Computer System= Hardware+ Software

Hardware:-Physical Components (We can touch and seen).

Software:-Logical Components (We can seen and do work something according to object/goal).

Computer Organization:-

What Is Computer:-

↓

Making for Calculations as well as measurement

↓

Based on Arithmetical and Logical.

↓

Arithmetic/Numeric Operation (+, -, *, /, ^, % (modulo operator/Remainder Operator)).

&

Logical Operation (True, False, Yes, No).

General Definition 1:-

It (Computer) can perform only those operations or calculations (Arithmetical and Logical), Measurement and controlling functions, which can be expressed there, result in terms of numerical or logical.

Program:-

The basic function/task/work of computer is the execution of program.

It is sequence of instructions, which operate on computer data to perform certain well-defined task or achieve a goal.

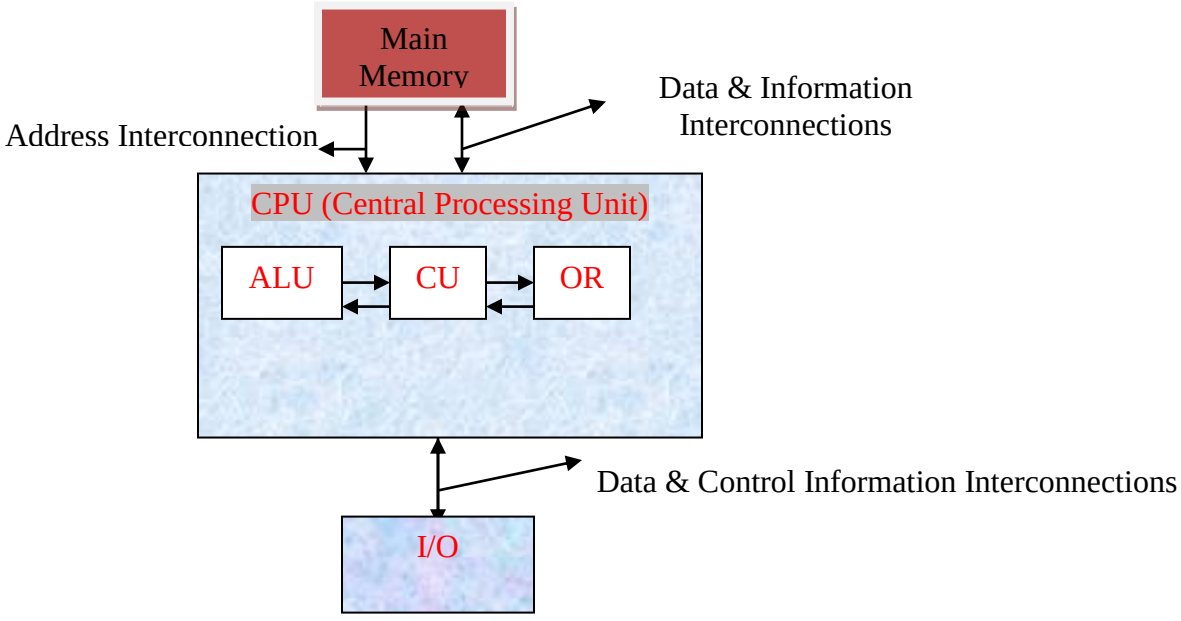
Data:- It is facts and figure that is represented in the form of 0 and 1.It is known as Bits(Binary Digits).

Modern Definition 2:-

It is an automatic electronic apparatus/Machine making for controlling operations and performing arithmetical and logical operations and can also perform measurement. Which can be expressed there result in terms of numerical or logical.

Structure of Computer

Structure of computer was given by **Von Neumann**. He was a British mathematician



Input Gives→ instructions & data after Ouput→Produce Result (Numerical or Logical)

CPU:-It is the **brain** of computer which consist of following three components

- ALU:- By This unit it can perform all types of arithmetical and logical operations
- CU:- It is used for controlling operations inside CPU.It accept signal from out side & produce their corresponding signal. It is the heart of computer.
- OR:- It is a special type of temporary storage area where actual processing is to be performed by CPU. The size of register determines processing speed of CPU.Processing speed measured in MIPS. (Million Instruction Per Seconds) as well as hertz unit.

ALU :-Arithmetic & Logic unit.
CU :-Control Unit.
OR :-Operational Register.

Main Memory:-

It is needed in a computer to store instructions and the data at the time of program execution.
Computer data represented in the form of 0 and 1.It is known as bits (Binary digits).
0 Off/false/No.
1 On/true/Yes.

1	0	1	0	0	1	0	1

8 bits=One Byte.

One Byte=One Character. Example:-ICSM=4 Bytes=32 4*8=Bits.

1 Kilo Byte (KB) =1024 Bytes=2^{10} Bytes.

1 Mega Byte (MB) =1024 *1024Bytes=2^{20} Bytes.

1 Giga Byte (GB) =1024*1024*1024 Bytes=2^{30} Bytes.

1 Tera Byte (TB) =1024*1024*1024 *1024 Bytes=2^{40} Bytes.

1 Peta Byte (PB) =1024*1024*1024 *1024* 1024Bytes=2^{50} Bytes.

1 Exabyte (EB) = 2^{60} Bytes.

1 Zettabyte (ZB) = 2^{70} Bytes.

1 Yottabyte (YB) = 2^{80} Bytes.

Notes:-Bits was invented by Lady ada.She had written first program of computer.

I/O Devices:- (Input/Output Devices)

Input ⟶ Processing (CPU) ⟶ OUTPUT

Input Devices(Instructions and Data)

- Keyboard
  i. Cherry Keyboard (Costly and repairable).
  ii. Membranes Keyboard (Cheaper & non repairable).
- Mouse
  i. Trackball Mouse.
  ii. Optical Mouse.(Better Quality).
- Joystick. (It is used for playing game).
- Light pen.(It is used for drawing on screen directly).
- OMR. (Optical Mark Recognition/Reader).

Example:-

A B C D
○ ○ ● ○

- OBR (Optical Bar Reader). It is used for identification of items. It interprets pencil marks on papers.

- MICR (Magnetic Ink Character recognition/Reader).→It is used in banking industry.
- Scanner:-It is used for scanning documents, graphics and images in digital form.
- Voice Speech Synthesizer:-It is used for recognizing audio sound/voice.(Used in cockpit of airlines).
- Mike.

Etc.

Output Devices:-

1. **VDU(Visual Display Unit)Produce Soft Copy/Electronic mode copy**
   a. CRT Screen (Cathode Ray Tube) **Pixels** (.) are fundamental elements of images.
   b. LCD Screen (Liquid Crystal Display) **Crystal rods** are used for creating graphical object on screen.
   c. LED Screen (Light Emitting Diode).
   d. PLASMA Screen.

2. **Printer(Produce Hard Copy).**
   a. **Impact Printer (Inked ribbon is used).**
      i. Dot Matrix Printer (DMP). (It prints only mono /single color).
      ii. Daisy Wheel Printer.
      iii. Drum Printer.
      iv. Line Printer.
   b. **Non Impact Printer(Used chemical for printing).**
      i. Inkjet printer (Cartridge is used).It print both types color and Mono color).
      ii. Laser Printer (Toner is used)It is the best printer in quality and speed.

Printing Speed measurement:-

- CPS (Character per Second).
- PPM (Page per Minute).

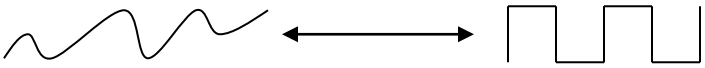
3. Plotter.
   It is used by architect engineer for graphical output on paper/Flex.
4. Speeker(It produces Sound).

Etc.

Both Input/Output devices:-

- MODEM (Modulator & Demodulator).
  It is used for internet connection.

Analog Signal ⟷ Digital Signal.



Analog Signal (Sine wave) 0 1 0 1 0 Digital Signal (Bit Trains).

- Touch Screen.
- Communication Port.

Characteristics of Computer:-

- Speed.
- Accuracy.
- Memory.
- High Remembering power.
- Deligency.
- No Intelligency.
- Emotionless.
- Feeling less.
- Versatility.

Types of Computer:-

- Analog Computer (Consist of Analog Signal).
- Digital Computer (Consist of Digital Signal).

- Hybrid Computer (Consist of both types of signal).

Analog Computer:-

Such types of computers are used for measuring temperature, pressure, speed etc.

Example:-

Thermometer :- It is used for measuring temperature.
Speedometer :- It is used for measuring speed.
Barometer :- It is used for measuring pressure of air.

Digital Computer→ Classification Of computer:-

- Micro Computer.
- Mini Computer.
- Mainframe Computer.
- Super Computer.

Micro Computer:-

It is small computer, which is used for personal work.

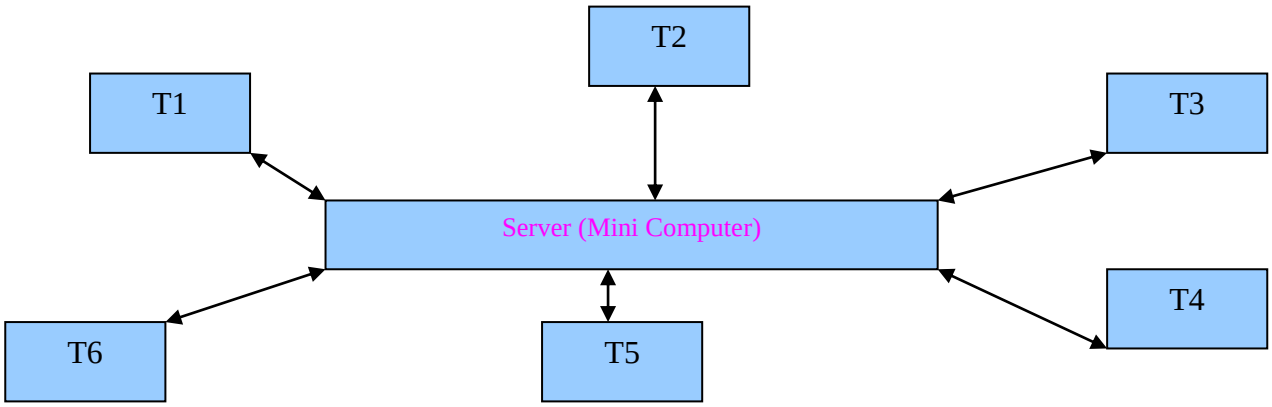
Example:-

- PC (Personal Computer)/Desktop PC.
- Laptop.
- Tablet.
- Palmtop.

Etc.

Mini Computer:-

It is larger than microcomputer which is used for small networking purpose.It may support 30 to 50 Terminals simultaneously. Example:-PDP-8,Wi-Fi(Wire -Fidelity), Small networking system in bank.



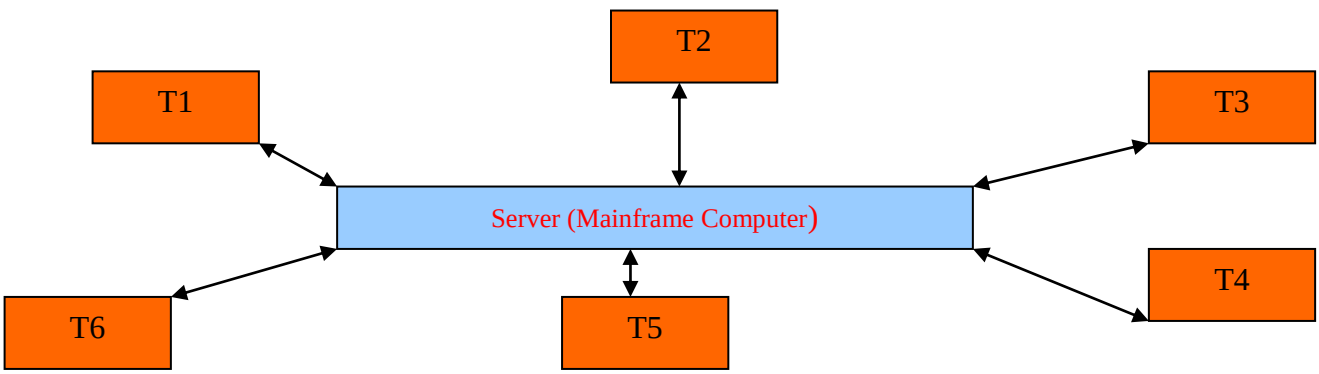
Mainframe Computer:-

It is also used for networking purpose. It is suited for big organization.

It may support above 500 terminals simultaneously.

Example:-

MEDHA.
DEC.
SPERRY etc.



Installation Arrangements:-

- ✓ Arrangements of Power.
- ✓ Arrangement of Special room and environment.
- ✓ Cost of Mainframes exists between 40000$ to 1 Million dollar.
- ✓ Arrangement of LAN for making large network.
- ✓ Arrangement of computer professionals for working on terminal.

Super Computer:-

It is the fastest computing device, which is used for solving complex problems.
There are many CPU are used in Super Computer.

Example:-

PARAM PADMA.
CORAY.
INDIGINIOUS, Hitachi, EKKA.
Etc.

Application of Super Computer:-

- ❖ Airlines Controlling.
- ❖ Weather forecasting.
- ❖ Shuttle space controlling. (Atalantice, Colambia).
- ❖ Medical Science.
- ❖ Satellite controlling.

Etc

Hybrid Computer:-

It is made by taking the best features of the analog computer and digital computer.
It is used in Hospital, Nuclear controlling system, Hydrogenic System.

Number System (Computer Data Representation Technique):-

- ✓ **Binary Number System**
  - 0 and 1 Base /Radix=2.
- ✓ **Octal Number System**
  - Numbers 0, 1,2,3,4,5,6,7 Base/Radix=8.
- ✓ **Decimal Number System**
  - Numbers 0,1,2,3,4,5,6,7,8,9 Base/Radix=10.
- ✓ **Hexadecimal Number System**
  - Numbers 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 Base/Radix=16/H.
  - 10=A,11=B,12=C,13=D,14=E,15=F

Conversion:-

Decimal To Binary Numbers System

- ✓ Division Method
- ✓ Tabular Method

… 2048 1024 512 256 128 64 32 16 8 4 2 1

Division Method:-

Example $(123)_{10}= (?)_2$

2	123	Remainder
2	61	1
2	30	1
2	15	0
2	7	1
2	3	1
2	1	1
2	0	1

$(123)_{10}= (1111011)_2$

Example $(209.45)_{10}= (?)_2$

2	209	Remainder
2	104	1
2	52	0
2	26	0
2	13	0
2	6	1
2	3	0
2	1	1
2	0	1

.45*2 =0.90 = 0
.90*2 =1.80 = 1
.80*2 =1.60 = 1
.60*2 =1.20 = 1

$(209.45)_{10}= (11010001.0111)_2$

Tabular Method

… 2048 1024 512 256 128 64 32 16 8 4 2 1

Example $(123)_{10}=(?)_2$

128	64	32	16	8	4	2	1
0	1	1	1	1	0	1	1

$(123)_{10}$ =$(1111011)_2$

Example $(17)_{10}=(?)_2$

16	8	4	2	1
1	0	0	0	1

$(17)_{10}$ =$(10001)_2$

Example $(27)_{10}=(?)_2$

16	8	4	2	1
1	1	0	1	1

$(27)_{10}= (11011)_2$

Example $(81)_{10}=(?)_2$

64	32	16	8	4	2	1
1	**0**	**1**	**0**	**0**	**0**	**1**

$(81)_{10}=(1010001)_2$

Example $(13)_{10}$= $(?)_2$

8 4 2 1

1 1 0 1

$(13)_{10}$= $(1101)_2$

Example $(9)_{10}$=$(?)_2$

8 4 2 1

1 0 0 1

$(9)_{10}$=$(1001)_2$

Example $(2)_{10}$=$(?)_2$

2 1

1 0 **$(2)_{10}$=$(10)_2$**

Binary To Decimal:-

Example:- $(1111011)_2$=$(?)_{10}$

$1*2^6+1*2^5+1*2^4+1*2^3+0*2^2+1*2^1+1*2^0$

= 64 +32 + 16 + 8 +0 +2 + 1

= $(123)_{10}$

Binary To Decimal:-

Example:- $(1111011.101)_2$=$(?)_{10}$

$1*2^6+1*2^5+1*2^4+1*2^3+0*2^2+1*2^1+1*2^0+1*2^{-1}+0*2^{-2}+1*2^{-3}$

= 64 +32 + 16 + 8 +0 +2 + 1 + .5 +0 + .125

= $(123.625)_{10}$

Binary To Decimal(Using Tabular Method):-

Example:- $(1111011)_2$=$(?)_{10}$

64 32 16 8 4 2 1

1 1 1 1 0 1 1

$64+32+16+8+0+2+1=(123)_{10}$

Binary Arithmetic Operations:-

- ✓ Addition.(+)
- ✓ Subtraction.(-)
- ✓ Multiplication.(*)
- ✓ Division.(/)

Addition:-

Example:-1 …16 8 4 2 1

111101

+ 111111

1111100 Carry=

Example:-2

111101

111101

+ 111111

10111001 Carry= 101

Subtraction/Minus:-

0-0 = 0 Borrow=0

1-0 = 1 Borrow=0

1-1 = 0 Borrow=0
0-1 = 1 Borrow=1

Example1:-

```
  111111
- 111110
--------
  0 000 01
--------
```

Example2:-

```
    1  1 0 0 1 0
  - 1  1 1 1 1 1
  ----------------
    1 1 1 0 01 1        B=1
  ----------------
  Negative result
```

Multiplication:-

```
      1 1 1. 1 1
        *111
  ---------------
      1 1 1 1 1
    1 1 1 1 1
  1 1 1 1 1
  ---------------
1 1 0 1 1 0 .0 1
  ---------------
```

Division:-

$(1101)_2/(11)_2=(?)_2$

```
11 | 1 1 0 1          | 1 0.0101     Quotient
   | 1 1              |
   |------------------|
   | 0 00100          |
   |      11          |
   |------------------|
   |    0 0 100       |
   |        11        |
   |------------------|
   |      0 0 001     | Remainder
```

Decimal to Octal:-

Example $(123)_{10}=(?)_8$

8	123	Remainder ▲
	15	3
	1	7
	0	1

$= (123)_{10}= (173)_8$

Example $(523)_{10}= (?)_8$

8	523	Remainder
8	6 5	3
8	8	1
8	1	0
8	0	1

$= (523)_{10}= (1013)_8$

Example $(503.66)_{10}= (?)_8$

8	503	Remainder
8	6 2	7
8	7	6
8	0	7

.66*8 =5.28 =5
.28*8 =2.24 =2
.24*8 =1.92 =1
.92*8 =7.36 =7

$= (503.66)_{10}=(767.5217)_8$

Octal to Decimal:-

Example:-1

$(1013)_8=(?)_{10}$
$1*8^3+0*8^2+1*8^1+3*8^0$
512 + 0 + 8 + 3
$= (523)_{10}$

Example:-2

$(313.35)_8=(?)_{10}$
$3*8^2+1*8^1+3*8^0+3*8^{-1}+5*8^{-2}$
192 + 8 + 3 + .375 + .0789
$=(203.4539)_{10}$

Binary To Octal :-

Example

$(11111111.110001)_2$

01 1 1 1 1 1 1 1. 1 1 0 0 01

011=3
111=7
111=7
110=6
001=1
$(377.61)_8$

Decimal To HexaDecimal :-

Example $(503)_{10}=(?)_{16}$

16	503	Remainder
16	31	7
16	1	15=F
16	0	1

$=(1F7)_{16}$

Example $(523.45)_{10}=(?)_{16}$

16	523	Remainder
16	32	11=B
16	2	0
16	0	2

.45*16 =7.20 =7
.2*16 =3.2 =3
.2*16 =3.2 =3

$=(20B.733)_{16}$

HexaDecimal To Decimal :-

$(1F7)_{16}=(?)_{10}$
$=1*16^2+F*16^1+7*16^0$
$=256+15*16+7$
$=256+240+7$
$=256+247$
$=(503)_{10}$

Binary to HexaDecimal:-

$(11111111.110001)_2$

1 1 1 1 1 1 1 1. 1 1 0 0 0 1 0 0

1 1 1 1=15=F
1 1 1 1=15=F
1 1 0 0=12=C
0 1 0 0=4
$(FF.C4)_{16}$

Compliments:-

1's Compliment.
2's Compliment.

Example:-1
1's Compliment= 1 → 0
0 → 1
$(10001)_2$ → 01110
1's compliment

Example:-2
2's Compliment=1's Compliment +1
$(10001)_2$ → 01110 +1=01111

1's compliment 2's compliment

Sign Magnitude:-

+	0
-	1

Example:-

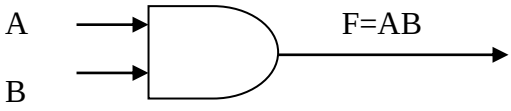
$(-17)_{10}$ → +17 → 010001 → 101110 (1's) → 101110+1=101111 (2's)

Logic Gates:-

It is used for making/Designing digital circuit. There are many types of gates.

- ✓ AND Gate.
- ✓ OR Gate.
- ✓ NOT Gate.
- ✓ NAND Gate.
- ✓ NOR Gate.
- ✓ EX-OR Gate.
- ✓ EX-NOR Gate.

AND GATE:-



TRUTH TABLE:-

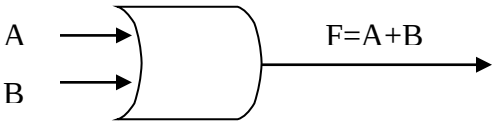
Let n=input variable

Input Signal=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	F=AB
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE:-



TRUTH TABLE:-

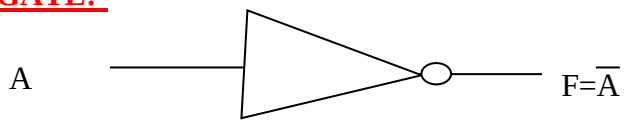
Let n=input variable

Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE:-



TRUTH TABLE:-

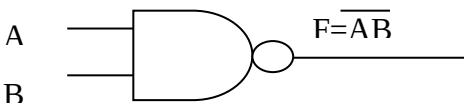
Let n=input variable

Input variable=2^n

n=1(A)= 2^1=2(0,1)

A	F=$\overline{A}$
0	1
1	0

NAND/NOT-AND GATE:-



TRUTH TABLE:-

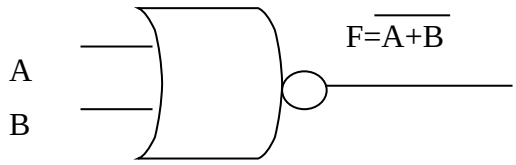
Let n=input variable

Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	F1=AB	F=$\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

NOR GATE:-



TRUTH TABLE:-

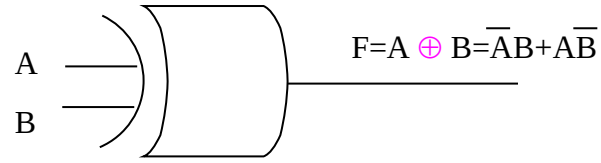
Let n=input variable

Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	F1=A+B	F=$\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

EX-OR GATE:-



TRUTH TABLE:-

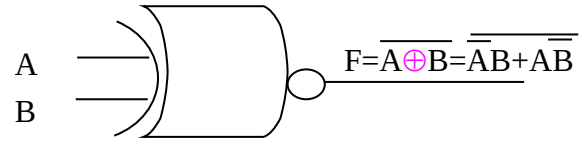
Let n=input variable

Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}$B	A$\overline{B}$	$\overline{A}$B+A$\overline{B}$
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

EX-NOR GATE:-



TRUTH TABLE:-

Let n=input variable

Input variable=2^n

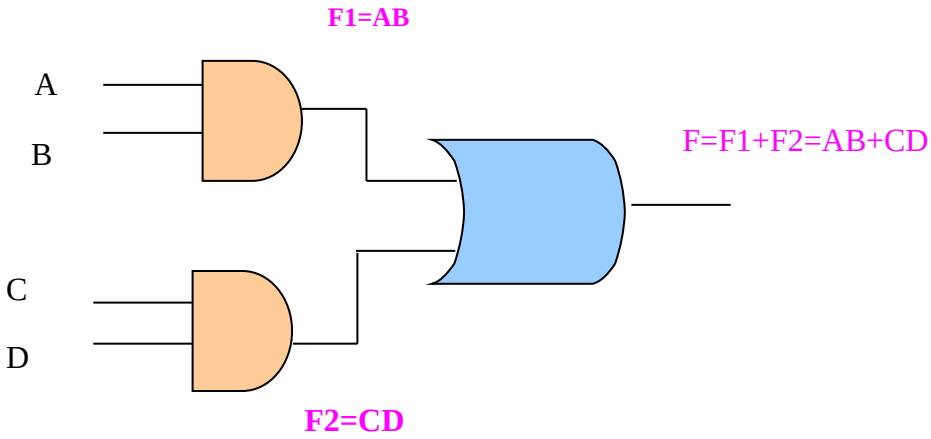
n=2(A,B)= 2^2=4(0,1,2,3)

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}B$	$A\overline{B}$	$\overline{A}B+A\overline{B}$	$\overline{\overline{A}B+A\overline{B}}$
0	0	1	1	0	0	0	1
0	1	1	0	1	0	1	0
1	0	0	1	0	1	1	0
1	1	0	0	0	0	0	1

Combinational Circuit:-

F=AB+CD is a Boolean Expression

Where A, B, C, D are called Boolean variables.

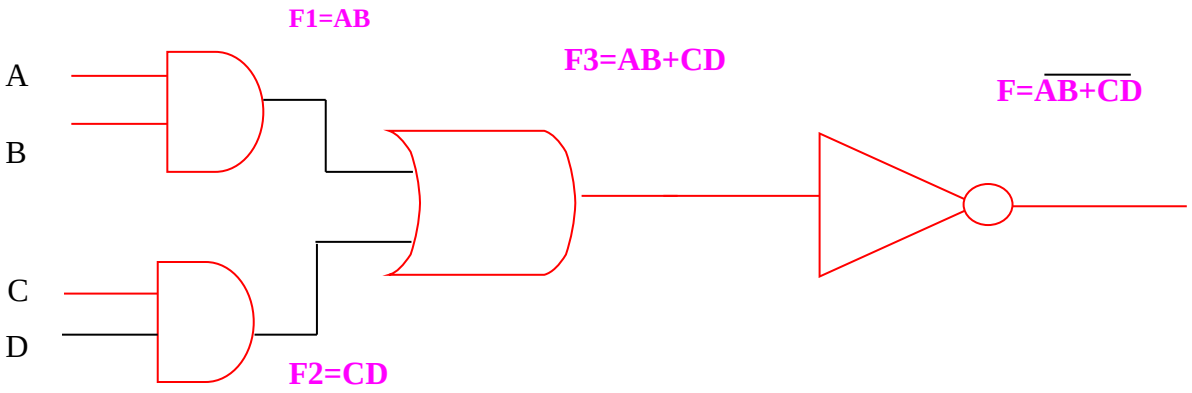


16 16 8 4 2 1

Truth Table

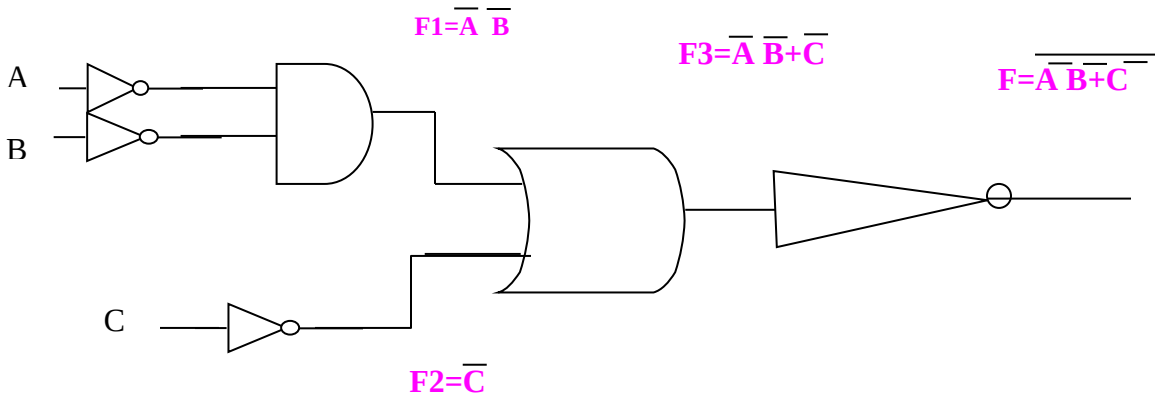
A	B	C	D	AB	CD	AB+CD
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

Example 2 :- $F=\overline{AB+CD}$

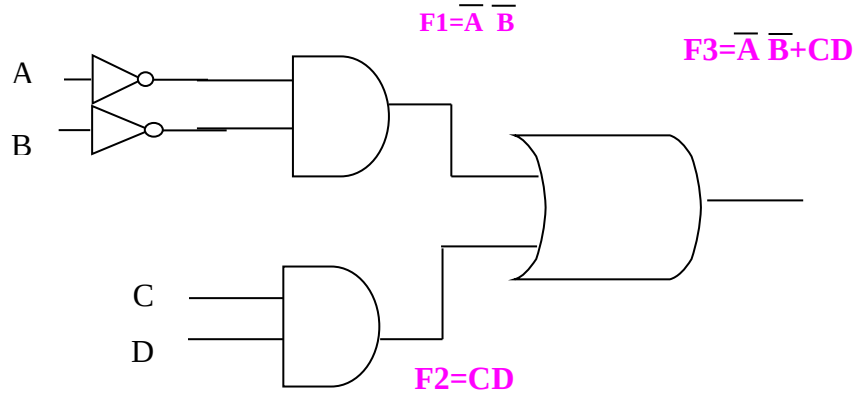


A	B	C	D	AB	CD	AB+CD	$\overline{AB+CD}$
0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	1
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1
1	0	1	0	0	0	0	1
1	0	1	1	0	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	0
1	1	1	1	1	1	1	0

Example 3 :- $F=\overline{\overline{A}\,\overline{B}+\overline{C}}$ Design circuit and truth table.



