UNIT -1 PART-B (10 MARKS)

1.Numbering conversion problems?

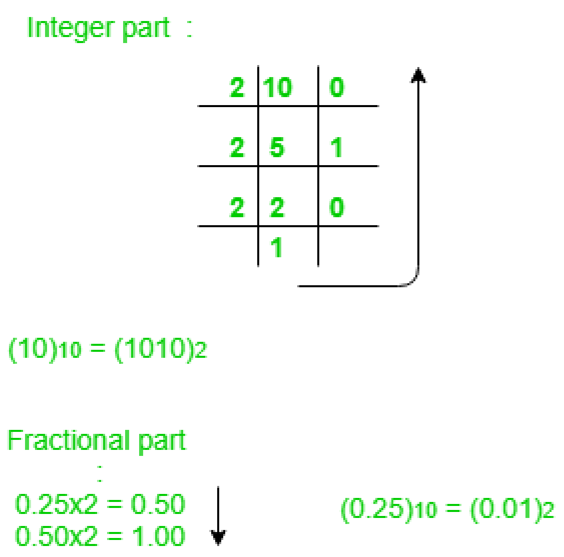
Number conversion problems are **problems that involve converting numbers from one system to another**. For example, converting from decimal to binary, or from hexadecimal to octal. There are different methods to perform these conversions, but one common way is to use repeated division and remainder operations. To check the correctness of the conversion, one can use the reverse process or a calculator.

1. Decimal to Binary Number System

To convert from decimal to binary, start dividing decimal number by 2, and whatever the reminder getting, writing down from bottom to top, and that will be the binary number representation of the decimal number. And the number contains fractional part, then multiply 2 in the fractional part.

Example

(10.25)10



Note: Keep multiplying the fractional part with 2 until decimal part 0.00 is obtained.
(0.25)10 = (0.01)2

Answer: (10.25)10 = (1010.01)2

2. Binary to Decimal Number System

To convert from binary to decimal, start multiplying the exponent of 2 with each digit of the number in decreasing order. If the number contains fractional part then will divide it by the exponent of 2.

Example

(1010.01)2
1x23 + 0x22 + 1x21+ 0x20 + 0x2 -1 + 1x2 -2 = 8+0+2+0+0+0.25 = 10.25
(1010.01)2 = (10.25)10

3. Decimal to Octal Number System

To convert from decimal to octal, start dividing decimal number by 8, and whatever the reminder getting, writing down from bottom to top, and that will be the octal number representation of the decimal number. And the number contains fractional part, then multiply 8 in the fractional part.

Example

(10.25)10
(10)10 = (12)8
Fractional part:
0.25 x 8 = 2.00

Note: Keep multiplying the fractional part with 8 until decimal part .00 is obtained.
(.25)10 = (.2)8

Answer: (10.25)10 = (12.2)8

4. Octal to Decimal Number System

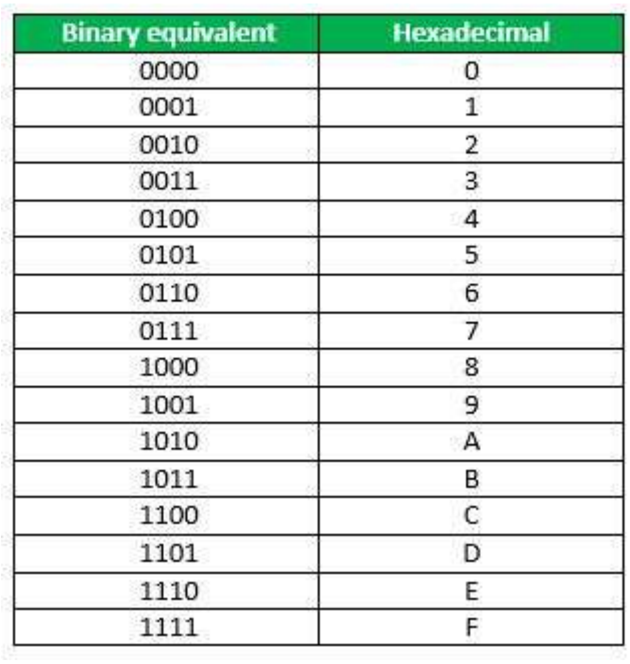
To convert from octal to decimal, start multiplying the exponent of 8 with each digit of the number in decreasing order. If the number contains fractional part then will divide it by the exponent of 8.

Example

(12.2)8
1 x 81 + 2 x 80 +2 x 8-1 = 8+2+0.25 = 10.25
(12.2)8 = (10.25)10

5. Hexadecimal to Binary Number System

To convert from Hexadecimal to Binary, write the 4-bit binary equivalent of hexadecimal.



Binary equivalent	Hexadecimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Example

(3A)16 = (00111010)2

6. Binary to Hexadecimal Number System

To convert from Binary to Hexadecimal, start grouping the bits in groups of 4 from the right-end and write the equivalent hexadecimal for the 4-bit binary. Add extra 0's on the left to adjust the groups.

Example

1111011011

<u>0011</u> <u>1101</u> <u>1011</u>

(001111011011)2 = (3DB)16

7. Binary to Octal Number System

To convert from binary to octal, start grouping the bits in groups of 3 from the right end and write the equivalent octal for the 3-bit binary. Add 0's on the left to adjust the groups.

Example

111101101

<u>111</u> <u>101</u> <u>101</u>

(111101101)2 = (755)8

2.explain k-mapping by using example problem?

In many digital circuits and practical problems, we need to find expressions with minimum variables. We can minimize Boolean expressions of 3, 4 variables very easily using K-map without using any Boolean algebra theorems.

K-map can take two forms:

1. Sum of product (SOP)
2. Product of Sum (POS)

According to the need of problem. K-map is a table-like representation, but it gives more information than the TABLE. We fill a grid of the K-map with 0's and 1's then solve it by making groups.

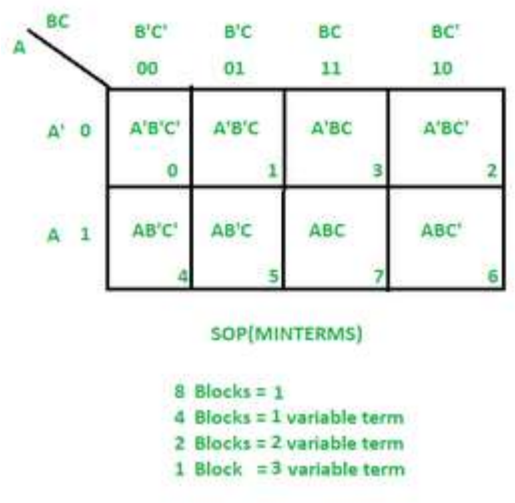
Steps to Solve Expression using K-map

1. Select the K-map according to the number of variables.
2. Identify minterms or maxterms as given in the problem.
3. For SOP put 1's in blocks of K-map respective to the minterms (0's elsewhere).
4. For POS put 0's in blocks of K-map respective to the max terms (1's elsewhere).

5. Make rectangular groups containing total terms in power of two like 2,4,8 ..(except 1) and try to cover as many elements as you can in one group.
6. From the groups made in step 5 find the product terms and sum them up for SOP form.

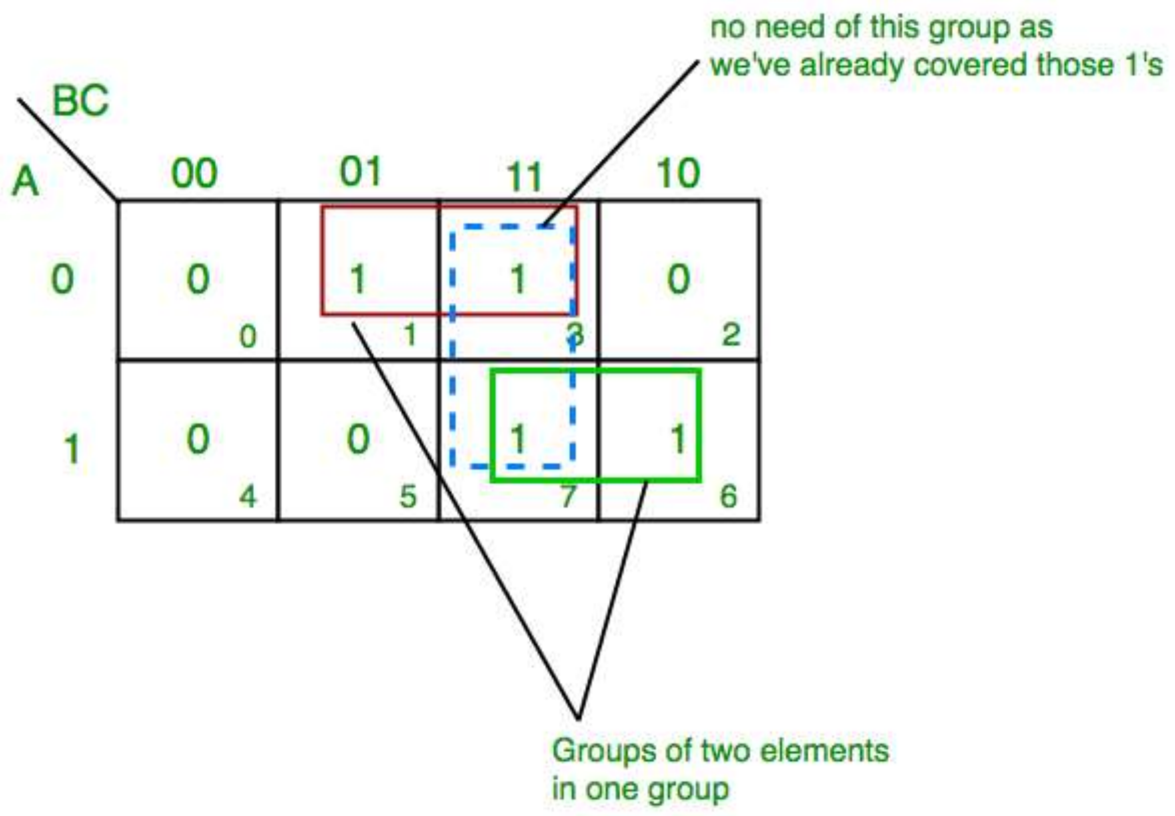
SOP FORM

1. K-map of 3 variables



K-map SOP form for 3 variables

Z= ?A,B,C(1,3,6,7)



From **red** group we get product term—

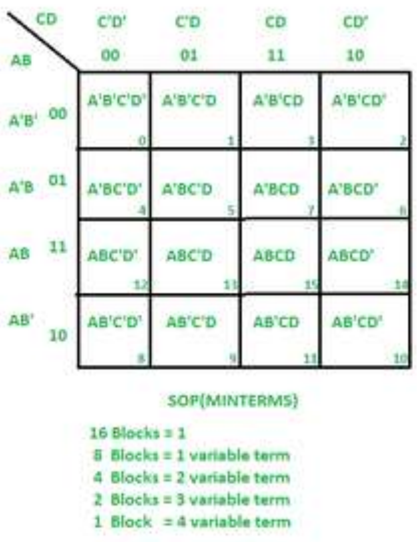
A’C

From **green** group we get product term—

AB

Summing these product terms we get- **Final expression (A’C+AB)**

2. K-map for 4 variables



K-map 4 variable SOP form

F(P,Q,R,S)=?(0,2,5,7,8,10,13,15)

From **red** group we get product term—

QS

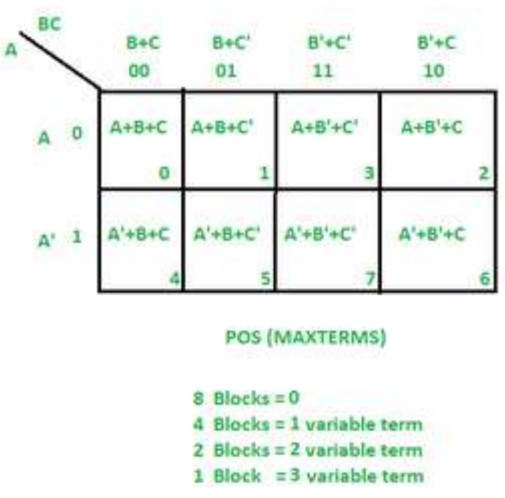
From **green** group we get product term—

Q’S’

Summing these product terms we get- **Final expression (QS+Q’S’)**.

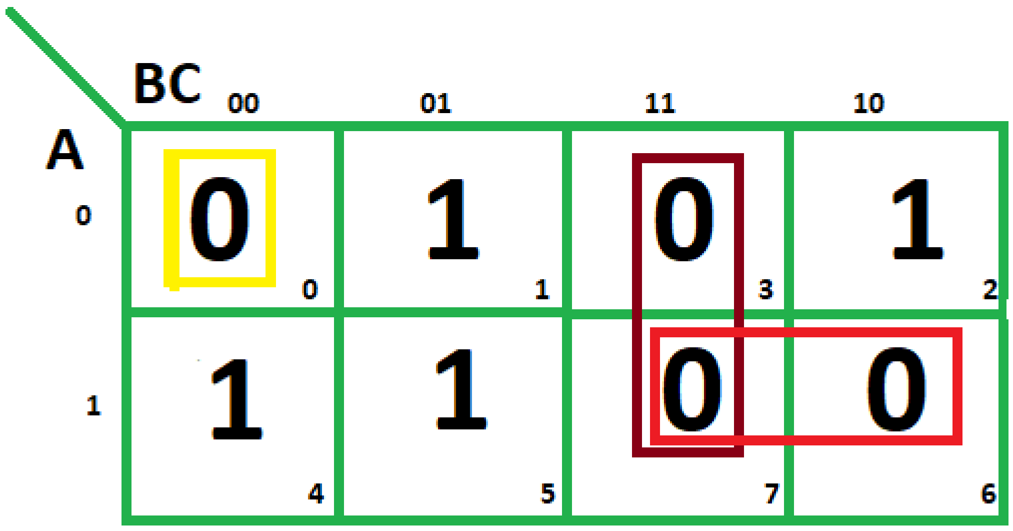
POS FORM

1. K-map of 3 variables



K-map 3 variable POS form

F(A,B,C)=?(0,3,6,7)



From **red** group we find terms

A B

Taking complement of these two

A' B'

Now **sum** up them

(A' + B')

From **brown** group we find terms

B C

Taking complement of these two terms

B’ C’

Now sum up them

(B’+C’)

From **yellow** group we find terms

A' B' C’

Taking complement of these two

A B C

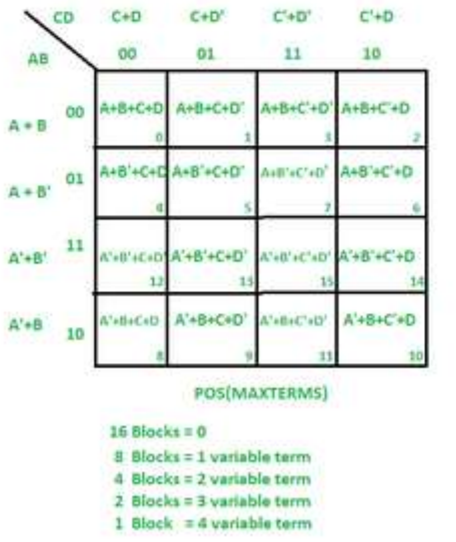
Now **sum** up them

(A + B + C)

We will take product of these three terms : **Final expression –**

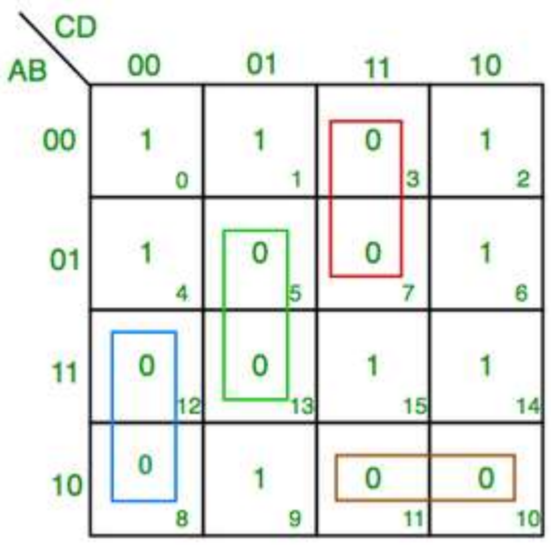
(A' + B’) (B’ + C’) (A + B + C)

2. K-map of 4 variables



K-map 4 variable POS form

F(A,B,C,D)=?(3,5,7,8,10,11,12,13)



From **green** group we find terms

C’ D B

Taking their complement and summing them

(C+D’+B’)

From **red** group we find terms

C D A’

Taking their complement and summing them

(C’+D’+A)

From **blue** group we find terms

A C’ D’

Taking their complement and summing them

(A’+C+D)

From **brown** group we find terms

A B’ C

Taking their complement and summing them

(A’+B+C’)

Finally we express these as product –

(C+D’+B’).(C’+D’+A).(A’+C+D).(A’+B+C’)

3. **Explain about once and twos compliment and example for subtraction of 1’s compliment?**

   1's Complement

- 1's complement is used when we want to alter and **invert all the significant binary digits of any number.**
- It **changes all 0 to 1 and all 1 to 0** in the given number and the final generated number is 1's complement.
- Let's take **Decimal 5 as an example** which is equivalent to 101 in Binary.
- If we generate its 1's complement then **it will be 010 which is 2 in Decimal.**

2's complement

- 2's complement is **used to store and signify signed and negative numbers.**
- If **we add 1 to LSB of any number 1's complement** then it will result in 2's complement.
- If we take **Decimal 5 as an example** then we know that 1's complement of it is 010
- So 2's complement will be 011 which **is equivalent to 3 in Decimal and it is the 2's complement for the given number.**

<u>Example for 1’s complement:-</u>

- 1's complement of "0111" is "1000"
- 1's complement of "1100" is "0011"

Binary subtraction using 1’s compliments:-

A number's 1's complement is derived by reversing every 0 to 1 and every 1 to 0 in a binary integer. For example, the binary number 1102 has a 1's complement of 0012. Please follow the instructions below to accomplish binary subtraction using 1's complement. Binary subtraction with 1's complement entails adding the complement of the subtrahend (the number being subtracted) to the minuend (the amount being removed). Here's an illustration:
Let's subtract (100010)2 from (110110)2. In this case, the binary equivalent of 34 is (100010)2 whereas the binary equivalent of 54 is (110110)2.
Step 1: First identify the Minuend and Subtrahend. In this Minuend is (110110)2 and subtrahend is (100010)2.
Step 2: Find the 1's complement of the subtrahend. The subtrahend is (100010)2 and after 1's complement it is (011101)2.
Step 3: Now add the minuend and the 1's complement of the subtrahend.

$$\begin{array}{rrl} & 110110 & \text{(Minuend)} \\ + & 011101 & \text{1's Complement of Subtrahend} \\ \hline & 1010011 & \\ \hline \end{array}$$

Step 4: This increment is shown in the left-most digit, 1. We add that to the result, which is (010011)2.

$$\begin{array}{rr} & 010011 \\ + & 1 \\ \hline & 010100 \\ \hline \end{array}$$

As a consequence, the answer is (10100)2. In addition, the difference between 54 and 34 is 20. The binary equivalent of 20 is (10100)2.

4. Convert the hexadecimal (1A31A31A3) number to binary ,octal and decimal numbers?

ans

Converting hexadecimal number to Binary number

Given hexadecimal number is

$(1A31A31A3)_{16}$

$(000110100011000110100011000110100011)_2$

The Binary equivalent of hexadecimal number is $(1A31A31A3)_{16} = (000110100011000110100011000110100011)_2$

Converting Hexadecimal number to octal number

Given hexadecimal number is

$(1A31A31A3)_{16}$

for converting hexadecimal number into octal number first we should write the binary Binary number for the given hexadecimal number as follows

$(000110100011000110100011000110100011)_2$

1

No we should ~~$~~ consider the binary number into 3 Bits as follows

$\underset{0}{\underline{000}}\ \underset{6}{\underline{110}}\ \underset{4}{\underline{100}}\ \underset{3}{\underline{011}}\ \underset{0}{\underline{000}}\ \underset{6}{\underline{110}}\ \underset{4}{\underline{100}}\ \underset{3}{\underline{011}}\ \underset{0}{\underline{000}}\ \underset{6}{\underline{110}}\ \underset{4}{\underline{100}}\ \underset{3}{\underline{011}}$

The octal equivalent of hexadecimal is

$(1A31A31A3)_{16} = (643064306643)_8$

Converting hexadecimal number to decimal number

The given hexadecimal number is

$(1A3\ 1A3\ 1A3)_{16}$

$\underset{16^8}{1}\ \underset{16^7}{A}\ \underset{16^6}{3}\ \underset{16^5}{1}\ \underset{16^4}{A}\ \underset{16^3}{3}\ \underset{16^2}{1}\ \underset{16^1}{A}\ \underset{16^0}{3}$

$1\times16^8 + A\times16^7 + 3\times16^6 + 1\times16^5 + A\times16^4 + 3\times16^3 + 1\times16^2 + A\times16^1 + 3\times16^0$

$1\times16^8 + 10\times16^7 + 3\times16^6 + 1\times16^5 + 10\times16^4 + 3\times16^3 + 1\times16^2 + 10\times16^1 + 3$

$= 7031370147$

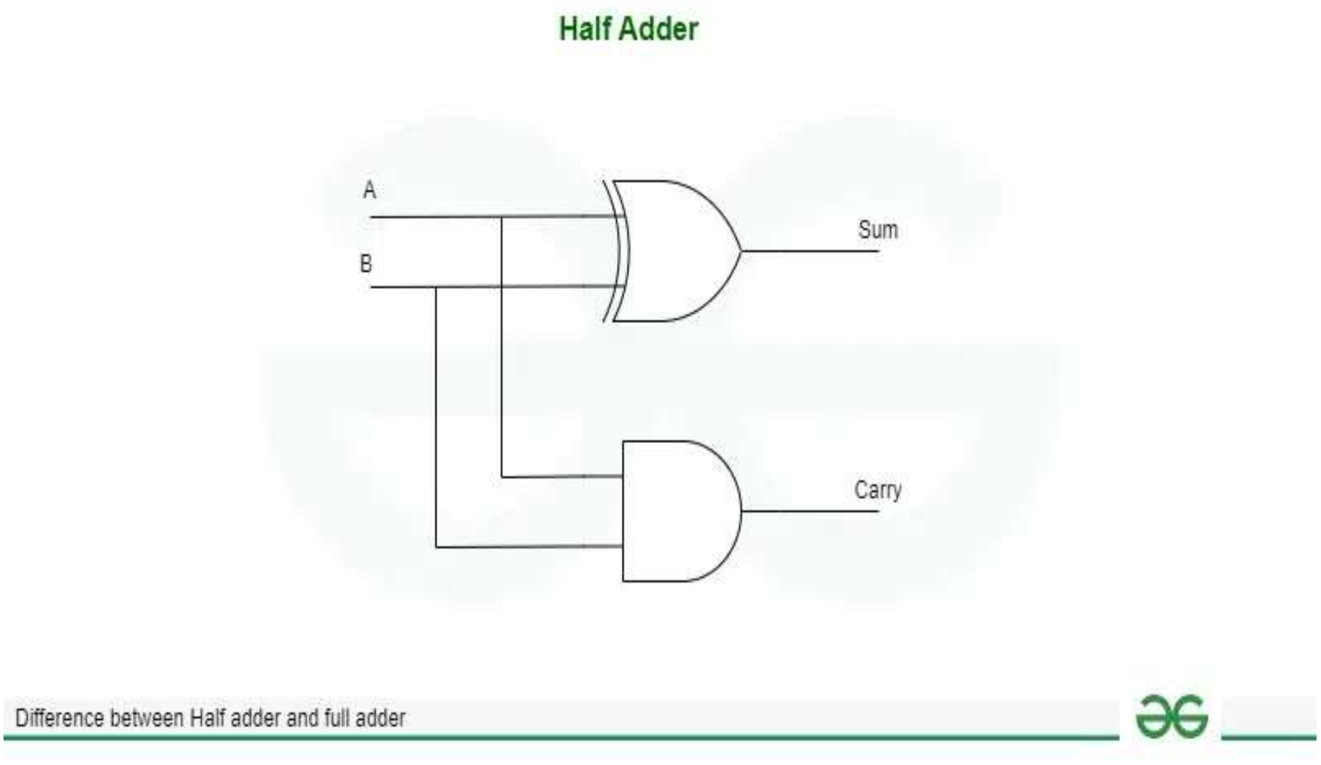
The decimal equivalent of hexadecimal is

$(1A31A31A3)_{16} = (7031370147)_{10}$

5. Explain about Half adder and full adder?

What is Half Adder?

Half Adder is a combinational logic circuit that is designed by connecting one EX-OR gate and one AND gate. The half-adder circuit has two inputs: A and B, which add two input digits and generate a carry and a sum.



Half Adder

The output obtained from the EX-OR gate is the sum of the two numbers while that obtained by AND gate is the carry. There will be no forwarding of carry addition because there is no logic gate to process that. Thus, this is called the Half Adder circuit.

Logical Expression of Half Adder

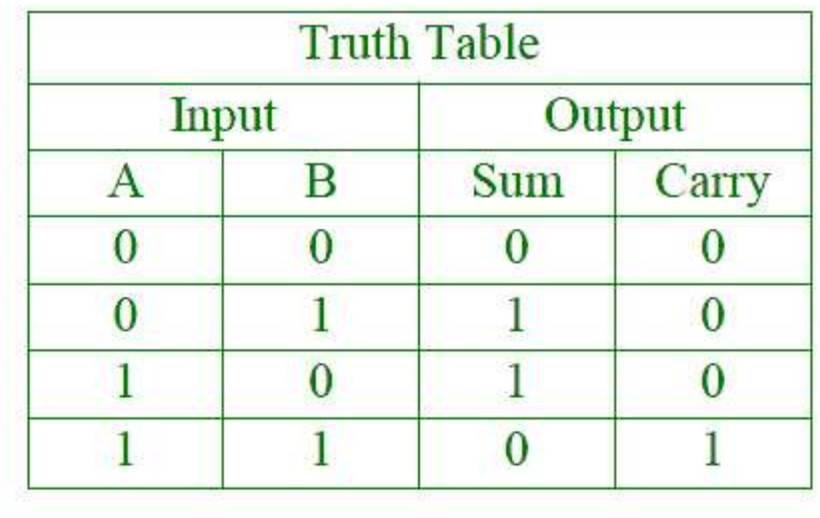
The Logical Expression for half added is given as

Sum = A $\oplus$ *B*

Carry = A AND B

Truth Table of Half Adder

The Truth Table for Half Added is Given as



Truth Table			
Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

What is Full Adder ?

Full Adder is the circuit that consists of two EX-OR gates, two AND gates, and one OR gate. Full Adder is the adder that adds three inputs and produces two outputs which consist of two EX-OR gates, two AND gates, and one OR gate. The first two inputs are A and B and the third input is an input carry as C-IN. The output carry is designated as C-OUT and the normal output is designated as S which is SUM.

Full Adder

The equation obtained by the EX-OR gate is the sum of the binary digits. While the output obtained by AND gate is the carry obtained by addition.

Logical Expression of Full Adder

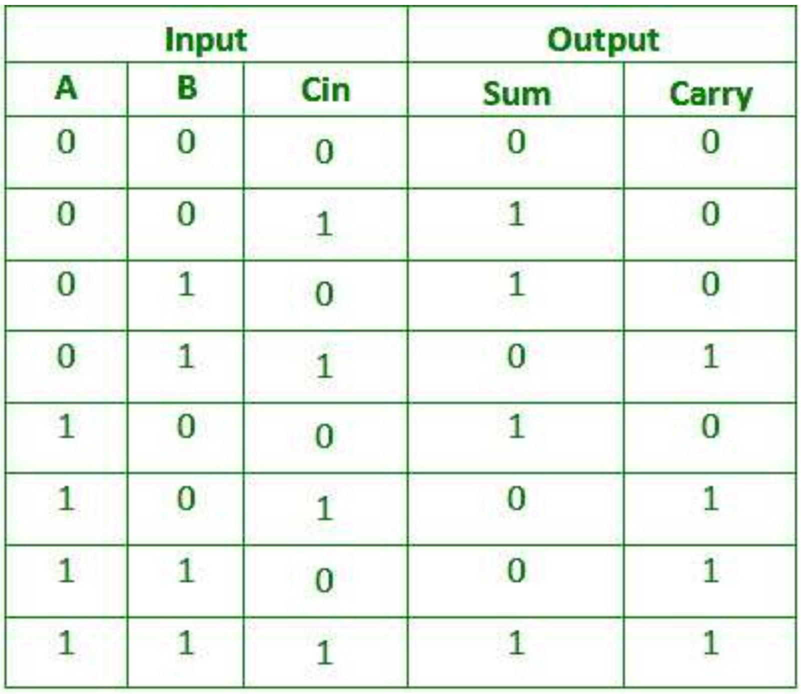
Given Below is the Logical Expression of Full Adder

SUM = (A XOR B) XOR Cin = $(A \oplus B) \oplus Cin$

CARRY-OUT = A AND B OR Cin(A XOR B) = $A.B + Cin(A \oplus B)$

Truth Table of Full Adder

Given Below is the truth Table of Full Adder



Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Sum and Carry Operation

In both Half Adders and Full Adders, Sum is the output of the addition of the two inputs and Carry is the output which is an overflow of the output position and needs to be shifted to the next higher position while adding successive bit inputs.

- **Sum (S)** : It results from the XOR gate, which is a logic gate that adds two or more bits together in the same way that you add in base 2 with no acknowledgement of carry from the previous bit.
- **Carry (C or Cout):** It is the output of the AND operation in the case of the Half Adder or both AND and OR Operations in the case of the Full Adder to indicate that a '1' has to be carried over to the next bit position.

Advantages and Disadvantages

Advantages of Half Adder

- Flexible and easy when it comes to design.
- Involves the use of fewer logic gates thus, is cheaper.

Disadvantages of Half Adder

- Fails to process a carry input from the previously added numbers.
- Restricted to the addition of only two bits.

Advantages of Full Adder

- Can add 3 bits, it includes one carry input and a carry output, which can perform more elaborate computations.
- It can be cascaded to produce adders for a number of bit additions which makes it suitable for multi bit arithmetic.

Disadvantages of Full Adder

- Complex and needs more gates, hence making the design more complicate and expensive.
- Yeah man, slightly slower because normally 2 gate process are used instead of 1.

Applications

Applications of Half Adder

- Arithmetic operations like addition, subtraction, and multiplication in low level dynamic circuits.
- Three types of rectifiers: half-wave, full-wave, and full-wave with a center tapped secondary. Used in small integration circuits.

Applications of Full Adder

- Carry-look ahead adders in digital processors that utilize multi-bit binary addition.

- Present in the arithmetic logic units (ALU) and other complicated digital systems

6. Explain about multiplexer and demultiplexer?

A Multiplexer (MUX) and a Demultiplexer (DEMUX) are essential digital circuits in communication systems, performing opposite functions. A multiplexer combines multiple input signals into a single output, while a demultiplexer takes a single input signal and routes it to one of many output lines.

In this article, we'll explore the differences between a multiplexer and a demultiplexer, their advantages, disadvantages, and applications.

What is a Multiplexer?

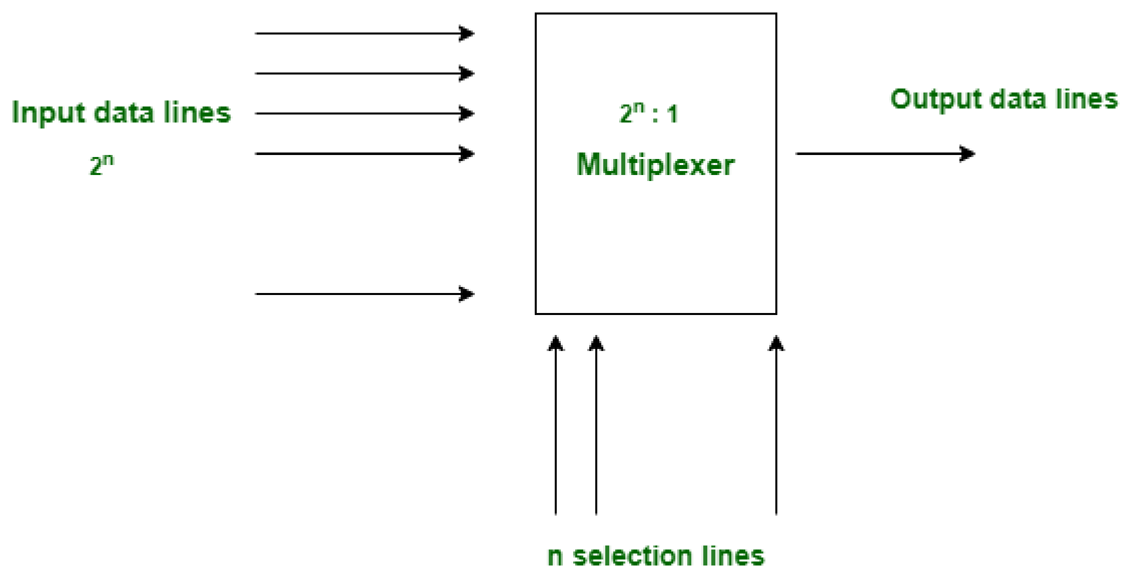
A **multiplexer** is a combinational circuit with multiple data inputs and a single output, determined by control or select lines. Often referred to as **MUX**, it requires log2(N)log2(*N*) selection lines for N*N* input lines, or equivalently, n*n* selection lines for 2n2*n* input lines.

Multiplexers are also known as:

- N-to-1 selectors
- Parallel-to-serial converters
- Many-to-one circuits
- Universal logic circuits

They are mainly used to increase the amount of data that can be sent over a network within a certain amount of time and bandwidth

Below is the Block Diagram of the Multiplexer, It will have 2n Input lines and will select output based on the Select line.



Advantages of Multiplexer(MUX)

- **Reduces the number of data lines**: MUX allows multiple signals to share a single communication line, saving space and resources.
- **Simpler Design**: It simplifies the system by reducing the number of data channels needed.
- **Efficient Use of Resources**: MUX optimizes the use of communication channels or buses.
- **Flexible Design**: MUX can be used for different types of data and communication formats.
- **Compact Systems**: It helps make systems smaller and simpler by reducing the number of connections.
- **Reduces Errors**: Fewer data lines mean less chance of interference or noise.

Disadvantages of Multiplexer(MUX)

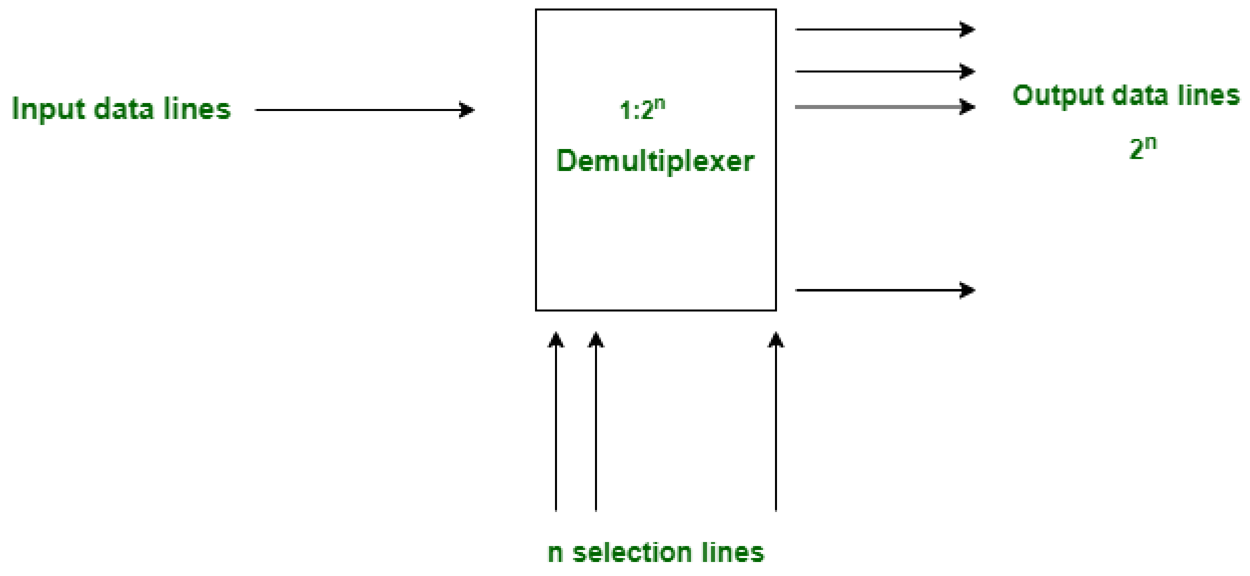
- **Complex Control**: It needs extra circuits to select which input is sent, making the design more complicated.
- **Limited Inputs**: The number of inputs depends on the number of control lines (e.g., 2 control lines allow only 4 inputs).
- **Propagation Delay**: Switching between inputs can introduce delays in high-speed systems.

- **Higher Power Use**: Larger multiplexers with more inputs can consume more power.
- **Signal Loss**: Some signal degradation may occur when combining multiple signals.

What is Demultiplexer (DEMUX)?

Demultiplexer is the opposite of multiplexer. It is also termed as DEMUX. It takes input from one source and also converts the data to transmit towards various sources. The demultiplexer has one data input line. The demultiplexer has several control lines (also known as select lines). These lines determine to which output the input data should be sent. The number of control lines determines the number of output lines.

