

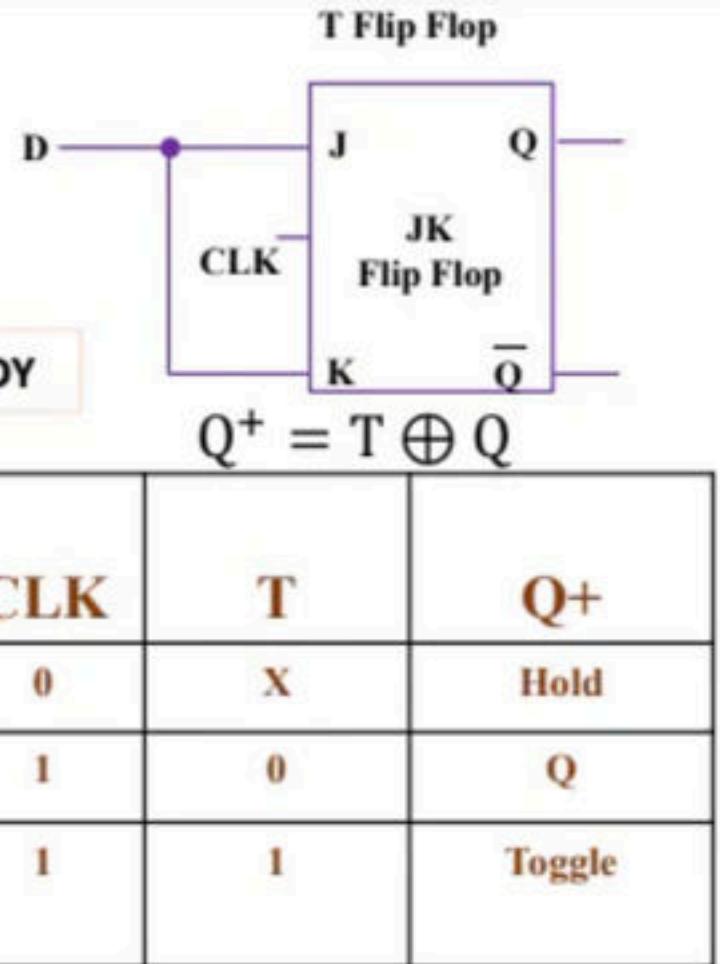
Q	Q <sup>+</sup>	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0





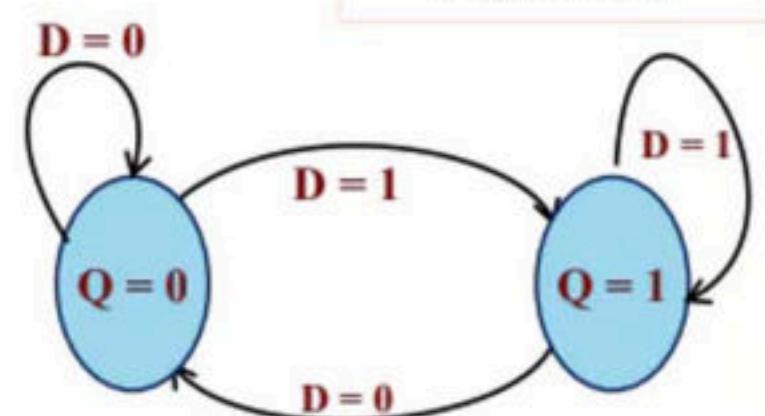
CLK	D	$Q^+$
0	X	Hold
1	0	0
1	1	1

CLK	D	Q	$Q^+$
0	X	Q	Q
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



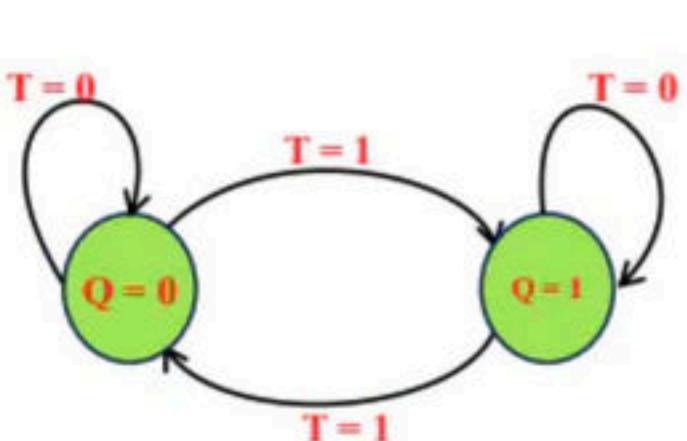
CLK	T	Q	$Q^+$
0	X	Q	Q
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

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



**Use the Code :**  
**BVREDDY**

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



**BVREDDY**

## Race Around Condition

The output of the FF changes to  $0 \rightarrow 1 \rightarrow 0 \dots$  Continuously at the starting of the next clock the output is uncertain , which is called as Race Around Condition (RAC )

RAC occurs in any FF if the following conditions satisfies

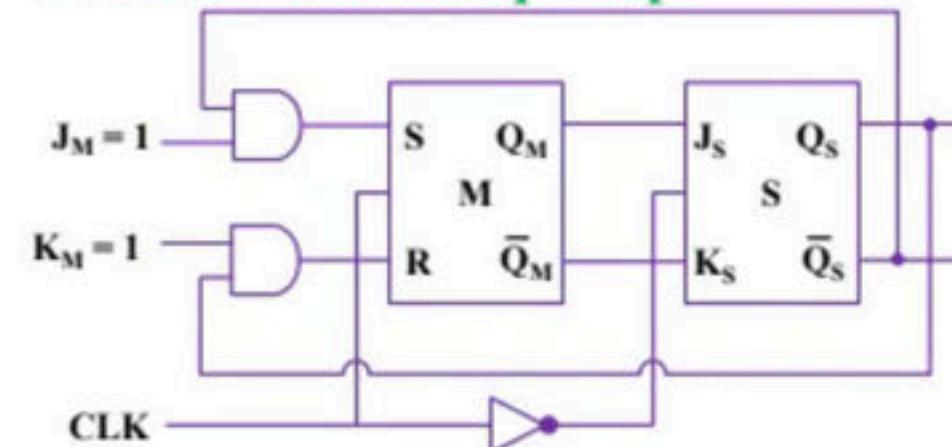
1. If the FFs are operated in level triggering
2. if  $(tpd) < (Tclk)_{on}$  ,
3. If the FFs are operated in Toggle mode

If the above 3 conditions satisfies simultaneously then there is a continuous race in the output of the FF between 0 and 1 to reach the next state , who will be the winner of the race is not certain , that depends on tpd and ( Tclk ) on .

## Remedy

1.  $(Tclk)_{on} < (tpd) < T$
2. By using Edge triggered FF
3. By using Master Slave FF

## Master – Slave Flip Flop



1. In case of Master Slave configuration , Master is applied with input clock and Slave is applied with inverted clock , so out of two FFs at a time only one of the FF respond and other will not respond . As a result, Many times toggling in a single clock cycle has been converted to one time toggle , hence RAC is avoided .
2. In Master Slave configuration , command signal is generated by master FF and the response of the command signal is given by slave FF
3. Master slave FF can store 1 – bit of data

### JK to SR

$$\begin{aligned} J &= S \\ K &= R \end{aligned}$$

### JK to D

$$\begin{aligned} J &= D \\ K &= \bar{D} \end{aligned}$$

### JK to T

$$\begin{aligned} J &= T \\ K &= T \end{aligned}$$

### SR to JK

$$\begin{aligned} S &= J\bar{Q} \\ R &= KQ \end{aligned}$$

### SR to D

$$\begin{aligned} S &= D \\ R &= \bar{D} \end{aligned}$$

### SR to T

$$\begin{aligned} S &= T\bar{Q} \\ R &= TQ \end{aligned}$$

### D to SR

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

### D to JK

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

### D to T

$$D = T \oplus Q$$

### T to SR

$$T = S\bar{Q} + RQ$$

### T to JK

$$T = J\bar{Q} + KQ$$

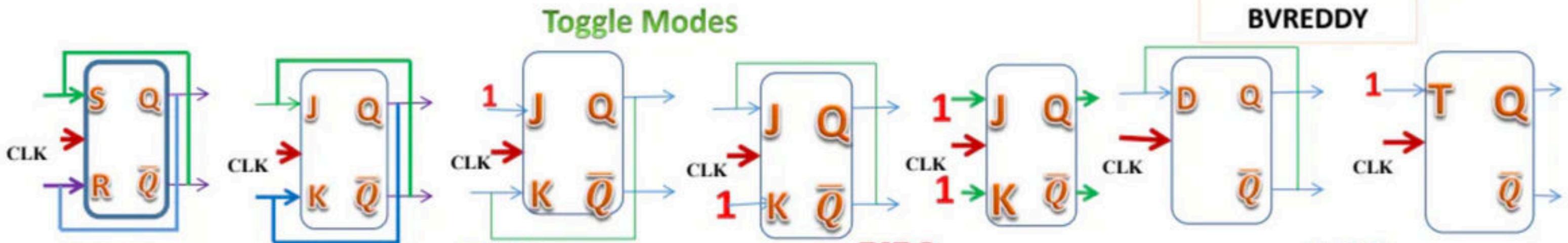
### T to D

$$T = D \oplus Q$$

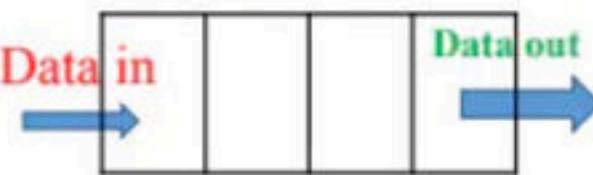
C  
O  
N  
V  
E  
R  
S  
A  
T  
I  
O  
N