Example 4 :- $F=\overline{A}\,\overline{B}+CD$



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

Boolean algebra:-

Simplification Method of Boolean functions:-

- ✓ Algebraic method.
- ✓ Karnaugh maps (K-Map)
- ✓ Quine McClusky Method

Algebraic method.

Theorem based on algebraic method:-

- ✓ Commutative law
  - A+B=B+A
  - A.B=B.A
- ✓ Associative law
  - A+(B+C)=(A+B)+C
  - A.(B.C)=(A.B).C
- ✓ Distributive law
  - A+(B.C)=(A+B).(A+C)
  - A.(B+C)=(A.B)+(A.C)
- ✓ Absorption Law
  - A+AB=A
- ✓ Identity law

A+0=A

A.1=A

- ✓ Inverse law

$A\bar{A}=0$

$A+\bar{A}=1$

- ✓ Demorgan's law

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

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

- ✓ Double compliment

$\overline{\overline{A}}=A$

Simplify the Following Boolean functions:-

Question 1:- A+1=1

Solution:-

$A+A+\bar{A}$

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

$=A+\bar{A}$

$=1$

Question 2:-

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

Solution:-

$=\overline{(\overline{A}+\overline{B})}.\overline{(A+\overline{B})}$ {By Using demargons law}

$=(\overline{A}+\overline{B}).(A+\overline{B})$ {By Using double compliment method}

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

$=0+\overline{B}(\overline{A}+A)+\overline{B}$

$=0+\overline{B}+\overline{B}$

$=\overline{B}$

Home Work:-

1. A * A = A
2. Simplify the Boolean expression (A+B+C)(D+E)' + (A+B+C)(D+E)
3. Simplify the Boolean property x + x'y = x + y

Karnaugh maps (K-Map):-

Karnaugh maps (K-Map):-

A Karnaugh map (K-map) is a pictorial method used to minimize Boolean expressions without having to use Boolean algebra theorems and equation manipulations. A K-map can be thought of as a special version of a truth table . Using a K-map, expressions with two to four variables are easily minimized.

Canonical and Standard Forms:-

- ✓ **SOP(Sum of product)** Σ It is called Minterm or Standard Sum
- ✓ **POS(product of Sum)** ∏ It is called Maxterm or standard Product

Σ SOP (Sum of product) For Two Boolean variables:-

A	B	Σ	Term
0	**0**	$\bar{A}\,\bar{B}$	M_0
0	**1**	$\bar{A}B$	M_1
1	**0**	$A\bar{B}$	M_2
1	**1**	AB	M_3

Example1 F= Σ(0,1,2,3,4,8,12) Example2 F= Σ(0,1,2,3,4,9,19)

∏ POS(product of Sum):-

A	B	∏	Term
0	**0**	**A+B**	**M_0**
0	**1**	**A+$\overline{B}$**	**M_1**
1	**0**	**$\overline{A}$+B**	**M_2**
1	**1**	**$\overline{A}$+$\overline{B}$**	**M_3**

Example1 F= Π(0,1,2,3,4,8,12) Example2 F= Π (0,1,2,3,4,9,19)

Σ SOP (Sum of product) For Three Boolean variables:-

A	B	C	Σ	Term
0	0	0	$\bar{A}\bar{B}\bar{C}$	M_0
0	0	1	$\bar{A}\bar{B}C$	M_1
0	1	0	$\bar{A}B\bar{C}$	M_2
0	1	1	$\bar{A}BC$	M_3
1	0	0	$A\bar{B}\bar{C}$	M_4
1	0	1	$A\bar{B}C$	M_5
1	1	0	$AB\bar{C}$	M_6
1	1	1	ABC	M_7

∏ POS (Product of Sum) For Three Boolean variables:-

A	B	C	∏	Term
0	0	0	$A+B+C$	M_0
0	0	1	$A+B+\bar{C}$	M_1
0	1	0	$A+\bar{B}+C$	M_2
0	1	1	$A+\bar{B}+\bar{C}$	M_3
1	0	0	$\bar{A}+B+C$	M_4
1	0	1	$\bar{A}+B+\bar{C}$	M_5
1	1	0	$\bar{A}+\bar{B}+C$	M_6
1	1	1	$\bar{A}+\bar{B}+\bar{C}$	M_7

Question:-

Make a table for min term and max term of following Boolean functions and also simplify these.

F= Σ (0, 1, 5, 8, 9, 12)

F=∏ (0, 1, 5, 8, 10, 14)

For Min term Σ

		A	B	C	D
M_0	=	0	0	0	0
M_1	=	0	0	0	1
M_5	=	0	1	0	1
M_8	=	1	0	0	0
M_9	=	1	0	0	1
M_{12}	=	1	1	0	0

For Max term ∏:-

		A	B	C	D
M_0	=	0	0	0	0
M_1	=	0	0	0	1
M_5	=	0	1	0	1
M_8	=	1	0	0	0
M_{10}	=	1	0	1	0
M_{14}	=	1	1	1	0

Minimization of Gates By Using K-MAP:-

Table for Two Variables (Say A and B):-

	$\overline{A}$	A
$\overline{B}$	0	1
B	2	3

Parity bits:-

It is an additional bits, Which is used for finding and correcting errors in system. There are two types of parity bits.

- ✓ Odd parity bit
- ✓ Even parity bit

Example:-

Find out odd and even parity bit of following numbers.

$(1000101)_2$

Solution:-

Odd parity bits of 1=0
Even parity bits of 1=1
Odd parity bits of 0=1
Even parity bits of 0=0

Example:-

Find out odd and even parity bit of following numbers.

$(00001)_2$

Solution:-

Odd parity bits of 1= 0
Even parity bits of 1= 1
Odd parity bits of 0= 1
Even parity bits of 0= 0

Coding System:-

There are following coding system uses in computer.

BCD → (Binary Coded Decimal)
EBCDIC → (Extended Binary Coded Decimal Interchange Code)
ASCII → (American Standard Code for Information Interchange)

BCD(Binary Coded Decimal):-

It is four bits of coding system. It is also known as 8-4-2-1.

d1	d2	d3	d4

Weight of $d1=2^3=8$.
Weight of $d2=2^2=4$.
Weight of $d3=2^1=2$.
Weight of $d4=2^0=1$.
One nibble= ½ of a byte.

EBCDIC (Extended Binary Coded Decimal Interchange):-

It is eight bits of coding system, which is used in only mainframe computer system.

d1	d2	d3	d4	d5	d6	d7	d8

ASCII (American Standard Code For Information Interchange)

(Or Alpha Numeric Codes)

It is also eight bits of coding system. It is used in microcomputer system.
Its value exist between 0 to 255.There are two categories of ASCII codes.

a:-Standard ASCII codes → (0-127)

Non Printing ASCII Codes	0-31
Printing ASCII Codes	32-127

b:-Extended ASCII codes 128-255

It is used for making symbols.

Example:-

65-90	A-Z
97-122	a-z

Alt+ASCII codes (Numeric Key Pads)

History of Computer:-

History of Computer:-

Computer developed before 300BC at China. This computing device was known as **abacus.**

- Mechanical Computer.
- Electromechanical computer.
- Electronic Computer.

Mechanical Computer:-

It was very first attempt towards automatic computing device. **Blaise Pascal** designed a mechanical computer that was known as **Pascaline** .It consist of a lot of gears and chains. It can perform repeated addition and subtraction. After some time the grand father (**Charles Babbage**) of modern computer also designed following two mechanical computer.

- Difference Engine.
- Analytical engine by Babbage.

Above both kinds of mechanical computer can perform repeated addition, subtraction, multiply, division, solving polynomial equation and trigonometric problems.

Blaise Pascal→Date of Birth	19 June 1623→ France	Date of Died	19 August 1662→Paris
Charles Babbage→ Date of Birth	26 Dec 1791→ London	Date of Died	18 October1871→London

Electromechanical computer.

It was the next attempt towards automatic electronic apparatus that was afforded by Howard Aiken at Harvard University. He designed first electromechanical computer that was known as **MARK-I**. It was sponsored by IBM and UN Navy.

Date of Birth	8 March 1900 →	Hoboken New Jersey		Died	14 March 1973

Electronic computer:-

Generation:-

- First generation → (1942 to 1955).
- Second Generation → (1955 to 1964).
- Third Generation → (1964 to 1975).
- Fourth Generation → (1975 to 1990).
- Fifth generation → (yet to come).

First generation:-

In this generation vacuum tube was used in processor technology. There are many types of **vacuum tubes**.

- Diode(Cathode & Anode Plate)
- Triode(Cathode , Anode Plate & Suppressor Grid).
- Tetrode(Four plates).
- Pentode(Five plates).

Using vacuum tubes following types of electronic computers was designed.

- ENIAC (Electronic Numerical Integrator and calculator).
- EDVAC (Electronic Discrete Variable automatic Computer).
- UNIVAC (Universal Automatic Computer).

Characteristics of ENIAC:-

- It was Very Giant Machine.
- Its weight was 30 tones.
- It may perform 5000 addition or 500 multiplications per minute.

- It occupies a number of rooms.
- It needed a lot of electricity.
- It needed a lot of cooling requirements.
- It emitted a lot of heat.
- Portability is very complex task.
- There were 18000 vacuum tubes were used.

Second generation:-

In this generation vacuum tube was replaced by **transistor**. By using transistor IBM (International business Machine) designed **IBM 700** series of processors. It consumes less electricity and cooling requirement system. Its speed was faster than first generation computer.

Example of transistor:-

n-p-n	Transistor.
p-n-p	Transistor.
n-p	Transistor.
p-n	Transistor.

Transistor based processor much faster than vaccum-based processor.

Third generation:-

In this generation, integration technology known as ***IC*** (Integrated Circuit) was used. It consists of a lot of transistor, capacitor and other electronics components.

IC Technology:-

SSI	(Small Scale Integrator).
MSI	(Medium Scale Integrator).
LSI	(Large Scale Integrator).
VLSI	(Very Large Scale Integrator).
ULSI	(Ultra Large Scale Integrator).

SSI	Number of gates below 100.
MSI	Number of gates upto 100 or more.
LSI	Number of gates upto 1000.
VLSI	Number of gates upto 10^6(Chips).
ULSI	Number of gates upto 10^8(Micro Chips).

Fourth generation:-

In this generation, **VLSI** based technology used in processor (CPU) used.

Feature:-

- Smaller.
- Faster.
- GUI (Graphic User Interface) .

Fifth generation:-

In this generation, **ULSI** based technology used in processor used.

Example:-

- ROBOT.
- Fighter Plane.
- Shuttle Space Plane (Its speed=40000KM/Hour).

Etc.

Intel Company designed first microprocessor 4004 in 1971 .It was specific purpose processor.

Intel Company designed second microprocessor 8088 in 1974 .It was general purpose processor.

Classification of computer(Digital Computer):-

- Micro computer.
- Minicomputer.
- Mainframe Computer.
- Super Computer.

Micro Computer:-

It is small computer, which is used for personal work.

Example:-

- PC
- Laptop
- Tablet
- Palmtop

etc.

Mini Computer:-

It is larger than microcomputer which is used for small networking purpose.

Mainframe Computer:-

It is also used for networking purpose. It is suited for big organization.

Example:-

MEDHA
DEC
SPERRY etc.

Super Computer:-

It is the fastest computing device, which is used for solving complex problems.

Example:-

PARAM
CORAY
INDIGINIOUS
Etc.

Application of Super Computer:-

- Airlines Controlling
- Weather forecasting
- Shuttle space controlling
- Medical Science

etc

Types of Computer:-

Analog Computer.
Digital Computer.
Hybrid Computer.

Analog Computer.

Such types of computers are used for measuring temperature, pressure, speed etc.

Example:-

Thermometer
Speedometer
Barometer

Digital Computer.

Classification Digital Computer:-

- Micro computer
- Minicomputer
- Mainframe Computer
- Super Computer

Hybrid Computer:-

32Such type of computers uses both categories of technology.

Example:-

- ECG Machines.
- Nuclear control System.
- Hydrogenic System

Uses of Computer in Digital devices:-

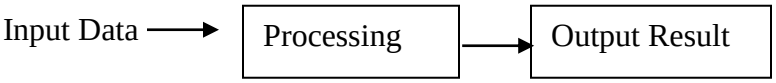
- In Mobiles

- In Bank
- In Music
- In Robotic

Computer System:-

- Hardware
  - Input Devices (Ex:-Keyboard,Mouse,Joystick etc)
  - Processor(CPU)
  - Output Devices(Ex Monitor,Speeker,Printer etc)
  - Storage Devices(Ex Hard Disk,CD ROM,DVD ,Floppy Disk)
- Software
  - System Software
    - Operating System Example :-MS Windows XP,VISTA etc
    - Application S/W Example :- MS word,MS excel,MS Power Point etc
    - Development S/w Example:- C++,C#,Java,COBOL,PASCA,BASIC etc

Working of Computer System:-



Process:-

Running state of program is called process.

Types of Processing:-

There are two types of processing in computer system

- On line processing/Real Time processing(A device directly connected with CPU).
- Off Line Processing/Batch Processing(A device not directly connected with CPU).

On line processing/Real Time processing:-

When processing takes place immediately as soon as the data is entered into the computer .It is called on line processing.

Off Line Processing/Batch Processing:-

When processing takes place on the stored data to get an output it is called **off line processing.**
Example:- Data Stored and processed later stage.

Assignment of BCA-01 Question 1

Processing speed and performance determine by using technique of OR.

Block diagram of Processor:-

CPU may consist of ALU,CU and OR.

Feature of IBM 1401 PC:-

- ✓ It was announced in late 1950.
- ✓ Memory capacity was 16000 characters
- ✓ MSI,LSI technology was used
- ✓ Processing speed measured in MIPS

Technology of Integration:-

- ✓ SSI
- ✓ MSI
- ✓ LSI
- ✓ VLSI
- ✓ ULSI

Feature of IBM PC AT:-

- ✓ It was announced in 1984.
- ✓ Memory capacity was 200 times more characters than IBM 1401.
- ✓ VLSI technology was used in this processor
- ✓ PC AT stand for personal computer advanced technology.
- ✓ Processing speed of PC At is more than IBM 1401 PC
- ✓ Size of operational register is more than IBM 1401 PC (Processing speed measured in MIPS)
- ✓ It is chips based processor.

Note:-Chips

Small components that contains a large amount of electronic circuitry. They are building blocks of A computer and perform various functions such as doing arithmetic ,serving as computer memory or controlling other chips.

Generation of Processor:-

First Generation.
Second generation.
Third Generation.
Fourth Generation.
Fifth generation.

Memory System in Computer/Memory Hierarchy in Computer:-

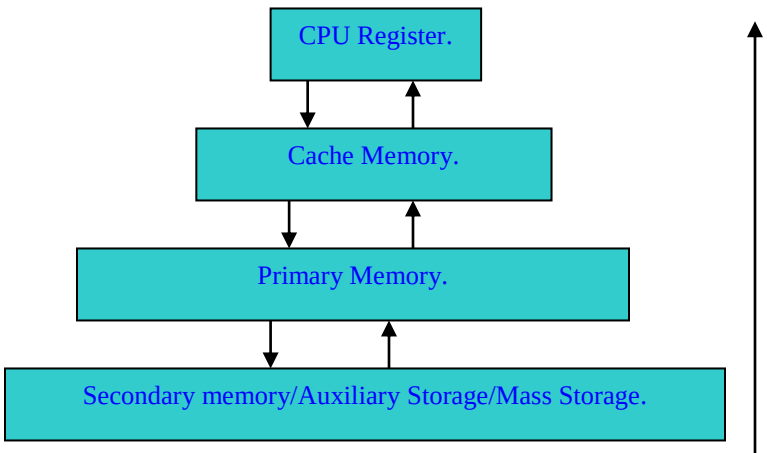
It is used for storing instructions and data permanently or temporarly for further use.It may also store audio, video, images, graphics ,s/w etc.

Memory in a computer system is required for storage and subsequent retrieval of the instructions and data.

Example:-

Hard Disk/Electromagnetic Disk.	CD ROM (Optical Memory).	DVD (Optical Memory).
BLU Ray Disk (Optical Memory).	Audio/Video Cassette (Magnetic tape)	
Pen Drive/Flash Memory.	Zip Disk.	
Flopy Disk (Magnetic Disk).	Memory Chips.	RAM, ROM Etc.

Memory Hierarchy:-



1:-Decreasing Size.

2:-Increasing Speed.

3:-Increasing Cost.

CPU Register/Operational register:-

It is the fastest memory in speed and smallest in size. It is very costly. It determines processing speed of CPU.Its speed measured in MIPS.

M Million
I Instruction
P Per
S Second

Larger the size of register faster may be the speed of CPU.

Example:- 32 bits register Size→Slower Speed.
64 bits register size→Relatively faster speed.

Cache memory/High Speed Memory:-

It is placed between CPU register and primary memory.It is used for increasing speed of CPU.

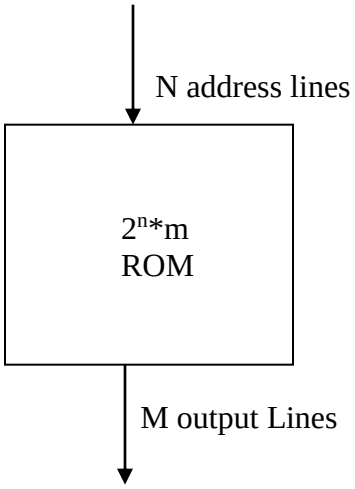
Primary Memory/Main Memory:-

It consists of semi conductor materials. Like Silicon(Si) and Germanium(Ge).
There are two types of primary memory.

- RAM(Random Access Memory).
  - SRAM(Static Random Access Memory).
  - DRAM(Dynamic Random Access Memory).
- ROM(Read Only Memory).
  - PROM (Programmable Read Only memory).
  - EPROM (Erasable PROM).
  - EEPROM (Electrically EPROM).
  - Flash Memory (Used in I/O storage device. Example I/O storage devices,MP3 music players, Digital camera) There are two kinds of flash memory
    - Code storage flash:-It stores programming algorithms and largely found in cell phones
    - Data Storage flash:-It stores data and comes digital cameras and MP3 players.

Example:-BIOS (Basic Input Output System)/Firmware.

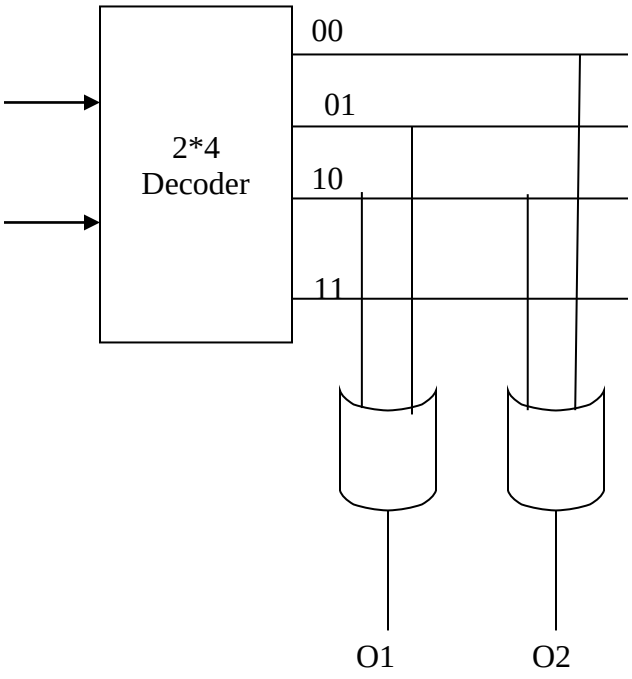
Block Diagram of ROM:-



I_1	I_2	O_1	O_2
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

Truth table of Simple ROM

Circuit diagram:-



A ROM is essentially a memory or storage device in which a fixed set of binary information is stored. The number of distinct addresses possible with n input variables is 2^n.A ROM is characterized by the number of words (2^n) and the number of bits(m) per word.
Example:-32*8 ROM which can be written as 2^5*8 consist of 32 words of 8 bit each. It means there are 8 output lines and 32 distinct words stored in the unit.

Difference Between RAM and ROM

RAM	ROM
Volatile Existence of data occur during running mode	Non Volatile(Permanently)
Uses by both users and System	Only for System
Consist of Si And Ge	Consist of Si And Ge

Difference Between SRAM and DRAM

SRAM(Static Random Access Memory)	DRAM(Dynamic Random Access Memory)
No loss of electrical Signal (1↔1,0↔0)	loss of electrical Signal(1→0,0→1)
No need of refreshment.	Need of refreshment.
It is made with transistors.	It is made with cells that stores data as charge capacitors.

Difference between ROM, PROM, EPROM and EEPROM

Name of Memory	Write Time	Access Time	Cycle
ROM	Many Hours	Nano Second(10^{-9})	1
PROM	Less Hour	Nano Second(10^{-9})	10
EPROM	Minute	Nano Second(10^{-9})	100
EEPROM	Second	Nano Second(10^{-9})	1000
Flash Memory	Mili Seconds	Nano Second(10^{-9})	10000

Secondary Memory/Auxiliary Memory/Backing Store:-

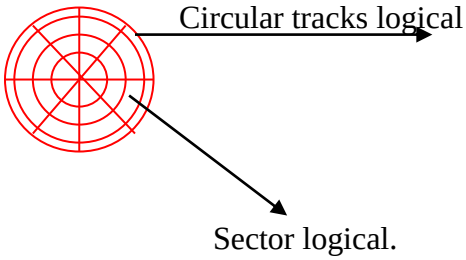
It is the largest storage device, which is used for storing data and information permanently.

Example:-

Hard Disk (Electro magnetic disk).
Pen Drive.
CD ROM (Optical Memory)/Compact Disk Read Only Memory.
DVD (Optical Memory)/Digital Video Disk/Digital Versatile Disk.
Blu Ray Disk (Optical Memory),CD-R.
CCD,Bubble Memories:-(They have arrays of cells that can hold charge packets of electron.A word is represented by a set of charge packets,the presence of each packets of represent the bit –value 1.
Audio/Video cassette (Magnetic tape).
Zip Drive.
Floppy Disk (Magnetic Disk) .
Memory Chips (It uses technology of flash).

Storage Mechanism In Magnetic Disk:-

In magnetic disk data stored in the form of logical circular tracks and sectors.



Circular Tracks and sectors.
One Sector=512 Bytes.

Access Time:-

Seek Time

Latency Time

For read and write, operation head is needed. There are two types of head

- Fixed head.
- Moveable head.

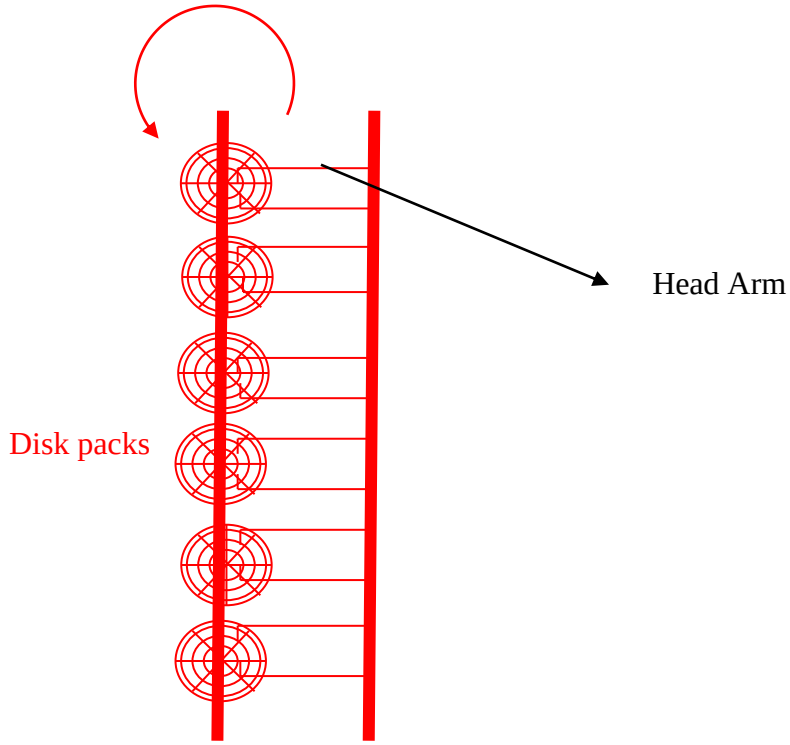
Seek Time:-

Time required to position the head on a specific track for read and write operation is called **seek time**.

Latency Time:-

The time required by a sector to reach below the read/write head is called **latency time.**

Storage Mechanism in hard disk:-



Access Speed Measurement Unit:-

RPM (Rotation/Revolution Per Minute) is used for measuring speed.

RPM may be 3600, 7200, above 10000.

Some Standard of Hard Disk

40 GB Hard Disk
80 GB Hard Disk
120 GB Hard Disk
320 GB Hard Disk
560 GB Hard Disk
1 TB Hard Disk
Etc

Name of Company:-

Samsung, LG., Philips.Sony, Moser Bear, IBM (International Business Machine).
Lenovo,Del,Segate ,I-Ball ,Compaq,HP,Apple,WD,Hitachi,Montec etc.

Storage Mechanism in Optical memory(Spiral tracks & sectors are used for storage):-

- CD ROM(Compact Disk read Only Memory) Low density
- DVD(Digital Video Disk) High Density
- BLU RAY DISK Very Much High Density
- WORM(Write Once Read many)

Read/Write Operations:-

Laser beams are needed.

For CD ROM Low Power of laser beams.
For DVD High Power of laser beams.
For BLU Ray Disk Very High Power of laser beams.

Data are stores in spiral tracks and sector type system/Organization.

Storage Capacity of CDROM (700 MB to 800 MB)
Storage Capacity of DVD (4 GB to 5 GB)
Storage Capacity of BLU Ray Disc (6 GB to 12 GB)

Storage Mechanism in Magnetic tape:-

Access Method:-

To retrieve (Read/Write operation) data from storage device by using following three technique.

- **Sequential Access Method.**
- **Direct Access Method.**
- **Random Access Method.**

Sequential Access Method.

To access data from storage system in sequential pattern.That is, One by One.

Example:- Magnetic Tape.

Direct Access Method.

To access data from storage device by using key value and corresponding address.

Example:-Magnetic disk,CD ROM.

Random Access Method.

To access data from storage device in random manner.

Example:-Semiconductor Memories.