Given below is the block diagram of the Demultiplexer, It will have one Input line and will give 2n output lines.



Advantages of Demultiplexers (DEMUX)

1. **Efficient Data Distribution**: Routes one input signal to multiple outputs effectively.
2. **Reduced Transmission Complexity**: Simplifies transmitter design by minimizing input lines.
3. **High-Speed Data Splitting**: Ideal for applications requiring rapid data distribution.
4. **Scalability**: Can handle more outputs by increasing control lines.

5. **Versatility**: Widely used in TV broadcasting, communication networks, and more.

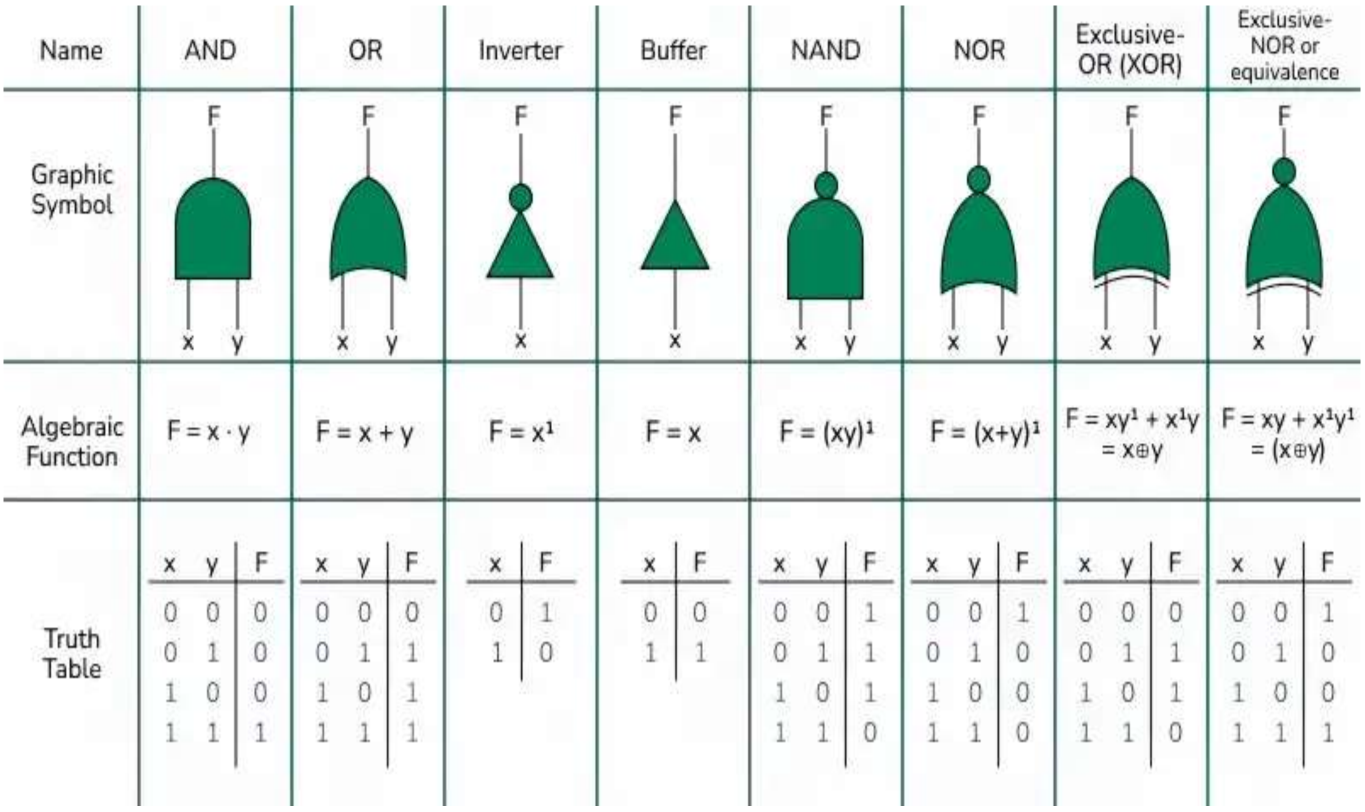
Disadvantages of Demultiplexers (DEMUX)

1. **Control Complexity**: Additional circuits are needed for output selection.
2. **Limited Outputs**: The number of outputs depends on available control lines.
3. **Propagation Delay**: Routing input to outputs may introduce delays in larger systems.
4. **Signal Degradation**: Signal strength may reduce when split into multiple outputs.
5. **Higher Power Consumption**: Power usage increases with additional outputs.
6. **Noise Susceptibility**: Splitting signals can lead to interference

7. Explain about logic gates types with diagrams and truth tables?

What is a Logic Gate?

A logic gates are an electronic circuit that are designed by using electrical components like diodes, transistors, resistors, and more. It is used to perform logical operations based on the inputs provided to it and gives logical output that can be either high(1) or low(0). The operation of logic gates is based on the Boolean algebra or mathematics. Logic gate founds its uses in our day to day basis such as in the architecture of our telephone, laptops, tablets an memory devices.



Name	AND	OR	Inverter	Buffer	NAND	NOR	Exclusive-OR (XOR)	Exclusive-NOR or equivalence
Graphic Symbol	F x y	F x y	F x	F x	F x y	F x y	F x y	F x y
Algebraic Function	$F = x \cdot y$	$F = x + y$	$F = x^1$	$F = x$	$F = (xy)^1$	$F = (x+y)^1$	$F = xy^1 + x^1y$ $= x \oplus y$	$F = xy + x^1y^1$ $= (x \oplus y)$
Truth Table	x y | F 0 0 | 0 0 1 | 0 1 0 | 0 1 1 | 1	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 1	x | F 0 | 1 1 | 0	x | F 0 | 0 1 | 1	x y | F 0 0 | 1 0 1 | 1 1 0 | 1 1 1 | 0	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 0	x y | F 0 0 | 0 0 1 | 1 1 0 | 1 1 1 | 0	x y | F 0 0 | 1 0 1 | 0 1 0 | 0 1 1 | 1

Types of Logic Gates

Logic gates can be broadly classified into three main categories

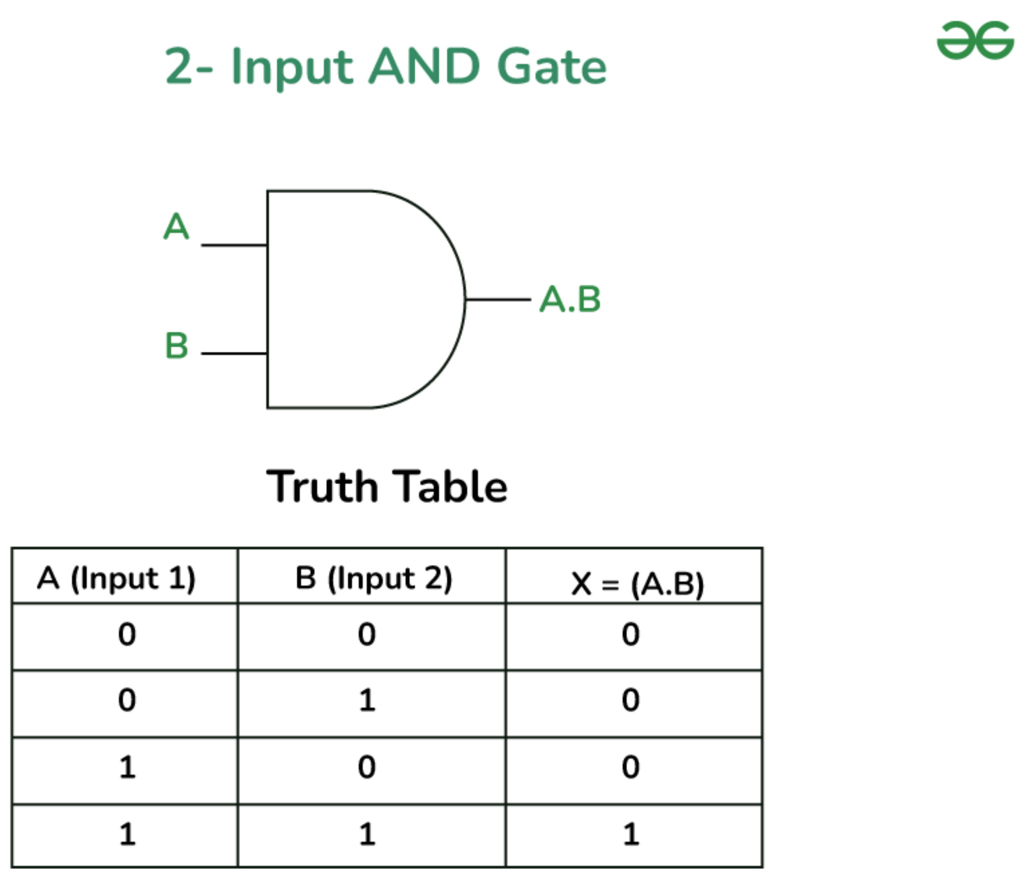
AND GATE

An AND gate is used to perform logical Multiplication of binary input. The Output state of the AND gate will be high(1) if both the input are high(1) ,else the output state will be low(0) if any of the input is low(0).

The Boolean Expression or logic for the AND gate is the logical multiplication of inputs denoted by a full stop or single dot as

A.B=X

The value of X will be True when both the inputs will be True.



Truth Table

A (Input 1)	B (Input 2)	X = (A.B)
0	0	0
0	1	0
1	0	0
1	1	1

Properties of AND Gate

The following are two main properties of the AND gate

- AND gate can accept two or more than two input values at a time.
- When all of the inputs are logic 1, the output of this gate is logic 1.

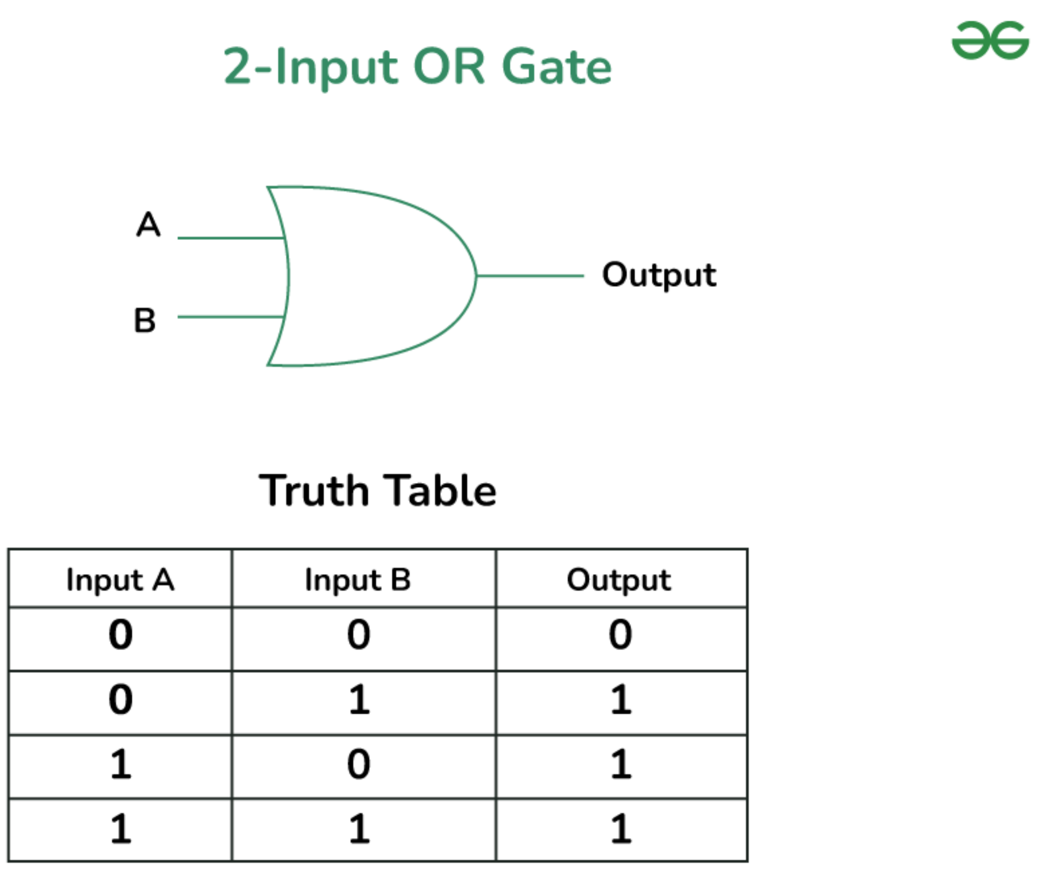
OR GATE

OR GATE is most widely used digital logic circuit. The output state of OR gate will be high i.e.,(1) if any of the input state is high or 1, else output state will be low i.e., 0.

The Boolean Expression for the OR gate is the logical addition of inputs denoted by plus sign(+) as

X= A+B

The value of X will be high(true) when one of the inputs is set to high (true).



Truth Table

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

Properties of OR Gate

An OR gate have the following two properties:

- It can have two or more input lines at a time.
- When all of the inputs to the OR gate are low or logic 0, the output of it is low or logic 0.

NOT GATE

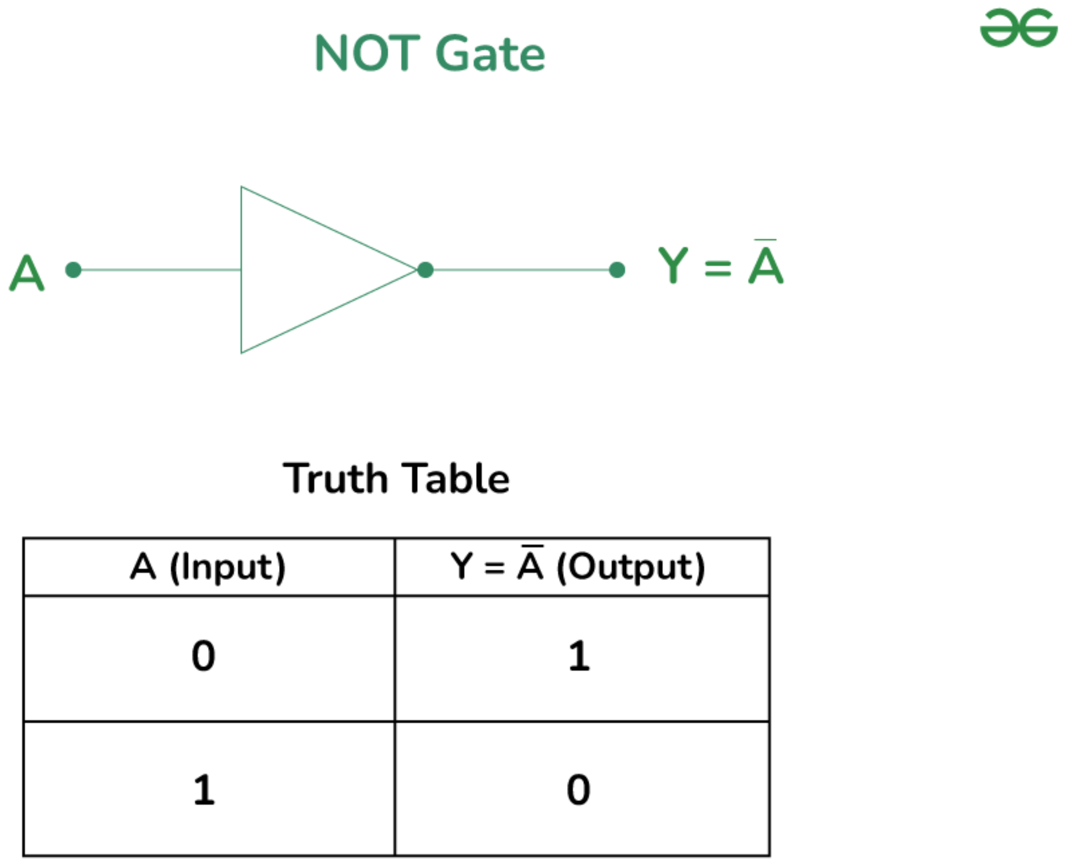
In digital electronics, the NOT gate is one of the basic logic gate having only a single input and a single output. It is also known as inverter or inverting buffer. When the input signal is "low" the output signal is "high" and vice-versa.

The Boolean expression of NOT Gate is as follows

$Y = \bar{A}$ *or*

$Y = A'$

the value of Y will be high when A will be low.



Truth Table

A (Input)	$Y = \bar{A}$ (Output)
0	1
1	0

Properties of NOT Gate

- The output of a NOT gate is complement or inverse of the input applied to it.
- NOT gate takes only one output.

NOR GATE

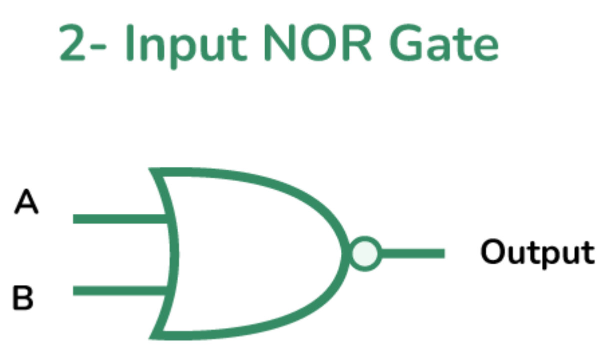
The NOR gate is the type of universal logic gate. It takes two or more inputs and gives only one output. The output state of the NOR gate will be high(1) when all the inputs are low(0). NOR gate returns the complement result of the OR gate. It is basically a combination of two basic logic gates i.e., OR gate and NOT gate.

The Boolean expression of NOR gate is as follows :

If A and B are considered as two inputs, and O as output, then the expression for a two input NOR gate will be

O = (A + B)’

The value of O will be true when all of its inputs are set to 0.



Truth Table

Input A	Input B	0 = (A + B)’
0	0	1
0	1	0
1	0	0
1	1	0

Properties of NOR Gate

The following are two important properties of NOR gate:

- A NOR gate can have two or more inputs and gives an output.
- A NOR gate gives a high or logic 1 output only when its all inputs are low or logic 0.

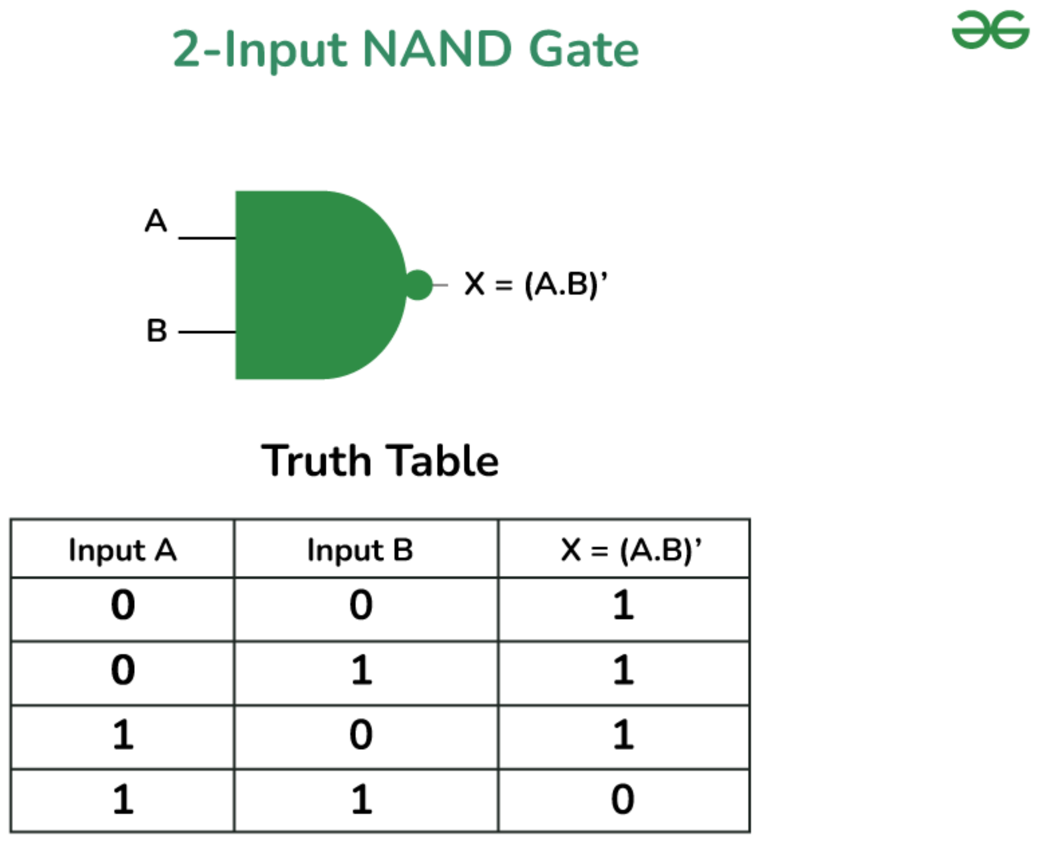
NAND GATE

The NAND Gate is another type of Universal logic gate. The NAND gate or “Not AND” is the combination of two basic logic gates AND gate and the NOT gate connected in series. It takes two or more inputs and gives only one output. The output of the NAND gate will give result high(1) when either of its input is high(1) or both of its input are low(0). In simple, it performs the inverted operation of AND gate.

The Boolean Expression of NAND Gate is as follows

Say we have two inputs, A and B and the output is called X, then the expression is

X = (A . B)'



Input A	Input B	X = (A.B)'
0	0	1
0	1	1
1	0	1
1	1	0

Properties of NAND Gate

The following are the two key properties of NAND Gate

- NAND gate can take two or more inputs at a time and produces one output based on the combination of inputs applied.
- NAND gate produces a low or logic 0 output only when its all inputs are high or logic 1.

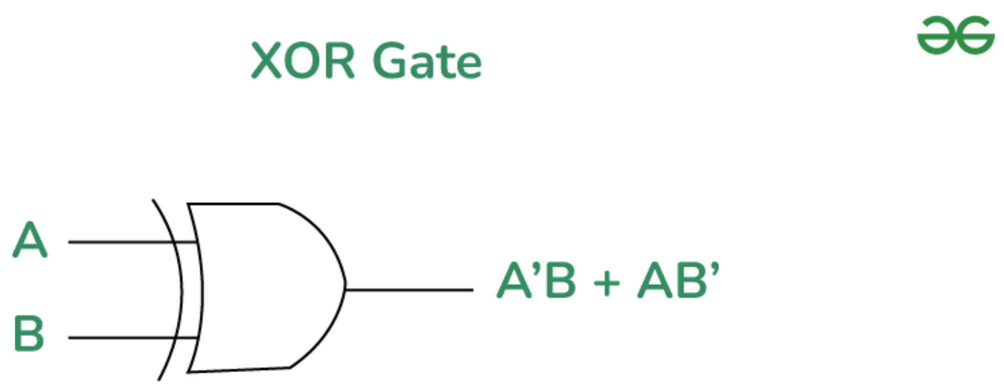
XOR GATE

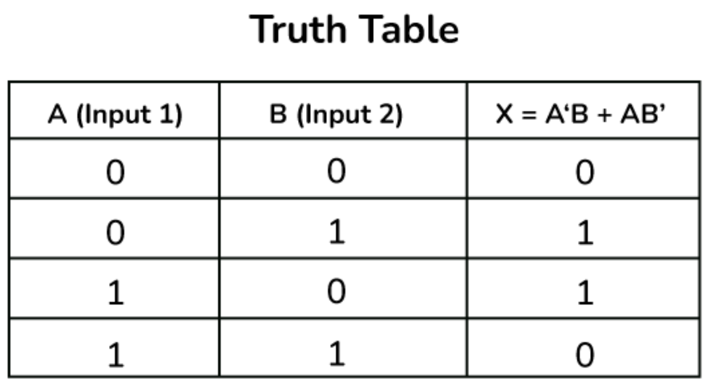
In digital electronics, there is a specially designed logic gate named, XOR gate, which is used in digital circuits to perform **modulo sum**. It is also referred to as **Exclusive OR gate or Ex-OR gate**. it is used extensively in arithmetic logic circuits., logic comparators and error detection circuits. The XOR gate can take only two inputs at a time and give an output. The output of the XOR gate is high(1) only when its two inputs are dissimilar i.e., if one of them is low(0) then other one will be high(1).

Say we have two inputs, A and B and the output is called X, then the expression is

The Boolean expression of XOR Gate is as follows

X = A’B + AB’





Truth Table

A (Input 1)	B (Input 2)	X = A‘B + AB’
0	0	0
0	1	1
1	0	1
1	1	0

Properties of XOR Gate

The following two are the main properties of the XOR gate:

- It can accept only two inputs at a time. There is nothing like a three or more input XOR gate.
- The output of the XOR gate is logic 1 or high, when its inputs are dissimilar.

XNOR GATE

The XNOR is the combination of XOR gate and NOT gate. The output of the XNOR gate is high(1) when both the inputs are high(1) or low(0). In other words, the output of the XNOR gate is high(1) when both the inputs are the same. the XNOR gate can be sometimes be called as Equivalence gate. In simple words, The XNOR gate is the complement of the XOR gate.

The following is the Boolean expression of the XNOR gate,

$Y = A \odot B$

Here, A and B are the input variables and Y is the output variable.

This expression can also be written as follows,

$Y = AB + A'B'$

We can also express the operation of an XNOR gate using XOR gate logic as follows:

$Y = (A \oplus B)'$

Properties of XNOR Gate

The following are two key properties of XNOR gate:

- XNOR gate takes only two inputs and produces one output.
- The output of the XNOR gate is high or logic 1 only when it has similar inputs.

Applications of Logic Gates

Logic gates are the fundamental building blocks of all digital circuits and devices like computers. Here are some key digital devices in which logic gates are utilized to design their circuits.

- Computers
- Microprocessors
- Microcontrollers
- Digital and smart watches

- Smartphones, etc.

Advantages of Logic Gates

- **Basic Functions :** Logic gates carry out basic logical functions like AND, OR, NOT, XOR, NAND, and NOR. All digital operations and dat respective processing rely on these functions.
- **Speed :** Their extremely high speed rates make them an essential feature in today's information processing systems that aim for quickness in data analysis.
- **Reliability :** Being elements whose behaviors are accurately defined means there is no uncertainty about how they behave when used as part of a system.
- **Scalability :** Digital systems complexity increases by interconnecting and replicating these components without significant variations in size or complexity.
- **Low Cost :** Logic Gate costs are relatively low from production viewpoint thus making it popular among those who want to construct digital circuits inexpensively.
- **Low Power Consumption :** Power consumption is minimal with logic gates; hence less energy is needed for operating hence suitable for use with gadgets without batteries or devices running low power consumption applications at all times.

Disadvantages of Logic Gates

Despite their numerous advantages, logic gates have their disadvantages. The following paragraphs discuss some shortcomings of logic gates.

- **Complexity :** The advancement and complexity of digital systems results in increasing number of logic gates and their interconnections, which causes designs that are very difficult to handle and troubleshoot.
- **Propagation Delay :** Small delay in the propagating signal is introduced with every logic gate. When several such gates are chained together, these

delays can add up and have adverse effects on the overall speed and performance of the circuit.

- **Noise Sensitivity :** Even noise, interference, and interfering fields can make logic gates sensitive to errors in the output signal. Proper shielding and conditioning of signals at times are needed to reduce these effects.
- **Power Dissipation :** While logic gates are essentially low power, their dissipation can grow with the complexity of the circuit. Heavy energy loss can generate thermal energy which necessitates supplementary cooling systems.

8. Explain all types of numbering systems briefly?

What is a Number?

A number is a mathematical value used for counting or measuring or labelling objects. Numbers are used to performing arithmetic calculations. Examples of numbers are natural numbers, whole numbers, rational and irrational numbers, etc. 0 is also a number that represents a null value.

A number has many other variations such as even and odd numbers, prime and composite numbers. Even and odd terms are used when a number is divisible by 2 or not, whereas prime and composite differentiate between the numbers that have only two factors and more than two factors, respectively.

In a number system, these numbers are used as digits. 0 and 1 are the most common digits in the number system, that are used to represent binary numbers. On the other hand, 0 to 9 digits are also used for other number systems. Let us learn here the types of number systems.

Types of Number Systems

There are various types of number systems in mathematics. The four most common number system types are:

1. Decimal number system (Base- 10)

2. Binary number system (Base- 2)
3. Octal number system (Base-8)
4. Hexadecimal number system (Base- 16)

Now, let us discuss the different types of number systems with examples.

Decimal Number System (Base 10 Number System)

The decimal number system has a base of 10 because it uses ten digits from 0 to 9. In the decimal number system, the positions successive to the left of the decimal point represent units, tens, hundreds, thousands and so on. This system is expressed in decimal numbers. Every position shows a particular power of the base (10).

Example of Decimal Number System:

The decimal number 1457 consists of the digit 7 in the units position, 5 in the tens place, 4 in the hundreds position, and 1 in the thousands place whose value can be written as:

$(1\times10^3) + (4\times10^2) + (5\times10^1) + (7\times10^0)$

$(1\times1000) + (4\times100) + (5\times10) + (7\times1)$

1000 + 400 + 50 + 7

1457

Binary Number System (Base 2 Number System)

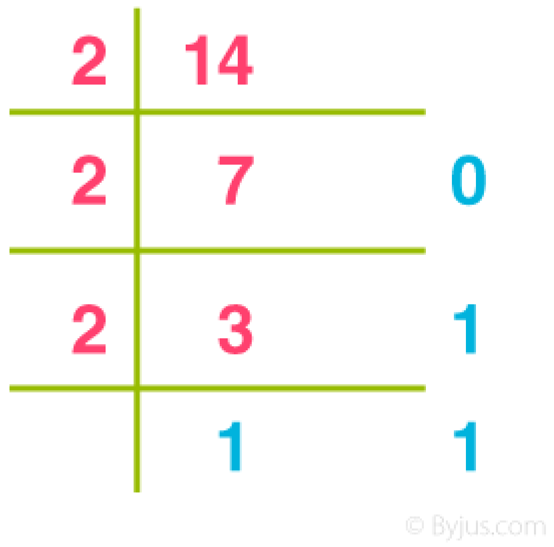
The base 2 number system is also known as the Binary number system wherein, only two binary digits exist, i.e., 0 and 1. Specifically, the usual base-2 is a radix of 2. The figures described under this system are known as binary numbers which are the combination of 0 and 1. For example, 110101 is a binary number.

We can convert any system into binary and vice versa.

Example

Write $(14)_{10}$ as a binary number.

Solution:



Base 2 Number System Example

∴ $(14)_{10} = 1110_2$

Octal Number System (Base 8 Number System)

In the octal number system, the base is 8 and it uses numbers from 0 to 7 to represent numbers. Octal numbers are commonly used in computer applications. Converting an octal number to decimal is the same as decimal conversion and is explained below using an example.

Example: Convert 215_8 into decimal.

Solution:

$215_8 = 2 \times 8^2 + 1 \times 8^1 + 5 \times 8^0$

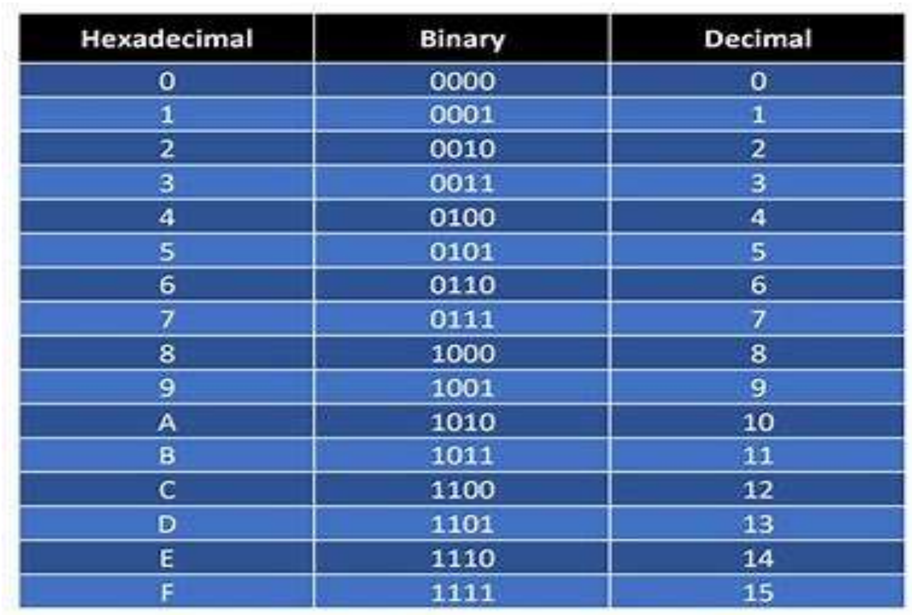
$= 2 \times 64 + 1 \times 8 + 5 \times 1$

$= 128 + 8 + 5$

$= 141_{10}$

Hexadecimal Number System (Base 16 Number System)

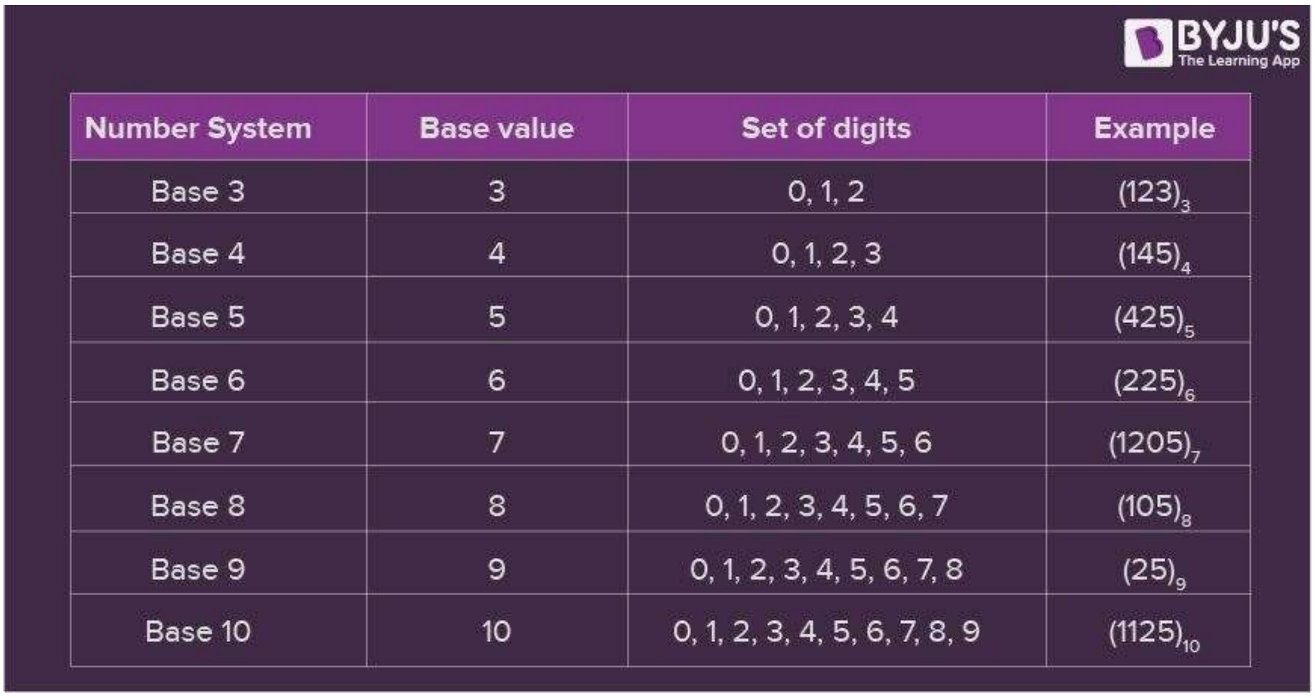
In the hexadecimal system, numbers are written or represented with base 16. In the hexadecimal system, the numbers are first represented just like in the decimal system, i.e. from 0 to 9. Then, the numbers are represented using the alphabet from A to F. The below-given table shows the representation of numbers in the hexadecimal number system.



Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Number System Chart

In the number system chart, the base values and the digits of different number systems can be found. Below is the chart of the numeral system.



Number System	Base value	Set of digits	Example
Base 3	3	0, 1, 2	$(123)_3$
Base 4	4	0, 1, 2, 3	$(145)_4$
Base 5	5	0, 1, 2, 3, 4	$(425)_5$
Base 6	6	0, 1, 2, 3, 4, 5	$(225)_6$
Base 7	7	0, 1, 2, 3, 4, 5, 6	$(1205)_7$
Base 8	8	0, 1, 2, 3, 4, 5, 6, 7	$(105)_8$
Base 9	9	0, 1, 2, 3, 4, 5, 6, 7, 8	$(25)_9$
Base 10	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	$(1125)_{10}$

Number System Chart

Number System Conversion

Numbers can be represented in any of the number system categories like binary, decimal, hexadecimal, etc. Also, any number which is represented in any of the number system types can be easily converted to another. Check the detailed lesson on the conversions of number systems to learn how to convert numbers in decimal to binary and vice versa, hexadecimal to binary and vice versa, and octal to binary and vice versa using various examples.

With the help of the different conversion procedures explained above, now let us discuss in brief about the conversion of one number system to the other number system by taking a random number.

Assume the number 349. Thus, the number 349 in different number systems is as follows:

The number 349 in the binary number system is 101011101

The number 349 in the decimal number system is 349.

The number 349 in the octal number system is 535.

