



DREAM STUDY

Smart Way of Easy Learning

Digital Electronics & Computer Organization (DECO)

www.dreamstudy.tk



dreamstudy123@gmail.com

Send us your query anytime!

BCA 2nd Sem

Digital Electronics

&

Computer Organisation

UNIT - 1Logic Gates & Circuits

* Logic gates :- The logic gates is the most basic building block of any digital system including computers.

Each one of the basic logic gates is a piece of electronic circuit that can be used to implement some basic logic expressions.

Law of boolean algebra could be used to do many manipulations with binary variables, and simplify logic expressions.

These are actually implemented in our digital system with the help of electronic circuit called logic gates.

There are three type of logic gates -

- 1- OR gate
- 2- AND gate
- 3- NOT gate

1- OR Gate :-

An OR gate perform 'OR' operation on two or more than two logic variables.

The OR operation on two independent logic variables A and B is written as-

$$Y = A + B$$

and read this expression Y equal A or B.

An OR gate is a logic circuit with two or more input and single output.

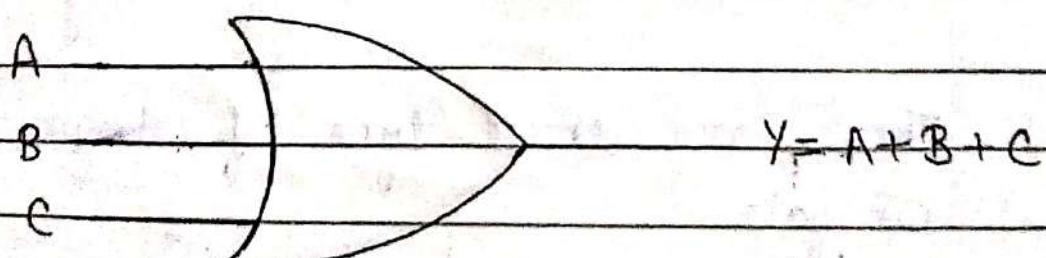
The output of an OR gate is low only when all the input are low. For all other possible input combination the output is high.

The output of an OR gate is logic '0' only when all the input are at logic '0'. For all other possible input combination the output is logic '1'.

$$Y = A + B$$

For more than two variables -

$$Y = A + B + \dots$$



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

2- AND gate :-

An AND gate is a logic circuit having two or more input and one output.

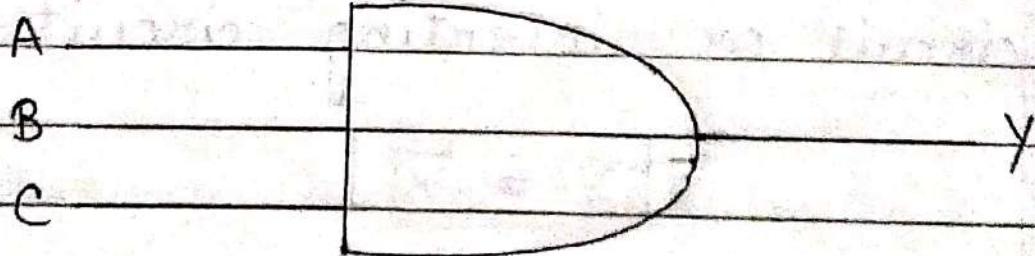
The output of AND gate is high only when all of its inputs are in high state.

In all other cases, the output is low when interrupted for a possible logic system, that means the output of AND is a logic '1' only when all of its input are in logic '1'.

In all other cases output will be logic '0'.

$$Y = A \cdot B$$

$$Y = A \cdot B \cdot C$$



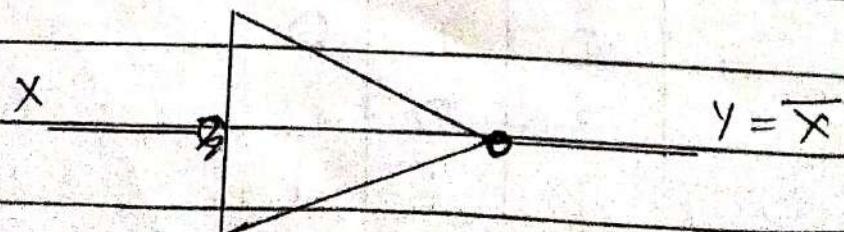
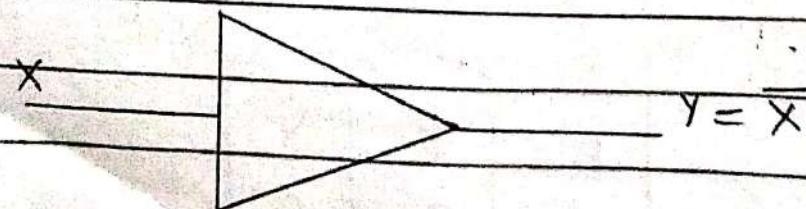
	A	B	Y
	0	0	0
	0	1	0
	1	0	0

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

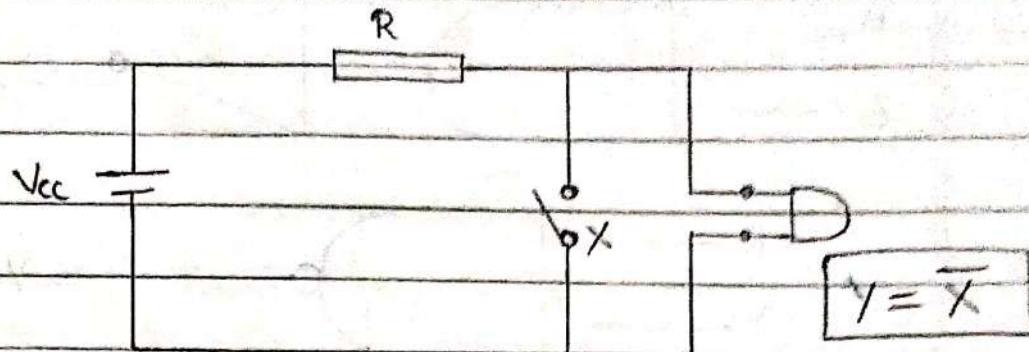
3- NOT Gate :-

NOT gate is a single input and single output logic circuit whose output is always the complement of the input, that is a low input produced a high output. It is known as complementing circuit or inverting circuit.

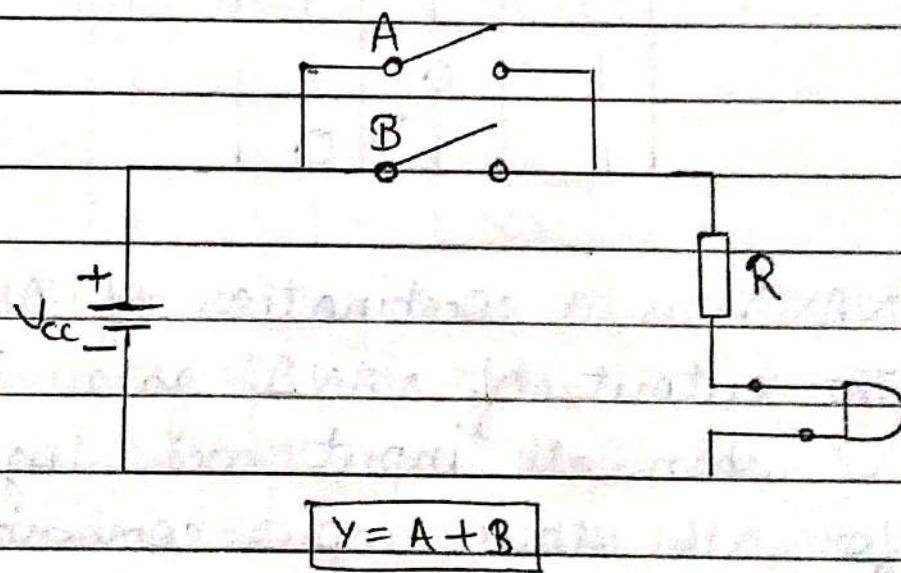
$$y = \bar{x}$$



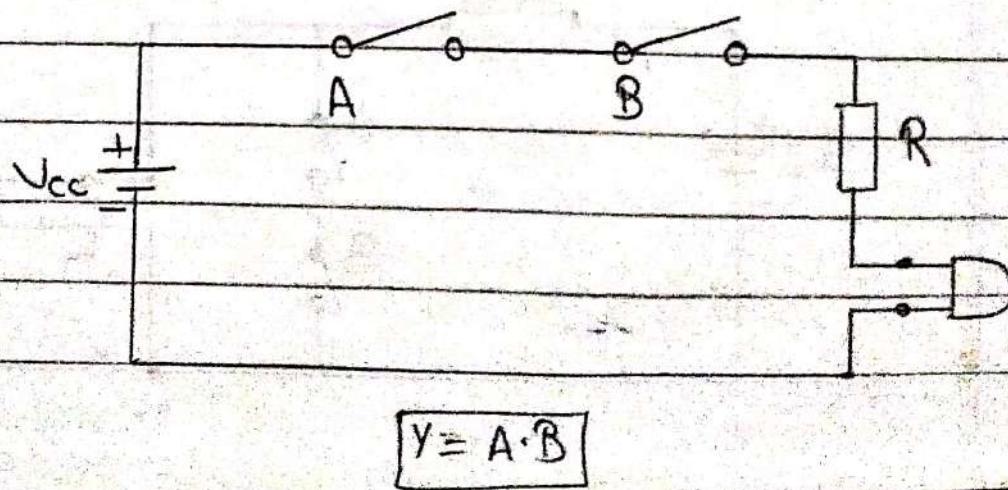
X	Y
0	1
1	0



* Circuit of OR gate :-

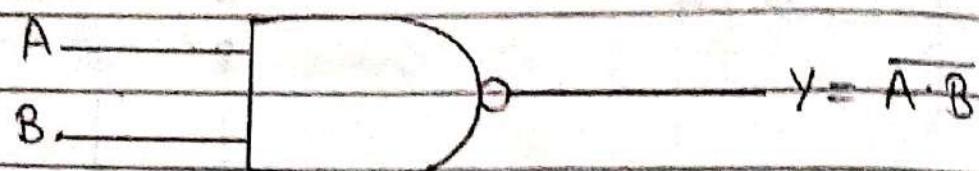
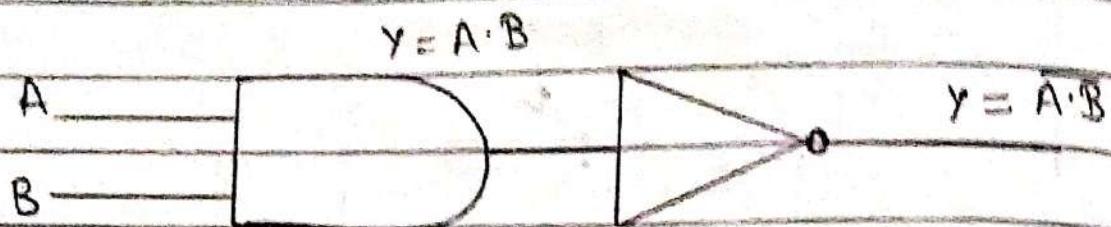


Circuit of AND gate :-



* NAND Gate :-

NAND = AND + NOT



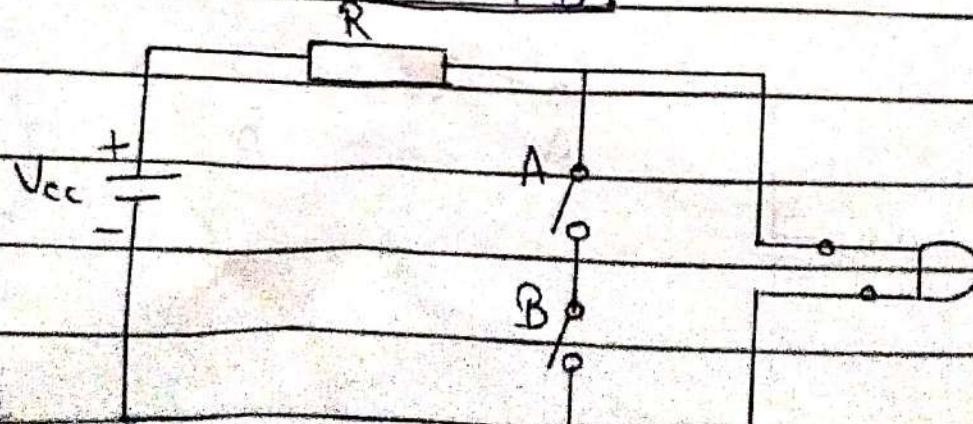
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

NAND is a combination of AND + NOT.

The output of NAND gate is a logic '0' when all input are logic '1' and for all other input combinations the output is a logic '1'.

NAND gate operation is logically expressed as

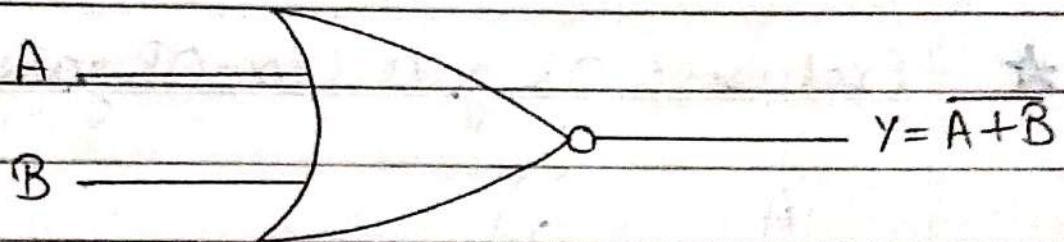
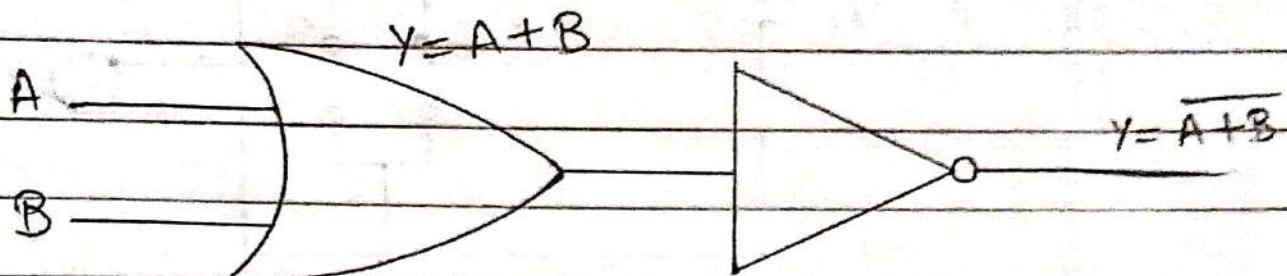
$$Y = \overline{A \cdot B}$$





* NOR gate :-

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



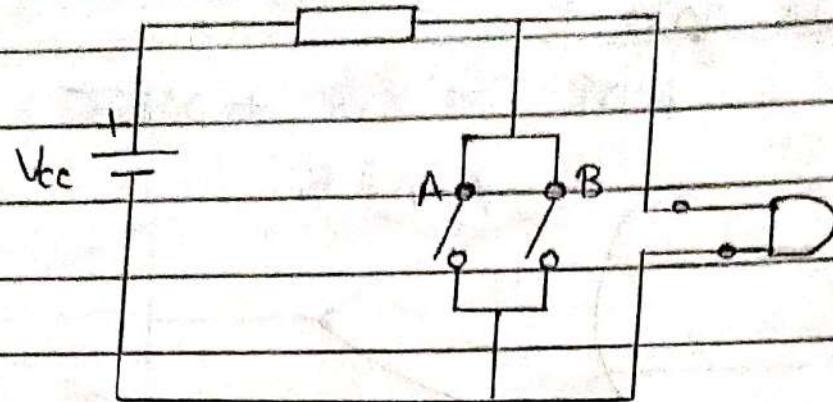
A	B	y
0	0	1
0	1	0
1	0	0
1	1	0

NOR is a combination of OR + NOT.

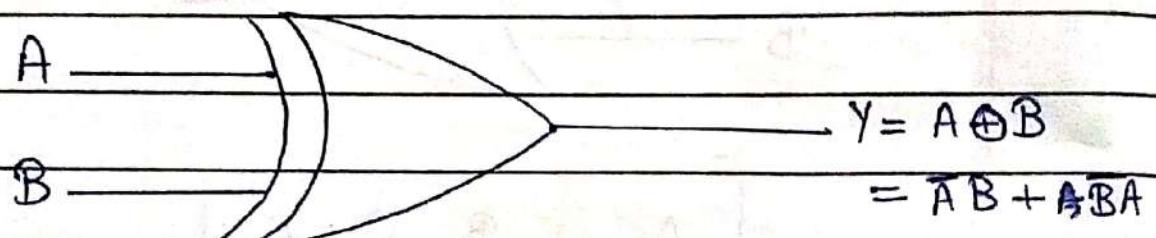
The output of NOR gate is logic '1' when all its input are logic '0' and for all other input combination the output is logic '0'.

