

## Digital Electronics —

It is an important part of electronics. Due to digital communication in transmission & reception of 'signals', Information technology rapidly increases. Digital watch, digital computer, T.v. Radio, make our daily life effective.

Signal :- (1) Analog Signal (2) Digital Signal

(1) Analog Signal :- It is defined as any physical quantity which varies continuously with respect to time.

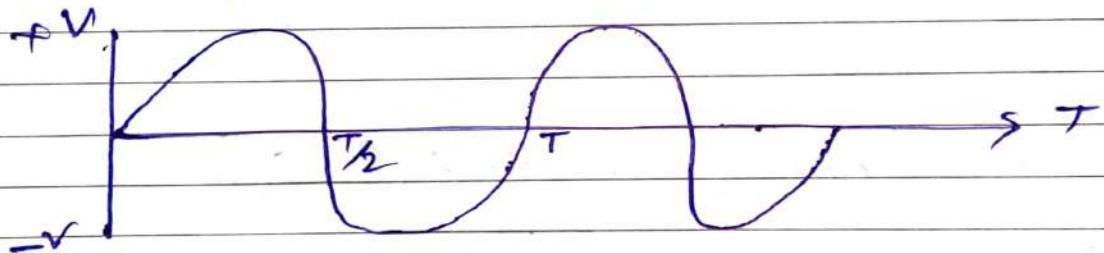


fig:- wave form of Analog Signal

(2) Digital Signal :- It is defined as any physical quantity having discrete values. If is the value of digital signal is one value from any two possible values (0, 1) or (Low, High) or (False, True) or (No, Yes).



fig:- wave form of a digital signal.

## Advantages of digital signal:-

- (1) Easy to operate (eg. On + OFF)
- (2) Easy to store information in digital system.
- (3) accuracy is high in digital system.
- (4) Digital circuits are more reliable.
- 5) Compression of data is possible in digital system

## Application of digital electronics:-

- (1) modern mobile & computers.
- (2) Electronic exchange of telephone.
- (3) Power electronic.
- (4) Traffic, railway signal communication.

## Diff. b/w Analog & digital Signal

### Analog Signal

Continuous change

Not based on Boolean Algebra

maximum effect of sound

difficult to store information

compression of data is impossible

Operation is slow

Low accuracy

### Digital Signal

Discontinuous change

Based on Boolean Algebra

Low effect of sound

Easy to store information

Possible

Operation is fast

High accuracy

# Digital electronics & computer organization

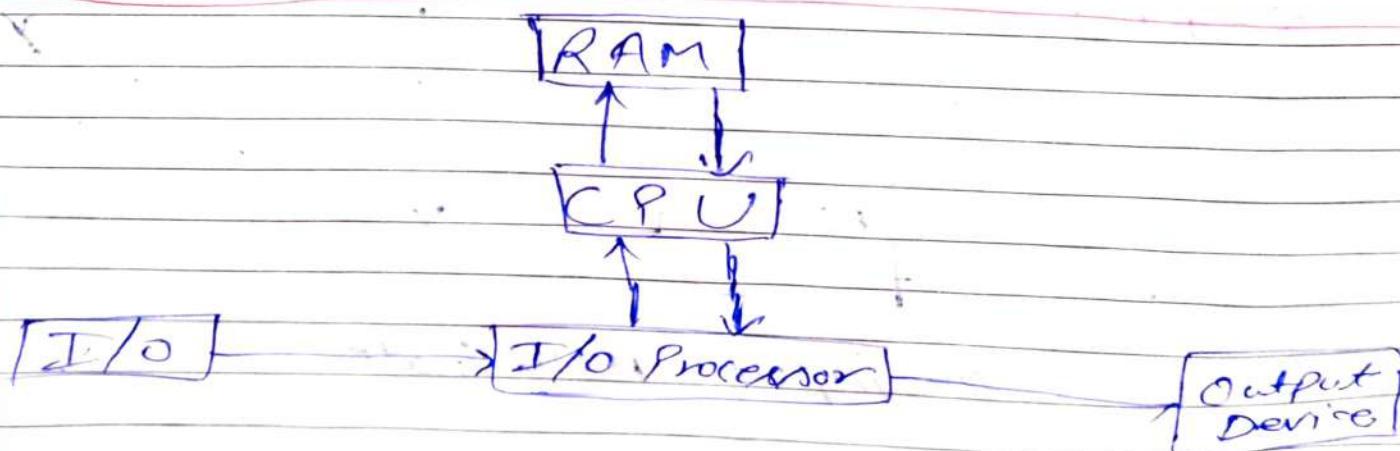
## Unit - I

Page No.

Digital Electronics:- digital Electronics is an area of Computer science. It manages with tools that can transmit out computer software. In digital electronics, we facilitate two-state or binary logic. There are two logic states including "0" (low) & "1" (high).

A computer facilitates a binary numbers, including 1 & 0, using two voltage levels in a machine known as a logic gate. Frequently two states can also be defined using boolean logic circuit. A logic gate creates two inputs & creates an individual output.

It contains the data mechanism, the instruction group, & methods for addressing memory. The structural design of a computer system is concerned with the descriptions of the multiple functional modules, including processor & memories, & managing them together in to an electronic system.



Block diagram of digital electronics

Logic gates:- Logic gates play an important role in circuit design & digital systems. It is a building block of a digital system & an electronic circuit that always have only one output. These gates can have one input or more than one input, but most of the gates have two inputs. On the basis of the relationship b/w the I/O, these gates are named as:-

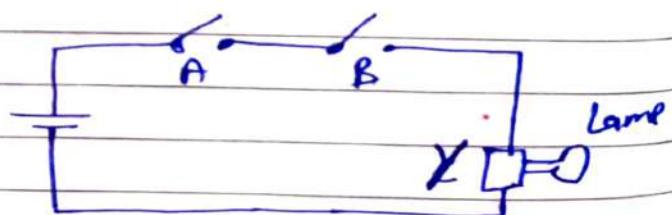
- (1) AND Gate
- (2) OR Gate
- (3) NOT Gate
- (4) NAND Gate
- (5) NOR Gate
- (6) XOR Gate
- (7) XNOR Gate

(1) AND Gate:- This gate works in the same way as the logical operator "and". The AND gate is a circuit that performs the AND operation of two inputs. This gate has a minimum of 2 input values & an output value.

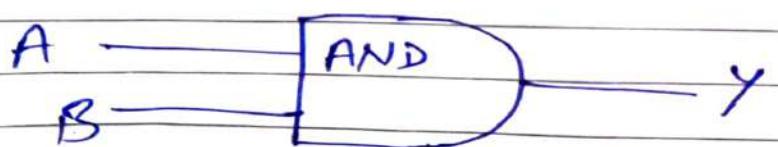
e.g:-  

$$Y = A \cdot B$$
  

$$Y = AB$$



Logic Design:-



2-input AND Gate

Truth Table :-

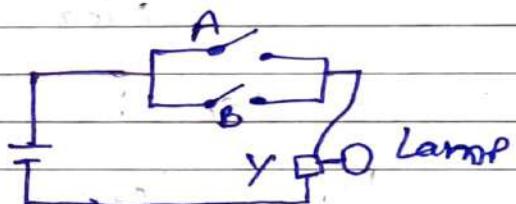
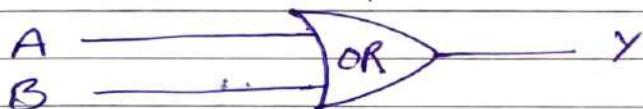
I/P		O/P
A	B	AB
0	0	0
0	1	0
1	0	0
1	1	1

(2) OR Gate :- This gate works in the same way as the logical operator OR.

$$\text{Eg:- } Y = A \text{ OR } B$$

$$Y = A + B$$

Logic Design:-



2- Input OR Gate

Truth Table

Input		Output
A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(3)

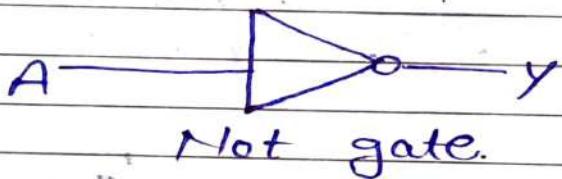
NOT Gate :- This is also called inverter.

This gate gives the inverse value of the input value. as a result. This gate has only one input & one output value.

$$Y = \text{NOT } A$$

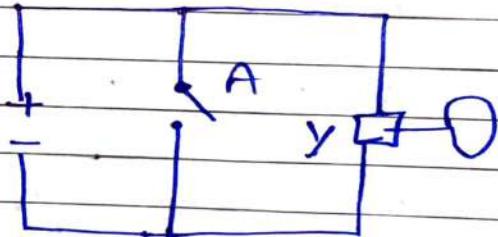
$$Y = A'$$

Logic Design:-



Truth Table:-

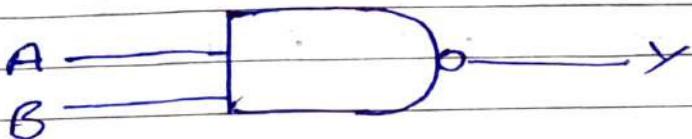
IP	OP
A	B
0	1
1	0



NAND Gate:- This is the combination of AND Gate & NOT Gate. This gate gives the same result as NOT-AND operation. This gate can have two or more than two input values & only one OP value.

Eg:-  $Y = A \text{ NAND } B$

## Logic Design:-



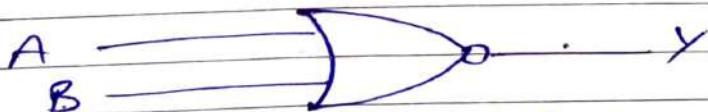
2-input NOR Gate

### Truth Table:-

A	B	$(AB)'$
0	0	1
0	1	1
1	0	1
1	1	0

(5) NOR Gate:- Combination of OR Gate & NOT gate. This gate gives the same result as the NOT-OR operation.

$$Y = A \text{ NOR } B$$



2-input NOR Gate

A	B	$(A+B)'$
0	0	1
0	1	0
1	0	0
1	1	0

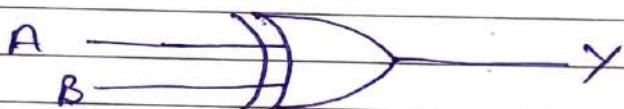
(6) XOR Gate :- The exclusive-OR gate is sometimes called as EX-OR & X-OR gate. XOR gate is used in half & full adder & subtractor.

$$Y = A \text{ XOR } B$$

$$Y = A \oplus B$$

$$Y = AB' + A'B$$

Logic Design:-



2-input XOR Gate

Truth Table :-

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

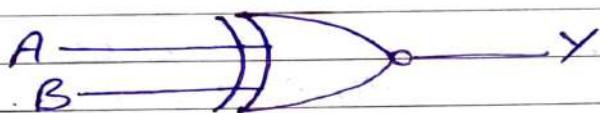
(7) XNOR Gate:- It is used in half & full adder & Subtractor.

$$Y = A \text{ XNOR } B$$

$$Y = A \ominus B$$

$$Y = A'B' + AB$$

Logic Design



2-input XNOR Gate

Truth Table

A	B	$A \ominus B$
0	0	1
0	1	0
1	0	0
1	1	1

## (\*) De-morgan's Theorem / Laws :-

A famous mathematician

De Morgan invented the two most important theorems of boolean algebra. The Demorgan's theorems are used for mathematical verification of the equivalency of the NOR & NAND gates. These theorems play an important role in solving various boolean algebra expressions.

Demorgan's first Law :- According to the first theorem, the complement result of the AND operation is equal to the OR operation of the complement of that variable. Thus, it is equivalent to the NAND function & it's a negative-OR function proving that  $(A \cdot B)' = A'$

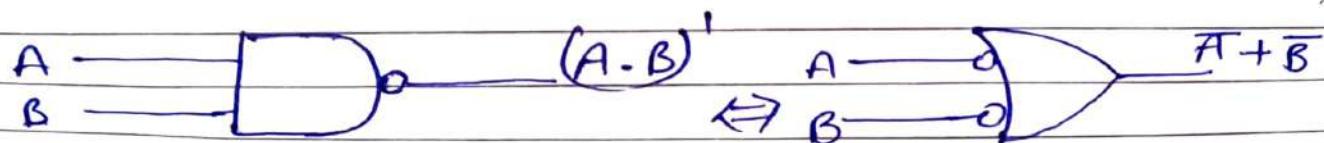
$$(A \cdot B)' = (A' + B')$$

~~NAND~~

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

~~OR~~

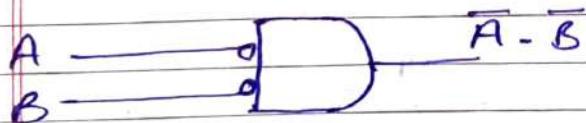
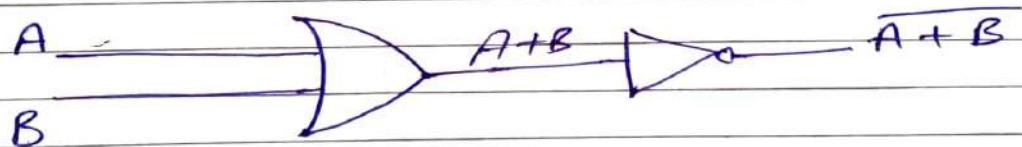
A	B	$A \cdot B$	$(A \cdot B)'$	$A'$	$B'$	$A' + B'$
0	0	0	1	1	1	1
0	1	0	1	1	0	1
1	0	0	1	0	1	1
1	1	1	0	0	0	0



De - Morgan's Second Theorem: - According to the second theorem, the complement result of the OR operation is equal to the AND operation of the complement of that variable.

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

A	B	$A + B$	$(A + B)'$	$A'$	$B'$	$A' \cdot B'$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0



## Laws of Boolean Algebra:-

- (1) Complementary or NOT Law (Inversion Law)
- (2) AND operation
- (3) OR operation Law
- (4) Commutative Law
- (5) Associative Law
- (6) Distributive Law

Note  $\rightarrow$  George Boole developed Boolean algebra.

- (1) Not law:- using NOT gate.



$$Y = \text{NOT } A = \bar{A}$$

$$\begin{array}{l} \bar{0} = 1 \\ \bar{1} = 0 \end{array}$$

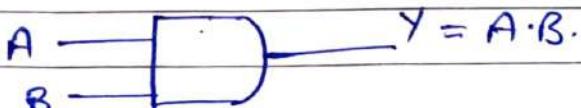
- (2) AND operation Law:- AND operation indicates by the dot (.) . If on doing AND operation of two input A & B, Y output is obtained then this can be expressed by following equation:-
- $$Y = A \cdot B.$$

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

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



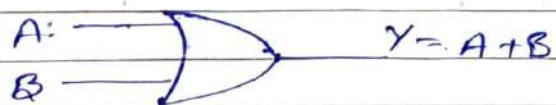
## (3) OR operation laws:-

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

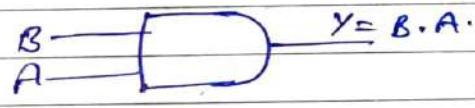
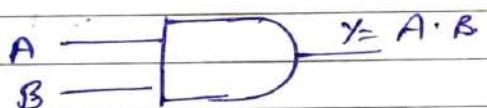
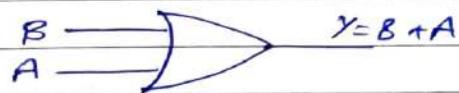
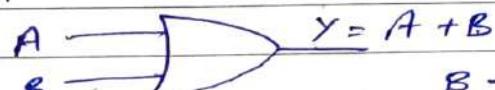
$$A + \bar{A} = 1$$



## (4) Commutative Law:-

$$A + B = B + A$$

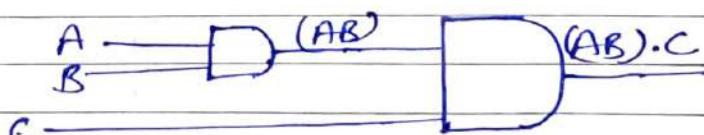
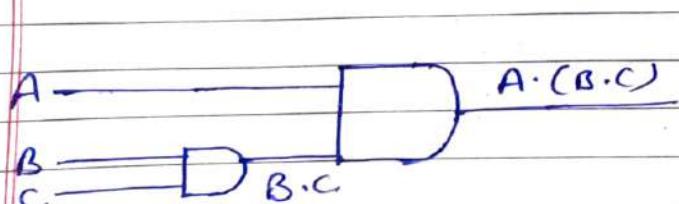
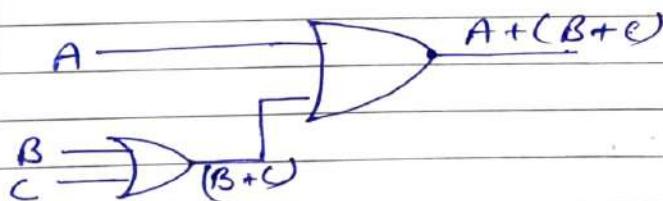
$$A \cdot B = B \cdot A$$



## (5) Associative Law:-

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

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



## (6) Distributive law :-

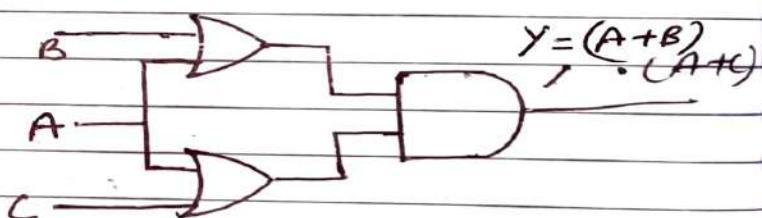
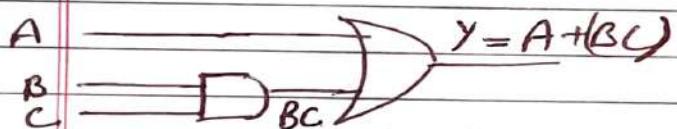
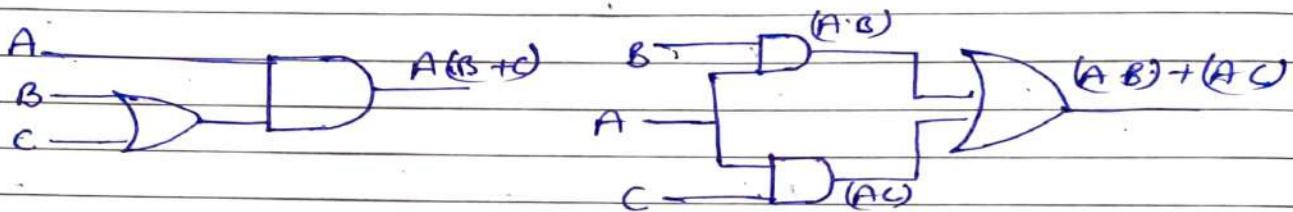
$$A(B+C) = (A \cdot B) + (A \cdot C)$$

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

Proof :-  $A + (B \cdot C) = (A+B)(A+C)$

by truth table

A  $\vee$  B  $\neq$  A  $\vee$  C



## (\*) Some other Laws :-

- (1)  $x + 0 = x$       (10)  $x + y = y + x$
- (2)  $x + 1 = x$       (11)  $x \cdot y = y \cdot x$
- (3)  $x + x = x$       (12)  $x + (y + z) = (x + y) + z = (x + z) + y$
- (4)  $x + \bar{x} = 1$       (13)  $x \cdot (y \cdot z) = (z \cdot y)x = x \cdot y \cdot z$
- (5)  $x \cdot 0 = 0$       (14)  $x \cdot (y + z) = x \cdot y + x \cdot z$
- (6)  $x \cdot 1 = x$       (15)  $(w+x)(y+z) = w \cdot y + w \cdot z + x \cdot y + x \cdot z$
- (7)  $x \cdot x = x$       (16)  $x + x \cdot y = x(1+y) = x \cdot 1 = x$
- (8)  $x \cdot \bar{x} = 0$       (17)  $x + y \cdot z = (x+y)(x+z)$
- (9)  $\bar{\bar{x}} = x$       (18)  $x + \bar{x} \cdot y = x + y$

$$(19) \bar{x} + x\bar{y} = \bar{x} + \bar{y}$$

$$(20) (\bar{x} + y) = \bar{x} \cdot \bar{y}$$

$$(21) \bar{x} \cdot y = \bar{x} + \bar{y}$$

$$(22) \bar{x} + xy = \bar{x} + y$$

$$(23) x + x\bar{y} = x$$

## (\*) Circuit Designing techniques:-

- (1) Sum of Product
- (2) Product of Sum
- (3) K-map

(1) Sum of products (SOP) form :- Group of product terms summed together.

$$Y = AB + AC + BC$$

Sum  $\rightarrow$  OR / (+)  
 Product  $\rightarrow$  AND / (.)

e.g:-  $Y = ABC + BCD + ABD$   
 $Y = \bar{P}\bar{Q} + PQR + PQR$

A, B, C, D, P, Q, R  $\rightarrow$  input terms / literals

(2) Product of sum (POS) form :- Group of sum terms multiplied together.

e.g:-  $Y = (A+B)(B+C)(A+C)$   
 $Y = (A+\bar{B}+\bar{C}) \cdot (A+B) \cdot (A+\bar{C})$   
 $Y = (P+Q) \cdot (P+R) \cdot (\bar{P}+R)$

↓ AND      ↓ OR

→ Standard/Canonical SOP form:- In this each product term consists of all the literals in the complemented or uncomplemented form.

e.g:-  $(A, B, C)$  are literals.

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

→ Standard/Canonical POS form:- In this each sum term consists of all the literals in the complemented or uncomplemented form.

e.g:-  $Y = (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (\bar{A} + \bar{B} + \bar{C})$

(Q) Check whether the following are standard form or not.

(1)  $Y = AB + ABC + \bar{A}BC$  [SOP] Non-standard (SOP)  
 ↳ 'C' is missing.

(2)  $Y = AB + A\bar{B} + \bar{A}\bar{B}$  → standard (SOP)

(3)  $Y = (\bar{A} + B) (A + B) (A + \bar{B})$  → standard (POS)

(4)  $Y = (\bar{A} + B) (A + B + C)$  → Non-standard  
 ↳ 'C' is missing.

(\*) Convert SOP to Standard SOP form:-

(i) for each term find missing literal.