The number 349 in the hexadecimal number system is 15D

Number System Solved Examples

Example 1:

Convert $(1056)_{16}$ to an octal number.

Solution:

Given, 1056_{16} is a hex number.

First we need to convert the given hexadecimal number into decimal number

$(1056)_{16}$

$= 1 \times 16^3 + 0 \times 16^2 + 5 \times 16^1 + 6 \times 160$

$= 4096 + 0 + 80 + 6$

$= (4182)_{10}$

Now we will convert this decimal number to the required octal number by repetitively dividing by 8.

8	4182	Remainder

8	522	6
8	65	2
8	8	1
8	1	0
	0	1

Therefore, taking the value of the remainder from bottom to top, we get;

$(4182)_{10} = (10126)_8$

Therefore,

$(1056)_{16} = (10126)_8$

Example 2:

Convert $(1001001100)_2$ to a decimal number.

Solution:

$(1001001100)_2$

$= 1 \times 2^9 + 0 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

$= 512 + 64 + 8 + 4$

$= (588)_{10}$

Example 3:

Convert 10101_2 into an octal number.

Solution:

Given,

10101_2 is the binary number

We can write the given binary number as,

010 101

Now as we know, in the octal number system,

010 → 2

101 → 5

Therefore, the required octal number is $(25)_8$

Example 4:

Convert hexadecimal 2C to decimal number.

Solution:

We need to convert $2C_{16}$ into binary numbers first.

2C → 00101100

Now convert 00101100_2 into a decimal number.

$101100 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

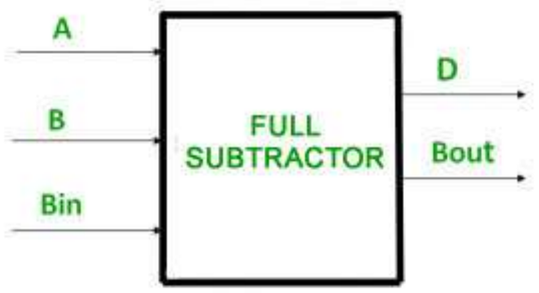
= 32 + 8 + 4

= 44

9. Explain about half subtractor and full subtractor?

Full Subtractor in Digital Logi:-

A full subtractor is a **combinational circuit** that performs subtraction of two bits, one is minuend and other is subtrahend, taking into account borrow of the previous adjacent lower minuend bit. This circuit **has three inputs and two outputs**. The three inputs A, B and Bin, denote the minuend, subtrahend, and previous borrow, respectively. The two outputs, D and Bout represent the difference and output borrow, respectively. Although subtraction is usually achieved by adding the complement of subtrahend to the minuend, it is of academic interest to work out the Truth Table and logic realisation of a full subtractor; x is the minuend; y is the subtrahend; z is the input borrow; D is the difference; and B denotes the output borrow. The corresponding maps for logic functions for outputs of the full subtractor namely difference and borrow.



Here's how a full subtractor works:

1. First, we need to convert the binary numbers to their two's complement form if we are subtracting a negative number.
2. Next, we compare the bits in the minuend and subtrahend at the corresponding positions. If the subtrahend bit is greater than or equal to the minuend bit, we need to borrow from the previous stage (if there is one) to subtract the subtrahend bit from the minuend bit.
3. We subtract the two bits along with the borrow-in to get the difference bit. If the minuend bit is greater than or equal to the subtrahend bit along with the borrow-in, then the difference bit is 1, otherwise it is 0.
4. We then calculate the borrow-out bit by comparing the minuend and subtrahend bits. If the minuend bit is less than the subtrahend bit along with the borrow-in, then we need to borrow for the next stage, so the borrow-out bit is 1, otherwise it is 0.

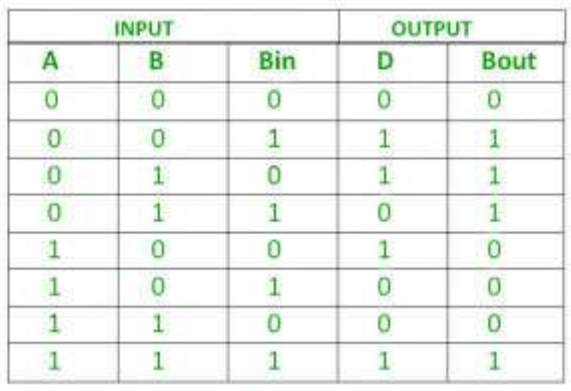
The circuit diagram for a full subtractor usually consists of two half-subtractors and an additional OR gate to calculate the borrow-out bit. The inputs and outputs of the full subtractor are as follows:

Inputs:

A: minuend bit
B: subtrahend bit
Bin: borrow-in bit from the previous stage
Outputs:

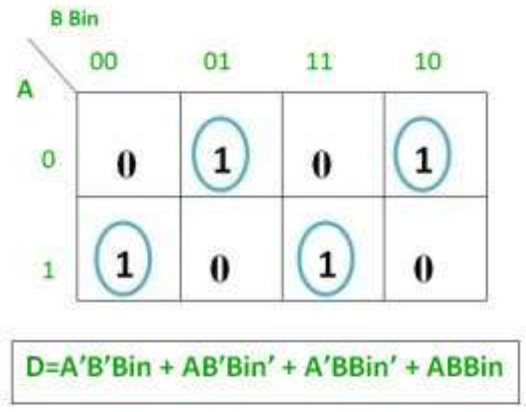
Diff: difference bit
Bout: borrow-out bit for the next stage

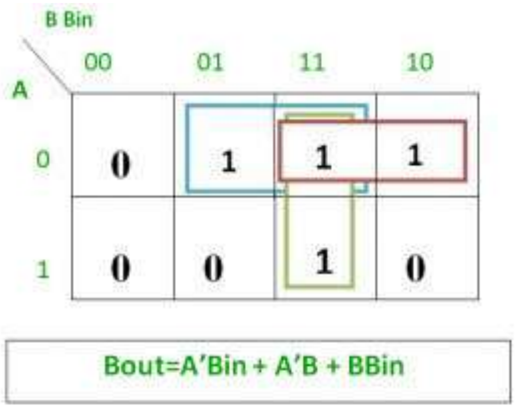
Truth Table –



INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

From above table we can draw the K-Map as shown for “difference” and “borrow”.





Logical expression for difference –

D = A’B’Bin + A’BBin’ + AB’Bin’ + ABBin
= Bin(A’B’ + AB) + Bin’(AB’ + A’B)
= Bin(A XNOR B) + Bin’(A XOR B)
= Bin (A XOR B)’ + Bin’(A XOR B)
= Bin XOR (A XOR B)
= (A XOR B) XOR Bin

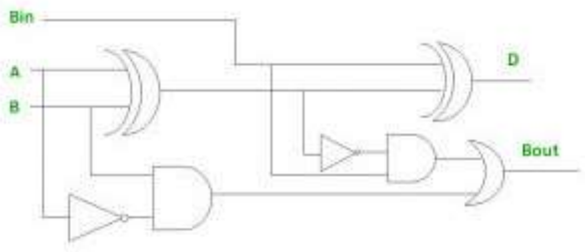
Logical expression for borrow –

Bout = A’B’Bin + A’BBin’ + A’BBin + ABBin
= A’B’Bin +A’BBin’ + A’BBin + A’BBin + A’BBin + ABBin
= A’Bin(B + B’) + A’B(Bin + Bin’) + BBin(A + A’)
= A’Bin + A’B + BBin

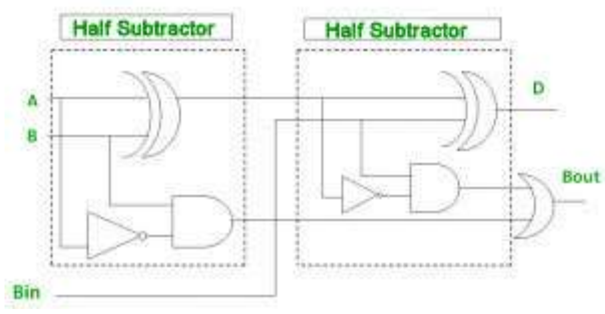
OR

Bout = A’B’Bin + A’BBin’ + A’BBin + ABBin
= Bin(AB + A’B’) + A’B(Bin + Bin’)
= Bin(A XNOR B) + A’B
= Bin (A XOR B)’ + A’B

Logic Circuit for Full Subtractor –



Implementation of Full Subtractor using Half Subtractors – 2 Half Subtractors and an OR gate is required to implement a Full Subtractor.



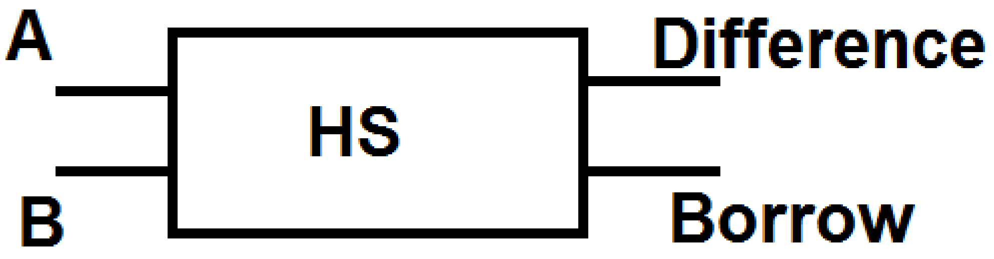
Half Subtractor in Digital Logic:-

A half subtractor is a digital logic circuit that performs binary subtraction of two single-bit binary numbers. It has two inputs, A and B, and two outputs, DIFFERENCE and BORROW. The DIFFERENCE output is the difference between the two input bits, while the BORROW output indicates whether borrowing was necessary during the subtraction.

The half subtractor can be implemented using basic gates such as XOR and NOT gates. The DIFFERENCE output is the XOR of the two inputs A and B, while the BORROW output is the NOT of input A and the AND of inputs A and B.

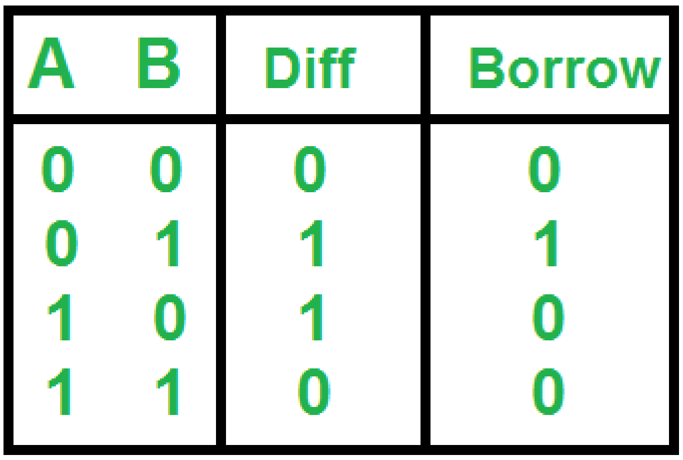
Half Subtractor

Half subtractor is a combination circuit with two inputs and two outputs that are **different** and **borrow**. It produces the difference between the two binary bits at the input and also produces an output (Borrow) to indicate if a 1 has been borrowed. In the subtraction (A-B), A is called a **Minuend bit** and B is called a **Subtrahend bit.**



Half Subtractor

Truth Table



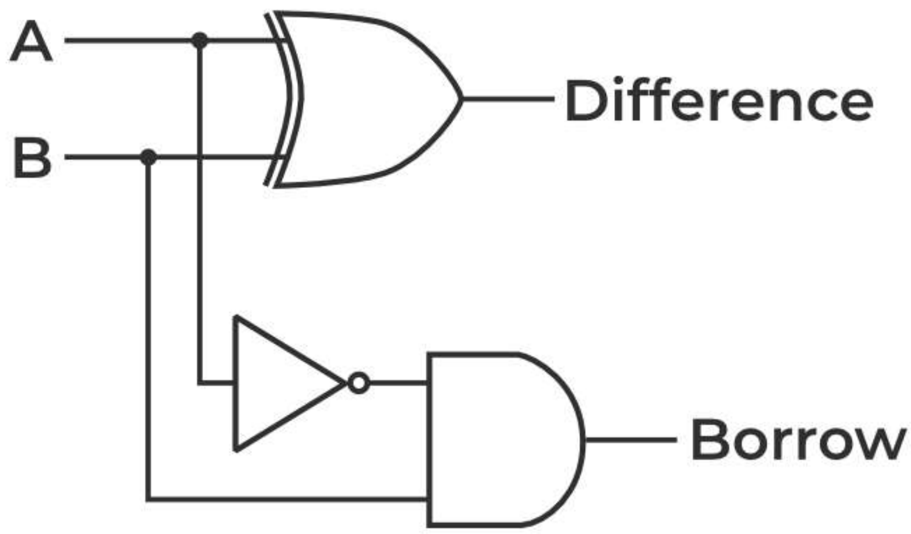
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

The SOP form of the Diff and Borrow is as follows:

Diff= A'B+AB'

Borrow = A'B

Implementation



Logical Expression

Difference = A XOR B
Borrow = \overline{A}B

Advantages of Half Adder and Half Subtractor

1. **Simplicity:** The half adder and half subtractor circuits are simple and easy to design, implement, and debug compared to other binary arithmetic circuits.
2. **Building blocks:** The half adder and half subtractor are basic building blocks that can be used to construct more complex arithmetic circuits, such as full adders and subtractors, multiple-bit adders and subtractors, and carry look-ahead adders.
3. **Low cost:** The half adder and half subtractor circuits use only a few gates, which reduces the cost and power consumption compared to more complex circuits.
4. **Easy integration:** The half adder and half subtractor can be easily integrated with other digital circuits and systems.

Disadvantages of Half Adder and Half Subtractor

1. **Limited functionality:** The half adder and half subtractor can only perform binary addition and subtraction of two single-bit numbers, respectively, and are not suitable for more complex arithmetic operations.
2. **Inefficient for multi-bit numbers:** For multi-bit numbers, multiple half adders or half subtractors need to be cascaded, which increases the complexity and decreases the efficiency of the circuit.
3. **High propagation delay:** The propagation delay of the half adder and half subtractor is higher compared to other arithmetic circuits, which can affect the overall performance of the system.

Application of Half Subtractor in Digital Logic:

1.Calculators: Most mini-computers utilize advanced rationale circuits to perform numerical tasks. A Half Subtractor can be utilized in a number cruncher to deduct two parallel digits from one another.

2.Alarm Frameworks: Many caution frameworks utilize computerized rationale circuits to identify and answer interlopers. A Half Subtractor can be utilized in these frameworks to look at the upsides of two parallel pieces and trigger a caution in the event that they are unique.

3.Automotive Frameworks: Numerous advanced vehicles utilize computerized rationale circuits to control different capabilities, like the motor administration framework, stopping mechanism, and theater setup. A Half Subtractor can be utilized in these frameworks to perform computations and examinations.

4.Security Frameworks: Advanced rationale circuits are usually utilized in security frameworks to identify and answer dangers. A Half Subtractor can be utilized in these frameworks to look at two double qualities and trigger a caution in the event that they are unique.

5.Computer Frameworks: Advanced rationale circuits are utilized broadly in PC frameworks to perform estimations and examinations. A Half Subtractor can be utilized in a PC framework to deduct two paired values from one another.

10. Once and twos compliment example for subtraction of 2’scomplement?

1's Complement

- 1's complement is used when we want to alter and **invert all the significant binary digits of any number.**
- It **changes all 0 to 1 and all 1 to 0** in the given number and the final generated number is 1's complement.
- Let's take **Decimal 5 as an example** which is equivalent to 101 in Binary.
- If we generate its 1's complement then **it will be 010 which is 2 in Decimal.**

2's complement

- 2's complement is **used to store and signify signed and negative numbers.**
- If **we add 1 to LSB of any number 1's complement** then it will result in 2's complement.
- If we take **Decimal 5 as an example** then we know that 1's complement of it is 010
- So 2's complement will be 011 which **is equivalent to 3 in Decimal and it is the 2's complement for the given number.**

Example for 2’s compliment:--

2's complement of "0111" is "1001"

2's complement of "1100" is "0100"

Another trick to finding two’s complement:

Step 1: Start from the Least Significant Bit and traverse left until you find a 1. Until you find 1, the bits stay the same

Step 2: Once you have found 1, let the 1 as it is, and now

Step 3: Flip all the bits left into the 1.

Illustration

Suppose we need to find 2s Complement of 100100

Step 1: Traverse and let the bit stay the same until you find 1. Here x is not known yet. Answer = xxxx00 –

Step 2: You found 1. Let it stay the same. Answer = xxx100

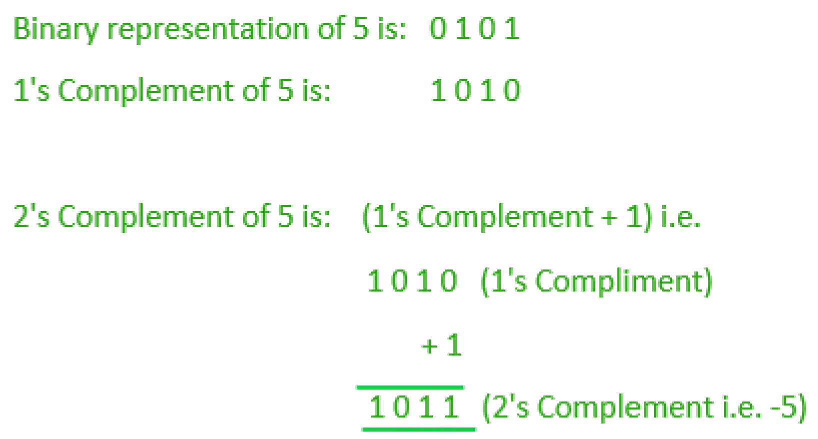
Step 3: Flip all the bits left into the 1. Answer = 011100.

Hence, the 2s complement of 100100 is 011100.

For one's complement, we simply need to flip all bits.
For 2's complement, we first find one's complement. We traverse the one's complement starting from LSB (least significant bit), and look for 0. We flip all 1's (change to 0) until we find a 0. Finally, we flip the found 0. For example, 2's complement of "01000" is "11000" (Note that we first find one's complement of 01000 as 10111). If there are all 1's (in one's complement), we add an extra 1 in the string. For example, 2's complement of "000" is "1000" (1's complement of "000" is "111").

Subtraction of two numbers using 2's Complement

Given two numbers a and b. The task is to subtract b from a by using 2's Complement method.
Note: Negative numbers represented as 2's Complement of Positive Numbers.
For example, -5 can be represented in binary form as 2's Complement of 5.
Look at the image below:



Binary representation of 5 is: 0 1 0 1

1's Complement of 5 is: 1 0 1 0

2's Complement of 5 is: (1's Complement + 1) i.e.

```
 1 0 1 0  (1's Compliment)
     + 1
 -------
 1 0 1 1  (2's Complement i.e. -5)
```

Examples:

Input : a = 2, b = 3

Output : -1

Input : a = 9, b = 7

Output : 2

To subtract b from a. Write the expression (a-b) as:

(a - b) = a + (-b)

Now (-b) can be written as (2’s complement of b). So the above expression can be now written as:

(a - b) = a + (2's complement of b)

So, the problem now reduces to “Add a to the 2’s complement of b“. The below image illustrates the above method of subtraction for the first example where a = 2 and b = 3.

Binary representation of 3 is: 0 0 1 1

1's Complement of 3 is: 1 1 0 0

2's Complement of 3 is: (1's Complement + 1) i.e.

```
1 1 0 0   (1's Compliment)
    + 1
-------
1 1 0 1   (2's Complement i.e. -3)
-------
```

```
Now 2 + (-3) =   0 0 1 0 (2 in binary)
               + 1 1 0 1 (-3)
                 -------
                 1 1 1 1 (-1)
                 -------
```

Now, to check whether it is -1 or not simply. takes 2's

Complement of -1 and kept -ve sign as it is.

-1 = 1 1 1 1

```
2's Complement = -(0 0 0 0)
                       + 1
                  ---------
                 -(0 0 0 1) i.e. -1
                  ---------
```

UNIT-1 PART-A(2MARKS)

1. Types of numbering system example?

There are Four types of Numbering systems they are

1. Decimal number system
2. Binary number system
3. octal lumber system
4. Hexadecimal number system

<u>1. Decimal number system:-</u>

A decimal number is a **number that consists of a whole number and a fractional part separated by a point**. The dot present between the whole number and fraction's part is called the decimal point. The number of digits in the decimal part determines the number of decimal places. Decimals can also be considered as fractions only when the denominators are 10, 100, 1000 etc. Examples of decimal numbers include 34.5, 3.5, 6.79, 78.32, 17.235, 0.149, 125.005, 2534.0.

<u>2. Binary number system:-</u>

Binary Number System: According to digital electronics and mathematics, a binary number is defined as a number that is expressed in the binary system or base 2 numeral system. It describes numeric values by two separate symbols; 1 (one) and 0 (zero). The base-2 system is the positional notation with 2 as a radix.

The binary system is applied internally by almost all latest computers and computer-based devices because of its direct implementation in electronic circuits using logic gates. Every digit is referred to as a bit.

Example: Convert 4 in binary.

Solution: 4 in binary is $(100)_2$.

<u>3. octal lumber system:-</u>

A number system which has its base as ‘eight’ is called an Octal number system. It uses numbers from 0 to 7. Let us take an example, to understand the concept. As we said, any number with base 8 is an octal number

example: -24_8, 109_8, 55_8, etc.

4. Hexadecimal number system:-

A **hexadecimal number** (or simply "hex") is a base-16 numbering system. Unlike the decimal system (base-10), which uses digits 0-9, the hexadecimal system uses sixteen symbols: the numbers 0 through 9 and the letters A through F, where:

- **A = 10**
- **B = 11**
- **C = 12**
- **D = 13**
- **E = 14**
- **F = 15**

Example:

The hexadecimal number **1A3** can be expanded as: $$1A3_{16} = (1 \times 16^2) + (10 \times 16^1) + (3 \times 16^0) = 256 + 160 + 3 = 419_{10}$$

So, **1A3 in hex is 419 in decimal**.

This system is widely used in computing and digital electronics because it can represent large numbers with fewer digits compared to decimal or binary.

2. Definition of one’s complement with example?

One's complement is a binary number representation technique primarily used in computer systems for signed number representations. It involves flipping all the bits of a binary number, which means changing every 0 to a 1 and every 1 to a 0. It's often used in the arithmetic of negative numbers.

Example:

If we have the binary number 1010 (in 4 bits):

- Its one's complement is obtained by inverting the bits: 0101.

3. What is meant by logic gate and how many types are ?

A **logic gate** is a fundamental building block in digital circuits. It is an electronic component that performs a logical operation on one or more binary inputs to produce a single binary output. Logic gates follow the principles of Boolean algebra, where inputs and outputs are in the form of **0s (false)** and **1s (true)**.

Types of Logic Gates:

There are seven basic types of logic gates:

1. **AND Gate**
    - Output is true (1) only if all inputs are true (1).
    - Example: A·BA \cdot B
2. **OR Gate**
    - Output is true (1) if at least one input is true (1).
    - Example: A+BA + B
3. **NOT Gate (Inverter)**
    - Output is the opposite of the input.
    - Example: A¯\overline{A}
4. **NAND Gate** (NOT + AND)
    - Output is the opposite of the AND gate.
    - Example: A·B¯\overline{A \cdot B}
5. **NOR Gate** (NOT + OR)
    - Output is the opposite of the OR gate.
    - Example: A+B¯\overline{A + B}
6. **XOR Gate (Exclusive OR)**
    - Output is true (1) if only one input is true (1).
    - Example: A⊕BA \oplus B

7. **XNOR Gate (Exclusive NOR)**
   - Output is true (1) if inputs are equal.
   - Example: A⊕B¯

4. Definition of 4' complement with example?

The **4's complement** is a concept specific to **quaternary (base-4) number systems**. It is the method used to find the complement of a number in base-4 arithmetic. The 4's complement of a number is similar to the 10's complement in decimal or the 2's complement in binary—it helps in performing subtraction by addition.

To find the 4's complement of a number:

1. Subtract each digit from 33 (since 4−1=34 - 1 = 3) to get the complement.
2. Add 11 to the least significant digit (LSD) of the resulting number.

Example:

Let's find the 4's complement of **2301** in base-4:

1. Subtract each digit from 33: 3−2=1,3−3=0,3−0=3,3−1=23 - 2 = 1, 3 - 3 = 0, 3 - 0 = 3, 3 - 1 = 2. So, the complement is 10321032.
2. Add 11 to the LSD of 10321032: 1032+1=10331032 + 1 = 1033.

Thus, the **4's complement of 2301** in base-4 is **1033**.

5. Definition of half adder and full adder?

A Half Adder and a Full Adder are basic building blocks in digital electronics used for performing binary addition.

Half Adder

- A Half Adder is a combinational circuit that adds two binary digits (A and B) and produces two outputs: Sum (S) and Carry (C).
- It does not account for any carry input from a previous stage of addition.

Logic:

- Sum = A⊕BA \oplus B (XOR operation)

- Carry = A·BA \cdot B (AND operation)

Example:

If A=1A = 1 and B=1B = 1:

- Sum = 1⊕1=01 \oplus 1 = 0
- Carry = 1·1=11 \cdot 1 = 1

Full Adder

- A Full Adder is a combinational circuit that adds three binary digits: A, B, and a Carry-in (Cin). It produces two outputs: Sum (S) and Carry-out (Cout).
- It is capable of handling carry inputs, which makes it suitable for multi-bit binary addition.

Logic:

- Sum = A⊕B⊕CinA \oplus B \oplus Cin
- Carry-out = (A·B)+(B·Cin)+(A·Cin)(A \cdot B) + (B \cdot Cin) + (A \cdot Cin)

Example:

If A=1A = 1, B=1B = 1, and Cin=1Cin = 1:

- Sum = 1⊕1⊕1=11 \oplus 1 \oplus 1 = 1
- Carry-out = (1·1)+(1·1)+(1·1)=1(1 \cdot 1) + (1 \cdot 1) + (1 \cdot 1) = 1

In summary:

- Half Adder: Adds two bits, no carry-in.
- Full Adder: Adds three bits (including carry-in).

6. Definition of 9's complement with example?

The **9's complement** is a method used in decimal (base-10) arithmetic to simplify subtraction by addition. It is calculated by subtracting each digit of a number from 9.

Example:

Let's find the 9's complement of **456**:

1. Subtract each digit from 9:
    - 9−4=59 - 4 = 5
    - 9−5=49 - 5 = 4
    - 9−6=39 - 6 = 3

So, the 9's complement of **456** is **543**.

7. Definition of Half subtractor and full subtractor?

A **Half Subtractor** and a **Full Subtractor** are combinational circuits used in digital electronics to perform binary subtraction.

Half Subtractor

- A Half Subtractor subtracts two binary digits (**A** and **B**) and produces two outputs: **Difference (D)** and **Borrow (B)**.
- It does not account for any borrow input from a previous stage.

Logic:

- **Difference** = A⨁BA \oplus B (XOR operation)
- **Borrow** = A¯·B\overline{A} \cdot B (NOT-AND operation)

Example:

If A=1A = 1 and B=1B = 1:

- Difference = 1⨁1=01 \oplus 1 = 0
- Borrow = 1¯·1=0\overline{1} \cdot 1 = 0

Full Subtractor

- A Full Subtractor subtracts three binary digits: **A**, **B**, and a **Borrow-in (Bin)**. It produces two outputs: **Difference (D)** and **Borrow-out (Bout)**.
- It accounts for borrow input, making it suitable for multi-bit binary subtraction.

Logic:

- **Difference** = A⨁B⨁BinA \oplus B \oplus Bin

- **Borrow-out** = A¯·(B+Bin)+(B·Bin)\overline{A} \cdot (B + Bin) + (B \cdot Bin)

Example:

If A=1A = 1, B=1B = 1, and Bin=1Bin = 1:

- Difference = 1⊕1⊕1=11 \oplus 1 \oplus 1 = 1
- Borrow-out = 1¯·(1+1)+(1·1)=1\overline{1} \cdot (1 + 1) + (1 \cdot 1) = 1

8. What is mean by K mapping and it is developed by?

K mapping, or **Karnaugh Mapping**, is a graphical method used to simplify Boolean algebra expressions. It organizes truth table values into a grid-like format, making it easier to minimize logic functions without using complex Boolean algebra theorems. This simplification is crucial in designing efficient digital circuits.

The Karnaugh Map was developed by **Maurice Karnaugh** in 1953 while he was working at Bell Labs. It is widely used in digital electronics to optimize logic circuits and reduce the number of gates required.

9. Why do we use base two for binary numbers?

In Binary number we have zeros and ones that's why we represent base as 2

Example:- (10010111)2

10.Definition of 2's complement with example?

The 2's complement is a method used in binary arithmetic to represent both positive and negative integers. It simplifies subtraction operations by allowing them to be performed as addition.

How to Find the 2's Complement:

1. Invert all the bits of the binary number (change 0 to 1 and 1 to 0).
2. Add 1 to the least significant bit (LSB) of the inverted number.

Example:

Let's find the 2's complement of 01010101 (binary for 5):

1. Invert the bits: 10101010.

2. Add 1: 1010+1=10111010 + 1 = 1011.

So, the 2's complement of 01010101 is 10111011, which represents −5-5 in binary.

UNIT-2: DIGITAL LOGICS AND COMPUTER ORGANIZATION

1.EXPLAIN ABOUT FLIP FLOPS ?

ANS: A flipflop is an electronic circuit with two stable states that can be used to store one bit of binary information. the stored data can be changed by applying various inputs.

A basic flipflop can be constructed using four nand gates or four nor gates. flip flop is known as basic digital memory circuit. It has two states as logic1(high) and logic 0(low) states. A flip flop is a sequential circuit consist of single binary state of data.

TYPES OF FLIPFLOPS:

1.SR FLIPFLOP

2.JK FLIPFLOP

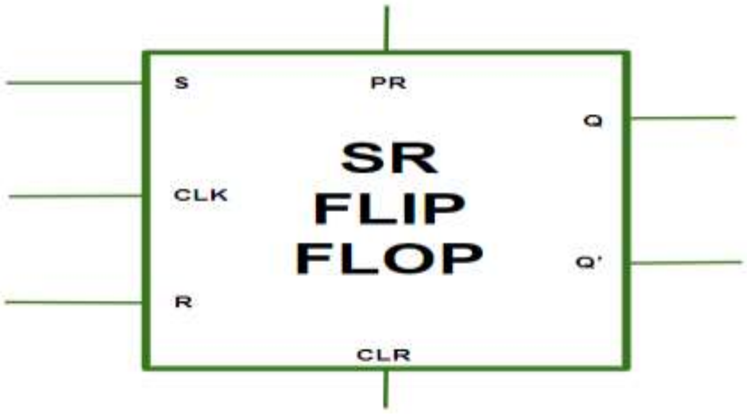
3.D FLIPFLOP

4.T FLIPFLOPL

S-R Flip Flop

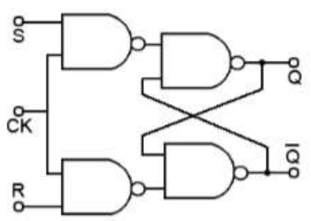
In the flip flop, with the help of preset and clear when the power is switched ON, the states of the circuit keeps on changing, that is it is uncertain. It may come to set(Q=1) or reset(Q’=0) state. In many applications, it is desired to initially set or reset the flip flop that is the initial state of the flip flop that needs to be assigned. This thing is accomplished by the preset(PR) and the clear(CLR).

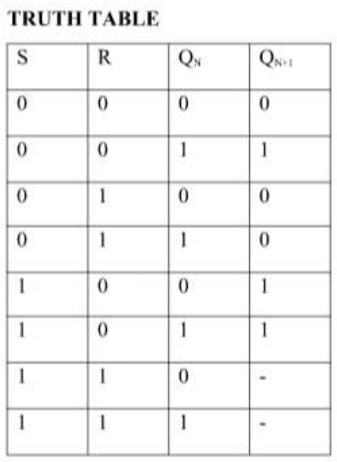
Block Diagram of S-R Flip Flop



Circuit Diagram and Truth Table of S-R Flip Flop

Given Below is the Diagram of S-R Flip Flop with its Truth Table





TRUTH TABLE

S	R	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	-
1	1	1	-

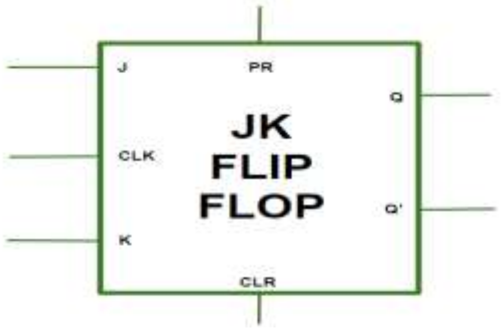
Characteristics Equation for SR Flip Flop

QN+1 = QNR’ + SR’

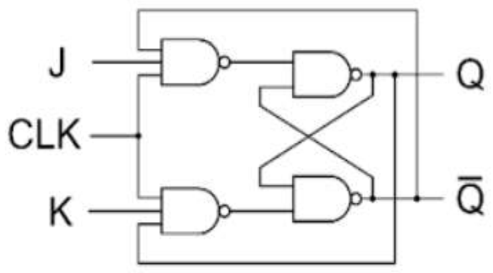
J-K Flip Flop

In JK flip flops, The basic structure of the flip flop which consists of Clock (CLK), Clear (clr)

Preset (PR).



Circuit Diagram and Truth Table of J-K Flip Flop



TRUTH TABLE

J	K	Q_N	Q_{N+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Characteristics Equation for JK Flip Flop

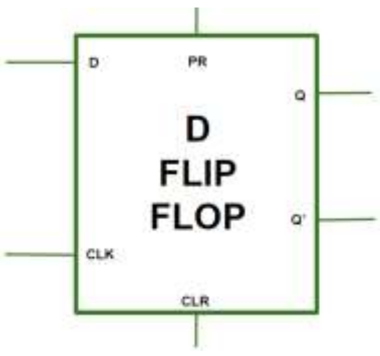
QN+1 = JQ’N + K’QN

<u>D Flip Flop:</u>

The D Flip Flop Consists a single data input(D), a clock input(CLK),and two outputs: Q and Q’ (the complement of Q).

Block Diagram of D Flip Flop

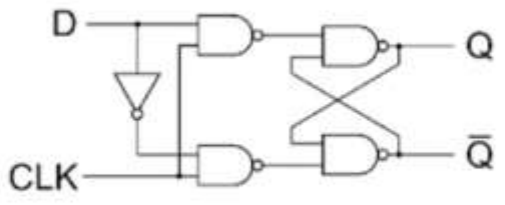
Given Below is the Block Diagram of [D Flip Flop]



D FLIP FLOP

Circuit Diagram and Truth Table of D Flip Flop

Given Below is the Diagram of D Flip Flop with its Truth Table



Q	D	Q(t+1)
0	0	0
0	1	1
1	0	0
1	1	1

Characteristics Equation for D Flip Flop

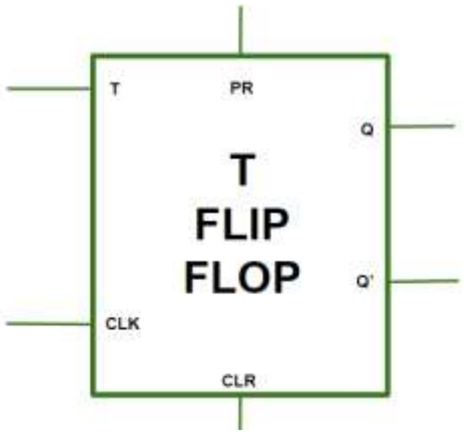
QN+1 = D

<u>T Flip Flop:</u>

The T Flip Flop consists of data input (T), a clock input (CLK), and two outputs: Q and Q' (the complement of Q).

Block Diagram of T Flip Flop

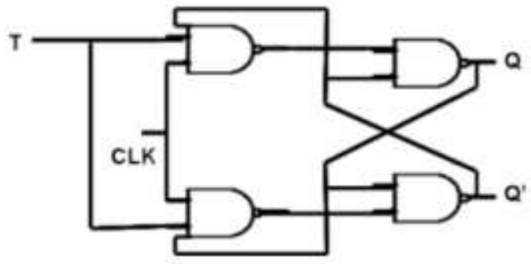
Given Below is the Block Diagram of T Flip Flop



T FLIP FLOP

Circuit Diagram and Truth Table of T Flip Flop

Given Below is the Circuit Diagram and Truth Table of T Flip Flop



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

characteristics Equation for T Flip Flop

QN+1 = Q’NT + QNT’ = QN XOR T

Applications of Flip-Flops:

Frequency Dividers

- ***Shift Registers***
- ***Storage Registers***
- ***Bounce elimination switch***
- ***Data storage***
- ***Data transfer.***
- ***Latch***
- ***Registers***
- ***Memory***

2.EXPLAIN ABOUT COUNTERS AND SEQUENCIAL CIRCUITS ?

A **Counter** is a device which stores (and sometimes displays) the number of times a particular event or process has occurred, often in relationship to a clock signal. Counters are used in digital electronics for counting purpose, they can count specific event happening in the circuit. they can also be designed with the help of flip flops. They are used as frequency dividers where the frequency of given pulse waveform is divided. Counters are sequential circuit that count the number of pulses can be either in binary code or BCD form. The main properties of a counter are timing , sequencing , and counting. Counter works in two modes .