of

F  
L  
I  
P  
F  
L  
O  
P



### SISO



➤ SISO Configuration has only

- 1- input
- 1- output

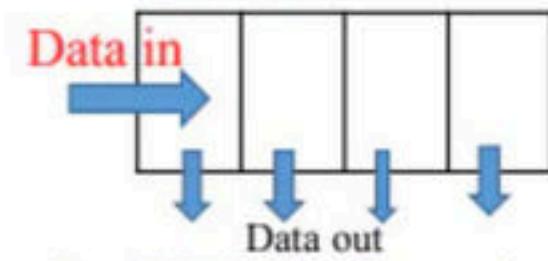
➤ For SISO configuration

for storing = (n) CP

for retrieving = (n-1) CP

Total number clock pulses = 2n-1

### SIPO



➤ SIPO Configuration has only

- 1- input
- 4- output

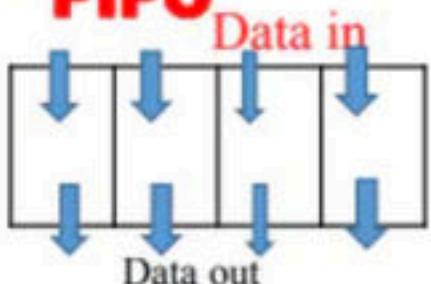
➤ For SIPO configuration

for storing = (n ) CP

for retrieving = 0 CP

Total number clock pulses = n

### PIPO



➤ PIPO Configuration has only

- 4- input
- 4- output

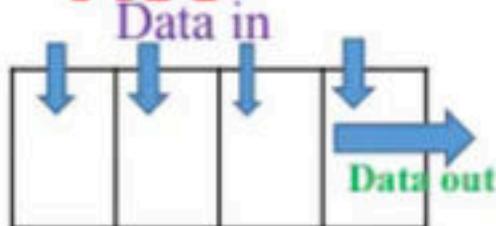
➤ For PIPO configuration

for storing = 1 CP

for retrieving = 0 CP

Total number clock pulses = 1

### PISO



➤ PISO Configuration has only

- 4- input
- 1- output

➤ For PISO configuration

for storing = 1 CP

for retrieving = (n-1)CP

Total number clock pulses = n

## Counters

**State of a Counter** : Any possible output of a counter is known as its state , for a n – bit counter the maximum possible states are  $2^n$

The states which are counted by the counter are called as *valid states* , and the states which are not counted (skipped) by the counter are called as *invalid states* .

**Modulus of a Counter** : The minimum number of clocks needed to get the counting pattern repeats is called as Modulus of a counter

Design equation of a counter

$$2^n \geq N$$

$$n \geq \log_2 N$$

n----> number of Flip Flops

N-----> MOD no. of a counter

BVREDDY

BVREDDY

## ASYNCHRONOUS COUNTER

BVREDDY

- Different FFs are applied with different clocks
- For only one FF external clock is applied ,which is LSB and output of one FF will acts as clock to next FFs
- FFs are operated in toggle mode

➤ Fixed counting sequence

1. up counter

2. down counter

- $\ominus$ ve Edge trigger and Q as a clock -----> Up counter
- $\ominus$ ve Edge trigger and  $\bar{Q}$  as a clock -----> Down counter
- $\oplus$ ve Edge trigger and Q as a clock -----> Down counter
- $\oplus$ ve Edge trigger and  $\bar{Q}$  as a clock -----> Up counter
- The disadvantages of the ripple counter is that transition states are present due to delay of the FF ( Decoding errors) .
- If only one FF changes its state ,then no transition states will be present , if more than one FF changes its states than transition states present.

BVREDDY

➤ To avoid decoding errors strobe signal is used .

➤ Strobe signal is kept low for 3tpd , for 3- bit counter , so that transition states are not reflected, and after 3tpd strobe signal is made high .

➤ If delay each FF is  $t_{pd}$  , then

$$T_{CLK} \geq n t_{pd}$$

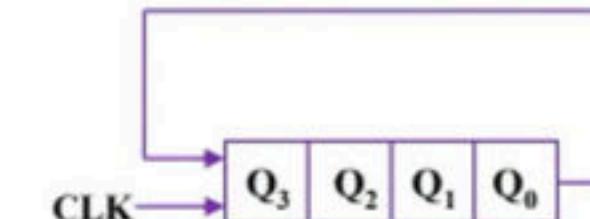
$$f_{CLK} \leq \frac{1}{t_{pd}}$$

**ये वक्त भी गुजर जाएगा**  
**This time will also pass**

## RING COUNTER

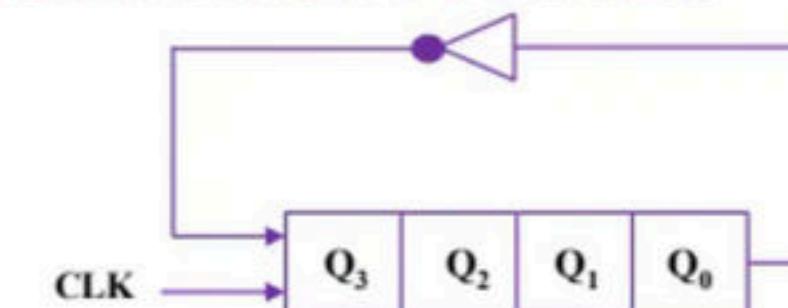
- Ring counter is a synchronous counter , it is a shift register in which last FF output is connected to the first FF input .
- In ring counter only one FF output is logic ‘1 ‘ and it will rotate with clock .
- Ring counter performs right shift operation .

BVREDDY



- Decoding logic of ring counter is simple and does not require any external logic circuit
- If all the outputs of FFs initially zero , then the Ring counter does not start .
- If more than one FF outputs' are high initially, then the ring counter enters into unused state and never come out of unused state , this is called as **Lock out problem** .

## JOHNSON RING COUNTER



BVREDDY

Johnson Ring counter

Twisted Ring counter

Switch tail counter

Walking Counter

Creeping counter

Mobies counter

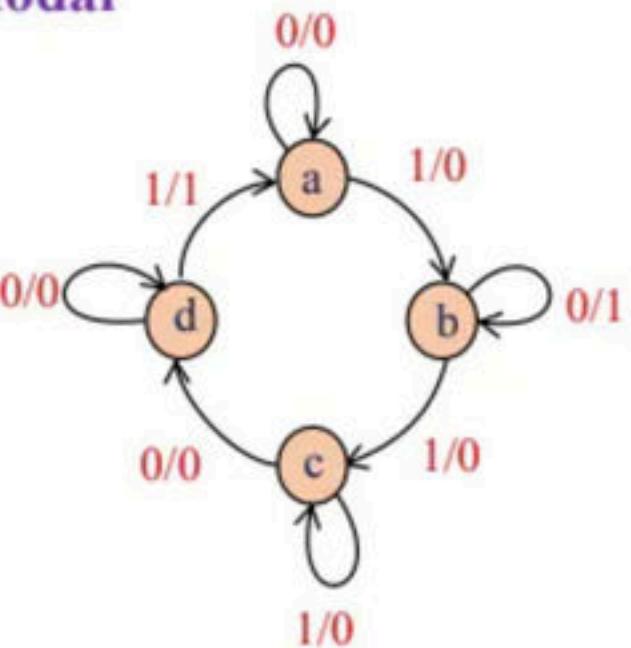
**Use the Code :**  
**BVREDDY**

### Ring counter

Ring counter		Johnson ring counter
1. Mod No = n	BVREDDY	1. Mod No = 2n
2. Number of used states= n Number of unused states = $2^n - n$		2. Number of used states= 2n Number of unused states = $2^{2n} - n$
3. Time period of each FF = n( $T_{CLK}$ )		3. Time period of each FF = 2n( $T_{CLK}$ )
4. Frequency of each FF = $\frac{f_{clk}}{n}$		4. Frequency of each FF = $\frac{f_{clk}}{2n}$
5. Suffer from lock out problem		5. Suffer from lock out problem
6. Decoding logic is simple		6. Decoding logic requires AND and NOR gates