RAID:-(Redundant Array of Independent Disks)

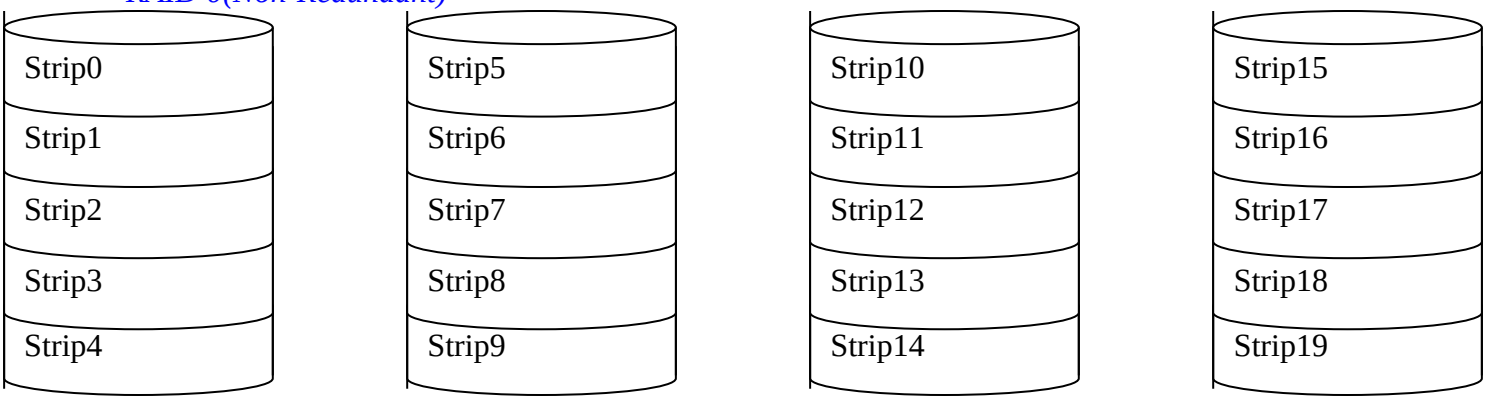
It is a technology to improve the secondary storage media by increasing the capacity, performance and reliability. Multiple –disks database schemes, is termed as RAID.
Some of the basic questions for such systems are:

How are the disks organized?
Ans:-May be as an array of disks.
Can separate I/O requests be handled by such a system in parallel?
Ans:-Yes, but only if the disk accesses are from separate disks.
Can a single I/O requests be handled in parallel.
Ans:-Yes but the data block requested should be available on separate disks.
Can this array of disks be used for increasing reliability?
Ans:-Yes but for that redundancy of data is essential.

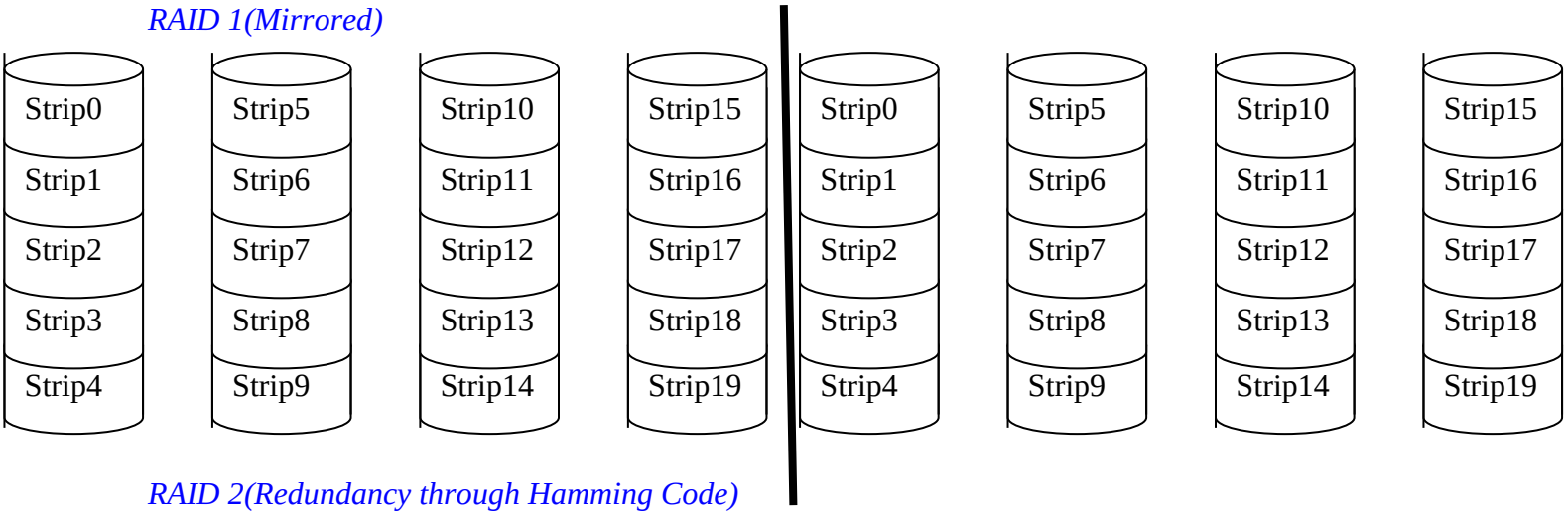
Characteristics of RAID disks:-

- OS considers the physical disks as a single logical drive.
- Data is distributed across the physical disks.
- In case of failure of a disk, the redundant information kept on redundant disks is used to recover the data.

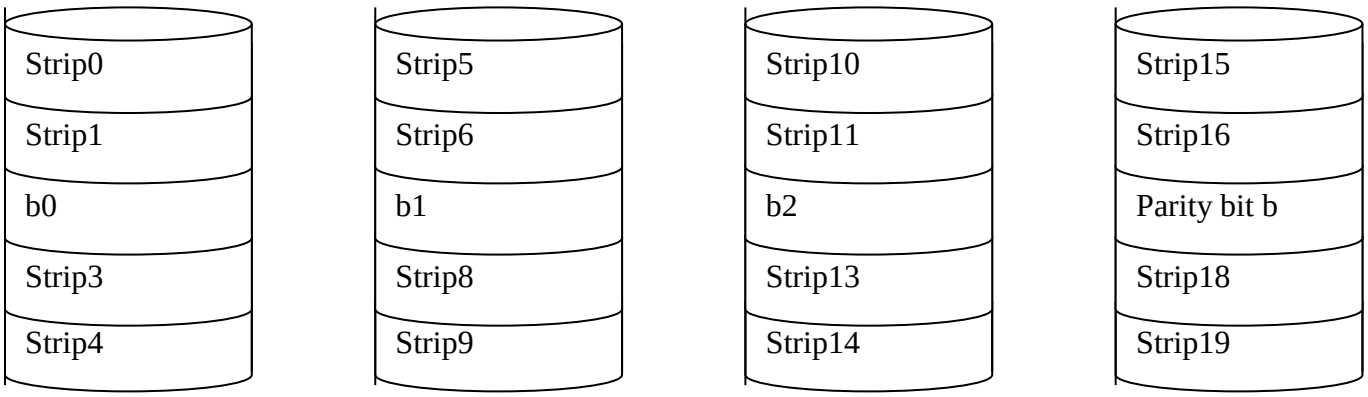
RAID 0(Non-Redundant)



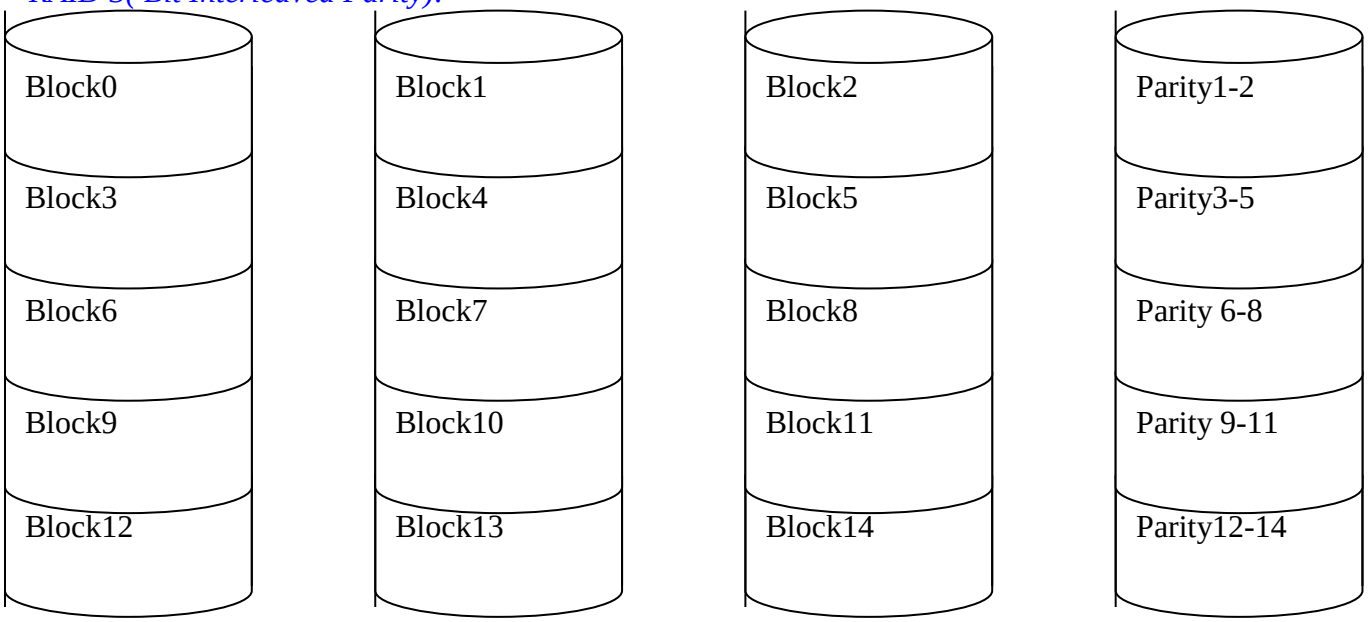
RAID 1(Mirrored)



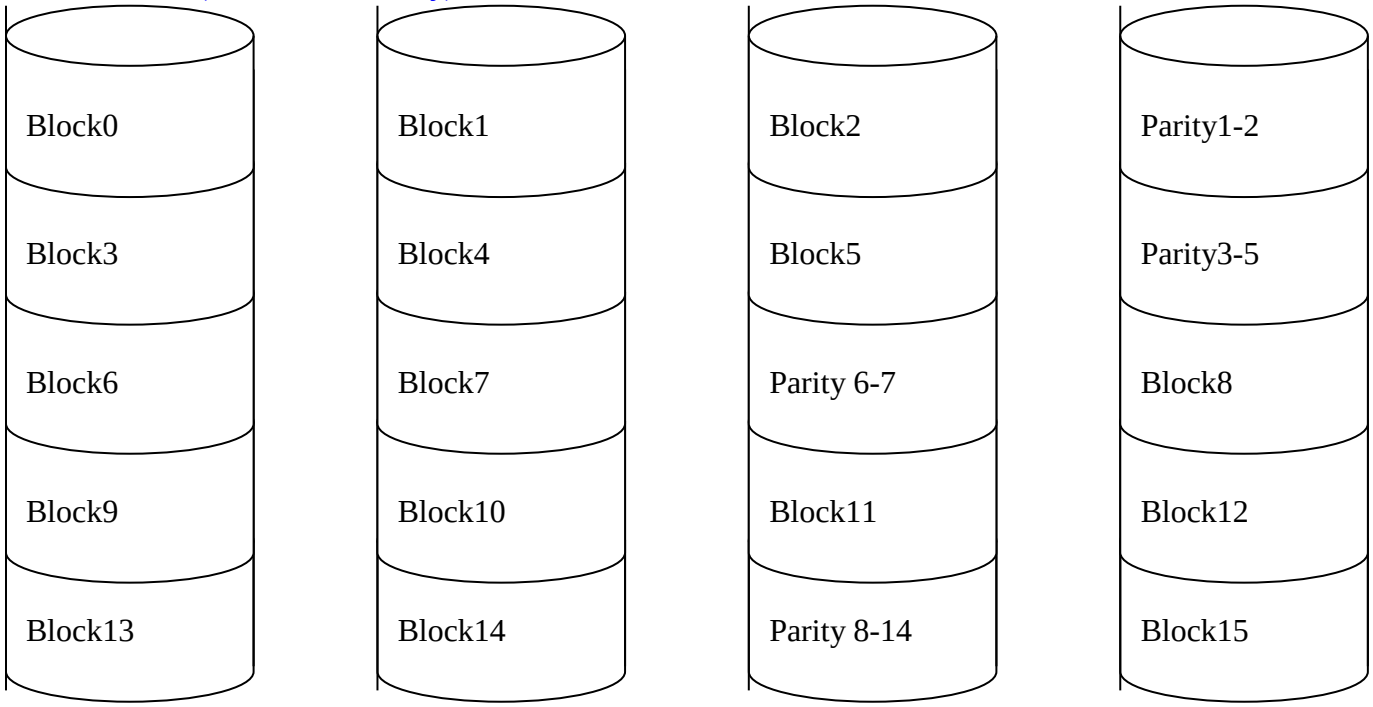
RAID 2(Redundancy through Hamming Code)



RAID 3(Bit Interleaved Parity):-



RAID 4(Block Level Parity):-



RAID 5(Block Level Distributed Parity)

RAID 6(Dual Redundancy)

In RAID technologies have two important performance considerations

- The data transfer rate.
- Input/Output request rate.

High data transfer rate dependent on:

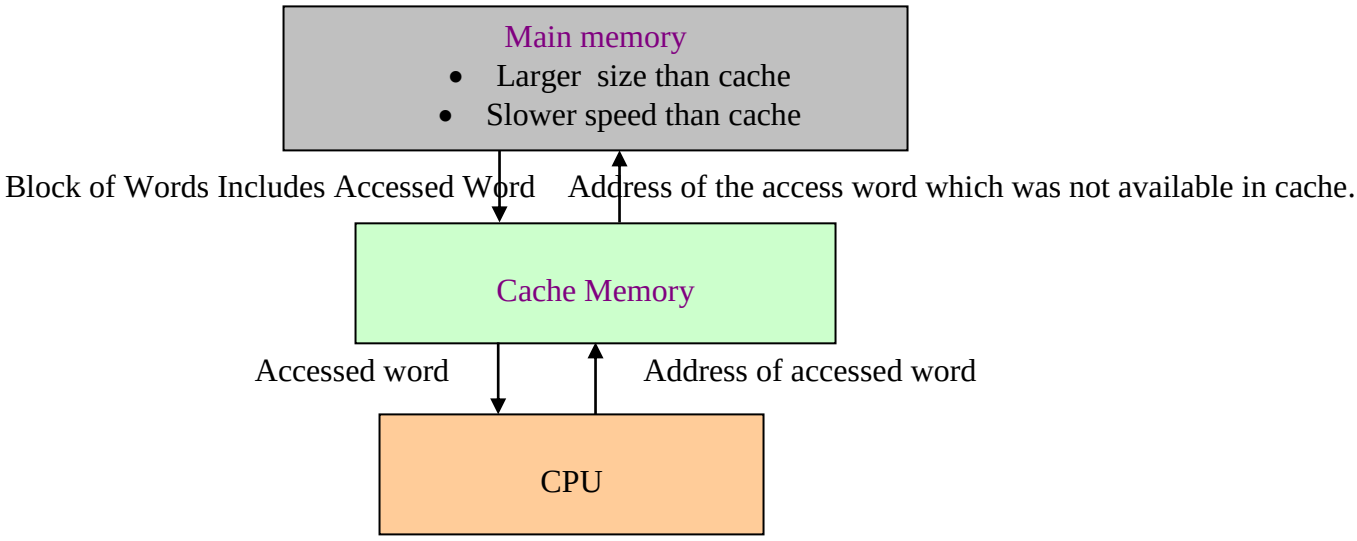
- Path between individual disks and memory.
- Fulfillment of an I/O request by multiple disks of array.

Concepts of High Speed memories:-

There are four possible answers to increase speed of CPU.

- Decrease the memory access time.
- Access more words in single access cycle.
- Insert a high speed memory termed as cache memory between processor and main memory.
- Use associate addressing in place of random access.

Operation of cache memory:-



The use of cache memory requires several design issues to be addressed. Some of design issues are briefly summarized below.

- **Cache Size:-** Example:-64 KB, 128 KB, 512 KB or 1 MB.
- **Block Size:-**It refers to the unit of data exchanged between cache and main memory.
- **Replacement policy:-**When a new block is to be fetched into the cache, another may have to be replaced to make room for new block.
- **Write policy:-**The write policy decides when the altered words of a block are written back to main memory.

Brief Description of Cache organization:-

Cache memories are found in almost all latest computers. They are very useful for increasing the speed of access of information from main memory. The idea of cache memory is that by keeping the most frequently accessed instructions and data in the fast cache memory.
The performance of cache memory is frequently measured in terms of quantity called **hit ratio**.
If the word is not found in cache, it is in the main memory and it counts as a miss.

Mapping/Relationship procedure:-

Data from main memory to cache memory is referred to as a mapping process. There are three types of mapping organizations:-

- Associative Mapping/Relationship.
- Direct Mapping/Relationship.
- Set Associative mapping/Relationship.

Associative Mapping:-

This memory stores both the address and data of the memory word.

Direct Mapping:-

The direct mapping cache organization uses the n-bit address to access the main memory and k-bit index to access the cache.

Set associative:-

This mapping is improvement on direct mapping organization in that each word of cache can store two or more words of memory under the same index address.

Chip Organization:-

Chip consist of semiconductor material. In chip it is an arrangement of decoder, bit line and word line. There are several technique used for a chip. Following two common technique uses mostly in chip.

- 2D Chip organization.
- 2 ½ D Chip Organization.

Figure Page Number (a) and (b) page Number 92 and 93.

Differences between 2D and 2 ½ D

2D	**2 ½ D**
It require more circuitry.	It require less circuitry.
Error detection and correction codes are not effectively used.	Error detection and correction codes are effectively used.
16 bits or 32 bits input/output pins are used.	Only one I/O pin is used.

I/O Organization or Input /Output System:-

Input → Processing (CPU) → OUTPUT

1. I/O Peripherals (Devices)✈Already Study in Von Neumann architecture.
2. I/O Modules.
3. I/O Technique.
4. I/O Processors.

I/O Peripherals (Devices) (We have already study in Von Neumann Archtitecture)

Input Devices

- Keyboard
  i. Cherry Keyboard(Costly and repairable)
  ii. Membrane Keyboard(Cheaper & non repairable)

- Mouse
  i. Trackball Mouse
  ii. Optical Mouse
- Joystick(It is used for playing game)
- Light pen
- OMR (Optical Mark Recognition/Reader).

Example:-

A B C D

○ ○ ● ○

- OBR (Optical Bar Reader).

||||||||||||||||||||||||||||

11 22 2233 4 2 2 2 333 44

- MICR (Magnetic Ink Character recognition).
- Scanner:-It is used for scanning documents, graphics and images in digital form.
- Voice speech Synthesizer:-It is used for recognizing audio sound.
- Mike.

Etc.

Output Devices

5. **VDU(Visual display Unit)Produce Soft Copy**
   a. CRT Screen (Cathode Ray Tube) ***Pixels*** (.) are fundamental elements of images.
   b. LCD Screen(Liquid Crystal Display) Crystal rods are used for creating graphical object on screen
   c. LED Screen(Light Emitting Diode)
   d. PLASMA Screen

Pixels (Picture Elements) are the fundamental elements of graphics.

6. Printer(Produce Hard Copy)
   a. Impact Printer(Inked ribbon is used)
      i. Dot Matrix Printer(DMP)
      ii. Daisy Wheel Printer
      iii. Drum Printer
      iv. Line Printer
   b. Non Impact Printer
      i. Inkjet printer (Cartridge is used both types color and Mono color printing).
      ii. Laser Printer (Toner is used) It is the best printer in quality and speed.

Printing Speed measurement:-

- ❖ CPS (Character per second).
- ❖ Page Per Minute.

7. Plotter.
   It is used by architect engineer for graphical output on paper.
8. Speeker

Etc.

Both Input/Output devices:-

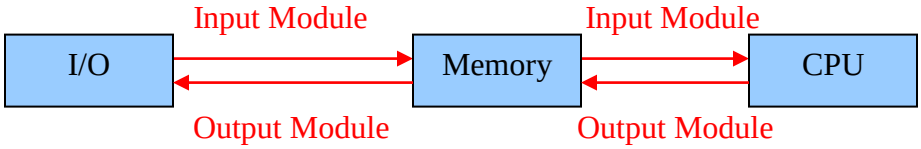
- MODEM (Modulator & Demodulator).
  It is used for internet connection.
- Touch Screen.
- Communication Port.

I/O Module:-

An I/O module is a mediator between the processor (CPU) and I/O devices.

Function/Task Of Module:-

- It should be able to provide control and timing signal.
- It should communicate with CPU.
- It should communicate with I/O devices.



Input/Output Technique:-

The I/O operations can be performed by three basic techniques. These techniques are:

- Programmed I/O
- Interrupt Driven I/O
- Direct Memory Access(DMA)

Programmed I/O:-

It is useful I/O method for computers where hardware costs need to be minimized. It perform following functions

- Transfer of data from I/O device to the CPU registers/OR.
- Transfer of data from CPU register/OR to Memory.

Interrupt (It is a program):-

It is a mechanism for transferring **a block of data from one memory** to **another memory**.It also performs following operations. The term interrupt is defined loosely to any exceptional event that cause CPU to temporarily transfer its control currently executing program to a different program which provide service to exceptional event.

- Initiation of I/O operation.
- Completion of I/O operation.
- Occurrence of H/W or S/W errors.

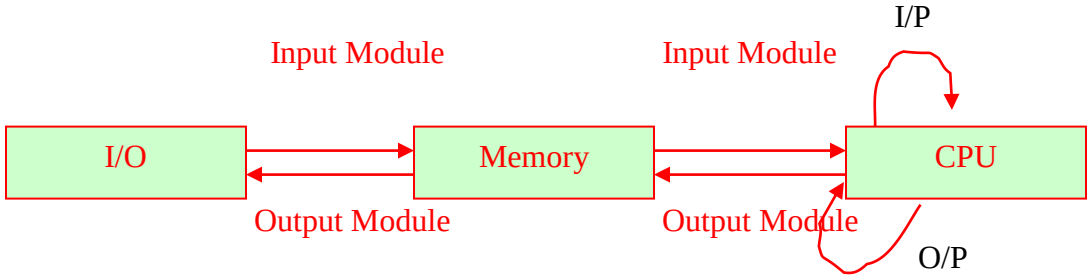
Direct Memory Access (DMA):-

It is a module for transfer large amount of data from CPU.DMA operate in the following way.

- Which operation to be performed.
- The address of I/O device which is to be used.

I/O Processors:-

It include following steps.
Step 1:-Direct Control of CPU on I/O device.
Step 2:- Introduction of I/O module.
Step 3:-CPU need not wait for I/O operations to complete.
Step 4:- Direct access of I/O module to the memory via DMA.



What Is scanners:-

It is an input device that allows we to capture drawing or photographs or text from tangible source (Paper, slides etc) into electronic form. There are various types of scanners.

- Drum scanners
- Hand scanners
- Video scanners

What Is SMPS:-

It converts AC current into DC current for computer. It is referred to as Switched Mode Power Supply.

AC (Alternative Current) → DC (Direct Current)

Advantage of SMPS:-

- It generates less heat.
- It produces power range between 150 to 200 watts.
- It uses less expensive transformers and circuit.

What Is Modems:-

It is used for converting analog signal to digital signal and digital to analog signal.
It is used for internet connection.
There are many types of MODEMs.

- Internal Modems.
- External Modems.
- Pocket Modem.
- PC Card Modems.

Modem understand a set of instructions called Hayes command set or AT command Set.

Technology of LCD:-

It is a technology for VDU.It consumes less electric power and produce less harmful radiation.Images are generated by crystal rods.

There are following three types of LCD technology.

- Reflective LCD (It is display generated by selectively blocking reflected light).
- Backlit LCD(It is display due to a light source behind LCD panel).
- Edgelit LCD (It is display due to a light source adjacent to the LCD panel).

Mechanism of CRT screen:-

It is a major technology on which monitors and televisions have been based. Following factors influence the quality of image of the monitor.

1:-The phosphor coating:-This effect the color and the persestitence (The period the effect of a single hit on a dot lasts).

2:-The Cathode (Electron Gun):-The sharpness of image depends on the good functioning of this gun.

3:-The Shadow Mask/Aperture Grill:-This determines the resolution of the screen in color monitor.

4:-The screen, glare and lighting of the monitor.

Bandwidth of CRT screen:- It is combination of address of each pixels and synchronization signal.

DPI of CRT screen:- It is a measure for the actual measure for the actual sharpness of the onscreen image.This depends both resolution and size of the image.

Monitor resolution:-It depends upon video cards. There are two types of frequency generated by CRT .

A:- Horizontal Frequency.
B:- Vertical frequency.

Both types of frequency measured in KHz.

Color Depth:-

It is clear that an image consists of an array of pixels.

Color Mode	Depth (Bits/pixel)
Monochrome	1
16 colors	4
256 colors	8
High color	16
True Color	24

External Interfaces:-

I/O module and the peripherals are characterized into two main categories:

A: - Parallel Interface.(It is used for connecting printer)

B: - Serial Interface.(It is used for connecting Mouse and MODEM)

Figure of Parallel Interface:-

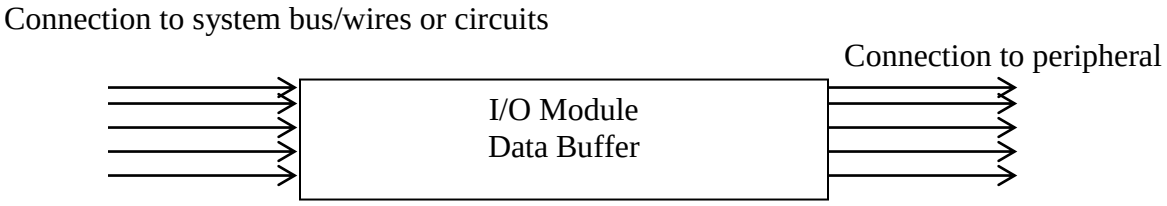
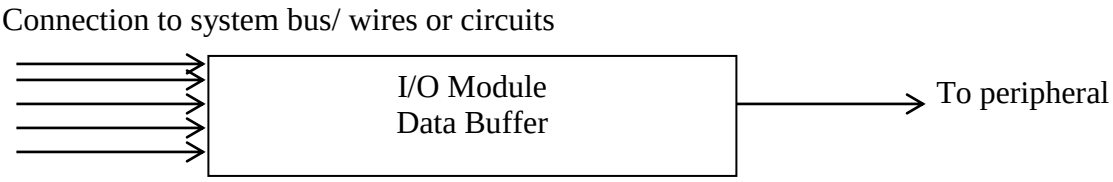


Figure of Serial Interface:-



What is Parallel Processing:-

Parallel computer is classified by According to M.J. Flynn's. There are four categories of computer.

- ✓ SISD (Single Instruction Single data).
- ✓ SIMD (Single Instruction Multiple data).
- ✓ MISD (Multiple Instruction Single data).
- ✓ MIMD (Multiple Instruction Multiple data).

SISD:-

Using this technology, Single Instruction applies on single data stream through pipeline.
Example:-Conventional Von Neumann architecture.

SIMD:-

Using this technology, Single Instruction applies multiple data stream or Instructions broadcast on multiple data stream.
Example:-Array Processors.

MISD:-

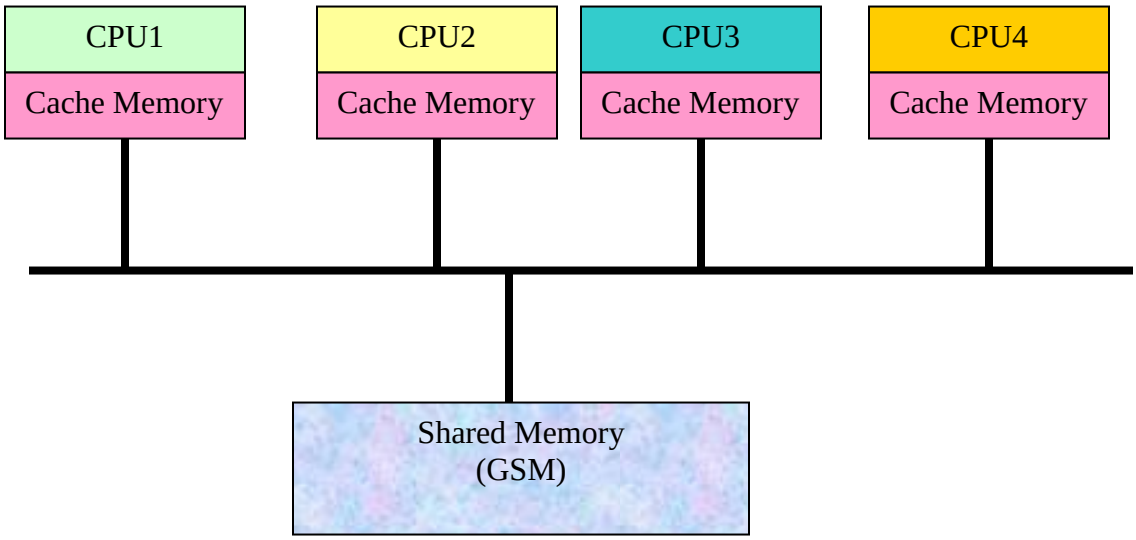
Using this technology, Multiple Instruction applies on single data stream.
Example:-Distributed architecture, Vector Processors.