The output of two input NOR gate is logically expressed as -

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



* Exclusive OR gate (Ex-OR gate) :-

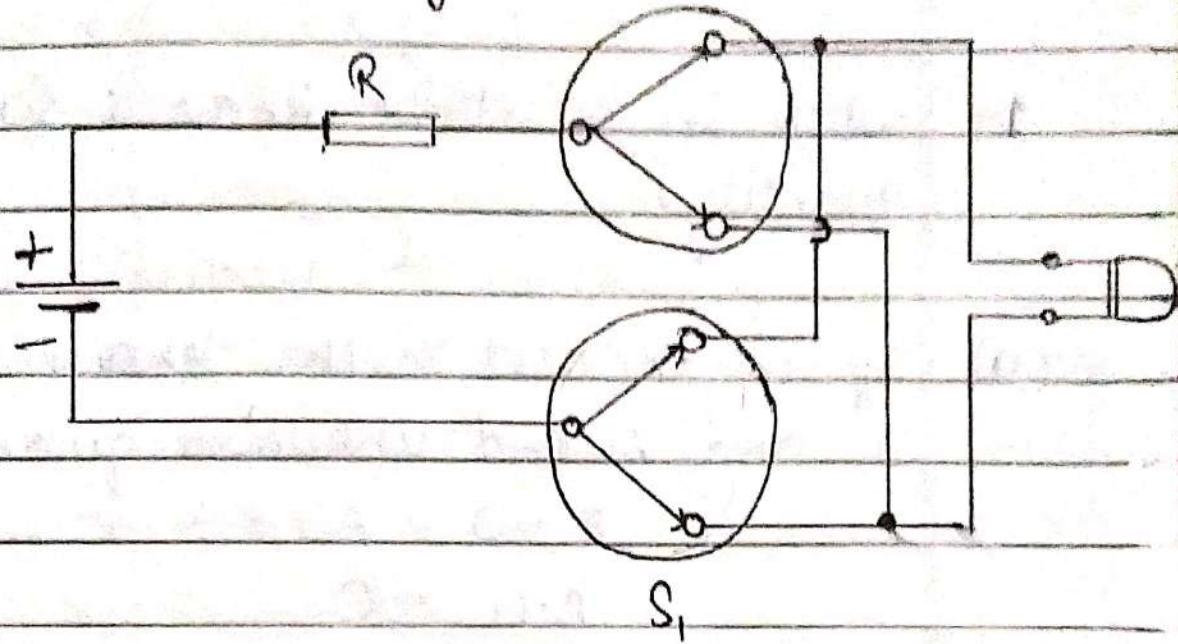


A	B	y
0	0	0
0	1	1
1	0	1
1	1	0

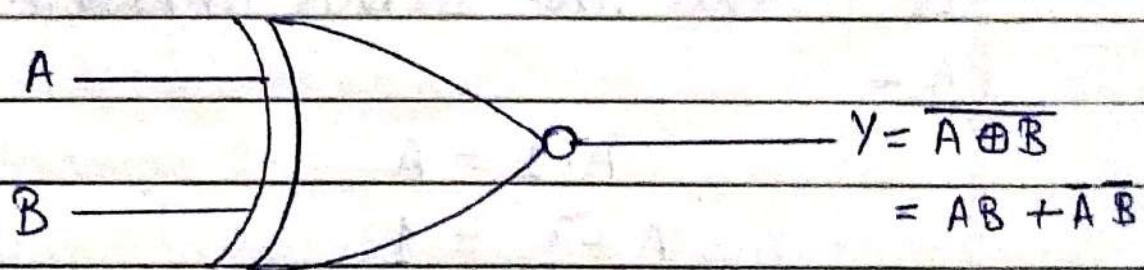
The exclusive OR gate commonly written as Ex-OR gate. It is a many input and one output gate. The output of an Ex-OR is a logic '1' when the inputs are unlike and a logic '0' when the inputs are like.



Circuit of Ex-OR gate :- S_2



* Ex-NOR gate :-



	A	B	$y = \overline{A \oplus B}$
	0	0	1
	0	1	0
	1	1	0
	1	0	1

1- When in a set we know a single variable quantity.

(a) If in the set X the known quantity is 0 and second unknown quantity A

$$A + 0 = A \quad - (1)$$

$$A \cdot 0 = 0 \quad - (2)$$

(b) Similarly in the set X the known quantity is 1 and the second unknown quantity is A -

$$A \cdot 1 = A \quad - (3)$$

$$A + 1 = 1 \quad - (4)$$

2- Commutative Law -

$$A + B = B + A \quad - (5)$$

$$A \cdot B = B \cdot A \quad - (6)$$

3- Distributive Law -

$$A \cdot (B + C) = (A \cdot B) + (A \cdot C) \quad - (7)$$

$$A + (B \cdot C) = (A + B) \cdot (A + C) \quad - (8)$$

4- Complementary Law -

$$A + \bar{A} = 1 \quad - (9)$$

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

5- Associative Law -

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

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

* Duality Theorem :- According to this theorem a Boolean expression can be changed in the other form of Boolean expression easily.

- (1) (+) changed in (.)
- (2) (.) changed in (+)
- (3) (1) changed in (0)

For example :-

Boolean expression

$$A + 1 = 1$$

$$A + A = A$$

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

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

Dual expression

$$A \cdot 0 = 0$$

$$A \cdot A = A$$

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

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

DeMorgan's Theorem :-

$$1) \overline{A+B} = \bar{A} \cdot \bar{B}$$

$$2) \overline{AB} = \bar{A} + \bar{B}$$

Q.1 $A + AB = A$

$$\text{L.H.S } A+AB = A(1+B)$$

$$= A \cdot 1$$

$$= A \quad \therefore \underline{\text{R.H.S.}}$$

$$\text{Q3} \quad A(A+B) = A$$

$$\text{L.H.S. } A(A+B) = A \cdot A + A \cdot B$$

$$= A + A \cdot B$$

$$= A(1+B)$$

$$= A \cdot 1$$

$$= A \quad = \text{R.H.S.}$$

$$\text{Q3} \quad A + \bar{A}B = A + B$$

$$\text{L.H.S. } A + \bar{A}B = A + AB + \bar{A}B$$

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

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

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

$$= (A+B) \cdot 1$$

$$= A + B \quad = \text{R.H.S.}$$

$$\text{Q4} \quad A \oplus B = \bar{A} \oplus \bar{B}$$

$$\text{L.H.S. } A \oplus B =$$

* Ckt designing techniques :- Before designing the circuit we need to simplify the circuit for efficient design. So for simplify the circuit, first we simplify the boolean expression from which the circuit will form and after the simplification there are basically two techniques which are used for simplify the boolean expression.

1- Canonical form SOP [Sum of Product]

POS [Product of Sum] π

2- K-map

Canonical form - There are a finite number of boolean function of an input variable. Yet an infinity number of possible logic expressions can be constructed with those an input values. Clearly there are no an infinity number of logic expressions that are equivalent.

Canonical form having a two type -

- (i) SOP [Sum of Product]
- (ii) POS [Product of Sum]

SOP [Sum of Product] - This form is also called DCF (Disjunction canonical form). In this form the function is a sum of

Q1: Find the sum of product expression for the function $f(x,y,z) = (x+y)\bar{z}$

$$\begin{aligned}
 f(x,y,z) &= (x+y)\bar{z} \\
 &= x\bar{z} + y\bar{z} \\
 &= x(y+\bar{y})\bar{z} + (x+\bar{x})y\bar{z} \\
 &= xy\bar{z} + x\bar{y}\bar{z} + xy\bar{z} + \bar{x}y\bar{z} \\
 &= xy\bar{z} + x\bar{y}\bar{z} + \bar{x}y\bar{z}
 \end{aligned}$$

Q2: Express the following boolean function in their SOP form and the function is -

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

POS [Product of Sum] - This form is also called CCF (Conjunction canonical form).

Product of sum is derived by considering the combination for which $f=0$. Each term is a sum of all the variables. A variable appear in uncomplemented form.

Q1: Express the boolean function $f = xy + \bar{x}z$ in the product of sum.

K-map :-

Q1. Simplify the expression $f = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + A\bar{B}\bar{C}$

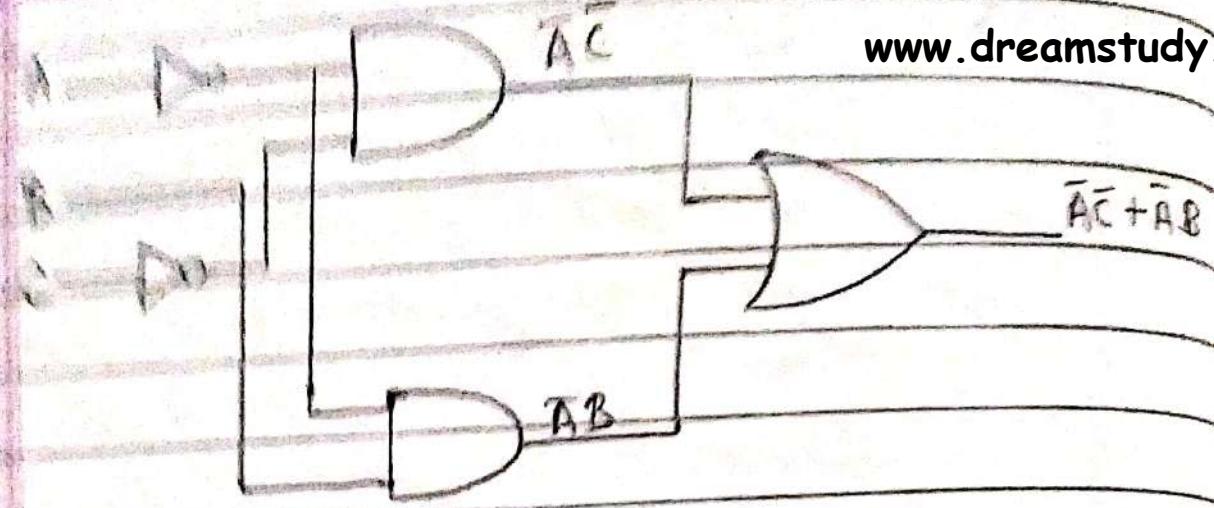
	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
\bar{C}	1	1	1	
C				

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

Q2. $f = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C}$

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
\bar{C}	1	1	1	
C				

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

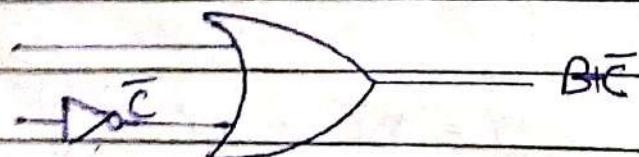


Simplify the following expression -

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

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
\bar{C}	1	1	1	1
C	1	1	1	1

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

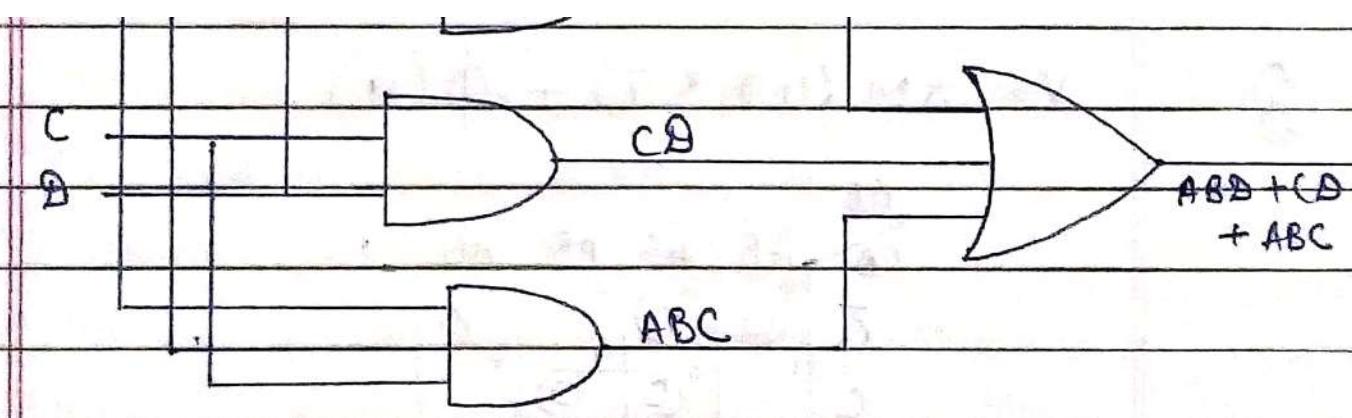


Q4.

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

	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
\bar{C}	1	1	1	1
C	1	1	1	1





Q6. $Y = \Sigma_m(4, 12, 13, 15)$

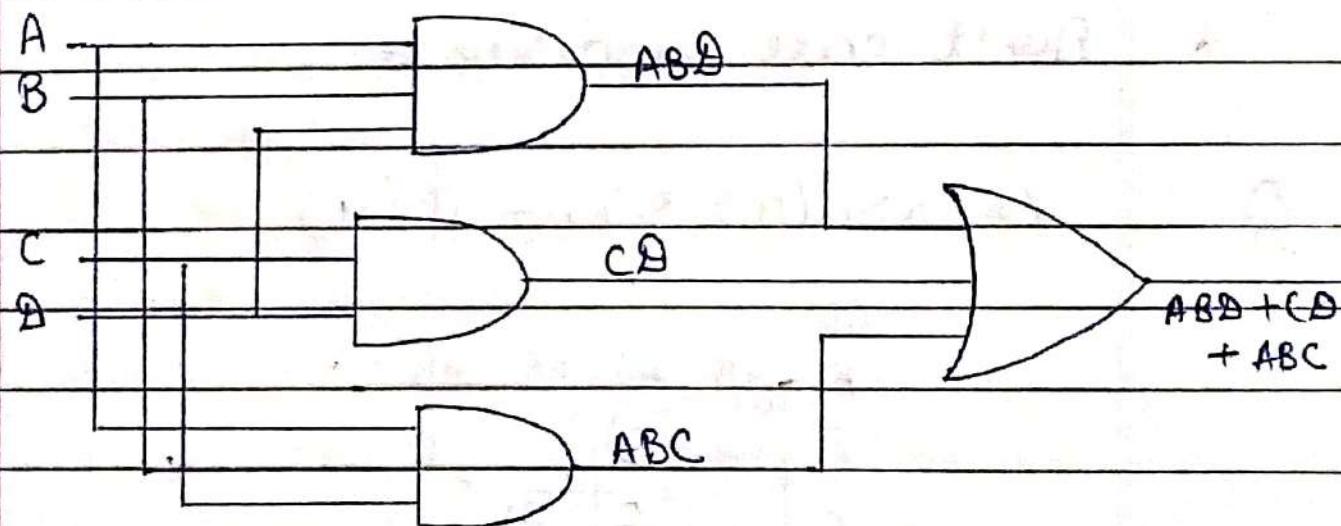
$\bar{A}B$	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{A}$	0	4	12	8
$\bar{C}A$	1	5	13	9

Q5.

$$f(ABCD) = \Sigma (3, 7, 11, 13, 14, 15)$$

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

$$Y = CD + ABD + ABC$$



Q7.

$$Y = \pi M (4, 5, 10, 12)$$

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

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

* Don't care condition -

Q8.

$$Y = \pi M (0, 2, 3, 7) + \phi(4)$$

$\bar{C}D$	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	0	2	6	4 X
$\bar{C}D$	1	3	7	5

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

UNIT - 2

Page
Page

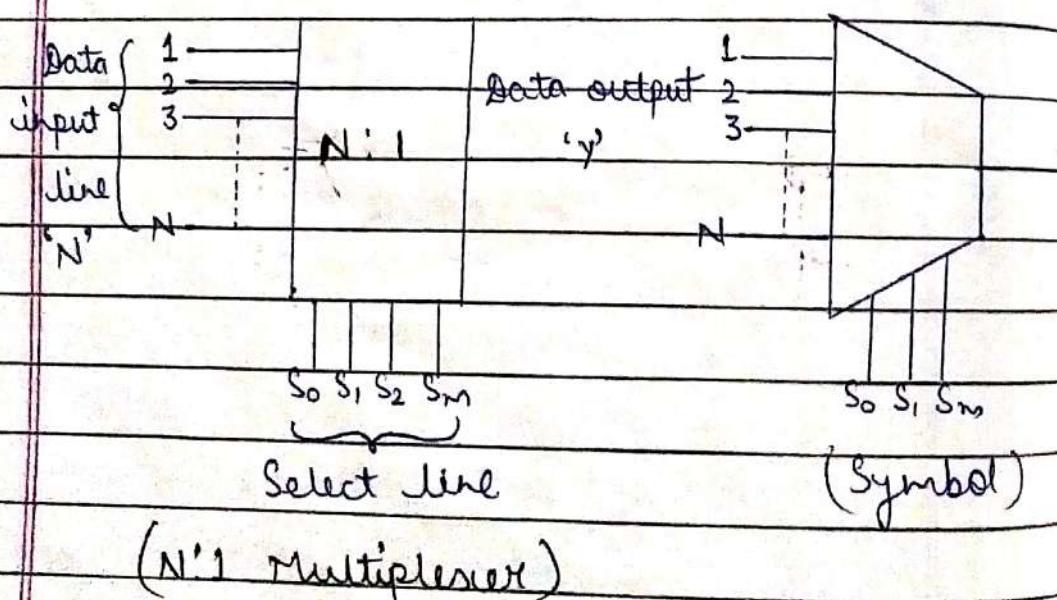
* Multiplexer :-

It is a type of combination circuit that selects binary information from one of the many input channels and transmit it to single output line.