### Mealy Modal

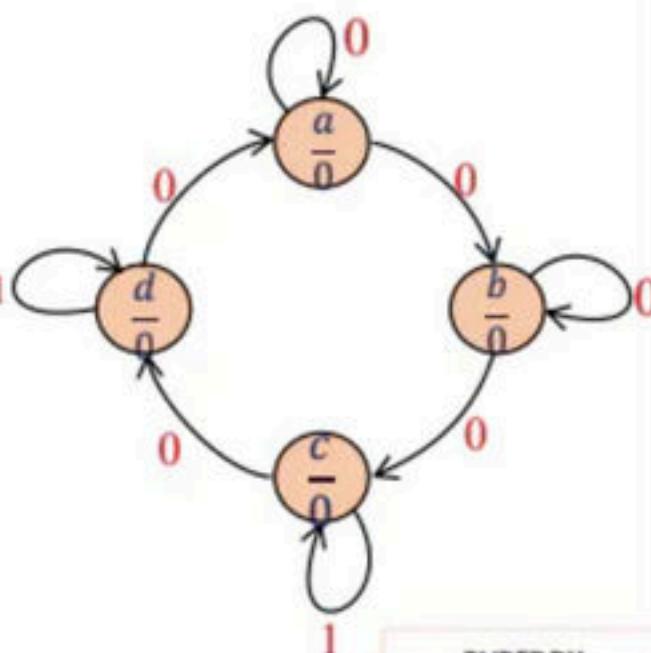
Present state	NS , O/P	
	X = 0	X = 1
a	a , 0	b , 0
b	b , 1	c , 0
c	d , 0	c , 0
d	d , 0	a , 1



BVREDDY

### Moore Modal

Present state	Next State		Output
	X = 0	X = 1	
a	a	b	0
b	b	c	0
c	d	c	0
d	a	d	1



BVREDDY

### FINITE STATE MACHINE

Synchronous Sequential circuits are also called as Finite State Machine ( FSM )

There are two types of FSMs

#### 1. Mealy State Machine

- The output of Mealy State Machine is a function of present state as well as present input
- to detect n – bit sequence by using Mealy modal n number of states are required

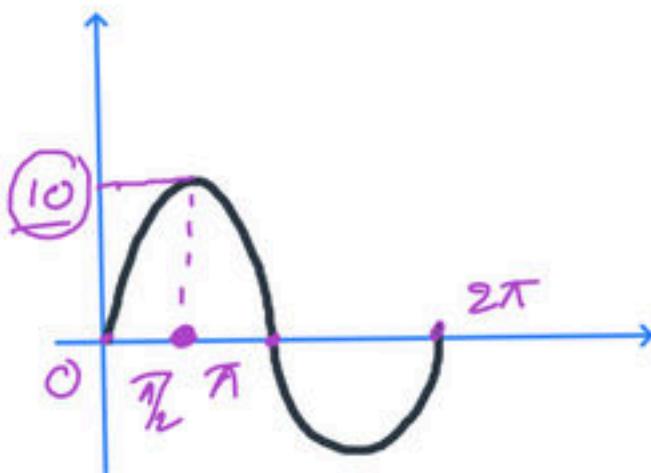
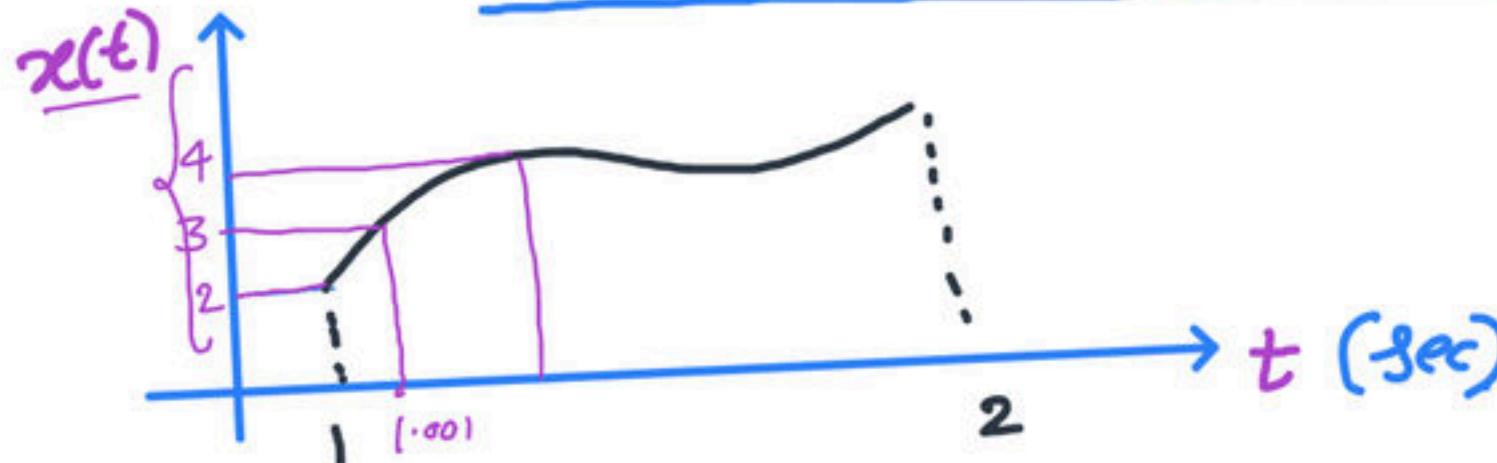
BVREDDY

#### 2. Moore State Machine

- The output of Moore State Machine is a function of present state only
- To detect n – bit sequence by using Moore modal (n+1) number of states are required