MIMD:-

Using this technology, Multiple Instruction applies on multiple data stream.
Example:-Multiprocessor System, Data Flow architecture.

Multiprocessor Interconnections:-

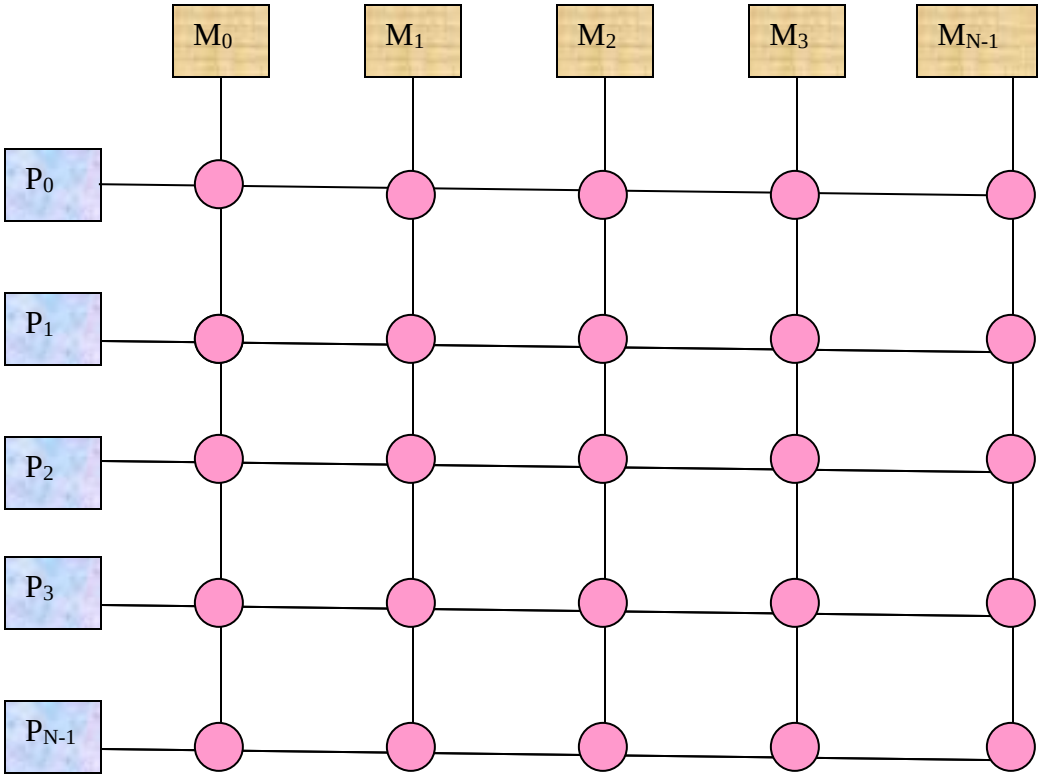
- Bus Oriented System.
- Crossbar Connected System.
- Hypercube.
- Multistage switched based system.



UMA (Uniformly Memory Access) tightly coupled system

Crossbar Connected System:-

The crossbar itself has no contention. It allows simultaneous access of N processors to N memories, Provided that processor accesses a different memory.



The cross point switch is the source of delay between a processor and memory. When more than one processor attempts to access the same memory at the same time. The crossbar scheme allows high degree of parallelism between unrelated task.It requires N^2 crosspoint switches to fully connect N endpoints to N other end points, such as processors and memory.

Instruction Set:-Most Important

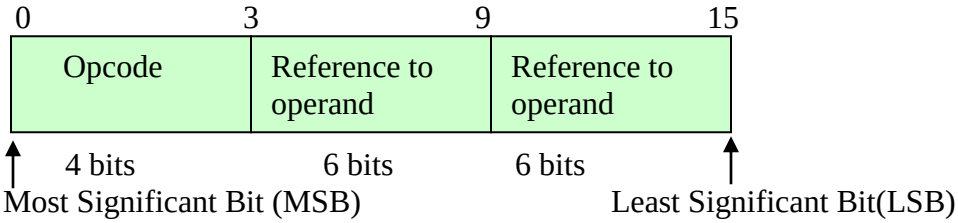
It is a collection of all the instructions/commands a CPU can execute. Each instruction consists of several elements.

Elements of Instruction sets:-

An instruction have following elements.

- An operation code also called opcode which specifies the operation to be performed.
- A reference to the operands on which data processing is to be performed.
- A reference to the operands which may store the results of data processing operation performed by instructions.
- A reference for the next instruction, to be fetched and executed.

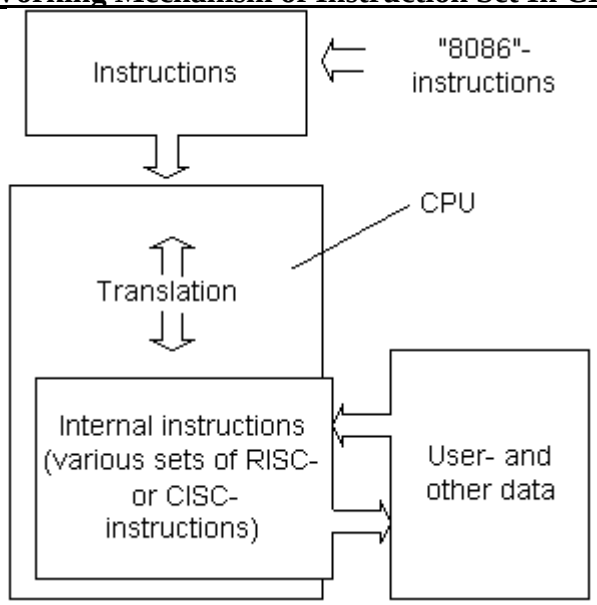
How is an Instruction represented:-



Types of Instruction:-

- Data Processing Instructions:- (These instructions are used for arithmetic and logic operations in machine)
- Data Storage/Retrieval Instructions :-(These instructions are used load and store instructions)
- Data Movement Instructions:- (These instructions are required to bring in programs and data from various devices to memory or to communicate the results to the input/output devices.)
- Control Instructions:-These instructions are used for testing status of computations through processor status.
- Miscellaneous Instructions :- (These instructions does not fit in any of the above category).

Working Mechanism of Instruction Set In CPU:-



Operand data Types:-

An operand data types specifies the type on which a particular operation can be performed. Following types of general categories of operand data types.

- Addresses (It is treated as form of data which is used in calculation of actual physical memory address of an operand)
- Numbers (It may be either integers, floating points or decimal numbers)
- Characters (sequence of characters are called string)
- Logical data types (True 1or false 0)

Operation Types:-

- Data transfer Operations.
- Arithmetic Operations.
- Logical & Shift Operations.
- Conversion Operations.
- I/O Operations.
- System Control Operations.
- Transfer Control Operations.

Data transfer operations:-

Example:-

MOVE or TRANSFER →Transfer a block of data from source to destination.
Example MOVE R1, R2 or TRANSFER R1, R2
STORE→Transfer a word from the processor to a specified location in the main memory.
LOAD or **FETCH**→Brings a word from locations of main memory to the processors.
EXCHANGE →Exchange the contents of the source with the destination.
CLEAR or RESET→Transfer a word containing all 0's to the destination.
SET→ Transfer a word containing all 1's to the destination.
PUSH→ Transfer a word from a source to top of stack.
POP→ Transfer a word from top of stack to destination.

Arithmetic operations:-

Example:-

ADD
SUBTRACT
MULTIPLY
DIVIDE
ABSOLUTE → (It convert negative value into positive value)
NEGATE → (It convert given value into negative)
INCREMENT → (It increment value by one)
DECREMENT→ (It decrement value by one)

Logical and Shift operations:-

Example:-

AND, OR, NOT, EX-OR, EX-NOR→ (Logical Operator)
LSH (Bit shift left side) →Shift operator
RSH (Bit shift right) →Shift operator
CSH (Bit shift in circular manner) →Shift operator

Let R1=7 and R2=9

R1=	0 0 0 0 1 1 1
R2=	0 0 0 1 0 0 1
R1 AND R2=	0 0 0 0 0 0 1
R1 OR R2=	0 0 0 1 1 1 1
R1 Ex-OR R2=	0 0 0 1 1 1 0
R1 Ex-NORR2=	1 1 1 0 0 0 1

Conversion operations:-

TRANSLATE (Translate ASCII to EBCDIC)
CONVERT (Convert one format to other format)

Input/Output operations:-

READ (or INPUT)
WRITE (or OUTPUT)
TEST I/O

System Control operations:-

OSCALL :-It is causes the instruction of execution of current program and passes the control to the OS.

Transfer of Control operations:-

Branch:-BRP, BRN, BRZ, BRO
Skip:-ISZ
Subroutine Call :-It is a self contained user program which contains the code often used frequently in large program.

Example :-CALL, RETURN commands are used in subroutine

Addressing Scheme:- Important

An operand may be specified as the part of the instructions or the reference of the memory location where the value is stored may be given.The addressing scheme refers to the mechanism employed for specifying operands. There are following types of addressing schemes.

- Immediate addressing

  Under this scheme, the actual operand D is A, the content of the operand field

  D=A

  It is used for initialize the value of a variable.

- Direct Addressing:-Under this scheme, the content A of the operand field specify EA,the effective address of the operand

  EA=A
  D=(EA) Reference of effective address.

- Indirect Addressing

Under this addressing scheme the effective address EA and contents of the operand field are related by

EA=(A)
D=(EA)

- Register Addressing

  In this scheme, the instruction specifies the address of the register containing the operand.

  EA=R
  D=(EA)

- Register Indirect Addressing

  Under this addressing scheme the operand field specifies the registers which contain the address of the operand.

  EA=(R)
  D= (EA)

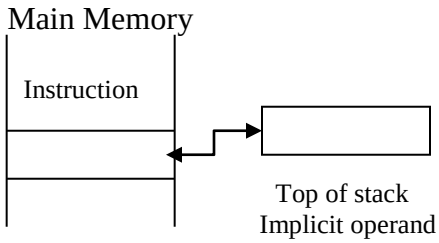
- Displacement addressing

  This is very powerful addressing scheme.It combines both the direct as well as the register indirect addressing schemes.

  EA=A+(R)

- Stack addressing scheme:-

  In this scheme, the address of an operand is not specified explicitly.
  The operand is found on the top of stack.(PUSH and POP operations are performed).



Comment:-

All the above addressing schemes are using memory locations and not registers their disadvantage is that the addressable memory location is dependent on the instruction code length i.e., if the code is of n bits the addressable memory is 2^n

Allocation of Bits among Opcode and Operand in instruction:-

Some of the factors that are considered for selection of addressing bits.

- Number of addressing modes.
- Number of operands.
- Register addressing versus memory addresses.
- Granularity of address: - It implies whether an address is referencing a byte or a word at a time.

  Example:-

  Memory of 8K words (1 word=16 bit)
  Word addressing =8K words.
  $=2^3*2^{10}$
  $=2^{13}$ Bytes.
  →13 bits are required for Word addressing.

  Byte Addressing $=2^{13}$Words.
  →2^{14}Words.
  →14 bits are required for byte addressing.

Commonly Used 8086 Instruction Set

AHC - Add with carry flag
ADD - Add two numbers
AND - Bitwise logical AND
CALL - Call procedure or function
CBW - Convert byte to word (signed)
CLI - Clear interrupt flag (disable interrupts)
CWD - Convert word to double word (signed)
CMP - Compare two operands
DEC - Decrement by 1
DIV - Unsigned divide
IDIV - Signed divide
IMUL - Signed multiply
IN - Input (read) from port
INC - Increment by 1
INT - Call to interrupt procedure
IRET - Interrupt return
LEA - Load effective address offset
MOV - Move data
MUL - Unsigned multiply
NEG - Two's complement negate
NOP - No operation
NOT - One's complement negate
OR - Bitwise logical OR
OUT - Output (write) to port
POP - Pop word from stack
POPF - Pop flags from stack
PUSH - Push word onto stack
PUSHF - Push flags onto stack
SAR - Bitwise arithmetic right shift (signed)
SBB - Subtract with borrow
SHL - Bitwise left shift (same as sal)
SHR - Bitwise right shift (unsigned)
STI - Set interrupt flag (enable interrupts)
SUB - Subtract two numbers
TEST - Bitwise logical compare
XOR - Bitwise logical XOR

AHC : Add with carry flag

```
Add With Carry Flag
```

ADD : Add two numbers

```
ADD     Add two numbers
```

AND : Bitwise logical AND

```
AND Bitwise logical AND
```

CALL : Call procedure or function

```
CALL Call procedure or function
```

CBW : Convert byte to word (signed)

```
CBW Convert byte to word (signed)
```

CLI : Clear interrupt flag (disable interrupts)

```

CLI Clear interrupt flag (disable interrupts)
```

CWD : Convert word to doubleword (signed)

```
CWD Convert word to doubleword (signed)
```

CMP : Compare two operands

```
CMP Compare two operands
```

DEC : Decrement by 1

```
 DEC Decrement by 1
```

DIV : Unsigned divide

```
DIV Unsigned divide
```

IDIV : Signed divide

```
 IDIV  Signed divide
```

IMUL : Signed multiply

```
IMUL Signed multiply
```

IN : Input (read) from port

```
IN Input (read) from port
```

INC : Increment by 1

```
INC  Increment by 1
```

INT : Call to interrupt procedure

```
INT Call to interrupt procedure
```

IRET : Interrupt return

```
IRET Interrupt return
```

JMP : Unconditional jump

```
JMP Unconditional jump
```

LEA : Load effective address offset

```
 LEA Load effective address offset
```

MOV : Move data

```
MOV Move data
```

MUL : Unsigned multiply

```
MUL Unsigned multiply
```

NEG : Two's complement negate

```
NEG Two's complement negate
```

NOP : No operation

```
NOP No operation
```

NOT : One's complement negate

```
NOT One's complement negate
```

OUT : Output (write) to port

```
OUT  Output (write) to port
```

POP : Pop word from stack

```
POP  Pop word from stack
```

POPF : Pop flags from stack

```
POPF  Pop flags from stack
```

PUSHF : Push flags onto stack

```
PUSHF  Push flags onto stack
```

RET : Return from procedure or function

```
RET  Return from procedure or function
```

SAL : Bitwise arithmetic left shift (same as shl)

```
SAL  Bitwise arithmetic left shift (same as shl)
```

SAR : Bitwise arithmetic right shift (signed)

```
SAR  Bitwise arithmetic right shift (signed)
```

SBB : Subtract with borrow

```
SBB Subtract with borrow
```

SHL : Bitwise left shift (same as sal)

```
SHL  Bitwise left shift (same as sal)
```

SHR : Bitwise right shift (unsigned)

```
SHR Bitwise right shift (unsigned)
```

STI : Set interrupt flag (enable interrupts)

```
STI  Set interrupt flag (enable interrupts)
```

SUB : Subtract two numbers

```
SUB Subtract two numbers
```

TEST : Bitwise logical compare

```
TEST Bitwise logical compare
```

XOR : Bitwise logical XOR

```
XOR  Bitwise logical XOR
```

AAA : ASCII Adjust after Addition

```
AAA  ASCII Adjust after Addition
```

AAD : ASCII Adjust before Division

```
AAD  ASCII Adjust before Division
```

AAM : ASCII Adjust after Multiplication

```
AAM  ASCII Adjust after Multiplication
```

AAS : ASCII Adjust after Subtraction

```
AAS  ASCII Adjust after Subtraction
```

CLC : Clear carry flag

```
CLC  Clear carry flag
```

CLD : Clear direction flag

```
CLD   Clear direction flag
```

CMC : Complement carry flag

```
CMC  Complement carry flag
```

CMPSB : Compare bytes in memory

```
CMPSB  Compare bytes in memory
```

CMPSW : Compare words

```
CMPSW  Compare words
```

DAA - Decimal adjust AL after addition

```
DAA  Decimal adjust AL after addition
```

DAS : Decimal adjust AL after subtraction

```
DAS  Decimal adjust AL after subtraction
```

HLT : Enter halt state

```
>HLT  Enter halt state
```

INTO : Call to interrupt if overflow

```
INTO  Call to interrupt if overflow
```

LDS : Load pointer using DS

```
LDS : Load pointer using DS
```

LES : Load ES with pointer

```


LES  Load ES with pointer
```

LODSB : Load string byte

```
LODSB  Load string byte
```

LODSW : Load string word

```
LODSW  Load string word
```

LOOP : Loop control

```
LOOP  Loop control
```

LOOPE : Loop control

```
LOOPE : Loop control
```

LOOPNE

```
LOOPNE
```

LOOPNZ : Loop control

```
LOOPNZ : Loop control
```

LOOPZ : Loop control

```
LOOPZ : Loop control
```

LEA : Load Effective Address

```
LEA  Load Effective Address
```

LES : Load ES with pointer

```
LES : Load ES with pointer
```

LODSB : Load string byte

```
LODSB  Load string byte
```

LODSW : Load string word

```
LODSW : Load string word
```

LOOP

```
LOOP
```

ROL : Rotate left

```
ROL  Rotate left
```

ROR : Rotate right

```
ROR  Rotate right
```

SAHF : Store AH into flags

```
SAHF  Store AH into flags
```

STC : Set carry flag

```
STC  Set carry flag
```

STD : Set direction flag

```
STD  Set direction flag
```

STOSB : Store byte in string

```
STOSB  Store byte in string
```

STOSW : Store word in string

```
STOSW   Store word in string
```

XCHG : Exchange data

```
XCHG : Exchange data
```

Basic Structure of CPU:-

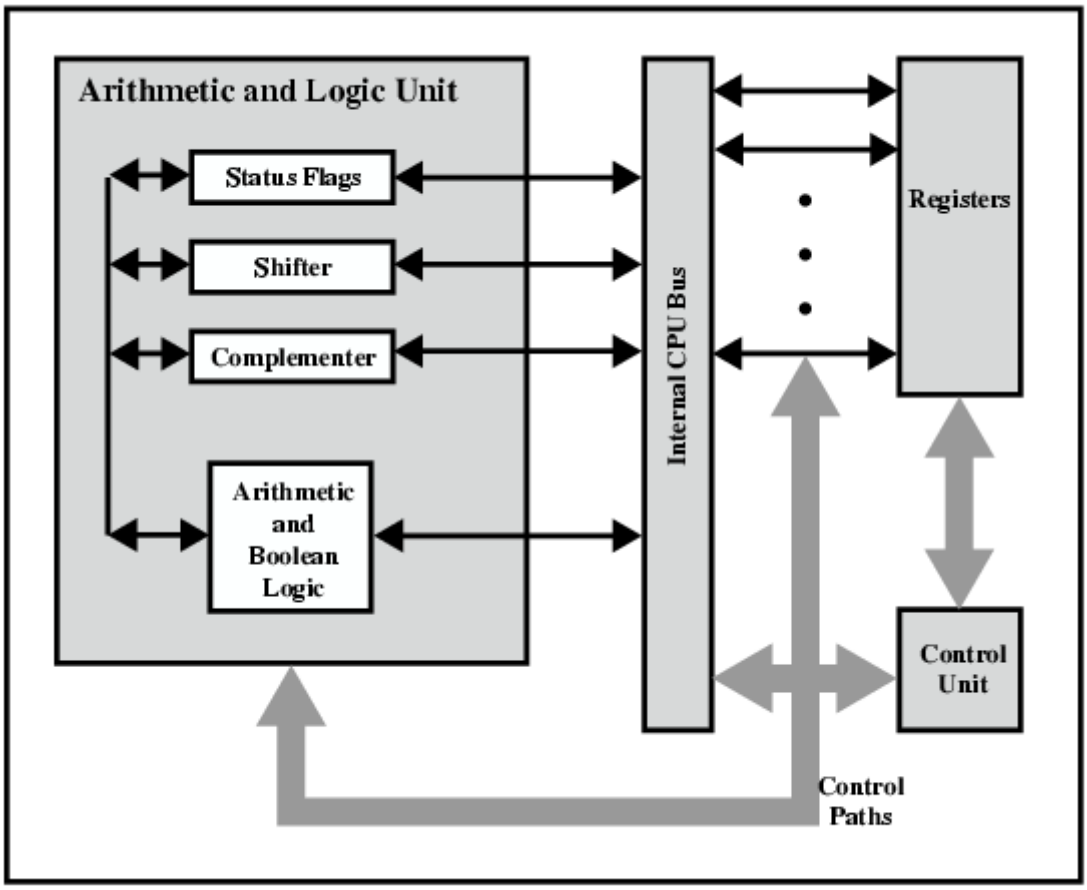
CPU basically consists of ALU and CU.

ALU:-It may perform all types of arithmetical and logical operations.

CU:-It is heart of computer which may perform controlling operations inside CPU.

There are following types of registers basically used in processor.

- AC (Accumulator):-It interacts with ALU and stores the input or output operand.
- DR (Data register):-It act as buffer between main memory and CPU.
- PC (Program Counter):-It contains the address of next instructions word to be executed.
- IR (Instruction Register):-It holds the current instructions.
- MAR (Memory Address Register):-It is used to provide address of memory location from where data is to be retrieved or to which data is to be stored.



Register Organization of CPU:-

There are basically two categories of registers.

- Programmer visible registers:-

  These registers are sub categorized into four types.

  1. GPR (General Purpose register):-It is used for calculation of address of operand for any operation code of an instruction.
  2. Data register:-It is used for storing intermediate results or data.
  3. Address registers:-It is similar to GPR but some dedicated address registers are used in several machines.
  4. Condition Codes Registers:-These registers contains condition codes which are also known as flags. These flags are set by CPU when hardware performing an operation.

- Status control and registers:-These registers can not be used in data manipulations. Some of control registers are given as :-

  PC, MAR, DR etc.

Some of commonly used status control flags are

Sign flags
Zero flag
Carry flag
Equal flag
Overflow flag
Supervisor Flag

Micro Operations In CPU:-

There are following four types of micro operations performed in CPU.

- Register Transfer Operations

  R1←R2

- Arithmetic Micro Operations

  R3←R1+R2

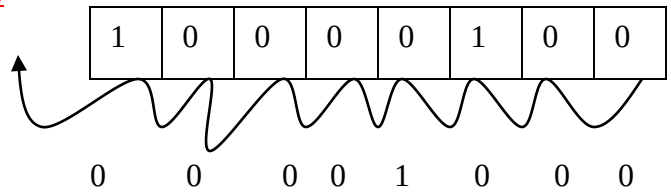
AC←AC+DR
R3←R1-R2
R1←R1+1
R1←R1-1

- Logic Micro Operations

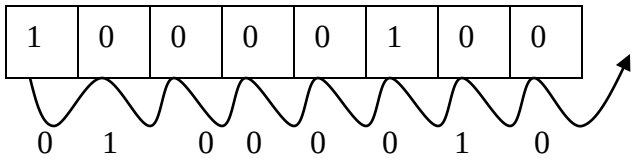
R1 =5 =00000101
R2 =6 =00000110
R1 AND R2 =00000100
R1 OR R2 =00000111
R1Ex-OR R2 =00000011
Complement R1=11111010

- Shift Micro Operations

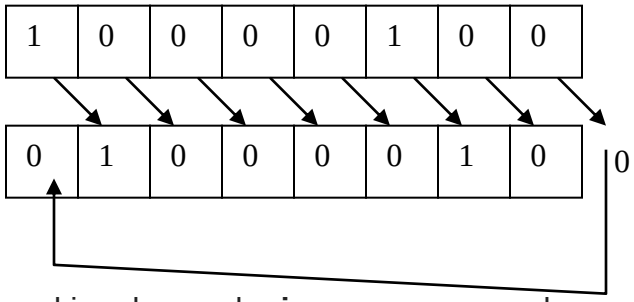
Left Shift



Right Shift



Circular Shift



What is Pipeline:-

A technique used in advanced **microprocessors** where the **microprocessor** begins executing a second instruction before the first has been completed. That is, several instructions are in the **pipeline** simultaneously, each at a different processing stage. The two techniques together are called a **pipeline.**

Instruction Execution and Micro Operations:-

Step1:- Fetch/Load Instructions
Step2:-Decode Instructions
Step3:-Execute Instructions
Step4:-Write Back Instructions
Step5:- Interrupt cycle

Above steps performed through pipeline.

Fetch Instructions Cycle :-

MAR←PC
DR←(M) It represents a memory read.
PC←PC+1
IR←DR

Indirect Cycle :-

MAR←DR
IR←DR
DR←(M)

Execute Cycle :-

MAR← PC

DR←(M)
AC←AC+DR
AC←AC+1
PC←PC +1

Interrupt Cycle :-

After completion of the execute cycle, Machine checks when the interrupt that was enabled has occurred interrupt cycle performed. If enabled has occurred then interrupt cycle is performed.Interrupt cycle in CPU register are

DR←PC (Register Transfer)
MAR←Address of location for saving return address.
(M) ←DR (Memory Write)
PC←Service programs first instruction address instruction.

ALU and Control Unit organization:-

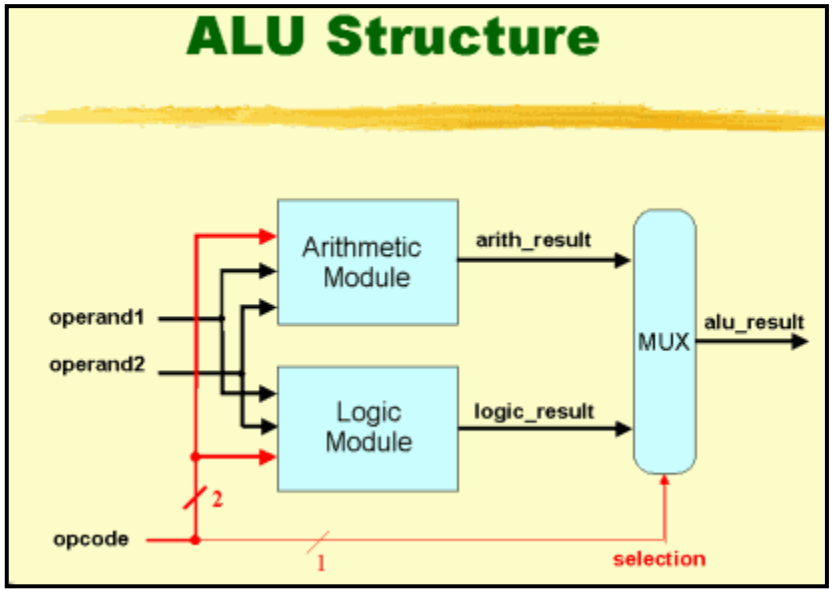
A simple ALU organization:-

It can perform all types of arithmetical and logical operations. Following micro-operations that can be defined in ALU.

AC←AC+DR	Addition
AC←AC-DR	Subtraction
AC←AC^DR	AND
AC←ACvDR	OR
AC←AC⊕DR	Exclusive OR
AC←$\overline{AC}$	NOT
Multiplication:-	AC.MQ←DR*MQ
Division :-	AC.MQ←MQ÷DR

MQ →Multiplier Quotient register It is special type of register used for multiplication and division.

Basic Structure of ALU:-



Floating Point ALU:-

It is used for performing floating point operation. Its representation shown in following figures.

Sign	Biased Exponent=8 Bits	Significant=23 Bits

Arithmetic Processor:-Important for short notes

It helps in reducing program complexity,as it provides a richer instruction set for a machine.Some of the instructions that can be assigned to arithmetic processors can be related to the addition, subtraction,multiplication and division of floating point

numbers,exponentiation,logarithms and other trigonometric functions.Arithmetic processor treated as one of the input/Output or peripheral units then it is termed as peripheral processor.

It performs the required operations on the data and communicates the result back to the CPU.