(ii) AND term with the term formed by ORing missing literal & its complement.

e.g:-  $Y = AB + A\bar{C} + BC$  here is three literals A, B, C

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

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

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

now it is a standard SOP.

e.g:-  $Y = A + BC + ABC$ , 3 literals A, B, C

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

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

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

(\*) Convert POS to Standard POS term:-

(i) for each term, find missing literal.

(ii) OR each term with the formed by ANDing missing literal in that term with its complement.

e.g:-  $Y = (A+B)(A+C)(B+\bar{C})$  - 3 literals A, B, C

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

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

$$[A \cdot A = A]$$

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

$$\text{Ans.}$$

$$Y = \frac{(A+B)}{C} \frac{(\bar{B}+C)}{A} - 3 \text{ literals} - ABC$$

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

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

(\*) Minterm & maxterm :-

(1) minterm :- Each individual term in the standard SOP form is called minterm.

e.g:-  $Y = \underline{ABC} + \underline{A\bar{B}\bar{C}} + \underline{\bar{A}BC}$   $\rightarrow$  minterms ( $m_i$ )

(2) Maxterm :- Each individual term in the standard POS form is called maxterm.

e.g:-  $Y = \underbrace{(A+B)}_{\text{+ 3 terms}} \underbrace{(A+\bar{B})}_{\text{+ 3 terms}}$   $\rightarrow$  maxterm ( $M_i$ )

Ques write minterms & maxterms for the following Truth Table:-

			S-SOP	S-POS
A	B	minterms $m_i$	maxterm $M_i$	
0	0	$\bar{A}\bar{B} \rightarrow m_0$	$A+B = M_0$	
0	1	$\bar{A}B \rightarrow m_1$	$A+\bar{B} \rightarrow M_1$	
1	0	$A\bar{B} \rightarrow m_2$	$\bar{A}+B \rightarrow M_2$	
1	1	$AB \rightarrow m_3$	$\bar{A}+\bar{B} \rightarrow M_3$	

SOP  $\rightarrow$  1

POS  $\rightarrow$  0

(\*) How to represent Logical expressions with minterms & maxterms.

$$(1) Y = ABC + \bar{A}BC + A\bar{B}\bar{C}$$

$$m_7 \quad m_3 \quad m_4$$

$$Y = m_7 + m_3 + m_4,$$

$$Y = \sum m(3, 4, 7)$$

Ans.

$$(2) Y = (A + \bar{B} + C) (A + B + C) (\bar{A} + \bar{B} + C)$$

$$M_2 \quad M_0 \quad M_6$$

$$Y = M_2 + M_0 + M_6$$

$$Y = \prod M(0, 2, 6)$$

Ans.

A	B	C	$(\Sigma)$ $m_i$	$(\Pi)$ $M_i$
0	0	0	$\bar{A}\bar{B}\bar{C} \rightarrow m_0$	$A + B + C \rightarrow M_0$
0	0	1	$\bar{A}\bar{B}C \rightarrow m_1$	$A + B + \bar{C} \rightarrow M_1$
0	1	0	$\bar{A}BC \rightarrow m_2$	$A + \bar{B} + C \rightarrow M_2$
0	1	1	$\bar{A}BC \rightarrow m_3$	$A + \bar{B} + \bar{C} \rightarrow M_3$
1	0	0	$A\bar{B}\bar{C} \rightarrow m_4$	$\bar{A} + B + C \rightarrow M_4$
1	0	1	$A\bar{B}C \rightarrow m_5$	$\bar{A} + B + \bar{C} \rightarrow M_5$
1	1	0	$A\bar{B}\bar{C} \rightarrow m_6$	$\bar{A} + \bar{B} + C \rightarrow M_6$
1	1	1	$ABC \rightarrow m_7$	$\bar{A} + \bar{B} + \bar{C} \rightarrow M_7$

Truth table to calculate the minterm & maxterm for given expression.

(\*) How to write standard SOP Expression for a given Truth Table.

- (1) Consider only input combinations whose op 'Y' is 1.
- (2) Write product term for each such combination.
- (3) 'OR' all these product terms.

e.g:-

A	B	Y	
0	0	0	
0	1	1	$\bar{A}B$
1	0	1	$A\bar{B}$
1	1	0	

$\bar{A}B$  } Product

$A\bar{B}$  } terms

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

$$Y = m_1 + m_2 \Rightarrow Y = \sum m(1, 2)$$

e.g:-

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

$$\bar{A}\bar{B}C \rightarrow m_1$$

$$A\bar{B}\bar{C} \rightarrow m_4$$

$$ABC \rightarrow m_7$$

$$Y = m_1 + m_4 + m_7$$

$$Y = \sum m(1, 4, 7) \text{ Ans.}$$

\* How to write standard POS expression of given truth table:-

- (1) Consider only those combinations of input which produces low O/P ( $Y=0$ ).
- (2) write maxterms only for such input combination.
- (3) 'AND' these maxterms.

e.g:-

A	B	C	Y	
0	0	0	0	$A+B+C \rightarrow M_0$
0	0	1	1	
0	1	0	1	
0	1	1	0	$A+\bar{B}+\bar{C} \rightarrow M_3$
1	0	0	1	
1	0	1	0	$\bar{A}+\bar{B}+\bar{C} \rightarrow M_5$
1	1	0	0	$\bar{A}+\bar{B}+C \rightarrow M_6$
1	1	1	1	

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

$$\left. \begin{aligned} Y &= \pi m(0, 3, 5, 6) \\ Y &= m_0 m_3 m_5 m_6 \end{aligned} \right] \text{Ans.}$$

(\*)

Algebraic Simplification of Boolean Expression :-

- Bring the given expression into SOP Form by Boolean laws & DeMorgan's theorems.
- Simplify SOP expression by checking the product terms for common factors.

Q1

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

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

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

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

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

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

$$\boxed{Y = B}$$

Ans

$$\therefore AB + AB = AB$$

$$\therefore B \cdot B = B$$

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

Q2

$$Y = \sum m(2, 4, 6)$$

↳ min term

$$Y = m_2 + m_4 + m_6$$

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

$$m_2 \rightarrow 2 \rightarrow 010$$

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

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

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

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

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

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

$$\because B + B = 1$$

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

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

$$\boxed{Y}$$

Ans.

## K-MAP (KARNAUGH-MAP)

K-map is a graphical method of simplifying Boolean expression.

for 'n' input variable  $\rightarrow 2^n$  boxes in K-map.

K-map is based on Gray Code (Unit distance).

### ② 2-variable K-map :-

A	B	0	1
0	0	1	
1	2	3	

### ③ 3-variable K-map :-

A	BC	00	01	11	10
0	0	1	3	2	
1	4	5	7	6	

BC	A	1
00		
01		
11		
10		

### ④ 4-variable K-map

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

## Relation b/w Truth Table & K-map Entries:-

Truth Table

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

A \ B	0	1
0	0	0
1	0	1

K-map Simplification Rules:-  
(SOP - 1, POS - 0)

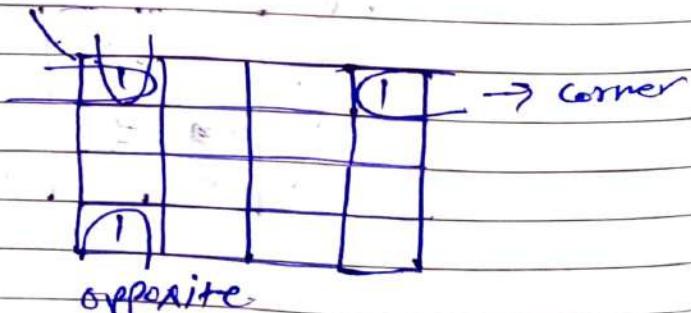
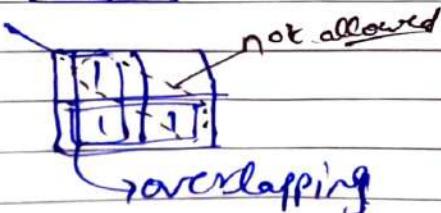
- (1) Groups may not contain zero.
- (2) Grouping in  $2^n$  cells.  $\therefore$  Adjacent cells  $\xleftarrow[2^n]{2^1}$
- (3) making group as large as possible.
- (4) Cells containing containing 1 must be grouped.
- (5) Groups may overlap.
- (6) opposite & corner grouping allowed.
- (7) Diagonal grouping not allowed.

A \ B	0	1
0	1	0
1	1	1

wrong

1	1	1
1	1	1

quad



opposite

(\*) Standard SOP form on K-map:-

- Enter 1's in the cells (Boxes) of K-map corresponding to each minterm present in expression.
- Remaining boxes with 0's.

e.g:-  $Y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ABC$ .

$A, B, C - 3 \text{ terms} \rightarrow 2^3 \rightarrow 8$

Soln:-

		BC	00	01	11	10
		A	10	11	03	02
			14	05	07	06
0	0					
1	1					

(\*) Standard POS form on K-map:-

- Enter 0's in the cells of K-map corresponding to each maxterm present in given expression.
- Remaining boxes with 1's.

$$\begin{bmatrix} 0 - A \\ 1 - \bar{A} \end{bmatrix}$$

e.g:-  $Y = (A+B+C)(A+\bar{B}+C)(\bar{A}+\bar{B}+C)$

$M_0 \quad M_2 \quad M_6$

		$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$	
		A	0	1	1	0
		A	1	1	1	0
0	0					
1	1					

		$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$	
		A	0	1	1	0
		A	1	1	1	0
0	0					
1	1					

IMP

## Simplify Boolean Expression with K-map:-

(i) Use Grouping Technique

(ii) Grouping means combining terms in adjacent cells.

(iii) Grouping adjacent 1's  $\rightarrow$  SOP(iv) Grouping adjacent 0's  $\rightarrow$  POS

Example of Grouping (Pairs) :-

$$(1) Y = \overline{A}BC + \overline{A}B\overline{C}$$

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

A	$\overline{B}C$	$B\overline{C}$	$\overline{B}C$	$BC$	$B\overline{C}$
$\overline{A}$	0	0	1	1	
A	0	0	0	0	

Now, in groups look for variables whose value is not changing.

 $(\overline{A}B)$  Ans.

$$(2) Y = \overline{A}\overline{B}\overline{C}D + A\overline{B}\overline{C}D$$

	$\overline{C}\overline{D}$	$\overline{C}D$	$CD$	$C\overline{D}$
$\overline{A}\overline{B}$	0	1	0	0
$\overline{A}B$	0	0	0	0
$AB$	0	0	0	0
$A\overline{B}$	0	1	0	0

 $(\overline{B}C D)$  Ans

Q.3  $Y = \Sigma m(1, 5, 7, 9, 11, 13, 15)$   
means here is 4 variable (0 to 15)

AB \ CD

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$	
$\bar{A}\bar{B}$	0 0	1 1	3 0	2 0	Quad - 3
$\bar{A}B$	4 0	5 1	8 1	7 0	$\hookrightarrow BD$
$A\bar{B}$	12 0	13 1	15 1	14 0	
$AB$	8 0	9 1	11 1	10 0	

↓ Quad 2      ↓ Quad 1      ↓ AP  
 $\hookrightarrow \bar{C}D$

$$\frac{\bar{C}D + AD + BC}{[D(\bar{C} + A + B)]} \text{ Ans}$$

Q.4  $Y = \Sigma m(1, 3, 5, 9, 11, 13)$ , 4 variable.

AB \ CD

	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$	
$\bar{A}\bar{B}$	0	1 1	1 1	0	
$\bar{A}B$	0	1	0	0	
$A\bar{B}$	0	1	0	0	
$AB$	0	1	1	0	

↓ Quad 2      Quad 1      ↓ CD  
 $\hookrightarrow BD$

$$\bar{B}D + \bar{C}D \Rightarrow [D(\bar{A}C + \bar{B})] \text{ Ans}$$

$$Q5 \quad Y = \Sigma_m (1, 2, 3, 4, 5, 7, 9, 11, 13, 15)$$

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$	Pairs
$\bar{A}\bar{B}$	0	1	1	2	1
$\bar{A}B$	4	1	1	1	0
$A\bar{B}$	13	0	13	15	14
$AB$	8	0	9	11	10

Octet  
D

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

$$Q6 \quad Y = \Sigma_m (1, 2, 9, 10, 11, 14, 15)$$

AB \ CD	$\bar{C}\bar{D}$	$\bar{C}D$	$C\bar{D}$	$CD$	Pairs
$\bar{A}\bar{B}$	0	1	1	2	1
$\bar{A}B$	4	5	7	6	
$A\bar{B}$	12	13	15	14	1
$AB$	8	9	11	10	1

Pair 1      Quad  
 $\bar{B}\bar{C}\bar{D}$       AC

$$\begin{aligned} Y &= \bar{B}\bar{C}\bar{D} + \bar{B}\bar{C}\bar{D} + AC \\ &= \bar{B}(\bar{C}\bar{D} + \bar{C}\bar{D}) + AC \end{aligned}$$

## Elimination of Redundant Group:-

(1)

if all the 1's in a group are already involved in some other groups.

(2) Redundant Group increase the no. of gates required.

$$Q.1 \quad Y = \Sigma m(0, 5, 6, 7, 11, 12, 13, 15)$$

AB\CD	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	3	2
$\bar{A}B$	4	5	7	6
$A\bar{B}$	12	13	15	14
$AB$	8	9	11	10

This Quad is a redundant group

$$Y = \bar{A}\bar{C}D + \bar{A}\bar{B}C + A\bar{B}\bar{C} + ACD + (BD)$$

↳ redundant.  
(to be removed)

$$Q.2 \quad Y = \Sigma m(0, 1, 2, 5, 13, 15)$$

AB\CD	$\bar{C}\bar{D}$	$\bar{C}D$	$CD$	$C\bar{D}$
$\bar{A}\bar{B}$	0	1	3	2
$\bar{A}B$	4	5	7	6
$A\bar{B}$	12	13	15	14
$AB$	8	9	11	10

$$Y = \bar{A}\bar{B}\bar{D} + \bar{A}\bar{C}D + ABD \quad Ans.$$

## (x) Don't Care Condition:-

denoted by (x) & may be assumed 0 or 1 as per requirement for simplification.

$$Q.1 \quad Y = \sum m(1, 3, 7, 11, 15) + d(0, 2, 5)$$

$\bar{A}B$	$\bar{C}D$	$\bar{C}D$	$CD$	$CD$
$\bar{A}\bar{B}$	0 x	1 1	3 1	2 x
$\bar{A}B$	4 0	5 x	7 1	6 0
$A\bar{B}$	12 0	13 0	15 1	14 0
$AB$	8 0	9 0	11 1	10 0

$$Q_2 = \bar{A}\bar{B}$$

$$Q_1 = CD$$

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

$$Q.2 \quad Y = \sum m(0, 1, 5, 9, 13, 14, 15) + d(3, 4, 7, 10, 11)$$

$\bar{A}B$	$\bar{C}D$	$\bar{C}D$	$CD$	$CD$
$\bar{A}\bar{B}$	0 1	1 1	3 x	2
$\bar{A}B$	4 x	5 1	7 x	6
$A\bar{B}$	12 1	13 1	15 1	14 1
$AB$	8 1	9 1	11 x	10 x

$$Q_1 = AC$$

octet D

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

Q. No. of Product terms in the minimized SOP expression obtained through the following K-map. 18 -

- (a) 2 ✓
- (b) 3
- (c) 4
- (d) 5

Product terms	1	0	0	1
	0	1	0	0
	0	0	1	1
	1	0	0	1
				Pair
				Quad

Q.  $F(A, B, C) = \sum m(2, 3, 4, 5) + \sum d(6, 7)$

		BC			
		$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$
$\bar{A}$	$\bar{A}$	0	0	1	1
	A	1	1	X	X

Ans: with out don't care condition

		BC			
		$\bar{B}\bar{C}$	$\bar{B}C$	$BC$	$B\bar{C}$
$\bar{A}$	$\bar{A}$	1	1	1	1
	A	1	1	X	X
					$y = \bar{A}B + A\bar{B}$
					or, $[y = A \oplus B]$
					XOR

with dont care condition.

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

$$\boxed{Y = A + B}$$
 Ans.

## (Ex) POS Simplification - K map :-

(1)  $Y = \text{PI M}(0, 2, 3, 5, 7)$

A \ BC	00	01	11	10
0	0	1	0	2
1	4	5	7	6

 $\therefore$  Grouping  $\rightarrow 0$ 

$0 \rightarrow x$   
 $1 \rightarrow \bar{x}$

Pair 3 =  $(A + C)$

Pair 1 =  $(\bar{B} + \bar{C})$

Pair 2 =  $(\bar{A} + \bar{C})$

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

(2)  $Y = \text{PI M}(0, 2, 3, 7)$

A \ BC	00	01	11	10
0	0	1	0	2
1	4	5	7	6

Pair 1 =  $(\bar{B} + \bar{C})$

Pair 2 =  $(A + C)$

$$Y = (\bar{B} + \bar{C})(A + C) \text{ Ans.}$$

Q.3

		CD	AB		
		00	01	11	10
		00	0		
		01			
		11			
		10	0		0

Sol:-

$$\text{Quadrant} = (B + D) \cdot m_3$$

(\*) K-map using MAX Terms [POS] :-  $0 - x, 1 - \bar{x}$   
 Grouping  $\rightarrow 0's$

Q.1

		BC	A			
		00	01	11	10	
		00	0	0	0	1
		01	1	1	1	1
		11				
		10				

$\Rightarrow G_1, \quad \Rightarrow G_2$

$$\text{Method 1} \rightarrow \bar{F} = G_1 + G_2 \quad \because \text{using SOP}$$

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

$$\begin{aligned} \bar{F} &= (\bar{A}\bar{B}) + (\bar{A}C) \\ F &= (A+B) \cdot (A+\bar{C}) \quad \boxed{\text{Ans}} \quad \because \text{using DeMorgan's law} \end{aligned}$$

$$\begin{aligned} \text{Method 2: } & G_1 = (A+B) \\ & G_2 = (A+\bar{C}) \\ & Y = G_1 \cdot G_2 \end{aligned}$$

$$Y = (A+B) \cdot (A+\bar{C}) \quad \boxed{\text{Ans}}$$

Q2

$$F(A B C D) = \prod M(0, 2, 6, 7, 8, 10, 12, 13)$$

. ↳ maxterms .

AB \ CD

	00	01	11	10
00	0 0	1 1	1 1	1 0
01	1 1	1 1	0 1	0 0
11	1 2	0 0	1 5 1	1 4 1
10	8 0	9 1	11 1	10 0

$$\text{Quad. 1} = (B + D)$$

$$\text{Quad. 1} = (A + \bar{B} + \bar{C})$$

$$\text{Quad. 2} = (\bar{A} + \bar{B} + C)$$

$$Y = (B + D)(A + \bar{B} + \bar{C})(\bar{A} + \bar{B} + C) \quad \text{Ans.}$$

# Combinational building Blocks.

DATE / /  
PAGE NO.

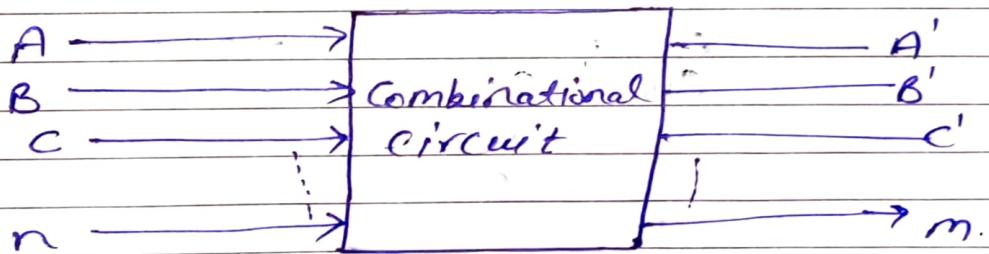
## Unit - II

Combinational circuit is a circuit in which we combine the different gates in circuit.  
eg:- Encoder, Decoder, multiplexer & demultiplexer.

Characteristics of combinational circuit :-

- (1) The output of combinational circuit at any instant of time, depends only on the levels present at input terminals.
- (2) The combinational circuit do not use any memory. The previous state of input does not have any affect on the present state of any the circuit.
- (3) A combinational circuit can have an 'n' number of inputs & m number of outputs.

Block diagram:-



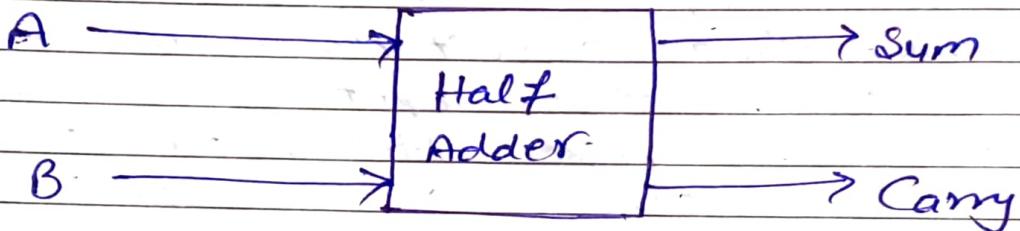
Designing of Combinational circuit

- (1) find the number of inputs & outputs
- (2) write the truth table.
- (3) write the logical Expression.
- (4) Minimize the logical Expression.
- (5) Hardware implementation.

(i)

Half Adder :- The half adder is a basic building block having two inputs & two outputs. The adder is used to perform OR operation of two single bit binary numbers. The Carry & Sum are two output states of the half adder.

Block diagram:-



Truth Table

Inputs		Outputs		
A	B	Sum	Carry	→ MSB
0	0	0	0	
0	1	1	0	
1	0	1	0	
1	1	0	1	

In the above table,

- (1) 'A' and 'B' are the input states, and 'sum' & 'Carry' are the output states.
- (2) The 'Carry' output is 0 in case where both the inputs are not 1.
- (3) The least significant bit of the sum is defined by the 'Sum' bit.