## Analog Signal :

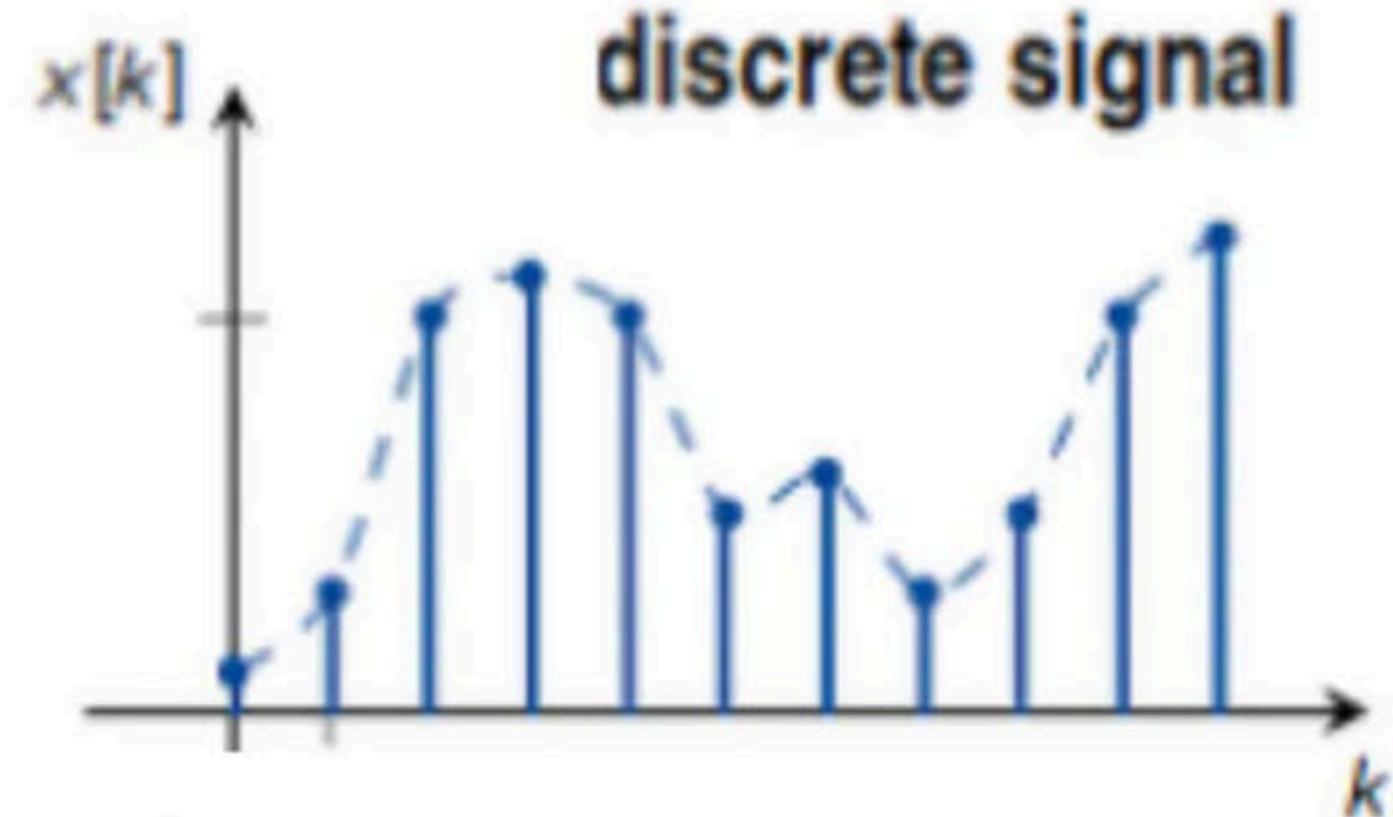
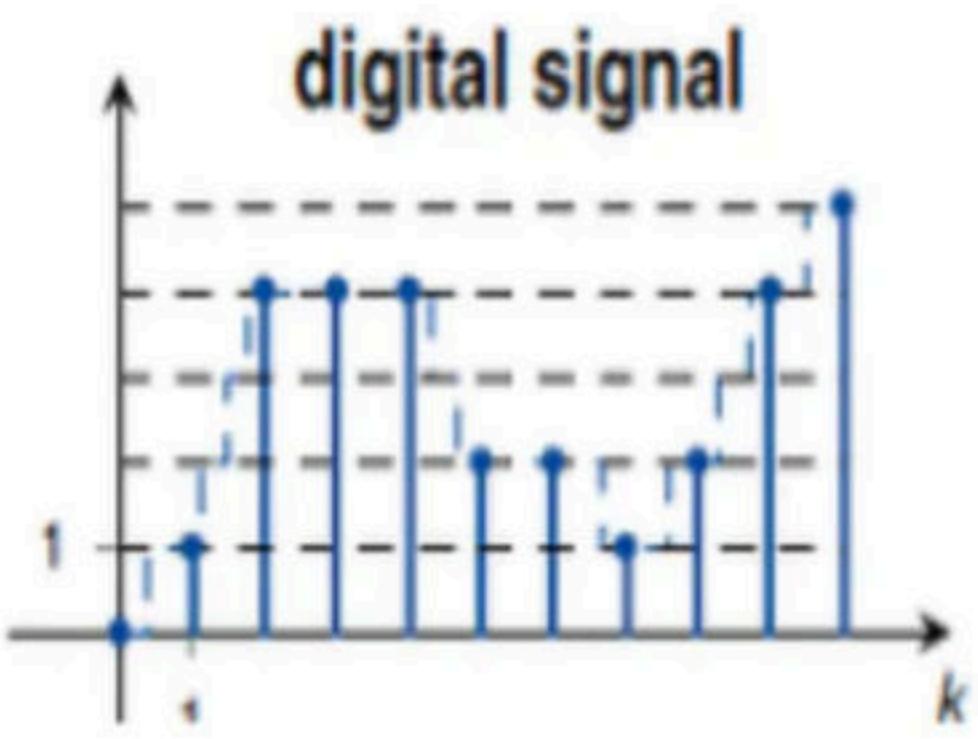
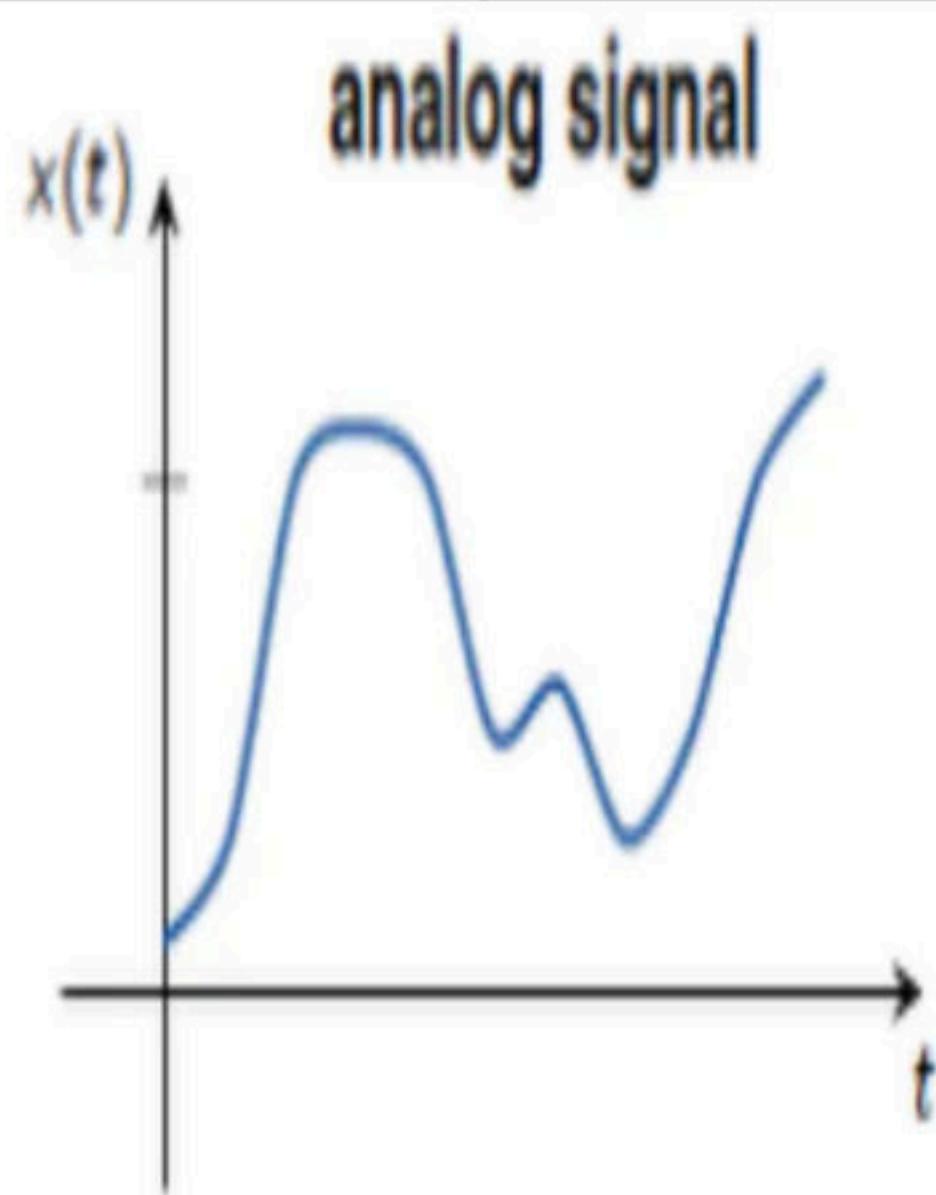
If the signal amplitude can take infinite number of possibilities then it is called as analog signal .



## Digital Signal :

If the signal amplitude can take only finite number of possibilities, then it is called as Digital signal .

Amplitude	Time	Signal
Continuous	Continuous	Analog signal
Continuous	Discrete	Discrete signal
Discrete	Discrete	Digital signal



- If the digital signals takes only two possible amplitudes , then it is called as **Binary Digital Signal**
- The system which process the analog signals is called as **analog system**.
- The system which process the digital signals is called as **digital system**.

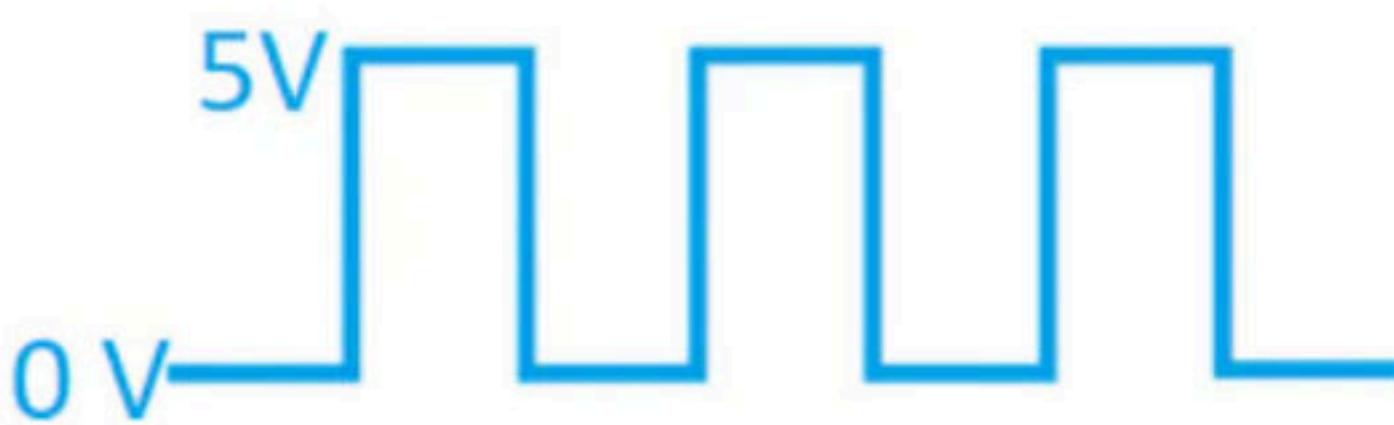
# Logic Systems

## 1. Positive logic system

High voltage corresponds to logic “ 1 ”

+5V ----> Logic '1' → High

0V ----> Logic '0' → Low.