Counter Classification

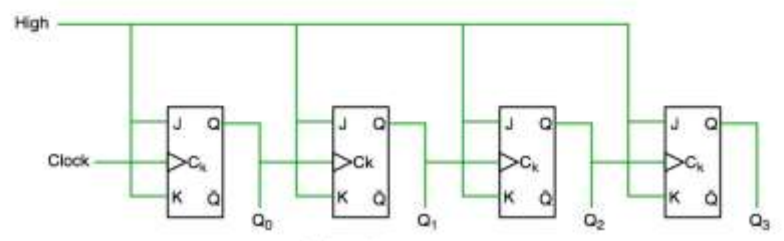
Counters are broadly divided into two categories

1. Asynchronous counter

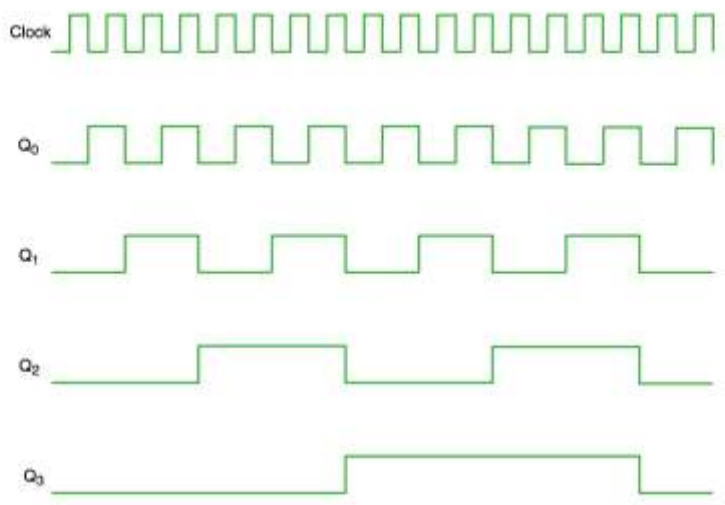
2. Synchronous counter

1. **Asynchronous Counter**

In asynchronous counter we don’t use universal clock, only first flip flop is driven by main clock and the clock input of rest of the following flip flop is driven by output of previous flip flops. We can understand it by following diagram-



(a) Asynchronous counter

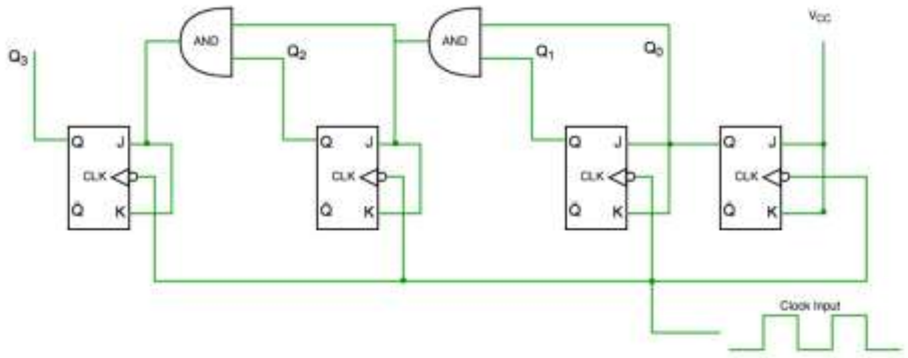


(b) Timing Diagram

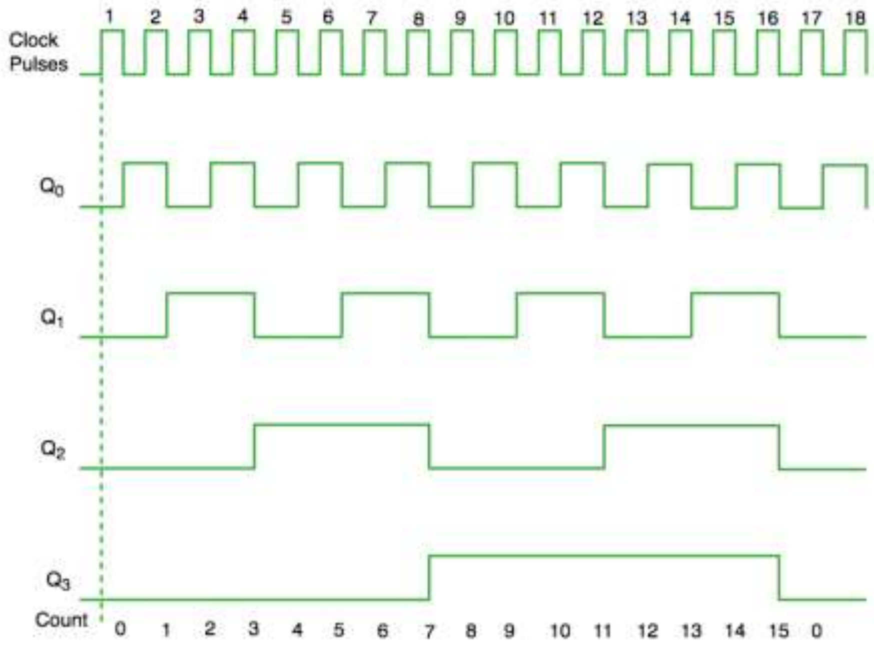
It is evident from timing diagram that Q0 is changing as soon as the rising edge of clock pulse is encountered, Q1 is changing when rising edge of Q0 is encountered(because Q0 is like clock pulse for second flip flop) and so on. In this way ripples are generated through Q0,Q1,Q2,Q3 hence it is also called **RIPPLE counter and serial counter.** A ripple counter is a cascaded arrangement of flip flops where the output of one flip flop drives the clock input of the following flip flop

2. **Synchronous Counter**

Unlike the asynchronous counter, synchronous counter has one global clock which drives each flip flop so output changes in parallel. The one advantage of synchronous counter over asynchronous counter is, it can operate on higher frequency than asynchronous counter as it does not have cumulative delay because of same clock is given to each flip flop. It is also called as parallel counter.



Synchronous counter circuit



Application of Counters

- Frequency counters
- Digital clock

- Time measurement
- A to D converter
- Frequency divider circuits
- Digital triangular wave generator.

<u>SEQUENCIAL CIRCUITS:</u> Sequential circuits are digital circuits that store and use previous state information to determine their next state. They are commonly used in digital systems to implement state machines, timers, counters, and memory elements and are essential components in digital systems design.

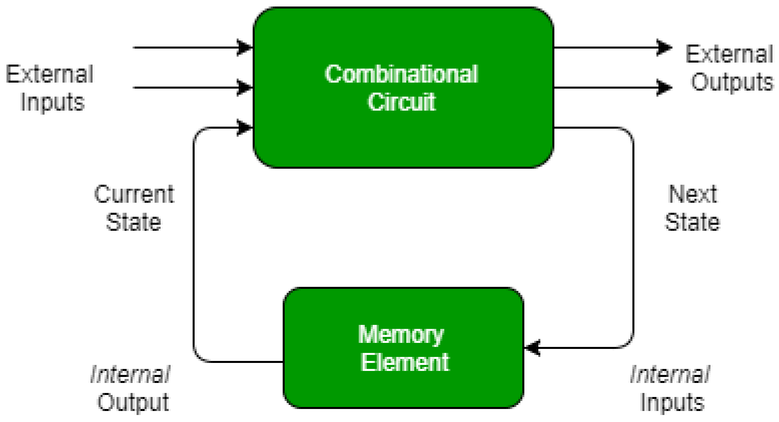


Figure: Sequential Circuit

Types of Sequential Circuits

There are two types of sequential circuits

Asynchronous Sequential Circuit

These circuits **do not use a clock signal** but uses the pulses of the inputs. These circuits are **faster** than synchronous sequential circuits because there is clock pulse and change their state immediately when there is a change in the input signal. We use asynchronous sequential circuits when speed of operation is important and **independent** of internal clock pulse.

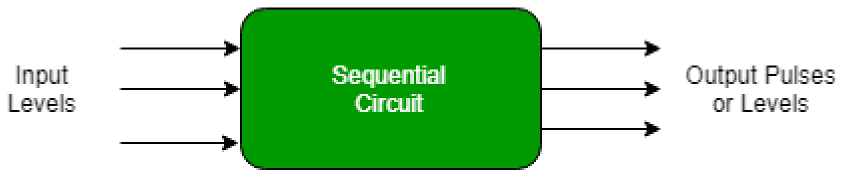


Figure: Asynchronous Sequential Circuit

But these circuits are more **difficult** to design and their output is **uncertain**.

Synchronous Sequential Circuit

These circuits **uses clock signal** and level inputs (or pulsed) (with restrictions on pulse width and circuit propagation). The output pulse is the same duration as the clock pulse for the clocked sequential circuits. Since they wait for the next clock pulse to arrive to perform the next operation, so these circuits are bit **slower** compared to asynchronous. Level output changes state at the start of an input pulse and remains in that until the next input or clock pulse.

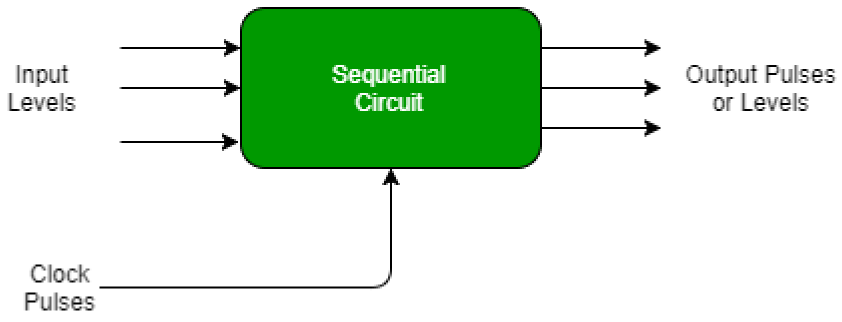


Figure: Synchronous Sequential Circuit

3.EXPLAIN ABOUT SHIFT REGISTERS ?

A **Register** is a device that is used to store such information. It is a group of flip-flops connected in series used to store multiple bits of data. The information stored within these registers can be transferred with the help of **shift registers**.

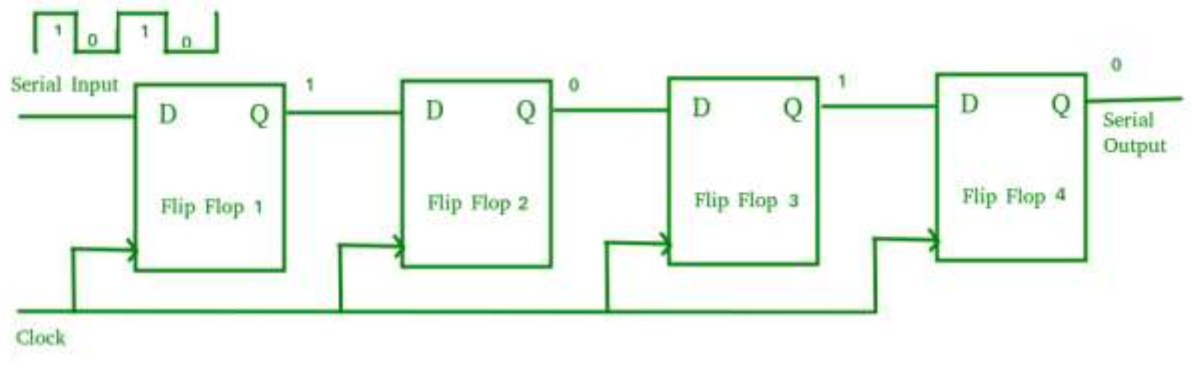
Shift Register is a group of flip flops used to store multiple bits of data. The bits stored in such registers can be made to move within the registers and in/out of the registers by applying clock pulses. An n-bit shift register can be formed by connecting n flip-flops where each flip-flop stores a single bit of data. The registers which will shift the bits to the left are called "**Shift left registers".**

Types of Shift Registers

- **Serial In Serial Out shift register**
- **Serial In parallel Out shift register**
- **Parallel In Serial Out shift register**
- **Parallel In parallel Out shift register**

Serial-In Serial-Out Shift Register (SISO):

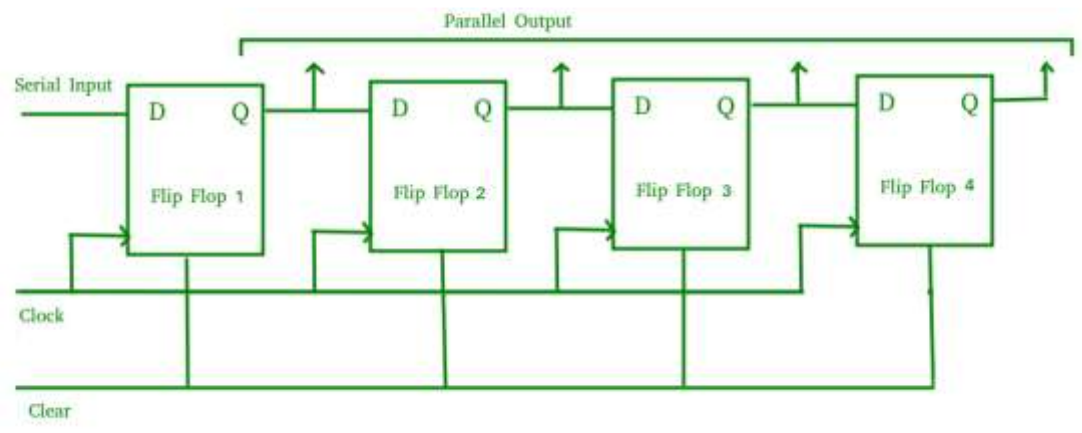
- The shift register, which allows serial input (one bit after the other through a single data line) and produces a serial output is known as a Serial-In Serial-Out shift register. Since there is only one output, the data leaves the shift register one bit at a time in a serial pattern, thus the name Serial-In Serial-Out Shift.



Serial-In Parallel-Out Shift Register (SIPO)

The shift register, which allows serial input (one bit after the other through a single data line) and produces a parallel output is known as the Serial-In Parallel-Out shift register.

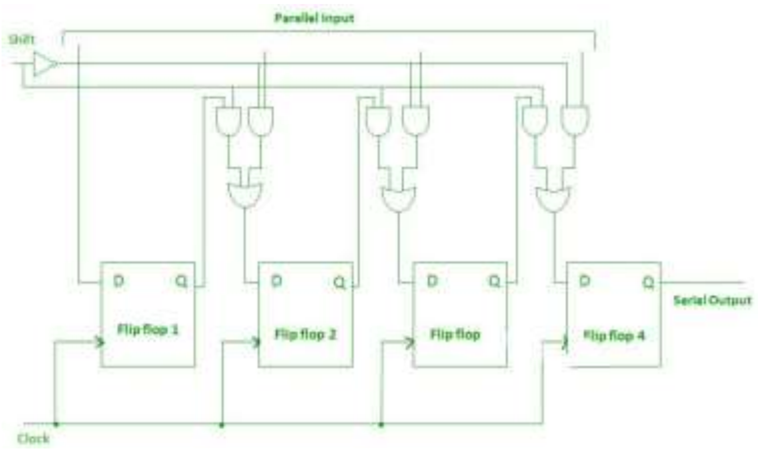
The circuit consists of four D flip-flops which are connected. The clear (CLR) signal is connected in addition to the clock signal to all 4 flip flops in order to RESET them. The output of the first flip-flop is connected to the input of the next flip flop and so on. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip-flop.



Parallel-In Serial-Out Shift Register (PISO)

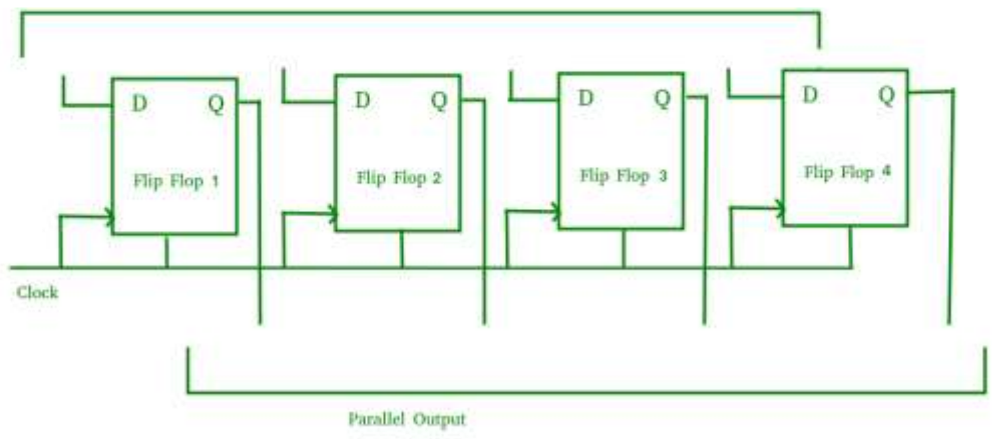
The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and produces a serial output is known as a Parallel-In Serial-Out shift register.

The circuit consists of four D flip-flops which are connected. The clock input is directly connected to all the flip-flops but the input data is connected individually to each flip-flop through a multiplexer at the input of every flip-flop. The output of the previous flip-flop and parallel data input are connected to the input of the MUX and the output of MUX is connected to the next flip-flop. All these flip-flops are synchronous with each other since the same clock signal is applied to each flip-flop.



Parallel-In Parallel-Out Shift Register (PIPO)

The shift register, which allows parallel input (data is given separately to each flip flop and in a simultaneous manner) and also produces a parallel output is known as Parallel-In parallel-Out shift register.



Applications of Shift Registers

- The shift registers are used for temporary data storage.
- The shift registers are also used for data transfer and data manipulation.
- The serial-in serial-out and parallel-in parallel-out shift registers are used to produce time delay to digital circuits.
- The serial-in parallel-out shift register is used to convert serial data into parallel data thus they are used in communication lines where demultiplexing of a data line into several parallel lines is required.
- A Parallel in Serial out shift register is used to convert parallel data to serial data.

4.EXPLAIN BREIFLY ABOUT TYPES OF COMPUTERS AND CHARACTERISTICS ?

COMPUTER:

A computer is an electronic device that has storage, computations, input (data), output (data) and networking capabilities. A computer processes the input according to the set of instructions provided to it by the user and gives the desired output.

Types of Computer

- Super Computer
- Mainframe computer
- Mini Computer

- Workstation Computer
- Personal Computer (PC)
- Digital Computer
- Hybrid Computer
- Tablets and Smartphone

They are the biggest and fastest computers (in terms of speed of processing data). Supercomputers are designed such that they can process a huge amount of data, like processing trillions of instructions or data just in a second. This is because of the thousands of interconnected processors in supercomputers. It is basically used in scientific and engineering applications such as weather forecasting, scientific simulations, and nuclear energy research. It was first developed by Roger Cray in 1976.

Super Computers

Characteristics of Supercomputers

- Supercomputers are the computers that are the fastest and they are also very expensive.
- It can calculate up to ten trillion individual calculations per second, this is also the reason which makes it even faster.
- It is used in the stock market or big organizations for managing the online currency world such as Bitcoin etc.

- It is used in scientific research areas for analyzing data obtained from exploring the solar system, satellites, etc.

Mainframe computer

Mainframe computers are designed in such a way that they can support hundreds or thousands of users at the same time. It also supports multiple programs simultaneously. So, they can execute different processes simultaneously. All these features make the mainframe computer ideal for big organizations like banking, telecom sectors, etc., which process a high volume of data in general.

Characteristics of Mainframe Computers

- It is also an expensive or costly computer.
- It has high storage capacity and great performance.
- It can process a huge amount of data (like data involved in the banking sector) very quickly.
- It runs smoothly for a long time and has a long life.

Minicomputer

Minicomputer is a medium size multiprocessing computer. In this type of computer, there are two or more processors, and it supports 4 to 200 users at one time. Minicomputer is similar to Microcontroller. Minicomputers are used in places like institutes or departments for different work like billing, accounting, inventory management, etc. It is smaller than a mainframe computer but larger in comparison to the microcomputer.

Characteristics of Minicomputer

- Its weight is low.
- Because of its low weight, it is easy to carry anywhere.
- less expensive than a mainframe computer.
- It is fast.

Workstation Computer

A workstation computer is designed for technical or scientific applications. It consists of a fast microprocessor, with a large amount of RAM and a high-speed graphic adapter. It is a single-user computer. It is generally used to perform a specific task with great accuracy.

Characteristics of Workstation Computer

- It is expensive or high in cost.
- They are exclusively made for complex work purposes.
- It provides large storage capacity, better graphics, and a more powerful CPU when compared to a PC.
- It is also used to handle animation, data analysis, CAD, audio and video creation, and editing.

Personal Computer (PC)

Personal Computers is also known as a microcomputer. It is basically a general-purpose computer designed for individual use. It consists of a microprocessor as a central processing unit(CPU), memory, input unit, and output unit. This kind of computer is suitable for personal work such as making an assignment, watching a movie, or at the office for office work, etc. For example, Laptops and desktop computers.

Digital Computer

Digital computers are designed in such a way that they can easily perform calculations and logical operations at high speed. It takes raw data as input and processes it with programs stored in its memory to produce the final output. It only understands the binary input 0 and 1, so the raw input data is converted to 0 and 1 by the computer and then it is processed by the computer to produce the result or final output. All modern computers, like laptops, desktops including smartphones are digital computers.

Hybrid Computer

As the name suggests hybrid, which means made by combining two different things. Similarly, the hybrid computer is a combination of both analog and digital computers. Hybrid computers are fast like analog computers and have memory and accuracy like digital computers. So, it has the ability to process both continuous and discrete data. For working when it accepts analog signals as input then it converts them into digital form before processing the input data. So, it is widely used in specialized applications where both analog and digital data are required to be processed. A processor which is used in petrol pumps that converts the measurements of fuel flow into quantity and price is an example of a hybrid computer.

Tablet and Smartphones

Tablets and Smartphones are the types of computers that are pocket friendly and easy to carry is these are handy. This is one of the best use of modern technology. These devices have better hardware capabilities, extensive operating systems, and better multimedia functionality. smartphones and tablets contain a number of sensors and are also able to provide wireless communication protocols.

5. EXPLAIN ABOUT FUNCTIONAL UNITS OF A COMPUTER ?

Computer: A computer is a combination of **hardware and software** resources which integrate together and provides various functionalities to the user. Hardware are the physical components of a computer like the processor, memory devices, monitor, keyboard etc. while software is the set of programs or instructions that are required by the hardware resources to function properly.

There are a few basic components that aids the working-cycle of a computer i.e. the Input- Process- Output Cycle and these are called as the functional components of a computer. It needs certain input, processes that input and produces the desired output. The input unit takes the input, the central processing unit does the processing of data and the output unit produces the output. The memory unit holds the data and instructions during the

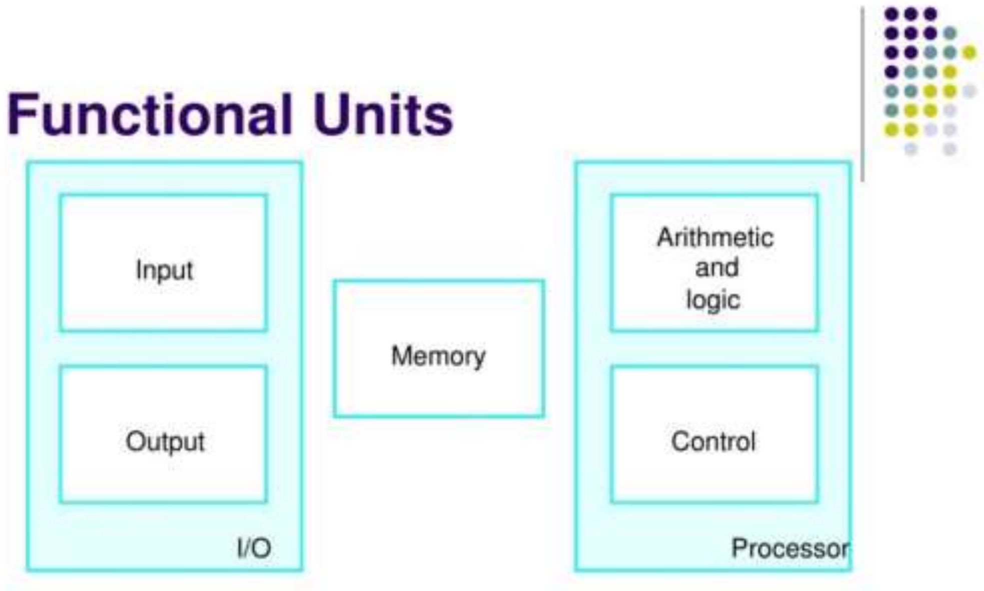


Figure 1.1. Basic functional units of a computer.

processing.

Input Unit : The input unit consists of input devices that are attached to the computer. These devices take input and convert it into binary language that the computer understands. Some of the common input devices are keyboard, mouse, joystick, scanner etc.

- **Central Processing Unit (CPU) :** Once the information is entered into the computer by the input device, the processor processes it. The CPU is called the brain of the computer because it is the control center of the computer. It first fetches instructions from memory and then interprets them so as to know what is to be done. If required, data is fetched from memory or input device. Thereafter CPU executes or performs the required computation and then either stores the output or displays on the output device. The CPU has three main components which are responsible for different functions – Arithmetic Logic Unit (ALU), Control Unit (CU) and Memory registers

- **Arithmetic and Logic Unit (ALU) :** The ALU, as its name suggests performs mathematical calculations and takes logical decisions. Arithmetic calculations include addition, subtraction, multiplication and division. Logical decisions involve comparison of two data items to see which one is larger or smaller or equal.
- **Control Unit :** The Control unit coordinates and controls the data flow in and out of CPU and also controls all the operations of ALU, memory registers and also input/output units. It is also responsible for carrying out all the instructions stored in the program. It decodes the fetched instruction, interprets it and sends control signals to input/output devices until the required operation is done properly by ALU and memory.

  Memory : Memory attached to the CPU is used for storage of data and instructions and is called internal memory The internal memory is divided into many storage locations, each of which can store data or instructions. Each memory location is of the same size and has an address. With the help of the address, the computer can read any memory location easily without having to search the entire memory. when a program is executed, it's data is copied to the internal memory and is stored in the memory till the end of the execution. The internal memory is also called the Primary memory or Main memory. This memory is also called as RAM, i.e. Random Access Memory. The time of access of data is independent of its location in memory, therefore this memory is also called Random Access memory (RAM).
- **Output Unit:** The output unit consists of output devices that are attached with the computer. It converts the binary data coming from CPU to human understandable form. The common output devices are monitor, printer, plotter etc.

6.EXPLAIN ABOUT COMPUTER GENERATIONS?

Generations of Computers

The modern computer took its shape with the arrival of your time. It had been around the 16th century when the evolution of the computer started. The initial computer faced many changes, obviously for the betterment. It continuously improved itself in terms of speed, accuracy, size, and price to urge the form of the fashionable day computer. **Phases of Computer Generations**

This long period is often conveniently divided into the subsequent phases called computer generations.

- First Generation Computers (1940-1956)
- Second Generation Computers (1956-1963)
- Third Generation Computers (1964-1971)
- Fourth Generation Computers (1971-Present)
- Fifth Generation Computers (Present and Beyond)

Generations of Computer	**Time-Period**	**Evolving Hardware**
First Generation	1940s – 1950s	Vacuum Tube Based
Second Generation	1950s – 1960s	Transistor Based
Third Generation	1960s – 1970s	Integrated Circuit Based
Fourth Generation	1970s – Present	Microprocessor Based

Generations of Computer	Time-Period	Evolving Hardware
Fifth Generation	Present – Future	Artificial Intelligence Based

First Generation Computers

The technology behind the primary generation computers was a fragile glass device, which was called a vacuum tube. These computers were very heavy and really large. These weren't very reliable and programming on them was a tedious task as they used low-level programming language and used no OS. First-generation computers were used for calculation, storage, and control purpose. They were too bulky and large that they needed a full room and consume a lot of electricity. Punch cards were used for improving the information for external storage. Magnetic card used. Machine and assembly language is developed.

Examples of some main first-generation computers are mentioned below.

- **ENIAC:** Electronic Numerical Integrator and Computer, built by J. Presper Eckert and John V. Mauchly was a general-purpose computer. It had been cumbersome, and large, and contained 18,000 vacuum tubes.
- **EDVAC:** Electronic Discrete Variable Automatic Computer was designed by von Neumann. It could store data also as instruction and thus the speed was enhanced.
- **UNIVAC:** Universal Automatic Computer was developed in 1952 by Eckert and Mauchly.

Second Generation Computers

Second-generation computers used the technology of transistors rather than bulky vacuum tubes. Another feature was the core storage. A transistor may be a device composed of semiconductor material that amplifies a sign or opens or closes a circuit.

Second Generation Computer

Transistors were invented in Bell Labs. The use of transistors made it possible to perform powerfully and with due speed. It reduced the dimensions and price and thankfully the warmth too, which was generated by vacuum tubes. Central Processing Unit (CPU), memory, programming language, and input, and output units also came into the force within the second generation.

The programming language was shifted from high level to programming language and made programming comparatively a simple task for programmers. Languages used for programming during this era were FORTRAN (1956), ALGOL (1958), and COBOL (1959).

Third Generation Computers

During the third generation, technology envisaged a shift from huge transistors to integrated circuits, also referred to as IC. Here a variety of transistors were placed on silicon chips, called semiconductors. The most feature of this era's computer was speed and reliability. IC was made from silicon and also called silicon chips.

A single IC has many transistors, registers, and capacitors built on one thin slice of silicon. The value size was reduced and memory space and dealing efficiency were increased during this generation. Programming was now wiped-out Higher-level languages like BASIC (Beginners All-purpose

Symbolic Instruction Code).

Fourth Generation Computers

In 1971 First microprocessors were used, the large-scale of integration LSI circuits built on one chip called microprocessors. The advantage of this technology is that one microprocessor can contain all the circuits required to perform arithmetic, logic, and control functions on one chip. LSI placed thousands of transistors onto a single chip.

The computers using microchips were called microcomputers. This generation provided even smaller size of computers, with larger capacities. That’s not enough, then Very Large Scale Integrated (VLSI) circuits replaced LSI circuits. The Intel 4004 chip, developed in 1971, located all the components of the pc from the central processing unit and memory to input/ output controls on one chip and allowed the dimensions to reduce drastically. VLSI placed several hundred thousand transistors on a single silicon chip. This silicon chip is known as the microprocessor.

Technologies like multiprocessing, multiprogramming, time-sharing, operating speed, and virtual memory made it a more user-friendly and customary device. The concept of private computers and computer networks came into being within the fourth generation.

It is often seen in programs like voice recognition, area of medicine, and entertainment. Within the field of game playing also it’s shown remarkable performance where computers are capable of beating human competitors.

Fifth-Generation-Computers

The speed is the highest, size is the smallest and area of use has remarkably increased within the fifth-generation computers. Though not a hundred percent AI has been achieved to date but keeping in sight the present developments, it is often said that this dream also will become a reality very soon.

To summarize the features of varied generations of computers, it is often said that a big improvement has been seen so far because of the speed and accuracy of functioning care, but if we mention the dimensions, it’s been small over the years. The value is additionally diminishing and reliability is increasing.

7.**EXPLAIN ABOUT VON-NEUMON ARCHITECTURE?**

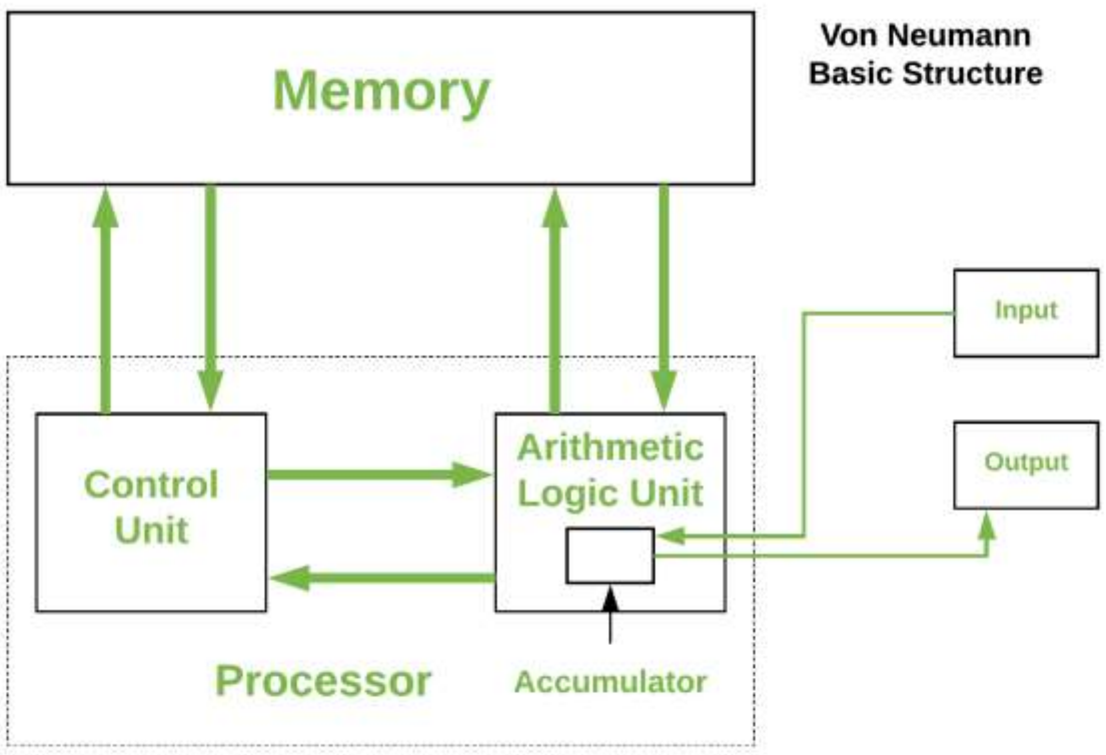
Computer Organization – Von Neumann architecture:

Computer Organization is like understanding the “blueprint” of how a computer works internally. One of the most important models in this field is the Von Neumann architecture, which is the foundation of most modern computers. Named after John von Neumann, this architecture introduced the concept of storing both data and instructions in the same memory.

Historically there have been 2 types of Computers:

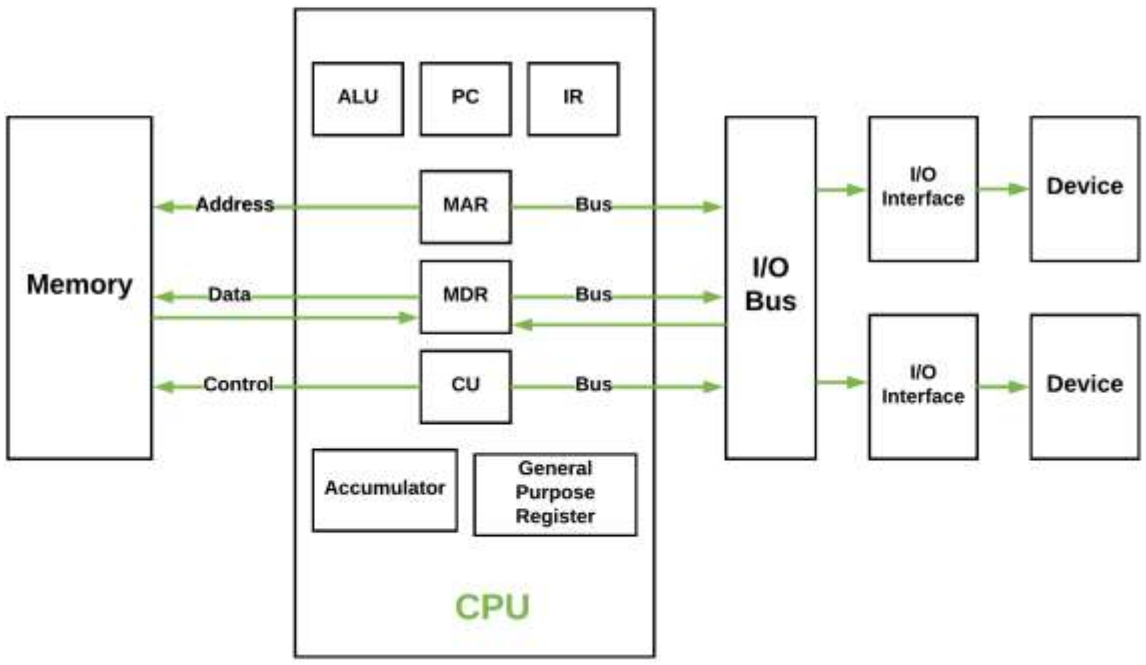
1. Fixed Program Computers – Their function is very specific and they couldn’t be reprogrammed, e.g. Calculators.
2. Stored Program Computers – These can be programmed to carry out many different tasks, applications are stored on them, hence the name.

The Von Neumann architecture popularized the stored-program concept, making computers more flexible and easier to reprogram. This design stores both data and instructions in the same memory, simplifying hardware design and enabling general-purpose computing.



The structure described in the figure outlines the basic components of a computer system, particularly focusing on the memory and processor. Here’s a breakdown of the components:

- Memory: This is where data and instructions are stored. It is a crucial part of the computer system that allows for the storage and retrieval of information.
- Control Unit: This component manages the operations of the computer. It directs the flow of data between the CPU and other components.
- Arithmetic Logic Unit (ALU): The ALU performs arithmetic and logical operations. It is responsible for calculations and decision-making processes.
- Input: This refers to the devices or methods through which data is entered into the computer system.
- Output: This refers to the devices or methods through which data is presented to the user or other systems.
- Processor: The processor, or CPU, is the central component that carries out the instructions of a computer program. It includes the ALU and Control Unit.
- Accumulator: This is a register in the CPU that stores intermediate results of arithmetic and logic operations.