It is also consider as many input single output circuit means many to one.

Many is used for various input and one is used for single output.

The multiplexer is used in telecommunication as it combines several input information signal into one output signal, which carry several communication channels by means of some multiplex techniques.



Here $N:1$ denotes as N stands for input

and one denotes the data output. The selection lines are denoted by $S_0, S_1, S_2, \dots, S_m$. The relation of select line with the input is given as $N = 2^m$ where $N = \text{No. of inputs}$ and $M = \text{select line}$, where N is the input and M is select line.

If $N=4$ then the value of M is $m=2$. That means two select lines are required to four inputs.

* 4:1 Multiplexer :-

It is defined as the multiplexer consisting of 4 input channels and information of one of the output line. According to the select input combination.

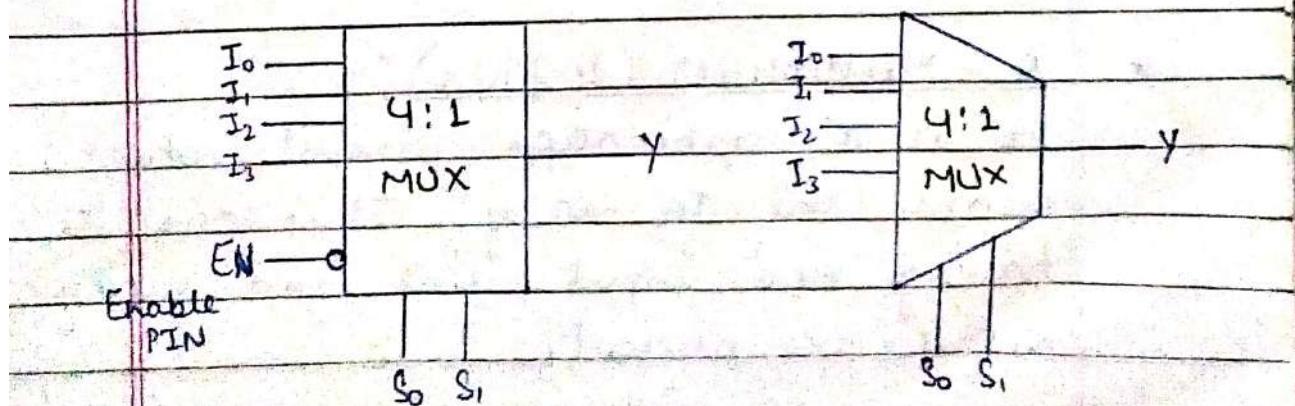
For 4:1 Multiplexer (MUX) the select lines required is

$$N = 2^m$$

$$4 = 2^m$$

$$M = 2$$

There are two select lines.



- 1- Input lines denoted by I_0, I_1, I_2 and I_3 ,
- 2- Select line are denoted by S_0, S_1 .
- 3- EN known as enable PIN slope. PIN is used for power control of operation or Multiplexer. It is a "active low" PIN means it work when it is low. (Because bubble is present which work as NOT function).

The data input I_3 is selected when $S_0 = 0$ and $S_1 = 1$ and output is given as -

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

0 → 0 0 0

1 → 0 0 1

2 → 0 1 0

3 → 0 1 1

4 → 1 0 0

5 → 1 0 1

6 → 1 1 0

7 → 1 1 1

Expression -

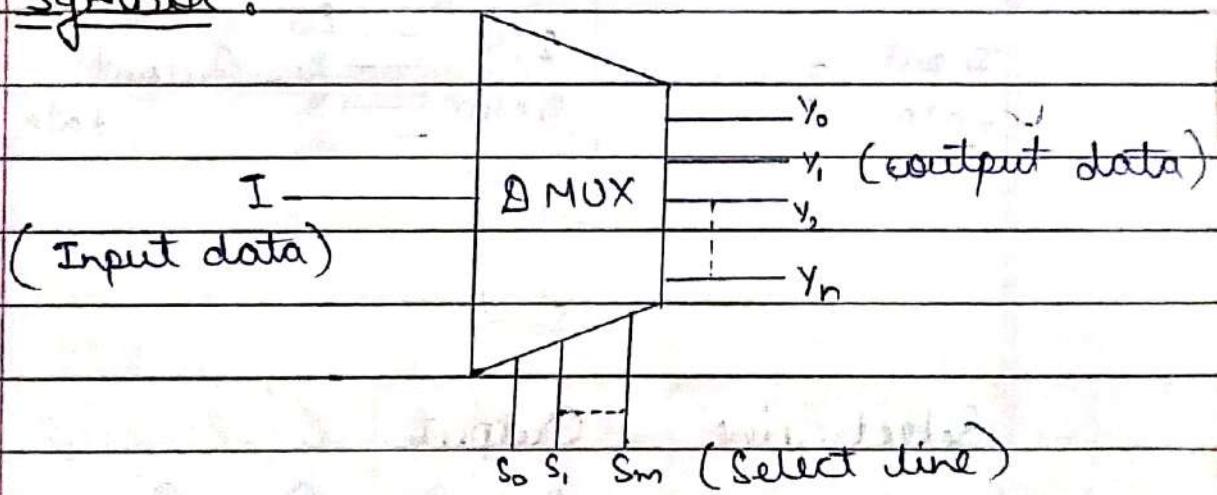
$$Y = I_0 \bar{S}_0 \bar{S}_1 \bar{S}_2 + I_1 \bar{S}_0 \bar{S}_1 S_2 + I_2 \bar{S}_0 S_1 \bar{S}_2 + I_3 \bar{S}_0 S_1 S_2 + \\ I_4 S_0 \bar{S}_1 \bar{S}_2 + I_5 S_0 \bar{S}_1 S_2 + I_6 S_0 S_1 \bar{S}_2 + I_7 S_0 S_1 S_2$$

* A - Multiplexer (A-MUX) :-

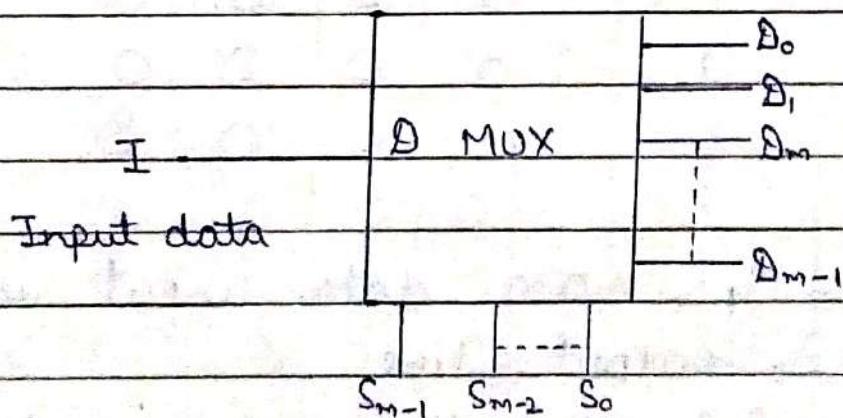
It is a combination circuit which means "one to many". It means it having one input line and many output are present.

Features of De-multiplexer - It is a logic circuit which receives information through a single input line and transmit the same information over one of the possible 2^m output lines. To which output line the data is passed will depend upon the select line combination. It generally represented as 1:N.

Symbol :-

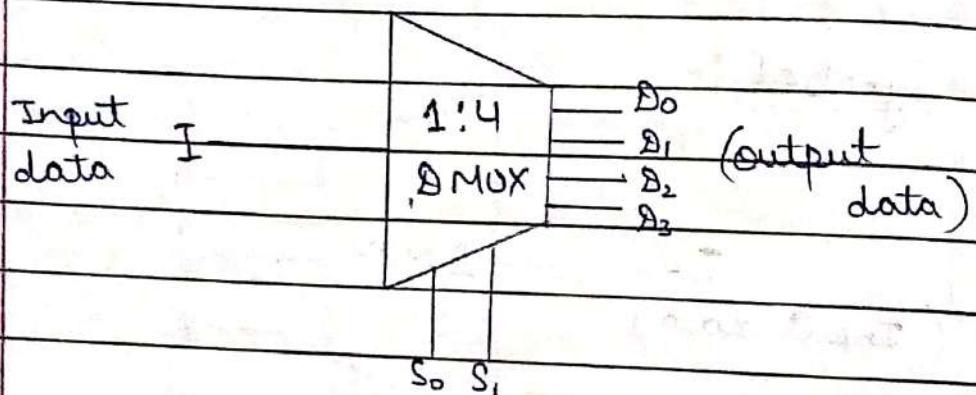


Block diagram of De-Multiplexer:-



* 1:4 line De-MUX :-

In this De-multiplexer it has 4 output and 2 select lines are present. The two data select line enable only one gate at a time and the data appearing on the data input line will pass through the selected gate to associated data of line.



Select line	Output
S ₁ S ₀	D ₀ D ₁ D ₂ D ₃
0 0	0 0 0 0
0 1	0 0 0 0
1 0	0 0 0 0
1 1	0 0 0 0

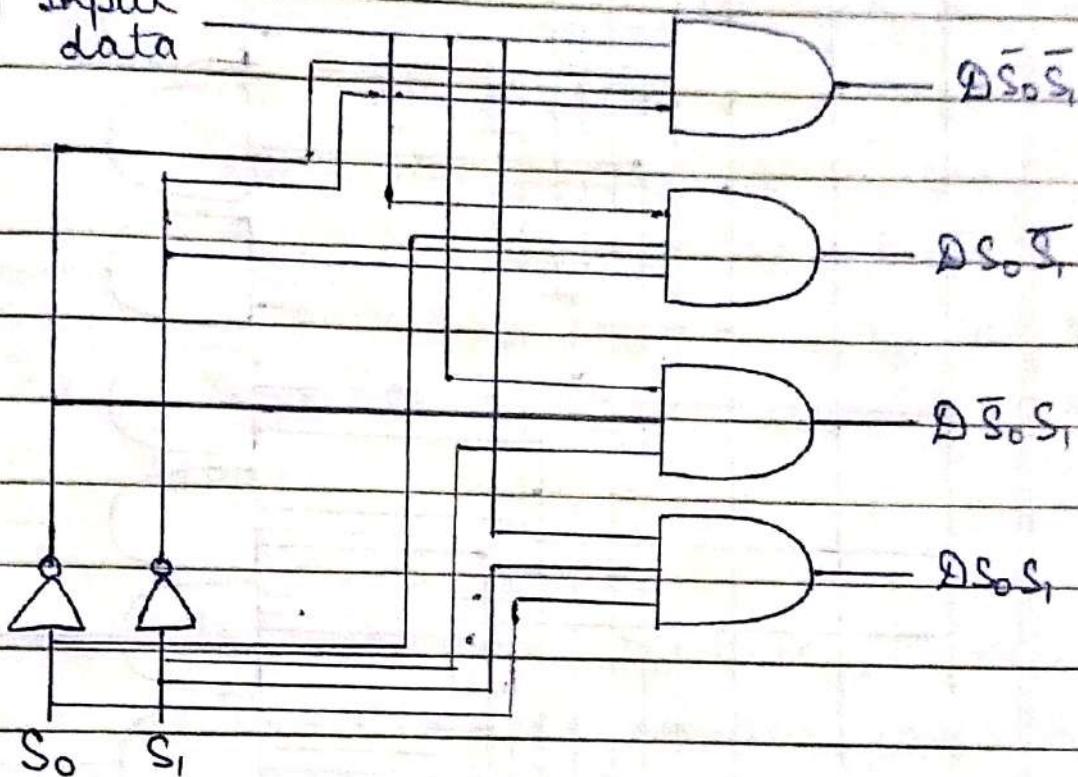
When S₁, S₀ = 0 0 data input is pass to D₀ output line.

When S₁, S₀ = 0 1 then data input is pass to D₁ output line.

When S₁, S₀ = 1 0 then data input is pass to D₂ output line.

When $S_1 S_0 = 11$ then data input is pass to D_5 output line.

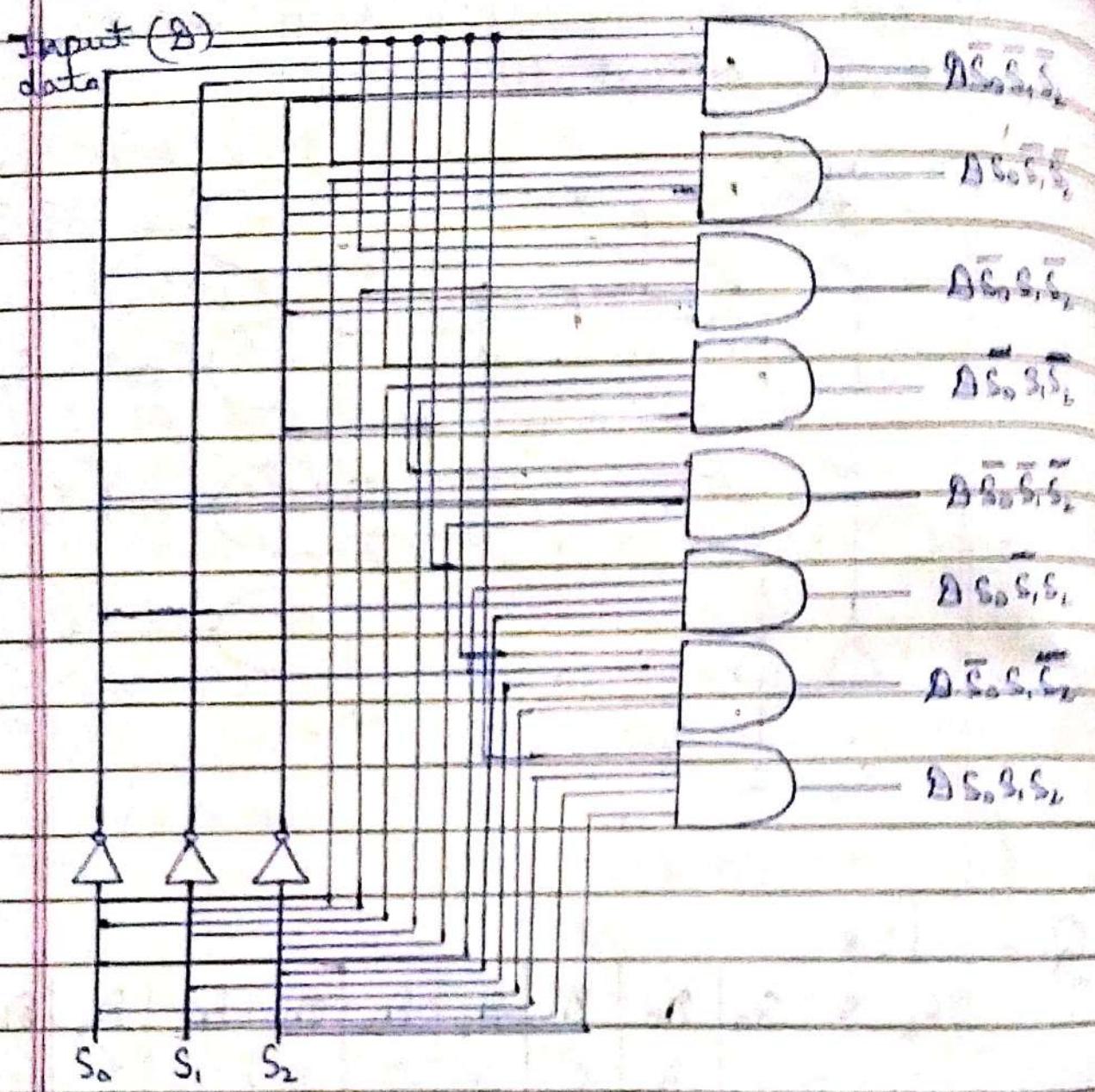
(D) Input data



Q.