## Positive logic system

$2V \rightarrow$  Logic '0'

$1.5V \rightarrow$  Logic '1'

## Positive logic system

$- 2V \rightarrow$  Logic '0'

$- 1.5V \rightarrow$  Logic '1'

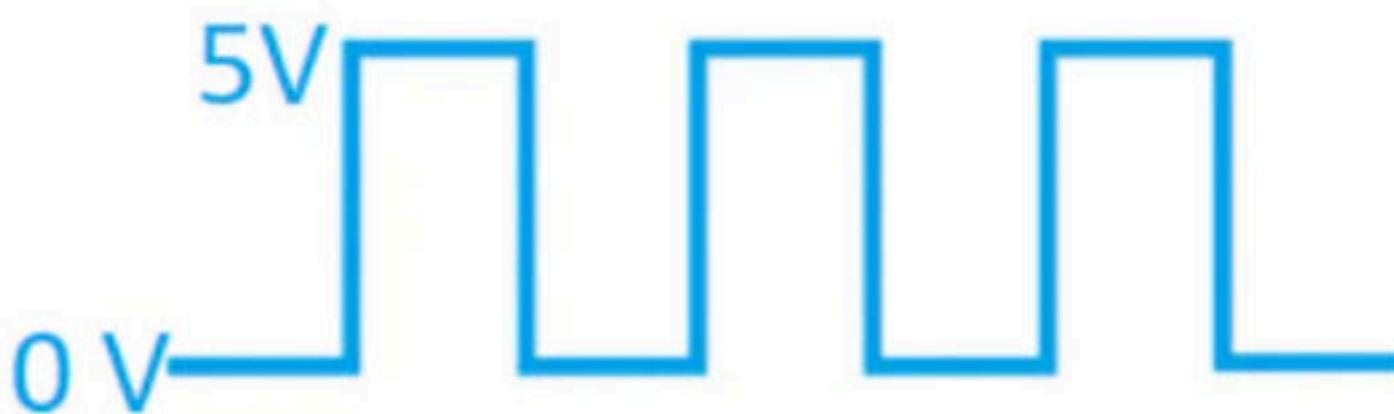
# Logic Systems

## 2. Negative logic system

High voltage corresponds to logic “ 0 ”

+5V ----> Logic ‘ 0 ’

0V ----> Logic ‘ 1 ’



## Negative logic system

$2V \rightarrow \text{Logic '1'}$

$1.7V \rightarrow \text{Logic '0'}$

## Negative logic system

$-3V \rightarrow \text{Logic '1'}$

$5V \rightarrow \text{Logic '0'}$



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

Positive Logic System

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

Negative Logic System

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

## Duality

- A **positive logic system** is converted into **negative logic system** by using the **concept of duality**.

1. \*  $\longleftrightarrow$  +

2. 0  $\longleftrightarrow$  1

3. Keep the variables as it is.

Q. Find the Dual of the expression  $f = AB + C$

$$f = (AB) + C$$

BODMAS

Bracket > AND > OR

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

Q. Find the Dual of the expression  $f = A(B+C)$

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

$$f_d = A + (BC)$$

# Boolean Algebra

$$Y(A, B) = AB \cdot A = \underline{O} / \underline{I}$$

- It is an analysis tool that is used for analyzing and designing of various digital system.
- The *i/p vs o/p* relationship in digital system is known as logic expression Boolean exp.

## QR -Operation

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

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

$$\begin{aligned} & \text{if } A=0 \\ & \quad \bar{A} = 1 \end{aligned} \quad A + \bar{A} = 1$$

$$1 + \text{Anything} = 1$$

$$1 + \text{Raj} = 1$$

$$A + A + A + A + A + A = A$$

## AND-Operation

$$A \cdot 1 = A$$

$$A \cdot 0 = 0$$

$$A \cdot A = A$$

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

# Practice Questions

## Day - 1

15 - Que

1 The dual of a Boolean theorem is obtained by

- (a) interchanging all zeros and ones only
- (b) changing all zeros to ones only
- (c) changing all ones to zeros only
- (d) interchanging operators and identity elements

2. In Boolean Algebra '1' is called

- (a) Additive identity      (b) Multiplicative identity
- (c) Either 1 or 2      (d) None

3. In Boolean Algebra '0' is called

- (a) Additive identity      (b) Multiplicative identity
- (c) Both 1 and 2      (d) None

4] What is dual of  $A+[B+(AC)] + D$

- (a)  $A+[B(A+C)] + D$
- (b)  $A[B+AC] D$
- (c)  $A+[B(A+C)] D$
- (d)  $A[B(A+C)] D$

5. In the following equations the equals sign means is equal to Which of the following is a positive logic?

- (a)  $0 = 0 \text{ V}$  and  $1 = +5 \text{ V}$
- (b)  $0 = 0 \text{ V}$  and  $1 = -5 \text{ V}$
- (c)  $0 = +5 \text{ V}$  and  $1 = 0 \text{ V}$
- (d) None of these

6. The dual of Boolean theorem  $x(y+z) = xy+xz$  is

- (a)  $x + yz = xy + xz$
- (b)  $x(y+z) = (x+y)(x+z)$
- (c)  $x+yz = (x+y)(x+z)$
- (d) None

7. Given Boolean theorem  $AB + A'C + BC = AB + A'C$  which of the following is true?

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

8. The voltage levels for positive logic system

- a) must necessarily be positive
- (b) must necessarily be negative
- (c) may be positive or negative
- (d) must necessarily be 0 V and 5 V

9. The voltage levels for negative logic system

- (a) must necessarily be negative
- (b) must necessarily be positive
- (c) need not be negative
- (d) must necessarily be 0 V and -5 V

**10. The dual of a Boolean expression is obtained by**

- (a) interchanging all 0s and 1s
- (b) interchanging all 0s and 1s, all + and ‘.’ signs
- (c) interchanging all 0s and 1s, all + and ‘.’ signs and complementing all the variables
- (d) interchanging all + and ‘.’ signs and complementing all the variables

11. which one of the following is the dual form of the Boolean identity?

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

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

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

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

(d)  $AB + \overline{A}C = AB + \overline{A}C + BC$

## 12. The Boolean theorem:

$AB + \overline{A}C + BC = AB + \overline{A}C$  corresponds to

- (a)  $(A+B).(\overline{A} + C).(B+C) = (A+B).(\overline{A} + C)$
- (b)  $AB + \overline{A} C + BC = AB + BC$
- (c)  $AB + \overline{A} C + BC = AB + BC$
- (d)  $(A+B).(\overline{A} + C).(B+C) = (AB).(\overline{A} C)$

13. Given Boolean theorem,  $AB + \overline{A}C + BC = AB + \overline{A}C$ . Which one of the following identities is true?

- (a)  $(A+B)(\overline{A}+C)(B+C) = (A+B)(\overline{A}+C)$
- (b)  $(AB + \overline{A}C + BC) = AB + BC$
- (c)  $AB + \overline{A}C + BC = (A+B)(\overline{A}+C)(B+C)$
- (d)  $(A+B)(\overline{A} + C)(B+C) = AB + \overline{A}C$

14.  $AB + \overline{A}C = (A + C)(\overline{A} + B)$  Which one of the following is the dual form of the Boolean identity given above?

- (a)  $AB + \overline{A}C = AC + \overline{A}B$
- (b)  $(A + B)(\overline{A} + C) = (A + C)(\overline{A} + B)$
- (c)  $(A + B)(\overline{A} + C) = AC + \overline{A}B$
- (d)  $AB + \overline{A}C = AB + \overline{A}C + BC$

15. If A and B are Boolean variables, then what is  $(A + B).(A + \bar{B})$  equal to?

- (a) B
- (b) A
- (c)  $A + B$
- (d)  $AB$

# Laws of Boolean Algebra

## 1. Commutative Law

$$AB = BA$$

$$A+B = B+A$$

## 2. Associative Law

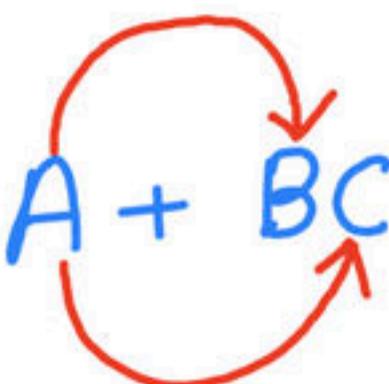
$$ABC = A(BC) = B(AC) = C(AB)$$

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

### 3.Distributive Law

$$A(B+C) = AB + AC$$

(Mingle Property)


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

## 4. De Morgan's Law

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

$$\overline{ABCDEF} = \overline{A} + \overline{B} + \overline{C} + \overline{D} + \overline{E} + \overline{F}$$

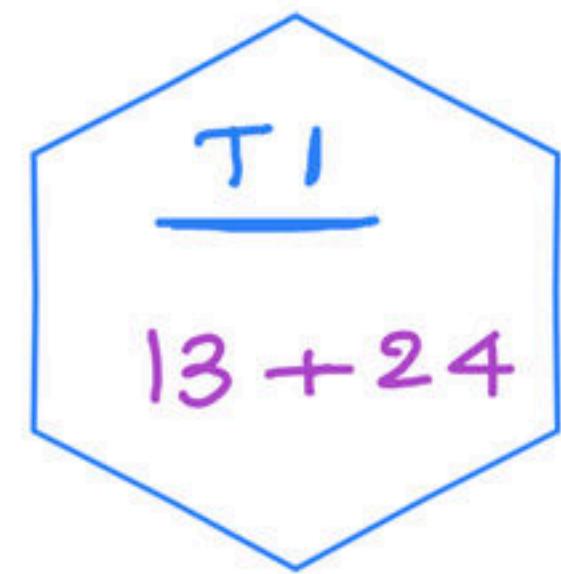
$$\overline{A+B} = \overline{A} \cdot \overline{B}$$

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

## 5. Transposition theorem ( T- 1)

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

①      ②      ③      ④



## 6. Transposition theorem ( T- 2)

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



T 2

14 + 23 .