The structure describes **Von Neumann Architecture**, which is a foundational design for modern computers. In this architecture, both data and instructions are stored in the same memory and share a common bus for communication. Here’s an explanation of the components of Von Neumann architecture:

Memory

- **Address**: Specifies the location in memory where data or instructions are stored or retrieved.
- **Data**: The actual information (either data or instructions) stored in memory.
- **Control**: Manages the flow of data and instructions between memory and the CPU.

CPU (Central Processing Unit)

The CPU is the core processing unit that executes instructions. It consists of:

- **ALU (Arithmetic Logic Unit)**: Performs arithmetic and logical operations (e.g., addition, subtraction, comparisons).

- **PC (Program Counter)**: Keeps track of the address of the next instruction to be executed.
- **IR (Instruction Register)**: Holds the current instruction being executed.
- **MAR (Memory Address Register)**: Stores the address of the memory location being accessed.
- **MDR (Memory Data Register)**: Temporarily holds data being transferred to or from memory.
- **CU (Control Unit)**: Coordinates the activities of the CPU, managing the flow of data and instructions.
- **Accumulator**: A register that stores intermediate results of arithmetic and logic operations.
- **General Purpose Registers**: Used for temporary storage of data during processing.

Bus

The bus is a communication system that transfers data, addresses, and control signals between the CPU, memory, and I/O devices. In Von Neumann architecture, a single bus is shared for both data and instructions, which can create a bottleneck (known as the Von Neumann bottleneck).

I/O Bus

- **I/O Interface**: Connects the CPU and memory to input/output devices.
- **Device**: Refers to external hardware like keyboards, monitors, or storage devices.

Key Characteristics of Von Neumann Architecture

1. **Single Memory for Data and Instructions**: Both data and program instructions are stored in the same memory.
2. **Shared Bus**: A single bus is used for transferring data, addresses, and control signals, which can limit performance.

3. **Sequential Execution**: Instructions are executed one at a time in a sequential manner.

Von Neumann bottleneck

Whatever we do to enhance performance, we cannot get away from the fact that instructions can only be done one at a time and can only be carried out sequentially. Both of these factors hold back the competence of the CPU. This is commonly referred to as the 'Von Neumann bottleneck'. We can provide a Von Neumann processor with more cache, more RAM, or faster components but if original gains are to be made in CPU performance, then an influential inspection needs to take place of CPU configuration.

Advantages of Von Neumann Architecture

- **Simplified Design**: Uses a single memory for data and instructions, reducing hardware complexity.
- **Cost-Effective**: Lower production costs due to fewer components.
- **Flexibility**: Can run various programs and makes it suitable for general-purpose computing.
- **Ease of Programming**: Unified memory structure simplifies software development.
- **Widely Adopted**: Forms the foundation of most modern computers hence, ensures widespread compatibility.

Limitations of Von Neumann Architecture

- **Memory Bottleneck**: Shared memory slows down data and instruction transfer.
- **Sequential Processing**: Cannot process data and instructions simultaneously.
- **Scalability Issues**: Struggles with high-performance tasks requiring rapid memory access.

- **Energy Inefficiency**: Frequent memory access increases power consumption.
- **Latency**: Data and instruction fetch delays reduce overall system efficiency.

Applications of Von Neumann Architecture

- **General-Purpose Computing**: Powers desktops, laptops, and smartphones.
- **Embedded Systems**: Used in simple devices where cost and simplicity are priorities.
- **Software Development**: Shapes programming tools and languages due to its unified structure.
- **Education**: A foundational concept in computer science courses.
- **Gaming and Multimedia**: Supports complex applications like video games and editing software.

8.EXPLAIN THE DIFFERENECE BETWEEN MULTI-PROCESSOR AND MULTI-COMPUTER ?

Features	Multiprocessor	Multicomputer
System Architecture	Architecture is based upon multiple processors having shared memory are connected for working together	In this multiple computer having processor and their own memory are connected by a interconnected to share data and have communication.
Memory Access	All the systems share single common memory	All have their own set of memory for storing data and

Features	Multiprocessor	Multicomputer
	and they all performs operations in it.	then the data is shared.
Communication Model	It uses implicit communication model because all the data structures and variables are in common memory.	It uses explicit message sharing model, which involves sending and receiving messages.
Construction	It is easy and lesser expensive since single memory is used.	It is comparatively more expensive and complex as they have multiple memories and all share data.
Network type	Its a dynamic network where data is read and write by all systems during runtime	It’s a static network.
Speed	its execution speed is fast	It is comparatively slower

Features	Multiprocessor	Multicomputer
Scalability	It has limited scalability since single memory is used. A large network will create problems	It has higher scalability and adding more nodes of processors is possible.
Fault Tolerance	If there is fault on memory it will affect all processors.	Faults can be handled individually by each node, it won't spoil the whole network.
Examples	Symmetric Multiprocessing (SMP) systems, where the whole system processes common tasks	Cluster computing, distributed computing, grid computing, cloud computing, parallel servers are
Programming	Easier programming as each utilizing parallel programming.	It requires distributed computing

9.**EXPLAIN BREIFLY ABOUT BUS STRUCTURE ?**

A bus is a collection of electrical pathways or conductors that carry data, addresses, and control signals between different hardware components.

These components are linked to the bus, allowing them to interact and collaborate effortlessly, and these all together are called bus architecture. The width (number of data lines), speed, and protocols of bus architectures can vary. A bus's width refers to the number of parallel data lines it contains, which defines how much data can be sent simultaneously. A wider bus offers faster data transfer but may require more physical connections.

Bus Structure in Computer Architecture

A system bus usually consists of a range of distinct lines, typically numbering from fifty to hundreds. Each line is designated for a specific function, and these lines can be divided into three main functional categories: data lines, address lines, and control lines. Let's discuss each of these in detail.

Data Lines (DL)

- Data Lines (DL) are electrical channels or conductors within a computer's bus architecture that are specifically dedicated to transferring actual data between different computer system components.
- These lines carry binary information in the form of digital signals, such as numbers, instructions, and other data.
- Data lines facilitate parallel data transfer, meaning multiple bits of data can be sent simultaneously.
- The width of the data bus, represented by the number of data lines, determines the quantity of data that can be conveyed in a single operation.
- Each data line can only transmit one bit at a time. For example, a computer system with a 32-bit data bus can transfer 32 bits of data in parallel.

Address Lines (AL)

- An address line is a collection of electrical channels or conductors within a computer's bus architecture specifically designated to carry memory addresses.
- These lines indicate the source or destination of data during memory read and write operations.
- The number of address lines in the address bus impacts the range of memory addresses that the computer system may access.
- The bus module is determined by the higher-order bits, while the address of memory locations or I/O ports is determined by the lower-order bits.
- When the processor needs to read a word from memory, it simply places the relevant word's address on the address line.

Control Lines (CL)

- In bus architecture, control lines are specific lines that transmit control signals between different computer system components.
- These control signals coordinate and govern the flow of data and instructions between various hardware components, ensuring that actions are carried out in the correct order, and the overall system runs smoothly.
- Control lines act as communication channels for signals that control the behaviour of the computer's internal components, such as the central processor unit (CPU), memory modules, input/output devices, and other peripherals.
- These signals are required for memory read and write operations, input/output operations, interrupts, and other control activities.

Some common control signals transmitted through control lines in a computer's bus architecture are as follows:

Control Signal	Description
Memory Write	This command moves the data on the data bus to the addre memory location.
Memory Read	This instruction sends the data from the addressed memor location to the data bus.
I/O Read	Enabling this control line sends data from the addressed I/C to the data bus.
I/O Write	When a command is sent over this control line, data from t data bus is sent to the designated I/O port.
Bus Request	The activation of this control line signifies that the compon has signalled its desire to take control of the bus.
Bus Grant	The activation of this control line signifies that the bus has allocated to the component that made the request.
Transfer ACK	This control line indicates that data has been received or pl on the data bus.
Interrupt Request	This control line indicates that there are pending interrupts
Interrupt ACK	When the pending interrupt is serviced, this control line acknowledges it.
Reset	This control line's bit information initializes all modules.

Single Bus Structure

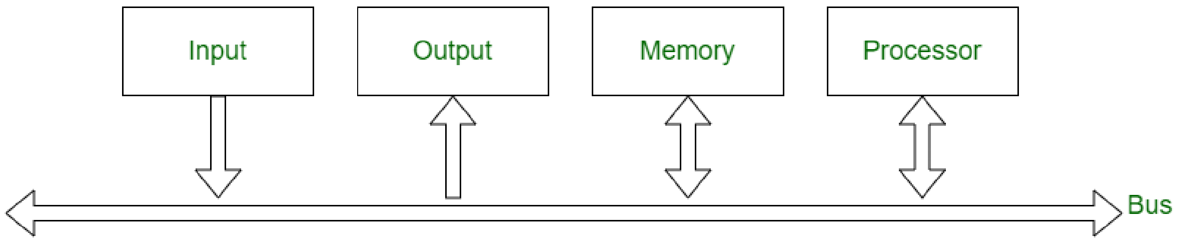
In a single bus structure, one common bus is used to communicate between peripherals and microprocessors. It has disadvantages due to the use of one common bus.

Advantages of Single Bus Structure

- **Simplicity:** The design is simplistic in nature and hence is easy to roll out and even administer.
- **Cost-Effective:** It is more expensive to have a large number of buses and wiring to power the smart grids, so fewer buses and less wiring leads to least expensive system.
- **Ease of Maintenance:** Since there are fewer components that are generally involved, it is easier to diagnose and even rectify any problems that may exist.

Disadvantages of Single Bus Structure

- **Bandwidth Limitation:** Because all the components feed off this bus, the rate at which data is transferred is rather slow and therefore creates a bottleneck.
- **Slower Performance:** This is because when many components in the computer request access to the central processing unit or the RAM at the same time then the system slows down.
- **Scalability Issues:** More components can intensify the bandwidth problem causing a problem if expansion of the system becomes necessary.



Single Bus Structure

Double Bus Structure

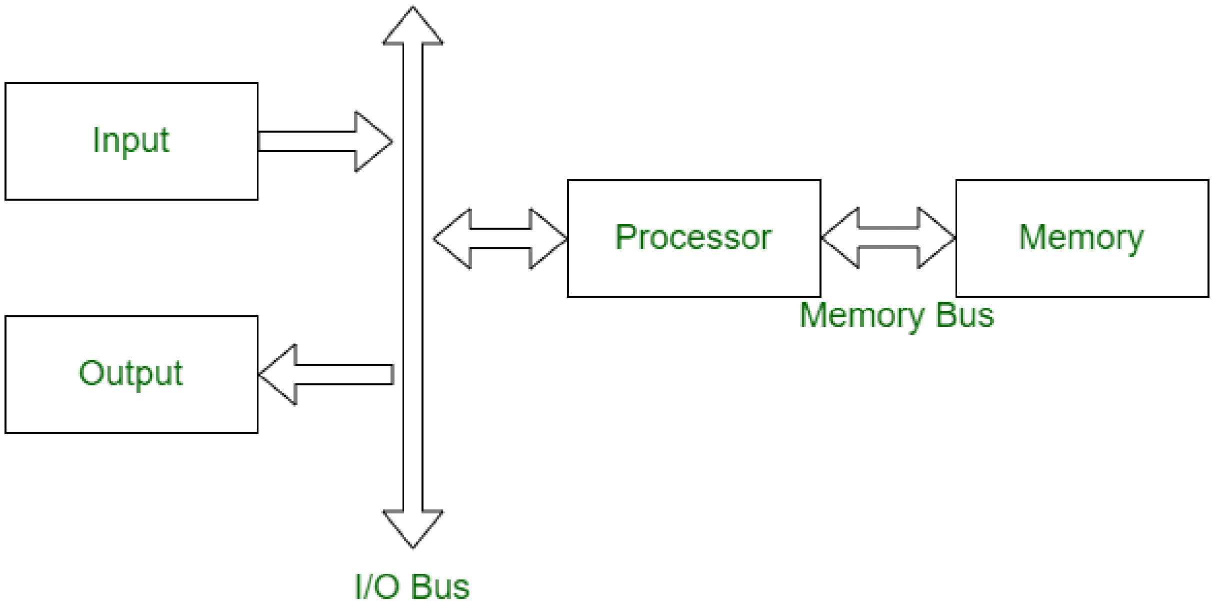
In a double bus structure, one bus is used to fetch instructions while other is used to fetch data, required for execution. It is to overcome the bottleneck of a single bus structure.

Advantages of Double Bus Structure

- **Improved Performance:** The efficiency is improved because the system can manage more data at once since it consists of two buses which are one for memory and the other for I/O.
- **Better Bandwidth Utilization:** Every bus can work on its own which will help to decrease the traffic and increase the speed of data exchange.
- **Scalability:** The system can be expanded more easily, as the two buses can accommodate more components without significant performance degradation.

Disadvantages of Double Bus Structure

- **Increased Complexity:** The design is slightly cumbersome requiring several parts which have to be fitted properly.
- **Higher Cost:** The major drawbacks of more buses and wiring are that it brings about an overall increase in the cost of the system.
- **Challenging Maintenance:** Identifying problems becomes even more challenging mainly because of the numerous parts and contact points.



Double Bus Structure

10.**EXPLAIN ABOUT THE BASIC OPERATIONAL CONCEPTS OF A COMPUTER ?**

<u>Basic Operational Concepts:</u>

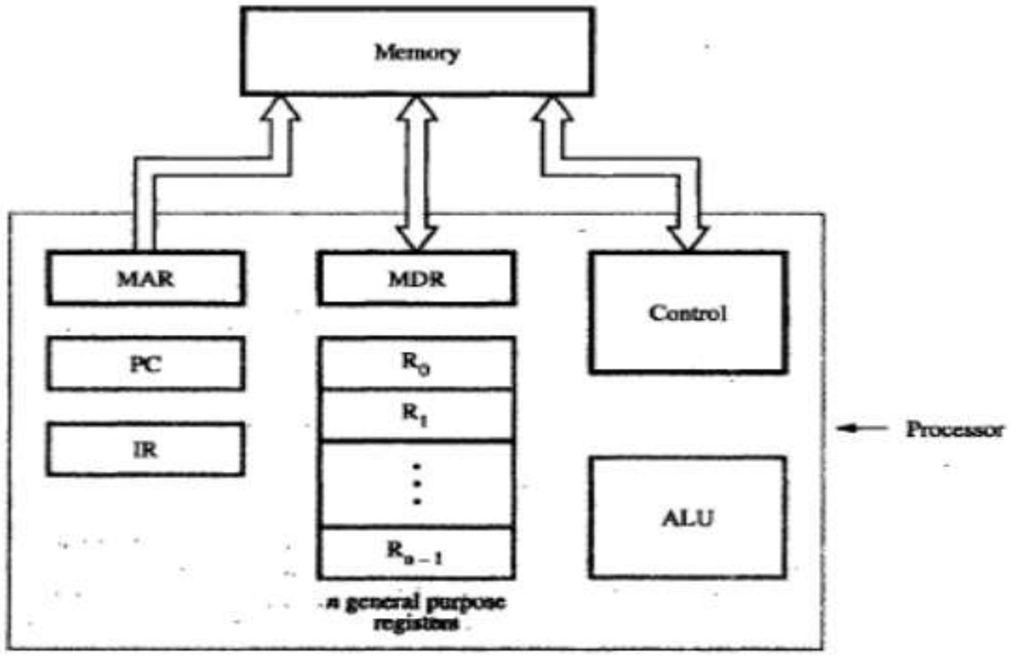


Figure 1.2 Connections between the processor and the memory.

Connection B / W Processor & Memory

Connection B/W Processor & Memory

- The above-mentioned block diagram consists of the following components

1) Memory
2) MAR
3) MDR
4) PC
5) IR
6) General Purpose Registers
7) Control Unit
8) ALU

MDR:

The Memory Data Register (MDR) is a crucial component of a computer's Central Processing Unit (CPU) that temporarily holds data being transferred to or from memory, facilitating efficient data processing. It acts as a buffer between the CPU and main memory, allowing for smoother execution of instructions by storing the operands during read and write cycles. Understanding the role of the MDR in the fetch-execute cycle enhances your knowledge of computer architecture and data handling efficiency.

Memory Address Register (MAR) :
It contains the address of a location of main memory from where information has to be fetched for information has to be stored. Contents of MAR are directly connected to the address bus. Apart from these registers, we may use other registers which may be invisible to the user, e.g., temporary Buffering registers.

Program Counter (PC) :
It contains the address of an instruction to be executed next. The PC is updated by the CPU after each instruction is executed so that it always points to the next instruction to be executed. A branch or skip instruction will also modify the content of the PC.

Instruction Register (IR) :

It contains the instruction most recently fetched or executed. The fetched instruction is loaded into an IR, where the opcode and operand specifier is analysed.

Arithmetic Logic Unit (ALU) is the component responsible for carrying out arithmetic operations (addition, subtraction, multiplication, and division) and logical operations (AND, OR, NOT) on data. It performs calculations and generates results that are integral to various processes.

The **Control Unit (CU)** serves as the manager of the CPU's activities. It interprets and coordinates instructions from the computer's memory, ensuring that each operation is executed in the correct sequence. The CU manages the flow of data between different components of the CPU and other hardware components, ensuring that the computer operates according to the program's instructions.

Computer **Memory** is just like the human brain. It is used to store data/information and instructions. It is a data storage unit or a data storage device where data is to be processed and instructions required for processing are stored. It can store both the input and output can be stored here.

Characteristics of Computer Memory

- It is faster computer memory as compared to secondary memory.
- It is semiconductor memories.
- It is usually a volatile memory, and main memory of the computer.
- A computer system cannot run without primary memory.

Types of Computer Memory

In general, computer memory is of three types:

- Primary memory
- Secondary memory
- Cache memory

Now we discuss each type of memory one by one in detail:

1. Primary Memory

It is also known as the main memory of the computer system. It is used to store data and programs or instructions during computer operations. It uses semiconductor technology and hence is commonly called semiconductor memory. Primary memory is of two types:

- **RAM (Random Access Memory):** It is a volatile memory. Volatile memory stores information based on the power supply. If the power supply fails/ interrupted/stopped, all the data and information on this memory will be lost. RAM is used for booting up or start the computer. It temporarily stores programs/data which has to be executed by the processor. RAM is of two types:
    - **S RAM (Static RAM):** S RAM uses transistors and the circuits of this memory are capable of retaining their state as long as the power is applied. This memory consists of the number of flip flops with each flip flop storing 1 bit. It has less access time and hence, it is faster.
    - **D RAM (Dynamic RAM):** D RAM uses capacitors and transistors and stores the data as a charge on the capacitors. They contain thousands of memory cells. It needs refreshing of charge on capacitor after a few milliseconds. This memory is slower than S RAM.
- **ROM (Read Only Memory):** It is a non-volatile memory. Non-volatile memory stores information even when there is a power supply failed/ interrupted/stopped. ROM is used to store information that is used to operate the system. As its name refers to read-only memory, we can only read the programs and data that is stored on it. It contains some electronic fuses that can be programmed for a piece of specific information. The information stored in the ROM in binary format. It is also known as permanent memory.

Secondary Memory

It is also known as auxiliary memory and backup memory. It is a non-volatile memory and used to store a large amount of data or information.

The data or information stored in secondary memory is permanent, and it is slower than primary memory. A CPU cannot access secondary memory directly. The data/information from the auxiliary memory is first transferred to the main memory, and then the CPU can access it.

Characteristics of Secondary Memory

- It is a slow memory but reusable.
- It is a reliable and non-volatile memory.
- It is cheaper than primary memory.
- The storage capacity of secondary memory is large.
- A computer system can run without secondary memory.
- In secondary memory, data is stored permanently even when the power is off.

Operating steps:

- The primary function of a computer system is to execute a program, sequence of instructions. These instructions are stored in computer memory.
- These instructions are executed to process data which are already loaded in the computer memory through some input devices.
- After processing the data, the result is either stored in the memory for further reference, or it is sent to the outside world through some output port.
- To perform the execution of an instruction, in addition to the arithmetic logic unit, and control unit, the processor contains a number of registers used for temporary storage of data and some special function registers.
- The special function registers include program counters (PC), instruction registers (IR), memory address registers (MAR) and memory and memory data registers (MDR).
- The Program counter is one of the most critical registers in CPU.

- The Program counter monitors the execution of instructions. It keeps track on which instruction is being executed and what the next instruction will be.
- The instruction register IR is used to hold the instruction that is currently being executed.
- The contents of IR are available to the control unit, which generate the timing signals that control, the various processing elements involved in executing the instruction.
- The two registers MAR and MDR are used to handle the data transfer between the main memory and the processor.
- The MAR holds the address of the main memory to or from which data is to be transferred.
- The MDR contains the data to be written into or read from the addressed word of the main memory.
- Whenever the processor is asked to communicate with devices, we say that the processor is servicing the devices. The processor can service these devices in one of the two ways.
- One way is to use the polling routine, and the other way is to use an interrupt.
- Polling enables the processor software to check each of the input and output devices frequently. During this check, the processor tests to see if any devices need servicing or not.
- Interrupt method provides an external asynchronous input that informs the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device.

  Example:

  Add LOCA, R0

- This instruction adds the operand at memory location LOCA to the operand which will be present in the Register R0.
- The above mentioned example can be written as follows:

  Load LOCA, R1

  Add R1, R0
- First instruction sends the contents of the memory location LOCA into processor Register R0, and meanwhile the second instruction adds the contents of Register R1 and R0 and places the output in the Register R1.

  The memory and the processor are are swapped and are started by sending the address of the memory location to be accessed to the memory unit and issuing the appropriate control signals.
- The data is then transferred to or from the memory.

<u>2 MARKS</u>

1.what is flip flop and types of flip of flip?

A flip-flop is a digital electronic circuit used for storing binary data, typically a single bit of information (0 or 1). It has two stable states, which makes it useful in memory storage and timing applications. Flip-flops are essential components in sequential logic circuits, such as registers, counters, and memory units.

Types of Flip-Flops:

1. SR Flip-Flop (Set-Reset Flip-Flop):

- It has two inputs, Set (S) and Reset (R), and two outputs, Q and Q' (complement of Q). It stores a bit based on the Set and Reset inputs.

2. D Flip-Flop (Data Flip-Flop):

- It has a single data input (D) and a clock input (CLK). The value of D is transferred to the output Q on the triggering edge of the clock.

3. T Flip-Flop (Toggle Flip-Flop):

- It has a single input, T (Toggle), and a clock input. When T is high (1) and the clock signal triggers, the state toggles between 0 and 1.

4. JK Flip-Flop:

- It has two inputs, J and K, and a clock input. The JK flip-flop can operate as an SR flip-flop, D flip-flop, or T flip-flop based on the input conditions.

2.**what is meant by binary counter?**

A **binary counter** is a digital circuit that counts in binary numbers (i.e., numbers represented using 0s and 1s). It is used to keep track of the number of occurrences of an event or to generate a sequence of binary numbers. Binary counters are commonly built using flip-flops and are used in applications like timers, clocks, and digital clocks.

4. How many types of computers are there?

1.super computer

2.digital computer

3.mini computer

4.main frame computer

5.note book

6.workstation

7.analog computer

5.**Defnition of sequential circuit?**

A **sequential circuit** is a type of digital circuit in which the output depends not only on the current inputs but also on the past sequence of inputs, meaning it has **memory**. In other words, a sequential circuit's behavior is determined by both its current state and the inputs it receives, making it different from a combinational circuit, where the output depends solely on the current inputs.

Key Characteristics:

- **Memory**: Sequential circuits store information about previous states and use that information to determine future outputs.
- **Clocking**: Many sequential circuits are synchronized with a clock signal, which controls when state changes occur. These are known as **synchronous sequential circuits**. If they don't rely on a clock, they are called **asynchronous sequential circuits**.
- **State**: The output of a sequential circuit depends on the combination of the present input and the sequence of past inputs, which is called its **state**.

Example:

- **Flip-flops** and **registers** are examples of sequential circuits since they store information and change states based on inputs over time.

6.fundamental components of a computer?

The fundamental components of a computer can be divided into several key parts, each responsible for a specific function. These components work together to perform tasks, process data, and run programs. Here are the primary components:

1. <u>Central Processing Unit (CPU)</u>

- Function: The CPU is the brain of the computer, responsible for executing instructions and performing calculations. It processes data, controls other components, and carries out the logic operations.
- Components:

    - Arithmetic Logic Unit (ALU): Performs mathematical and logical operations.
    - Control Unit (CU): Directs the operation of the processor, controlling the flow of data and instructions.
    - Registers: Small, fast storage areas within the CPU that temporarily hold data or instructions.

2. <u>Memory (RAM and ROM)</u>

- Function: Memory stores data and instructions that the CPU needs to access quickly.
- Types:
    - RAM (Random Access Memory): Temporary, volatile memory used by the CPU to store data that is currently being processed or used.
    - ROM (Read-Only Memory): Non-volatile memory that stores firmware or software that doesn’t change, such as the computer's boot-up instructions.

3. <u>Input Devices</u>

- Function: Devices used to send data and instructions into the computer.
- Examples: Keyboard, mouse, microphone, scanner, touchpad.

4. <u>Output Devices</u>

- Function: Devices that allow the computer to present or display processed data.
- Examples: Monitor, printer, speakers, and projector.

5. <u>Storage Devices</u>

- Function: Devices used to store data permanently (or semi-permanently) for later use.
- Examples: Hard Drive (HDD), Solid-State Drive (SSD), optical drives (CD/DVD), flash drives.

6. <u>Motherboard</u>

- Function: The main circuit board that holds the CPU, memory, and other essential components together. It also provides connections for input/output devices and other peripheral hardware.
- Components: Connectors for power, storage, expansion cards, and various communication interfaces.

7. Power Supply Unit (PSU)

- Function: Provides the electrical power needed for the computer to operate. It converts AC (alternating current) from an outlet into the DC (direct current) used by the computer's components.

8. Bus (Data Bus, Address Bus, Control Bus)

- Function: The bus is a communication pathway that allows different components of the computer to transfer data and control signals.
- Types:
    - Data Bus: Carries the data between components.
    - Address Bus: Carries the address location in memory or I/O for data retrieval.
    - Control Bus: Carries control signals that manage the operations of other components.

7.why note book computers are very much famous?

Notebook computers, also known as laptops, have become very popular due to a combination of factors that make them highly convenient, portable, and versatile. Here are the key reasons why notebook computers are so famous:

1. Portability

2.Convenience

3.Improved Performance

4.Versatility

5.Wireless Connectivity

6.Affordability

7.Battery Life

8.Touchscreen and 2-in-1 Options

8.elaborate ASCII and EBCDIC?

ASCII (American Standard Code for Information Interchange)

EBCDIC (Extended Binary Coded Decimal Interchange Code)

9.what are the types of units present in computer?

Here are the main types of units present in a computer:

1. Input Unit

- Function: The input unit is responsible for receiving data and instructions from the external environment and sending them to the computer system for processing.
- Examples:
    - Keyboard
    - Mouse
    - Scanner
    - Microphone
    - Camera

2. Output Unit

- Function: The output unit is responsible for converting the processed data into a human-readable form or a format that can be understood by other devices.
- Examples:
    - Monitor
    - Printer

    - Speakers
    - Projector

3. Central Processing Unit (CPU)

- Function: The CPU is often referred to as the "brain" of the computer. It performs most of the processing inside the computer. It executes instructions from programs and performs calculations.
- Components:
    - Arithmetic Logic Unit (ALU): Handles arithmetic and logical operations.
    - Control Unit (CU): Directs the operation of the processor, telling it when and how to execute instructions.
    - Registers: Small, fast storage areas used for temporary data storage within the CPU.

4. Memory Units

- Function: Memory units store data and instructions that the CPU needs to access. Memory is generally categorized into two main types:
    - Primary Memory (Volatile): Temporary memory that is directly accessible by the CPU. This includes:
        - RAM (Random Access Memory): Stores data and programs currently in use.
        - Cache Memory: Small, fast memory located inside the CPU to store frequently accessed data.
    - Secondary Memory (Non-Volatile): Permanent storage that retains data even when the computer is turned off. This includes:
        - Hard Drive (HDD)
        - Solid-State Drive (SSD)
        - Optical Disks (CD/DVD)
        - USB Flash Drives

5. Control Unit

- Function: The control unit (CU) is part of the CPU and is responsible for directing the flow of data between the CPU, memory, and other peripherals. It decodes and executes instructions and controls the timing of operations.

6. Arithmetic Logic Unit (ALU)

- Function: The ALU performs mathematical operations (like addition, subtraction, multiplication, and division) and logical operations (like AND, OR, NOT) on the data. It is a critical component of the CPU.

10.what happened in 4th generation of a computer?

The 4th generation of computers (roughly spanning from the 1970s to the 1990s) brought significant advancements in computing technology, largely driven by the development of microprocessors. Here's an overview of what happened during this era:

1. Introduction of Microprocessors

- The microprocessor was the hallmark of the 4th generation. A microprocessor is a single chip that contains the entire central processing unit (CPU) of a computer.
- The first commercially available microprocessor was the Intel 4004, released in 1971. It could perform basic calculations and control operations, making it the foundation for personal computers and other embedded systems.

2. Miniaturization of Hardware

- With the advent of microprocessors, computers became significantly smaller, faster, and more affordable than previous generations.
- The size of computers shrank from room-sized machines to desktop-sized systems, making them more accessible to businesses and eventually consumers.

3. Development of Personal Computers (PCs)

- The 4th generation saw the emergence of personal computers (PCs), which were affordable and could be used by individuals or small businesses. Notable examples include:
    - Apple II (1977): One of the first successful personal computers.
    - IBM PC (1981): Set the standard for the PC market, leading to the widespread adoption of personal computers.
    - Commodore 64 (1982): A home computer that became incredibly popular for personal and educational use.

4. Operating Systems and Software
5. Improved Graphics and User Interfaces
6. Increased Storage Capacity
7. Introduction of Networking and the Internet
8. Increased Processing Power.

UNIT – 3 :- COMPUTER ARITHMETIC

PART-A (10 MARKS)

1. Perform Binary addition and subtraction of signed numbers?
PO2

Sol:- To perform binary addition and subtraction with signed numbers using two's complement, convert the numbers to two's complement, add them as if they were unsigned, and then convert the result back to signed representation if needed.

1. Two's Complement Representation:

- **Positive Numbers:** Represent a positive number as its binary equivalent with a leading '0'.
- **Negative Numbers:**
    - Find the binary equivalent of the absolute value of the negative number.
    - Invert all the bits (0s become 1s and 1s become 0s).
    - Add 1 to the inverted number.

2. Addition:

- Add the two two's complement representations as if they were unsigned binary numbers.
- **Carry-out:** Discard any carry-out from the most significant bit (MSB).
- **Result Interpretation:**
    - If the MSB of the result is '0', the result is positive.
    - If the MSB of the result is '1', the result is negative (and is in two's complement form).

3. Subtraction:

- Convert the subtrahend (the number being subtracted) to its two's complement representation.
- Add the minuend (the number being subtracted from) to the two's complement of the subtrahend.
- Follow the same addition rules as described above.

Example:

Let's perform the following subtraction: 5 - 3.

1. **Convert to Binary:**
   - 5 (decimal) = 00000101 (binary)
   - 3 (decimal) = 00000011 (binary)
2. **Two's Complement of -3:**
   - Invert the bits of 00000011: 11111100
   - Add 1: 11111100 + 1 = 11111101
3. **Addition:**
   - 5 (00000101) + (-3 (11111101)) = 00000101 + 11111101 = 00000010
4. **Result Interpretation:**
   - The result (00000010) is positive (MSB is 0) and represents 2 in decimal.

2. Explain about multi bus organization?
   P02

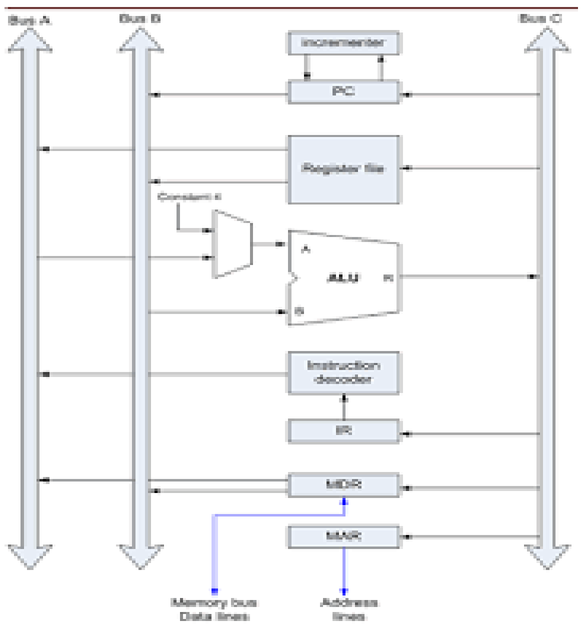
Sol:- In a multi-bus organization, multiple dedicated data paths (buses) are used for instruction fetching, operand retrieval, and result storage, allowing for concurrent operations and improved performance compared to a single-bus system.

Here's a more detailed explanation:

- **Purpose:**

Multi-bus organization aims to overcome the limitations of single-bus systems where a single bus serves all communication needs, leading to bottlenecks and reduced performance.

- **How it works:**
    - **Multiple Buses:** Instead of relying on a single bus, multiple buses are used, each dedicated to a specific purpose, such as fetching instructions, operands, or storing results.
    - **Concurrent Operations:** This allows the CPU to fetch instructions, retrieve operands from memory, and store results concurrently, leading to faster execution cycles.
    - **Example:** A three-bus system might use one bus for fetching instructions, another for retrieving operands, and a third for storing results.
- **Benefits:**
    - **Improved Performance:** By allowing concurrent operations, multi-bus systems can significantly improve the overall performance of the processor.
    - **Reduced Bus Contention:** With dedicated buses, the likelihood of bus contention (multiple devices trying to access the bus simultaneously) is reduced, leading to smoother data transfers.
    - **Scalability:** Multi-bus architectures are more scalable than single-bus architectures, as additional buses can be added to accommodate more devices and higher data transfer rates.
- **Types of Multi-Bus Organizations:**
    - **Two-Bus Organization:** Uses two buses, one for fetching operands and the other for storing results.
    - **Three-Bus Organization:** Uses three buses, one for fetching instructions, one for fetching operands, and one for storing results.



3. Explain about types of Computer and functional unit?
PO2

Sol:-

A computer is an electronic device that has storage, computations, input (data), output (data) and networking capabilities. With the growing AI, computers also have learning capabilities from the data provided. The input and output data can be in different forms like text, images, audio and video. A computer processes the input according to the set of instructions provided to it by the user and gives the desired output. Computers are of various types and they can be categorized in two ways on the basis of size and on the basis of data handling capabilities.

Types of Computer

There are two bases on which we can define the types of computers. We will discuss the type of computers on the basis of size and data handling capabilities. We will discuss each type of computer in detail. Let's see first what are the types of computers.

- Super Computer
- Mainframe computer
- Mini Computer

- Workstation Computer
- Personal Computer (PC)

Now, we are going to discuss each of them in detail.

Supercomputer

When we talk about speed, then the first name that comes to mind when thinking of computers is supercomputers. They are the biggest and fastest computers (in terms of speed of processing data). Supercomputers are designed such that they can process a huge amount of data, like processing trillions of instructions or data just in a second. This is because of the thousands of interconnected processors in supercomputers. It is basically used in scientific and engineering applications such as weather forecasting, scientific simulations, and nuclear energy research. It was first developed by Roger Cray in 1976.

Super Computers

Characteristics of Supercomputers

- Supercomputers are the computers that are the fastest and they are also very expensive.

- It can calculate up to ten trillion individual calculations per second, this is also the reason which makes it even faster.
- It is used in the stock market or big organizations for managing the online currency world such as Bitcoin etc.
- It is used in scientific research areas for analyzing data obtained from exploring the solar system, satellites, etc.

Mainframe computer

Mainframe computers are designed in such a way that they can support hundreds or thousands of users at the same time. It also supports multiple programs simultaneously. So, they can execute different processes simultaneously. All these features make the mainframe computer ideal for big organizations like banking, telecom sectors, etc., which process a high volume of data in general.

Characteristics of Mainframe Computers

- It is also an expensive or costly computer.
- It has high storage capacity and great performance.
- It can process a huge amount of data (like data involved in the banking sector) very quickly.
- It runs smoothly for a long time and has a long life.

Minicomputer

Minicomputer is a medium size multiprocessing computer. In this type of computer, there are two or more processors, and it supports 4 to 200 users at one time. Minicomputer is similar to Microcontroller. Minicomputers are used in places like institutes or departments for different work like billing, accounting, inventory management, etc. It is smaller than a mainframe computer but larger in comparison to the microcomputer.

Characteristics of Minicomputer

- Its weight is low.
- Because of its low weight, it is easy to carry anywhere.

- less expensive than a mainframe computer.
- It is fast.

Workstation Computer

A workstation computer is designed for technical or scientific applications. It consists of a fast microprocessor, with a large amount of RAM and a high-speed graphic adapter. It is a single-user computer. It is generally used to perform a specific task with great accuracy.

Characteristics of Workstation Computer

- It is expensive or high in cost.
- They are exclusively made for complex work purposes.
- It provides large storage capacity, better graphics, and a more powerful CPU when compared to a PC.
- It is also used to handle animation, data analysis, CAD, audio and video creation, and editing.