The SOP form of the sum & carry are as follows:-

$$\text{Sum} = \bar{x}'y + xy' \quad \text{min term (SOP) } \bar{x}=0, \bar{y}=1$$

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

Construction of half Adder circuit :-

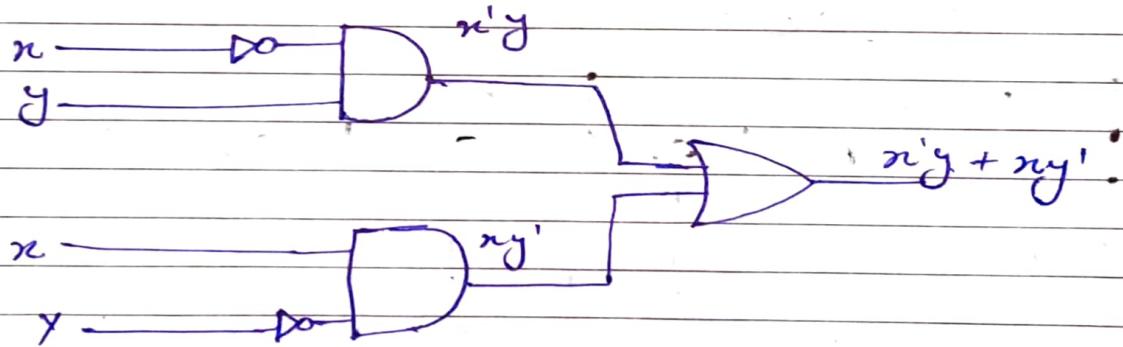
The half adder is designed with the help of the following two logic gates:-

(1) 2-input AND gate.

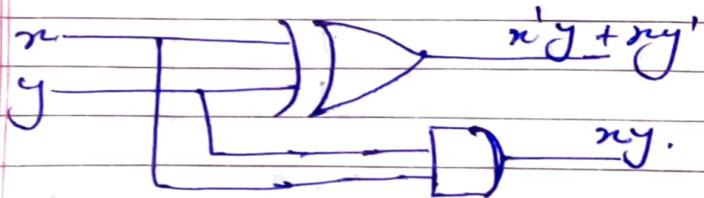
(2) 2-input Exclusive-OR gate or En-OR gate.

$$\text{Sum} = \bar{x}'y + xy' \quad \text{OR} \quad \text{Sum} = \bar{x} \oplus y$$

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



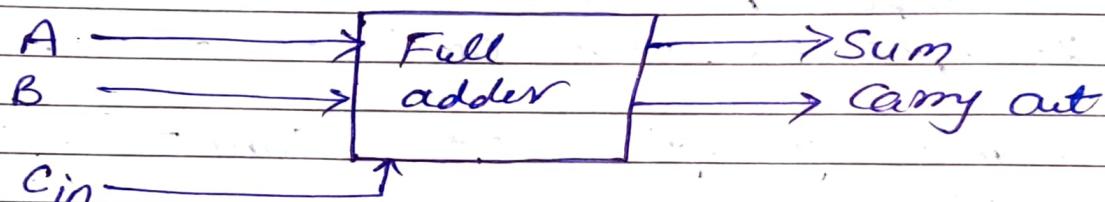
OR



Half-Adder circuit

**Full Adder :-** The half adder is used to add only two numbers. To overcome this problem, the full adder was developed. The full adder is used to add three 1 bit binary numbers A, B & carry C. The full adder has three input states & two output States i.e; Sum & Carry.

Block diagram :-



Truth Table

Inputs			Outputs	
A	B	Cin	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

(1) 'A' & 'B' are the input variables. These variables represent the two significant bits which are going to be added.

- (2) 'Cin' is the third input which represents the carry from the previous lower significant position, the carry bit is latched.
- (3) The sum & carry are the output variables that define the output values.
- (4) The eight rows under the input variable designate all possible combinations of 0 & 1 that can occur in these variables.

The SOP form can be obtained with the help of K-map as:-

	$x'y'z$	$x'yz$	$xy'z'$	$xyz$
$x'y'z$	00	01	11	10
$x'yz$	0	1	1	1
$xy'z'$	1	1	1	1
$xyz$	1	1	1	1

$$\text{Sum} = x'y'z + x'yz + xy'z' + xyz$$

	$x'y'z$	$x'yz$	$xy'z'$	$xyz$
$x'y'z$	00	01	11	10
$x'yz$	0	1	1	1
$xy'z'$	1	1	1	1
$xyz$	1	1	1	1

$$\text{Carry} = xy + xz + yz$$

Construction of full adder using half adder:-

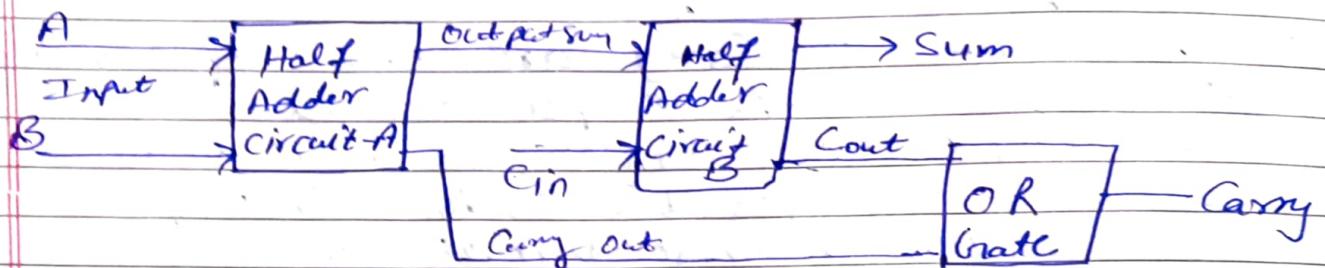
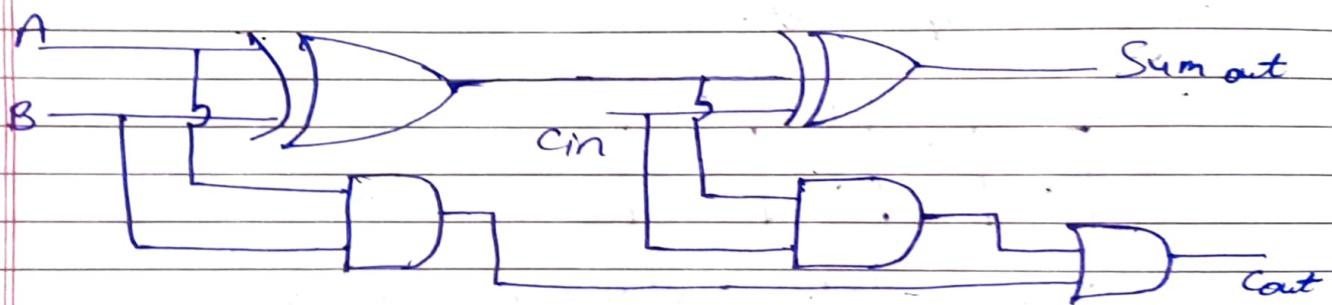


Fig:- Block diagram of full adder



full adder circuit

Sum :- \* Perform the XOR operation of input A & B.

(\*) Perform the XOR operation of the outcome with Cin. So, the sum is  $(A \oplus B) \oplus Cin$  which is also represented as  $(A \oplus B) + Cin$ .

Carry:- \* perform the 'AND' operation of input A & B.

(\*) Perform the 'XOR' operation of input A & B.

(\*) Perform the 'OR' operations of both the OP that come from the previous two steps. So the 'Carry' can be represented as:-  $A \cdot B + (A \oplus B)$

**Subtractor :-** A subtractor is a combinational logic circuit that can perform the subtraction of two numbers (binary) & produce the difference between them. It is a combinational circuit that means its output depends on its present inputs only. There are two types of subtractors.

- (1) Half Subtractor
- (2) Full Subtractor

(1) Half Subtractor :- It is a combinational circuit which is used to perform subtraction of two bits. It has two inputs & two outputs. Output is known as Difference & Borrow. The logic symbol & truth table are shown below:-

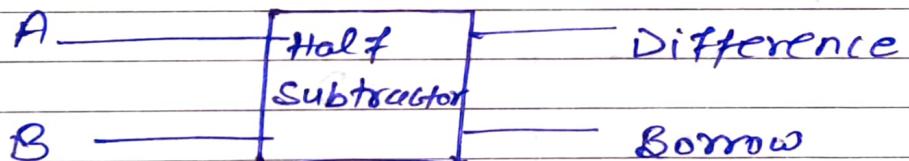


fig:- Logic symbol. of Half Subtractor

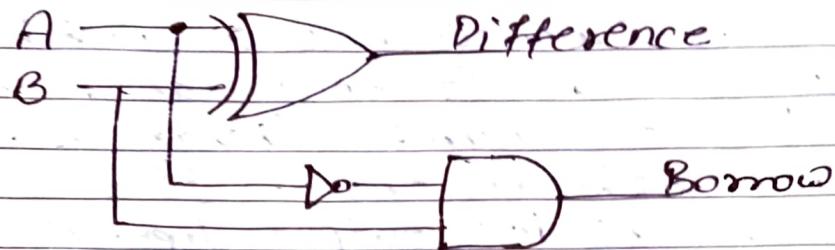
Truth Table :-

Inputs		Outputs	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Boolean Expression

$$\text{Diff} = A \oplus B$$

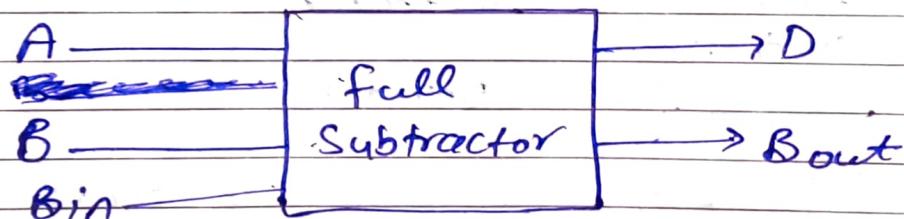
$$\text{Borrow} = A'B$$



\* Circuit diagram \*

② full subtractor :- It is a combinational circuit that performs subtraction involving three bits, namely A, B & Bin (Borrow-in). It accepts three inputs A, B & Bin & it produces two outputs: D (difference) & Bout (Borrow out).

Logic Symbol :-

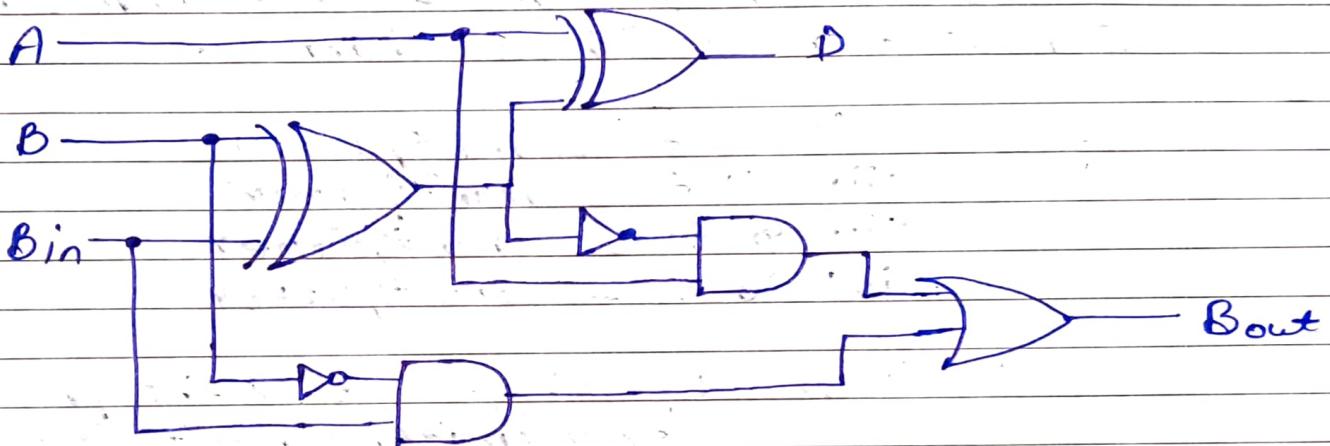


A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Boolean Expression :-

$$D = A \oplus B \oplus B_{in}$$

$$B_{out} = A'B_{in} + A'B + BB_{in}$$

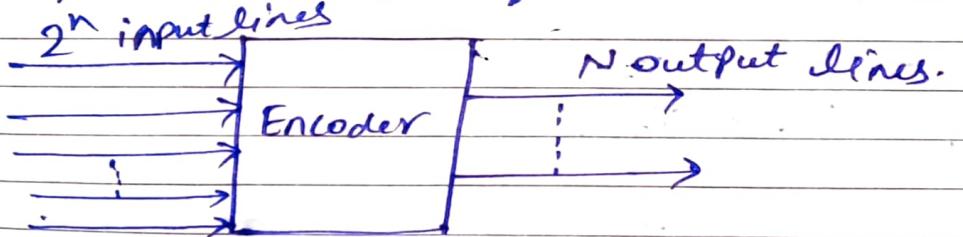


circuit diagram of full subtractor

**Encoder in Digital Logic :-** An encoder is a digital circuit that converts a set of binary inputs in to a unique binary code. The binary code represents the position of the input & is used to identify the specific input that is active. Encoders are commonly used in digital system to convert a parallel set of inputs in to a serial code.

The combinational circuits that change the binary information into  $N$  output lines are known as Encoders. The binary information is passed in the form of  $2^N$  input lines. The output lines define the  $N$ -bit code for the binary information.

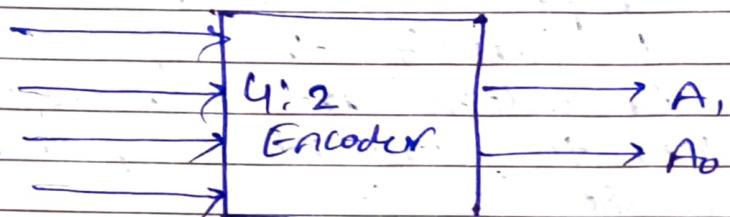
Encoder performs the reverse operation of the decoder. At a time, only one input line is activated for simplicity. The produced  $N$ -bit output code is equivalent to the binary information.



There are various types of encoders which are as follows:-

- (1) 4 to 2 line Encoder :- In 4 to 2 line encoder, there are total of four inputs i.e.,  $Y_0, Y_1, Y_2, Y_3$  & two outputs i.e.,  $A_0, A_1$ .

In 4-input lines, one input line is set to true at a time to get the respective binary code.

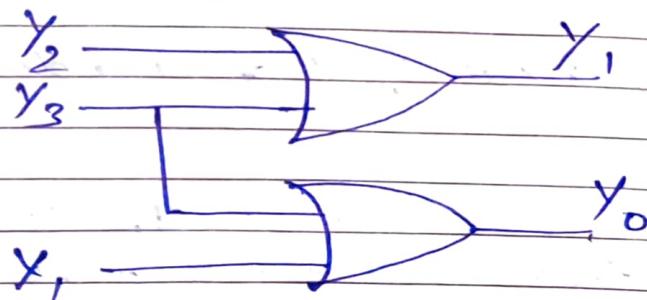


INPUTS				Outputs	
$y_0$	$y_1$	$y_2$	$y_3$	$A_1$	$A_0$
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

Boolean Expression:-

$$\begin{aligned} A_1 &= Y_2 + Y_3 \\ A_0 &= Y_1 + Y_3 \end{aligned}$$

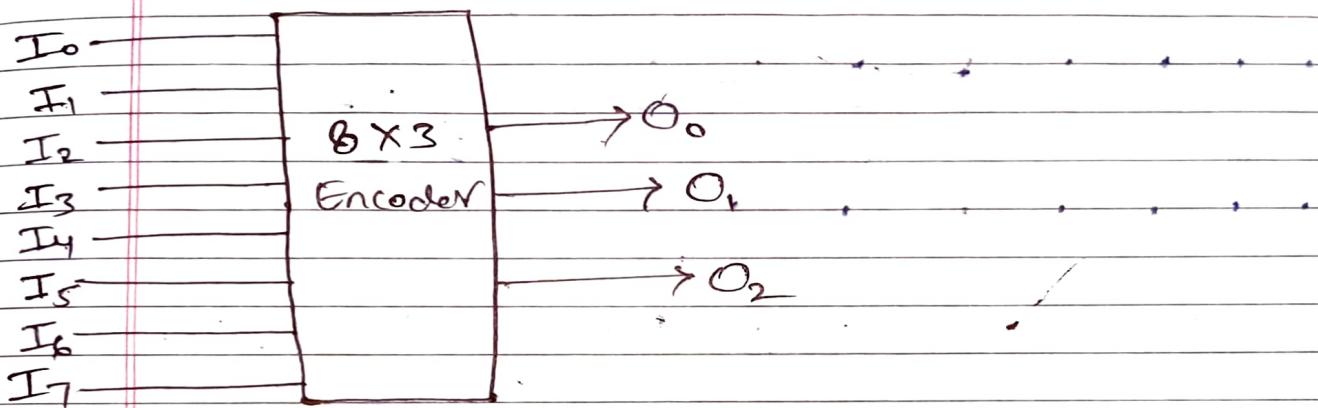
Logic Diagram



(\*) 8 : 3 Encoder (Octal to binary Encoder) :-

8 → Input line

3 → Output line.



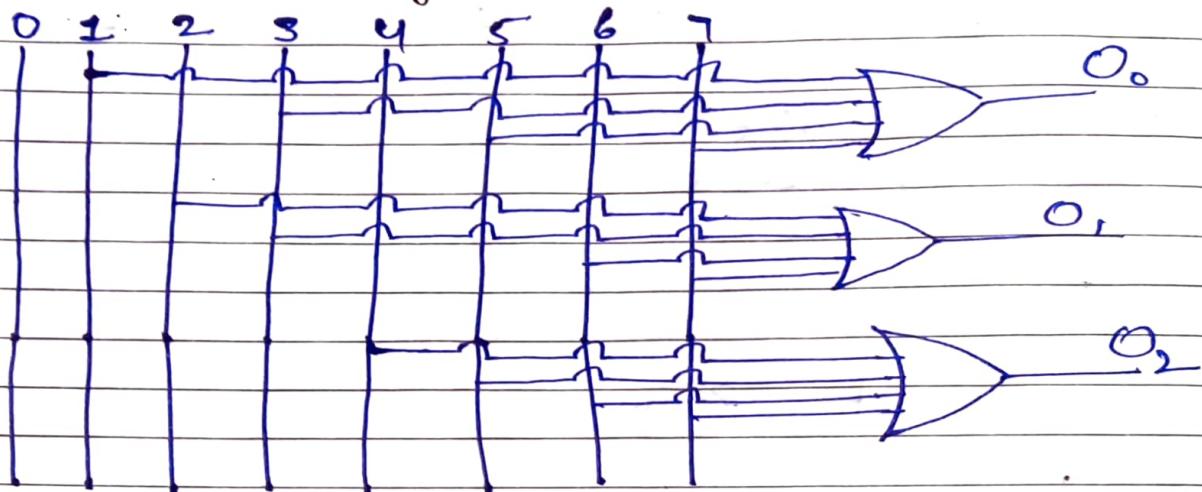
	Input								Output		
	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1

$$O_2 \rightarrow I_4 + I_5 + I_6 + I_7$$

$$O_1 \rightarrow I_2 + I_3 + I_6 + I_7$$

$$O_0 \rightarrow I_1 + I_3 + I_5 + I_7$$

# Logic diagram



(+) Decimal to BCD Encoder :- 10; 4 line Encoder

10 - Input line

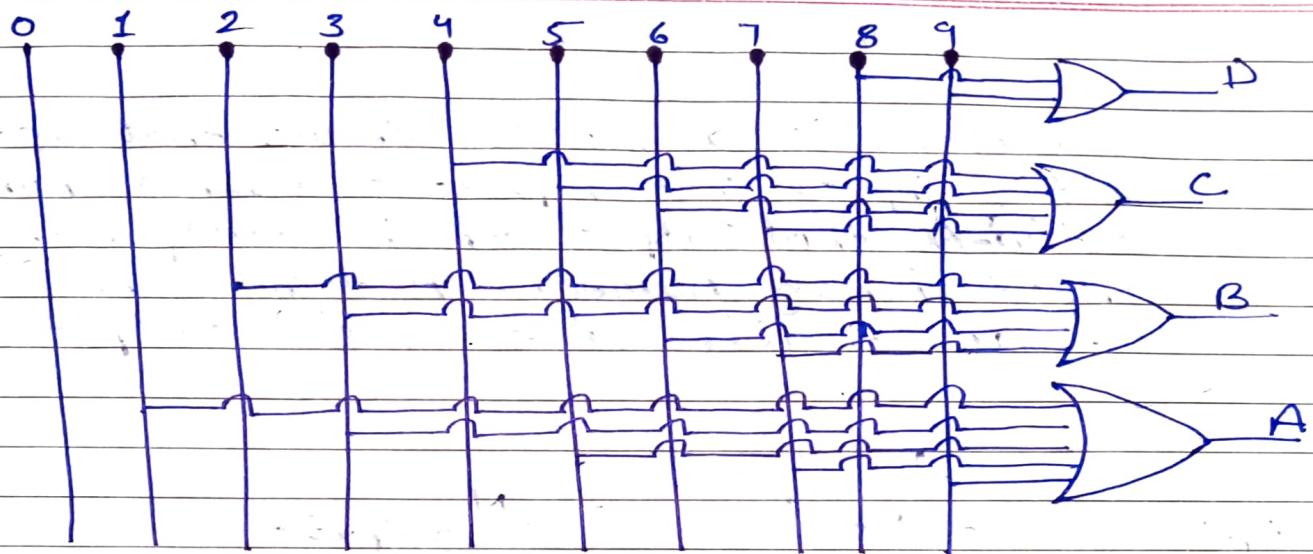
4 - output line

Input	Output								
I <sub>0</sub>	D	C	B	A		1	3	Decimal	A
0	0	0	0	0		1	3	to	B
1	0	0	0	1		1	1	BCD	C
2	0	0	1	0		1	1	encoder	D
3	0	0	1	1		9	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					

Decimal to BCD encoder

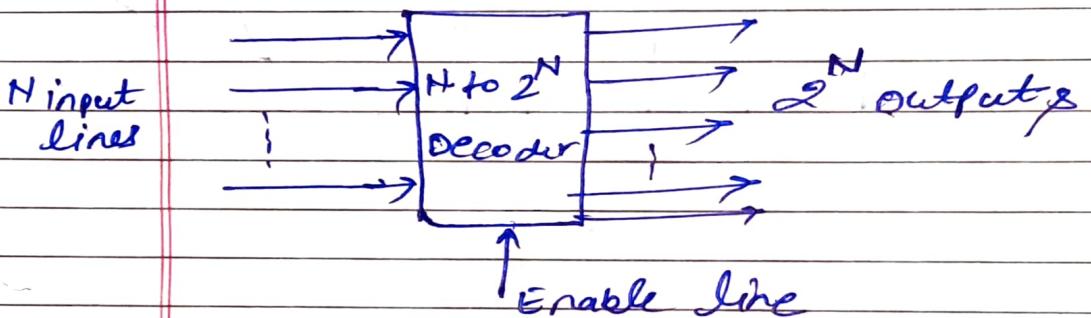
$D \rightarrow 8 + 9$   
 $C \rightarrow 4 + 5 + 6 + 7$   
 $B \rightarrow 2 + 3 + 6 + 7$   
 $A \rightarrow 1 + 3 + 5 + 7 + 9$

## Decimal



## Decoder in digital Electronics:-

A decoder is a logic circuit that accepts a set of inputs that represent a binary number & activates that output which corresponding to the input binary number. A decoder has 'n' inputs & an enable line & ' $2^n$ ' output lines.