# Problems

Q) Minimize the following

$$\underbrace{(x+y)(x+\bar{y})(\bar{x}+y)}_{1 \quad 2 \quad 3 \quad 4}$$

$$\frac{T1}{13+24}$$

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

$$y\bar{y} = 0$$

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

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

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

$$xy$$

Q) Minimize the following

$$\frac{(x+y+z)(x+y+\bar{z})}{\textcircled{1} \quad \textcircled{2} \quad \textcircled{3} \quad \textcircled{4}}$$

$$\underline{\tau_1}$$

$$13+24$$

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

$$z\bar{z} = 0$$

$$x+y$$

$$\underline{\underline{\hspace{1cm}}}$$

Q) Minimize the following

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

1    2    3    4

T1

13 + 24.

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

A

Q) Minimize the following

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

Mingle

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

$$A + \overline{A} = 1.$$

$$= A + B$$

=

Q) Minimize the following

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

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

Mingle

$$A + \overline{B}$$

Q) Minimize the following

$$\bar{A} + AB$$

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

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

Q) Minimize the following

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

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

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



Q. Find the complement of the expression  $f = AB + C$

Q. Find the complement of the expression  $f = A(B+C)$

Q. Find the complement of the expression  $f = a[b + z(x + \bar{a})]$

Q. Find the complement of the expression  $f = a(b + c) + \bar{a}b$

# Consensus Theorem ( Rajinikanth Wala)

Q) Minimize the following

$$\overline{A}B + AC + BC$$

Q) Minimize the following

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

Q) Minimize the following

$$AB + AC + B\bar{C} =$$

Q) Minimize the following

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

Q) Minimize the following

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

Q) Minimize the following

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

Q) Minimize the following

$$AB + \bar{A}CD + BCD$$

Q) Minimize the following

$$AB + \bar{A}CD + BCD$$

Q) Minimize the following

$$ABC + \bar{A}D + \bar{B}D + CD$$

Q) If  $X^* Y = \overline{XY}$  , then the minimized expression of  $[(x + y) * y] * z$  is ..

Q) If  $f(A, B) = \bar{A} + B$ , then the simplified expression of  
 $f[(f(x + y, y)), z]$  is .....

Q) Minimize the following Boolean expression  $(A + B + C)(A + B + \bar{C})(A + \bar{B} + C)$

Q) Minimize the following Boolean expression

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

Q) Minimize the following Boolean expression       $v + \bar{v}w + \bar{v}\bar{w}x + \bar{v}\bar{w}\bar{x}g$

Q) Minimize the following Boolean expression       $A + \bar{A}B + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}D$

Q) Minimize the Boolean expression

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

# **Practice Questions**

## **Day - 2**

16. Which of the following Boolean Algebra rules is correct?

- (a)  $A \cdot \bar{A} = 1$
- (b)  $A + AB = A + B$
- (c)  $A + \bar{A} \cdot B = A + B$
- (d)  $A(A + B) = B$

17. The Boolean equation  $X = [(A + \overline{B})(B + C)] B$  can be simplified to

- (a)  $X = \overline{A} B$
- (b)  $X = A \overline{B}$
- (c)  $X = A B$
- (d)  $X = \overline{A} \overline{B}$

18. Logic function  $(\bar{A} + B)(A + \bar{B})$  can be reduced to:

- (a) B
- (b)  $\bar{B}$
- (c) A
- (d)  $\bar{A}$

19. The simplified form of the Boolean expression  $AB + A(B + C) + B(B + C)$  is given by

(a)  $AB + AC$

(b)  $B + AC$

(c)  $BC + AC$

(d)  $AB + C$

20. The expression  $(X+Y)(X+\bar{Y})(\bar{X}+Y)$  is equivalent to

(a)  $\bar{X}\bar{Y}$

(b)  $\bar{X}Y$

(c)  $X\bar{Y}$

(d)  $XY$

21. In Boolean algebra if  $F = (A+B)(\bar{A}+C)$  then

(a)  $F = AB + \bar{A}C$

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

(c)  $F = AC + \bar{A}B$

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

22. Which of the following expression is not correct?

(a)  $X + \overline{X}Y = X$

(b)  $X \cdot (\overline{X} + Y) = XY$

(c)  $X + X\overline{Y} = X$

(d)  $ZX + Z\overline{X}Y = ZX + ZY$

23. What is the simplified form of the Boolean expression  $T = (X+Y)(X+\bar{Y})(\bar{X}+Y)$

- (a)  $\bar{X}\bar{Y}$
- (b)  $\bar{X}Y$
- (c)  $XY$
- (d)  $X\bar{Y}$

24.  $(A' + B' + C')$  is equal to

- (a)  $A' B' C'$
- (b)  $ABC$
- (c)  $A+B+C$
- (d)  $A'+B'+C'$

25. The Boolean expression  $(x+y)(x+z)$  is equal to

- (a)  $x+z$
- (b)  $x+y$
- (c)  $x+yz$
- (d)  $y+xz$

26. **Expression**

$$A + \bar{A}B + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D}E$$

**would be simplified to**

- (a)  $A + \bar{A}B + CD + E$
- (b)  $A + B + CDE$
- (c)  $A + BC + CD + DE$
- (d)  $A + B + C + D + E$

27. If  $X\bar{Y} + \bar{X}Y = Z$  then  $X\bar{Z} + \bar{X}Z$  is equal to

(a)  $\bar{Y}$

(b) Y

(c) 0

(d) 1

28. **If A = 0 in logic expression**

$$Z = [A + EF + \bar{B}C + D] \cdot [A + \bar{D}\bar{E} + \bar{B}C + \bar{D}\bar{F}] \text{, then}$$

- (a)  $Z = 0$
- (c)  $Z = \bar{B}C$

- (b)  $Z = 1$
- (d)  $Z = B\bar{C}$

**29. What does the expression  $AD + ABCD + ACD + \bar{A}B + A\bar{C}D + \bar{A}\bar{B}$  on minimization result into?**

- (a)  $A + D$
- (b)  $AD + \bar{A}$
- (c)  $AD$
- (d)  $\bar{A} + D$

30.  $A + AB + ABC + ABCD + ABCDE + \dots =$

(a) 1

(b) A

(c)  $A + AB$

(d)  $AB$

# **Boolean function representation**

**1. Canonical form**

**2. Minimal Form**

# Boolean function representation

**Canonical form :** Each minterm ( maxterms ) contains all the Boolean variables

$$F(A, B, C) =$$

$$F(A, B, C) =$$

**Minimal Form** : The minimized form of Boolean expression

$$F(A, B, C) =$$

$$F(A, B, C) =$$

**Literal :** A Boolean variable either in normal form (or ) complemented form is known as literal

**Minterm :** Each term in canonical SOP representation is known as minterm

**Maxterm:** Each term in canonical POS representation is known as maxterm



With **n**- variable number of possible input combinations =

# Sum of Product (SOP)

# Product of Sum (POS )

## Note :

1. Maximum possible minterms =
2. Maximum possible maxterms =
3. Number of minterm's + number of maxterm's =

4. The sum of all maximum possible minterms =

5. The product of all maximum possible maxterms =

6. Minterm's and maxterm's are of same index are complement to each other

7. The product of two minterms of different index is .....

8. The sum of two arbitrary maxterms of different index is .....

Q) Find the Minterns and Maxterms of the following

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

Q) Find the Minterns and Maxterms of the following

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

Q) Find the Minterns and Maxterms of the following

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

Q) Find the Minterns and Maxterms of the following

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

Q) Find the Minterns and Maxterms of the following

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

Q) Find the Minterns and Maxterms of the following

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

Q) Find the Minterns and Maxterms of the following

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

Q) Find the Minterns and Maxterms of the following

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

Q. Find minterms and maxterms of the logic expression  $F(P, Q, R) = \bar{P} + QR$

Q.  $f(P, Q, R, S) = PQ + \bar{P}QR + \bar{P}Q\bar{R}S$ , the function is equivalent to

- a)  $PQ + QR + QS$
- b)  $P + Q + R + S$
- c)  $\bar{P} + \bar{Q} + \bar{R} + \bar{S}$
- d)  $\bar{P}R + \bar{P}\bar{R}S + P$