1:8

	S_2	S_1	S_0	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	0	0	0	0	0	0	0	0	0	0	0
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	0	0	1	0	0	0	0	0	0	0	0
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	0	1	0	0	0	0	0	0	0	0	0
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	0	1	1	0	0	0	0	0	0	0	0
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	1	0	0	0	0	0	0	0	0	0	0
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	1	0	1	0	0	0	0	0	0	0	0
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	1	1	0	0	0	0	0	0	0	0	0
$D\bar{S}_0 \bar{S}_1 \bar{S}_2$	1	1	1	0	0	0	0	0	0	0	0

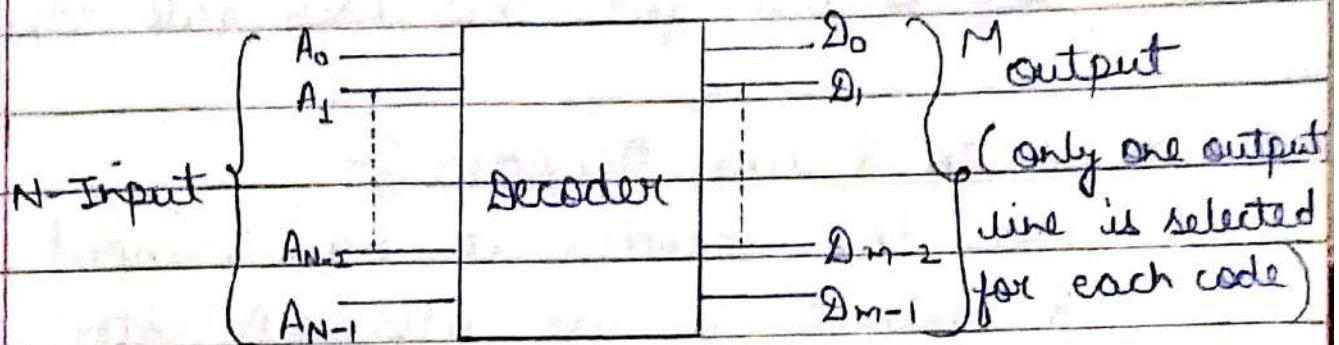


* Decoder :-

Decoder is a combinational circuit that converts an N-bit binary input code into m output line such that only one output line is activated for each of the possible combinations of input.

In other words a decoder

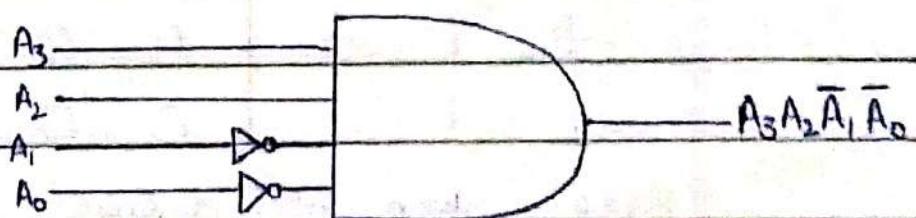
has n input lines to handle N-bits and from one to 2^n output lines to indicate the presence of one or more N bit combination.



For example:- Suppose we have to determine when 1100 will occur on the input of its digital circuit. Then we have to use AND gate as decoding element because it produce a high output when all of its input are high. So in order to make this we have to make sure that all inputs of AND gate are equal to high.

$A_3 \quad A_2 \quad A_1 \quad A_0$

1	1	0	0
---	---	---	---



So the logic eqn is -

$$Y = A_3A_2\bar{A}_1\bar{A}_0$$

- * Features :-
 - Use to convert binary format into non coded format.
 - Some decoders are use to provide active low output for this AND gate are used.

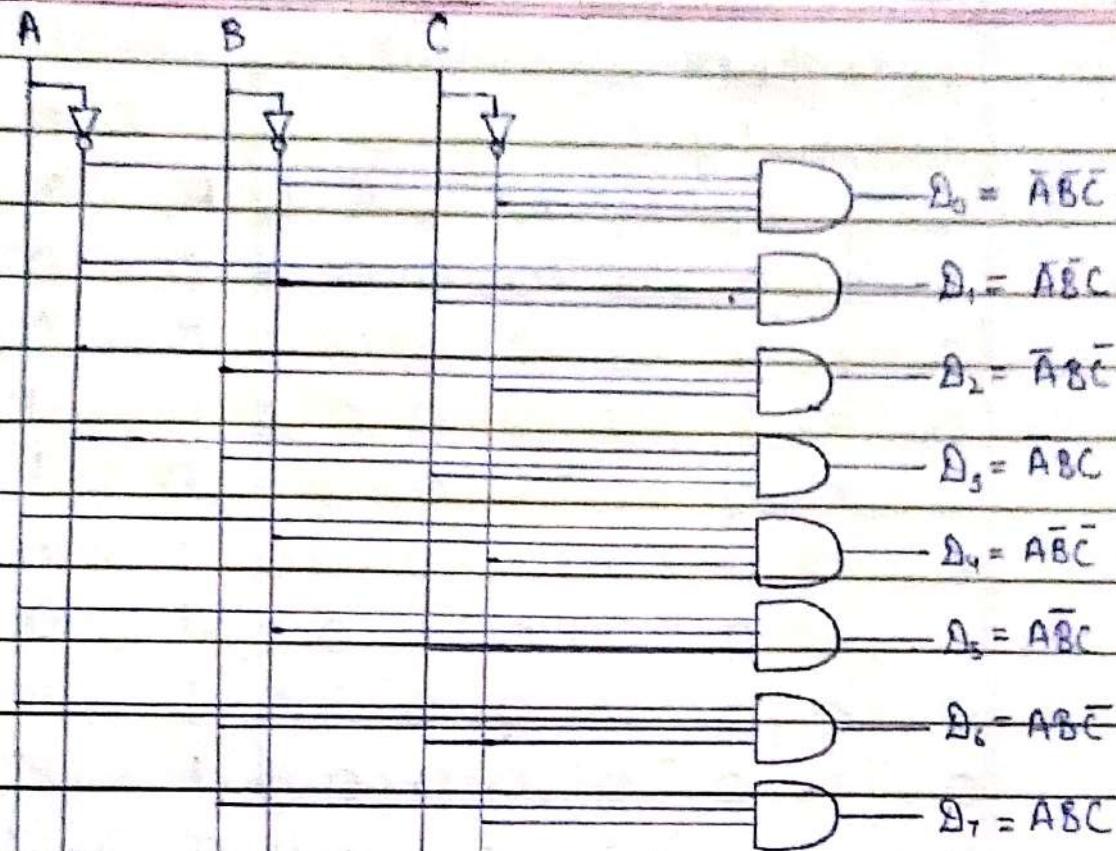
* 3 to 8 line Decoder :-

For this decoder it has 3 input and 8 output in we all AND gates. So the output is high. For acting low output AND gate are used. It is also known as binary to octal decoder.

Truth table -

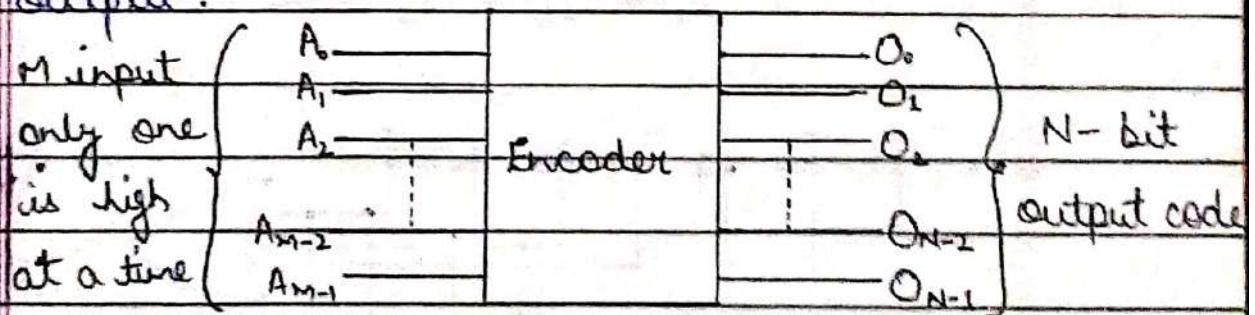
Input	Output
A B C	D ₀ D ₁ D ₂ D ₃ D ₄ D ₅ D ₆ D ₇
0 0 0	1 0 0 0 0 0 0 0
0 0 1	0 1 0 0 0 0 0 0
0 1 0	0 0 1 0 0 0 0 0
0 1 1	0 0 0 1 0 0 0 0
1 0 0	0 0 0 0 1 0 0 0
1 0 1	0 0 0 0 0 1 0 0
1 1 0	0 0 0 0 0 0 1 0
1 1 1	0 0 0 0 0 0 0 1

Here when ABC = 000 then D₀ is selected line and provided at the output $\bar{A}\bar{B}C$. When ABC = 001 then D₁ is selected line and provided at the output $\bar{A}BC$. and so on.....



* Encoder :-

It is a circuit which converts decimal digits or alphabetic characters into coded representation like - binary code as an output.



* Octal to binary encoder :- It is also known as 8 line to 3 line encoder. It accepts 8 input line and produce 3 bit output code. The 8 inputs are 0, 1, 2, 3, 4, 5, 6 and 7.

Truth Table -

Input								Output		
I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	A	B	C
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	1	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Here in first case when I_0 is 1 then ABC is 000 because I_0 is selected.

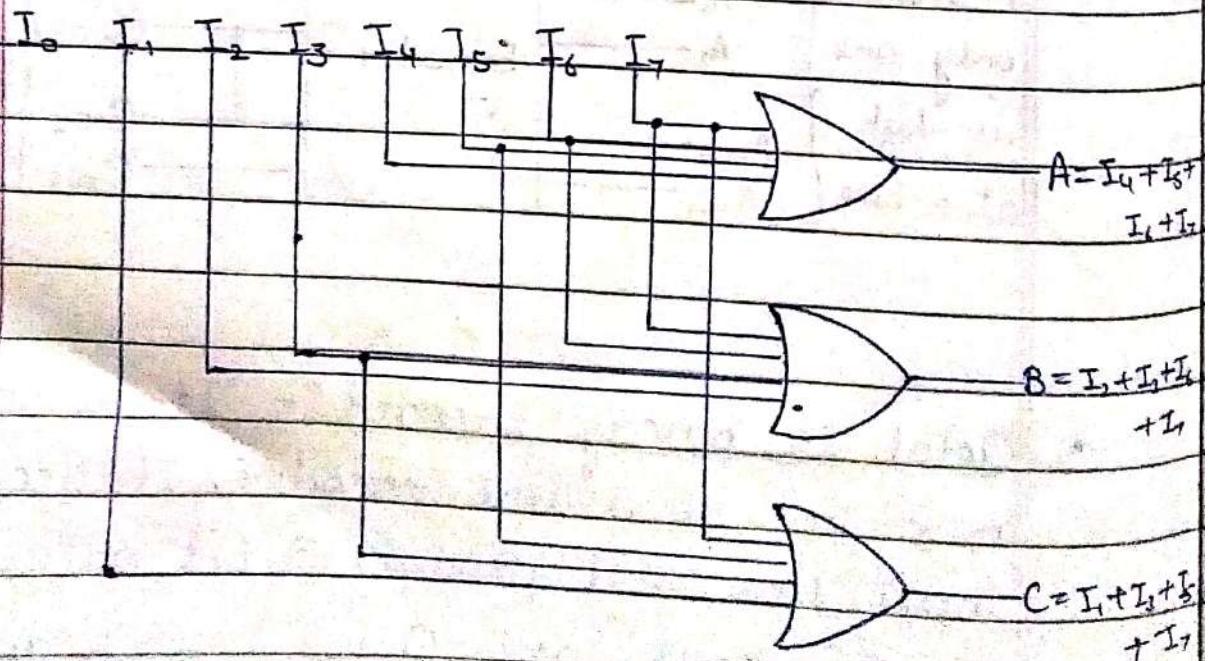
When I_7 is 1 then ABC is 111.

The expression for ABC is given as -

$$A = I_4 + I_5 + I_6 + I_7$$

$$B = I_1 + I_2 + I_3 + I_6 + I_7$$

$$C = I_1 + I_3 + I_5 + I_7$$



* Decimal to Binary encoders-

Input	Output
$I_0 \ I_1 \ I_2 \ I_3 \ I_4 \ I_5 \ I_6 \ I_7 \ I_8 \ I_9$	$A \ B \ C \ D$
1 0 0 0 0 0 0 0 0 0	0 0 0 0
0 1 0 0 0 0 0 0 0 0	0 0 0 0 1
0 0 1 0 0 0 0 0 0 0	0 0 0 1 0
0 0 0 1 0 0 0 0 0 0	0 0 0 0 1 1
0 0 0 0 1 0 0 0 0 0	0 1 0 0
0 0 0 0 0 1 0 0 0 0	0 1 0 1
0 0 0 0 0 0 1 0 0 0	0 1 1 0
0 0 0 0 0 0 0 1 0 0	0 1 1 1
0 0 0 0 0 0 0 0 1 0	1 0 0 0
0 0 0 0 0 0 0 0 0 1	1 0 0 1

$$A = I_8 + I_9$$

$$B = I_4 + I_5 + I_6 + I_7$$

$$C = I_2 + I_3 + I_6 + I_7$$

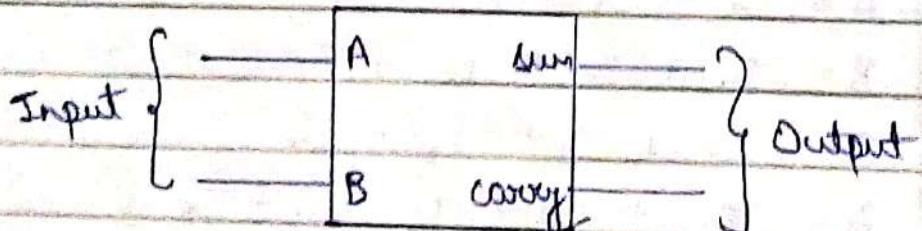
$$D = I_1 + I_3 + I_5 + I_7 + I_9$$

* Adder :-

- 1- Adder is use to perform the addition function of two numbers.
- 2- They are use in many digital circuit in which numerical data is to be process.
- 3- They are classified as -
 - (i) Half adder
 - (ii) Full adder

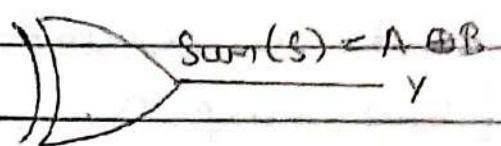
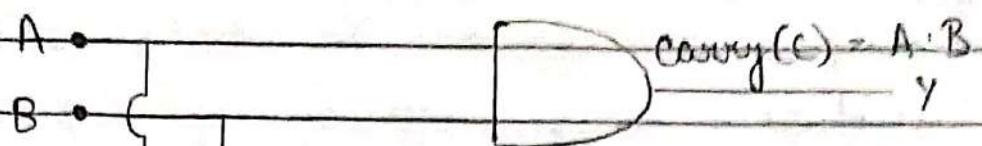
Half Adder - It is a type of combination circuit that perform the addition of two bits and produce two outputs as sum and carry.

The symbol of half adder is -



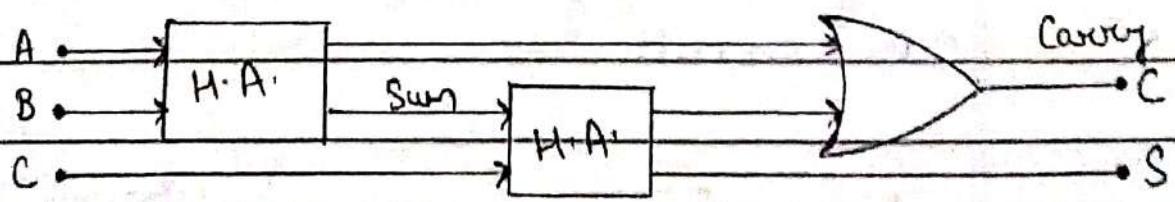
Designing of half adder -

Input	Output		
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

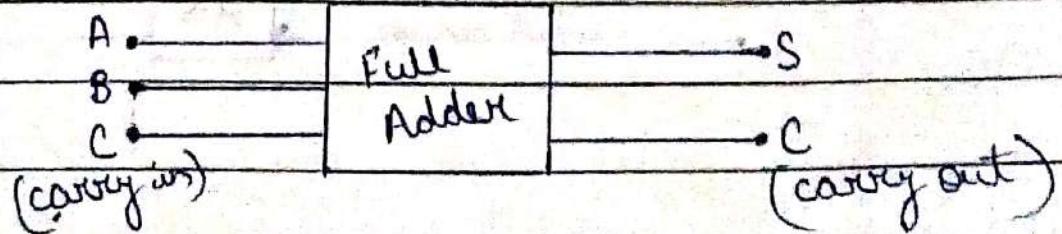


Full adder — The half adder accomplishes the first step in the addition of bits. Two half adder are combine into a full adder in order to add the carry bit to the sum.