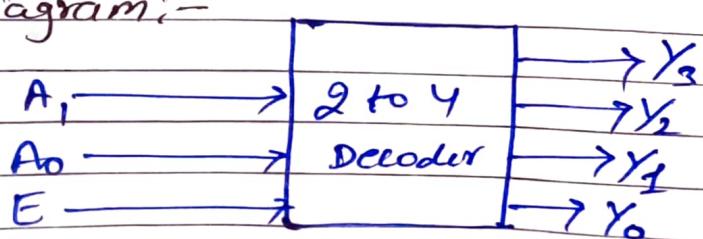
### ④ 2 to 4 Decoder in Digital Electronics

It is a combinational circuit that converts the 2 bit binary information into 4 bit binary info. on basis of Enable signal.

Inputs :-  $A_0, A_1, E$  (enable line)

Outputs :-  $Y_0, Y_1, Y_2, Y_3$

Block Diagram:-



Truth Table :- when the Enable Signal (E) is 1,  
One of the outputs is 1 & the rest  
corresponds to 0. Here is

Enable	Input		Output			
E	A <sub>1</sub>	A <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

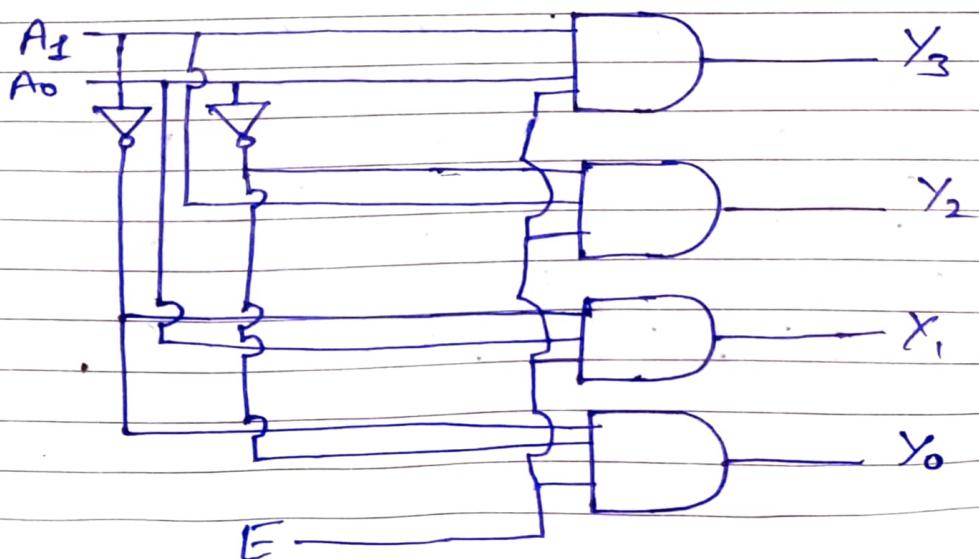
$$Y_3 = E, A_1 \cdot A_0$$

$$Y_2 = E A_1' A_0'$$

$$Y_1 = E A_1' A_0$$

$$Y_0 = E A_1 A_0'$$

Logic circuit :-

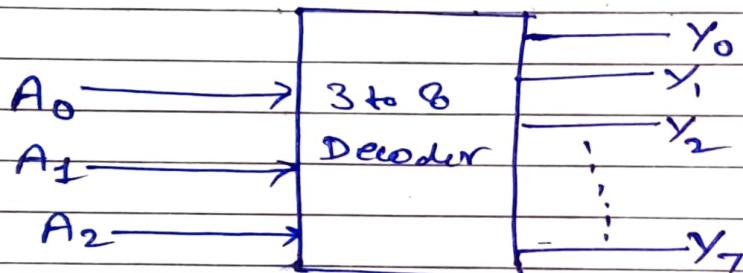


3 to 8 Decoder :- The 3 to 8 Decoder is responsible for converting 3-bit data to 8-bit data. It can be better understood by keeping in mind, that from 3 bits of data, maximum 8 numbers of combinations are possible.

Input:-  $A_0, A_1, A_2$

Output:-  $y_0, y_1, y_2, y_3, y_4, y_5, y_6, y_7$

## Block Diagram :-



## Truth Table

$$Y_0 = A'_0 \quad A'_1 \quad A'_2$$

$$Y_1 = A_0 \quad A'_1 \quad A'_2$$

$$Y_2 = A'_0 \quad A_1 \quad A'_2$$

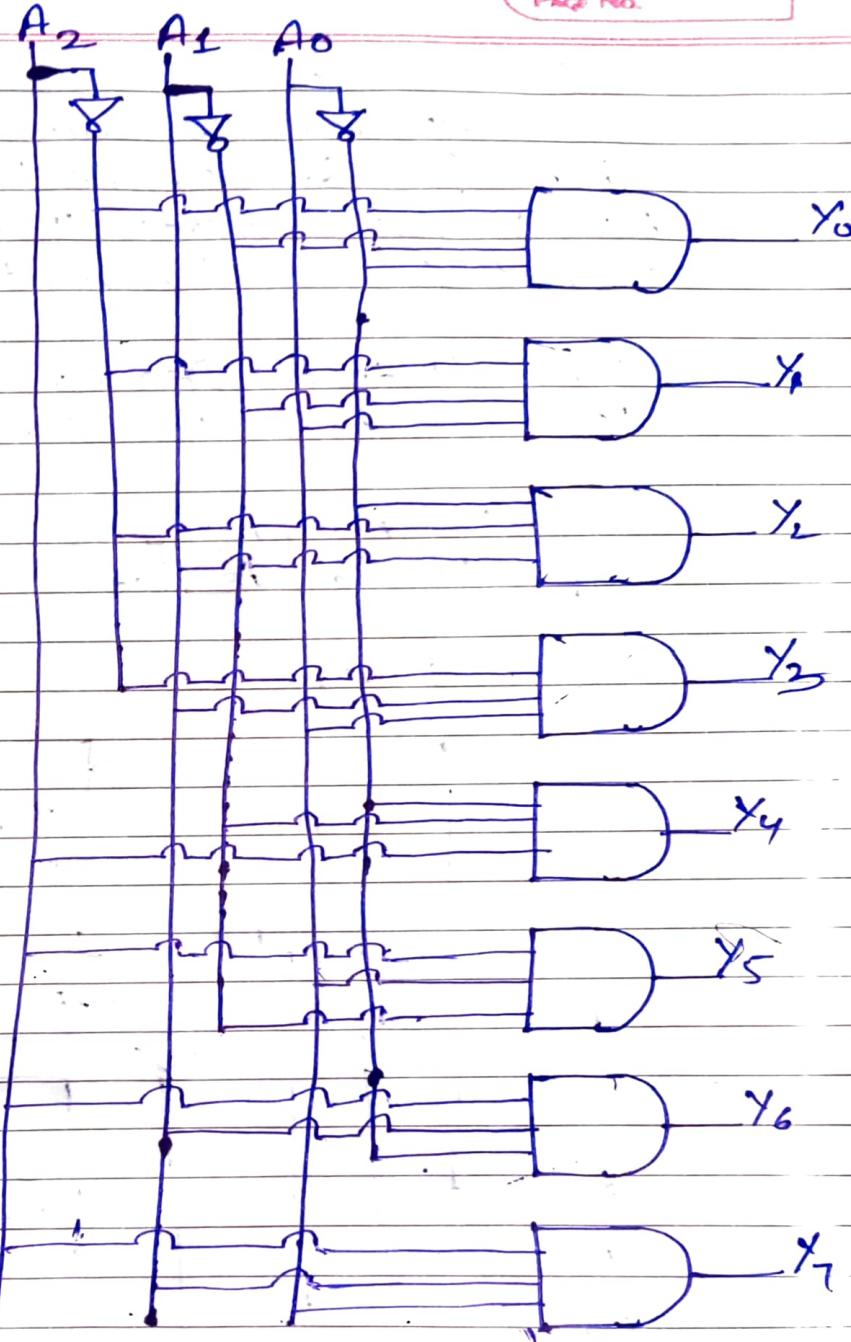
$$Y_3 = A_0 \quad A_1 \quad A'_2$$

$$Y_4 = A'_0 \quad A'_1 \quad A_2$$

$$Y_5 = A_0 \quad A'_1 \quad A_2$$

$$Y_6 = A'_0 \quad A_1 \quad A_2$$

$$Y_7 = A_0 \quad A_1 \quad A_2$$



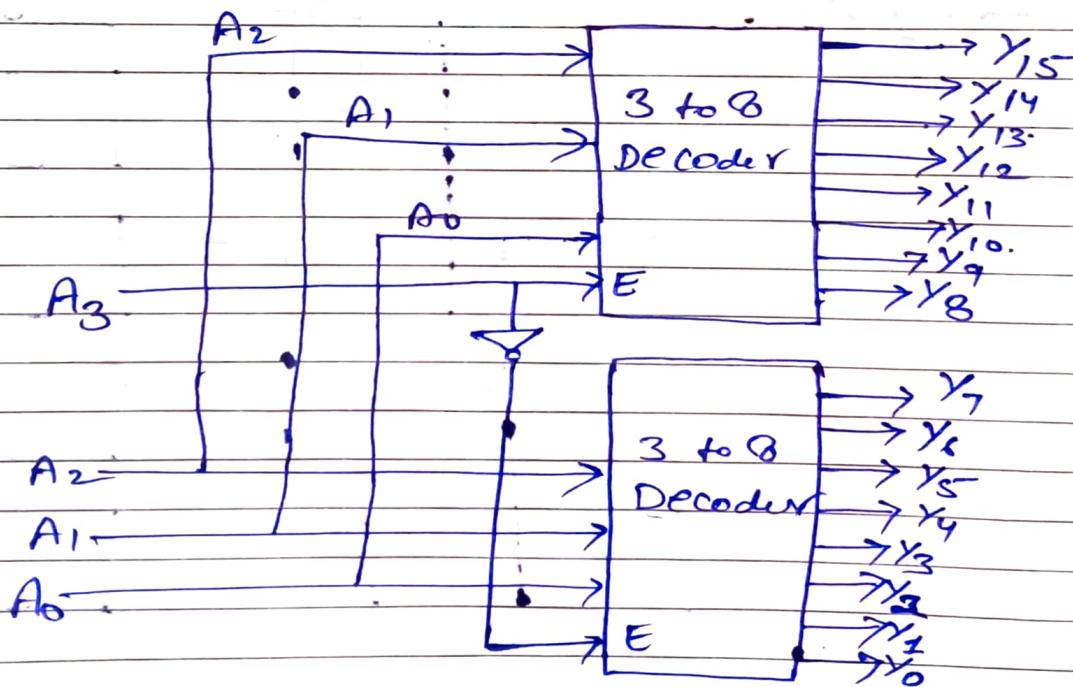
② 4 to 16 Decoder :- Now since the maximum number of combinations possible from 4 bits is 16. So, the 4 to 16 Decoder in digital Electronics converts 4-bit input data into 16-bit output binary information.

This 4 to 16 Decoder is constructed using two 3 to 8 Decoders.

Inputs:-  $A_0, A_1, A_2$

outputs:-  $y_0$  to  $y_{15}$

Block Diagram:-



Inputs	$A_3$	$A_2$	$A_1$	$A_0$	Outputs
— — — —	—	—	—	—	$Y_5$
— — — —	—	—	—	—	$Y_4$
— — — —	—	—	—	—	$Y_3$
— — — —	—	—	—	—	$Y_2$
— — — —	—	—	—	—	$Y_1$
— — — —	—	—	—	—	$Y_0$
— — — —	—	—	—	—	$Y_5$
— — — —	—	—	—	—	$Y_4$
— — — —	—	—	—	—	$Y_3$
— — — —	—	—	—	—	$Y_2$
— — — —	—	—	—	—	$Y_1$
— — — —	—	—	—	—	$Y_0$
— — — —	—	—	—	—	$Y_5$
— — — —	—	—	—	—	$Y_4$
— — — —	—	—	—	—	$Y_3$
— — — —	—	—	—	—	$Y_2$
— — — —	—	—	—	—	$Y_1$
— — — —	—	—	—	—	$Y_0$

Multiplexer:- Multiplexer, also known as MUX, are essential components in digital electronics. Multiplexer in digital electronics is widely used for data selection, signal routing, & address decoding in microprocessors & microcontrollers. They are also used in communication systems to transmit multiple signals over a single channel.

Types:- (i) 2 to 1 multiplexer

(ii) 4 to 1 multiplexer

(iii) 8 to 1 multiplexer

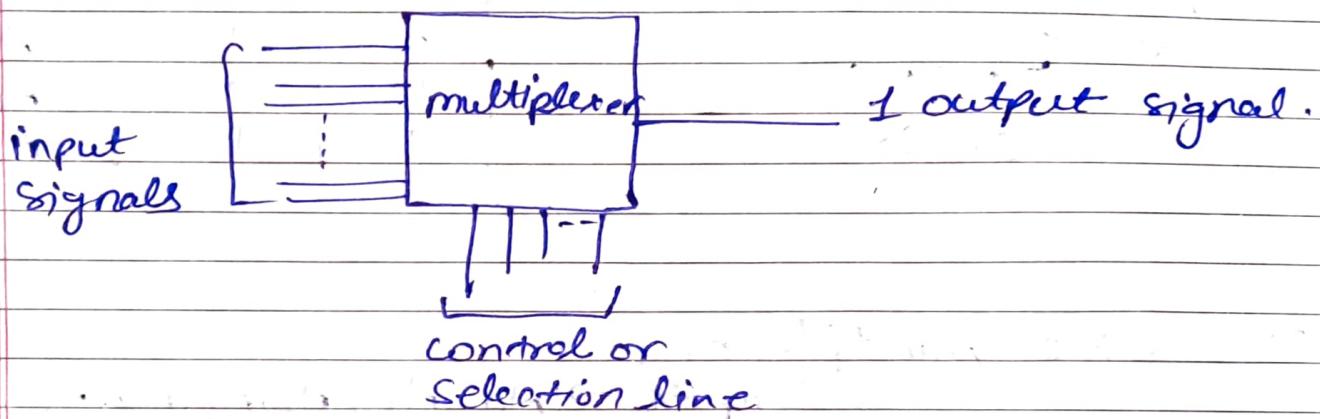
(iv) 16 to 1 multiplexer

What is multiplexing:- It is the process of transmitting multiple signals or data streams over a single communication channel or transmission medium. This is achieved by combining multiple input signals into a single output signal and then transmitting it over a shared communication channel.

Multiplexer in Digital Electronics:- A multiplexer in digital electronics is known as a data selector. It is a combinational logic circuit having multiple input lines, one output line, & many select line or control lines. It receives binary information from several input lines & routes it to a single output line based on a set of select/control lines. Multiplexer in digital electronics is also known as many to one

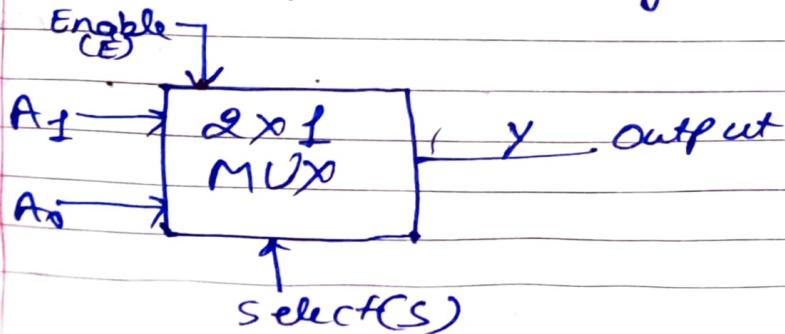
combinational circuits.

The block diagram below depicts a multiplexer with  $n$  input lines,  $m$  selection lines, and one output line. If there are  $m$  selection/control lines, the number of possible input lines are  $2^m$ .



Types:-

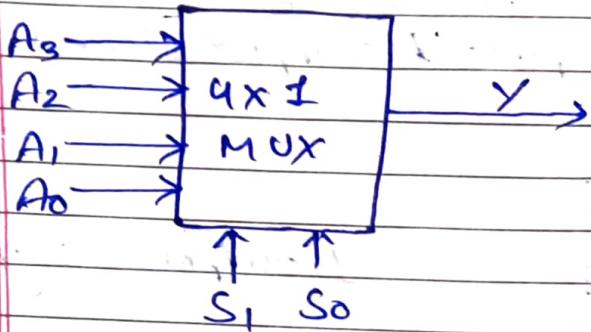
(1)  $2 \times 1$  multiplexer:- It is a combinational logic circuit that has only two inputs i.e.  $A_0$  &  $A_1$ , 1 selection line, i.e.  $S_0$  & single output  $Y$ . On the basis of the combination of inputs that are present at the selection line  $S_0$ , one of these 2 inputs forwards to the output line. It is the simplest type of multiplexer in digital electronics.



Truth Table:-

Inputs	Outputs
$S_0$ 0	$y$ $A_0$
1	$A_1$

(2)

4x1 multiplexer :- input  $\rightarrow A_0, A_1, A_2, A_3$ output  $\rightarrow Y$ selection line  $\rightarrow S_1, S_0$ 

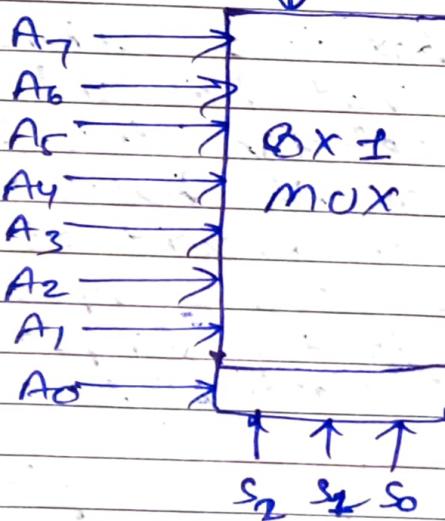
Inputs		Outputs	
S <sub>1</sub>	S <sub>0</sub>	S <sub>0</sub>	Y
0	0	0	A <sub>0</sub>
0	1	1	A <sub>1</sub>
1	0	0	A <sub>2</sub>
1	1	1	A <sub>3</sub>

(3)

8x1 multiplexer :-

Inputs  $\rightarrow A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7$ Selection line  $\rightarrow S_2, S_1, S_0$ Output line  $\rightarrow Y$ 

Enable(E)



Inputs			Outputs	
S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	S <sub>0</sub>	Y
0	0	0	0	A <sub>0</sub>
0	0	1	1	A <sub>1</sub>
0	1	0	0	A <sub>2</sub>
0	1	1	1	A <sub>3</sub>
1	0	0	0	A <sub>4</sub>
1	0	1	1	A <sub>5</sub>
1	1	0	0	A <sub>6</sub>
1	1	1	1	A <sub>7</sub>

Memory:- A memory is just like a human brain. It is used to store data & instructions. Computer memory is the storage space in the computer, where data is to be processed & instructions required for processing are stored. The memory is divided into large number of small parts called cells. Each cell/location has a unique address, which varies from zero to m/r size minus one.

Memory is categorised in 3 following types-

- (1) Cache memory
- (2) Primary memory/main memory
- (3) Secondary memory.

(1) Cache memory:- It is very high speed semiconductor memory which can speed up the CPU. It acts as a buffer between the CPU & the main memory. It is used to hold those parts of data & program which are most frequently used by the C.P.U. The parts of data & programs are transferred from the disk to Cache memory by operating system, from where the C.P.U. can access them.

Advantages:- (1) faster than main memory

- (2) It consumes less access time as compared to main memory.
- (3) It stores the program that can be executed within a short period of time.
- (4) It stores data for temporary use.

Disadvantage:- (1) It has limited capacity

- (2) very expensive.

(2) Primary memory:- primary memory holds only those data & instructions on which the computer is currently working. It has a limited capacity & data is lost when power is switched off. It is generally made up of semiconductor device. These memories are not fast as registers. The data & instruction required to be processed resides in the main memory. It is divided into two categories:-

- ① RAM
- ② ROM

## Characteristics of main memory:-

1. These are semiconductor memories.
2. It is known as the main m/r.
3. Usually volatile memory.
4. Data is lost in case of power is switched off.
5. It is the working memory of the computer.
6. Faster than secondary memory.
7. A computer can not run without the primary m/r.

3. Secondary memory:- It is also known as external memory or non-volatile.  
It is slower than the main memory. These are used for storing data permanently. CPU directly does not access these memory, instead they are accessed via input output routines. The contents of secondary memories are first transferred to the main memory, & then the CPU can access it.  
e.g:- HDD, DVD etc.

## Characteristics of secondary memory:-

1. These are magnetic & optical memories.
2. also known as backup memory.
3. non-volatile memory.
4. Data is permanently stored even if power is off.
5. used to store data in computer.
6. Computer may run without secondary memory.
7. slower than primary memories.

## ⊗ RANDOM ACCESS MEMORY [R.A.M.] :-

RAM is the internal memory of the C.P.U. for storing data, program & program result. It is a read/write memory which stores data until the machine is working. As soon as the machine is switched off, data is erased. RAM is small, both in terms of its physical size & in the amount of data it can hold.

RAM is of two types:- (1) Static RAM [SRAM]

(2) Dynamic RAM [DRAM]

(1) Static RAM:- The word static indicates that the memory retains its contents as long as power is being supplied. SRAM chips use a matrix of 6-transistors & no capacitors.

Transistors do not require power to prevent leakage, ~~so~~ so SRAM need not be refreshed on regular basis. SRAM is thus used as cache memory & has very fast access.