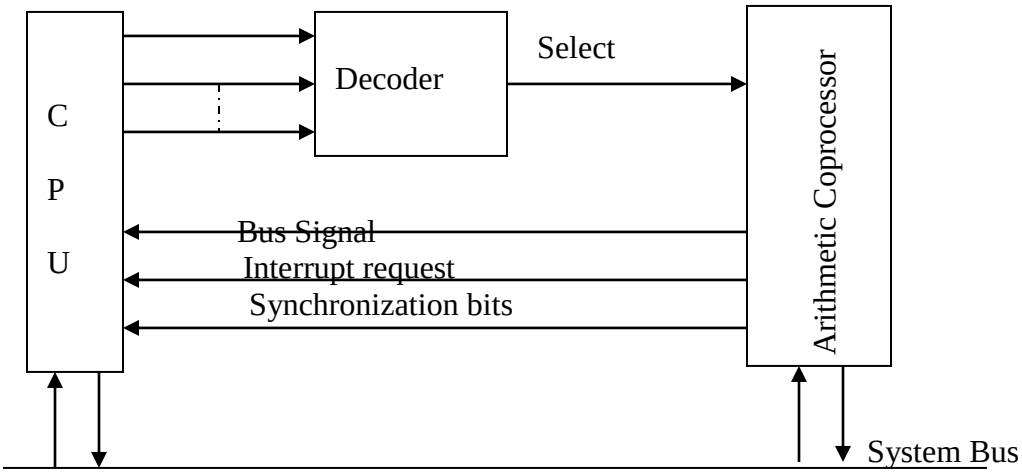
Function of peripheral processor:-

1:- Data transfer instruction.
2:- Decode & Execute.
3:- Check status in side CPU operations.
4:- Data transfer execution.

Coprocessor:-

It is peripheral for that are tailor made for a particular family of CPU.Each CPU is designed to have a coprocessor interface.The CPU H/W takes care of instruction execution by coprocessor.

Structure:CPU with Arithmetic Coprocessor:-



The Control Unit:-

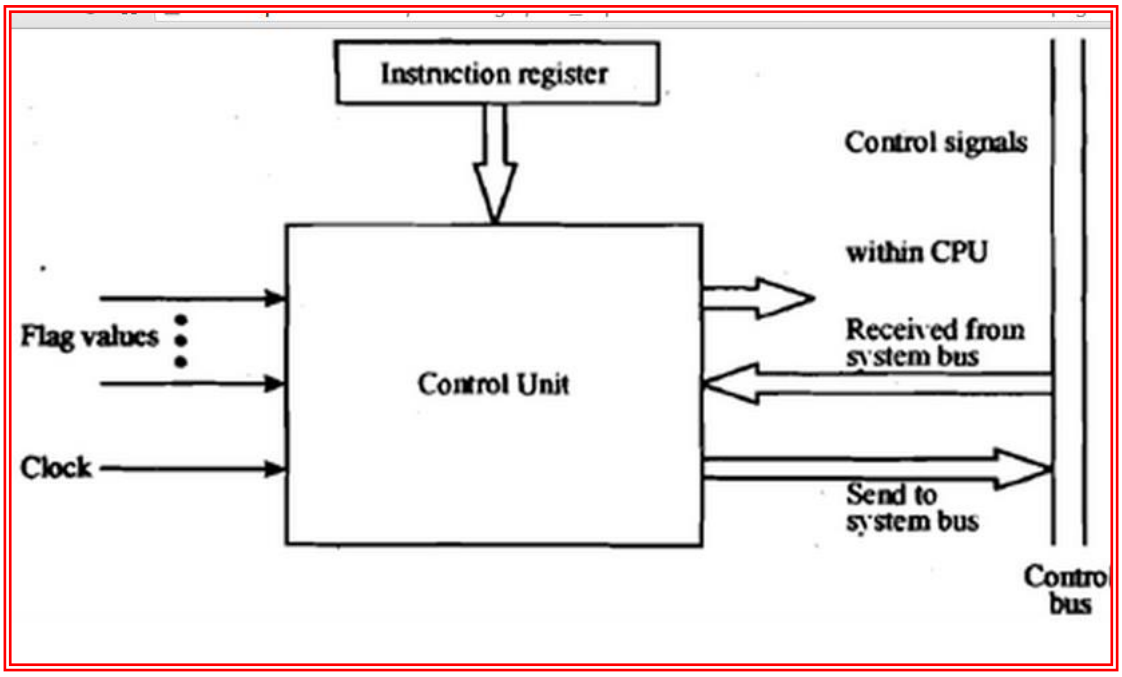
It is heart of computer which is used for controlling operations inside CPU.The basic responsibilities of the control unit are to control:

- Data exchange of CPU with the memory or I/O modules.
- Internal operations in the CPU such as:
  - Moving data between registers.
  - Making ALU to perform a particular operation on the data.
  - Regulating other internal operations.

A control unit must know about the:

- Basic Components of CPU
- Micro-Operation

Structure of Control Unit:-



Input of Control Unit:-Important for short note

- The master Clock Signal
- The Instruction register
- Flags (Stratus control of operation inside CPU)
- Control Signal From Control Bus

Fetch Cycle Inside CPU:-

T1:MAR←PC
T2:MBR←MAR
PC←PC+1
T3:IR←MBR

The Indirect Cycle Inside CPU:-`

T1:MAR←IR
T2:MBR←MAR
T3:IR(Address)←MBR(Address)

The execute Cycle Inside CPU:-

T1:MAR←IR
T2:MBR←[MAR]
T3:R1←R1+MBR

The Interrupt Cycle Inside CPU:-

T1:MAR←PC
T2:MAR←Save-address
PC←ISR
T3:[MAR]← MBR

The Instruction Cycle Inside CPU:-

00	:	Fetch
01	:	Indirect
10	:	Execute
11	:	Interrupt

The Hardwired Control Unit:-

A variety of techniques have been used to organize a control unit.Most of them fall into two major categories.

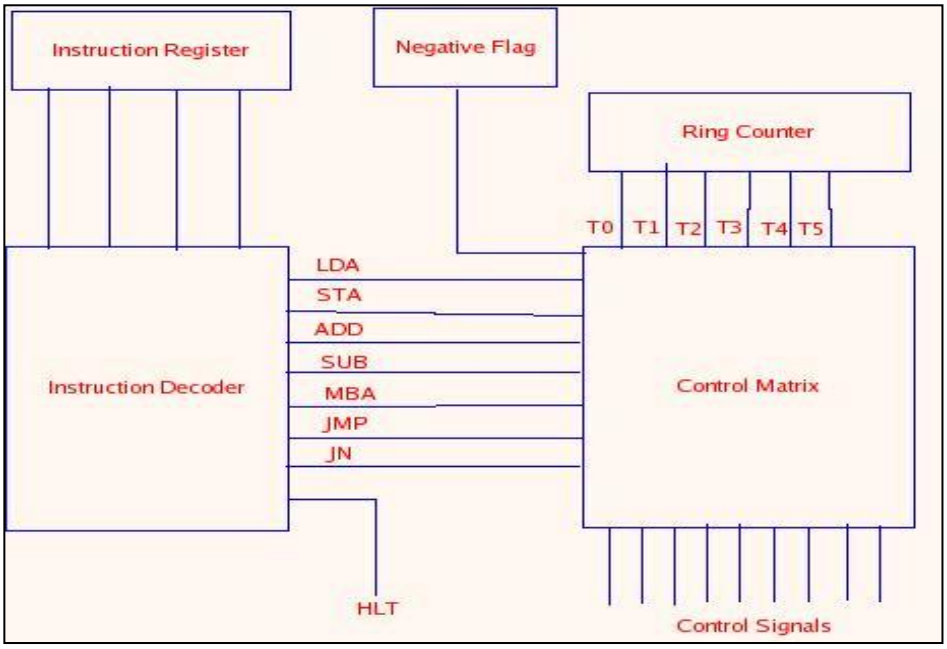
- Hardwired Control Organization
- Micro programmed control organization

In the hardwired organization, the control unit is designed as a combinational circuit.That is, The control unit is implemented by gates, flip-flops, decoder and other digital circuits.

Wilkes Control :-

Prof M.V. Wilkes of the Cambridge University Mathematical laboratory coined term microprogramming in 1951.He provided a systematic alternative procedure for designing control unit of a digital computer.It has following two components.

- Control Field, which indicates the control lines which are to be activated
- Address Field, Which provides the address of the next microinstruction to be executed.



The Execution of Microprogram in Control Unit:-

There are two categories of microinstructions are used in control unit for micro program.

- Unencoded Microinstructions
  - Large number of bits are used
  - Difficult to program
  - No decoding logic
  - Optimizes machines performance
  - Detailed hardware view
- Highly Encoded micro-instructions
  - Relatively less bits.
  - Easy to program.
  - Need decoding logic.
  - Optimizes programming effort.
  - Aggregated view.

Various Components of Computer System:-

- Motherboard (PCB)→Printed Circuit Board.
- CPU→Central Processing Unit.
- UPS →Uninterruptible Power Supply.
- SMPS→ (Switch Mode Power Supply).
- POST→(Power on Self Test).
- VDU→ (Visual Display Unit).
- MODEM (for Internet).
- USB Port(For Connecting USB devices).→Universal Serial Bus
- CPU Fan (For Cooling CPU).
- DVD Drive/CD Drive/Blu Ray Disk Drive.
- Speaker.
- Printer(For Printing).
- Scanner(For Scanning graphical document).
- LAN card for making LAN network.
- RAM And ROM.
- Hard Disk.(Secondary Memory).
- Flat Ribbon Cable.
- Web Cam.
- Mike.
- Mouse (Trackball mouse and Optical Mouse).
- Keyboard (Cherry and Membranes Keyboard).

Instruction Execution Register(Temporary memory):-

1:-MAR (MEMORY ADDRESS REGISTER)
2:-MBR (MEMORY BUFFER REGISTER)
3:-PC (PROGRAM COUNTER)
4:-IR (INSTRUCTION REGISTER)

MAR:-

It specifies the address of memory location from which data or instruction is to be accessed (for read operation) or to which the data is to be stored (for write operation).

MBR:-

It is a register, which contains the data to be written in the memory (for write operation) or It receives the data from memory (for read operation).

PC:-

It keeps track of the instruction that is to be executed next, after the execution of an on-going instruction.

IR:-

Here the instructions are loaded before the execution.

System Bus(Collection of wires or printed circuit on board):-

It is a circuit in which data transfer from:-

Input→Memory→CPU.
CPU→ Memory→Output.

Types of Buses:-

- ✓ Data Bus/Line Bus.
- ✓ Address Bus/Address line.
- ✓ Control Bus/Control Line.

Pipeline:- :- It provides a way to start a new task before an old one has been completed. This technique is called pipeline.There are following two types of pipelining technique.

1:-Instruction Pipeline 2:-Arithmetic pipeline

Fetch→Decode→Execute→Store (By means of pipelines)

Instruction Pipelines (Cycle):-

Step1 Calculate the address for next instructions.

Step2 Fetch the instruction.

Step3 Decode the operation required by the instructions.

Step4 Calculate the address for operand.

Step5 Perform the operation on the data.

Step6 Calculate the address of the operand.

Step7 Store the operand.

Interrupt:-

It is a program, which is generated by a number of sources, which may be either internal, or external to the cpu.It provides to number of sources.Some of the Internal with some event in which they may be occur.

Types of interrupt:-

- Program Interrupt(Traps)
- Time Interrupt
- I/O Interrupt
- H/W failure interrupt

Digital Logic Circuit:-

Boolean Algebra:-

It is an attempt of representing the true –false logic of humans in mathematical form.

True(T) → 1

False(F) → 0

Boolean Theorem:-

- Commuulative law
  - $AB=BA$
  - $A+B=B+A$
- Distributive law
  - $A(B+C)=AB+AC$
  - $A+(BC)=(A+B)(A+C)$
- Identity Law
  - $1.A=A$
  - $0+A=A$
- Inverse law
  - $A+\overline{A}=1$
  - $A\ \overline{A}.=0$
- Associative law:-
  - $A+(B+C)=(A+B)+C$
  - $A(BC)=(AB)C$
- Demorgan's law:-
  - $\overline{(A+B)}=\overline{A}.\overline{B}$
  - $\overline{(A.B)}=\overline{A}+\overline{B}$
- Absorption Law:-
  - $A+AB=A$
- $A+A+A+A+A\ldots=A$
- $A.A.A.A.A\ldots=A$

1:-Prove That:-

$X+1=1$

LHS

$X+1$

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

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

$=1$ RHS

2:-Simplify That:-

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

using Demorgan's law

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

$=(A+B)B$

$=AB+BB$

$=AB+B$

$=B$

3:-Simplify That:-

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

using Demorgan's law

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

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

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

$=AB$

Method For Truth Table:-

Example:-1

F=AB

Let n=input variable

Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

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

Example:-2

F=ABC

n=3(A, B, C) = 2^3=8(0, 1, 2, 3, 4, 5, 6, 7)

A	B	C	ABC
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

Example:-3

$F=\overline{AB+CD}$

n=4(A, B, C, D) = 2^4=16(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15)

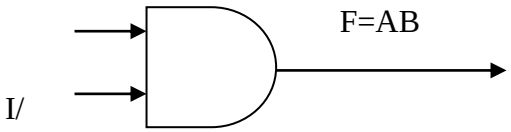
A	B	C	D	AB	CD	AB+CD	$\overline{AB+CD}$
0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	1
0	0	1	0	0	0	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	0	1
0	1	1	0	0	0	0	1
0	1	1	1	0	1	1	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1
1	0	1	0	0	0	0	1
1	0	1	1	0	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0
1	1	1	0	1	0	1	0
1	1	1	1	1	1	1	0

Logic Gates:-

It is used for making digital circuit. There are many types of gates.

- ✓ AND GATE.
- ✓ OR GATE.
- ✓ NOT GATE.
- ✓ NAND GATE.
- ✓ NOR GATE.
- ✓ EX-OR GATE.
- ✓ EX-NOR GATE.

AND GATE:-



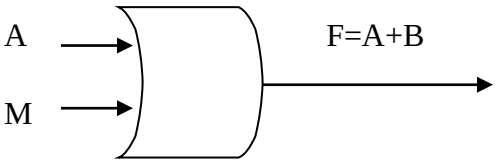
TRUTH TABLE:-

Let n=input variable
Input Signal=2^n

n=2(A, B)= 2^2=4(0, 1,2,3)

A	B	F=AB
0	0	0
0	1	0
1	0	0
1	1	1

OR GATE:-



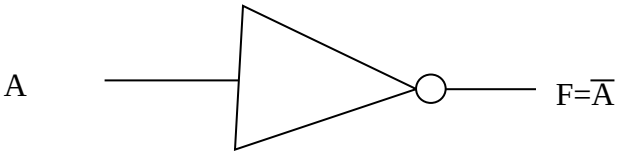
TRUTH TABLE:-

Let n=input variable
Input variable=2^n

n=2(A,B)= 2^2=4(0, 1, 2,3)

A	B	F=A+B
0	0	0
0	1	1
1	0	1
1	1	1

NOT GATE:-



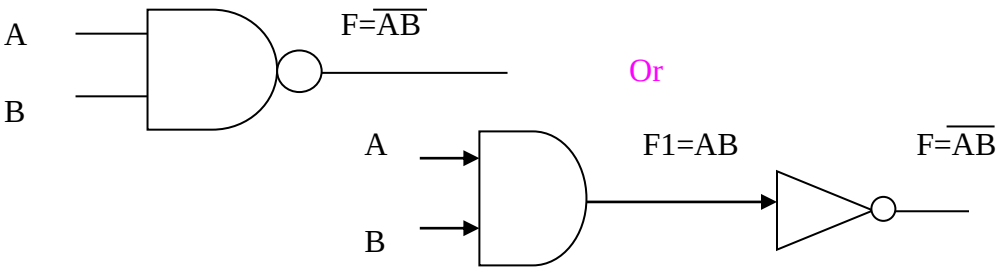
TRUTH TABLE:-

Let n=input variable
Input variable=2^n

n=1(A)= 2^1=2(0,1)

A	F=$\overline{A}$
0	1
1	0

NAND GATE:-



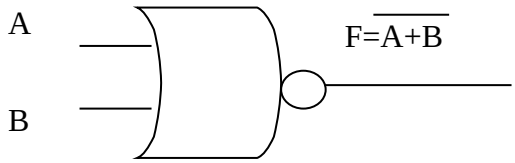
TRUTH TABLE:-

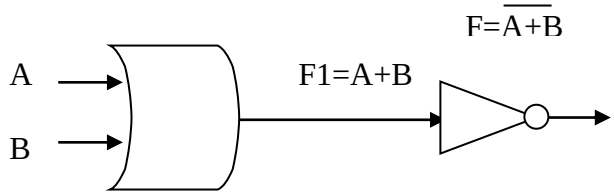
Let n=input variable
Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	F1=AB	$F=\overline{AB}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

NOR GATE:-





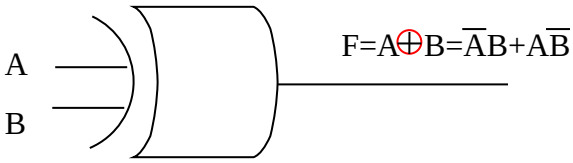
TRUTH TABLE:-

Let n=input variable
Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	F1=A+B	$F=\overline{A+B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

EX-OR GATE:-



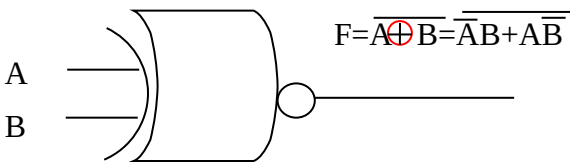
TRUTH TABLE:-

Let n=input variable
Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}$B	A$\overline{B}$	$\overline{A}$B+A$\overline{B}$
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	0	1	1
1	1	0	0	0	0	0

EX-NOR GATE:-



TRUTH TABLE:-

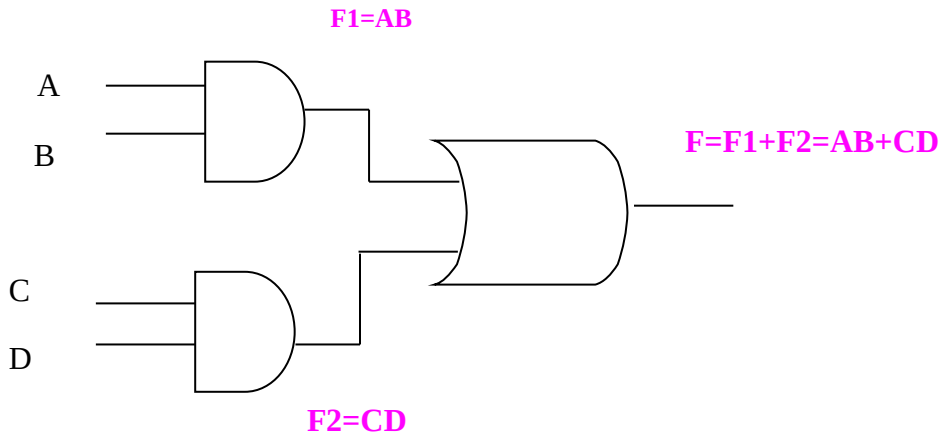
Let n=input variable
Input variable=2^n

n=2(A, B) = 2^2=4(0, 1, 2, 3)

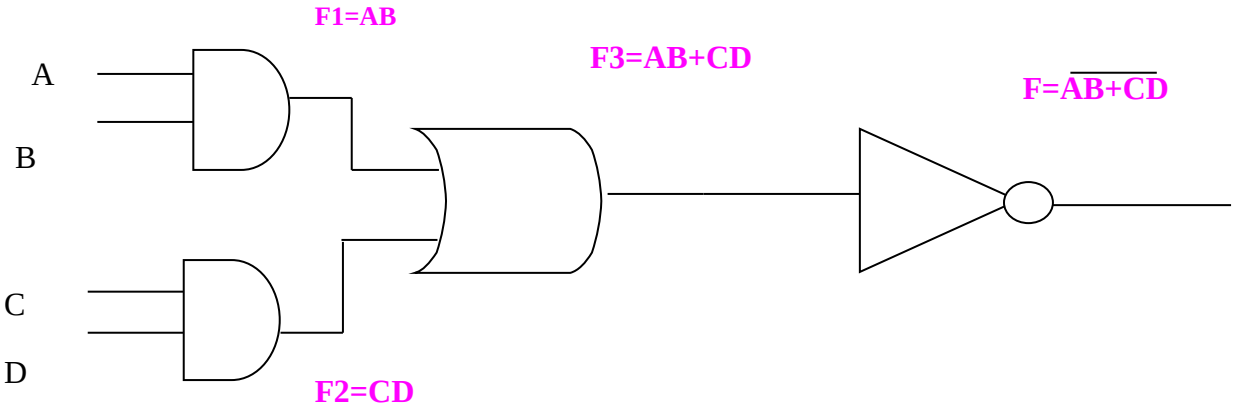
A	B	$\overline{A}$	$\overline{B}$	$\overline{A}B$	$A\overline{B}$	$\overline{A}B+A\overline{B}$	$\overline{\overline{A}B+A\overline{B}}$
0	0	1	1	0	0	0	1
0	1	1	0	1	0	1	0
1	0	0	1	0	1	1	0
1	1	0	0	0	0	0	1

Combinational Circuit:-

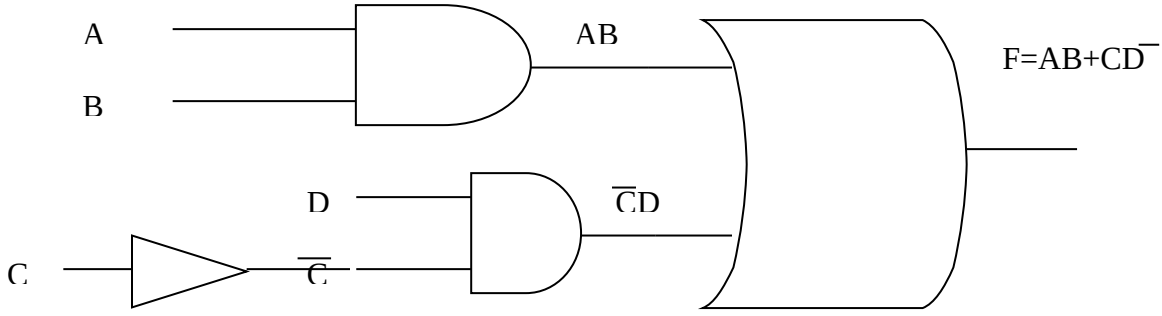
F=AB+CD



F=$\overline{AB+CD}$



F=AB+$\overline{C}$D



Simplification of Digital Circuit:-

- Algebraic Simplification.
- Karnaugh Maps.

Karnaugh Maps:-

- SOP ∑ (SUM OF PRODUCT) → Min term
- POS ∏ (Product OF SUM) → Max term

Table For SOP (MIN TERM) ∑ :-

- For variable Two

	$\overline{A}$	A
$\overline{B}$	0	1
B	2	3

- Table for variable Four

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

Table For variable Five

$\overline{M}$=0

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

M=1

	$\overline{A}\,\overline{B}$	$\overline{A}\,B$	$A\,B$	$A\,\overline{B}$
$\overline{C}\,\overline{D}$	16	17	19	18
$\overline{C}\,D$	20	21	23	22
$C\,D$	28	29	31	30
$C\,\overline{D}$	24	25	27	26

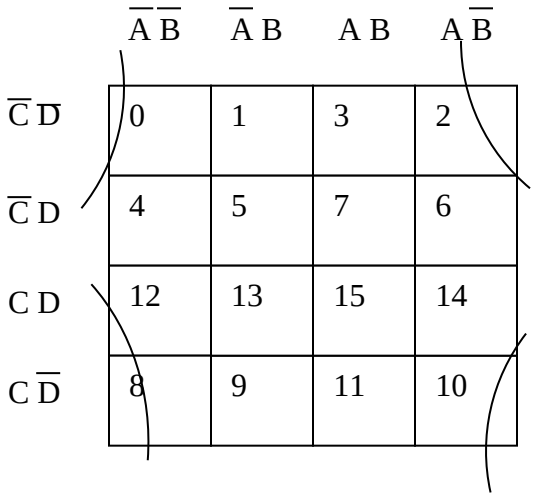
How to design Digital Circuit by Using K-Map

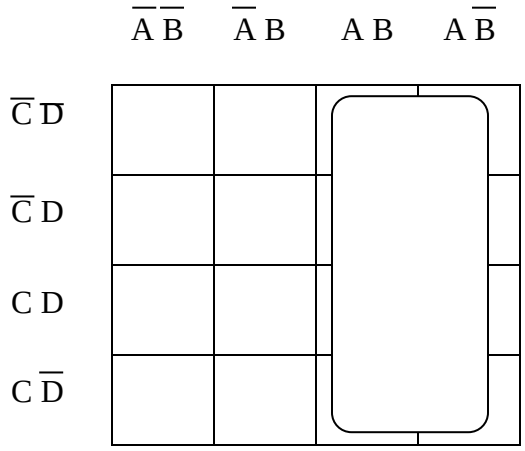
Making possible pairing of numbers in k-map table:-

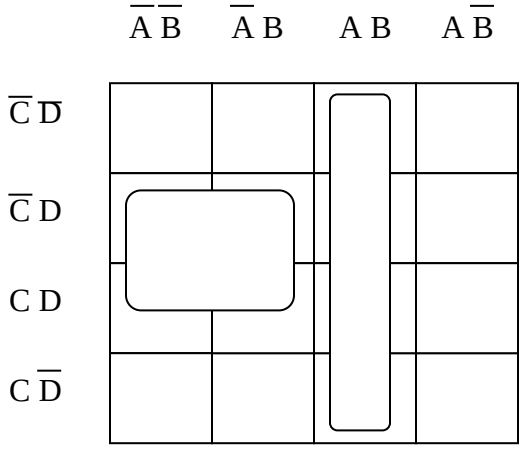
$2^n \rightarrow$n indicate number of variables
Where n=0, 1, 2, 3,4,5,6…
2^0=1 Possible pair
2^1=2 Possible pair
2^2=4 Possible pair
2^3=8 Possible pair
2^4=16 Possible pair
2^5=32 Possible pair
…..

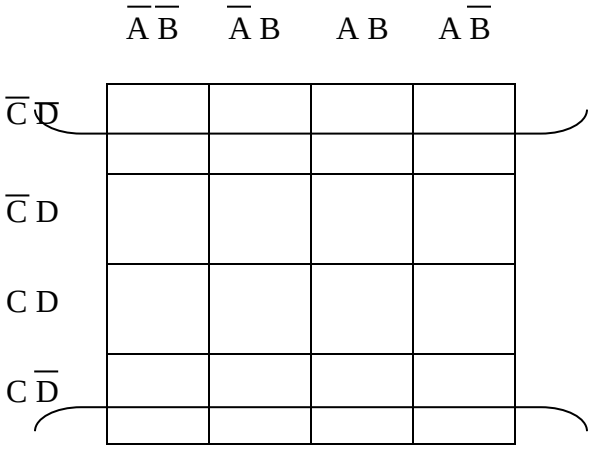
Pairing Rules in k-map table:-

- ✓ Diagonal pairs not allowed.
- ✓ Always consider the largest number of pairs such as 2, 4, 8, 16, 32….
- ✓ Corner pairs should first consider.




$\overline{A}\,\overline{B}$
$\overline{A}\,B$
$A\,B$
$A\,\overline{B}$
$\overline{C}\,\overline{D}$
$\overline{C}\,D$
$C\,D$
$C\,\overline{D}$


$\overline{A}\,\overline{B}$
$\overline{A}\,B$
$A\,B$
$A\,\overline{B}$
$\overline{C}\,\overline{D}$
$\overline{C}\,D$
$C\,D$
$C\,\overline{D}$


$\overline{A}\,\overline{B}$
$\overline{A}\,B$
$A\,B$
$A\,\overline{B}$
$\overline{C}\,\overline{D}$
$\overline{C}\,D$
$C\,D$
$C\,\overline{D}$

Combinational Circuit:-

Adders:-

A combinational circuit which performs addition of two bits is called half adders, While combinational circuit which performs arithmetic addition of three bits is called a full adder.

Definition of half adder:-

It produces sum and carry in following way.

$S=\overline{x}y+x\overline{y}$
$C=xy$

Truth Table:-

X	Y	Carry	Sum
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

	$\overline{Y}$	y
X		1
$\overline{X}$	1	

Circuit of half adder:-

Page no:-74

Definition of Full adder:-

It produces sum and carry using three bits in following way.