Input	Output
A B C	S C
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
.	.

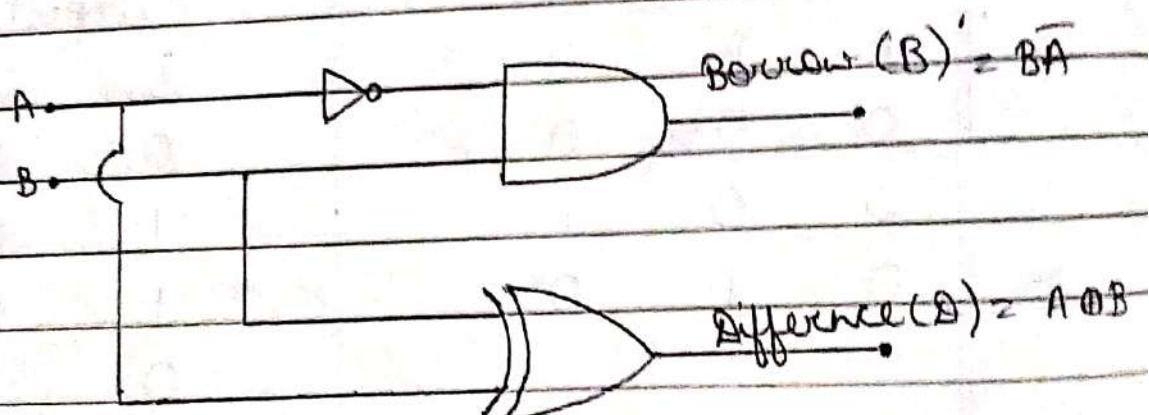


Full Adder

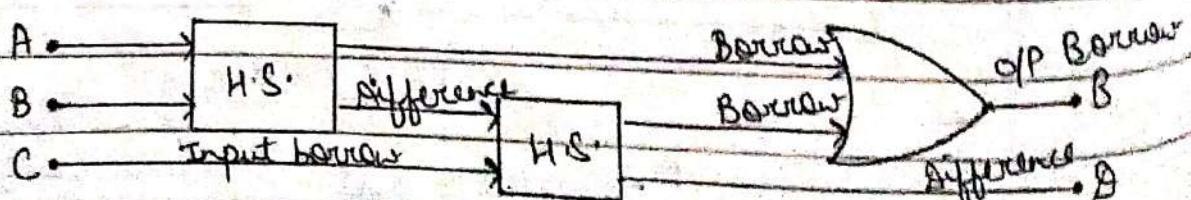


* Half Subtractor - The half subtractor subtracts only two binary digits at a time and produced an output of a difference and a borrow. It has two inputs and two outputs.

Input		Output	
A	B	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Full Subtractor - Like a full adder, full subtractor also handle 3 bits and can be designed using two half subtractor and an OR circuit.



★ ROM :-

- 1- Stable and impervious to power supply.
- 2- Non-volatile.
- 3- For permanent storage of information.
- 4- It stores computer BIOS.

BIOS - Code tells processor how to access its resources upon powering up the system. Code for embedded system is application.

Different types of ROM -

1. PROM - Programmable read only memory

- 1- Program is written to ROM at the time of manufacturing.
- 2- Contents can be programmed by user with a special PROM programmer.
- 3- Flexible and economical storage for fixed program and data.

2. EEPROM -

- 1- Allows erasability of ROM and reprogram.
- 2- Erasability is possible with UV light beam.
- 3- Store information for longer duration.

3. EEPROM -

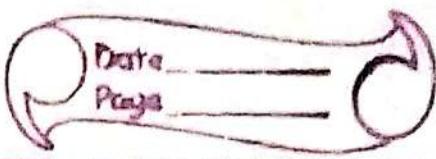
- 1- Contents can be erased electrically with the application of high voltage.

4. Floppy disk (Diskette) -

- 1- Convenient bulk storage device.
- 2- Can be taken out of computer.
- 3- Size available 5.25 inches or 3.5 inches.
- 4- Contain in rigid plastic case.
- 5- Read / Write of this drive can write or read information from both sides of the disk.
- 6- Data is stored in magnetic form.
- 7- 3.55 inch disk has storage upto 1.44 MB.
- 8- Disk is organised into sectors and tracks.
- 9- 3.5 inch disk has 80 concentric circles called tracks and each track is divided into 18 sectors. Each sector store 512 bytes of data + address.

5. CD-ROM -

- 1- CD-ROM is for read only purpose.
- 2- 800 MB of data can be stored including address.
- 3- CD-ROM is reliable for distribution.
- 4- Information is evenly distributed across segments of same size.
- 5- Data stored on track increases as we go outwards.



- 6- CD-ROM is rotated at variable speed for reading data.
- 7- Information is return by creating pits on the disk surface by shining a laser beam.
- 8- As disk rotate, the laser beam traces out a continuous spiral. One is return on disk circular pit of around $0.8\text{ }\mu\text{m}$ (micrometre).
- 9- No pit for 0 is created.
- 10- Laser beam for reading is used.

Advantage -

- 1- Large data storage.
- 2- Fast application, mass application and inexpensive.
- 3- Removable.

Disadvantages -

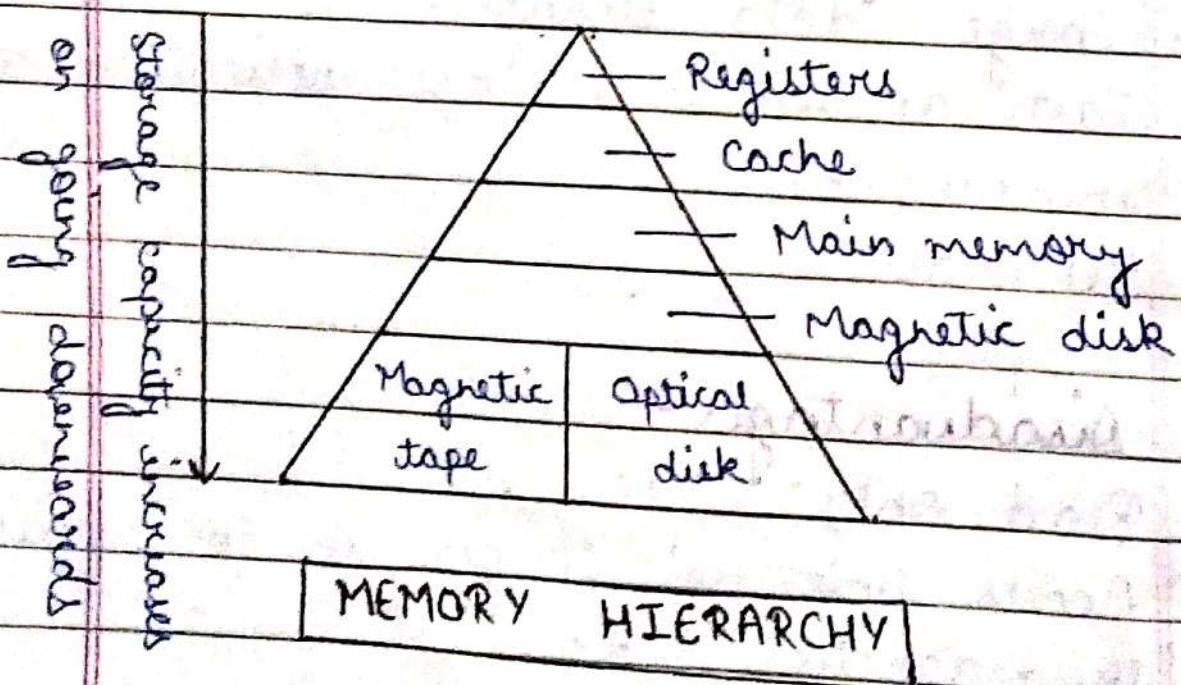
- 1- Read only type.
- 2- Access time is longer in comparison to magnetic disk.

* Memory - Electronic circuitry which allows data to be stored and retrieved when required.

Auxiliary memory - Back-up storage provider. Large storing capacity but slow access rate.

Main memory - It is the memory unit that communicate directly with CPU.

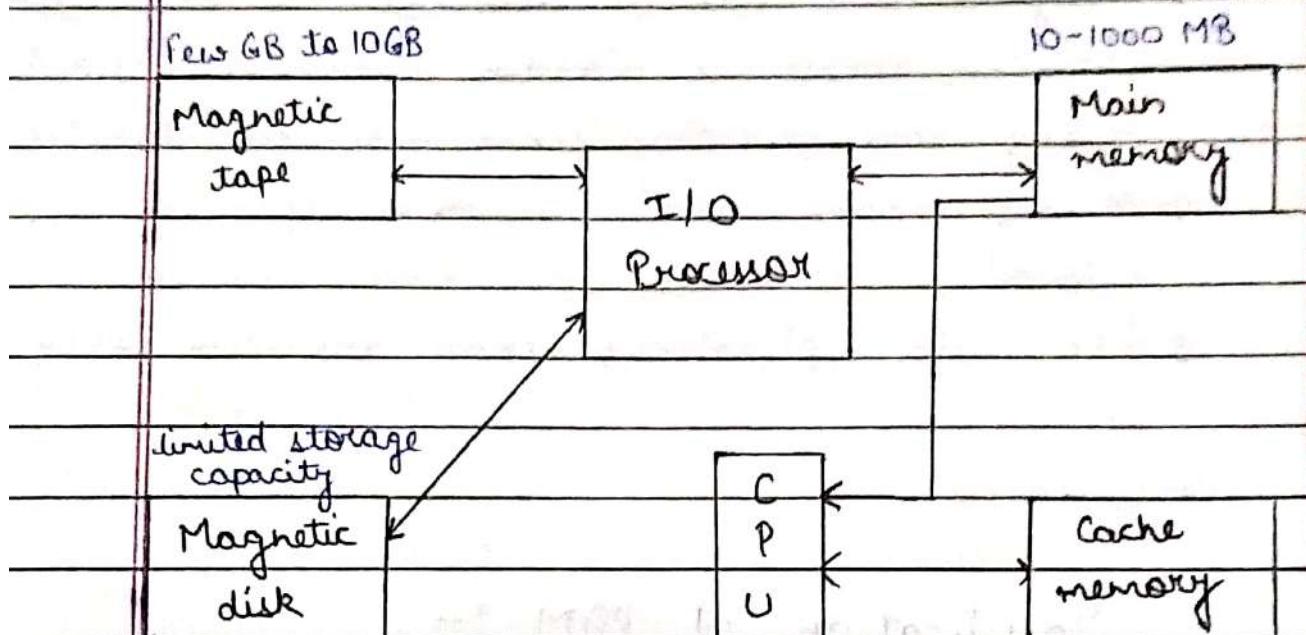
Cache memory - Very small but very high access speed and expensive.



CPU Registers - CPU registers are the CPU local memory and are accessed in few nano seconds.

I/O Processor - Translating I/O requests.

(Page)
auxillary and main memory.



MEMORY ORGANISATION

* Classification of computer memory :-

1) Location

- * CPU
- * Internal
- * External

2) Access method

- * Sequentially
- * Direct
- * Random

3) Performance

- * Access time
- * Cycle time
- * Transfer time

4) Physical type

- * Semi conductor
- * Magnetic

5) Physical characteristics

- * Volatility based
- * Erasability/Non erasability

6) Capacity

- * Word size
- * No. of words

7) Unit of transfer

- * Word
- * Block

* RAM :-

- 1- Any address can be accessed at any time it can be access in random manner without going through any other memory location.
- 2- Access search time for each location is same.
- 3- It store applications that are currently in use.
- 4- Volatile.

Classification of RAM :-

① Static RAM -

- 1- It contains array of flip flops.
- 2- Each flip flop maintains a single bit of data within a single memory address. It is called static RAM because it doesn't require external refresh circuit as long as power is required.
- 3- It is a semiconductor type memory.
- 4- Used for high speed registers, cache and small memory banks such as - counter buffers.
- 5- A time of 10-30 nano seconds is required to access a data.

② Dynamic RAM -

- 1- It holds its data if it is continuously

access by special logic called refresh circuit.

- 2 This circuit reads many 100 times per second contents of each memory cell.
- 3 Reading action itself refreshes contents of memory even if power is supplied to it.

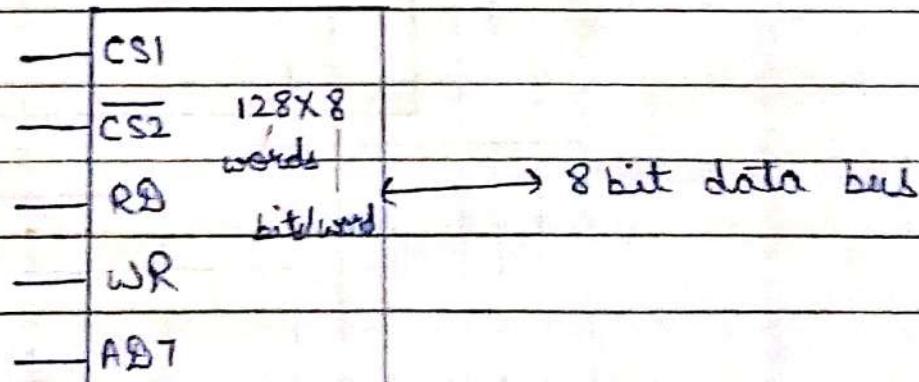
Advantages -

- 1 Used instead of SRAM
- 2 Much higher capacity.
- 3 Cheaper than SRAM.