Characteristics:- (1) Long life. (2) No Need to refresh  
(3) Faster (4) used as Cache memory  
(5) Large size (6) Expensive

(7) High Power consumption

(2) Dynamic RAM:- Unlike SRAM, DRAM must be continuously refreshed in order to maintain the data. This is done by placing the memory on a refresh circuit that rewrites the data several hundred times per second.

DRAM is used for most system memory as it is cheap & small. All DRAMs are made up of memory cells, which are composed of one capacitor & one transistor.

### Characteristics of DRAM:-

- (1) Short data lifetime.
- (2) refreshed continuously
- (3) slower than SRAM
- (4) smaller in size
- (5) less expensive.
- (6) less power consumption

### \* Read Only Memory [ROM]:-

The memory from which we can only read but cannot write on it. This type of memory is non-volatile. The information is stored permanently in such memories during manufacture. A ROM stores such instructions that are required to start a computer. This operation is referred to as bootstrap.

#### Types of ROMs:-

- (1) MROM [MASKED ROM]:- The very first ROMs were hard-wired devices that contained a preprogrammed set of data or instructions. These kinds of ROMs are known as MROM, which are inexpensive. It is programmed by IC manufacturer.

## 2. PROM [Programmable Read only memory]:-

It is a computer memory chip that can be programmed once after it is created. Once the PROM is programmed, the information written is permanent & can not be erased or deleted. PROM was first developed by Wen Tsing Chow in 1956. Programmed by user.

e.g:- BIOS in early systems.

Today's, PROM in computers has been replaced by EEPROM.

## 3. EPROM [Erasable Programmable Read only Memory]:-

It is a memory chip that does not lose data even when the power is switched off. This is a non-volatile memory. Each EPROM is individually programmed by an electronic device. After that the data can be erased by exposing the ~~EPROM~~ EPROM to strong ultraviolet light.

Advantage :-

(\*) Non volatile

(\*) quite effective

(\*) It is reprogrammable.

Disadvantage :- Transistors used in EPROM have higher resistance.

(\*) Needs U.V. light to erase the data.

(\*) not possible to erase a particular byte of data in EPROM. Whole data is deleted.

(\*) It takes some time to erase the data in EPROM.

Secondary Storage Devices:- If we need to store large amount of data permanently, we need a cheaper & permanent memory. Such memory called secondary memory.

Types of Secondary storage devices:-

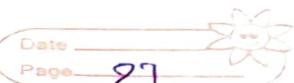
1. Floppy Disk [F.D.]- It is a type of storage media that reads data storage. It is also known as a floppy diskette, floppy, or floppy disk that is used to store data. It was extremely expensive as it was one of first types of hardware storage created in 1967 by IBM.

It contained four basic components:-

- (\*) Magnetic read/write heads.
- (\*) Containing a lever with a frame that helps to open & close the device.
- (\*) Containing all of the electronics, it includes a circuit board.
- (\*) It is placed through a spindle clamping device because it is spinning 300 to 360 rotations every minute.

Types of floppy Disk:- There are three type-

(1) 8-Inch Drive:- In the early 1970's, 8-inch was the first floppy design that was used as read only format then became able for both read & write storage 80 kb.



2.  $5\frac{1}{4}$  inch Drive:- during 1980s, a  $5\frac{1}{4}$ -inch floppy disk drive was produced that was widely in use on PCs. In 1990s,  $5\frac{1}{4}$  inch floppy were also included on computers that could have the ability to store data between 360 KB & 1.2 MB.

3.  $3\frac{1}{2}$  Inch Drive:- This drive is enclosed in plastic, which can hold 1.44 mb on high-density disk & 730 KB on a double density disk.

Advantages:- (1) Portability (2) compatibility

Disadvantages:- (1) Speed (2) Storage (3) File corrupt  
(4) Lack of Reliability (5) Physical damages.

2. C.D. [Compact Disc]- A circular disk introduced by James Russell. It is 4.75 in diameter, which is a flat, round, portable storage medium used to record, store & playback audio, video & other data. On 17-Aug-1982, in Germany, the first CD was created at a Philips factory. It can store data up to 700 mb. It stores data as small notches & read with the help of laser from an optical drive & notches are converted into usable data by drives.

different types of C.D. :-

(1) CD-ROM :- It allows the computer to read data, which is already stored in CD, it can not be deleted or change.

(2) Recordable CD (CD-R) :- Also known as ED-WORM (Write once read many) or CD-WO (write once) - Sony + Philips jointly developed it.

3. CD+R :- A group of companies developed the +R format. It was developed to increase the amount of storage available on a compact disc. ~~CD~~ CD+R has twice storage than CD-R.

4. Rewritable CD (CD-RW) :- It can be used to write data a number of times, erased & reused, and also used as normal CD-R.

5. Video CD (VCD) :- It was a CD including moving images & pictures. It had a capacity of 650 mb / 700 mb.

6. mini-CD :- It is wide around 3 inches & can store 210 mb data.

3. Hard Disk:- Also known as magnetic disc that stores data. It is located in a drive unit. It is non-volatile storage device that contains platters & magnetic disks rotating at high speed.

It is designed to store data permanently including large storage capacity compared to primary memory.

H.D. was introduced in the year 1956 by IBM. The first PC contains a hard drive of less than 1 mb. while modern PC containing a H.D. of 1TB.

Advantages:- (\*) Low cost    (\*) faster than optical drive    (\*) Large storage.

Disadvantages:- (\*) slower than RAM.

(\*) HDD is noisy. (\*) consume more power

4. Pen Drive:- It is a small removable storage device. It is a portable storage device that you can use anywhere & connect to any computer. Also known as USB flash drive. USB Pen drive can be of three type namely USB 1.0, 2.0, 3.0

use of Pen drive:- ① data transfer ② back up  
③ updating motherboard firmware ④ Booting O/S  
⑤ store digital data.

Date - 21  
Page - 30

disadvantage:- ① not Capable to store large data.  
② data may corrupt due to virus.

## Unit - IV

Sequential circuit)— The output of a sequential logic circuit depends on the present input & past outputs. So we need a m/r element to store the past output. The outputs of the m/r elements, is given to the inputs of Combinational logic circuits. Eg:- flip flops, counter etc.

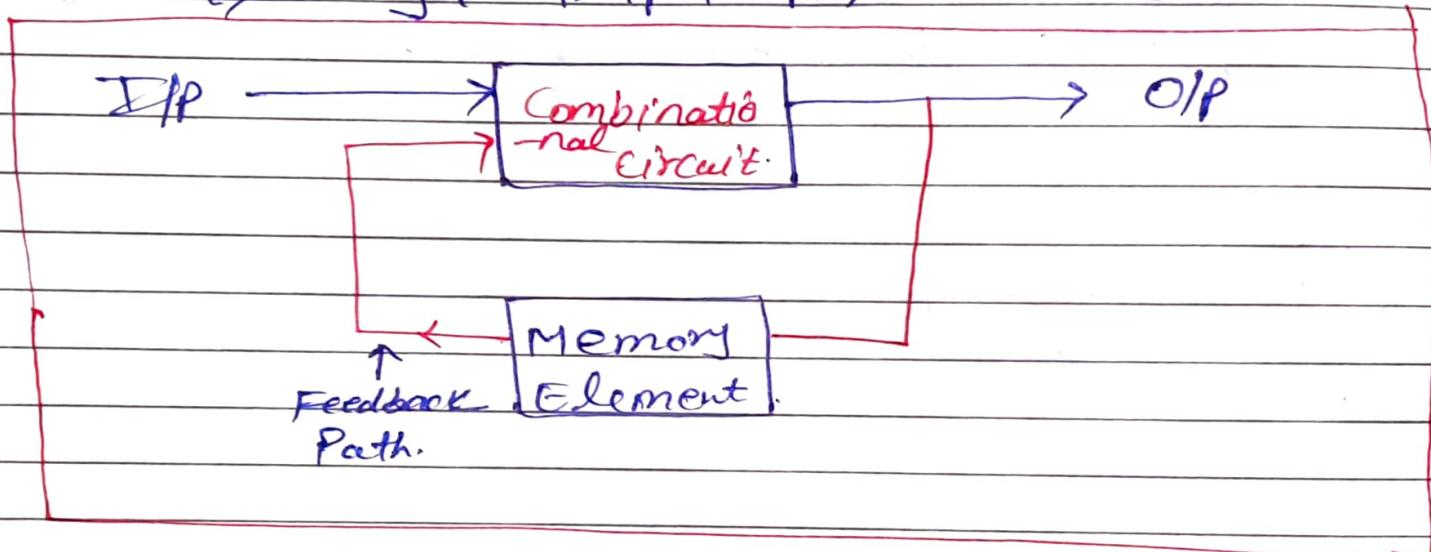


fig:- Sequential circuit

### Types of Sequential circuits:-

- 1) Asynchronous circuit do not synchronize with positive edge or negative edge of the clock signal, it means the output of the sequential circuits do not change or affect at the same time & change their states immediately when there is a change in the input signal. so, these circuits ~~do not change at different~~ are faster & independent of the internal clock pulses. These are difficult to design.

External input

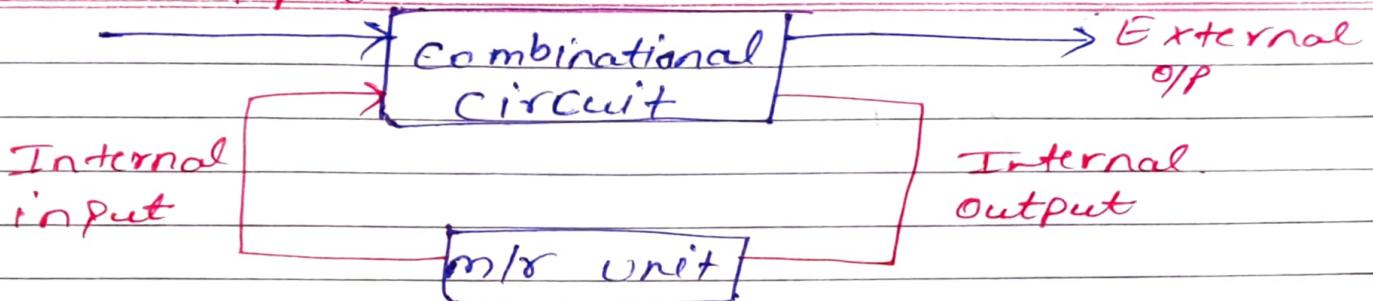


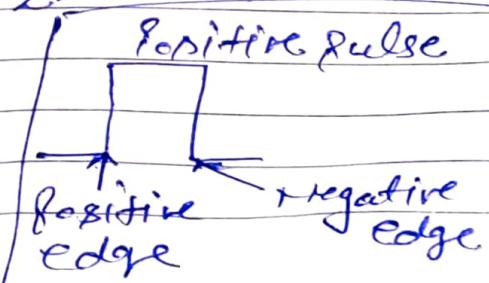
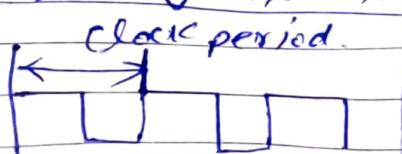
Fig:- Asynchronous Sequential circuit

~~flip flop, register, counter etc.~~

(2) Synchronous Sequential Circuits:- The output of synchronous sequential circuit changes according to the condition, at particular time before signal. Their output can be change only by giving system clock pulse.

Clock :- A clock is a special device that continuously outputs 0's & 1's.

- The time it takes the clock to change from 1 to 0 & back to 1 is called the clock period or clock cycle time.
- The clock frequency is the inverse of the clock period. The unit of measurement for frequency is the hertz.



-ive pulse.



Flip flop

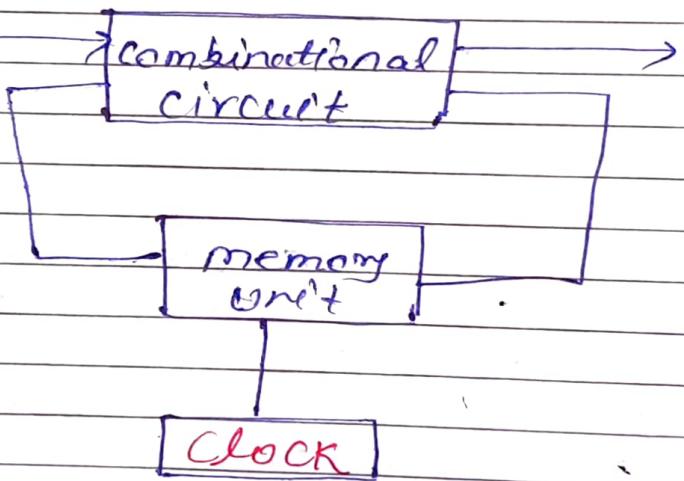


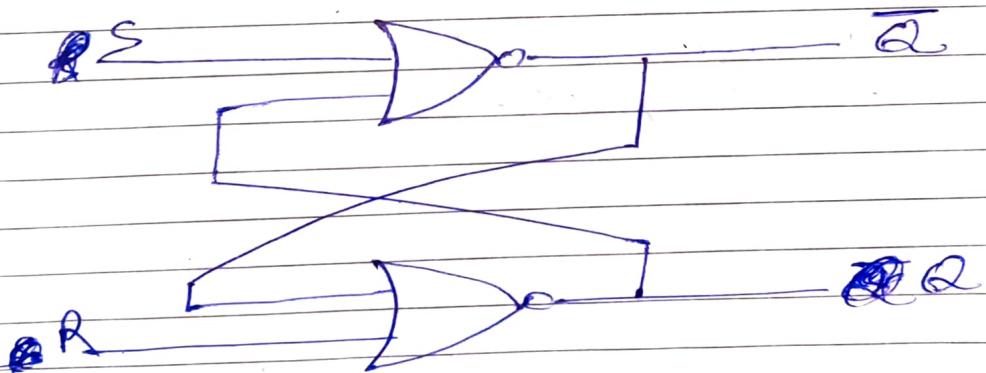
fig:- Synchronous sequential circuit.

flip flops:-

Latches :- A Latch is a basic memory element that operates with signal levels & stores 1 bit of data. Latches are said to be sensitive devices. Latches are useful for storing information & for the design of asynchronous sequential circuits.

SR Latch :- The SR latch is a circuit with two cross-coupled NOR-gate or two cross coupled NAND gates with two input labelled S (Set) & R (Reset) & with two complementary outputs Q &  $\bar{Q}$ .

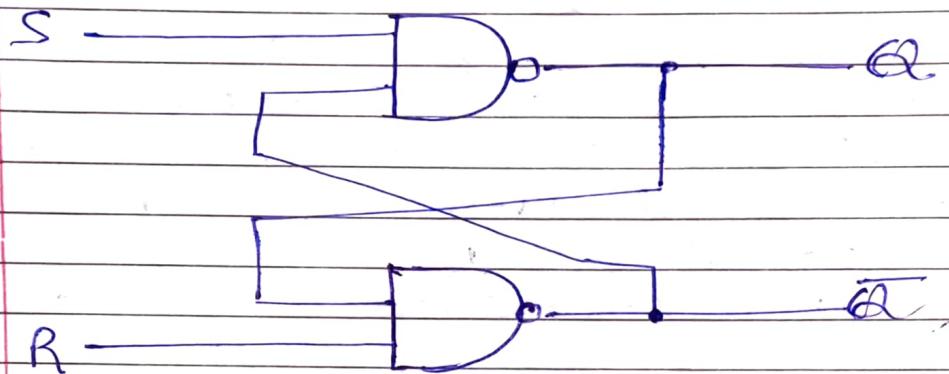
SR Latch Using NOR gates:-



S	R	Q	$\bar{Q}$
0	0	Held State	
0	1	0	1
1	0	1	0
1	1	invalid state	

I/P	I/P	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

## SR latch using NAND gate :-



S	R	Q	Q'
0	0	invalid	
0	1	1	0
1	0	0	1
1	1	Hold State	

I <sub>P</sub>	I <sub>T</sub> <sub>P</sub>	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

Latches are the basic building blocks of flip flops:-

**Flip flop :-** A circuit that has two stable states (0, 1) or (High & Low) is treated as a flip flop. These stable states are used to store binary data that can be changed by applying varying inputs. The flip flops are the fundamental building blocks of digital system.

(1) **S-R flip-flop :-** most common flip flop used in the digital system. It is also known as Set-Reset flip flop. The S & R inputs control the state of the flip flop when the clock pulse goes from Low to High. The flip flop will not change until the clock pulse is on a rising edge. When S-R (both) are simultaneously High, it is uncertain whether the output will be high or low.

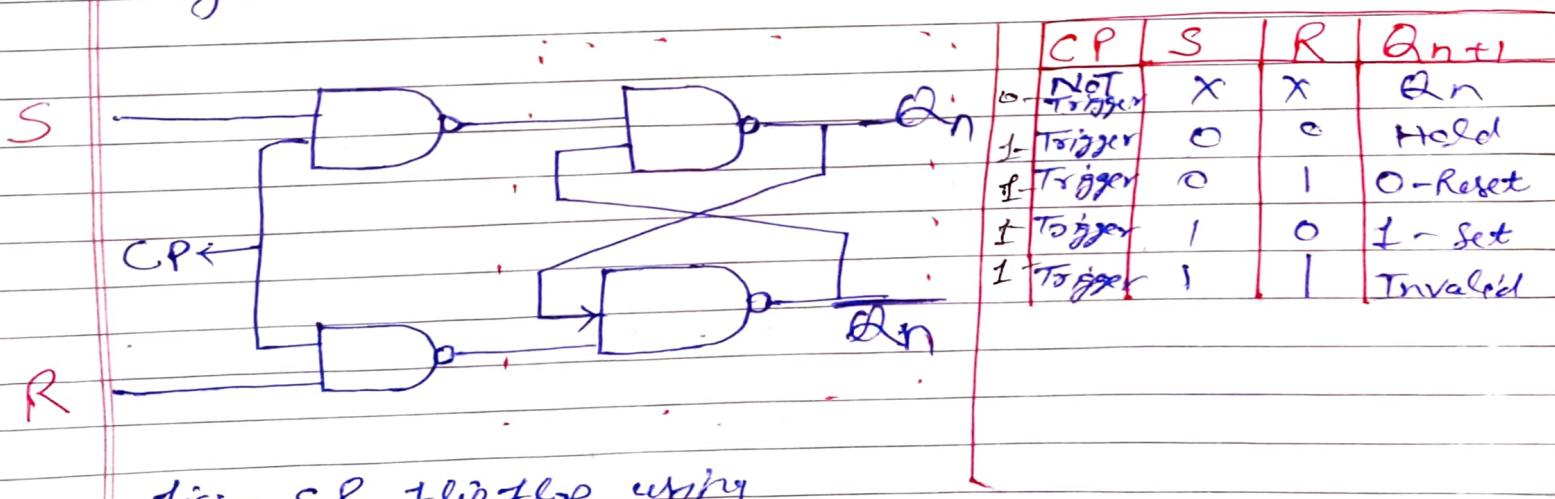
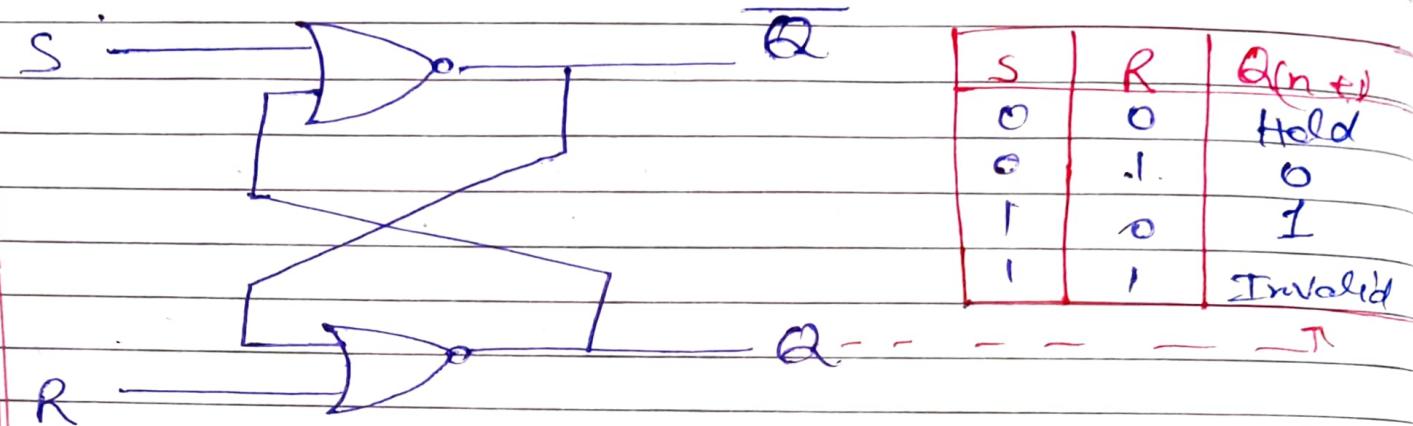


fig:- SR flip flop using  
NAND gate \*

## SR Latch using NOR Gate



S	R	$Q(n+1)$
0	0	Hold
0	1	0
1	0	1
1	1	Invalid

Table for NOR Gate

Input	I OR	H OR
0 0	0	1
0 1	1	0
1 0	1	0
1 1	1	0

Characteristic table for SR Flip-flop:-

S	R	$Q_n$	$Q_{n+1}$	
0	0	0	0	→ hold
0	0	1	1	
0	1	0	0	→ reset
0	1	1	0	
1	0	0	1	→ set
1	0	1	1	
1	1	0	X	→ don't care
1	1	1	X	