$Qf_1(A, B, C) = \sum m(2, 3, 6)$  and

$f_2(A, B, C) = \sum m(1, 2, 5, 7)$

Then  $f_3 = f_1 f_2$        $f_4 = f_1 + f_2$

Q) Find the Minterms  $\bar{f}$

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

Q) Using two Boolean Variables , how many different Boolean functions are possible

1. By using 2- Boolean variables  
total number of possible Boolean functions =

2. By using n- Boolean variables  
total number of possible Boolean functions =

3. By using 2- Boolean variables total number of possible Boolean functions having at most 3- minterms =

4. By using 2- Boolean variables total number of possible Boolean functions having at most 3- maxterms =

5. By using 2- Boolean variables total number of possible Boolean functions having 3- minterms =

6. By using n- Boolean variables total number of possible Boolean functions having 2- minterms =

7. By using 5- Boolean variables total number of possible Boolean functions having at most 3- minterms =

Q)  $Y(A, B, C) = \sum m(1, 4, 6, 7)$  identify the correct statements

1.  $Y = m_1 + m_4 + m_6 + m_7$

2.  $Y = M_0 M_2 M_3 M_5$

3.  $Y = \overline{m_0 + m_2 + m_3 + m_5}$

4.  $Y = \overline{M_1 M_4 M_6 M_7}$

5.  $Y = \overline{m_0} \ \overline{m_2} \ \overline{m_3} \ \overline{m_5}$

6.  $Y = \overline{M_1} + \overline{M_4} + \overline{M_6} + \overline{M_7}$

7.  $Y = m_0 m_2 m_3 m_5$

8.  $Y = M_1 + M_4 + M_6 + M_7$

9.  $Y = \overline{m_1 m_4 m_6 m_7}$

10.  $Y = \overline{M_0 + M_2 + M_3 + M_5}$

Q. If  $A^* B = AB + \bar{A}\bar{B}$  , let  $C = A^* B$  , then which of the following is correct .

- a)  $B^* C = A$
- b)  $A^* C = B$
- c)  $A^* B^* C = 1$
- d)  $A^* B = B^* A$

## **Neutral Function**

If the number of minterms and number of maxterms are equal , then the Boolean function is called as neutral function.

# Mutually Exclusive terms

# Self Dual Expression

# Self Dual Expression

If one time dual of the Boolean expression result the same expression , then it is called as self dual expression

$$f = AB + BC + AC$$

## **Conditions for the given expression is Self Dual**

1. The given Boolean function must be Neutral function

i.e The number of minterms = number of maxterms

number of minterms + number of maxterms =

number of minterms = number of maxterms =

2. It should not contains mutually exclusive terms

i.e If  $m_i$  belongs to f , then  $m_{2^n-i-1}$  should not belongs to f

Q) Verify the given Boolean functions are self dual or not

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

Q) Verify the given Boolean functions are self dual or not

$$f(A, B, C) = m(1, 2, 4, 7)$$

Q) Verify the given Boolean functions are self dual or not

$$f(A, B, C) = m(0, 1, 2, 5)$$

## Note:

1.Number of Boolean functions =

2.Maximum Number of minterms =

3.Maximum Number of maxterms =

4.Number of Neutral functions =

5.Number of self dual expressions =

Q) A logic circuit have 3 inputs A , B , C and output Y . Output Y is logic 1 for the following

1. A and C are true
2. B and C are false
3. A, B and C are true
4. A, B and C are false

then the minimized expression Y is-----

Q) A logic circuit have 3 inputs A , B and C . Output is F . F is logic 1 when majority number of inputs are at logic 1, then the minimized expression for F is

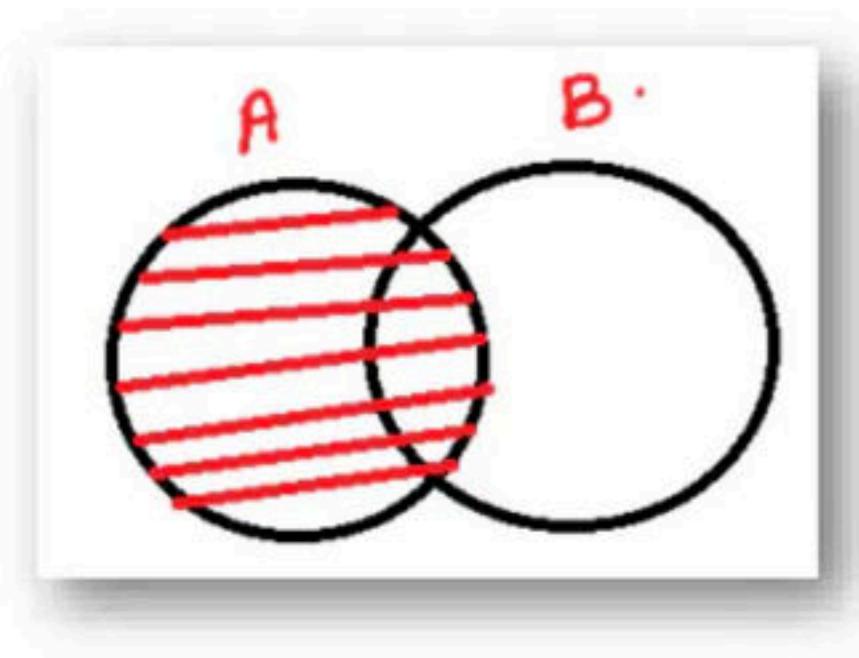
Q) A logic circuit have 3 inputs A , B and C . Output is F . F is logic 1 when minority number of inputs are at logic 1, then the minimized expression for F is

Q. A car alarm system is designed considering 4 inputs, Door closed (D) Key in (K), Seat pressure (S) and Seat belt closed (B). The alarm (A) should sound if

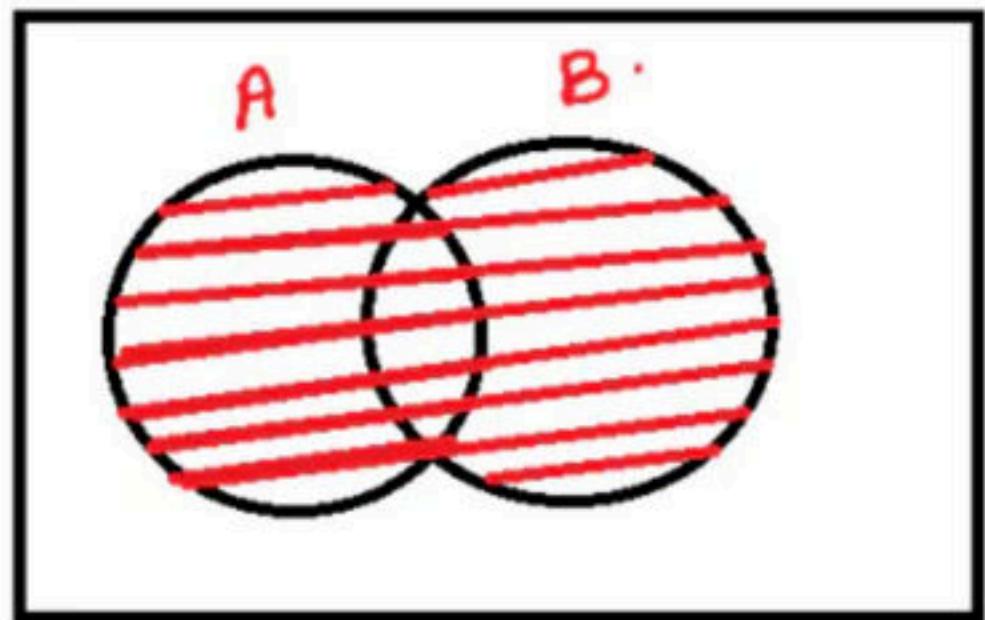
1. The key is in and door is not closed (or)
2. The door is closed, the key is in, driver in the seat and seat belt is not closed.