Personal Computer (PC)

Personal Computers is also known as a microcomputer. It is basically a general-purpose computer designed for individual use. It consists of a microprocessor as a central processing unit(CPU), memory, input unit, and output unit. This kind of computer is suitable for personal work such as making an assignment, watching a movie, or at the office for office work, etc. For example, Laptops and desktop computers.

Characteristics of Personal Computer (PC)

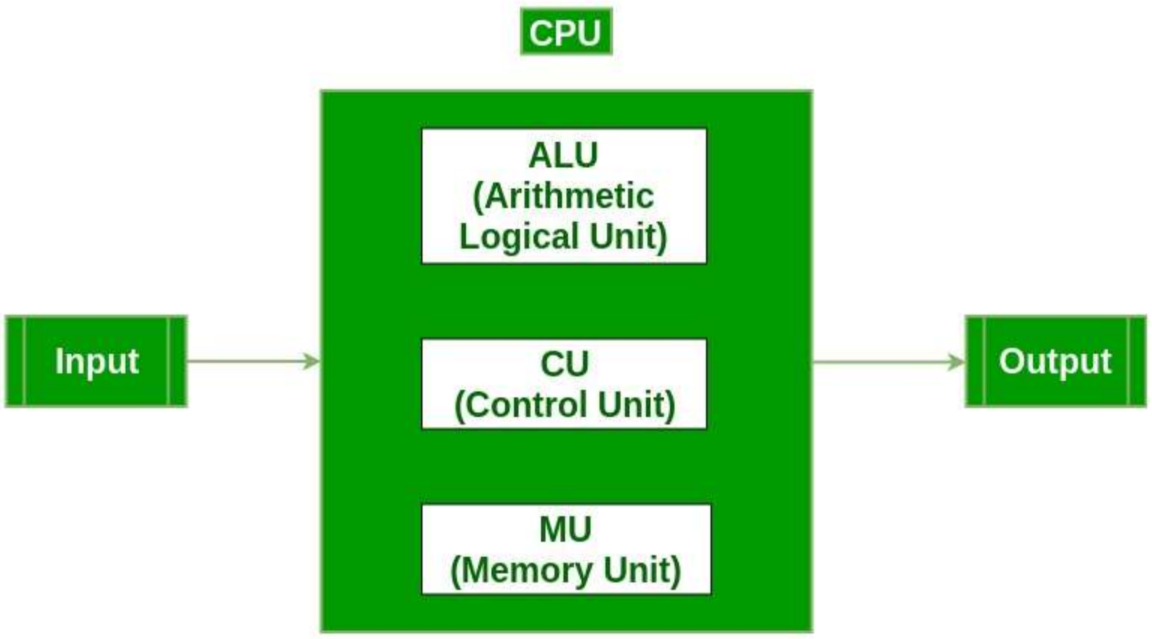
- In this limited number of software can be used.
- It is the smallest in size.
- It is designed for personal use.
- It is easy to use.

Functional Components of a Computer:-

Computer: A computer is a combination of **hardware and software** resources which integrate together and provides various functionalities to the user. Hardware are the physical components of a computer like the processor, memory devices, monitor, keyboard etc. while software is the set of programs or instructions that are required by the hardware resources to function properly.
There are a few basic components that aids the working-cycle of a computer i.e. the Input- Process- Output Cycle and these are called as the functional components of a computer. It needs certain input, processes that input and produces the desired output. The input unit takes the input, the central processing unit does the processing of data and the output unit produces the output. The memory unit holds the data and instructions during the processing.

Digital Computer: A digital computer can be defined as a programmable machine which reads the binary data passed as instructions, processes this binary data, and displays a calculated digital output. Therefore, Digital computers are those that work on the digital data.

Details of Functional Components of a Digital Computer



- **Input Unit :**The input unit consists of input devices that are attached to the computer. These devices take input and convert it into binary language that the computer understands. Some of the common input devices are keyboard, mouse, joystick, scanner etc.
- **Central Processing Unit (CPU) :** Once the information is entered into the computer by the input device, the processor processes it. The CPU is called the brain of the computer because it is the control center of the computer. It first fetches instructions from memory and then interprets them so as to know what is to be done. If required, data is fetched from memory or input device. Thereafter CPU executes or performs the required computation and then either stores the output or displays on the output device. The CPU has three main components which are responsible for different functions – Arithmetic Logic Unit (ALU), Control Unit (CU) and Memory registers
- **Arithmetic and Logic Unit (ALU) :** The ALU, as its name suggests performs mathematical calculations and takes logical decisions. Arithmetic calculations include addition, subtraction, multiplication and division. Logical decisions involve comparison of two data items to see which one is larger or smaller or equal.
- **Control Unit :** The Control unit coordinates and controls the data flow in and out of CPU and also controls all the operations of ALU, memory registers and also input/output units. It is also responsible for carrying out all the instructions stored in the program. It decodes the fetched instruction, interprets it and sends control signals to input/output devices until the required operation is done properly by ALU and memory.
- **Memory Registers :** A register is a temporary unit of memory in the CPU. These are used to store the data which is directly used by the processor. Registers can be of different sizes(16 bit, 32 bit, 64 bit and so on) and each register inside the CPU has a specific function like storing data, storing an instruction, storing address of a location in memory etc. The user registers can be used by an assembly language programmer for storing operands,

intermediate results etc. Accumulator (ACC) is the main register in the ALU and contains one of the operands of an operation to be performed in the ALU.

- **Memory :** Memory attached to the CPU is used for storage of data and instructions and is called internal memory The internal memory is divided into many storage locations, each of which can store data or instructions. Each memory location is of the same size and has an address. With the help of the address, the computer can read any memory location easily without having to search the entire memory. when a program is executed, it's data is copied to the internal memory and is stored in the memory till the end of the execution. The internal memory is also called the Primary memory or Main memory. This memory is also called as RAM, i.e. Random Access Memory. The time of access of data is independent of its location in memory, therefore this memory is also called Random Access memory (RAM). Read this for different types of RAMs
- **Output Unit :** The output unit consists of output devices that are attached with the computer. It converts the binary data coming from CPU to human understandable form. The common output devices are monitor, printer, plotter etc.

4. Explain about hard wired and multi programmed control?
PO2

Sol:- In computer architecture, the control unit is responsible for directing the flow of data and instructions within the CPU. There are two main approaches to implementing a control unit: hardwired and micro-programmed.

A hardwired control unit is a control unit that uses a fixed set of logic gates and circuits to execute instructions. The control signals for each instruction are hardwired into the control unit, so the control unit has a

dedicated circuit for each possible instruction. Hardwired control units are simple and fast, but they can be inflexible and difficult to modify.

On the other hand, a micro-programmed control unit is a control unit that uses a microcode to execute instructions. The microcode is a set of instructions that can be modified or updated, allowing for greater flexibility and ease of modification. The control signals for each instruction are generated by a microprogram that is stored in memory, rather than being hardwired into the control unit.

The control unit is the brain of the CPU, and it can be implemented in two ways: hardwired or micro-programmed. Understanding the differences between these implementations is essential for those studying computer organization.

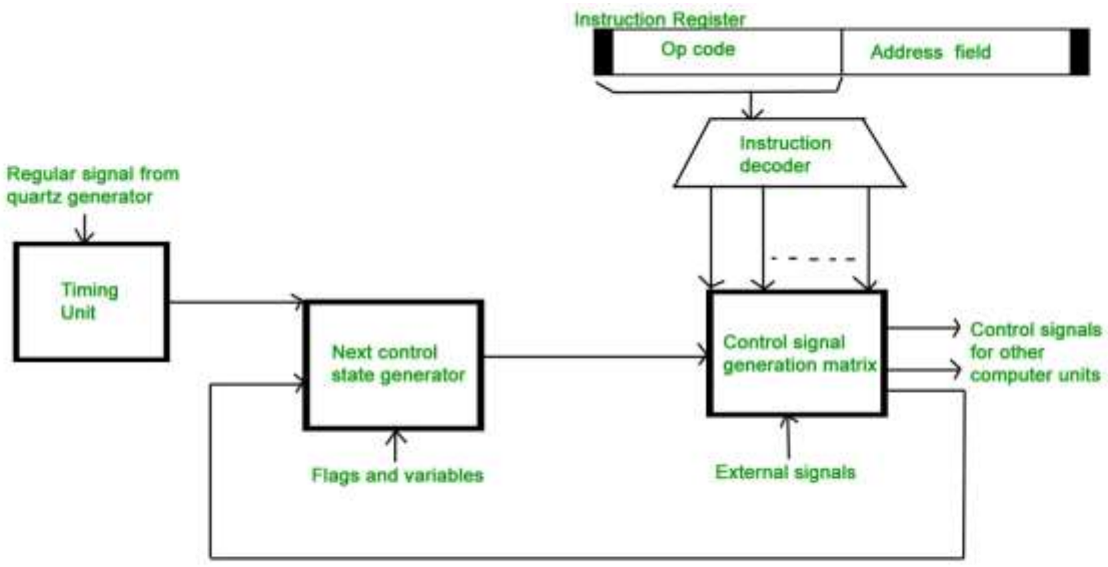
If we talk about Micro-programmed control units they are generally slower than hardwired control units because they require an extra step of decoding the microcode to generate control signals, but they are more flexible and easier to modify. They are commonly used in modern CPUs because they allow for easier implementation of complex instruction sets and better support for instruction set extensions.

To execute an instruction, the control unit of the CPU must generate the required control signal in the proper sequence. There are two approaches used for generating the control signals in proper sequence as Hardwired Control unit and the Micro-programmed control unit.

Hardwired Control Unit: The control hardware can be viewed as a state machine that changes from one state to another in every clock cycle, depending on the contents of the instruction register, the condition codes, and the external inputs. The outputs of the state machine are the control signals. The sequence of the operation carried out by this machine is determined by the wiring of the logic elements and hence named "hardwired".

- **Fixed logic circuits that correspond directly to the Boolean expressions are used to generate the control signals.**

- **Hardwired control is faster than micro-programmed control.**
- **A controller that uses this approach can operate at high speed.**
- **RISC architecture is based on the hardwired control unit**


Instruction Register
Op code
Address field
Instruction decoder
Regular signal from quartz generator
Timing Unit
Next control state generator
Control signal generation matrix
Control signals for other computer units
Flags and variables
External signals

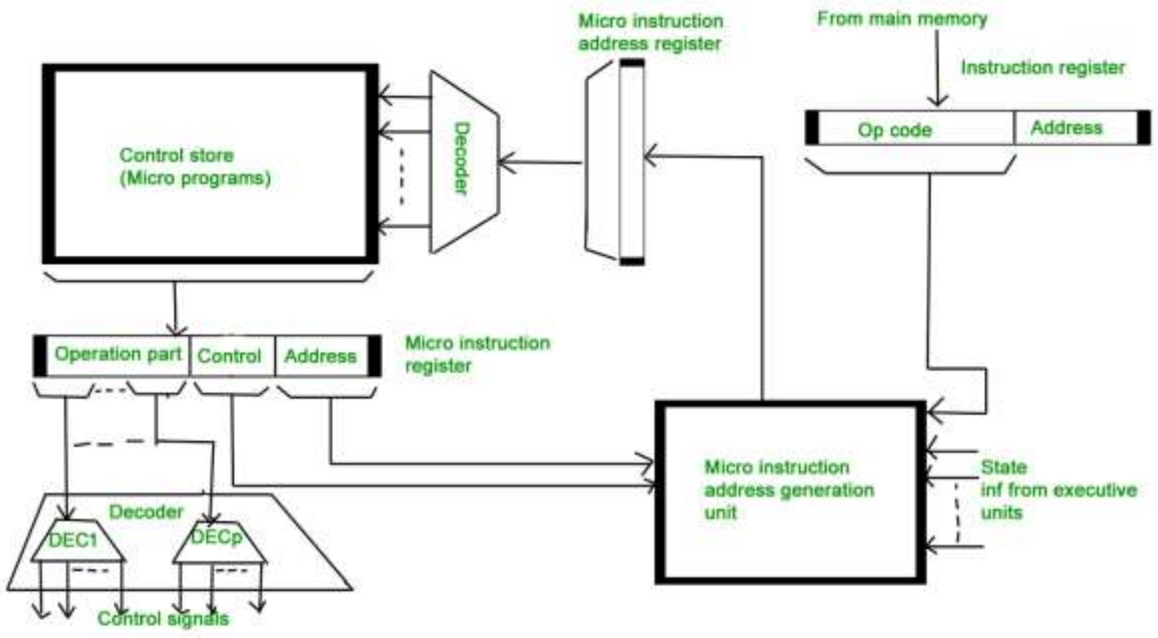
Micro-programmed Control Unit –

- **The control signals associated with operations are stored in special memory units inaccessible by the programmer as Control Words.**
- **Control signals are generated by a program that is similar to machine language programs.**
- **The micro-programmed control unit is slower in speed because of the time it takes to fetch microinstructions from the control memory.**

Some Important Terms

1. **Control Word: A control word is a word whose individual bits represent various control signals.**
2. **Micro-routine: A sequence of control words corresponding to the control sequence of a machine instruction constitutes the micro-routine for that instruction.**
3. **Micro-instruction: Individual control words in this micro-routine are referred to as microinstructions.**

4. **Micro-program: A sequence of micro-instructions is called a micro-program, which is stored in a ROM or RAM called a Control Memory (CM).**
5. **Control Store: the micro-routines for all instructions in the instruction set of a computer are stored in a special memory called the Control Store.**


Micro instruction address register
From main memory
Instruction register
Op code
Address
Control store (Micro programs)
Decoder
Operation part
Control
Address
Micro instruction register
Micro instruction address generation unit
State inf from executive units
Decoder
DEC1
DECp
Control signals

The differences between hardwired and micro-programmed control units:

	Hardwired Control Unit	**Micro-programmed Control Unit**
Implementation	**Fixed set of logic gates and circuits**	**Microcode stored in memory**
Flexibility	**Less flexible, difficult to modify**	**More flexible, easier to modify**

	Hardwired Control Unit	Micro-programmed Control Unit
Instruction Set	Supports limited instruction sets	Supports complex instruction sets
Complexity of Design	Simple design, easy to implement	Complex design, more difficult to implement
Speed	Fast operation	Slower operation due to microcode decoding
Debugging and Testing	Difficult to debug and test	Easier to debug and test
Size and Cost	Smaller size, lower cost	Larger size, higher cost
Maintenance and Upgradability	Difficult to upgrade and maintain	Easier to upgrade and maintain

Types of Micro-programmed Control Unit – Based on the type of Control Word stored in the Control Memory (CM), it is classified into two types :

1. Horizontal Micro-programmed Control Unit :
The control signals are represented in the decoded binary format that is 1 bit/CS. Example: If 53 Control signals are present in the processor then 53 bits are required. More than 1 control signal can be enabled at a time.

- It supports longer control words.
- It is used in parallel processing applications.
- It allows a higher degree of parallelism. If degree is n, n CS is enabled at a time.
- It requires no additional hardware(decoders). It means it is faster than Vertical Microprogrammed.
- It is more flexible than vertical microprogrammed

2. Vertical Micro-programmed Control Unit :
The control signals are represented in the encoded binary format. For N control signals- Log2(N) bits are required.

- It supports shorter control words.
- It supports easy implementation of new control signals therefore it is more flexible.
- It allows a low degree of parallelism i.e., the degree of parallelism is either 0 or 1.
- Requires additional hardware (decoders) to generate control signals, it implies it is slower than horizontal microprogrammed.
- It is less flexible than horizontal but more flexible than that of a hardwired control unit.

5. Explain multiplication of positive numbers example?
PO2

Sol:-

MULTIPLICATION OF POSITIVE NUMBERS :-

The usual algorithm for multiplying integers by hand is illustrated in Figure: 1

for the binary system.

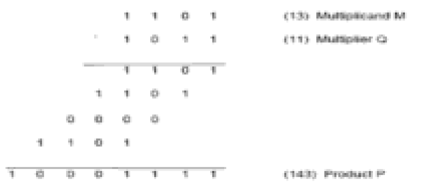

1 1 0 1 (13) Multiplicand M
1 0 1 1 (11) Multiplier Q
1 1 0 1
1 1 0 1
0 0 0 0
1 1 0 1
1 0 0 0 1 1 1 1 (143) Product P

Figure: 1 Manual multiplication algorithm

This algorithm applies to unsigned numbers and to positive signed numbers.

The product of two n-digit numbers can be accommodated in 2n digits, so the

product of the two 4-bit numbers in this example fits into 8 bits. In the binary

system, multiplication of the multiplicand by one bit of the multiplier is easy. If the

multiplier bit is 1, the multiplicand is entered in the appropriate position to be

added to the partial product. If the multiplier bit is 0, then 0s are entered, as in the

third row of the example.

Binary multiplication of positive operands can be implemented in a combinational, two-dimensional logic array, as shown in Figure2.

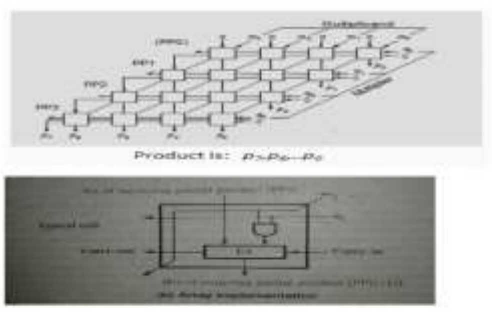


Figure: 2

The main component in each cell is a full adder FA. The AND gate in each

cell determines whether a multiplicand bit, mj, is added to the incoming partial-

product bit, based on the value of the multiplier bit, qi. Each row i, where 0 < i < 3,

adds the multiplicand (appropriately shifted) to the incoming partial product, PPi,

to generate the outgoing partial product, PP (i+ 1)), if qi, = 1. If qi, = 0, PPi is

passed vertically downward unchanged. PP0 is all 0s and PP4 is the desired

product. The multiplicand is shifted left one position per row by the diagonal

signal path.

The worst case signal propagation delay path is from the upper right corner of the array to the high-order product bit output at the bottom left corner of the array. The path consists of the staircase pattern that includes the two cells at the

right end of each row, followed by all the cells in the bottom row. Assuming that

there are two gate delays from the inputs to the outputs of a full adder block, the

path has a total of 6(n -1)-1 gate delays, including the initial AND gate delay in all

cells, for the n x n array. Only the AND gates are actually needed in the first row of the array because the incoming partial product PP0 is zero. This has been taken

into account in developing the delay expression.

Multiplication is usually provided in the machine instruction set of a processor.

High-performance processor chips use an appreciable area of the chip to perform

arithmetic functions on both integer and floating-point operands. Although the

preceding combinational multiplier is easy to understand, it uses many gates for

multiplying numbers of practical size, such as 32- or 64-bit numbers.

Multiplication can also be performed using a mixture of combinational array

techniques, similar to those shown in Figure: 1 and sequential techniques requiring

less combinational logic.

The simplest way to perform multiplication is to use the adder circuitry in

the ALU for a number of sequential steps. The block diagram in Figure: 3 shows

the hardware arrangement for sequential multiplication.

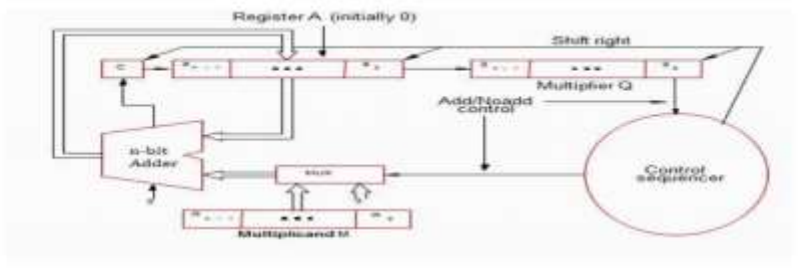


Figure: 3 Register Configuration

This circuit performs multiplication by using a single n-bit adder n times to

implement the spatial addition performed by the n rows of ripple-carry adders of

Figure: 2.

Registers A and Q combined hold PPi while multiplier bit qi generates the

signal Add/Noadd. This signal controls the addition of the multiplicand, M to PPi

to generate PP (i+ 1). The product is computed in n cycles. The partial product

grows in length by one bit per cycle from the initial vector, PP0, of n 0s in register

A. The carry-out from the adder is stored in flip-flop shown at the left end of

register A. At the start, the multiplier is loaded into register Q, the multiplicand

into register M, and C and A are cleared to 0.

At the end of each cycle, C, A, and Q are shifted right one bit position to allow

for growth of the partial product as the multiplier is shifted out of register Q.

Because of this shifting, multiplier bit qi appears at the LSB position of Q to

generate the Add/Noadd signal at the correct time, starting with q0 during the first

cycle, qi during the second cycle, and so on. After they are used, the multiplier bits

are discarded by the right-shift operation. The carry-out from the adder is the

leftmost bit of PP (i +1) and it must be held in the C flip-flop to be shifted right

with the contents of A and Q. After n cycles, the high-order half of the product is

held in register A and the low-order half is in register Q. The multiplication

example of Figure: 1 is shown in Figure: 4 as it would be performed by this

hardware arrangement.

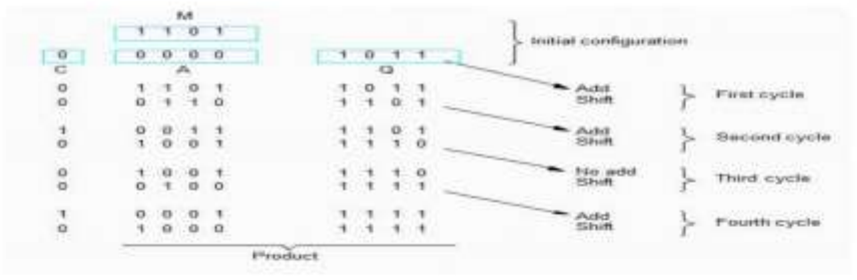


Figure: 4 Multiplication Example (Sequential circuit binary multiplier

6. Explain about signed operant multiplication?
PO2

Sol:- Multiplication of two fixed point binary number in *signed magnitude representation* is done with process of *successive shift* and *add operation*.

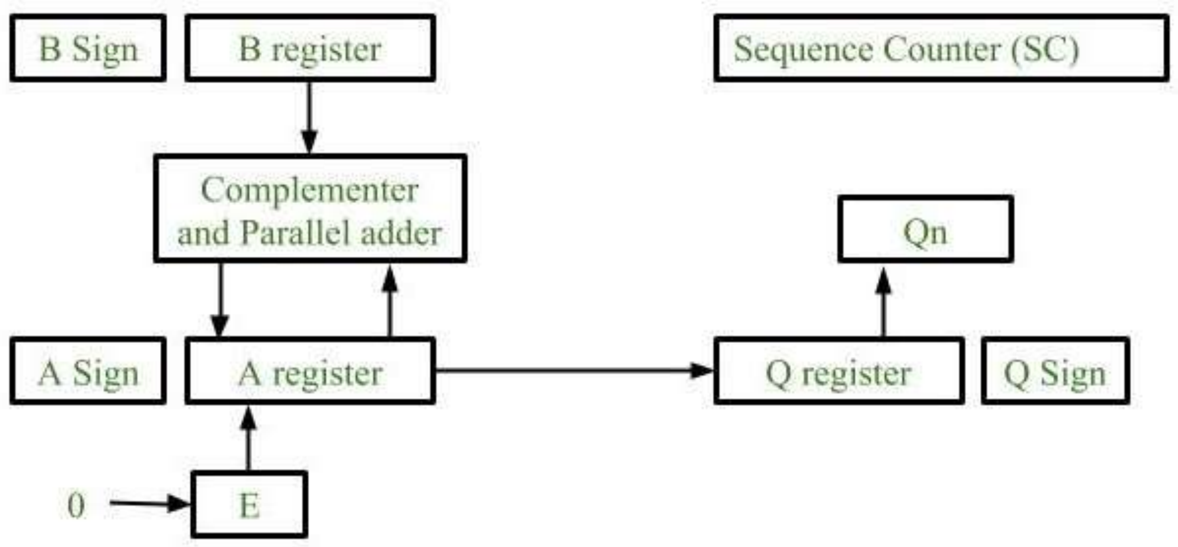
```
      10111 (Multiplicand)
    x 10011 (Multiplier)
     10111
    10111
   00000
  00000
 10111
 011011010     (Product)
```

In the multiplication process we are considering successive bits of the multiplier, least significant bit first.
If the multiplier bit is 1, the multiplicand is copied down else 0's are copied down.

The numbers copied down in successive lines are shifted one position to the left from the previous number.
Finally numbers are added and their sum form the product.

The sign of the product is determined from the sign of the multiplicand and multiplier. If they are alike, sign of the product is positive else negative.

Hardware Implementation :
Following components are required for the *Hardware Implementation* of multiplication algorithm :

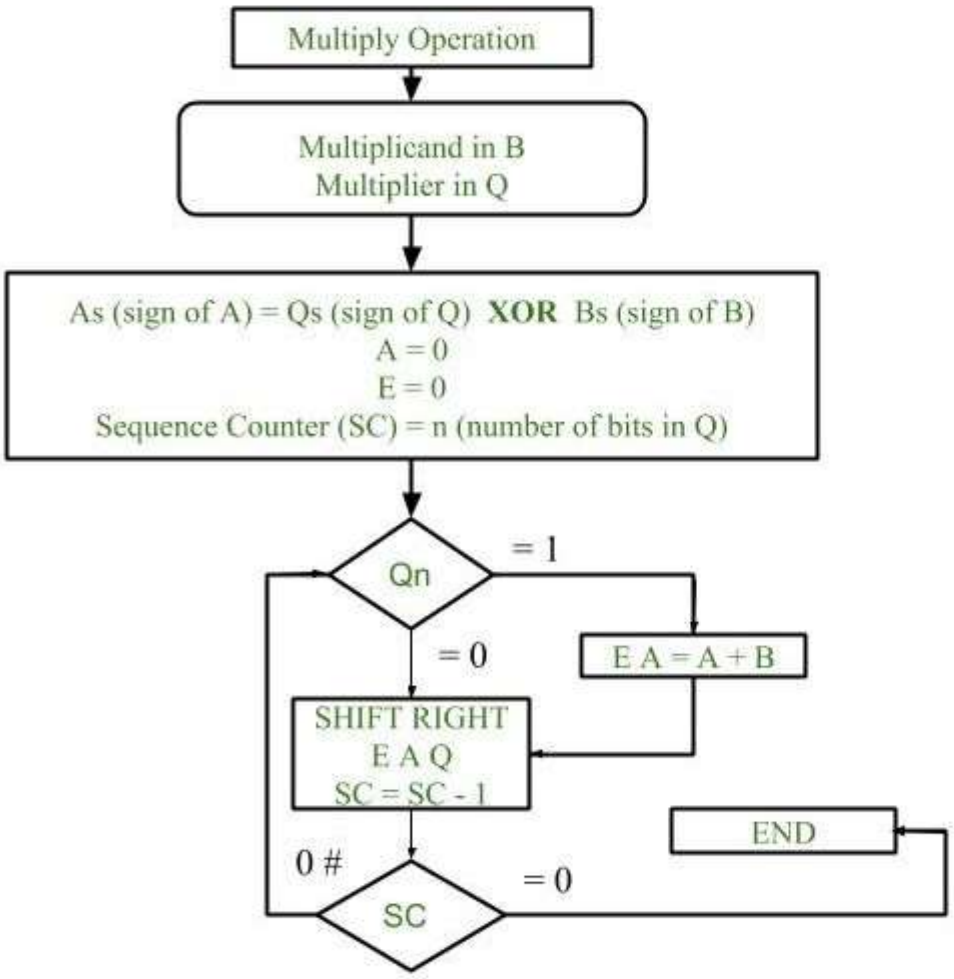


1. Registers:
   Two Registers B and Q are used to store multiplicand and multiplier respectively.
   Register A is used to store partial product during multiplication.
   Sequence Counter register (SC) is used to store number of bits in the multiplier.
2. Flip Flop:
   To store sign bit of registers we require three flip flops (A sign, B sign and Q sign).

Flip flop E is used to store carry bit generated during partial product addition.

3. **Complement and Parallel adder:**
   This hardware unit is used in calculating partial product i.e, perform addition required.

   Flowchart of Multiplication:



1. Initially multiplicand is stored in B register and multiplier is stored in Q register.
2. Sign of registers B (Bs) and Q (Qs) are compared using XOR functionality (i.e., if both the signs are alike, output of XOR operation is 0 unless 1) and output stored in As (sign of A register).

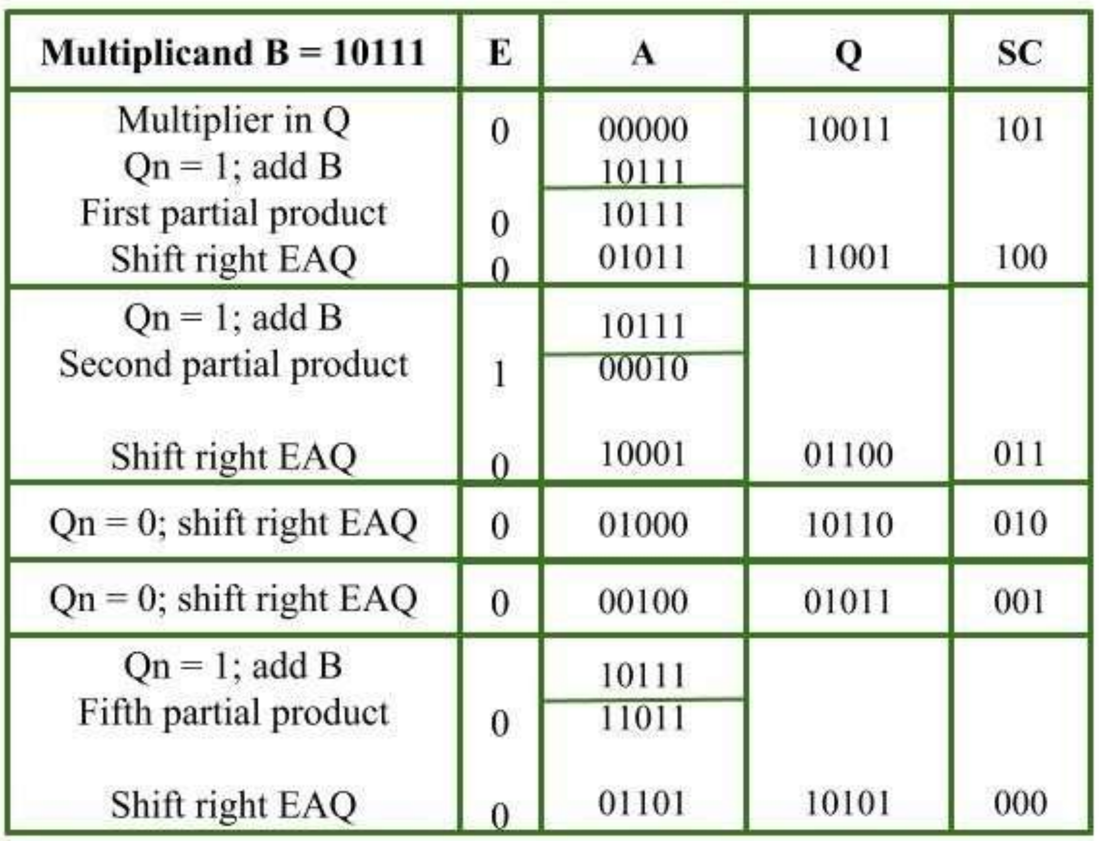
   Note: Initially 0 is assigned to register A and E flip flop. Sequence counter is initialized with value n, n is the number of bits in the Multiplier.

3. Now least significant bit of multiplier is checked. If it is 1 add the content of register A with Multiplicand (register B) and result is assigned in A register with carry bit in flip flop E. Content of E A Q is shifted to right by one position, i.e., content of E is shifted to most significant bit (MSB) of A and least significant bit of A is shifted to most significant bit of Q.
4. If Qn = 0, only shift right operation on content of E A Q is performed in a similar fashion.
5. Content of Sequence counter is decremented by 1.
6. Check the content of Sequence counter (SC), if it is 0, end the process and the final product is present in register A and Q, else repeat the process.

Example:

Multiplicand = 10111

Multiplier = 10011



Multiplicand B = 10111	E	A	Q	SC
Multiplier in Q	0	00000	10011	101
Qn = 1; add B		10111		
First partial product	0	10111		
Shift right EAQ	0	01011	11001	100
Qn = 1; add B		10111		
Second partial product	1	00010		
Shift right EAQ	0	10001	01100	011
Qn = 0; shift right EAQ	0	01000	10110	010
Qn = 0; shift right EAQ	0	00100	01011	001
Qn = 1; add B		10111		
Fifth partial product	0	11011		
Shift right EAQ	0	01101	10101	000

Final product in AQ
0110110101

7. Explain fast multiplication with example?
PO2

Sol:- Karatsuba algorithm for fast multiplication using Divide and Conquer algorithm

Given two binary strings that represent value of two integers, find the product of two strings. For example, if the first bit string is "1100" and second bit string is "1010", output should be 120.

For simplicity, let the length of two strings be same and be n.

A Naive Approach is to follow the process we study in school. One by one take all bits of second number and multiply it with all bits of first number. Finally add all multiplications. This algorithm takes O(n^2) time.

```
        x = 1 0 1 0 0 1  = 41
        y = 1 0 1 0 1 0  = 42
   ------------------------------
            1 0 1 0 0 1 0
          1 0 1 0 0 1
      + 1 0 1 0 0 1
   ------------------------------
      1 1 0 1 0 1 1 1 0 1 0  = 1722
```

Using Divide and Conquer, we can multiply two integers in less time complexity. We divide the given numbers in two halves. Let the given numbers be X and Y.

For simplicity let us assume that n is even

X = Xl*2n/2 + Xr [Xl and Xr contain leftmost and rightmost n/2 bits of X]
Y = Yl*2n/2 + Yr [Yl and Yr contain leftmost and rightmost n/2 bits of Y]

The product XY can be written as follows.

XY = (Xl*2n/2 + Xr)(Yl*2n/2 + Yr)
= 2n XlYl + 2n/2(XlYr + XrYl) + XrYr

If we take a look at the above formula, there are four multiplications of size n/2, so we basically divided the problem of size n into four sub-problems of size n/2. But that doesn't help because the solution of recurrence T(n) = 4T(n/2) + O(n) is O(n^2). The tricky part of this algorithm is to change the middle two terms to some other form so that only one extra multiplication would be sufficient. The following is tricky expression for middle two terms.

XlYr + XrYl = (Xl + Xr)(Yl + Yr) - XlYl- XrYr

So the final value of XY becomes

XY = 2n XlYl + 2n/2 * [(Xl + Xr)(Yl + Yr) - XlYl - XrYr] + XrYr

With above trick, the recurrence becomes T(n) = 3T(n/2) + O(n) and solution of this recurrence is O(n1.59).

***What if the lengths of input strings are different and are not even?* To handle the different length case, we append 0's in the beginning. To handle odd length, we put *floor(n/2)* bits in left half and *ceil(n/2)* bits in right half. So the expression for XY changes to following.**

XY = 22ceil(n/2) XlYl + 2ceil(n/2) * [(Xl + Xr)(Yl + Yr) - XlYl - XrYr] + XrYr

The above algorithm is called Karatsuba algorithm and it can be used for any base.

Following is C++ implementation of above algorithm.

C++JavaPythonC#JavaScript

```
// C++ implementation of Karatsuba algorithm for bit string multiplication.

#include<iostream>

#include<stdio.h>
```

```
using namespace std;

// FOLLOWING TWO FUNCTIONS ARE COPIED FROM http://goo.gl/q0OhZ
// Helper method: given two unequal sized bit strings, converts them to
// same length by adding leading 0s in the smaller string. Returns the
// the new length
int makeEqualLength(string &str1, string &str2)
{
    int len1 = str1.size();
    int len2 = str2.size();
    if (len1 < len2)
    {
        for (int i = 0 ; i < len2 - len1 ; i++)
            str1 = '0' + str1;
        return len2;
    }
    else if (len1 > len2)
    {
        for (int i = 0 ; i < len1 - len2 ; i++)
            str2 = '0' + str2;
    }
    return len1; // If len1 >= len2
}
```

```
// The main function that adds two bit sequences and returns the
addition
string addBitStrings( string first, string second )
{
   string result;  // To store the sum bits

   // make the lengths same before adding
   int length = makeEqualLength(first, second);
   int carry = 0;  // Initialize carry

   // Add all bits one by one
   for (int i = length-1 ; i >= 0 ; i--)
   {
      int firstBit = first.at(i) - '0';
      int secondBit = second.at(i) - '0';

      // boolean expression for sum of 3 bits
      int sum = (firstBit ^ secondBit ^ carry)+'0';

      result = (char)sum + result;

      // boolean expression for 3-bit addition
      carry = (firstBit&secondBit) | (secondBit&carry) | (firstBit&carry);
   }
```

```
    // if overflow, then add a leading 1
    if (carry)  result = '1' + result;

    return result;
}

// A utility function to multiply single bits of strings a and b
int multiplyiSingleBit(string a, string b)
{  return (a[0] - '0')*(b[0] - '0');  }

// The main function that multiplies two bit strings X and Y and returns
// result as long integer
long int multiply(string X, string Y)
{
    // Find the maximum of lengths of x and Y and make length
    // of smaller string same as that of larger string
    int n = makeEqualLength(X, Y);

    // Base cases
    if (n == 0) return 0;
    if (n == 1) return multiplyiSingleBit(X, Y);

    int fh = n/2;   // First half of string, floor(n/2)
    int sh = (n-fh); // Second half of string, ceil(n/2)
    // Find the first half and second half of first string.
    // Refer http://goo.gl/lLmgn for substr method
    string Xl = X.substr(0, fh);
    string Xr = X.substr(fh, sh);

    // Find the first half and second half of second string
    string Yl = Y.substr(0, fh);
    string Yr = Y.substr(fh, sh);

    // Recursively calculate the three products of inputs of size n/2
    long int P1 = multiply(Xl, Yl);
    long int P2 = multiply(Xr, Yr);
    long int P3 = multiply(addBitStrings(Xl, Xr), addBitStrings(Yl, Yr));

    // Combine the three products to get the final result.
    return P1*(1<<(2*sh)) + (P3 - P1 - P2)*(1<<sh) + P2;
}

// Driver program to test above functions
int main()
{
    printf ("%ld\n", multiply("1100", "1010"));
    printf ("%ld\n", multiply("110", "1010"));
    printf ("%ld\n", multiply("11", "1010"));
    printf ("%ld\n", multiply("1", "1010"));
    printf ("%ld\n", multiply("0", "1010"));
    printf ("%ld\n", multiply("111", "111"));
    printf ("%ld\n", multiply("11", "11"));
}
```

Output

```
120
60
30
10
0
49
9
```

Time Complexity: Time complexity of the above solution is O(nlog23) = O(n1.59).
Time complexity of multiplication can be further improved using another Divide and Conquer algorithm, fast Fourier transform. We will soon be discussing fast Fourier transform as a separate post.