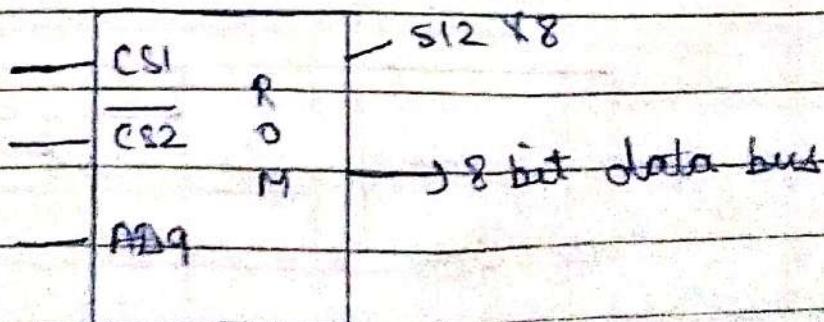
Disadvantage -

Slower than SRAM due to overhead circuit

* RAM chip (Block diagram) -



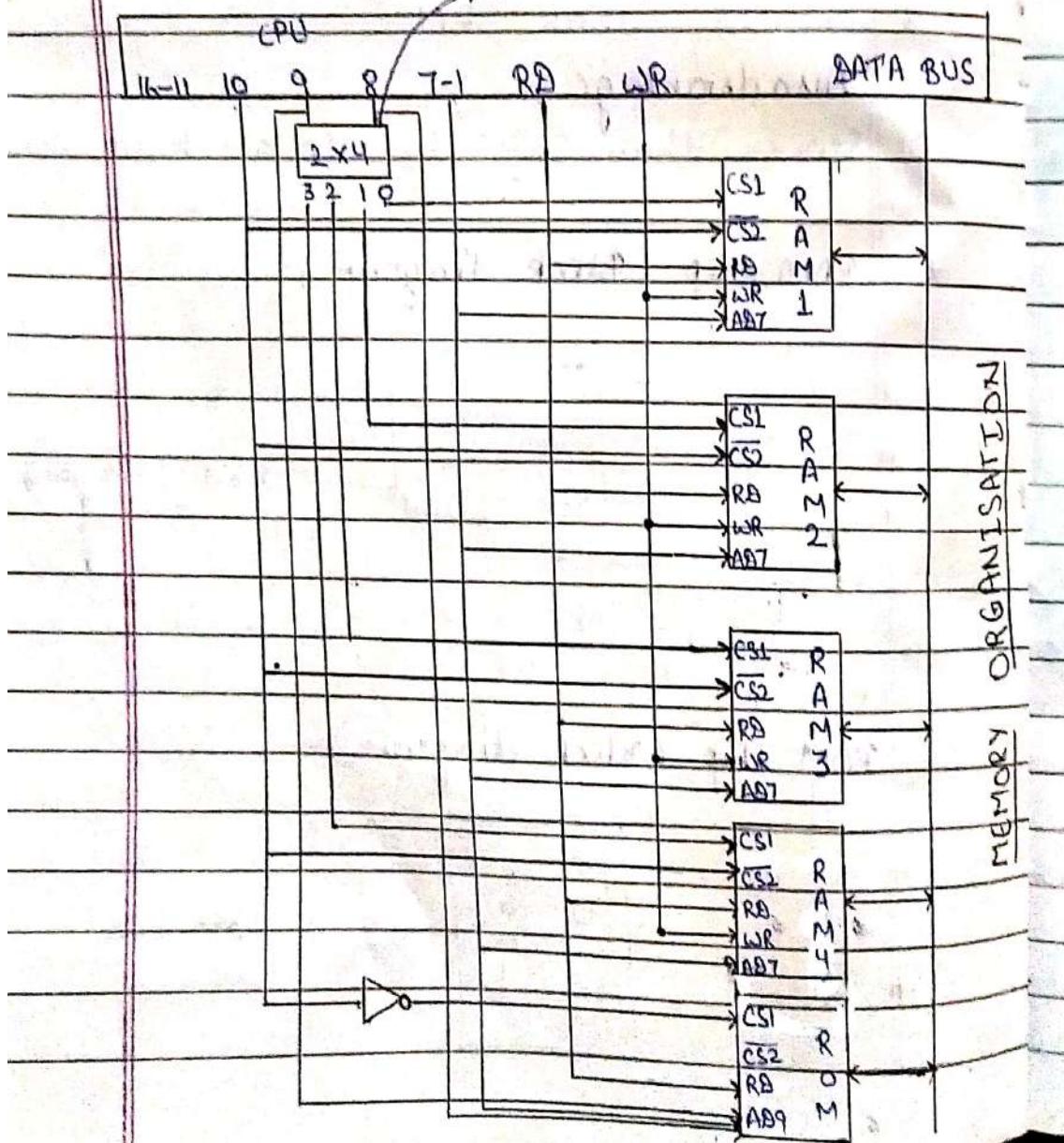
ROM chip (Block diagram) -



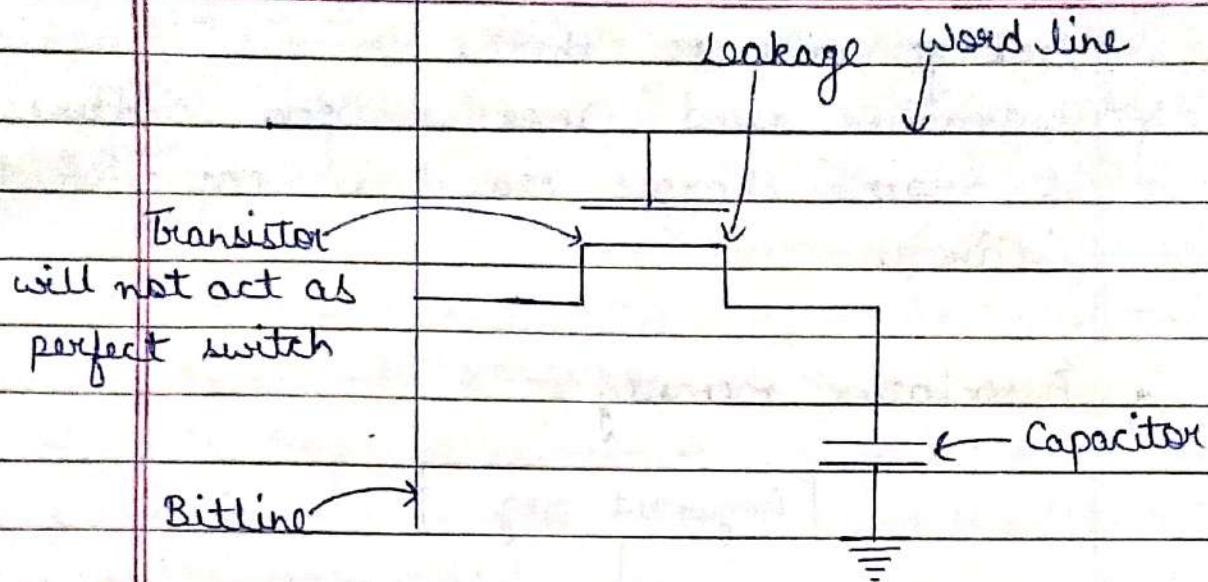
Function Table (RAM)

CS1	CS2	RD	WR	Memory function	Data bus
0	0	X	X	Inhibit	High Impedance
0	1	X	X	Inhibit	High Impedance
1	0	0	0	Inhibit	High Impedance
1	0	0	1	Write	Input to RAM
1	0	1	X	Read	Output data from RAM
1	1	X	X	Inhibit	High Impedance

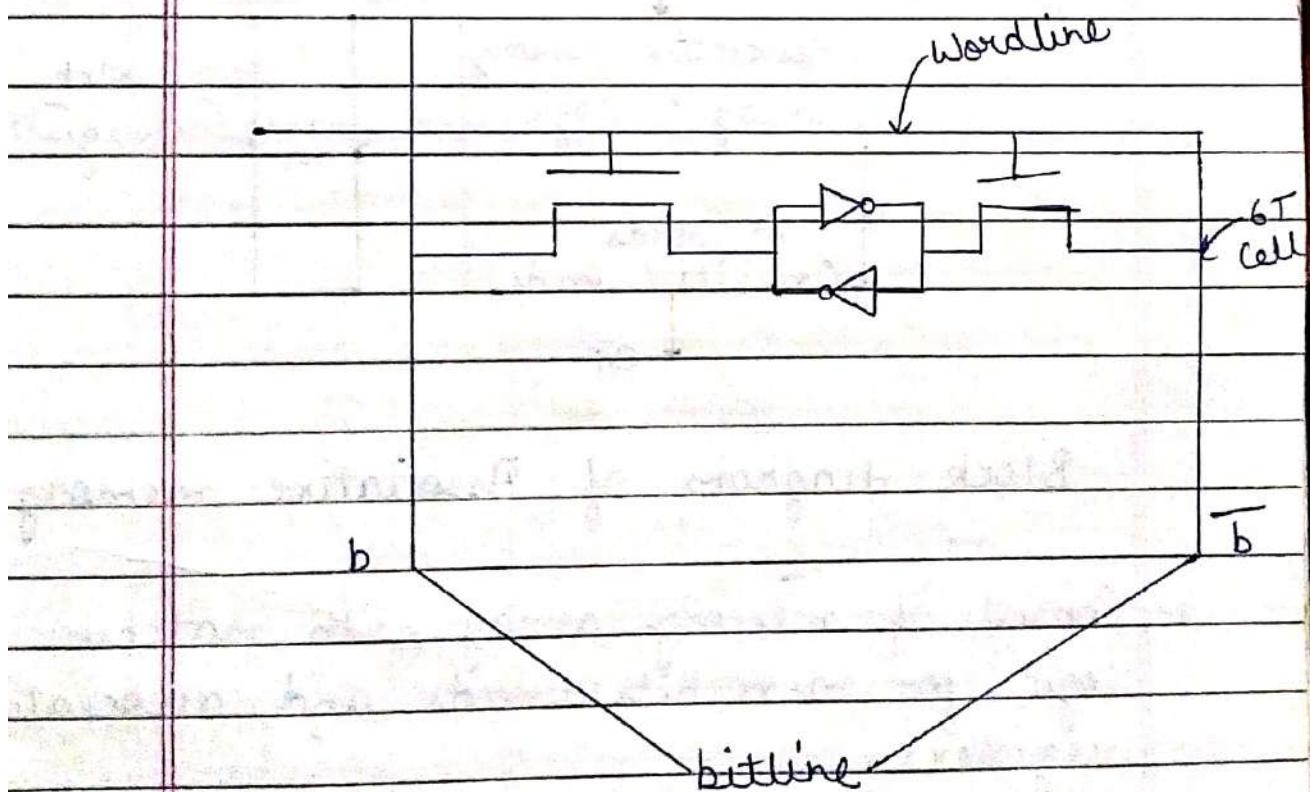
Decoder



Date 21/May/2018
Page _____



(Basic cell of SRAM)

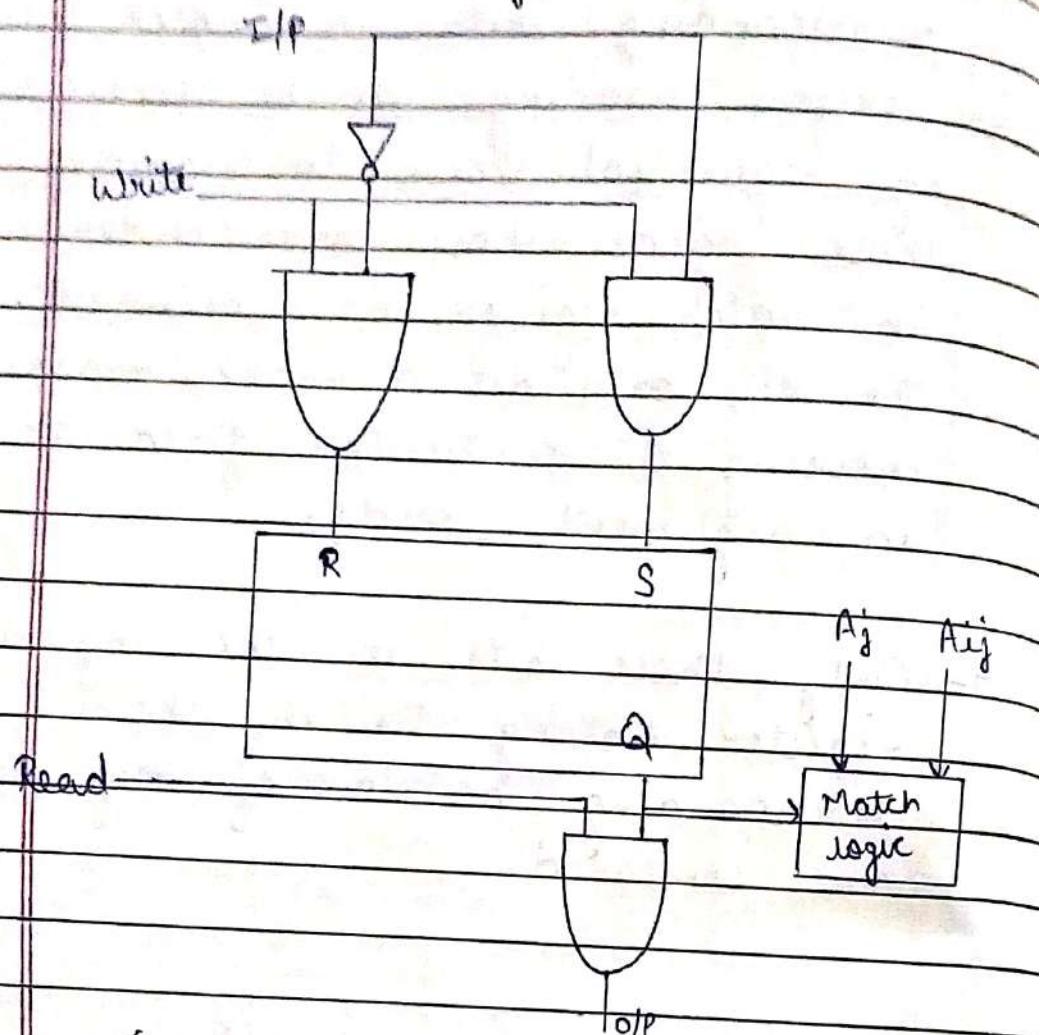


(Basic cell of DRAM)

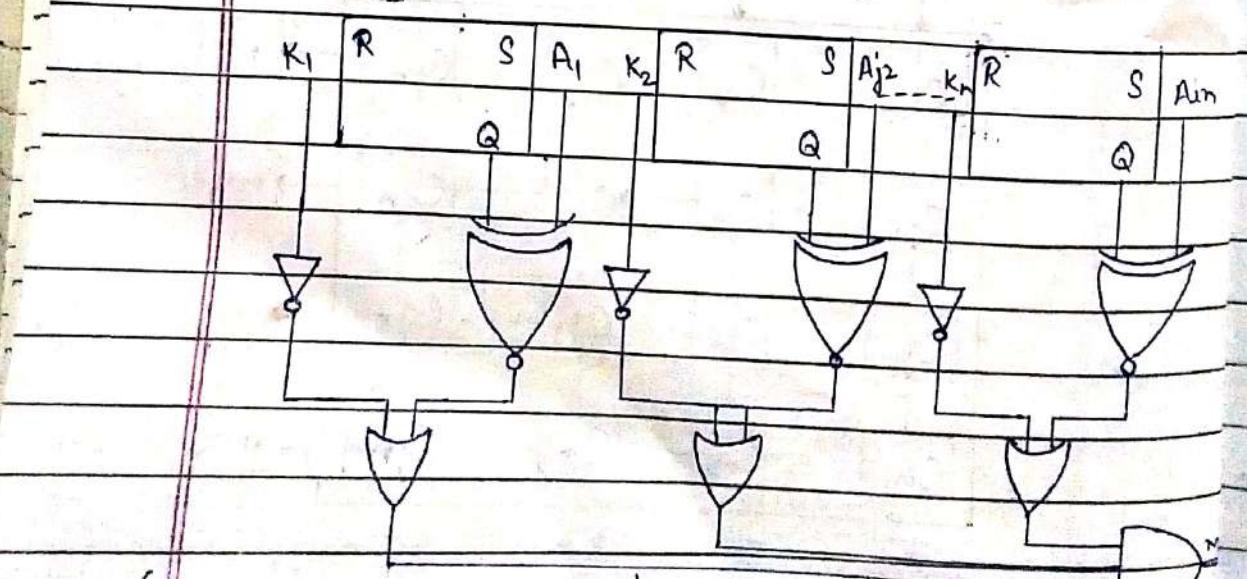
* Disadvantages of SRAM -

- 1- The transistor will not work as perfect switch.

* Associative memory -



(Structure of Associative memory using logic gates)



(Expanded figure showing the opⁿ working)

1- R-S flipflop :- storage element (reading, writing and matching the contents of cell)
Match logic - Compares the bit in storage cell with corresponding unmasked bit of arguments and provides output for decision logic that sets the corresponding bit in match register.

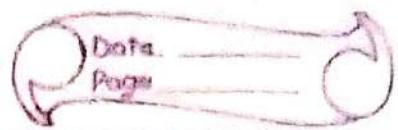
2- The cell bit O_{ij} is compared with A_{ij} (Ex-NOR when input is same Ex-NOR give 1).

Output of Ex-NOR is valid when corresponding bit in key register is logic 1.

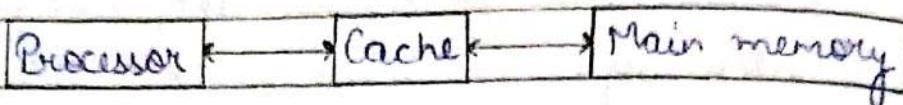
The condition is implemented using OR gate.

When corresponding bit in key register = 1, the inverter gate gives 0 which is the input of OR gate, and follows second input that is the comparison output, otherwise output of OR-gate is 1.

The output are AND-ed with n input AND gate to check whether all bits are matching with A_{ij} register contents.

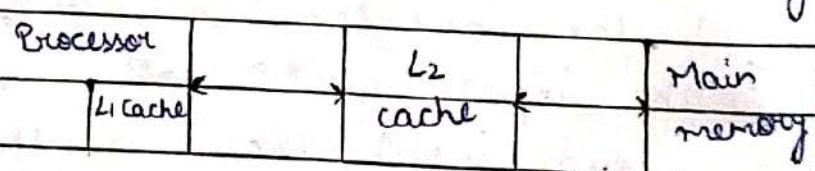


* Cache memory

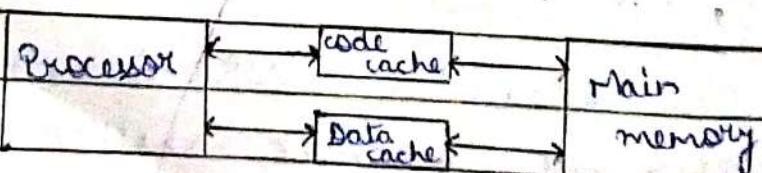


Cache memory organisation

- 1- It is a very high speed memory that is used in computer system to compensate the speed difference between the main memory access time and processor logic.
- 2- The cache memory access time is less than access time of the main memory by factor 5 to 10.
- 3- It stores program segment currently being executed in the CPU and data frequently used.
- 4- Placement can be done in 2 ways:-



L₁ and L₂ cache placement



Split cache organisation

Split cache organisation is done to be use in pipelining.



- * Basic operations of cache memory :-
When CPU needs to access memory, the cache is examined first. If word is found in cache it is read from the memory, if not then main memory is accessed.
- * Performance of cache :- Is frequently measured in terms of quantity called hit ratio.

Hit - When CPU refers to memory (cache) and find the word in cache it is said to produce hit.