Then the minimized expression is

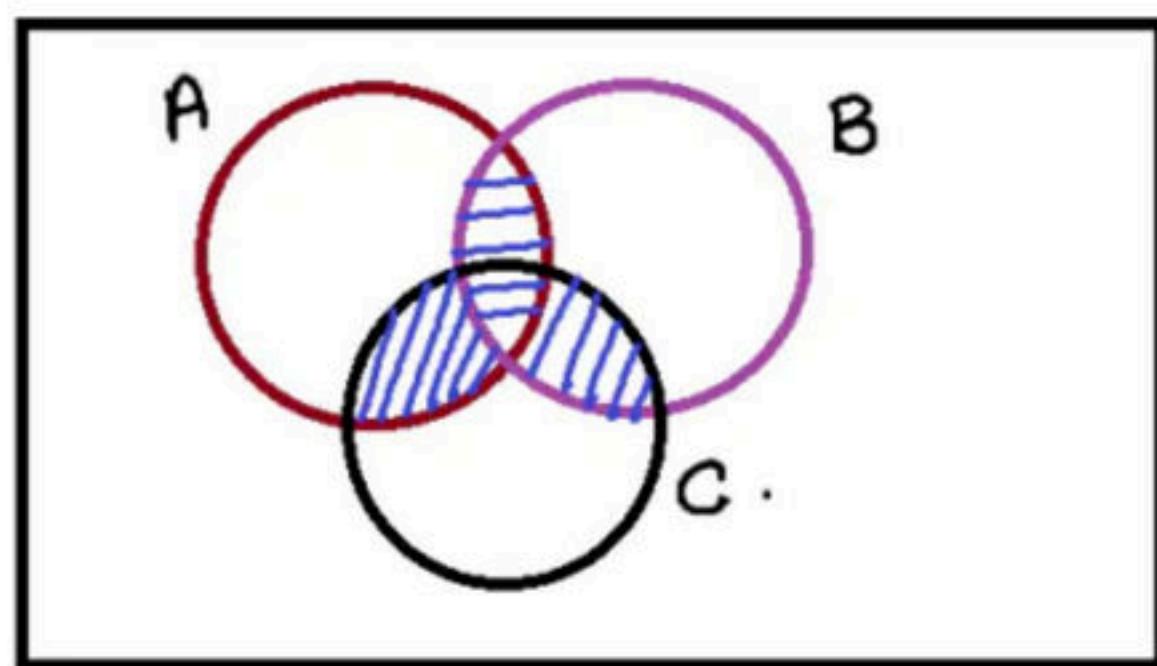
Q) For the given venn diagrams , find the minimized logical expression



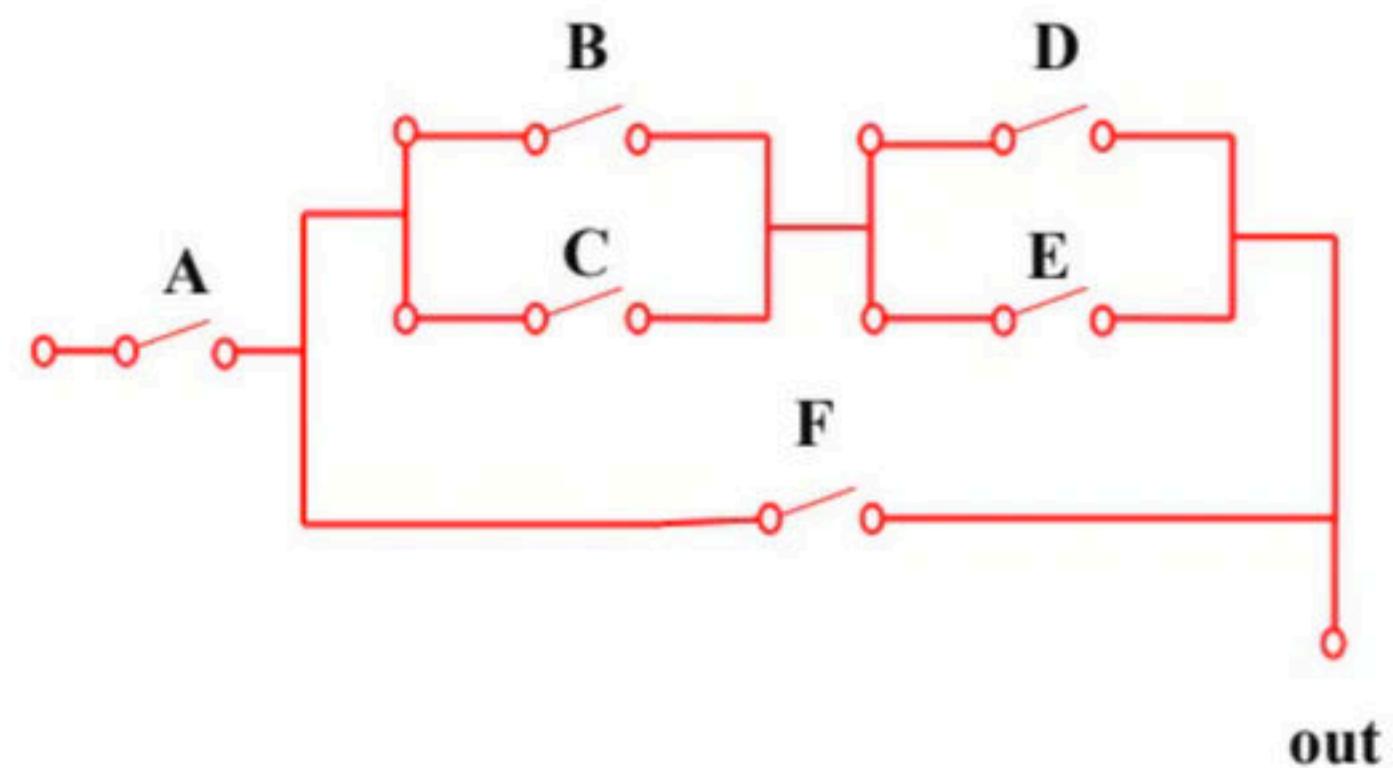
Q) For the given venn diagrams , find the minimized logical expression



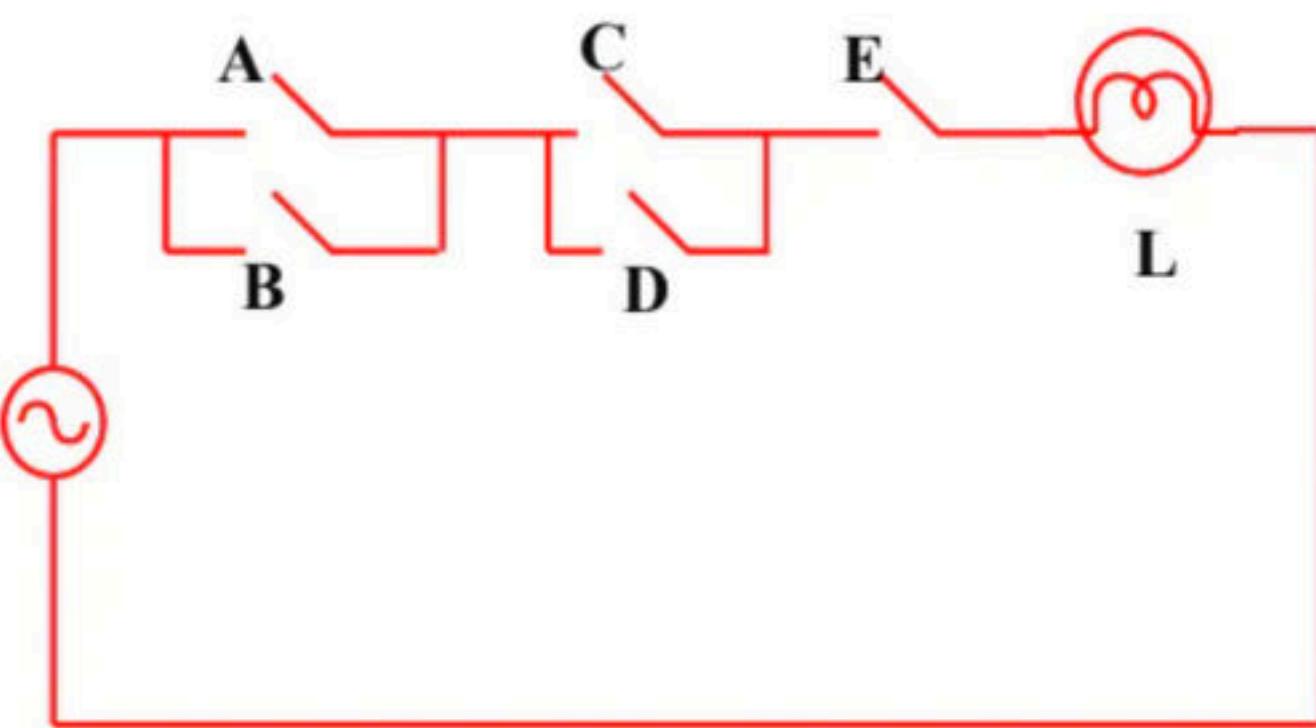
Q) For the given venn diagrams , find the minimized logical expression



**Q.** What Boolean function does the following circuit represents:



**Q.** The switching circuit given in the figure can be expressed in binary logic notation as



Q. How many Boolean functions of the type  $f(x, y) = f(\bar{x}, \bar{y})$  are available with three variables

- a) 4   b) 15   c) 3
- d) 16

Q. How many Boolean functions of the type  $f(x, y, z) = f(\bar{x}, \bar{y}, \bar{z})$  are available with three variables

- a) 4   b) 15   c) 32   d) 16