## Characteristic equation -

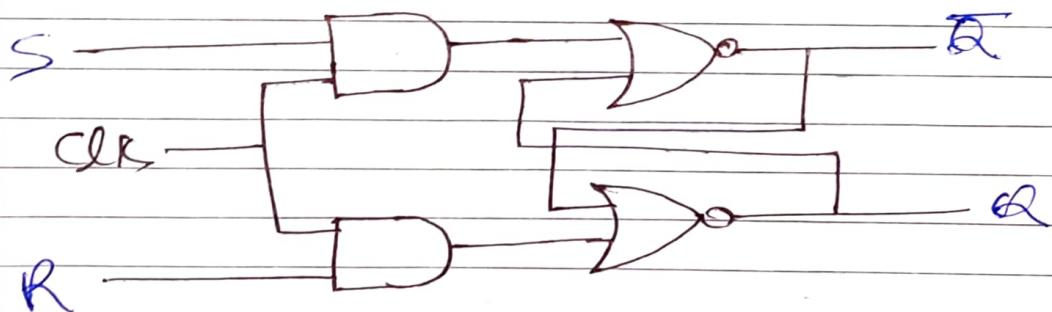
S \ R Qn	00	01	11	10
0	0	1	1	0
1	1	1	*	*

$$Q_{n+1} = S + \bar{R} Q_n$$

## Excitation Table :-

Q(n)	Q(n+1)	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

SR flip flop using Nor Gate OR NOR Latch

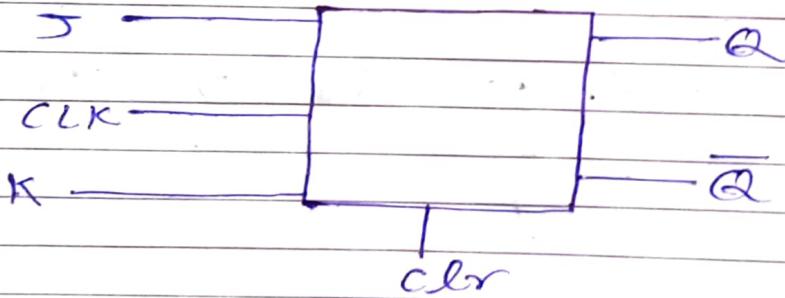


Clock	S	R	Q <sub>n+1</sub>
not trig.	X	X	Q <sub>n</sub>
Triggered	0	0	Hold
"	0	1	0
"	1	0	1
"	1	1	Invalid

## What is JK flip-flop:-

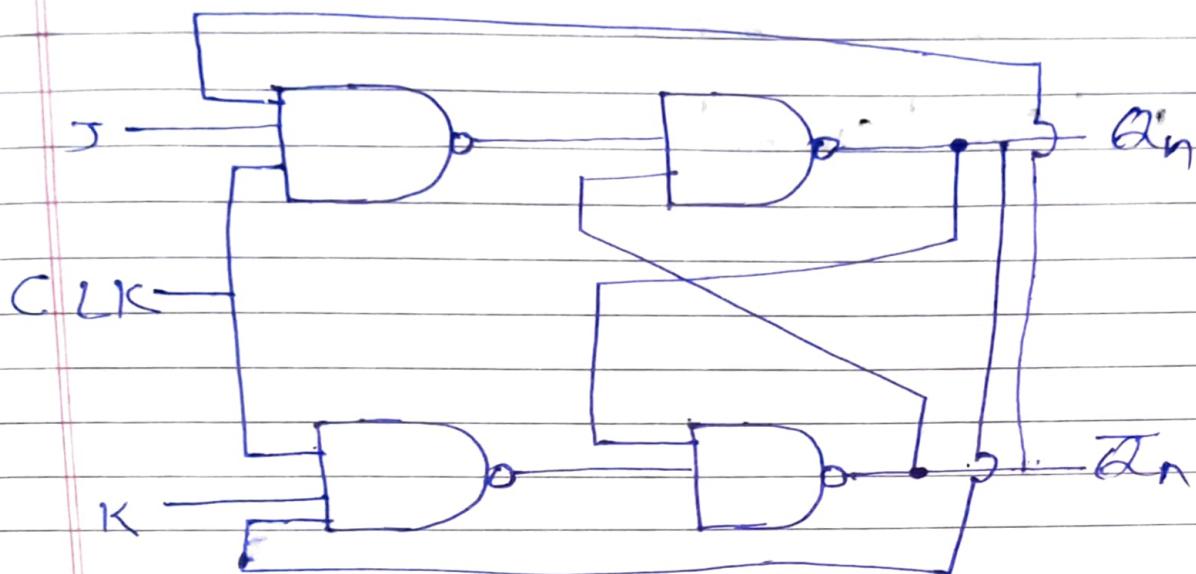
Jack Kilby flop, can be used as a basic memory element. It can store binary information & toggle functionality with a diversity of making application with it.

It is one kind of sequential logic circuit which stores binary information in bit wise manner. It consists of two inputs & two outputs. Inputs are J(set) & K(Reset) & their corresponding outputs are Q & Q'. JK flip flop block diagram as shown below:-



The JK flip flop is a refinement of S-R flip flop in which the S-R type's invalid state is defined.

S-R flip flop is a most basic flip flop. We can design further flip flops using S-R flip flops.



S R Flip flop Truth Table:-

S	R	$Q_{n+1}$
0	0	Hold
0	1	0
1	0	1
1	1	Invalid

→ This problem of S-R  
will resolve by J-K flip flop.

It will toggle i.e.  $\bar{Q}_n$

Case I  $Q_n = 1, Q_{n+1} = 0 \quad \left. \right\} \text{Toggle value.}$

Case II  $Q_n = 0, Q_{n+1} = 1 \quad \left. \right\} \text{Toggle value.}$

J	K	$Q_{n+1}$
0	0	Hold
0	1	0
1	0	1
1	1	Toggle.

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

Characteristic equation:-

$$\cancel{J} \cancel{K} \bar{Q}_n = \bar{K} \bar{Q}_n + \bar{Q}_n \bar{K} + \bar{Q}_n \bar{J}$$

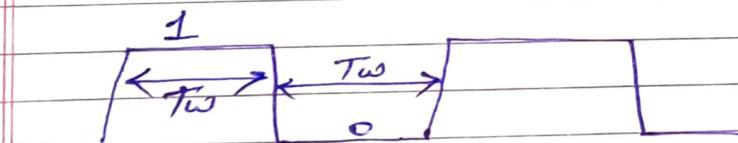
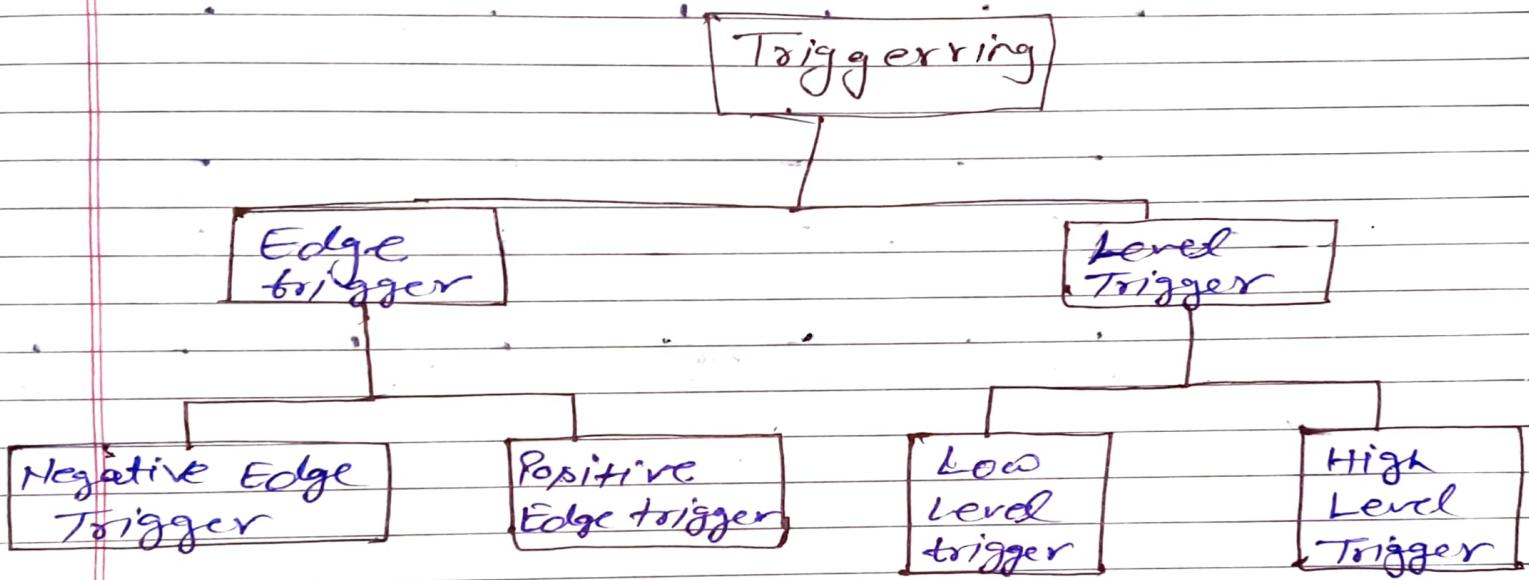
$\bar{J}$	$\bar{I}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	$\vdots$
$\bar{J}$	$\bar{I}$	$\boxed{1}$	$\boxed{1}$	$\boxed{1}$	

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

Excitation Table :-

$Q_n$	$Q_{n+1}$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

# Types of Triggering :-



$$T_W = T/2$$

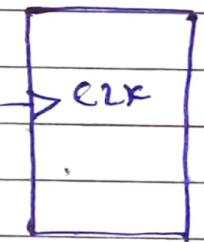
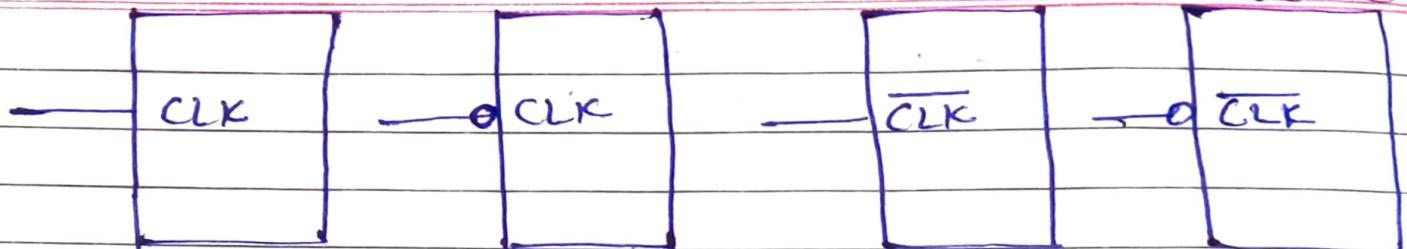
$T$  = total time for clock cycle.

+ve level

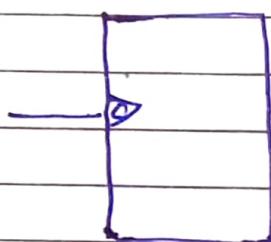
-ve level

-ve level

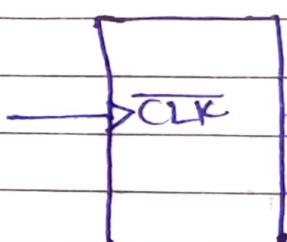
-ve level



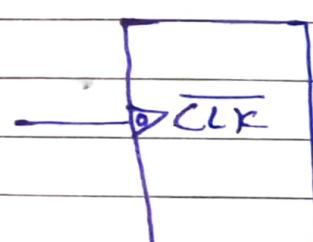
+ve Edge



-ve edge



-ve Edge



-ve Edge.

Race Around Condition :- in JK flip flop

condition 1:- Level triggered J-K flip flop

condition 2:- when  $J=K=1$  (Toggle Mode)

condition 3:-  $T_w \gg T_d$

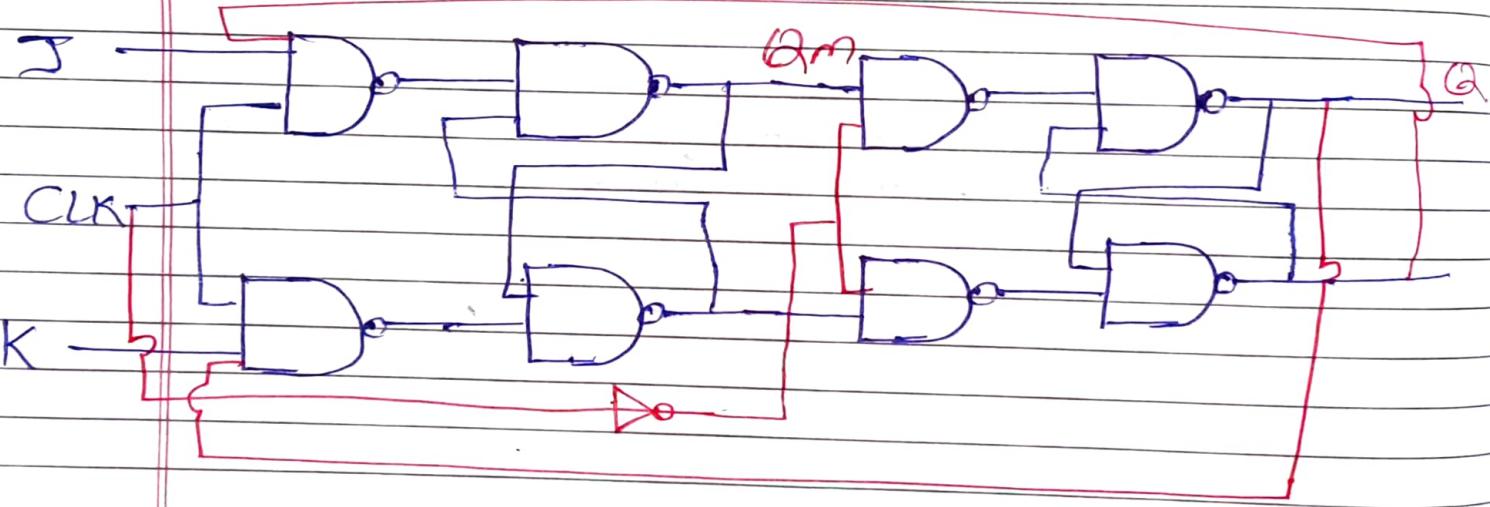
if all three conditions are true then the JK flip flop is in race around condition.



## Master slave JK flip flop -

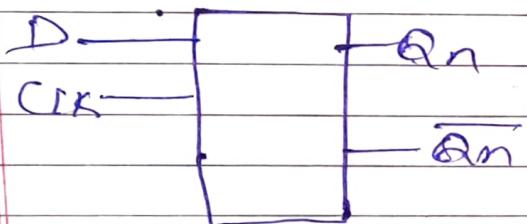
Master slave

J K flip flop is to resolve the race around condition of J K flip flop.



## (\*) D flip flop or Data flip flop:-

It is simply known as storage.  
Block Diagram :-



Truth Table:-

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

Characteristic Table:-

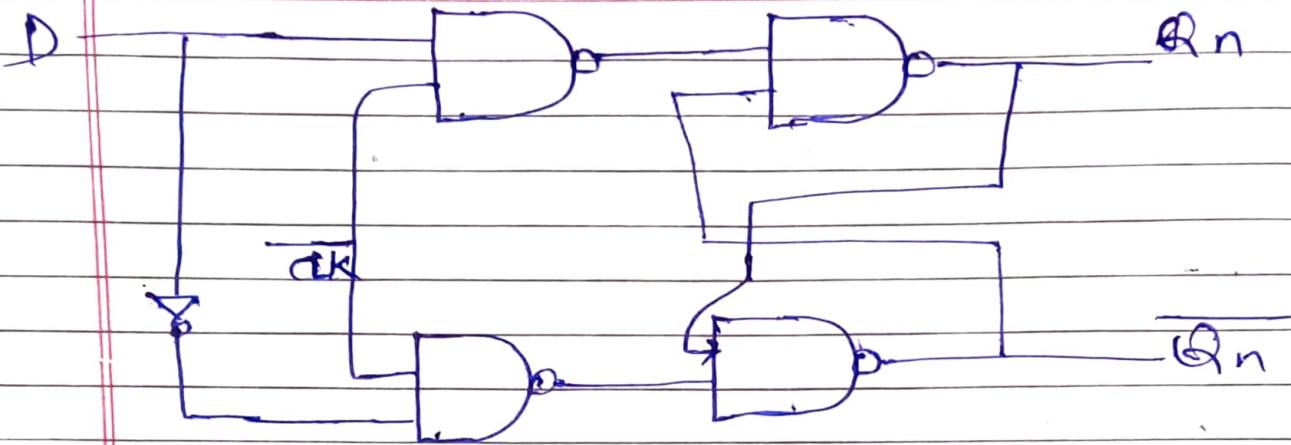
D	$Q_n$	$Q_{n+1}$
0	0	0
0	1	0
1	0	1
1	1	1

Characteristic equation:-

$$Q_{n+1} = D$$

Excitation Table:-

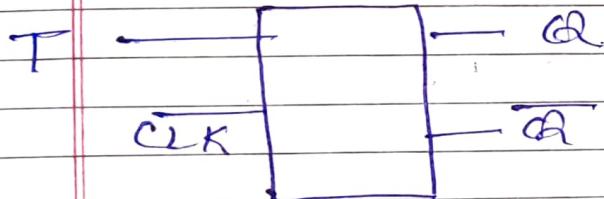
$Q_n$	$Q_{n+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1



D flip flop by SR flip flop.

(\*) T flip flop :- Also known as Toggle flip flop

• Block diagram:-



- Truth Table :-

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

(\*) Characteristic Table :-

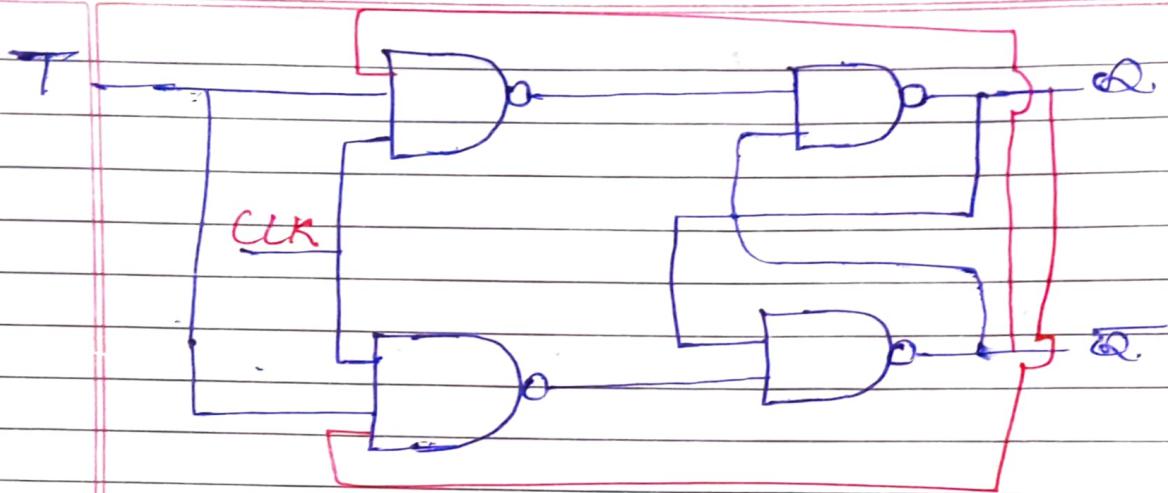
T	$Q_n$	$Q_{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic Table :-  
equation :-

$$\text{Q}_{n+1} = T \oplus Q_n$$

(\*) Excitation Table :-

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



## Q) Conversion of flip flops:-

### SR to D flip flop:-

SR  $\rightarrow$  given

D  $\rightarrow$  Required.

Use of SR - Excitation Table  
~~or~~ D - characteristic Table

### Q) D - flip flop Characteristic Table + S.R. Excitation

D	Qn	Qn+1	S	R	Qn	Qn+1	S	R
0	0	0	0	X	0	0	0	X
0	1	0	0	1	0	1	1	0
1	0	1	1	0	1	0	0	1
1	1	1	X	0	1	1	X	0

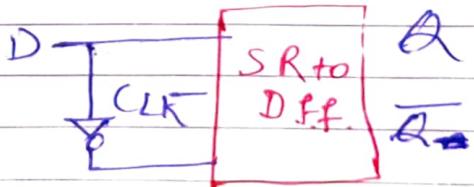
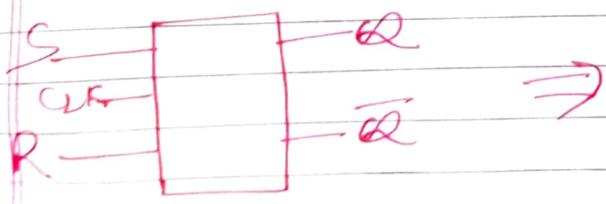
Characteristic equation according to 'S'  $\neq$  'R'

S	Qn	Qn
D		
D	X	X

$$\boxed{S = D}$$

R	Qn	Qn
D	X	D
D		

$$\boxed{R = \bar{D}}$$



Convert T to JK flip flop:-

T-given  $\rightarrow$  Excitation Table

JK-required  $\rightarrow$  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

$Q_n$	$Q_{n+1}$	<del><math>Q_{n+1}</math></del>
0	0	0
0	1	1
1	0	1
1	1	0

$\Rightarrow K \bar{Q}_n$

$\bar{J}$	$\bar{K} \bar{Q}_n$	$\bar{K} Q_n$	$K \bar{Q}_n$	$K Q_n$
J	D	D	D	D

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



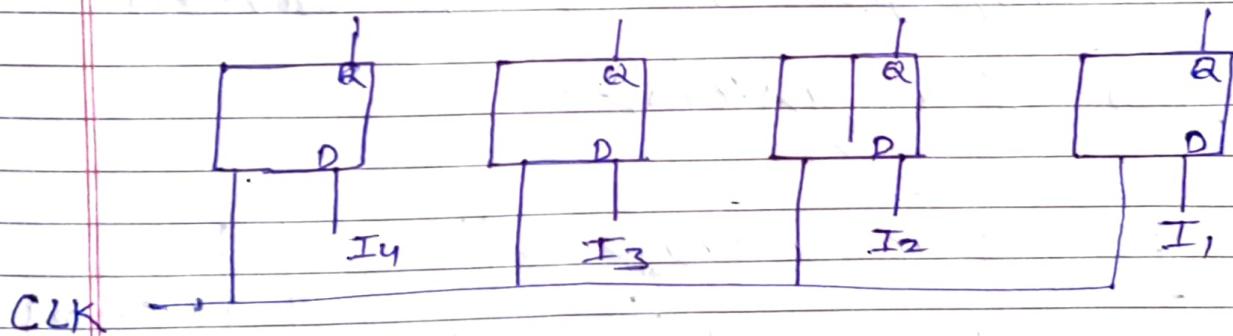
Registers:- (\*) It is a group of flip flops.