Miss - When data is not found in cache memory it is said to produce miss.

t_c = cache access time

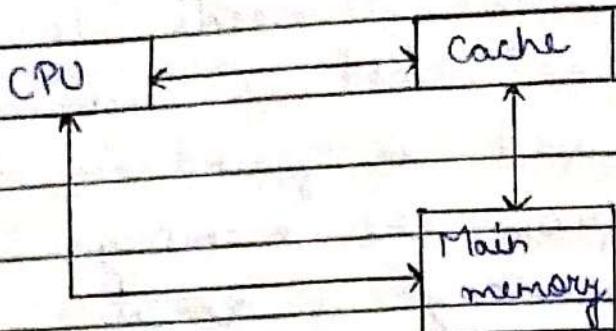
h = hit ratio = $h/(htm)$

t_m = main memory access time

$$A = \frac{t_c}{t_m} = \frac{t_c}{htc + (1-h)(t_c + t_m)}$$

$$= \frac{t_c}{t_c [1 + (1-h)(1 + t_m/t_c)]} = \frac{1}{Hr(1-h)}$$

* Mapping (cache)



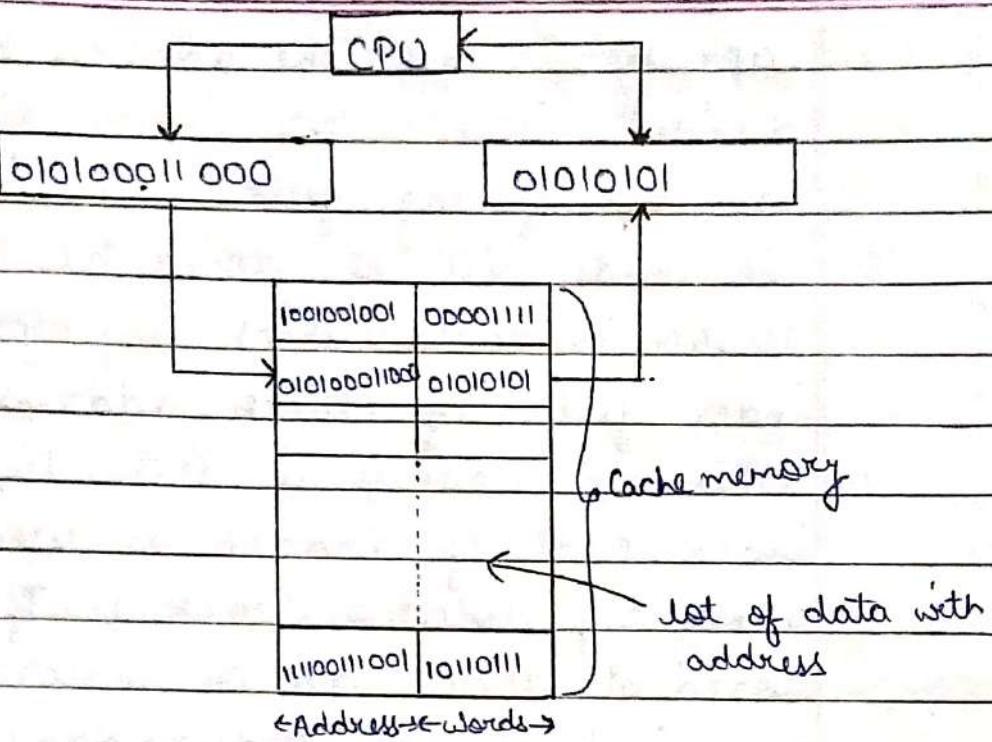
CPU will send n-bit address to cache memory in order to find data (word).

* Memory mapping (cache) :- Transformation of data from main memory to cache memory is mapping. Mapping can be done in three ways -

- 1) Associative mapping
- 2) Direct mapping
- 3) Set associative mapping

1) Associative mapping -

- 1- Both words and address of word are stored in cache memory.
- 2- The address bit send by CPU get matched then corresponding word is sent to central processor, if not then the word is searched in main memory. The word along with address is copied from main memory to cache memory.



(Associative mapping)

2) Direct mapping -

- 1- Suppose main memory size = 4K (4×1024 bytes) and cache memory is of 1k.
- 2- Thus a cache memory needs 10 bits to address a word and main memory require 12 bits to address a word.
- 3- In direct mapping 12 bit address sent by CPU is divided into two parts called tag field and index field.
- 4- The index field has the number of bits equal to number of bits required to address a word in cache.
- 5- Thus if a computer has main memory of capacity 2^m and cache memory of

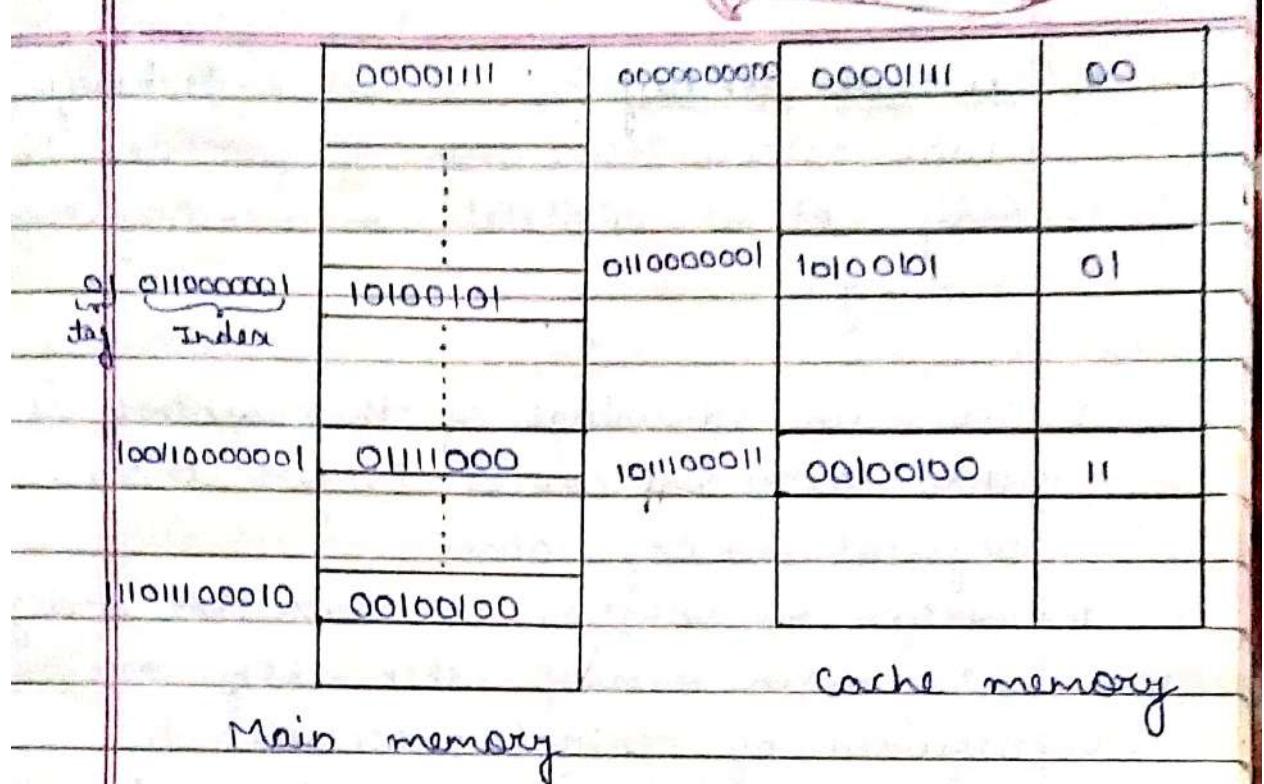


capacity 2^n , then the address bit will be divided into n bits index field and $m-n$ bit of tag field.

- 6- The words will be stored at that location in cache which is represented by index field of their address.
- 7- When an address is sent to CPU, the index part of address is used to get a memory location (cache). If tag stored at that location matches the tag field of requested address, then word is fetched. Else if tag does not match the word is search in main memory.
- 8- When word is move in cache from main memory then its address is divided into index and tag field.

Disadvantage of Direct Mapping :-

Hit ratio drops if two or more words whose address have same index but different tags are accessed.



3) Set associative memory -

- 1- Each word of cache memory can store 2 or more words of main memory under the same index address.
- 2- Each data is stored along with its tag and the number of tag, data pair in one word of cache is said to form a set.

Tag	Data	Tag	Data
0000110010	00	10101010	10
0000111011	01	11000011	10

Two way set associative memory mapping

* Virtual memory :- It is a technique that allows execution of processes that may not be completely in memory (main memory)

- 1- The main advantage of this scheme is that programs can be larger than physical memory capacity.
- 2- Portion of program or data are brought into main memory when they are required by central processor.
- 3- Virtual memory is separation of user logical memory from physical memory.
- 4- This separation allows an extremely large virtual memory to be provided for programmers when physical memory is small.
- 5- The address generated by user program is called virtual address and the set of virtual address may virtual address space.
- 6- A main memory called location or physical address and set of such local addresses are called memory space (physical address space).

Note - size of virtual memory $>$ size of physical memory

Suppose a computer of main memory capacity of 32 k words.

Since $32k = 2^{15}$, 15 bits will be needed to specify a physical address.

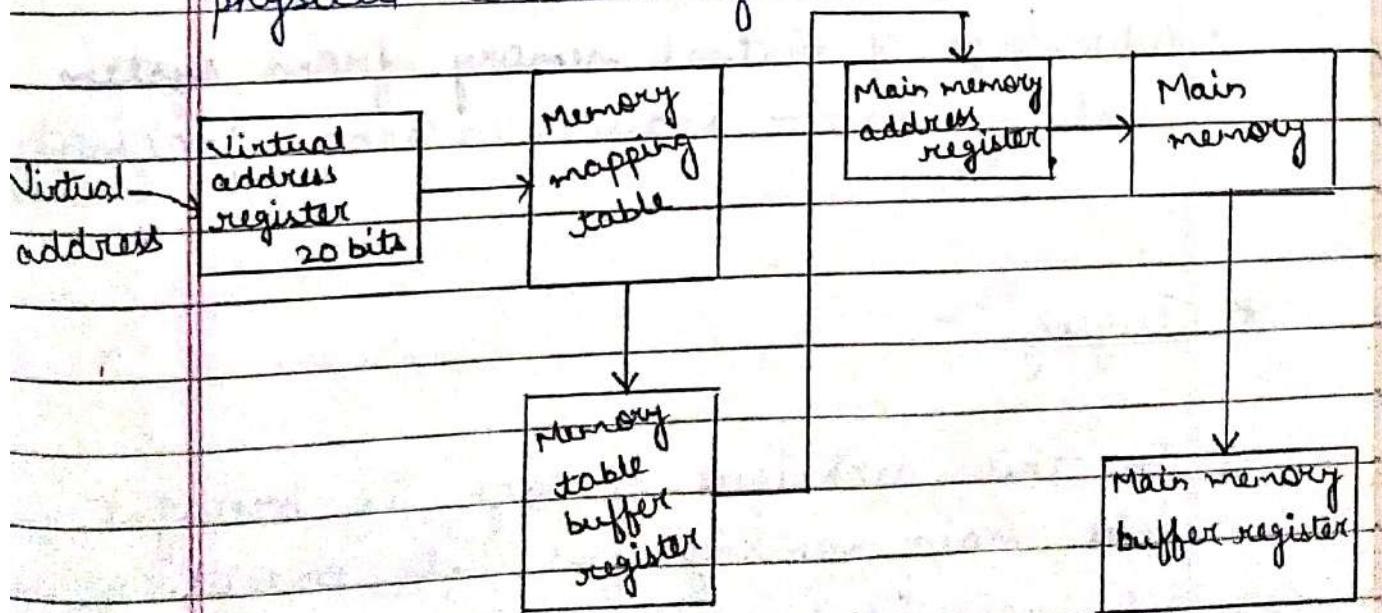
Suppose that computer has auxiliary memory for storing 2^{20} words.

Let N = Address space and M = memory space

$$N = 1024k \text{ and } M = 32k$$

The address bit of instruction code will consist of 20 bits but physical memory addresses must be specified using only 15 bits.

Therefore CPU will reference the instructions and data with 20 bit addresses, but the information at this address is taken from physical memory rather than auxiliary memory. Thus it is required to map a virtual address of 20 bit to a physical address of 15 bit.





Advantages of Virtual memory :-

- 1- Only a part of program needs to be in memory (main). Swapping can be done for execution of program.
- 2- It increases degree of multiprogramming.
- 3- It needs to allow pages to be swapped-in or swapped-out.
- 4- Logical address space is greater than physical address space.

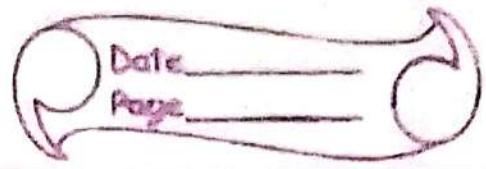
Disadvantage :-

We need to maintain entire image of process in disk storage.

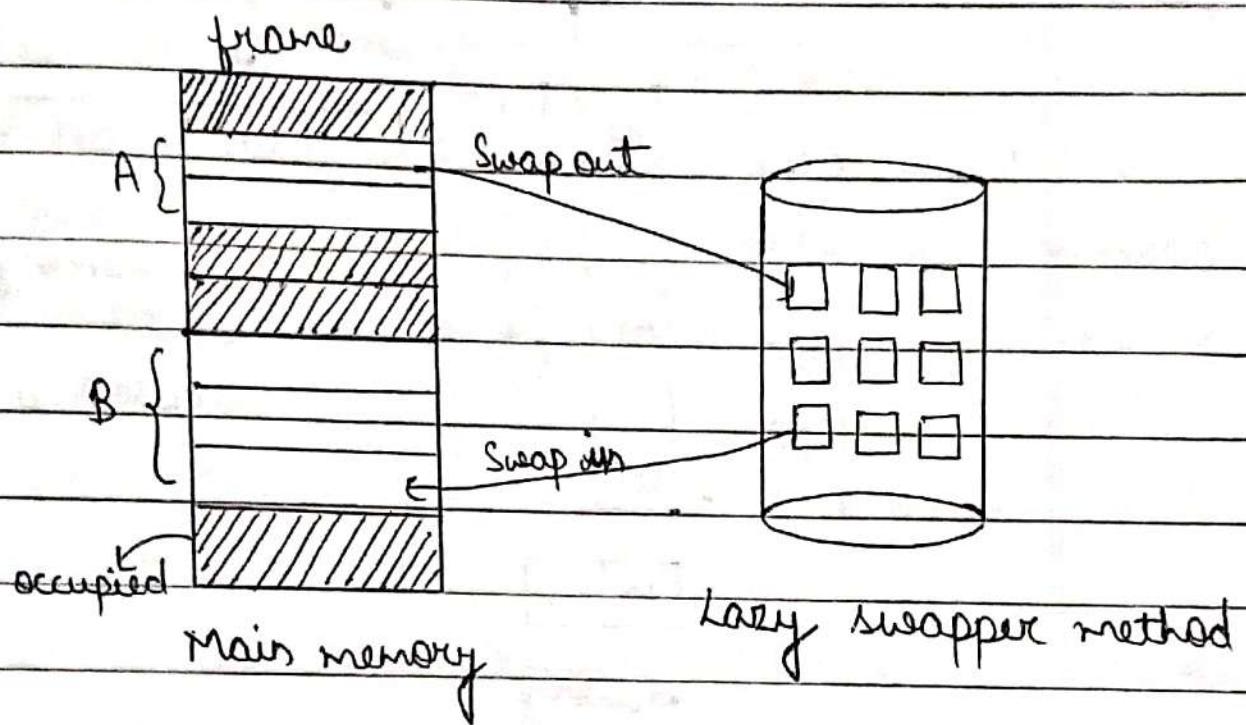
- Advantage of virtual memory with user point of view - logical address space is big.
- Advantage of virtual memory from system point of view - Reduces external fragmentation

* Paging :-

- 1- In this technique a page is brought into main memory for its execution only when it is demanded.