Auxiliary Space: O(n)

Exercise:
The above program returns a long int value and will not work for big strings. Extend the above program to return a string instead of a long int value.

Solution:
Multiplication process for large numbers is an important problem in Computer Science. Given approach uses Divide and Conquer methodology.
Run the code to see the time complexity comparison for normal Binary

Multiplication and Karatsuba Algorithm.
You can see the full code in this repository

Examples:

First Binary Input :
101001010101010010101001010100101010010101010010101
Second Binary Input :
101001010101010010101001010100101010010101010010101
Decimal Output : Not Representable
Output : 2.1148846e+30

First Binary Input : 1011
Second Binary Input : 1000
Decimal Output : 88
Output : 5e-05

8. Explain about floating point numbers and operation with example? PO2

Sol:- Floating Point Representation – Basics

There are posts on representation of floating point format. The objective of this article is to provide a brief introduction to floating point format.

The following description explains terminology and primary details of IEEE 754 binary floating-point representation. The discussion confines to single and double precision formats.

Usually, a real number in binary will be represented in the following format,

ImIm-1…I2I1I0.F1F2…FnFn-1

Where Im and Fn will be either 0 or 1 of integer and fraction parts respectively.

A finite number can also represented by four integers components, a sign (s), a base (b), a significant (m), and an exponent (e). Then the numerical value of the number is evaluated as

(-1)s x m x be ________ *Where m < |b|*

Depending on base and the number of bits used to encode various components, the IEEE 754 standard defines five basic formats. Among the five formats, the binary32 and the binary64 formats are single precision and double precision formats respectively in which the base is 2.

What is Floating Point Representation?

The Floating point representation is a way to the encode numbers in a format that can handle very large and very small values. It is based on scientific notation where numbers are represented as a fraction and an exponent. In computing, this representation allows for trade-off between range and precision.

Format: A floating point number is typically represented as:

Value=Sign × Significand × BaseExponent

where:

- Sign: Indicates whether the number is positive or negative.
- Significand (Mantissa): Represents the precision bits of the number.
- Base: Usually 2 in binary systems.
- Exponent: Determines the scale of the number.

Need for Floating Point Representation

The Floating point representation is crucial because:

- Range: It can represent a wide range of values from the very large to very small numbers.

- Precision: It provides a good balance between the precision and range, making it suitable for the scientific computations, graphics and other applications where exact values and wide ranges are necessary.
- Flexibility: It adapts to different scales of numbers allowing for the efficient storage and computation of real numbers in the computer systems.

Number System and Data Representation

- Number Systems: The Floating point representation often uses binary (base-2) systems for the digital computers. Other number systems like decimal (base-10) or hexadecimal (base-16) may be used in the different contexts.
- Data Representation: This includes how numbers are stored in the computer memory involving binary encoding and the representation of the various data types.

Table – Precision Representation

Precision	Base	Sign	Exponent	Significant
Single precision	2	1	8	23+1
Double precision	2	1	11	52+1

Single Precision Format

The single precision format has 23 bits for significant (1 represents implied bit, details below), 8 bits for exponent and 1 bit for sign.

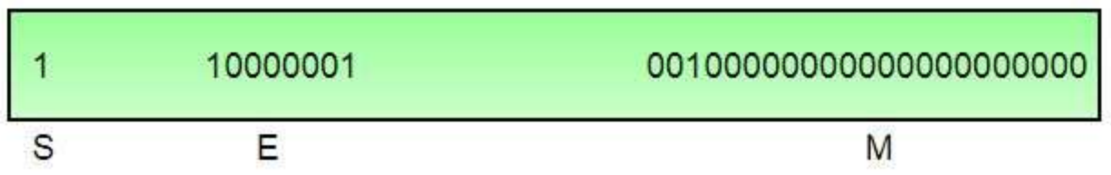
For example, the rational number 9÷2 can be converted to single precision float format as following,

9(10) ÷ 2(10) = 4.5(10) = 100.1(2)

The result said to be *normalized*, if it is represented with leading 1 bit, i.e. 1.001(2) x 22. (Similarly when the number 0.000000001101(2) x 23 is normalized, it appears as 1.101(2) x 2-6). Omitting this implied 1 on left extreme gives us the *mantissa* of float number. A normalized number provides more accuracy than corresponding *de-normalized* number. The implied most significant bit can be used to represent even more accurate significant (23 + 1 = 24 bits) which is
called *subnormal* representation. *The floating point numbers are to be represented in normalized form*.

The subnormal numbers fall into the category of de-normalized numbers. The subnormal representation slightly reduces the exponent range and can't be normalized since that would result in an exponent which doesn't fit in the field. Subnormal numbers are less accurate, i.e. they have less room for nonzero bits in the fraction field, than normalized numbers. Indeed, the accuracy drops as the size of the subnormal number decreases. However, the subnormal representation is useful in filing gaps of floating point scale near zero.

In other words, the above result can be written as (-1)0 x 1.001(2) x
22 which yields the integer components as s = 0, b = 2, significant (m) = 1.001, mantissa = 001 and e = 2. The corresponding single precision floating number can be represented in binary as shown below,



Where the exponent field is supposed to be 2, yet encoded as 129 (127+2) called *biased exponent*. The exponent field is in plain binary format which also represents negative exponents with an encoding (like sign magnitude, 1's complement, 2's complement, etc.). The biased exponent is used for the representation of negative exponents. The biased exponent has advantages over other negative representations in

performing bitwise comparing of two floating point numbers for equality.

A *bias* of (2n-1 – 1), where n is # of bits used in exponent, is added to the exponent (e) to get biased exponent (*E*). So, the biased exponent (*E*) of *single precision* number can be obtained as

E = e + 127

The range of exponent in single precision format is -128 to +127. Other values are used for special symbols.

Note: When we unpack a floating point number the exponent obtained is the biased exponent. Subtracting 127 from the biased exponent we can extract unbiased exponent.

Double Precision Format

The double precision format has 52 bits for significant (1 represents implied bit), 11 bits for exponent and 1 bit for sign. All other definitions are same for double precision format, except for the size of various components.

Precision

The smallest change that can be represented in floating point representation is called as precision. The fractional part of a single precision normalized number has exactly 23 bits of resolution, (24 bits with the implied bit). This corresponds to log(10) (223) = 6.924 = 7 (the characteristic of logarithm) decimal digits of accuracy. Similarly, in case of double precision numbers the precision is log(10) (252) = 15.654 = 16 decimal digits.

Accuracy

Accuracy in floating point representation is governed by number of significant bits, whereas range is limited by exponent. Not all real numbers can exactly be represented in floating point format. For any numberwhich is not floating point number, there are two options for

floating point approximation, say, the closest floating point number *less than x* as x_ and the closest floating point number *greater than x* as x+. A *rounding* operation is performed on number of significant bits in the mantissa field based on the selected mode. The *round down* mode causes x set to x_, the *round up* mode causes x set to x+, the *round towards zero* mode causes x is either x_ or x+ whichever is between zero and. The *round to nearest* mode sets x to x_ or x+ whichever is nearest to x. Usually *round to nearest* is most used mode. The closeness of floating point representation to the actual value is called as *accuracy*.

Special Bit Patterns

The standard defines few special floating point bit patterns. Zero can't have most significant 1 bit, hence can't be normalized. The hidden bit representation requires a special technique for storing zero. We will have two different bit patterns +0 and -0 for the same numerical value zero. For single precision floating point representation, these patterns are given below,

0 00000000 00000000000000000000000 = +0

1 00000000 00000000000000000000000 = -0

Similarly, the standard represents two different bit patterns for +INF and -INF. The same are given below,

0 11111111 00000000000000000000000 = +INF

1 11111111 00000000000000000000000 = -INF

All of these special numbers, as well as other special numbers (below) are subnormal numbers, represented through the use of a special bit pattern in the exponent field. This slightly reduces the exponent range, but this is quite acceptable since the range is so large.

An attempt to compute expressions like 0 x INF, 0 ÷ INF, etc. make no mathematical sense. The standard calls the result of such expressions as Not a Number (NaN). Any subsequent expression with NaN yields NaN. The representation of NaN has non-zero significant and all 1s in the

exponent field. These are shown below for single precision format (x is don't care bits),

x 11111111 m0000000000000000000000

Where *m* can be 0 or 1. This gives us two different representations of NaN.

0 11111111 00000000000000000000001 _____________ Signaling NaN (SNaN)

0 11111111 10000000000000000000001 _____________Quiet NaN (QNaN)

Usually QNaN and SNaN are used for error handling. QNaN do not raise any exceptions as they propagate through most operations. Whereas SNaN are which when consumed by most operations will raise an invalid exception.

Overflow and Underflow

***Overflow* is said to occur when the true result of an arithmetic operation is finite but larger in magnitude than the largest floating point number which can be stored using the given precision. *Underflow* is said to occur when the true result of an arithmetic operation is smaller in magnitude (infinitesimal) than the smallest normalized floating point number which can be stored. Overflow can't be ignored in calculations whereas underflow can effectively be replaced by zero.**

Endianness

The IEEE 754 standard defines a binary floating point format. The architecture details are left to the hardware manufacturers. The storage order of individual bytes in binary floating point numbers varies from architecture to architecture.

Advantages

- **Wide Range: Can represent very large and very small numbers.**

- Efficient Calculation: The Suitable for a wide range of the scientific, engineering and graphics applications where precision and range are important.
- Standardization: The Floating point representation is standardized (IEEE 754) which ensures consistency and compatibility across the different systems and programming languages.

Disadvantages

- Precision Issues: The Floating point numbers can suffer from the precision errors due to rounding and truncation.
- Complexity: The More complex than fixed-point representation requiring more computational resources.
- Overhead: Operations involving the floating point numbers can be slower and require more memory compared to integer operations.

Applications

- Scientific Computations: Used in simulations, modeling and calculations requiring the high precision and large ranges.
- Graphics: Essential in the rendering and manipulating graphical data where precise calculations are needed.
- Engineering: The Applied in fields such as aerospace, mechanical engineering and electronics for the accurate measurements and simulations.

Conclusion

The Floating point representation is a powerful tool for handling real numbers in computing offering the balance between range and precision. While it has its drawbacks such as the precision issues and increased computational complexity its advantages make it indispensable for the various scientific, engineering and graphical applications. Understanding floating point representation helps in making informed decisions about numerical computations and data representation.

9. Explain about design of fast order?
PO2

Sol:- Designing for "fast order" in a general sense involves optimizing processes to deliver products or services quickly and efficiently, which can be achieved through various techniques, including streamlining workflows, leveraging technology, and focusing on key performance indicators (KPIs).

Here's a breakdown of design considerations for "fast order" in different contexts:

1. Understanding the Context:

- **What is "fast order" in your context?**

  Is it about:

  - **E-commerce: Quick order processing, fast shipping, and efficient returns?**
  - **Restaurant/Food Delivery: Rapid order taking, fast cooking, and on-time delivery?**
  - **Manufacturing: Short lead times for production and delivery?**
  - **Customer Service: Quick resolution of issues and inquiries?**
  - **Digital Products: Fast downloads, instant access to content?**

- **What are the key performance indicators (KPIs) for "fast order"?**

  Examples include:

  - **Average order processing time**
  - **Order fulfillment time**
  - **Delivery time**
  - **Customer satisfaction with speed**
  - **Error rates**

2. Design Principles for Fast Order:

- Streamline Workflows:
    - Identify bottlenecks: Analyze the entire process from order placement to delivery and identify areas where delays occur.
    - Automate tasks: Use technology to automate repetitive tasks, such as order entry, data processing, and shipping label generation.
    - Simplify processes: Remove unnecessary steps and make the process as intuitive and efficient as possible.
    - Cross-functional collaboration: Ensure seamless communication and coordination between different departments or teams involved in the process.
- Leverage Technology:
    - Use technology to streamline processes: Implement systems that can automate tasks, track orders, and provide real-time updates.
    - Data analytics: Use data to identify trends, predict demand, and optimize processes.
    - Cloud computing: Leverage cloud infrastructure for scalability and flexibility.
    - AI and Machine Learning: Use AI to personalize experiences, predict demand, and automate tasks.
- Focus on Customer Experience:
    - Provide clear and timely updates: Keep customers informed about the status of their order and any potential delays.
    - Make it easy to track orders: Provide a simple and intuitive way for customers to track their order progress.
    - Offer flexible options: Provide customers with a variety of options for order placement, payment, and delivery.

- Be responsive to customer needs: Address customer inquiries and issues quickly and efficiently.

- **Optimize Resources:**
    - **Inventory management: Ensure that you have the right products in the right quantities at the right time.**
    - **Logistics and transportation: Optimize your logistics and transportation network to ensure fast and reliable delivery.**
    - **Staff training: Ensure that your staff is trained to handle orders quickly and efficiently.**
- **Continuous Improvement:**
    - **Regularly review and analyze your processes: Identify areas for improvement and implement changes to optimize performance.**
    - **Gather feedback from customers: Use customer feedback to identify areas where you can improve the customer experience.**
    - **Stay up-to-date with the latest technologies and trends: Continuously learn and adapt to stay ahead of the competition.**

3. Examples of "Fast Order" Design:

- **E-commerce:**

  Amazon's Prime shipping, Instacart's same-day delivery, and DoorDash's fast food delivery are examples of businesses that have successfully designed for "fast order".
- **Restaurant/Food Delivery:**

  Apps like Uber Eats and Grubhub have streamlined the ordering and delivery process, making it faster and easier for customers.
- **Manufacturing:**

  Just-in-time inventory management and lean manufacturing techniques are used to reduce lead times and improve efficiency.

10. Write algorithm for booths with flow charts?
PO2

Sol:- Booth's Algorithm

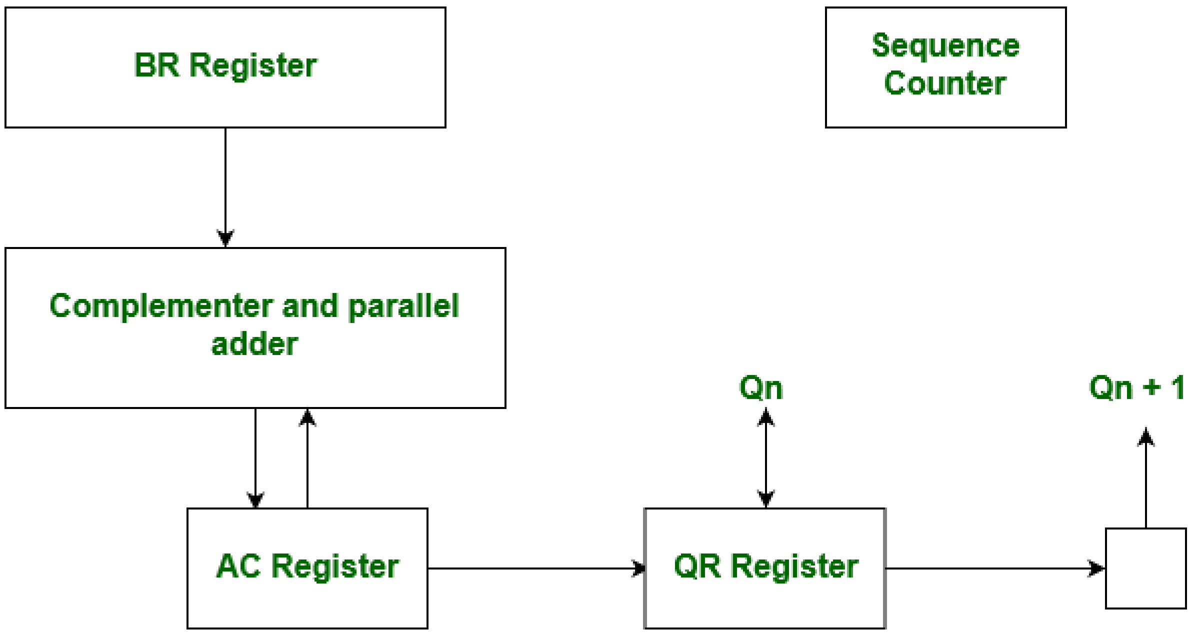
Booth algorithm gives a procedure for multiplying binary integers in signed 2's complement representation in efficient way, i.e., less number of additions/subtractions required. It operates on the fact that strings of 0's in the multiplier require no addition but just shifting and a string of 1's in the multiplier from bit weight 2^k to weight 2^m can be treated as 2^(k+1) to 2^m. As in all multiplication schemes, booth algorithm requires examination of the multiplier bits and shifting of the partial product. Prior to the shifting, the multiplicand may be added to the partial product, subtracted from the partial product, or left unchanged according to following rules:

1. The multiplicand is subtracted from the partial product upon encountering the first least significant 1 in a string of 1's in the multiplier
2. The multiplicand is added to the partial product upon encountering the first 0 (provided that there was a previous '1') in a string of 0's in the multiplier.
3. The partial product does not change when the multiplier bit is identical to the previous multiplier bit.

Booth's Algorithm optimizes binary multiplication. For a better grasp of computer organization, the GATE CS Self-Paced Course provides clear explanations and examples.

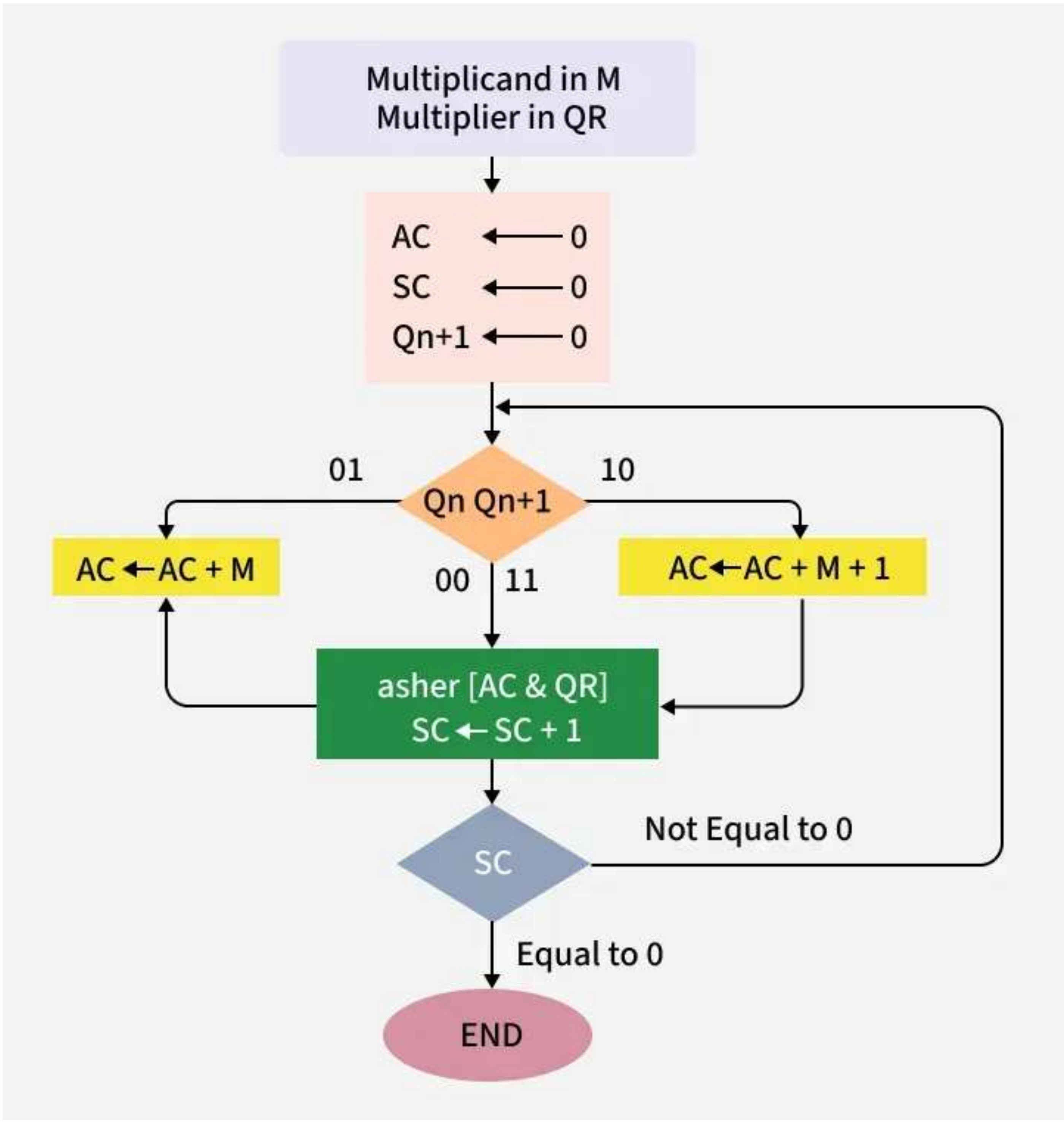
Hardware Implementation of Booths Algorithm

The hardware implementation of the booth algorithm requires the register configuration shown in the figure below:



Booth’s Algorithm Flowchart

We name the register as A, B and Q, AC, BR and QR respectively. Qn designates the least significant bit of multiplier in the register QR. An extra flip-flop Qn+1is appended to QR to facilitate a double inspection of the multiplier. The flowchart for the booth algorithm is shown below:



Booth's Algorithm Flowchart

AC and the appended bit Qn+1 are initially cleared to 0 and the sequence SC is set to a number n equal to the number of bits in the multiplier. The two bits of the multiplier in Qn and Qn+1are inspected. If the two bits are equal to 10, it means that the first 1 in a string has been encountered. This requires subtraction of the multiplicand from the partial product in AC. If the 2 bits are equal to 01, it means that the first 0 in a string of 0's

has been encountered. This requires the addition of the multiplicand to the partial product in AC. When the two bits are equal, the partial product does not change. An overflow cannot occur because the addition and subtraction of the multiplicand follow each other. As a consequence, the 2 numbers that are added always have a opposite signs, a condition that excludes an overflow. The next step is to shift right the partial product and the multiplier (including Qn+1). This is an arithmetic shift right (ashr) operation which AC and QR to the right and leaves the sign bit in AC unchanged. The sequence counter is decremented and the computational loop is repeated n times. Product of negative numbers is important, while multiplying negative numbers we need to find 2's complement of the number to change its sign, because it's easier to add instead of performing binary subtraction. product of two negative number is demonstrated below along with 2's complement.

Example – A numerical example of booth's algorithm is shown below for n = 4. It shows the step by step multiplication of -5 and -7.

BR = -5 = 1011,
BR' = 0100, <– 1's Complement (change the values 0 to 1 and 1 to 0)
BR'+1 = 0101 <– 2's Complement (add 1 to the Binary value obtained after 1's complement)
QR = -7 = 1001 <– 2's Complement of 0111 (7 = 0111 in Binary)
The explanation of first step is as follows: Qn+1
AC = 0000, QR = 1001, Qn+1 = 0, SC = 4
Qn Qn+1 = 10
So, we do AC + (BR)'+1, which gives AC = 0101
On right shifting AC and QR, we get
AC = 0010, QR = 1100 and Qn+1 = 1

__mask-th-td__index=1____mask-th-td__index=2____mask-th-td__index=3____mask-th-td__index=4____mask-th-td__index=5____mask-th-td__index=6____mask-th-td__index=7____mask-th-td__index=8____mask-th-td__index=9____mask-th-td__index=10____mask-th-td__index=11____mask-th-td__index=12____mask-th-

td__index=13____mask-th-td__index=14____mask-th-td__index=15____mask-th-td__index=16____mask-th-td__index=17____mask-th-td__index=18____mask-th-td__index=19____mask-th-td__index=20____mask-th-td__index=21____mask-th-td__index=22____mask-th-td__index=23____mask-th-td__index=24____mask-th-td__index=25____mask-th-td__index=26____mask-th-td__index=27____mask-th-td__index=28____mask-th-td__index=29____mask-th-td__index=30____mask-th-td__index=31____mask-th-td__index=32____mask-th-td__index=33____mask-th-td__index=34____mask-th-td__index=35____mask-th-td__index=36____mask-th-td__index=37____mask-th-td__index=38____mask-th-td__index=39____mask-th-td__index=40____mask-th-td__index=41____mask-th-td__index=42____mask-th-td__index=43____mask-th-td__index=44____mask-th-td__index=45__

Product is calculated as follows:

Product = AC QR
Product = 0010 0011 = 35

Advantages

- **Faster than traditional multiplication: Booth's algorithm is faster than traditional multiplication methods, requiring fewer steps to produce the same result.**
- **Efficient for signed numbers: The algorithm is designed specifically for multiplying signed binary numbers, making it a more efficient method for multiplication of signed numbers than traditional methods.**
- **Lower hardware requirement: The algorithm requires fewer hardware resources than traditional multiplication methods, making it more suitable for applications with limited hardware resources.**

- **Widely used in hardware: Booth's algorithm is widely used in hardware implementations of multiplication operations, including digital signal processors, microprocessors, and FPGAs.**

Disadvantages

- **Complex to understand: The algorithm is more complex to understand and implement than traditional multiplication methods.**
- **Limited applicability: The algorithm is only applicable for multiplication of signed binary numbers, and cannot be used for multiplication of unsigned numbers or numbers in other formats without additional modifications.**
- **Higher latency: The algorithm requires multiple iterations to calculate the result of a single multiplication operation, which increases the latency or delay in the calculation of the result.**
- **Higher power consumption: The algorithm consumes more power compared to traditional multiplication methods, especially for larger inputs.**

Application of Booth's Algorithm

1. **Chip and computer processors: Corner's Calculation is utilized in the equipment execution of number-crunching rationale units (ALUs) inside microchips and computer chips. These parts are liable for performing number juggling and coherent procedure on twofold information. Proficient duplication is fundamental in different applications, including logical registering, designs handling, and cryptography. Corner's Calculation lessens the quantity of piece movements and augmentations expected to perform duplication, bringing about quicker execution and better in general execution.**
2. **Digital Signal Processing (DSP): DSP applications frequently include complex numerical tasks, for example, sifting and convolution. Duplicating enormous twofold numbers is a principal activity in these errands. Corner's Calculation permits DSP frameworks to perform**

duplications all the more productively, empowering ongoing handling of sound, video, and different sorts of signs.

3. Hardware Accelerators: Many particular equipment gas pedals are intended to perform explicit assignments more productively than broadly useful processors. Corner's Calculation can be integrated into these gas pedals to accelerate augmentation activities in applications like picture handling, brain organizations, and AI.
4. Cryptography: Cryptographic calculations, like those utilized in encryption and computerized marks, frequently include particular exponentiation, which requires proficient duplication of huge numbers. Corner's Calculation can be utilized to speed up the measured augmentation step in these calculations, working on the general proficiency of cryptographic tasks.
5. High-Performance Computing (HPC): In logical reenactments and mathematical calculations, enormous scope augmentations are oftentimes experienced. Corner's Calculation can be carried out in equipment or programming to advance these duplication tasks and improve the general exhibition of HPC frameworks.
6. Implanted Frameworks: Inserted frameworks frequently have restricted assets regarding handling power and memory. By utilizing Corner's Calculation, fashioners can upgrade augmentation activities in these frameworks, permitting them to perform all the more proficiently while consuming less energy.
7. Network Parcel Handling: Organization gadgets and switches frequently need to perform estimations on bundle headers and payloads. Augmentation activities are regularly utilized in these estimations, and Corner's Calculation can assist with diminishing handling investment utilization in these gadgets.
8. Advanced Channels and Balancers: Computerized channels and adjusters in applications like sound handling and correspondence frameworks require productive augmentation of coefficients with input tests. Stall's

Calculation can be utilized to speed up these increases, prompting quicker and more precise sifting activities.

Basically, Corner's Calculation finds its application any place productive paired duplication is required, particularly in situations where speed, power proficiency, and equipment streamlining are significant elements.

Best Case and Worst Case Occurrence: Best case is when there is a large block of consecutive 1's and 0's in the multipliers, so that there is minimum number of logical operations taking place, as in addition and subtraction. Worst case is when there are pairs of alternate 0's and 1's, either 01 or 10 in the multipliers, so that maximum number of additions and subtractions are required.

PART-B (2 MARKS)

1. What are arithmetic operators?
PO1

Sol:- Arithmetic operators are symbols that perform mathematical operations on numbers. They are used to calculate results in mathematical equations and programs.

Examples of arithmetic operators

- **Addition: (+) Adds two values together**
- **Subtraction: (-) Subtracts one value from another**
- **Multiplication: (*) Multiplies two values together**
- **Division: (/) Divides one value by another**
- **Remainder: (%) Shows the remainder after division**
- **Exponentiation: Raises a value to a power**

Name	Operator	Arithmetic Operation	Syntax
Addition	+	Add two operands.	x + y
Subtraction	–	Subtract the second operand from the first operand.	x – y
Multiplication	*	Multiply two operands.	x * y
Division	/	Divide the first operand by the second operand.	x / y
Modulus	%	Calculate the remainder when the first operand is divided by the second operand.	x % y

2. What is mean by bus structure?
PO1

Sol:- In computer architecture, a "bus structure" refers to the arrangement and organization of buses, which are shared communication pathways, that connect multiple components of a computer system, enabling data and control signal exchange.

Here's a more detailed explanation:

- **What is a Bus?**

  A bus is a set of wires or traces on a circuit board that allows different components within a computer system to communicate with each other.

- **Why is it Important?**

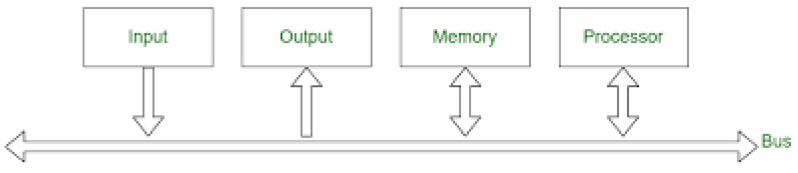
  Buses are essential for enabling data transfer, address identification, and control signal transmission between the CPU, memory, input/output devices, and other components.

- **Types of Buses:**
  - **Data Bus: Carries data between components.**
  - **Address Bus: Identifies the memory locations or I/O devices that the data is being sent to or retrieved from.**
  - **Control Bus: Transmits control signals, such as read/write commands, to coordinate data transfer.**

- **Bus Structures:**
  - **Single Bus Structure: Uses a single shared bus for all communication, which can lead to bottlenecks when multiple components need to access the bus simultaneously.**
  - **Multiple Bus Structure: Uses multiple buses to improve performance by allowing parallel communication between different components.**

- **Example:**

  A typical PC has a system bus, which consists of control, data, and address buses.



Single Bus Structure

3. Define hard wired and multi programmed control?
PO1

Sol:- In computer architecture, a hardwired control unit uses fixed logic circuits to generate control signals, while a microprogrammed control unit uses a stored sequence of microinstructions (microcode) to generate these signals.

Here's a more detailed explanation:

Hardwired Control Unit:

- **Definition:**

  A hardwired control unit uses a network of logic gates and circuits to directly generate the control signals necessary for executing instructions.

- **Characteristics:**
  - **Speed: Generally faster because the control signals are generated directly by hardware.**
  - **Flexibility: Less flexible and harder to modify, as changes require physical alterations to the circuitry.**
  - **Cost: Can be more expensive to design and implement due to the complexity of the circuitry.**

- **Example:**

  Imagine a circuit that, when a specific input is received, directly activates a specific output, like a light turning on when a button is pressed.

Microprogrammed Control Unit:

- **Definition:**

  A microprogrammed control unit uses a stored sequence of microinstructions, or microcode, to generate control signals.

- **Characteristics:**
  - **Speed: Generally slower than hardwired control units because it involves fetching and executing microinstructions.**

    - Flexibility: More flexible and easier to modify, as changes can be made by altering the microcode rather than the hardware.
    - Cost: Can be more cost-effective, as the hardware required is simpler.
- Example:

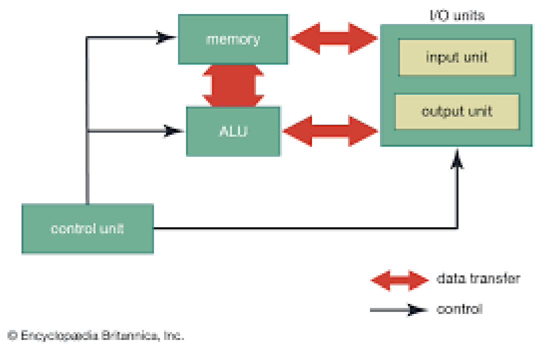
  Think of a computer program that, when a specific input is received, executes a series of instructions, each of which activates a specific output.

  Here's a table summarizing the key differences:

Feature	Hardwired Control Unit	Microprogrammed Control Unit
Control Signal Generation	Fixed logic circuits	Stored microinstructions (microcode)
Speed	Generally faster	Generally slower
Flexibility	Less flexible, harder to modify	More flexible, easier to modify
Cost	Can be more expensive	Can be more cost-effective

4. What is the process happening in control unit? PO1

Sol:- control unit, subcomponent of a central processing unit (CPU) that manages a computer's operations. The control unit fetches instructions from the CPU's memory, represented in bits, and translates those instructions into control signals in the form of pulses of electricity or light.



5. What is multiprocessor?
    PO1

Sol:- A multiprocessor system is a computer system that utilizes two or more central processing units (CPUs) or processors to execute instructions concurrently, aiming to improve overall system performance and speed.

Here's a more detailed explanation:

- Multiple Processors:

  Unlike a traditional computer with a single CPU, a multiprocessor system has multiple CPUs that can work simultaneously.

- Shared Resources:

  These processors typically share access to a common memory space (RAM) and peripherals.

- Increased Performance:

  The primary goal of using a multiprocessor system is to increase the system's execution speed by allowing multiple tasks to be processed concurrently.

- Types of Multiprocessors:

  - Shared Memory Multiprocessors: All CPUs share the same memory space.
  - Distributed Memory Multiprocessors: Each CPU has its own private memory.
- Multiprocessing vs Multiprogramming:

  Multiprocessing involves executing multiple processes simultaneously on multiple processors, while multiprogramming involves running multiple programs concurrently on a single processor.
- Examples:

  Modern multicore processors, where a single chip contains multiple processing cores, are a common form of multiprocessor systems.

6. Define fast multiplication?
   PO1

Sol:- "Fast multiplication" refers to algorithms and techniques that compute the product of two numbers more efficiently than the standard "grade school" multiplication method, often using divide-and-conquer strategies or specialized hardware.

7. Explain about relational operators?
   PO1

Sol:- Relational operators are symbols used in programming to compare two values and determine a relationship (like equality, greater than, less than) resulting in a boolean value (true or false).

Here's a breakdown:

- Purpose:

  Relational operators are used to compare two values or expressions and determine if a specific relationship holds true.

- **Common Operators:**
    - **== (equal to)**
    - **!= (not equal to)**
    - **> (greater than)**
    - **< (less than)**
    - **>= (greater than or equal to)**
    - **<= (less than or equal to)**
- **Boolean Output:**

  Relational operators always evaluate to either true or false, representing whether the comparison is valid or not.
- **Usage:**

  They are commonly used in conditional statements (like if, else if, else) and loops to control program flow based on the outcome of the comparison.
- **Examples:**
    - **if (age >= 18) { ... } (checks if age is 18 or older)**
    - **while (count != target) { ... } (continues the loop as long as count is not equal to target)**

8. What is mean by integer division?
PO1

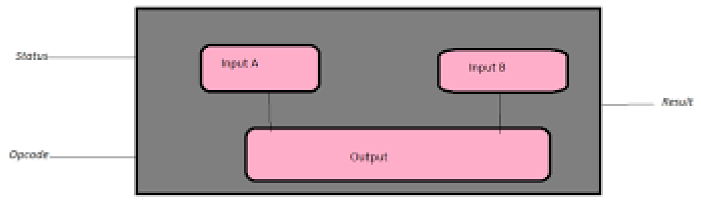
Sol:- Integer division refers to the process of dividing two integers and obtaining the quotient without any fractional part.