$S=\overline{x}\,\overline{y}\,p+x\,\overline{y}\,\overline{p}+\overline{x}\,y\,\overline{p}+\overline{x}\,\overline{y}\,\overline{p}$
$C=xp+xy+yp$

Circuit of full adder:-

Page no:-75

Subtractor:- Most Important

Subtractor is the one which used to subtract two binary number and provides Difference and Borrow as a output.basically we have two types of subtractor.

1. Half Subtractor
2. Full Subtractor

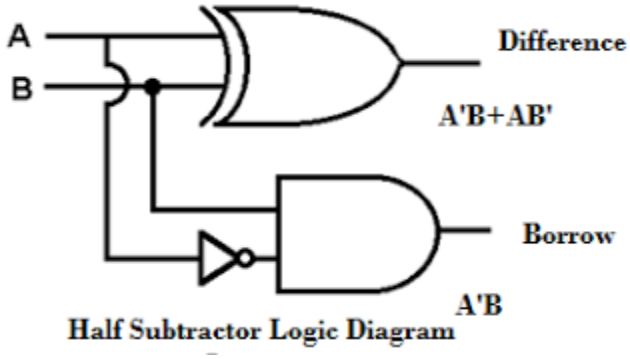
Half Subtractor :

Half Subtractor is used for subtracting one single bit binary number from another single bit binary number.The truth table of Half Subtractor is shown below.

Difference = $A'B+AB'=A\oplus B$
Borrow=A'B

Half Subtractor-Truth Table			
Input		Output	
A	B	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

The logic Diagram of Half Subtractor is shown below:-



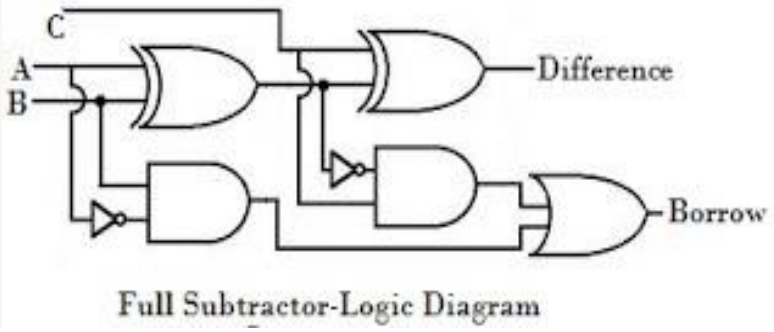
Half Subtractor Logic Diagram

Full Subtractor: A logic Circuit which is used for Subtracting Three Single bit Binary numbers is known as Full Subtractor.The Truth Table of Full Subtractor is Shown Below.

Full Subtractor-Truth Table				
Input			Output	
A	B	C	Difference	Borrow
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

From the Truth Table The Difference and Borrow will written as
Difference=A'B'C+A'BC'+AB'C'+ABC
Reduce it like adder
Then We got
Difference=A$\oplus$B$\oplus$C
Borrow=A'C+A'B+BC

The logic diagram of Full Subtractor is Shown below



Full Subtractor-Logic Diagram

Decoder:-

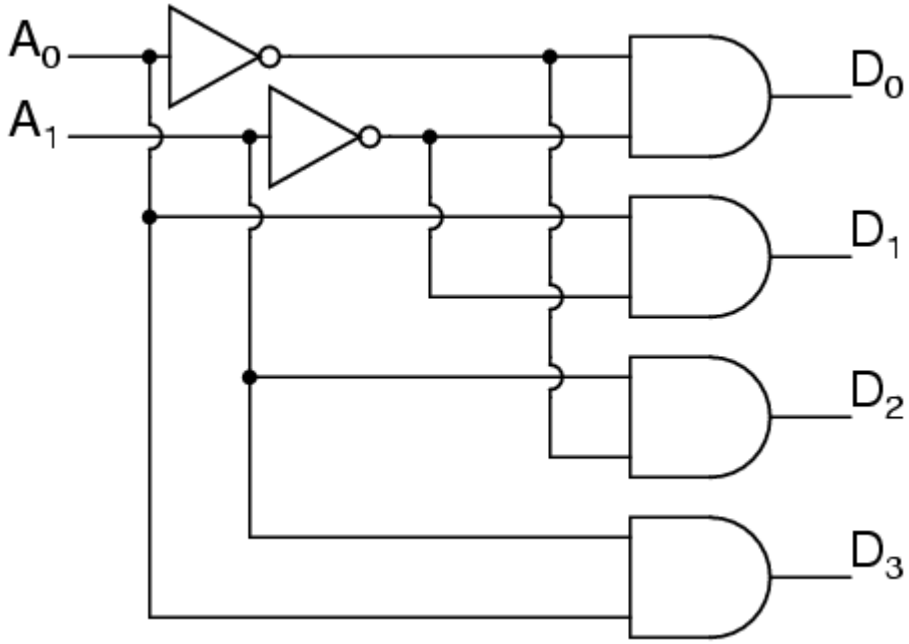
It converts one type of coded information to another form. A decoder has 'n' inputs and an enable line(s sort selection line) and 2^n output lines.

Or

A decoder is a circuit that changes a code into a set of signals. It is called a decoder because it does the reverse of encoding, but we will begin our study of encoders and decoders with decoders because they are simpler to design.

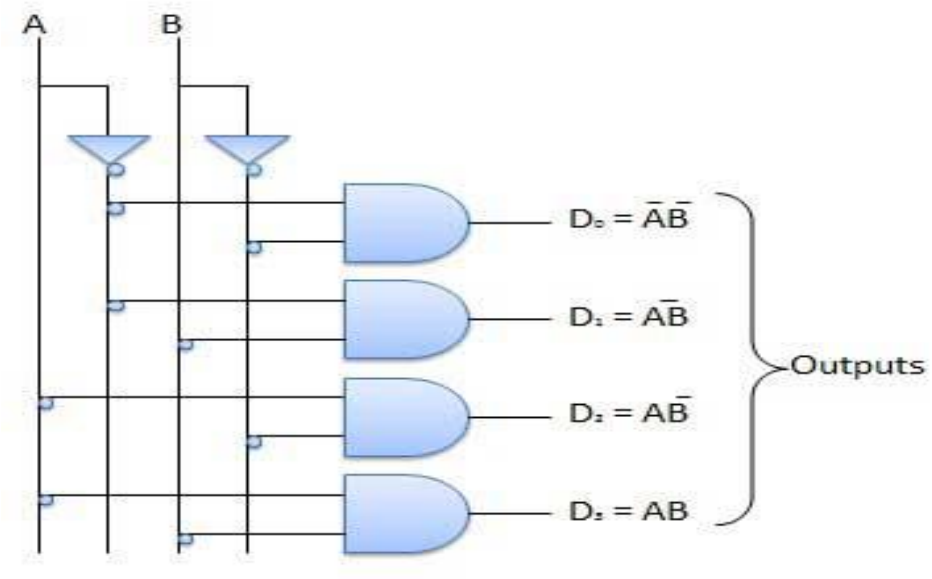
Example:-Possible decoder

n=1	⇨2^1=2	⇨1*2 Decoder
n=2	⇨2^2=4	⇨2*4 Decoder
n=3	⇨2^3=8	⇨3*8 Decoder
n=4	⇨2^4=16	⇨4*16 Decoder



Encoder:-

An encoder is a device which transforms the data into some bits known only to it and the decoder is a device which transforms those coded bits to generate the original data again. These two are mainly used in computer technology but the underlying concept can be used anywhere



Multiplexer:-

It is one of the basic building units of a computer system which in principle allows sharing of a common line by more than one input lines.It is defined by 2^n*1.

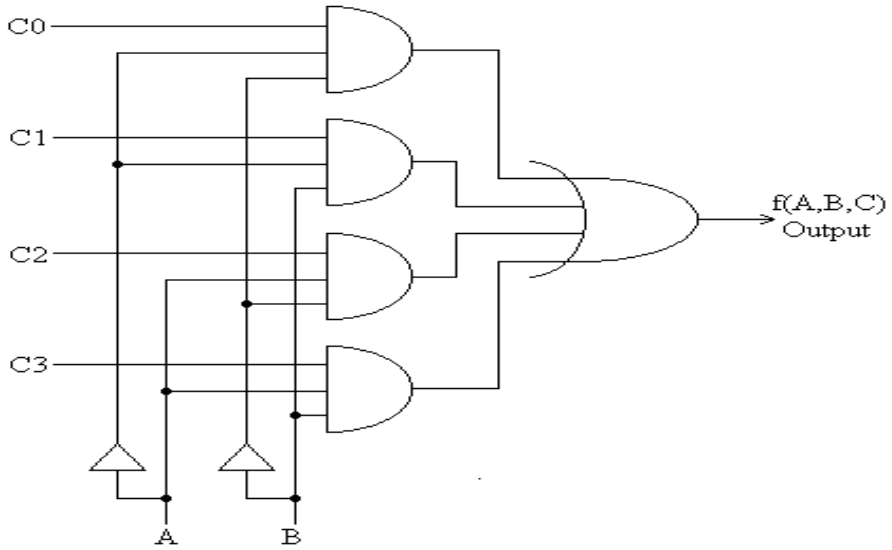
Or

A multiplexer performs the function of selecting the input on any one of 'n' input lines and feeding this input to one output line.

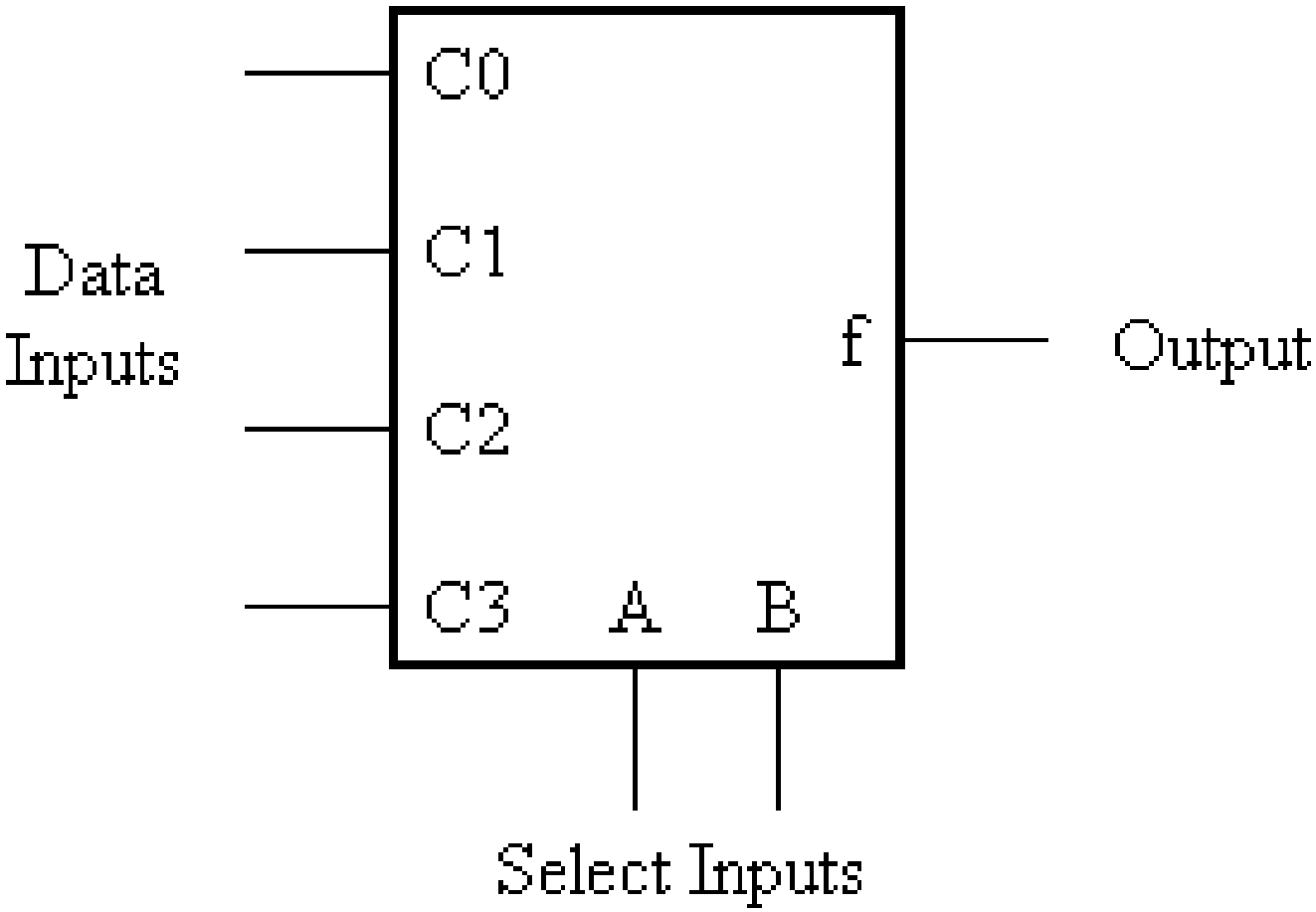
Example:-Possible Multiplexer

n=1 ⇨2^1*1=2 ⇨2*1 MUX
n=2 ⇨2^2*1=4 ⇨4*1 MUX
n=3 ⇨2^3*1=8 ⇨8*1 MUX

Assume that we have four lines, ***C_0, C_1, C_2 and C_3***, which are to be multiplexed on a single line, ***Output (f)***. The four input lines are also known as the ***Data Inputs***. Since there are four inputs, we will need two additional inputs to the multiplexer, known as the ***Select Inputs***, to select which of the ***C*** inputs is to appear at the output. Call these select lines ***A and B***.
The gate implementation of a 4-line to 1-line multiplexer is shown below:-



The circuit symbol for the above multiplexer is:


C0
C1
C2
C3
A
B
f
Data
Inputs
Output
Select Inputs

Encoders:-

An encoder performs the reverse function of the decoder.An encoder has 2^n input lines and 'n' out put lines.

Example:-Possible Encoder

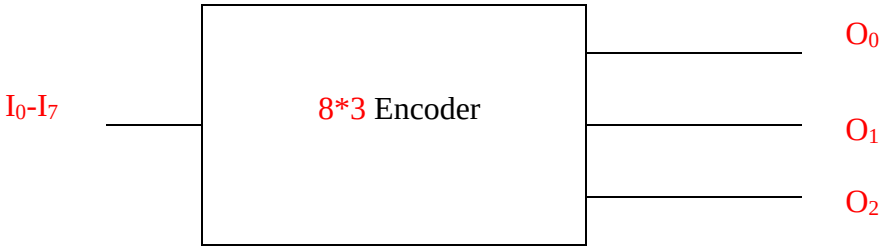
2^n*n

$2^1*1=2*1$ Encoder

$2^2*2=4*2$ Encoder

$2^3*3=8*3$ Encoder

$2^4*4=16*4$ Encoder



I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7		O_2	O_1	O_0
1	0	0	0	0	0	0	0	D_0	0	0	0
0	1	0	0	0	0	0	0	D_1	0	0	1
0	0	1	0	0	0	0	0	D_2	0	1	0
0	0	0	1	0	0	0	0	D_3	0	1	1
0	0	0	0	1	0	0	0	D_4	1	0	0
0	0	0	0	0	1	0	0	D_5	1	0	1
0	0	0	0	0	0	1	0	D_6	1	1	0
0	0	0	0	0	0	0	1	D_7	1	1	1

$O_0=I_1 + I3 + I_5 + I_7$

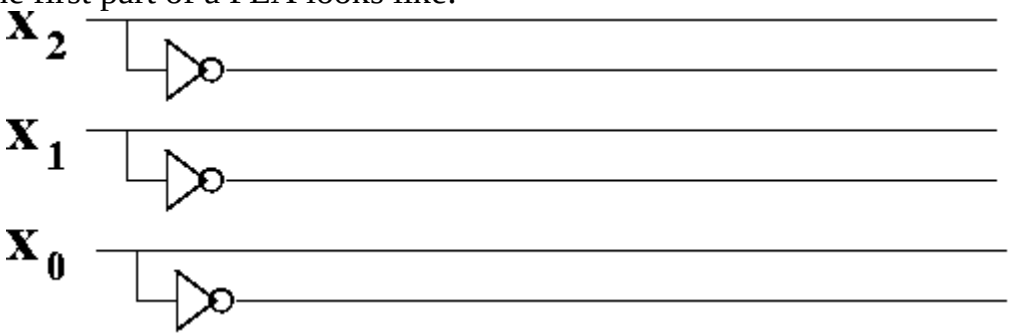
$O_1=I_2 + I_3 + I_6 + I_7$

$O_2=I_4 + I_5+ I_6 + I_7$

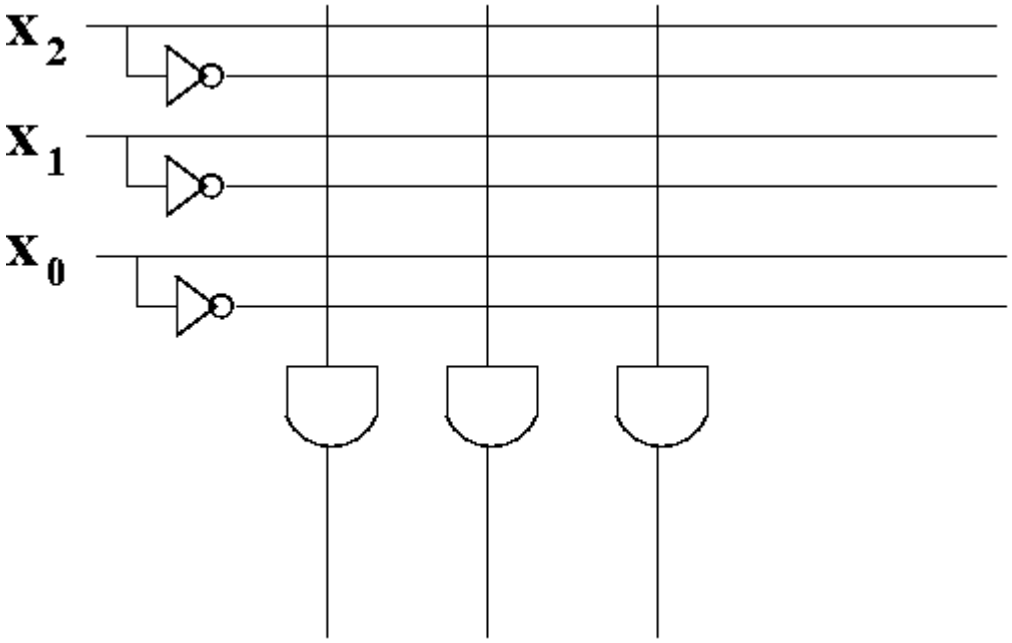
PLA (Programmable Logic Array):-

It is defined for SOP (**Σ**) form for Boolean function and consist of regular arrangements of NOT, AND & OR gate on a chip. Each input to the chip is passed through a NOT gate, Thus the input and its compliment are available to each OR gate and the output of each OR gate is treated as chip output. Implementation of such logic circuit is called PLA.

The first part of a PLA looks like:-



The next part is to draw a vertical wire with an AND gate. I've drawn 3 of them.



Let's try to implement a truth table with a PLA.

x_2	x_1	x_0	z_1	z_0
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	0
1	1	0	0	0
1	1	1	0	1

Each of the vertical lines with an AND gate corresponds to a minterm. For example, the first AND gate (on the left) is the minterm: $\backslash x_2 \backslash x_1 x_0$.

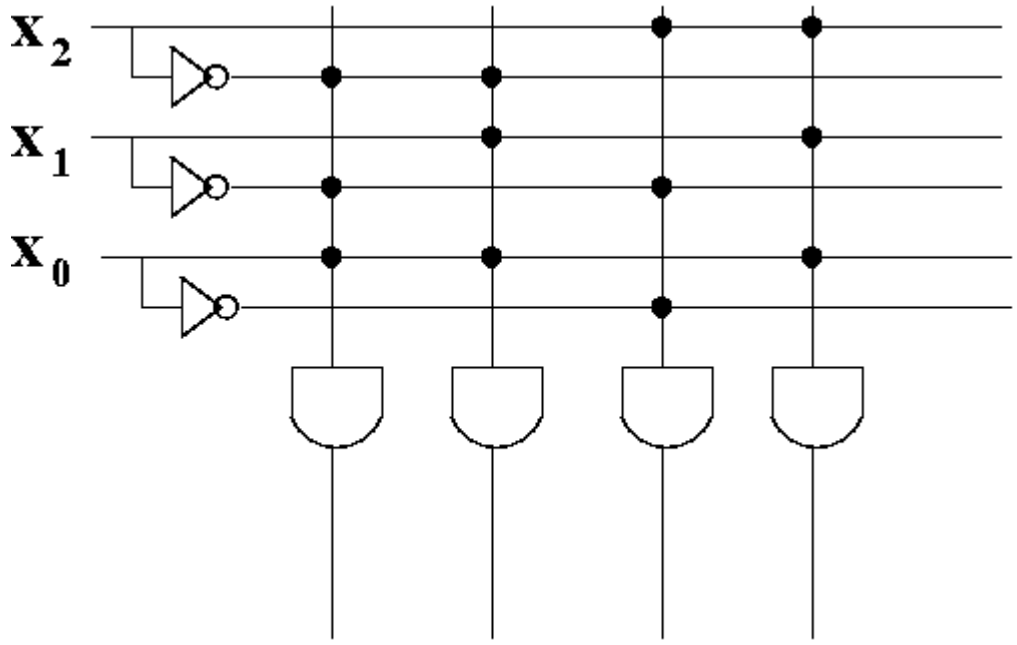
The second AND gate (from the left) is the minterm: $x_2 x_1 x_0$.

The third AND gate (from the left) is the minterm: $x_2 x_1 x_0$.

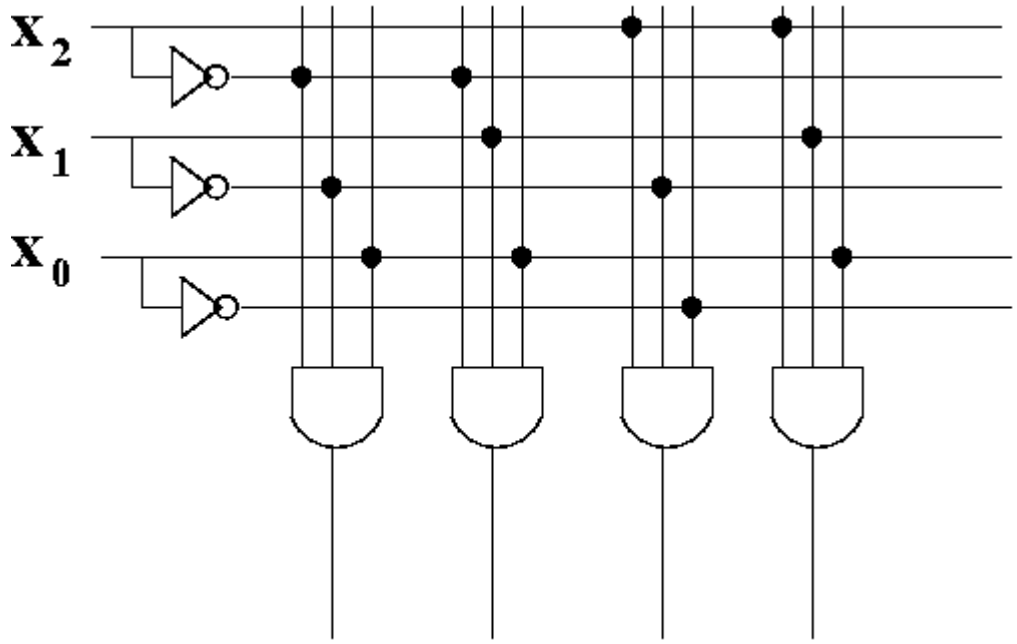
I've added a fourth AND gate which is the minterm: $x_2 x_1 x_0$.

The first three minterms are used to implement z_1. The third and fourth minterm are used to implement z_0.

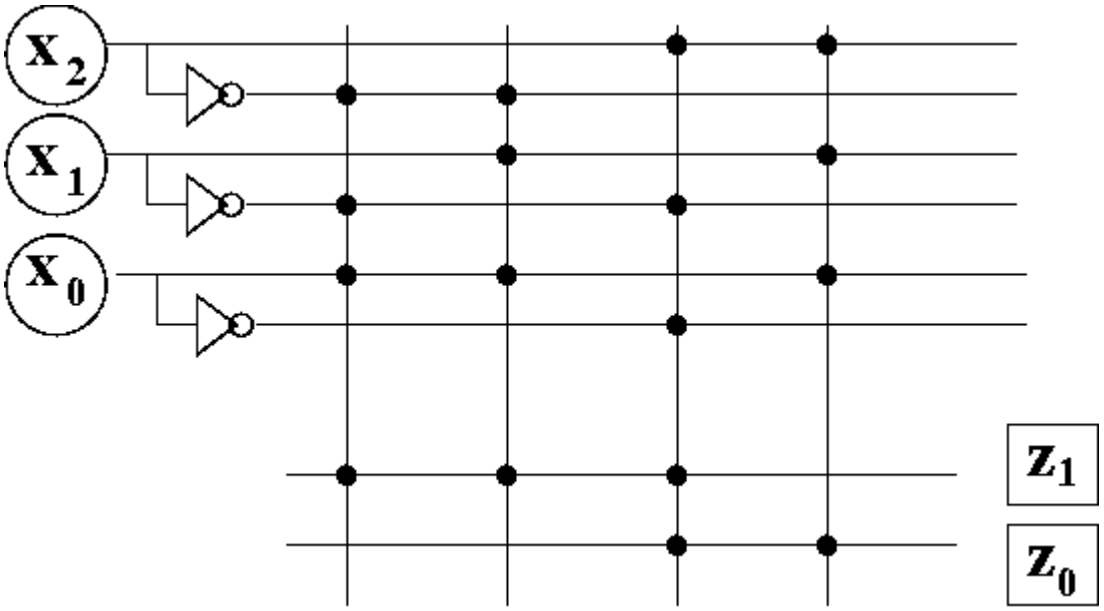
This is how the PLA looks after we have all four minterms.



We draw one wire just to make it look neat.



his is how the PLA looks when we leave out the AND gates and the OR gates. It's not that the AND gates and OR gates aren't there---they are, but they've been left out to make the PLA even easier to draw.



Sequential Circuit:-

PORT:-

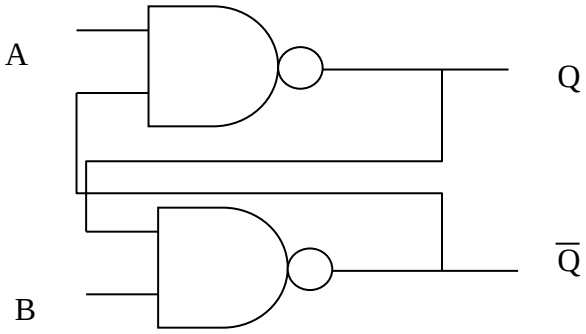
Serial port	For Connecting Mouse /modem.
Parralel Port	For Connecting Printer.
USB	Universal Serial bus for connecting both types of devices.

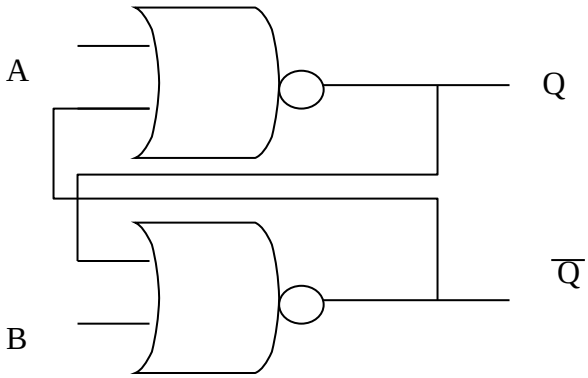
Sequential Circuit:-

It is an interconnection of combinational circuits and storage elements. The storage elements, called flip-flops.

Flip-Flop:-

It is a binary cell, which stores a bit of information. It consists of either two NAND gates or two NOR Gates.