2 Combination of paging and swapping are used.

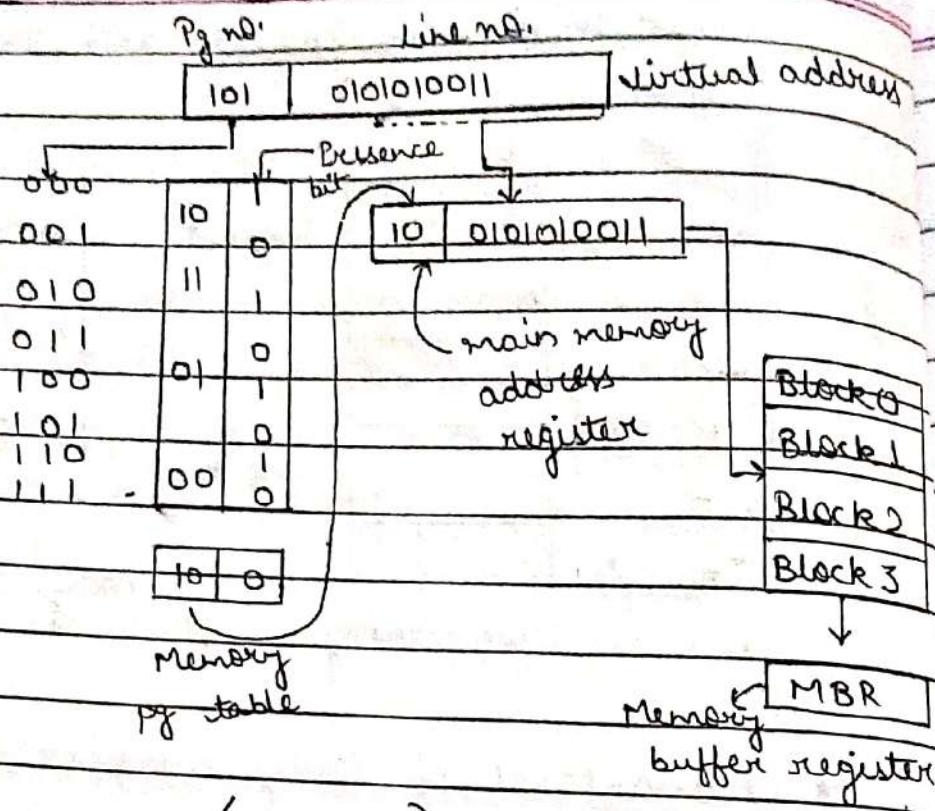


* Advantages of Lazy Swapper method :-

- 1- It reduces memory burden.
- 2- Swap time is reduced.
- 3- Increases degree of multiprogramming.

Disadvantage :-

Page fault may occur.



(Mapping)

The memory page table consist of 8 words.
 The table address denotes the page no.
 The content of words give the block no.
 where the page is stored in main memory.

Presence bit:- Content of word at this address is read out into the MTBR (Memory table buffer register) along with the presence bit.

If presence bit = 1 the contents are read and transferred to main memory.

* Page replacements:- For efficient utilization of main memory, page replacement is



used along with virtual memory technique.

- a) When new page is brought to main memory and if it is requested to remove a particular page from main memory then it is replaced by the another page.
- b) Where the page is placed in main memory.
- c) When the page is brought (send) outside of main memory to auxiliary memory.
- When the program starts execution some pages are transferred to main memory.
- The page table indicates the position of pages into main memory.
- When an attempt is made to refer a page that is in auxiliary memory page fault occurs, execution of current program is suspended until page is brought into main memory, which is I/O operation hence task is assigned to I/O processor.

* Replacement Techniques :-

1) FIFO (First in First out) -

- Oldest page is chosen, when page is

Data
Page

loaded to memory and Id number is pushed into FIFO step.

- The page whose Id number is at top of FIFO step is remove.

2.) LRU (Least Recently used pages) -

- Page that has not been used for longest period of time is replaced.
- It uses counter, at regular time interval, it is incremented by 1.

UNIT -4

Sequential Building Blocks

* Flip-flop:- A flip-flop is a binary cell which can store a bit of information. It itself is a sequential circuit. A flip-flop maintains anyone of the two stable states which can be treated as 0 or 1, depending on presence and absence of output signal.

The difference among various type of flip-flop is in the number of input they posses and in the manner in which the input affects the binary state. A flip-flop can be constructed using two NAND or NOR gates.

* Types of flip-flop:-

- 1- Clock SR flip-flop
- 2- J.K. flip-flop
- 3- T flip-flop
- 4- A flip-flop.
- 5- Master slave flip-flop

1- Clock SR flip-flop - It is an important circuit because other flip-flops are constructed from it. When the clock input is low the state of the latch is not change even if the data input changes.

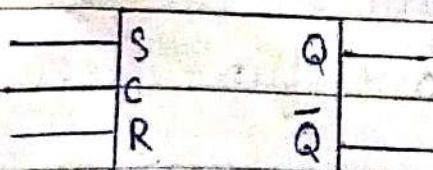
A SR flip-flop has additional two NAND gates with 3 inputs which are labelled S (for set condition), R (for reset condition) C (for clock condition).

The enable input C determines when the S and R input become effective. The logic diagram with function table of SR flip-flop is given below -

Function-table -

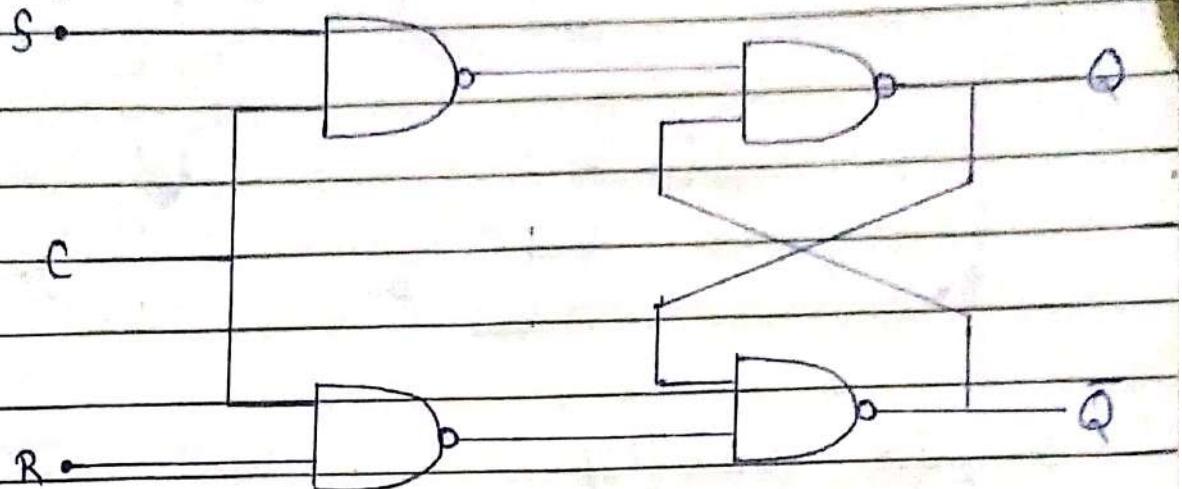
C	R	S	Q	\bar{Q}
0	x	x	No change	
1	0	0	No change	
1	0	1	0	1
1	1	0	1	0
1	1	1	Undefined	

Symbol of clock SR flip-flop -



*Basha
Page*

Logic diagram -

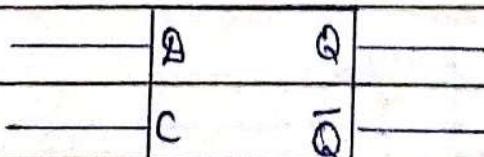


2- D flip-flop = A flip-flop is a modified version of SR. We can convert a SR flip-flop to a D flip-flop by inserting an inverter between S and R.

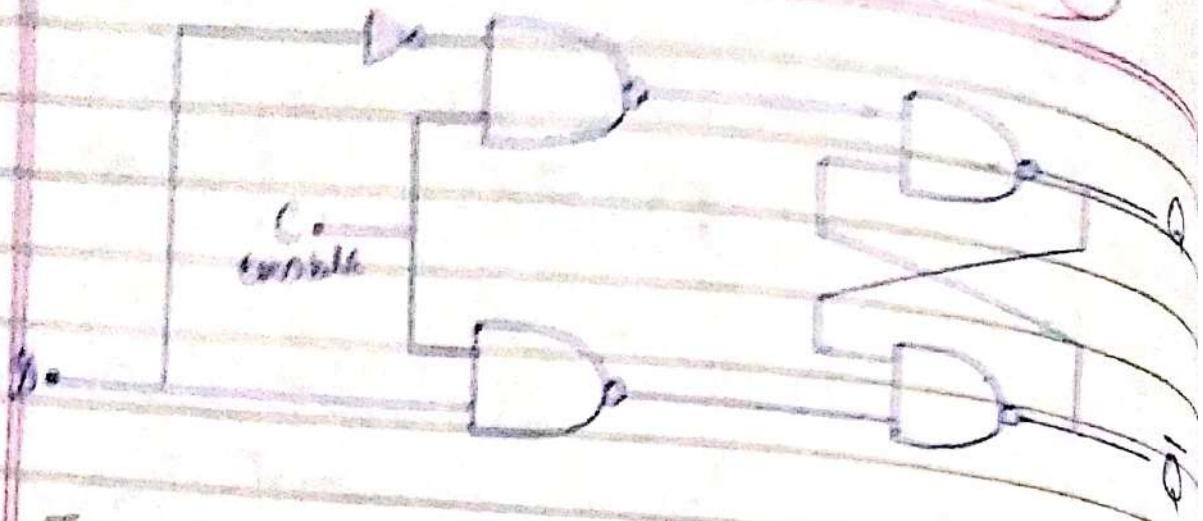
Function-table -

C	A	Q
0	X	No change.
1	0	0 (reset)
1	1	1 (set)

Symbol -



Logic diagram -

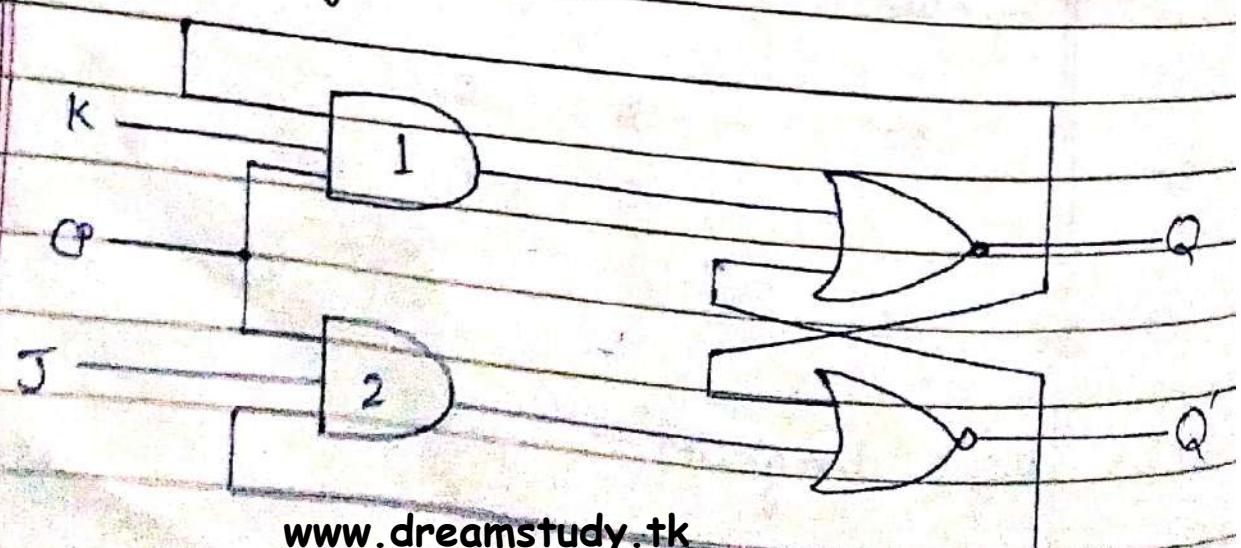


3. JK flip-flop :-

Truth table -

Q	J	K	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Logic diagram -

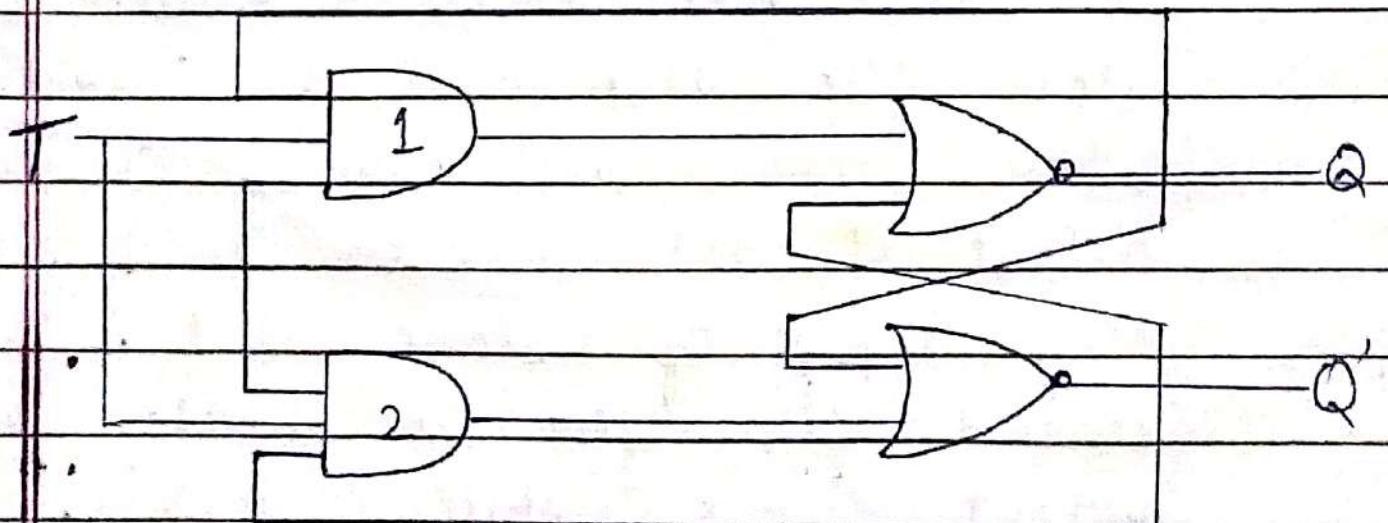


4- T flip-flop:-

Truth table -

Q	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

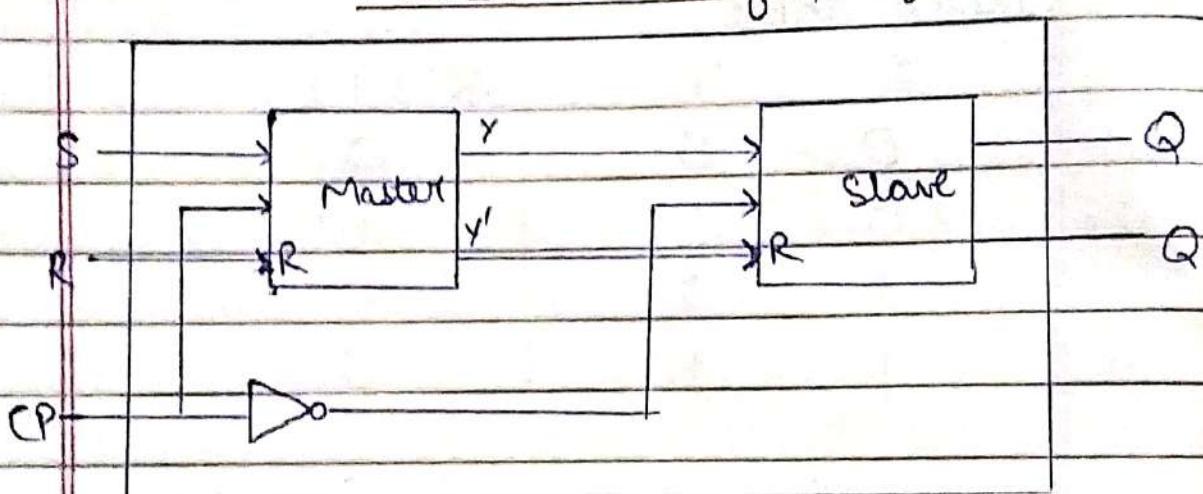
Logic diagram -



T.P

5- Master-slave flip-flop:- This flip-flop is constructed from two separate flip-flops. One circuit serve as a master and other as a slave. and the overall circuit is called master-slave flip-flop.

Master-slave flip-flop



This flip-flop consist a master flip-flop, slave flip-flop and an inverter.

When the clock pulse is 0 the output of the inverter is 1. So the clock input of slave is 1. The second flip-flop is enable and output $Q = Y$ while $Q' = Y'$.

Master flip-flop is disable because $CP = 0$. The information at R and S is transmitted to the master flip-flop when pulse becomes 1.



* Register :- A register is a device capable of storing a bit. The data can be serial or parallel. A register can convert a data from serial to parallel and vice-versa. Shifting the digits to left and right is an important aspect of arithmetic operation.



END

Hope the study material was helpful , to stay connected with us : Visit Us

www.dreamstudy.tk



<https://www.facebook.com/dreamstudy>



<https://www.instagram.com/dream.study/>



<https://www.youtube.com/channel/UC8I0Dfy2YekfiigaXEtBAow> or
search on youtube with - dreamstudy website



https://twitter.com/DreamStudy_



www.dreamstudy.tk