(\*) It is used to store information.

(\*) n-bit register will contain n flip flops & will store n-bit word.

In addition to flip flops, a register may have combinational gates that perform data processing tasks.

The gates control when & how new information is transferred into the register.



4-bit Register

A register can be 2-bit register, 4-bit register, 8-bit register.

There are the following operations which are performed by the registers:-

(1) Fetch:- (\*) It is used to take the instructions given by the users.

(2) To fetch the instruction stored in to the main memory.

(2) Decode:- The decode operation is used to interpret the instructions. The operation performed on the instructions is identified by the CPU.

In simple words, the decode operation is used to decode the instructions.

(3) Execute:- This operation is used to store the result produced by the CPU into the memory. After storing the result, it is displayed on the user screen.

### Types of Registers:-

PC (Program Counter)		
AR (Address Register)	CPU 16 bit	Memory 4096 words 16 bits per word
IR (Instruction Register)		
TR (Temporary Register)		DR (Data Register)
I N P R	B U T R	A.C. (Accumulator)

### MAR or Memory Address Register:-

It is a special type of register that contains the memory address of the data & instruction.

The main task of the MAR is to access instruction & data from memory in the execution phase. The MAR stores the address of memory location where the data is to be read or to be stored by the CPU.

Program Counter :- The program counter is also called an instruction address register or instruction pointer. The next memory address of the instruction, which is going to be executed after completing the execution of current instruction is contained in the program counter.

means the program counter contains the memory address of the location of the next instruction.

Accumulator Register :- CPU mostly used Accumulator Register. It is used to store the system result. All the results will be stored in the accumulator reg. whenever CPU produces some results.

MDR or Memory Data Register :-

It is a part of computer's control unit. It contains the data that we want to store in the computer storage or data fetched from the computer storage.

The MDR works as a buffer that contains anything for which the processor is ready to use it. The MDR contains the copied

data of the memory for the processor.

The data which is to be read out or written into the address location is contained in the Memory Data Register.

**Index Register:-** It is a hardware element that holds the number. The number adds to the computer instruction's address to create an effective address.

In CPU, the index register is a processor register used to modify the operand address during the running program.

**Memory Buffer Register:- MBR**

It contains the metadata of the data & instruction written in or read from M/R.

In other words, it is used to store the data upcoming data/instruction from the M/R & going to M/R.

**Data Register:-** The data register is used to temporarily store the data. This data transmits to or from a peripheral Device.

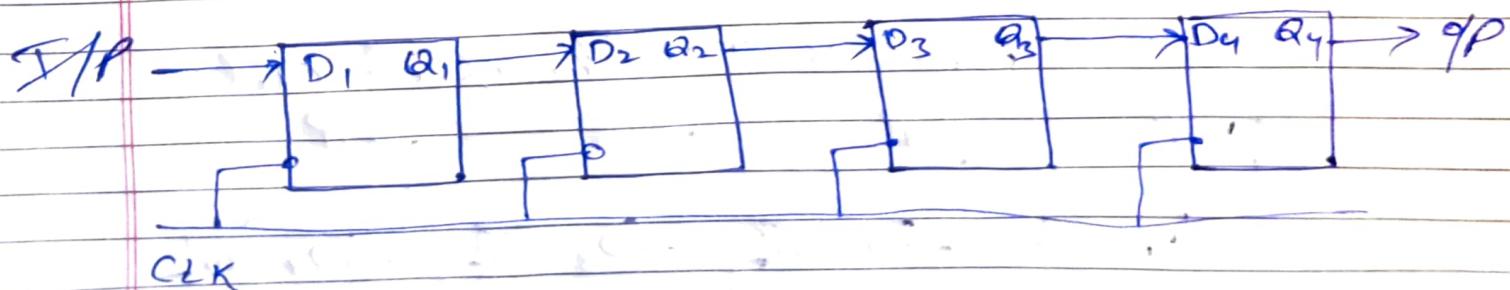
## Shift Registers:-

- (\*) Shift Registers are used to implement arithmetic operations.  
eg - Left shift & Right shift.
- (\*) Basic flip flop used in the register is D flip flop.

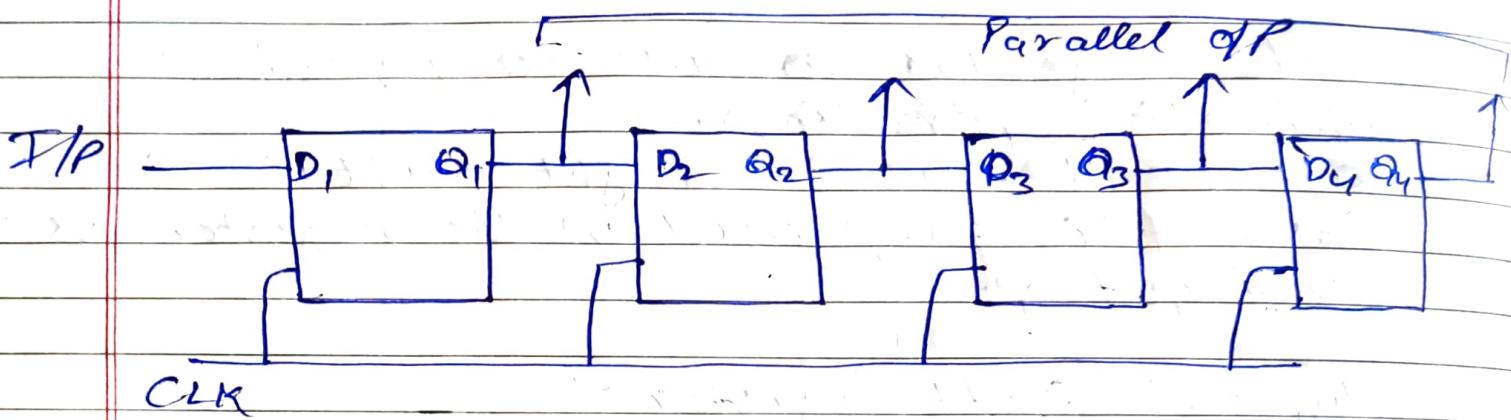
### Types of Shift Registers:-

- (1) SISO
- (2) SIPO
- (3) PISO
- (4) PIPO

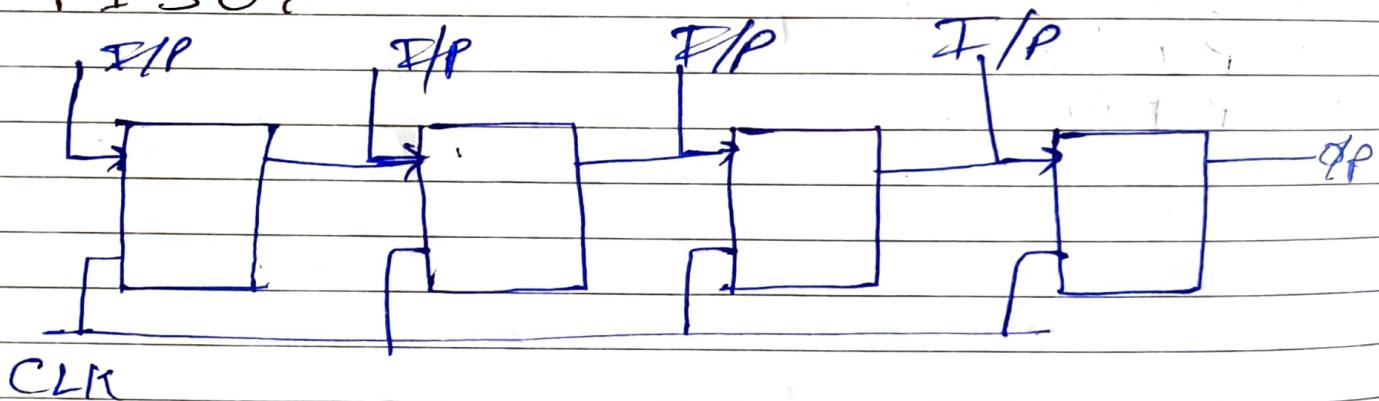
(1) SISO :- Serial Input serial Output



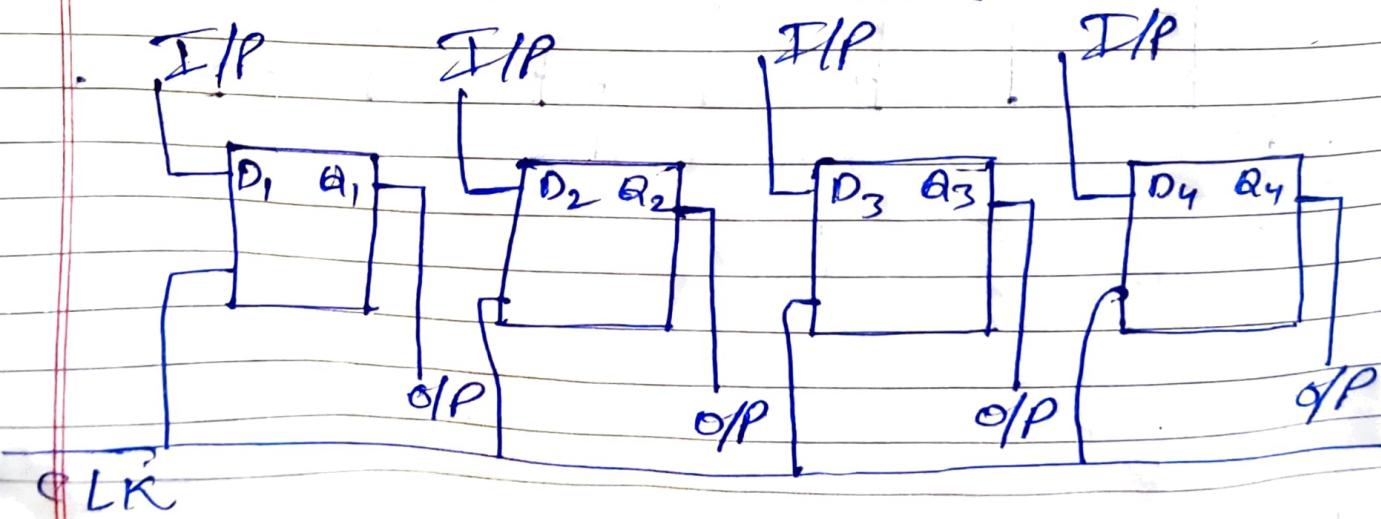
(2) SIPO :- Serial In + Parallel out



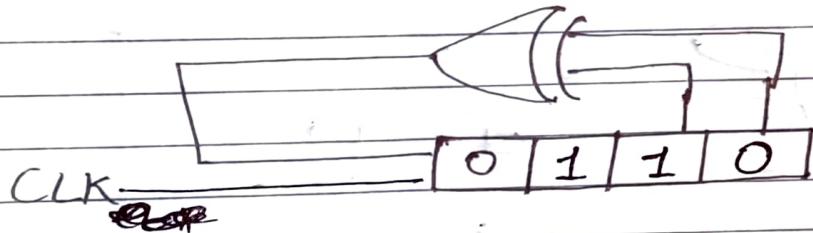
(3) PISO:-



(4) PIPO:- Parallel IN + Parallel out



MODE	Clocks needed for n-bit Shift Register		
	Loading	Reading	Total
SISO	n	n-1	2n-1
SIPO	n	0	n
PISO	1	n-1	n
PIPO	1	0	1



In a 4 bit right shift register. How many clock pulses are required to change the content of register all 1's?

Introduction to counters:- It is a sequential circuit.

- (\*) Flip flops are essential part of counter.
- (\*) used for counting of clock pulses.
- (\*) If pulses are generated by an event, then it is called Event counter.
- (\*) n No. of flip flops can count up to  $2^n$  No. of pulses.

Types of counters:-

Asynchronous counter  
or Ripple counter

Synchronous Counter  
or Parallel counters

(1) flip flops are connected in a way that QP of 1st flip flop drives the clock of the next FF.

There is no connection between QP of the 1st flip flop & the clock of the next flip flop.

(2) Flip flops are not clocked simultaneously.

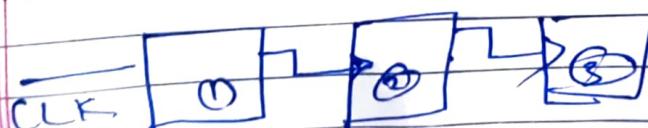
Flip flops are clocked simultaneously.

(3) Ckt is simple for more no. of states

Ckt becomes complicated as no. of states increases.

(4) Speed is slow as the clock is propagated through no. of stages

Speed is high as the clock is given at the same time.



→ we Toggle mode  
JK, T flip flop

any flip flop. Can be used

## Synchronous/Asynchronous counters

↓  
UP counter

↓  
down counter

↓  
UP/down counter

Unit - 5

RAM



SRAM

DRAM

Usage

Cache memory

main memory

Speed

very fast

fast

Cost

Costly

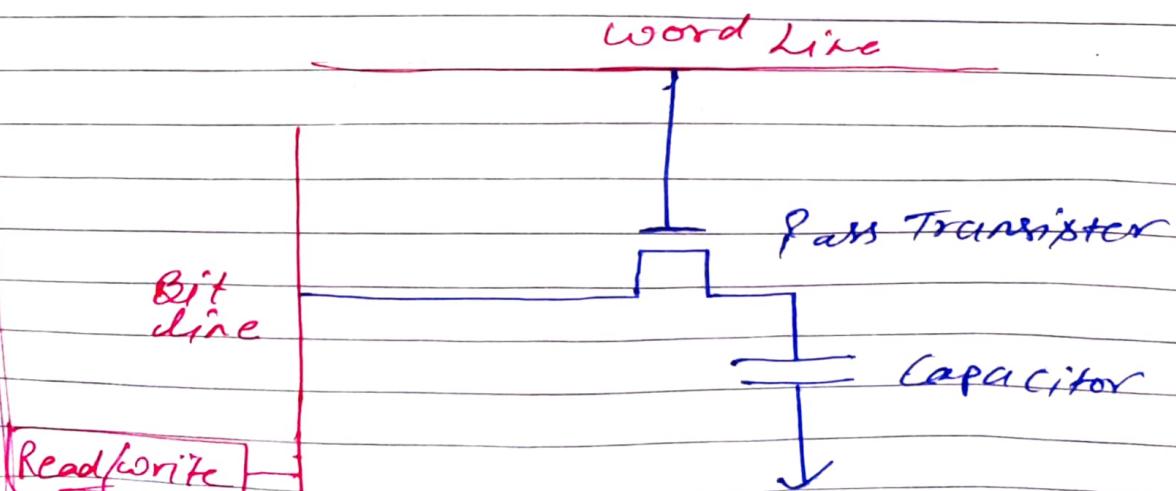
cheaper than SRAM

Density

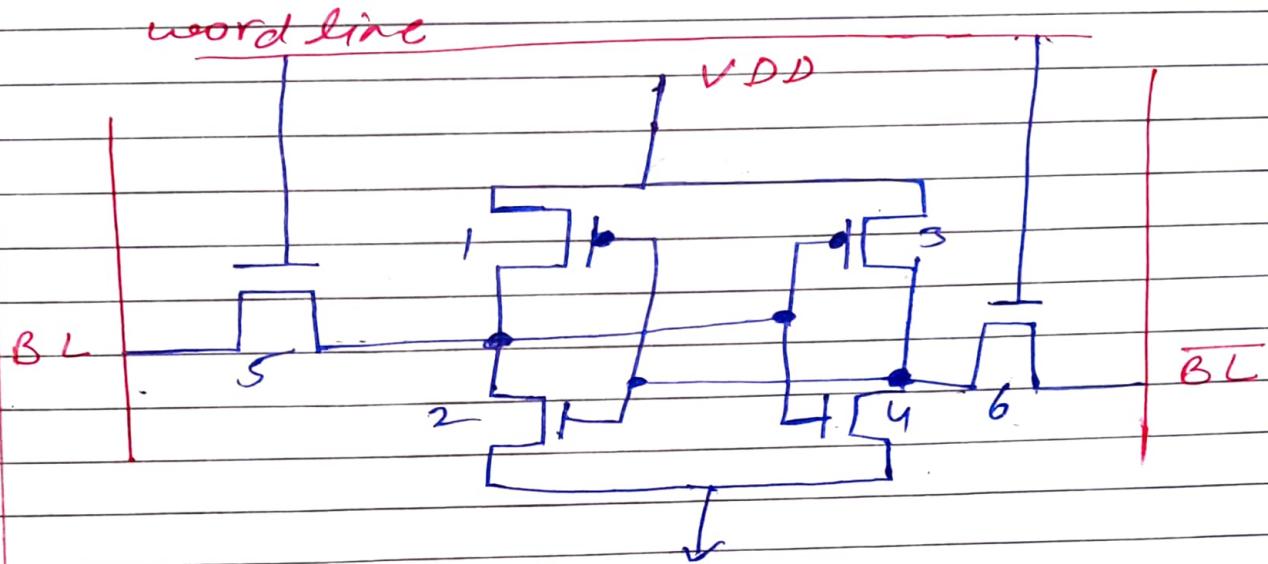
Low

High

## DRAM cell:-



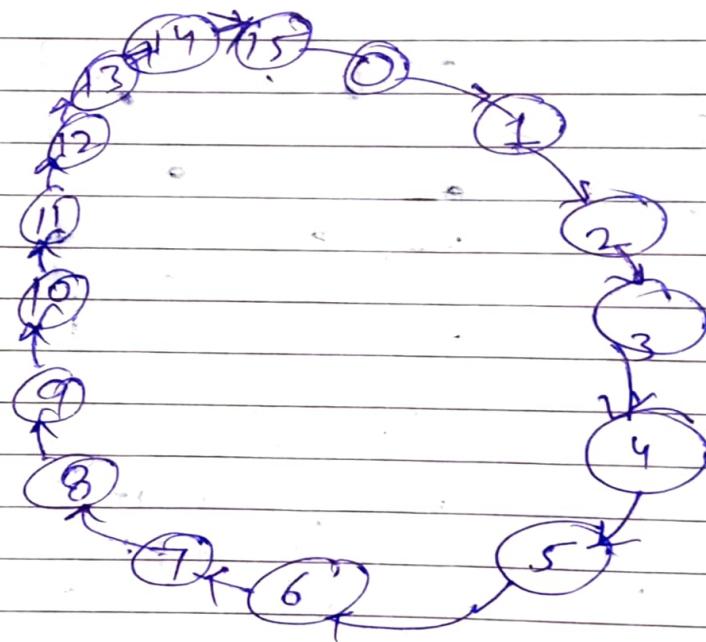
## SRAM Cell :-



- (\*) Constructed using flip flops.
- (\*) As long as power is applied it is able to retain its data.
- (\*) This reason helps avoiding refreshing.
- (\*) It has :- (1) 6 transistors
- (\*) Cache m/r org<sup>n</sup> (2) Control lines
- (\*) Virtual m/r org<sup>n</sup> (3) 1 DC voltage
- (\*) Associative m/r. (4) Bit lines (B, B̄)
- (5) Address line.

(\*) 4 bit binary synchronous counter with D FF.