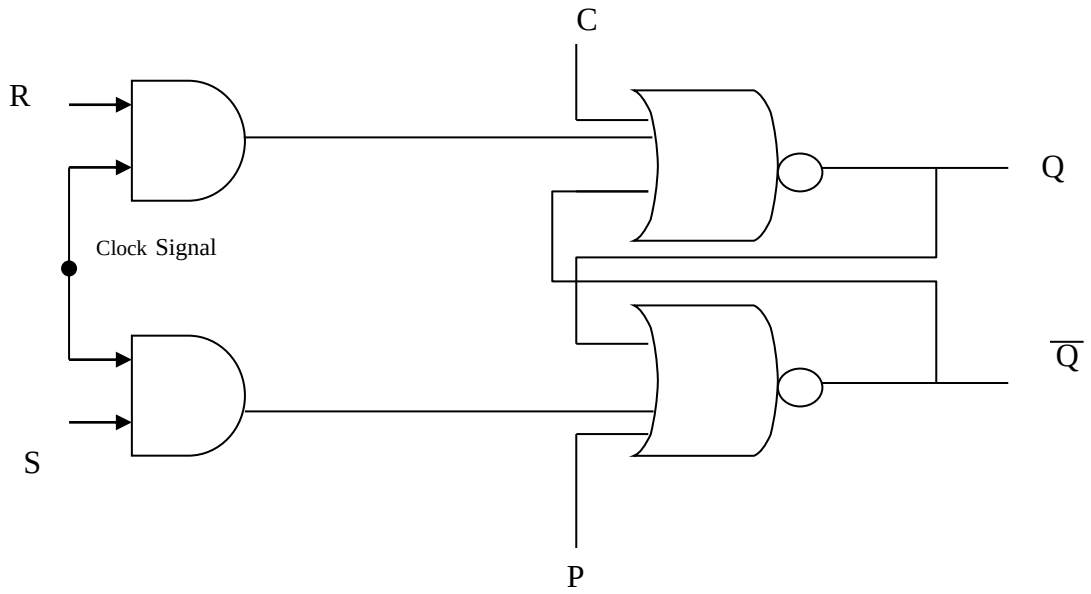




Types of Flip Flops:-

- R-S(Reset-Set) flip flops.
- D (Data)flip flops.
- J-K flip flops.
- T(Toggle) flip flops.
- Master-Slave flip flops using J-K flip flops.
- Trigger flip –flops.

R-S flip-flops:- An **SR Flip Flop** is an arrangement of logic gates that maintains a stable output even after the inputs are turned off. This simple **flip flop** circuit has a set input (S) and a reset input (R). The set input causes the output of 0 (top output) and 1 (bottom output).



S	R	State on Complete at clock Cycle
0	0	No Change in State
0	1	Clear the Flip Flop (State 0)
1	0	Set Flip Flop (State 1)
1	1	Not Allowed

Explanation of R-S flip Flop:-

A:-If no clock signal C=0 then output can not change irrespective of R &S values.

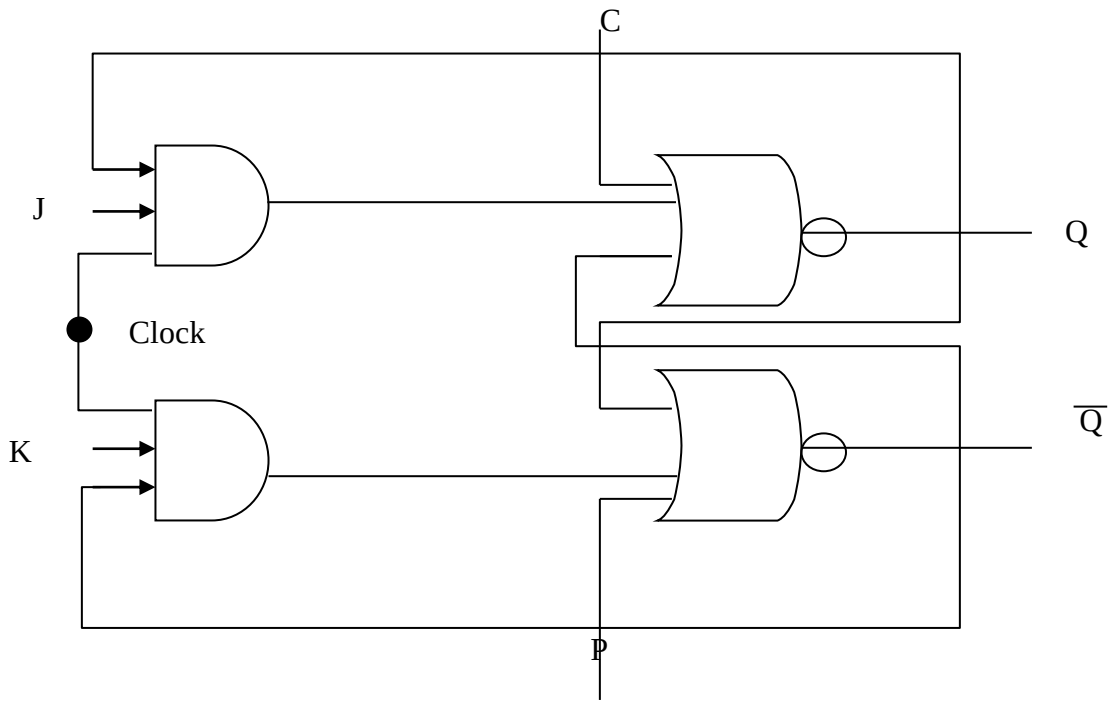
B:-If clock signal changes from 0 to 1 and S=1 & R=0 Then output Q=1 and $\overline{Q}$=0 (Set State).

C:-If clock signal changes from 0 to 1 and S=0 & R=1 Then output Q=0 and $\overline{Q}$=1 (Reset State).

D:-During positive clock transition if both R=1 & S=1 then output is not defined.

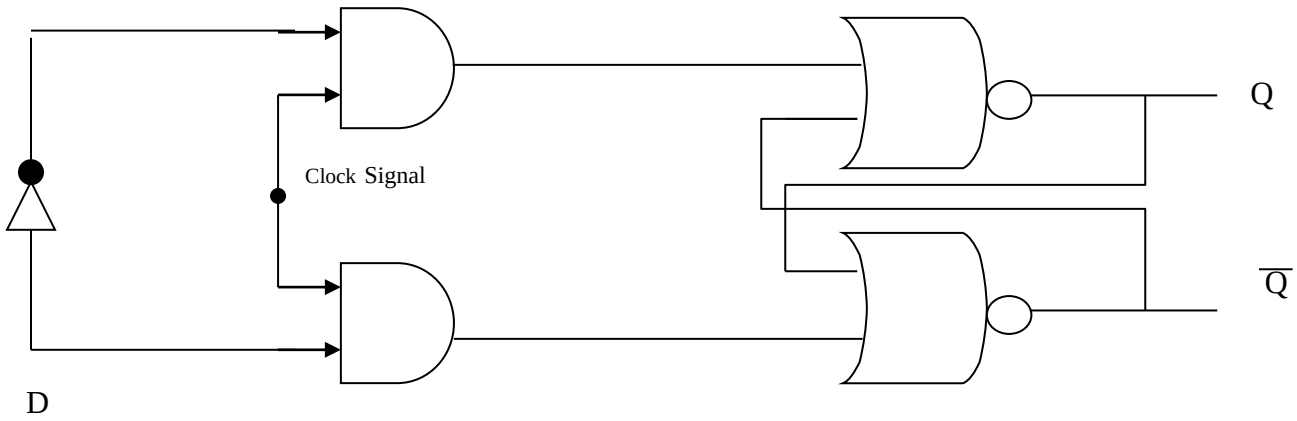
J-K flip-flops:-

The **JK flip flop** is basically a gated SR **flip-flop** with the addition of a clock input circuitry that prevents the illegal or invalid output condition that can occur when both inputs S and R are equal to logic level "1".



J	K	State on Complete at clock Cycle
0	0	No Change In State
0	1	Clear The Flip Flop (State 0)
1	0	Set Flip Flop (State 1)
1	1	Compliment the state of flip flops

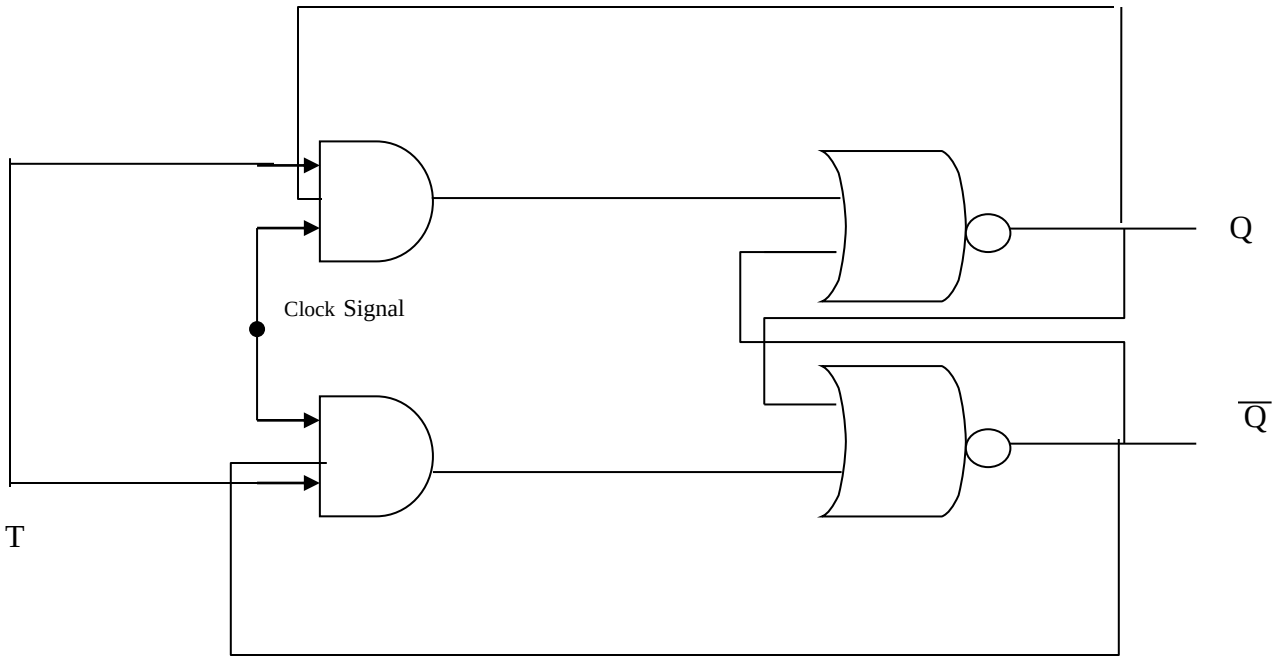
D (Data) flip-flops :-(It is modification of RS flip-flop . The problem of undefined output in RS flip flop when both R and S become 1 avoided in D flips flop)



Truth table:-

D	Output	
0	0	Clear
1	1	Set

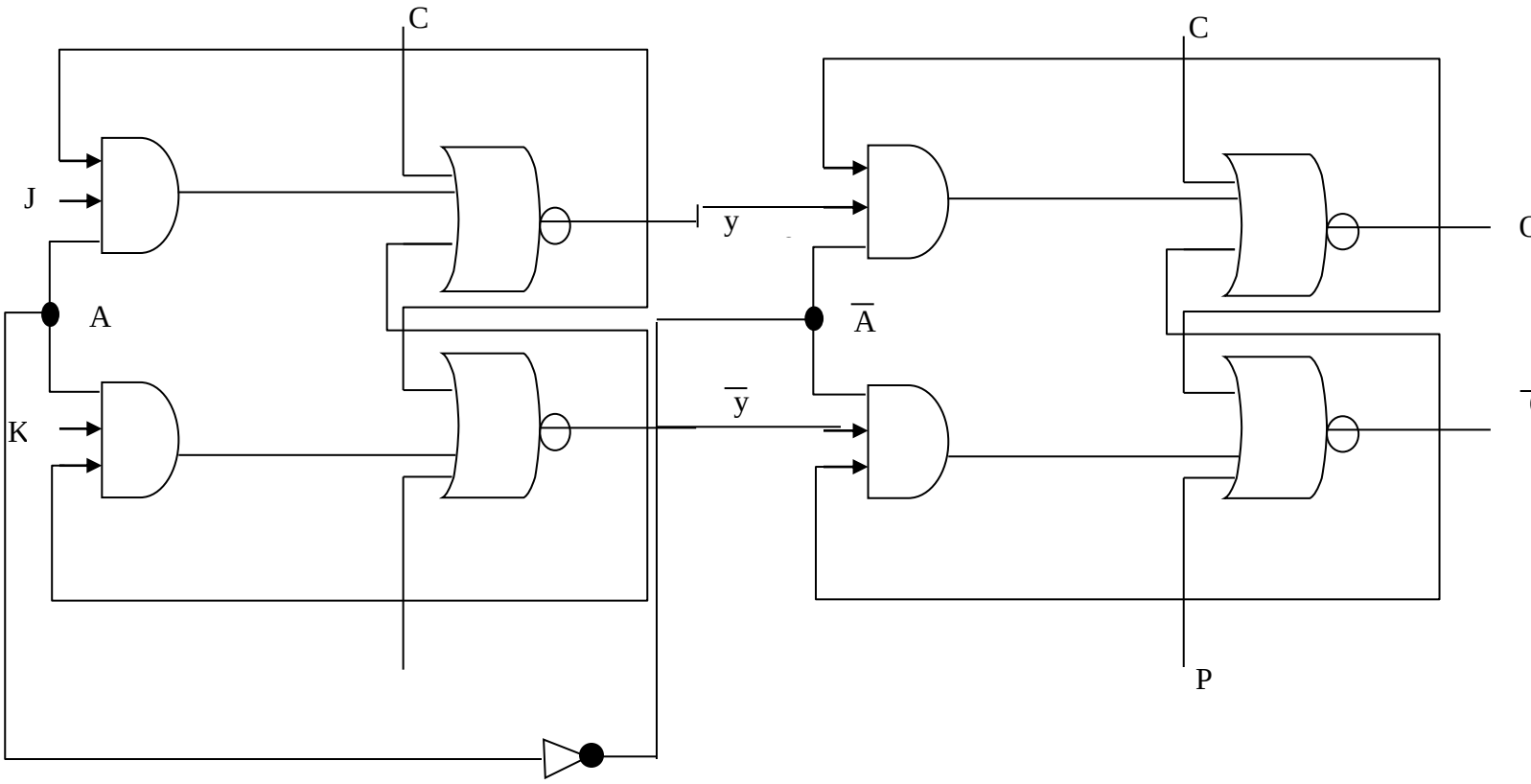
T (Toggle) flip-flops:- (It is obtained from joining inputs j and k together).



T	Output
0	No change
1	Compliment

Master-Slave flip-flops:-

It consists of two flip-flops. One is the master & other is called the slave flip flop. This Flip-flop shown in following figures.



Master Slave

Note:-Master-slave flip flop can be constructed using D or RS flip flop in the same manner.

Explanation:-

State Flip-flop:-

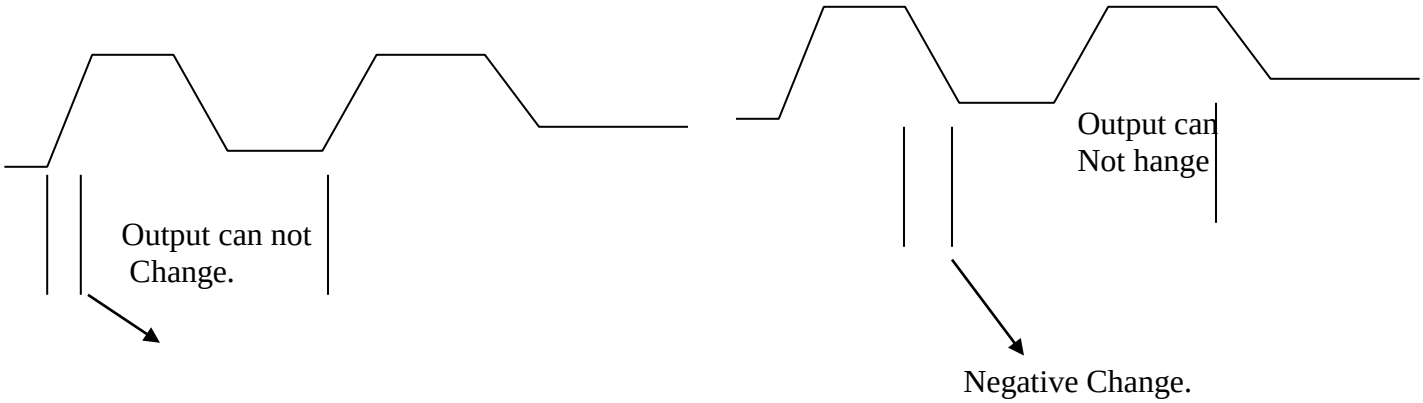
J=1, K=0 It means Q=1 and $\overline{Q}$=0

Clear Flip-flop:-

J=0, K=1 It means Q=0 and $\overline{Q}$=1

Edge Triggered Flip-Flops:-

It is used to synchronize the change during a clock pulse transition instead of constant level.



Positive Transition.

Example of **Sequential Circuit Design**:-

- ✓ Register
- ✓ Counter

In sequential circuit it is specified by a time sequence of external inputs, external outputs and internal flip-flop binary states.

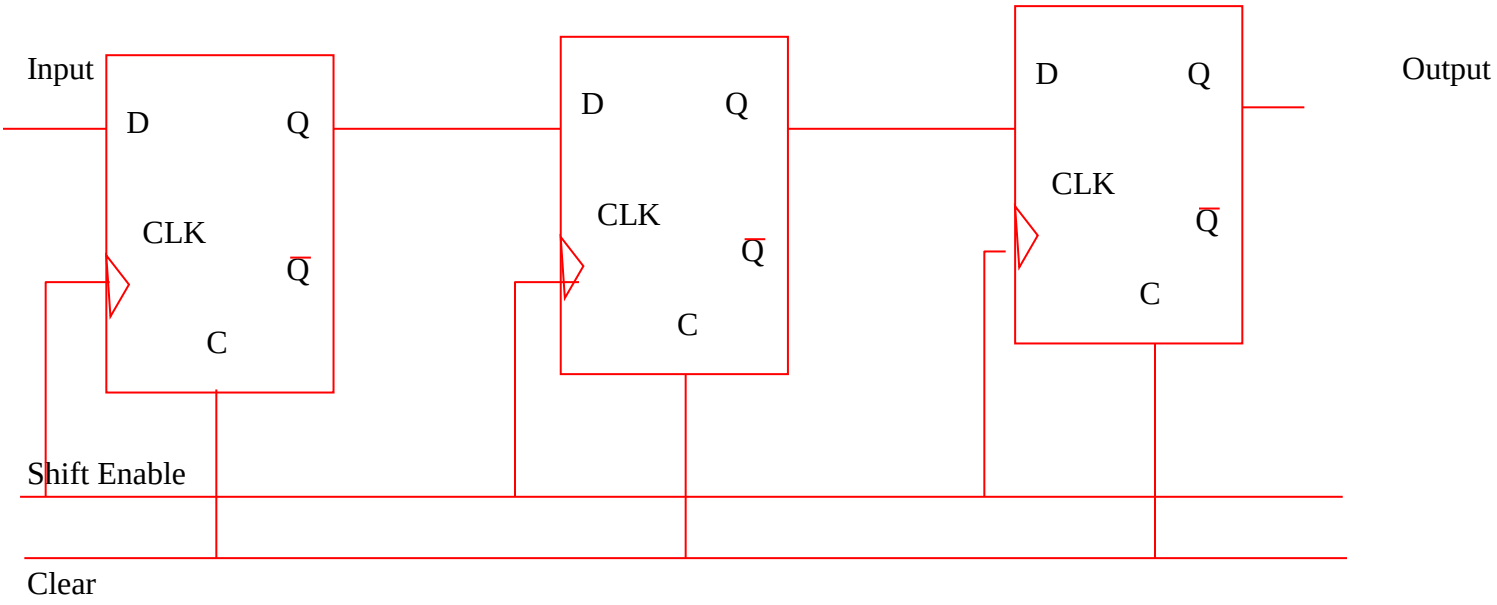
Registers:-

It is a group of flip-flops, which store binary information and gates. An n-bits register has n flip-flops and stores n-bits of binary information. There are two types of registers.

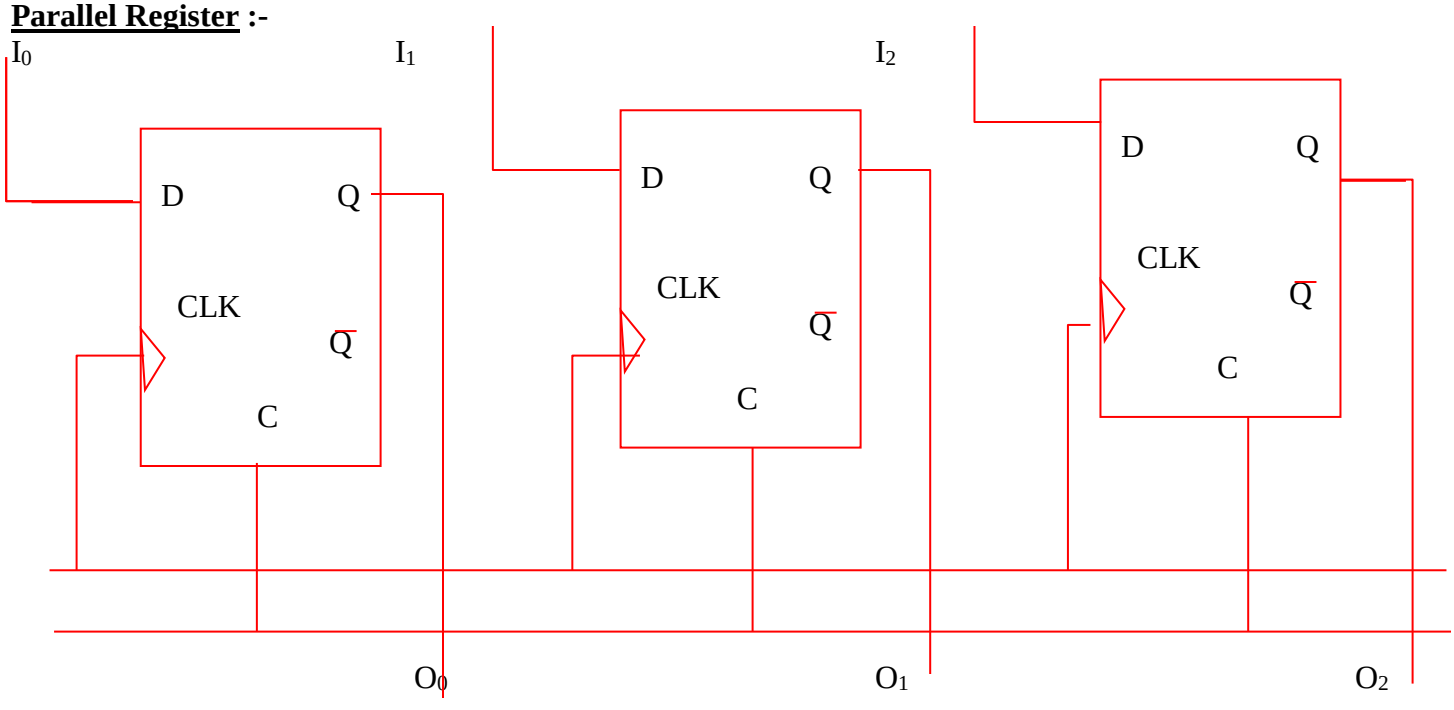
Types of registers:-

- ✓ Serial register.
- ✓ Parallel register.

Serial Register:-



Parallel Register :-

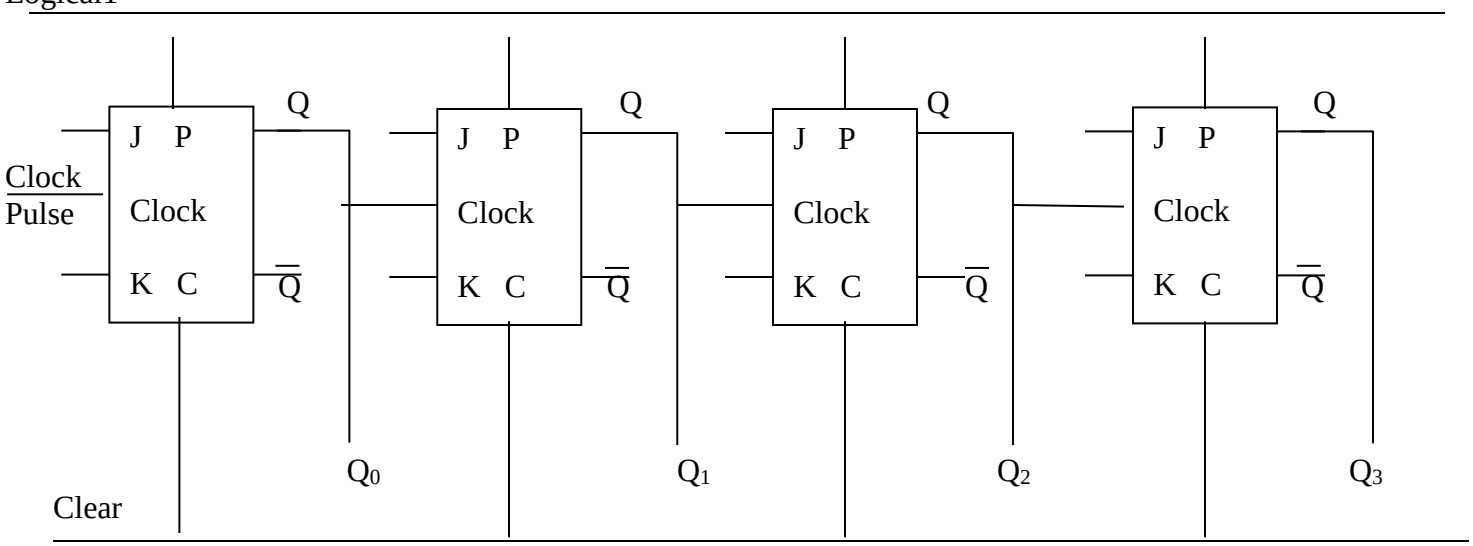


Counter:-

It is a sequential circuit in which value is incremented by one on the occurrence of some event.It is used for counting the number of times of an event occurs and are useful for generating the timing signals for controlling the sequence of operations in digital computers.There are two types of counters.

- Asynchronous Counter
- Synchronous Counter

Example of 4-bit ripple counter

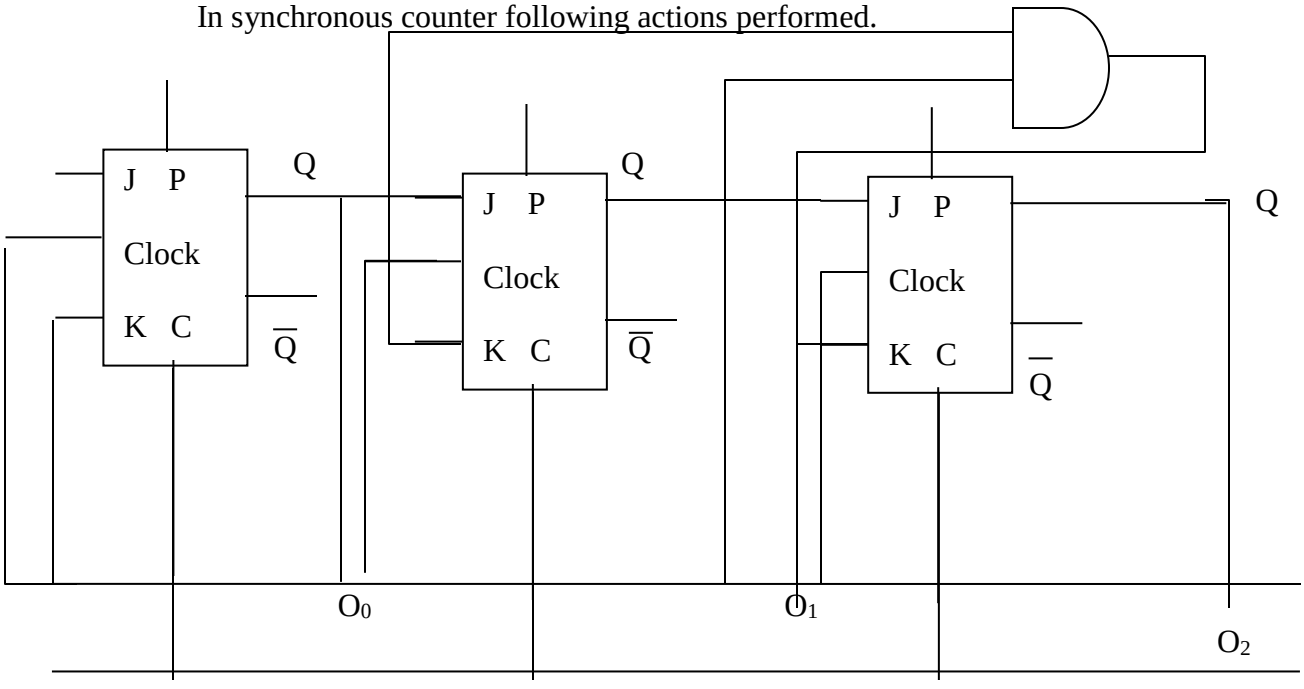


Asynchronous Counter:-

It is termed as ripple counter, since the change that occurs in order to increment the counter ripples through the counter from one end to other.Clock pulse count from 0 to 15.

Synchronous Counter:-

In synchronous counter following actions performed.



- The first flip-flop is always complemented.
- The second flip flop is complimented in the next clock pulse if the current state of the first flip flop is set (One).
- The third flip flop is fed by AND gate which is connected with the output of the first and second flip-flops.
- Here master slave flip flop used for changing the state simultaneously.

Truth Table:-

O_0	O_1	O_2
0	0	0
1	0	0
0	0	0
0	1	0
1	1	0
0	0	1
1	1	1
0	1	1
1	1	1
0	0	0

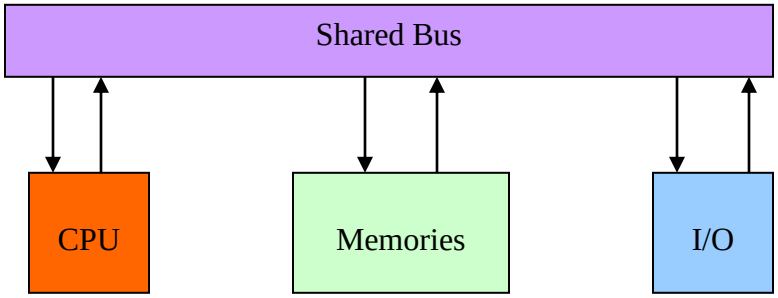
Interconnection Structures:-

A computer consist of three basic components.

- CPU.
- Memory.
- Input/Output Components.

Interconnection structure must support the transfer of:-

- An instruction or a unit of data from memory to CPU.
- A unit of data from CPU to memory.
- Reading of data from I/O device.
- CPU sending data to input/output device.
- Memory to input/output device.
- Input/output device to memory.



Types of Buses

There are three types of buses are used for connecting component of computer.

- Data Lines/Data Bus.
- Address Lines/Address Bus.
- Control Lines/Control Bus.

Buses normally consist of 8, 16 or 32 bits.

Aspects Related to Buses:-

Bus arbitration:-

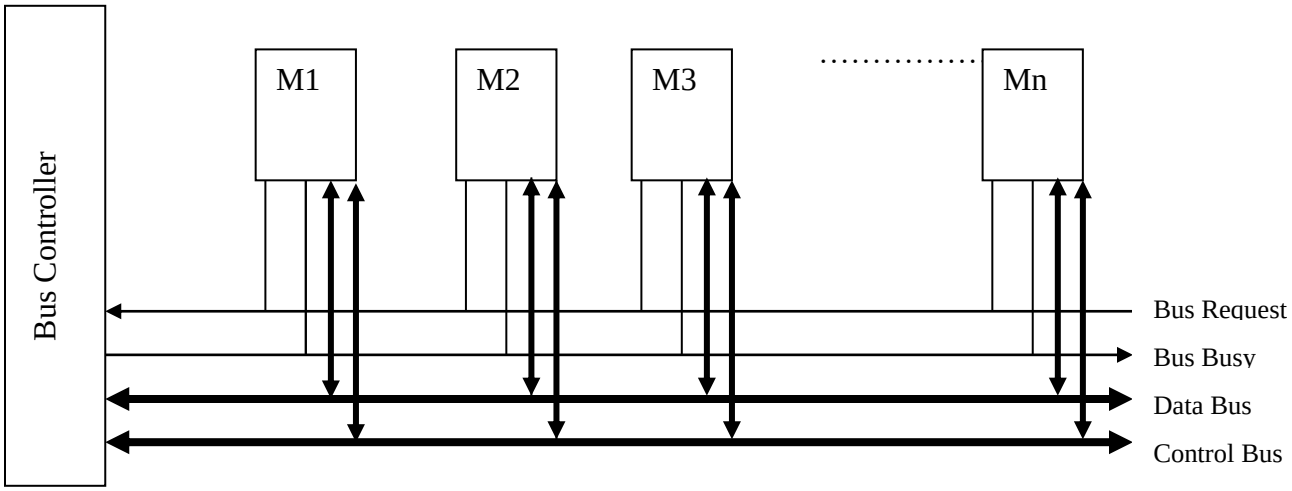
A bus arbiter is a device used in a multi-master bus system to decide which bus master will be allowed to control the bus for each bus cycle. The most common kind of bus arbiter is the memory arbiter in a system bus system.

- Dedicated or Multiplexed buses:- A dedicated bus line assigned permanently to a function or to a physical subset of the components of the computer.

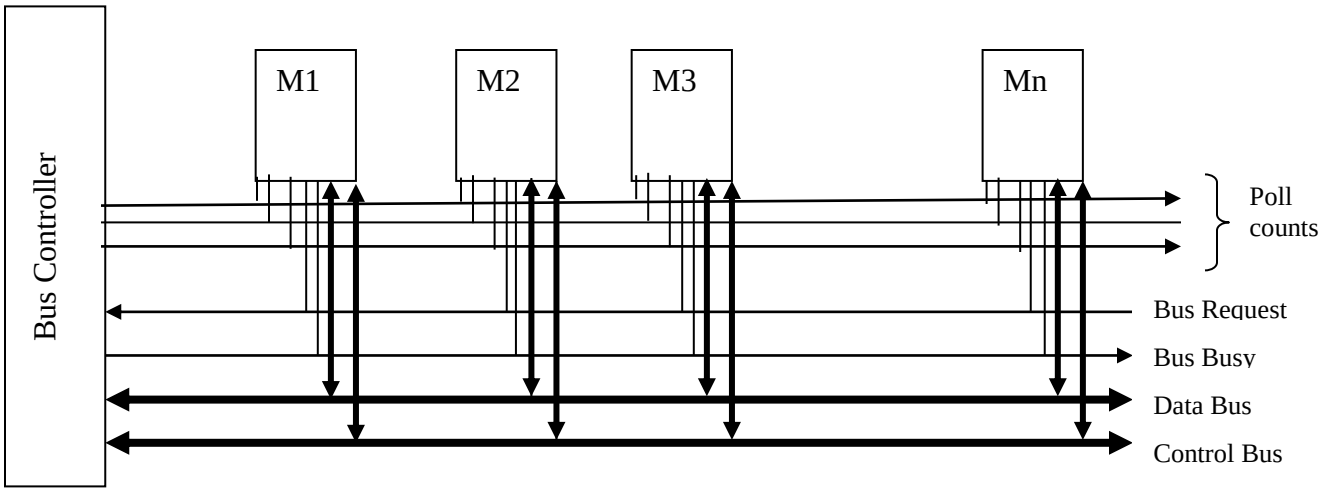
- Synchronous or Asynchronous timing:-It is another important aspect of buses in which data transfer is timed. Transfer of data performed during specific time that is known as source and destination. Where as in asynchronus buses transfer of data performed separate control signal.
- Bus arbitration:-It is an important aspect of the system where buses are used for the control of a bus.Buses are associated by module.
- Daisy chaining:-It is an arbitration in which two control signals are used

  A:-Bus request.

  B:-Bus busy.

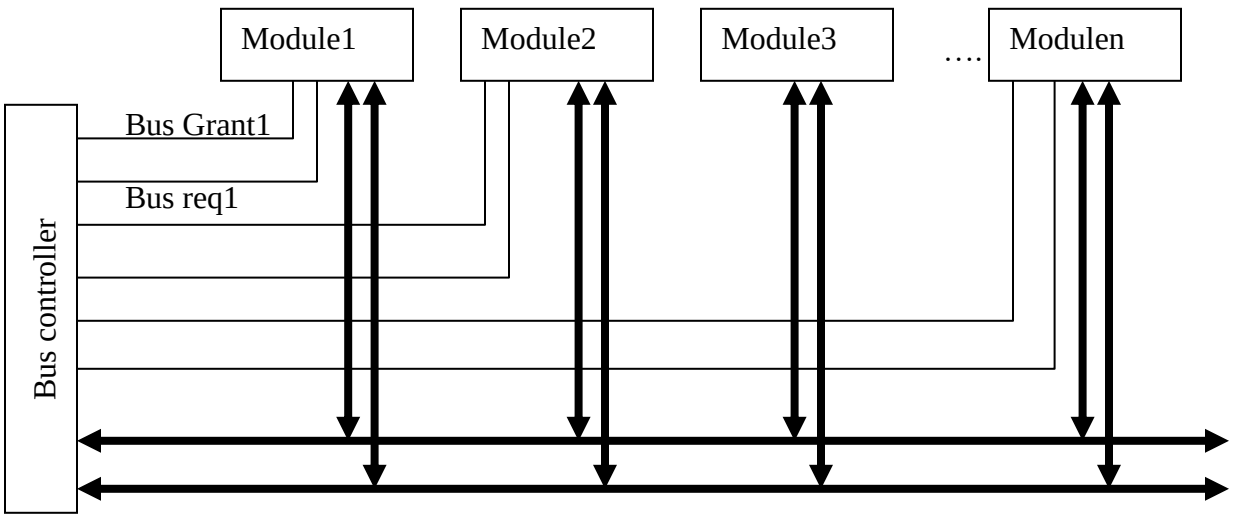


- Polling:-



In polling there are poll counts are used in bus grant instead of single bus grant

- Independent requesting



In this arbitration scheme, each module has its independent bus request and bus grant line.

Question Bank:-

Question1:-Describe about von Neumann architecture.

Question2:-Discuss history of computer.

Question3:-What are different types of computer?

Question4:-What are different categories of number system? Explain in brief.

Question5:-What are different types of coding system?

Question6:-What is the role of compliments? Explain it in brief

Question7:-What are different categories of I/O organizations?

Question8:-What is memory hierarchy? Explain in brief.

Question9:-Differentiate between primary and secondary memory.

Question10:-Differentiate between RAM and ROM memory.

Question11:-Differentiate between Impact and non-Impact printer?

Question12:-What are different types of gates.?Explain with suitable example.

Question13:-How instructions are executed in register?.

Question14:-What is the role of system bus?

Question15:-What is the role of interrupt?

Question16:-What is the role of I/O processors?

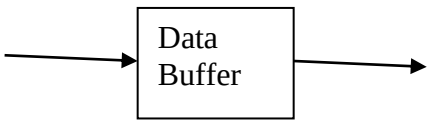
Question17:-What is flip flop?

Question18:-Explain various components of computer system?

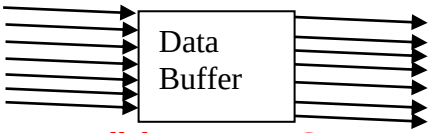
Question19:-Describe mechanism of storage capacity in hard disk and optical memory?

Question20:-What is the of mainframe and supercomputer in different area?

Parallel Processing/Flyns Classification of Computers:-



Serial Execution Concept



Parallel Execution Concept

A technique that are used to provides simultaneous data processing tasks for the purpose of increasing the computational speed of computer.

According to M.J. Flynns, Computers are classified four categories.

- ✓ SISD(Single Instruction Single data)
- ✓ SIMD(Single Instruction Multiple data)
- ✓ MISD(Multiple Instruction Single data)
- ✓ MIMD(Multiple Instruction Multiple data)

SISD:-

Using this technology, Single Instruction applies on single data stream through pipeline.

Example:-Conventional Von Neumann architecture.

SIMD:-

Using this technology, Single Instruction applies multiple data stream or Instructions broadcast on multiple data stream.

Example:-Array Processors.

MISD:-

Using this technology, Multiple Instruction applies on single data stream.

Example:-Distributed architecture, Vector Processors.

MIMD:-

Using this technology, Multiple Instruction applies on multiple data stream.

Example:-Multiprocessor System, Data Flow architecture.

Vector Processing:-

Such types of computing system basically used in following areas.

- Weather forecasting.
- Petroleum Explorations.
- Flight simulations.
- Artificial & Expert System. (Knowledge Based System).
- Image processing.

Concept of RISC & CISC Technology:-

RISC (Reduced Instruction Set Computer).

CISC (Complex Instruction Set Computer).

Above both types of technology used in making Processors.

In both types of CPU technology instructions passes through pipelines.

Pipeline:- It provides a way to start a new task before an old one has been completed. This technique is called pipeline.There are following two types of pipelining technique.

1:-Instruction Pipeline 2:-Arithmetic pipeline

Fetch→Decode→Execute→Store (By means of pipelines)

Instruction Pipelines (Cycle):-

Step1 Calculate the address for next instructions.

Step2 Fetch the instruction.

Step3 Decode the operation required by the instructions.

Step4 Calculate the address for operand.

Step5 Perform the operation on the data.

Step6 Calculate the address of the operand.

Step7 Store the operand.

Risc Architecture:-RISC having designed the following way.

A:-One instruction Per Cycle.

B:-Register to Register Operand.

C:-Simple Addressing Mode.

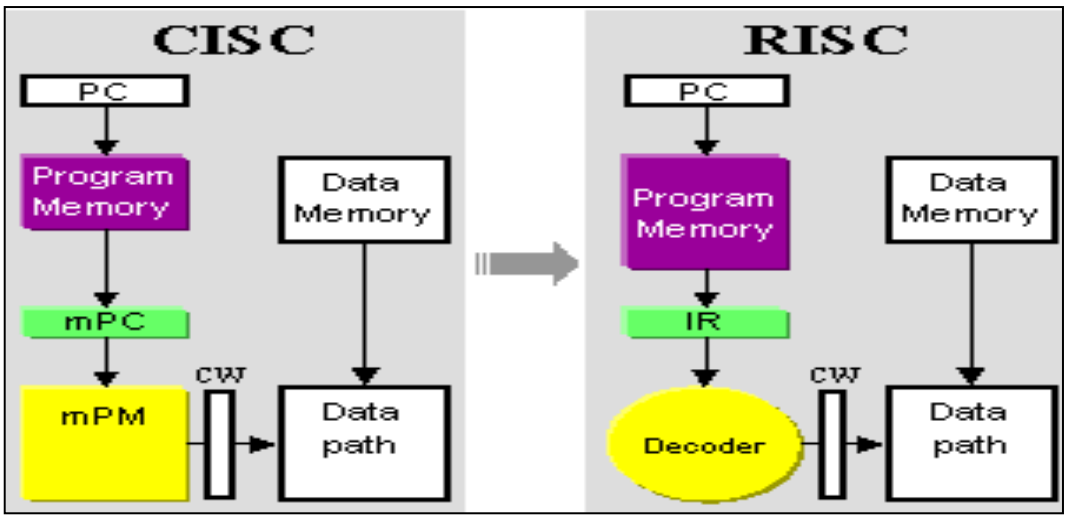
D:-Simple Instruction format.

E:-Implementation of VLSI technology.

F:-High performance of instruction execution.

Difference between RISC & CISC:-

RISC	CISC
1:-Size Of register is Smaller, less than 100 instructions per cycle	1:-Size of register is larger per cycle
2:-CPI is lesser	2:-CPI is larger
3:-Low code density	3:-High Code Density
4:-It Support HLL	4:-It also support HLL
5:-It uses microcode	5:-It also uses microcode
6:-It gives relatively less performance	6:-It gives relatively peak performance
7:-Large number of General purpose Register	7:-GPR varies from 8-32
8:-Hardwired control unit	8:-Microprogrammed control unit



The Microinstruction:-

An instruction of microprogram is called microinstruction.It specifies one or more micro operations, which can be executed simultaneously.When microinstructions are executed, a set of control signals are generated.

Types of Microinstructions:-

There are two categories of microinstruictions.

1:-Branching Microinstructions:-

Instructions are executed depending upon conditions/flags.

2:-Non-Branching Microinstruction

Instructions are executed without conditions/flags.

Format of Microinstruction

There are following three formats of microinstructions

1:-Horizontal Microinstruction

2:-Vertical Microinstruction

3:-Dual Use bits Microinstruction

Draw Figure :-

(a) →Page Number 77

(b) →Page Number 77

(c) →Page Number 77

Machine Startup:-

The control unit is responsible for initializing various registers during the startup of machine.The control unit loads a H/W generated address in the program counter and starts execution the instruction stored in that location. This address of the first instruction is known ***Reset vector***.

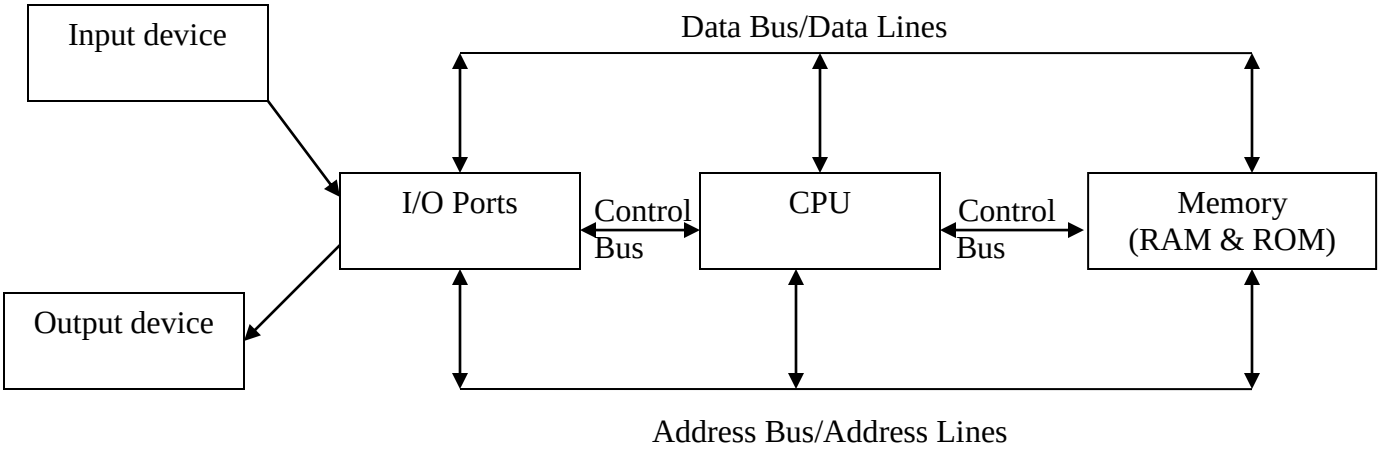
Microprocessor Architecture:- (Intel 8085/8086 Processor)

The word micro is used in microscopes, microwaves, microprocessors, microcomputers, microprogramming, microcodes etc.It means small.
Microprocessor is examples of VLSI technology in which 10^6 gates are integrated.There are three components are used in microcomputers.

- I/O devices
- CPU
- Memory

These components are connected by using **system buses**. There are three types of buses.

- Address Bus/Address Line
- Data Bus/Data Line
- Control Bus/Control Line



Bus Sizes of 8085/8086 Processor:-

16 bit lines are used. It can access 2^{16} =64 K Bytes →→For 8085
The address bus of 8086 microprocessor has a 20 bits address bus. →For 8086

Advantage of Microprocessors:-

- Compact but powerful.
- Easily programmable and maintainable due to small size.
- Useful in distributed application.

Demonstration of Microprocessors:-

- More throughput.
- More addressing capability.
- Powerful addressing modes.
- Powerful instruction set.
- Virtual memory management for large program.

Execution of Microprocessors:-

All microprocessors execute a continuous loop of fetch and execute cycles.

```
while(1)
    {
    fetch (byte);
    execute (using byte);
    }
```

8086 microprocessor consist of two independent units.

- The Bus Interface Unit (BIU).
    - Calculate the physical address.

- Fetch the instruction.
- Reading or Writing data memory or I/O port from memory or I/O.

- The Execution Unit (EU).
  - It performs all operations inside ALU.

Register Set of 8086:-

8086 register have five groups of registers. These groups are

- **GPR :-**8086 microprocessor consist of 4 general purpose register.These are AX, BX,CX and DX.
  AX→Accumulator register.
  BX→Base register.
  CX→Counter register.
  DX→I/O port register.
- **Segment register:-**It is used for calculating physical address of instruction or memory.
- **Pointer And Index register:-**It has three pointer and address register.
  BP→Base Pointer.
  SI→Source Index.
  DI→Destination address.
- **Special register:-**There are two types of special register in 8086.
  SS→Stack segment
  SP→Stack Pointer
- **Flags register :-**It is used for checking status control.

Instruction Set of 8086:-

- Data Transfer instructions
  - MOV dest,source→move data from source to destination
  - PUSH operand→Pushes the operand into a stack.
  - POP destination→Pop a word from stack
  - XCHG destination,source→It exchange byte or word from source to destination
  - LEA register,source→Load effective address
  - LDS dest-reg→Load data segment
  - PUSHF→Pushes flag register to top of the stack
  - POPF→Pops the stack top to flag register
- Arithmetic Instructions
  - ADD dest, src Add contents of dest into src.
  - ADC des,src→Add byte+byte+carry flag
  - INC destination→It increments specified byte or word operand by one.
  - SUB dest,src→Subtract from destination to source
  - DEC src→ It decrements specified byte or word operand by one.
  - NEG src→It complement the source value.
  - CMP dest,src
- Bit Manipulation Instructions
  - NOT dest
  - AND dest,src
  - OR dest,src
  - XOR dest,src
  - SHL/SAL des,count
  - SHR→Shift bits of word or byte right,put zero in MSB.
  - RCL→Rotate bits of word or byte left, MSB to CF and CF to LSB.
  - RCR→Rotate bits of word or byte right, LSB to CF and CF to MSB.
  - ROL→Rotate bits of word or byte left, MSB to LSB and To CF.
- Program Execution Transfer Instructions
  - JMP→ Unconditionally go to specified address to get next instruction.
  - RET→ It return procedure to calling program.it work with the pair of CALL
  - LOOP→Execute a set of instructions until CX is equal to Zero.
  - LOOPE→It execute a sequence of instructions while zero flag=1 and CX not equal zero
- String Instructions

- REP
- MOVS/MOVSB/MOVSW
- CMPS/CMPSB/CMPSW
- LODS/LODSB
- STOS/STOSB/STOSW

- Processor Control Instructions

STC	Set clear flag to 1.
CLC	Clear the carry flag to 0.
CMC	Compliment the state of carry flag.
STD	Set the direction flag to 1.
CLD	Clear the direction flag to 0.
STI	Set interrupt enable flag to 1.
CLI	Clear interrupt enable flag to zero.
HLT	Do nothing until interrupt or reset.
NOP	No action except fetch and decode.

ADDRESSING MODES of 8086

The basic set of operands in 8086 may reside in register, memory and immediate operand.Addressing modes helps in addressing complex data structure with ease.Some of specific terms and registers roles for addressing are:

- Base register (BX, BP)→These register are used for pointing to base of an array, stack etc.
- Index register (SI, DI) →These registers are used as index registers in data/or extra segments.
- Displacement→It represent offset from the segment address.

Table of addressing modes:-

MODE	Description
Register Indirect	Effective address is the displacement of memory variable
Based	Effective address is the contents of register
Indexed	Effective address is the sum of an index register and a displacement
Based indexed	Effective address is the sum of a base and an index register
Based indexed with displacement	Effective address is the sum of a base register,an index register and a displacement

Register addressing Mode:-

Example:-

AX,BX,CX,DX,SI,DI,BP,IP,CS,ES,SS etc.

Immediate addressing Mode:-

The only constraint is that the assembler must be able to determine the value of an immediate operand at assembly time.

Example:-

MOV AL,11.

Direct addressing Mode:-

It is also known as direct operands are also called as relocatable operands .

Example:-

MOV COUNT,CL.

Indirect addressing Mode:-

In indirect addressing modes,operands use registers to point to locations in memory.It is useful mode for handling strings/arrays etc.There are two types of indirect addressing modes.

- Base register BX,BP.
- Index register SI,DI.

Introduction of assembly Language:-

We know that machine language program written into, by using 0 and 1.It is very complex language for writing computer programs.After then 0 and 1 replced by using symbols,codes and other types of representing instructions.For this reason this language is comparatively simple and easy to understanding.But assembly language program dependes upon architecture of CPU.

Pros and Cons of Assembly language:-

- Assembly language provides more control over handling particular h/w and s/w.It allows us to study instructions set, addressing modes and interrupts.
- Assembly programming generates smaller, more compact executable modules.
- It is long and tedious to write initially.
- Our bugs can be very difficult to chase.
- We can access machine-dependent registers and I/O.
- We can control the exact code behavior in critical sections that might otherwise involve deadlock between multiple software threads or hardware devices.
- A small change in algorithmic design might completely invalidate all our existing assembly code. So that either we are ready (and able) to rewrite it all, or you're tied to a particular algorithmic design.

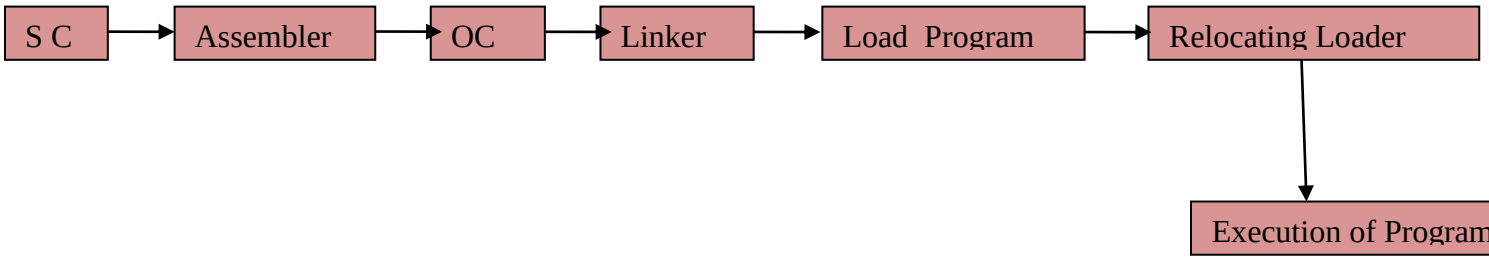
How to execute assembly language program:-

- Step 1:-When source code translate by using assembler it generate an intermediate object file(.obj)
- Step 2:-In second step .obj file convert into .exe
- Step 3:-Load the program into memory for execution

Assembly language performed by Two-Pass assembler.
Object module passes Two phases.

- Pass-1
  - It includes following three steps.
    - Separate Symbols, mnemonic codes and operational fields.
    - Determine storage requirements for every assembly statements.
    - Build The symbol.(It is used to store corresponding values).
- Pass-2
  - It generates only object code. If no any bugs found in source codes.

Implementation of Assembler:-



Program In Assembly Language:-

1:-Addition of two Numbers:-

```
.model small
.data
 opr1 dw 1234h
 opr2 dw 0002h
 result dw 01 dup(?),'$'
.code
        mov ax,@data
        mov ds,ax
        mov ax,opr1
        mov bx,opr2
        clc
        add ax,bx
        mov di,offset result
        mov [di], ax

        mov ah,09h
        mov dx,offset result
        int 21h

        mov ah,4ch
        int 21h
        end
```

2:-Check Number is Eve/Odd:-

```
model small
.stack 200H
.data
msg1 DB 10,13,' Number is odd.$'
msg2 DB 10,13,' Number is even.$'
msg3 DB 10,13,'Enter the no:$'
newline DB 10,13,'$'
sum DW 0
count DW ?
num DW ?
.code
print MACRO msg
PUSH AX
PUSH DX
```

```
MOV AH,09H
MOV DX,offset msg
INT 21H
POP DX
POP AX
ENDM
.startup
print newline
print msg3
CALL readnumtoAX
MOV BL,02
DIV BL
CMP AH,00
JE even1
print newline
print msg1
JMP last
even1:
print newline
print msg2
last:
.exit
readnumtoAX PROC NEAR
PUSH BX
PUSH CX
MOV CX,10
MOV BX,00
back:
MOV AH,01H
INT 21H
CMP AL,'0'
JB skip
CMP AL,'9'
JA skip
SUB AL,'0'
PUSH AX
MOV AX,BX
MUL CX
MOV BX,AX
POP AX
MOV AH,00
ADD BX,AX
JMP back
skip:
MOV AX,BX
POP CX
POP BX
RET
readnumtoAX ENDP
```

END
3:-Check Year is Leap/Not:-