Sol:- State diagram:-



Excitation table of D flip flop:-

$A$	$A_{n+1}$	D
0	0	0
0	1	1
1	0	0
1	1	1

# Excitation Table of 4 bit Counter using D flip-flop

Present State				Next State				Flip-flop Input			
$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_3(n+1)$	$Q_2(n+1)$	$Q_1(n+1)$	$Q_0(n+1)$	$D_3$	$D_2$	$D_1$	$D_0$
0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	1	0	0	1	1
0	0	1	1	0	1	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	0	0	1	1	0
0	1	1	0	0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	1	1	0	0	1
1	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	1	0	0	1	1	0	1
1	1	0	0	1	1	0	1	1	1	0	0
1	1	0	1	1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0	0

Kmap for  $D_3$

$D_3$	$Q_3 Q_2$	$Q_1 Q_0$	00	01	11	10
00	00	00				
01	01	11	11	11	11	11
10	11	11	11	11	11	11

$$D_3 = Q_3 Q_2' + Q_3 Q_1' + Q_3 Q_0' + Q_3' Q_2 Q_1 Q_0$$

$$= Q_3 (Q_2' + Q_1' + Q_0') + Q_3' (Q_2 Q_1 Q_0)$$

$$= Q_3 (Q_2 Q_1 Q_0)' + Q_3' (Q_2 Q_1 Q_0)$$

$$D_3 = Q_3 \oplus (Q_2 Q_1 Q_0)$$

Kmap for  $D_2$

		$Q_2 Q_1$	$Q_2 Q_0$	$Q_1 Q_0$	$D_2$
		00	01	11	10
$Q_3 Q_2$	00	1	1	1	1
	01	1	1	1	1
11	1	1	1	1	1
10	1	1	1	1	1

$$D_2 = Q_2 Q_1' + Q_2 Q_0' + Q_2' Q_1 Q_0$$

$$= Q_2 (Q_1' + Q_0') + Q_2' (Q_1 Q_0)$$

$$= Q_2 (Q_1 Q_0)' + Q_2' (Q_1 Q_0)$$

$$D_2 = Q_2 \oplus (Q_1 Q_0)$$

Kmap for  $D_1$  :-

		$Q_2 Q_1$	$Q_2 Q_0$	$Q_1 Q_0$	$D_1$
		00	01	11	10
$Q_3 Q_2$	00	1	1	1	1
	01	1	1	1	1
11	1	1	1	1	1
10	1	1	1	1	1

$$D_1 = Q_1' Q_0 + Q_1 Q_0'$$

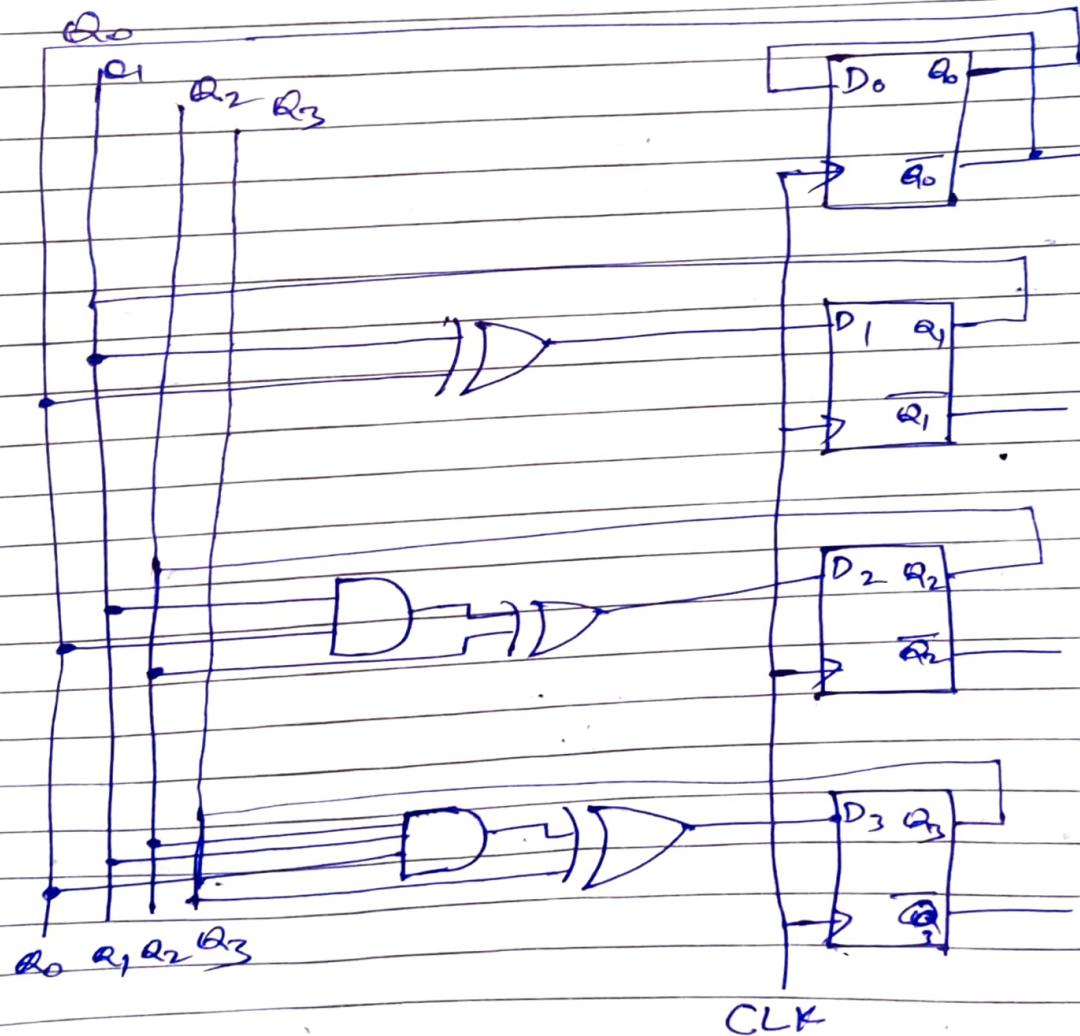
$$D_1 = Q_1 \oplus Q_0$$

K-map for  $D_0$  :-

$Q_3 Q_2 \swarrow Q_1 Q_0$

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

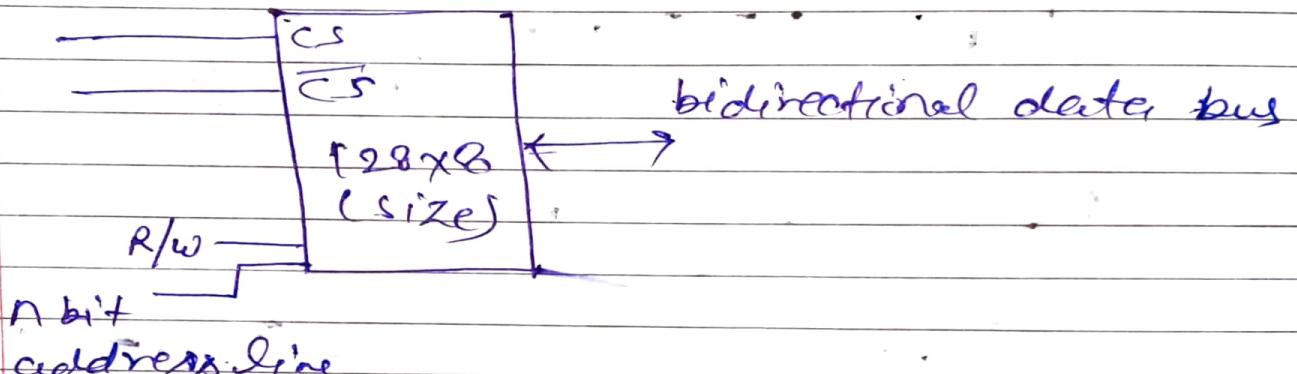
$$D_0 = \overline{Q_0}$$



Logic circuit - 4 bit binary synchronous counter using D-flip flop.

## Building large memories using chips:-

↓  
Primary memory (RAM)



$128 \rightarrow \text{words}$   
Size of 1 word = 8 bit

Size of Ram chip = no. of words  $\times$  size of each word

$128 \times 8$

↓  
→ address line

Ram organization  $\rightarrow$  making large memory by using small chips.

- (1) make  $256 \times 8$  Ram using  $128 \times 8$  Ram
  - (2)  $128 \times 8$  Ram using  $128 \times 1$  Ram
  - (3)  $256 \times 8$  using  $128 \times 1$
- 3 types  
7 Questions

- (i) No. of chips required?
- (ii) No. of Address bits required
- (iii) Decoder size
- (iv) Pictorial Representation.

d) Design a  $512 \times 8$  Ram using  $128 \times 8$  Ram Chips.

Sol:- (i) No. of chips Required

$$\frac{\text{Size of desired chip}}{\text{Size of basic Ram chip}} \Rightarrow \frac{512 \times 8}{128 \times 8} = ①$$

4 Ram chips will be required.

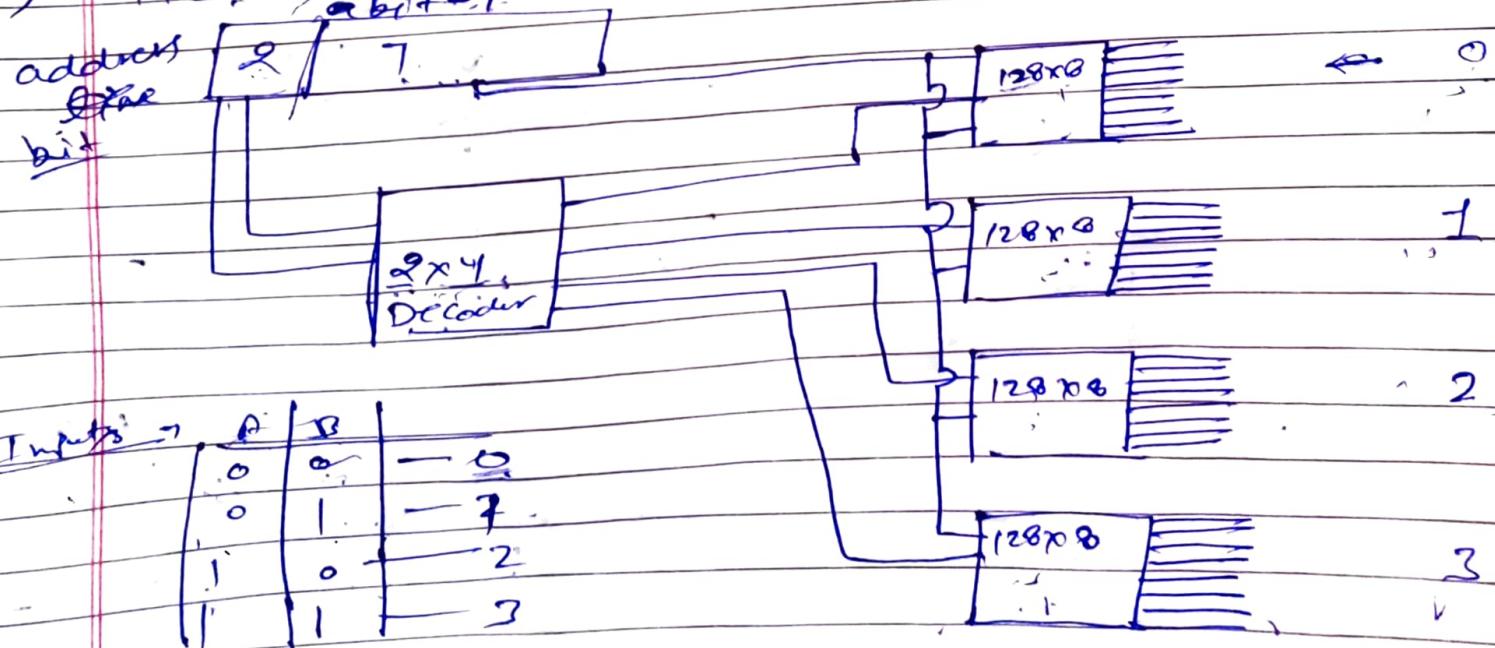
(ii) Address bits required:-

$$512 \times 8 \Rightarrow 2^9 \rightarrow 9 \text{ address bits required.}$$

(iii) Decoder size - No. of words increasing. = ④

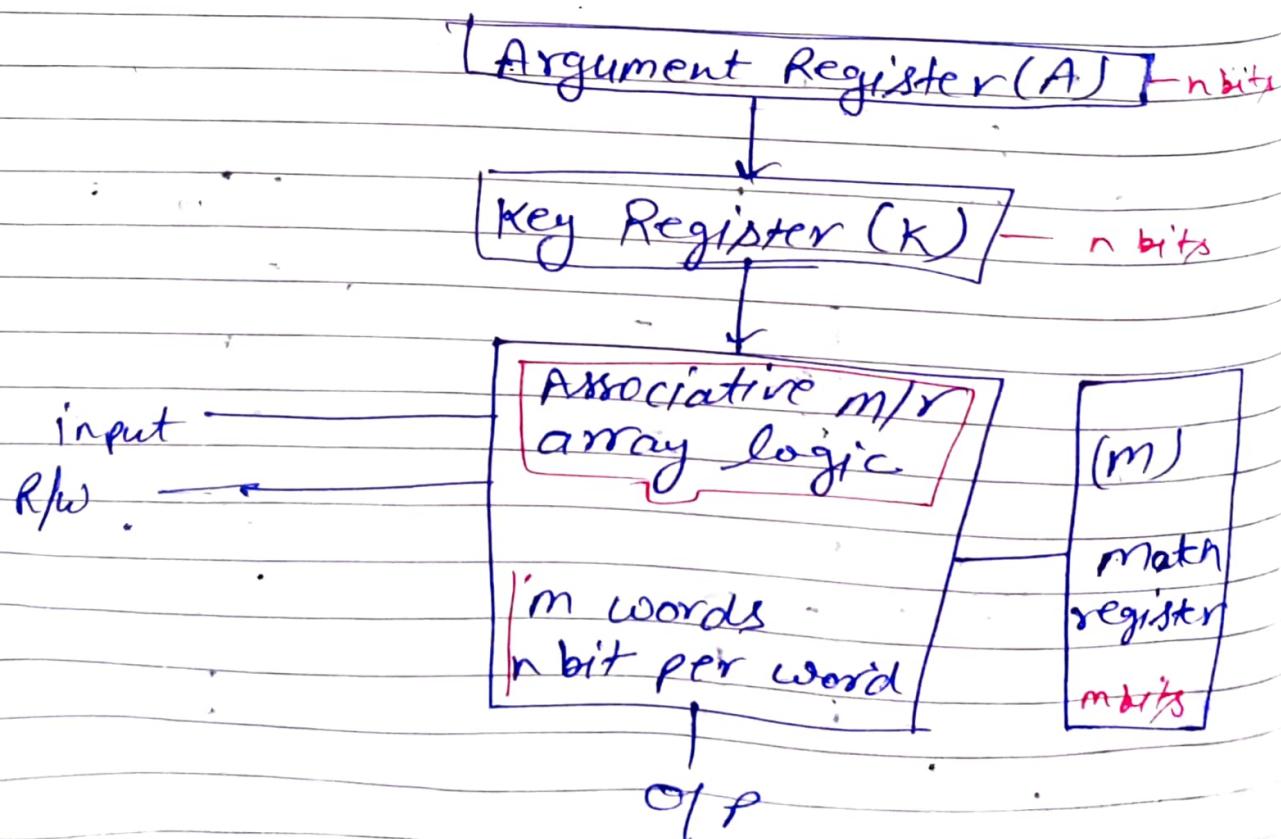
$$2^9 \rightarrow 4 \Rightarrow 2 \times 4 \text{ decoder.}$$

(iv) Pictorial Representation:-



**Associative memory:** - when data is accessed by data Content rather than data address, then the memory is called associative memory or content addressable memory (CAM).

- (\*) Data is stored at the very first empty location found in memory.
- (\*) In associative memory when data is stored at a particular location but no address is stored along with it.
- (\*) When the stored data need to searched then only the key (i.e. data or part of data) is provided.



## Applications of Associative memory:-

- (i) It can only used in mtr allocation format.
- (ii) It is widely used in the database management etc

**Advantages:-** It is used where search time needs to be less or short.

- (\*) It is suitable for parallel searches.
- (\*) It is often used to speedup database.
- (\*) It is used in page tables used by the virtual memory.

### **Disadvantages:-**

- (\*) More expensive than Ram.
- (\*) Each cell must have storage capability & logical circuits for matching its content with external argument.

## Associative memory :-

- (1) A memory unit accessed by content is called an associative memory or Content Addressable memory (CAM).
- (2) Associative memory is accessed simultaneously & in parallel on the basis of data content rather by specific address or location.
- (3) When a word is written in an associative memory, no address is given.
- (4) This memory is capable of finding an empty unused location to store word.
- (5) When a word is to be "read" from an associative memory, the content of word is specified or part of word is specified.
- (6) The memory locates all words which match the specified content & marks them for reading.
- (7) Because of its organization, the associative memory is uniquely suited to do parallel searches by data association.
- (8) An associative is more expensive than RAM because each cell must have storage capability as well as logic circuits for matching its content with an external argument.

(9) That's why, Associative memory is useful in application where search time is very critical & must be very short.

Argument :- It contain words to be searched.

Register :- It has  $n$  bits (one for each bit of word)

Key Register :- It has  $n$  bits (one for each bit of word)  
It provides a mask for choosing a particular field/key in arguments.  
or it specifies which part of the argument word needs to be compared with words in memory.

If all bits in key register are 1's, the entire word should be compared, otherwise, only the bits having 1's in their corresponding position are compared.

Associative :- It contains the word that are to be compared with the ~~memory array~~ argument word in parallel.

Memory Array :- It consists of  $m$  words with  $n$  bits per word.

Match logic(M) :- It has  $m$  bits, one bit corresponding to each word in the memory array.

After the matching process, the bits corresponding to matching words in match register are set to 1.

Reading is accomplished by sequential access in memory for those words whose match bits are set (or 1).

eg if A 1 0 1 1 1 1 0 0

K 1 1 1 0 0 0 0 0

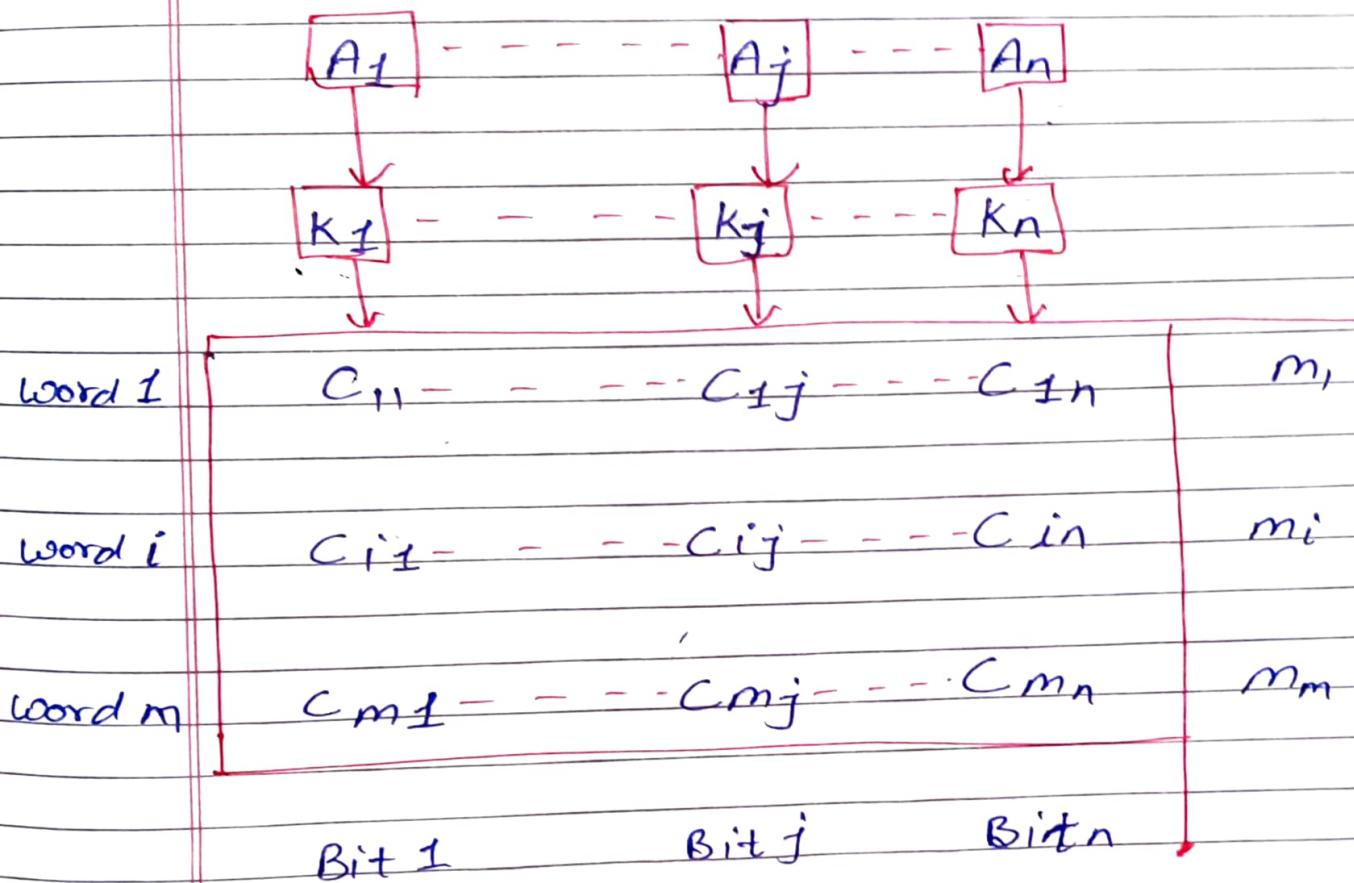
		M
Word 1	1 0 0 1 1 1 1 0 0	<input type="checkbox"/> 0
Word 2	1 0 1 0 0 0 0 0 1	<input type="checkbox"/> 1
Word 3	1 0 1 1 1 1 0 0	<input type="checkbox"/> 1
Word 4	1 1 0 0 0 1 0 1 0	<input type="checkbox"/> 0
Word 5	1 1 1 0 0 0 1 1 1	<input type="checkbox"/> 0

Associative memory of m word, n cells per word

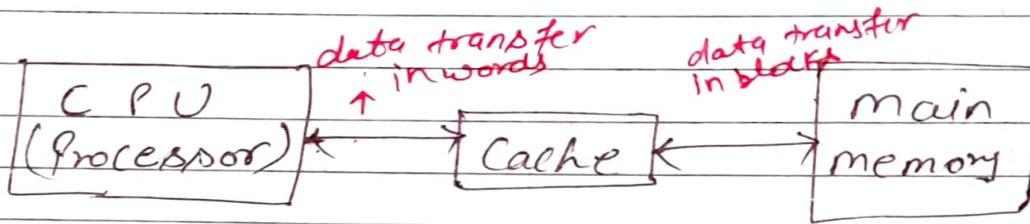
The cell in array is represented by  $c_{ij}$ : a cell for bit 'j' in word 'i'

$i$  = word number

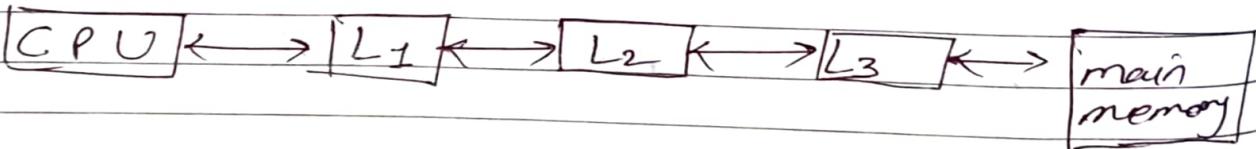
$j$  = bit position in the word.



## (\*) Cache memory :-



- Small sized fast memory.
- Placed between main memory & CPU.
- High speed volatile memory.
- Contains most frequently accessed instruction & data.
- Located inside the CPU chip or motherboard.  
 L<sub>1</sub>, L<sub>2</sub>    L<sub>3</sub>



## Working of Cache :-

- (\*) The CPU initially looks in the Cache for the data it needs.
- (\*) If the data is there, it will retrieve it & process it.
- (\*) If the data is not there, then the CPU access the system main memory & then puts a copy of the new data in Cache before processing it.
- (\*) Next time, if the CPU needs to access the same data again, it will just retrieve the data from the Cache instead of going through the whole loading process again.

**Cache Performance:-** It is measured in terms of Hit Ratio.

**Cache hit :-** If the required word is found in Cache is called hit.

$$\text{Hit Ratio} = \frac{\text{Hits}}{\text{Hits} + \text{miss}} = \frac{\text{no. of hits}}{\text{Total no. CPU references}}$$

**Cache Miss:-** If the required word is not found in Cache is called cache miss.

$$\text{Miss Ratio} = \frac{\text{miss}}{\text{Hits} + \text{miss}} = \frac{\text{no. of Miss}}{\text{Total no. of Reference}}$$

**Cache Access Time:-** Time required to access (Cache hit time) : word from the Cache.

**Miss Penalty:** (Cache miss Time Penalty)  
The Time required to fetch the required block from main memory.

Virtual memory:- It appears to be present but actually it is not. It provides illusion of a large memory. virtual memory technique allows users to use more memory for a program than the real memory of a Computer.

Virtual memory is the concept that gives the illusion to the user that they will have main memory equal to the capacity of secondary storage media (or auxillary memory).

Need of virtual memory:-