9. Define multi computer?
PO1

Sol:- A "multicomputer" refers to a computer system composed of multiple, independent computers connected via a network, where each computer has its own memory and processors, and communication between them is explicit, unlike multiprocessors that share memory.

10. What is the purpose of arithmetic and logic unit?
PO1

Sol:- The purpose of an Arithmetic Logic Unit (ALU) is to perform arithmetic and logical operations on data, serving as a fundamental building block of a computer's CPU, executing tasks like addition, subtraction, bitwise operations, and comparisons.



Unit 4

MEMORY ORGANISATION

***Random Access Memory (RAM) and Read Only Memory (ROM)**

Memory is a fundamental component of computing systems, essential for performing various tasks efficiently. It plays a crucial role in how computers operate, influencing speed, performance, and data management.

Random Access Memory (RAM) is primary-volatile memory and Read Only Memory (ROM) is primary-non-volatile memory. It is also called as read write memory or the main memory or the primary memory. The programs and data that the CPU requires during execution of a program are stored in this memory

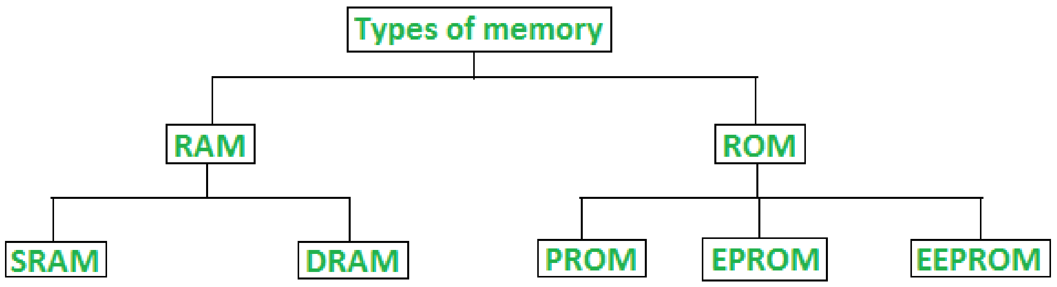
Types of Memory:

Memory is the most essential element of a computing system because without it computer can't perform simple tasks. Both types of memory (RAM and ROM) are important for the computer, but they serve different purposes.

RAM is used to store data that the computer is currently using, while ROM is used to store data that the computer needs to boot and operate. RAM is faster than ROM, as the data stored in it can be accessed and modified in any order, while data stored in ROM can only be read.

Computer memory is of two basic types:

1. **Primary memory (RAM and ROM)**
2. **Secondary memory (Hard Drive, CD, etc).**



Classification of computer memory

Random Access Memory (RAM)

Random Access Memory (RAM) is a type of computer memory that is used to temporarily store data that the computer is currently using or processing. RAM is volatile memory, which means that the data stored in it is lost when the power is turned off. RAM is typically used to store the operating system, application programs, and data that the computer is currently using.

- It is also called read-write *memory* or the *main memory* or the *primary memory* .
- The programs and data that the CPU requires during the execution of a program are stored in this memory.
- It is a volatile memory as the data is lost when the power is turned off.

Types of Random Access Memory (RAM)

- Static RAM (SRAM)
- Dynamic RAM (DRAM)

1. Static RAM: SRAM stands for Static Random Access Memory. It is a type of semiconductor which is widely used in computing devices and microprocessors.

2. Dynamic RAM: DRAM stands for Dynamic Random Access Memory. It is made of Capacitors and has smaller data life span than Static RAM.

Advantages of Random Access Memory (RAM)

- **Speed: RAM is much faster than other types of storage, such as a hard drive or solid-state drive, which means that the computer can access the data stored in RAM+ more quickly.**
- **Flexibility: RAM is volatile memory, which means that the data stored in it can be easily modified or deleted. This makes it ideal for storing data that the computer is currently using or processing.**
- **Capacity: The capacity of RAM can be easily upgraded, which allows the computer to store more data in memory and thus improve performance.**
- **Power Management: RAM consumes less power compared to hard drives, and solid-state drives, which makes it an ideal memory for portable devices.**

Disadvantages of Random Access Memory (RAM)

- **Volatility: RAM is volatile memory, which means that the data stored in it is lost when the power is turned off. This can be a problem for important data that needs to be preserved, such as unsaved work or files that have not been backed up.**
- **Capacity: The capacity of RAM is limited, and although it can be upgraded, it may still not be sufficient for certain applications or tasks that require a lot of memory.**
- **Cost: RAM can be relatively expensive compared to other types of memory, such as hard drives or solid-state drives, which can make upgrading the memory of a computer or device more costly.**

Read-Only Memory (ROM)

Read Only Memory (ROM) is a type of computer memory that is used to permanently store data that does not need to be modified. ROM is non-volatile memory, which means that the data stored in it is retained even when the power is turned off. ROM is typically used to store the computer's BIOS (basic input/output system), which contains the instructions for booting the computer, as well as firmware for other hardware devices.

- Stores crucial information essential to operate the system, like the program essential to boot the computer.
- It is non-volatile.
- Always retains its data.
- Used in embedded systems or where the programming needs no change.
- Used in calculators and peripheral devices.
- ROM is further classified into four types- M *ROM* , *PROM* , *EPROM* , and *EEPROM* .

Types of Read-Only Memory (ROM)

1. PROM (Programmable Read-Only Memory)
2. EPROM (Erasable Programmable Read Only Memory)
3. EEPROM (Electrically Erasable Programmable Read Only Memory)
4. MROM (Mask Read Only Memory)

1. PROM (Programmable read-only memory): It can be programmed by the user. Once programmed, the data and instructions in it cannot be changed.

2. EPROM (Erasable Programmable read-only memory): It can be reprogrammed. To erase data from it, expose it to ultraviolet light. To reprogram it, erase all the previous data.

3. EEPROM (Electrically erasable programmable read-only memory): The data can be erased by applying an electric field, with no need for ultraviolet light. We can erase only portions of the chip.

4. MROM(Mask ROM): Mask ROM is a kind of read-only memory, that is masked off at the time of production. Like other types of ROM, mask ROM cannot enable the user to change the data stored in it. If it can, the process would be difficult or slow.

Advantages of Read Only Memory (ROM)

- **Non-volatility:** ROM is non-volatile memory, which means that the data stored in it is retained even when the power is turned off. This makes it ideal for storing data that does not need to be modified, such as the BIOS or firmware for other hardware device.
- **Reliability:** Because the data stored in ROM is not easily modified, it is less prone to corruption or errors than other types of memory.
- **Power Management:** ROM consumes less power compared to other types of memory, which makes it an ideal memory for portable devices.

Disadvantages of Read Only Memory (ROM)

- **Limited Flexibility:** ROM is read-only memory, which means that the data stored in it cannot be modified. This can be a problem for applications or firmware that need to be updated or modified.
- **Limited Capacity:** The capacity of ROM is typically limited, and upgrading it can be difficult or expensive.
- **Cost:** ROM can be relatively expensive compared to other types of memory, such as hard drives or solid-state drives, which can make upgrading the memory of a computer or device more costly.

Difference Between RAM and ROM

Parameter	RAM	ROM
Storage Type	Temporary Storage.	Permanent Storage.
Storage Capacity	Store data in MBs.	Store data in GBs.
Data Volatility	Volatile.	Non-volatile.
Usage	Used in normal operations.	Used for startup process of computer.
Data Writing Speed	Writing data is faster.	Writing data is slower.

*Main Memory

The main memory acts as the central storage unit in a computer system. It is a relatively large and fast memory which is used to store programs and data during the run time operations.

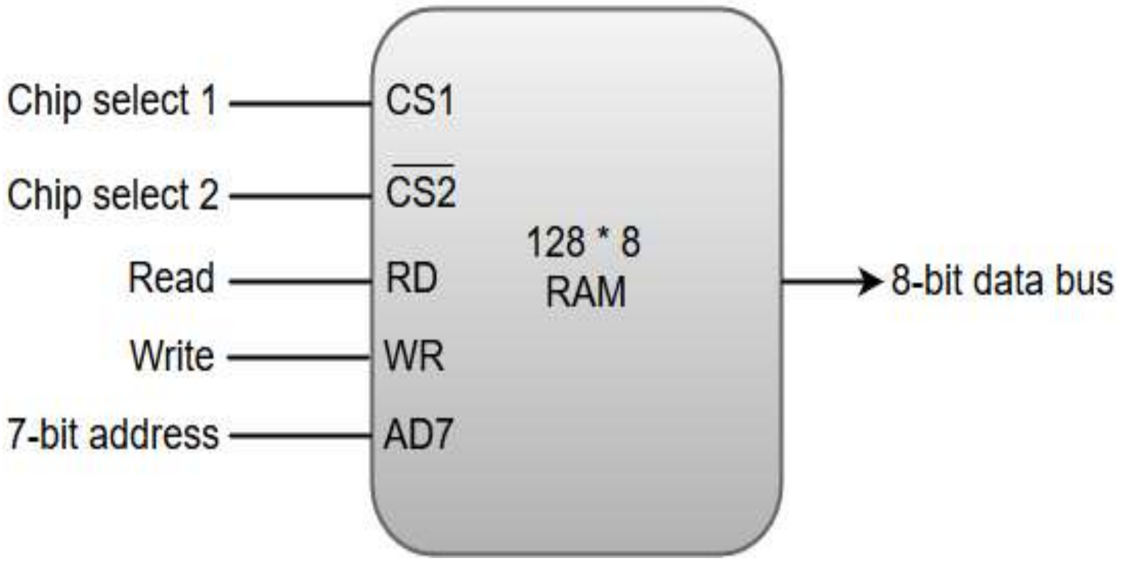
The primary technology used for the main memory is based on semiconductor integrated circuits. The integrated circuits for the main memory are classified into two major units.

1. RAM (Random Access Memory) integrated circuit chips
2. ROM (Read Only Memory) integrated circuit chips

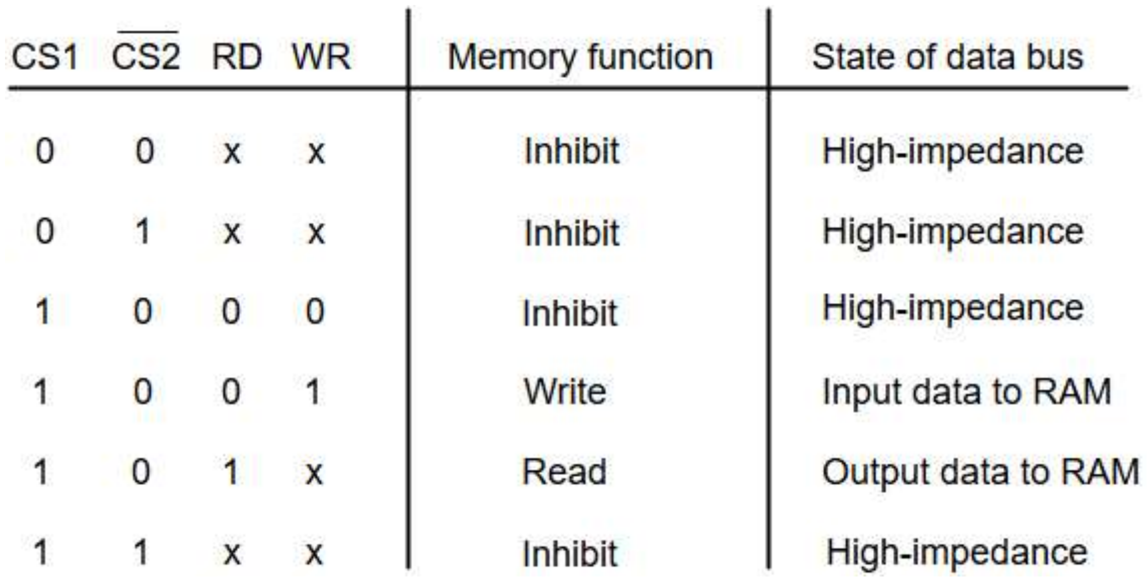
RAM integrated circuit chips

3. The RAM integrated circuit chips are further classified into two possible operating m
odes, **static** and **dynamic**.
4. The primary compositions of a static RAM are flip-flops that store the binary information. The nature of the stored information is volatile, i.e. it remains valid as long as power is applied to the system. The static RAM is easy to use and takes less time performing read and write operations as compared to dynamic RAM.
5. The dynamic RAM exhibits the binary information in the form of electric charges that are applied to capacitors. The capacitors are integrated inside the chip by MOS transistors. The dynamic RAM consumes less power and provides large storage capacity in a single memory chip.
6. RAM chips are available in a variety of sizes and are used as per the system requirement. The following block diagram demonstrates the chip

Typical RAM chip:



- A 128 * 8 RAM chip has a memory capacity of 128 words of eight bits (one byte) per word. This requires a 7-bit address and an 8-bit bidirectional data bus.

    - The 8-bit bidirectional data bus allows the transfer of data either from memory to CPU during a **read** operation or from CPU to memory during a **write** operation.
    - The **read** and **write** inputs specify the memory operation, and the two chip select (CS) control inputs are for enabling the chip only when the microprocessor selects it.
    - The bidirectional data bus is constructed using **three-state buffers**.
- The output generated by three-state buffers can be placed in one of the three possible states which include a signal equivalent to logic 1, a signal equal to logic 0, or a high-impedance state.
- The following function table specifies the operations of a 128 * 8 RAM chip.
- 

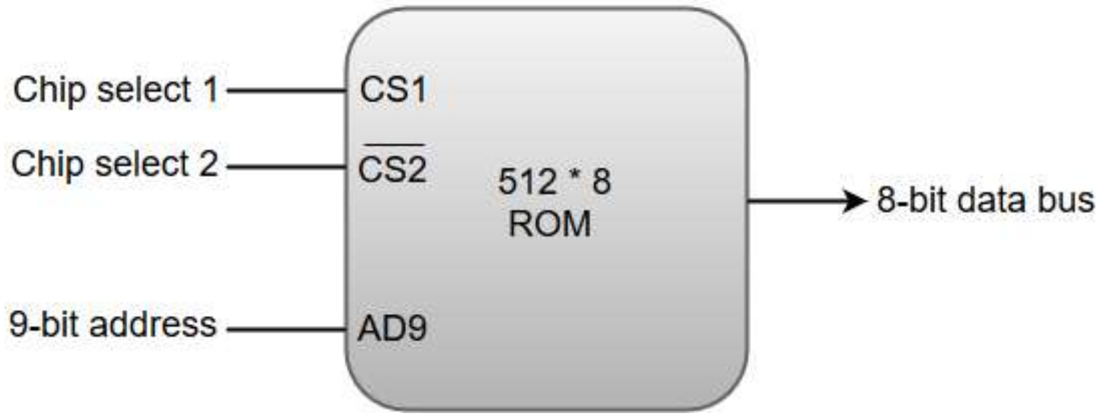
CS1	$\overline{CS2}$	RD	WR	Memory function	State of data bus
0	0	x	x	Inhibit	High-impedance
0	1	x	x	Inhibit	High-impedance
1	0	0	0	Inhibit	High-impedance
1	0	0	1	Write	Input data to RAM
1	0	1	x	Read	Output data to RAM
1	1	x	x	Inhibit	High-impedance

- From the functional table, we can conclude that the unit is in operation only when CS1 = 1 and CS2 = 0. The bar on top of the second select variable indicates that this input is enabled when it is equal to 0.
- ROM integrated circuit
- The primary component of the main memory is RAM integrated circuit chips, but a portion of memory may be constructed with ROM chips.
- A ROM memory is used for keeping programs and data that are permanently resident in the computer.
- Apart from the permanent storage of data, the ROM portion of main memory is needed for storing an initial program called a **bootstrap loader**. The

primary function of the **bootstrap loader** program is to start the computer software operating when power is turned on.

- ROM chips are also available in a variety of sizes and are also used as per the system requirement. The following block diagram demonstrates the chip interconnection in a 512 * 8 ROM chip.

Typical ROM chip:



- A ROM chip has a similar organization as a RAM chip. However, a ROM can only perform read operation; the data bus can only operate in an output mode.
- The 9-bit address lines in the ROM chip specify any one of the 512 bytes stored in it.
- The value for chip select 1 and chip select 2 must be 1 and 0 for the unit to operate. Otherwise, the data bus is said to be in a high-impedance state.

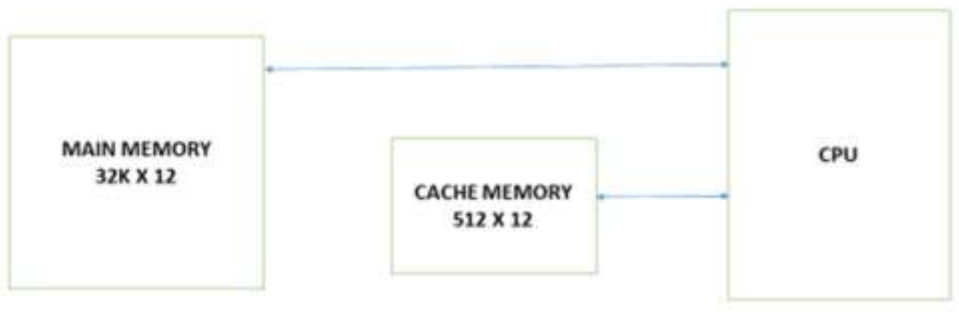
*Cache Memory

Introduction:

- When a program instruction is executed, the CPU repeatedly refers to the set of instructions in the memory.
- Every time a subroutine is called, its instructions are fetched from the memory.
- Over a short interval of time, the address generated by a program refers to few localized areas of memory repeatedly.

- That is ,only some portion of memory will be accessed repeatedly at a particular time while the remaining memory in the main memory is accessed less frequently.
- It takes more time for the CPU to access the main memory each and every time .
- If the active portions of the program and data are placed in a fast small memory, the average memory access time can be reduced, which reduces the total execution time of the program.
- Such a fast small is called *"Cache Memory".*
- Cache memory is placed between CPU and main memory.
- Cache memory access time is less than that of main memory by a factor 5 to 10.
- Cache is the fastest components in the memory hierarchy

Example of a cache memory:



Basic Operation of Cache memory:

- When the CPU needs to access some memory, first the cache is examined.
- If the word to be accessed is found in the cache memory, then it is read from the fast memory.

- If the word is not found in the cache memory, then it is read from the main memory.
- A block of words just accessed in the main memory is automatically transferred to cache memory.
- Cache memory contains a replica or the copy of the main memory.
- When the CPU refers to a memory and finds that word in the cache, it is called “*hit”.*
- When the word is not found in the cache and it is present in the main memory, then it is counted as “*miss”.*
- The ratio of no.of hits to the no.of total CPU references to memory(hit+ misses) is called as “*hit ratio*”.
- The performance of the cache memory is frequently measured in terms of hit ratio.
- Hit ratios are found to be more than 0.9.
- Hit ratio = (hit)/(hit+miss).

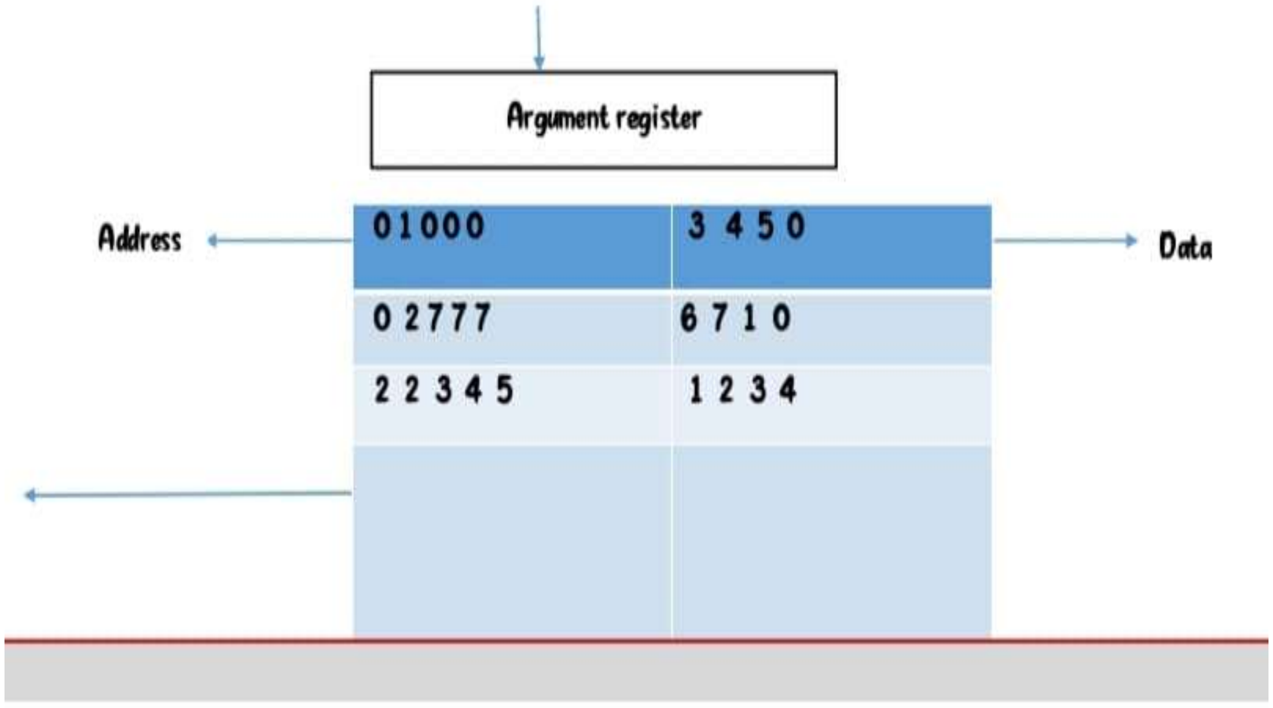
Mapping process:

- The transformation of data from main memory to cache memory is called “mapping process”.
- There are three types of mapping procedures in the organization of cache memory:

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

Associative Mapping:

- The fastest and most flexible cache organization uses an associative memory.
- The associative cache memory stores both the address and the content i.e., data of the memory word.



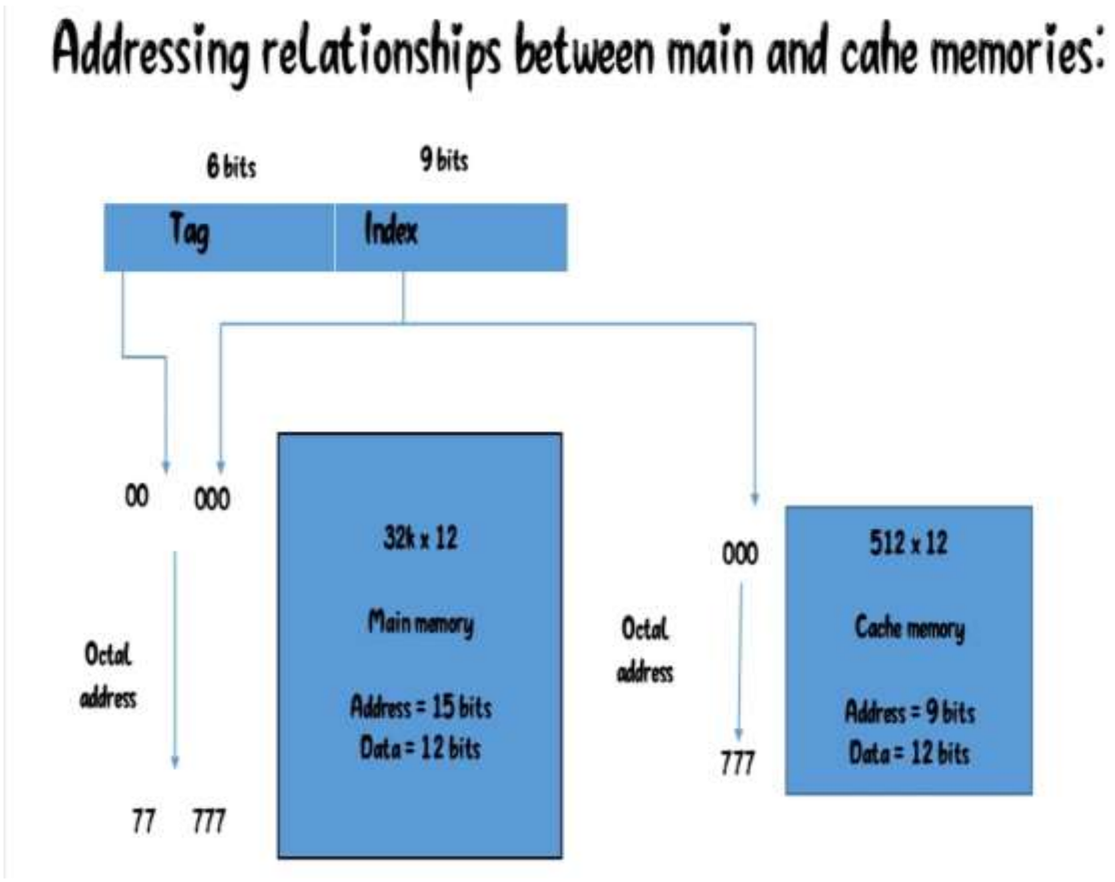
- The diagram shows three words presently stored in the cache.
- The address value is of 15 bits which is shown in 5 digit octal number.
- Its corresponding 12-bit word is show in 4-digit octal number.
- A CPU address of 15-bits is stored in the argument register and the associative memory is searched for the matching address .
- If the address is found, it is *hit* and the word is read into the CPU , if the address is not found, then it is miss and the word is read from the main memory and the address data pair is then transferred to the cache memory.
- If the cache is full, then the words in the cache are to be displaced to make room for new words.

  Direct Mapping:

- The drawback of Associative mapping is it is more expensive as it has an added logic associated with each cell.
- To overcome this drawback ,we move to Direct Mapping.
- In direct mapping we use the random access memory.
- To organize direct mapping, the CPU address of 15 bits is divided into two fields:

    1. the nine least significant bits constitute the *index* field.

    2. the remaining s



.

Direct mapping cache organization:

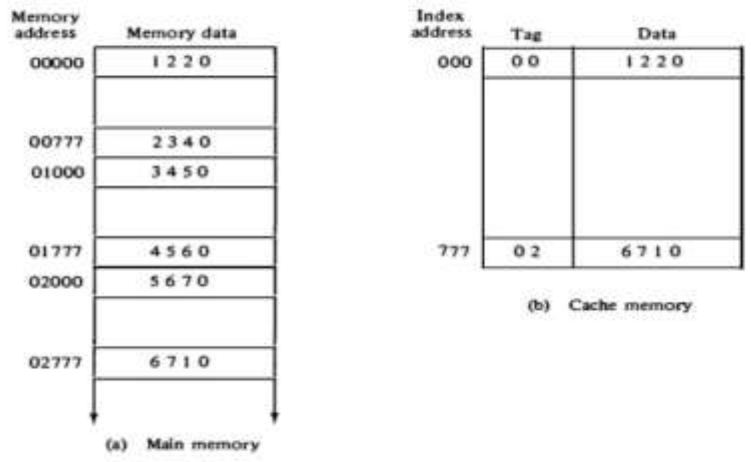
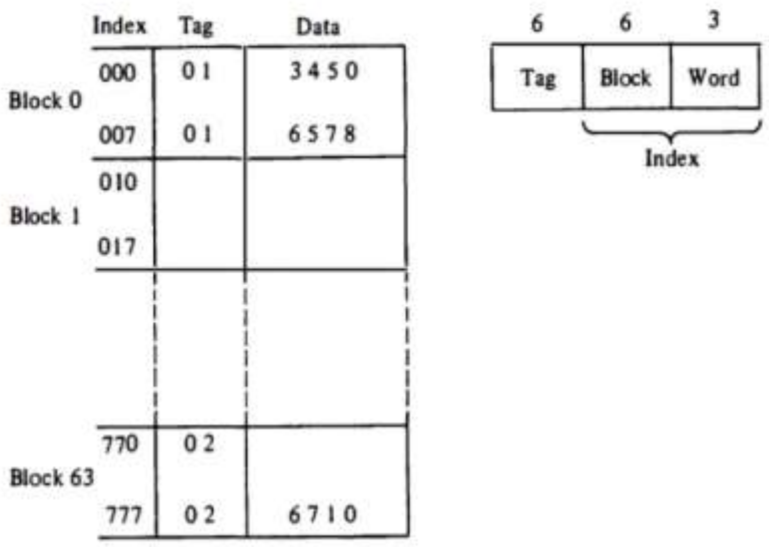


Figure 12-13 Direct mapping cache organization.

12

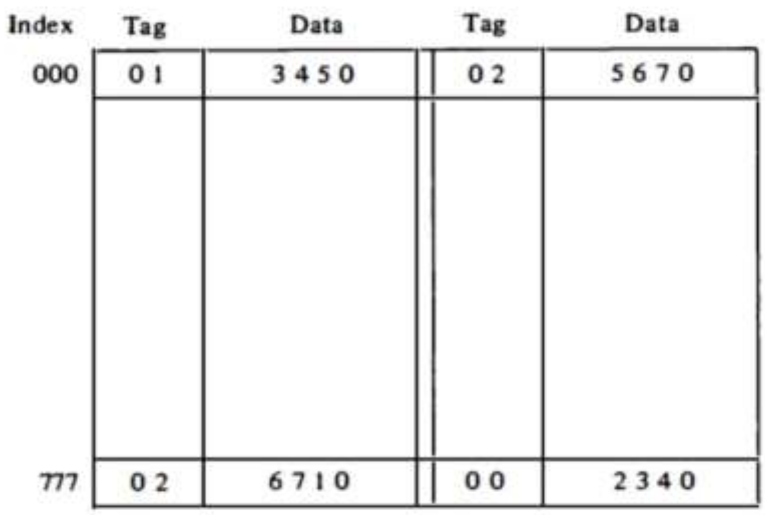


13

Figure 12-14 Direct mapping cache with block size of 8 words.

Set-Associative Mapping:

- The drawback of direct mapping is two words is two words with same index but with different tag values cannot reside in the cache memory for the same time.
- Set-Associative mapping is an improvement of direct mapping in which each word of cache can store two or more words of memory under the same index address.
- Each data word is stored together with its tag and the number of tag-data items in word of cache is said to form a set.



Index	Tag	Data	Tag	Data
000	0 1	3 4 5 0	0 2	5 6 7 0
777	0 2	6 7 1 0	0 0	2 3 4 0

Figure 12-15 Two-way set-associative mapping cache.

Writing into cache:

There are two methods:

1. Write-through method.

2.Write-back method.

1.Write-through method:

In this method , main memory is updated with every memory write operation and the cache memory is also updated in parallel if it contains that word in the specific address.

It has an advantage that main memory always contains the same data as that of the cache memory.

2.Write-Back method:

In this method only the cache memory is updated.

The location where the memory is updated is marked by a flag

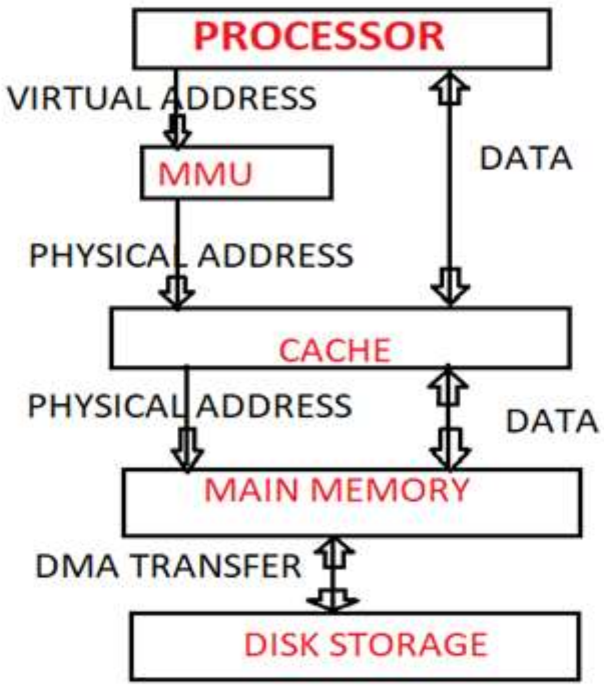
So that when the word is removed from the cache , it can be added to main memory easily.

The advantage of this method is we can update the memory several times easily during the program execution and when the word id removed from the cache the accurate copy is re-written into the main memory.

Cache Initialization:

- The cache memory is initialized when the power is supplied to the computer or when the main memory is loaded with a complete set of programs from the auxiliary memory.
- After initialization , ideally the cache has to be empty , but it instead it contains some invalid data.
- Therefore, a *"valid bit"* is included with each word in cache to check whether the word contains valid data or not.
- Firstly , all the valid bits are set to 0 when the cache is initialized.
- When a new word is loaded from main memory to cache , the valid bit is set to 1.
- 1 indicates valid data and 0 indicates invalid data.
- Virtual Memory
- The physical memory which we are using in our computer system is in megabytes not in gigabytes. If we try to run a program which is not completely fit into main memory, then, in this case, we try the parts of currently being executed are stored in main memory and remaining portion is stored in a secondary device such as hard disk.
- If this case occurs, so it's necessary that all parts of a program which are needed for execution are first brought into the main memory. When a new segment of a program is brought and memory is full, it must replace another segment already in the memory. So the techniques which are used to move program and data block into the main memory when they required for execution are called **virtual memory techniques**.

Virtual Memory Organization



- Programmers discern a larger memory which is allocated on the disk this memory is referred to as **virtual memory**. A programmer enjoys a huge virtual space to develop his or her program or software.
- Program or a processor references a data space which is independent of available physical memory space. The address issued by a processor is called a virtual address.
- The virtual address can be translated into the physical address by a combination of hardware and software components.
- If a virtual address is a part of the program in the physical memory then it can be accessed immediately and if the address is not in the main memory then its content will be brought into the main memory before it can be used.
- To improve the performance hardware are added to the system, these hardware units are known as Memory Management Unit (MMU).

- In the paging system, page table helps us to find the physical address form virtual address since the virtual address space is used to develop a process. So the work of the MMU is to translate the virtual address to physical address.

  Paging and Virtual Memory Address Translation

  The mechanism for reading a data from memory involves translation of virtual address to physical address. In this basically, the concept of paging is used for doing the translation. Following steps are there for address translations which are given below:

1. Virtual address space of the program is divided into fixed length units called, pages.
2. Each page will contain a block of words that will occupy the locations in the main memory. After this, each page will be mapped into the fixed location in the main memory called page frame.
3. The address generated by the processors to fetch the word memory can be divided into the two parts:

Virtual page number	Offset

4. In this case, high order bits are interpreted as virtual page number and these bits are used to fetch corresponding page frame number from the page table.
5. Low order bits of the address gives the offset and it specifies as the particular byte within in a page.
6. By adding this virtual page number to the contents of the content page table, the address of the corresponding entry in the page table is obtained.
7. Each entry in a page includes a control bit that describes a validity of a page, the status of the page, and whether the page has been modified.

   Advantages of Virtual Memory:

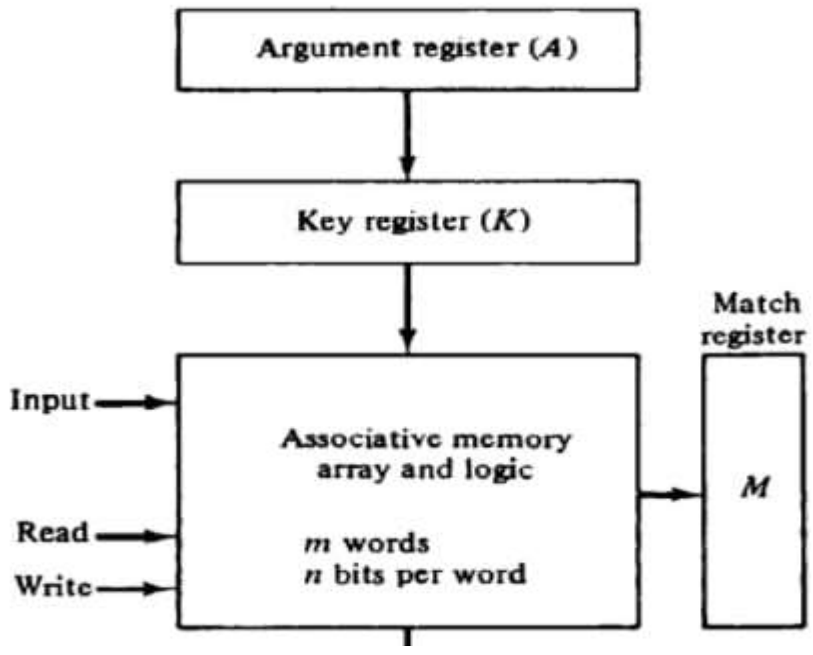
1. A process which is larger than the main memory, it will be executed because of the demand paging.
2. By the concept of a virtual memory, many processes can be maintained in the main memory.
3. It will allow greater programming level by using only less space for a particular process.

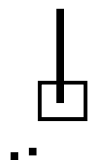
AUXILIARY MEMORY:

The time required to find an item stored in memory can be reduced considerably if stored data can be identified for access by the content of the data itself rather than by an address. A memory unit accessed by content is called an associative memory or content addressable memory (CAM). • CAM is accessed simultaneously and in parallel on the basis of data content rather than by specific address or location • Associative memory is more expensive than a RAM because each cell must have storage capability as well as logic circuits • Argument register –holds an external argument for content matching • Key register –mask for choosing a particular field or key in the argument word

Hardware Organization:

Figure 12-6 Block diagram of associative memory.

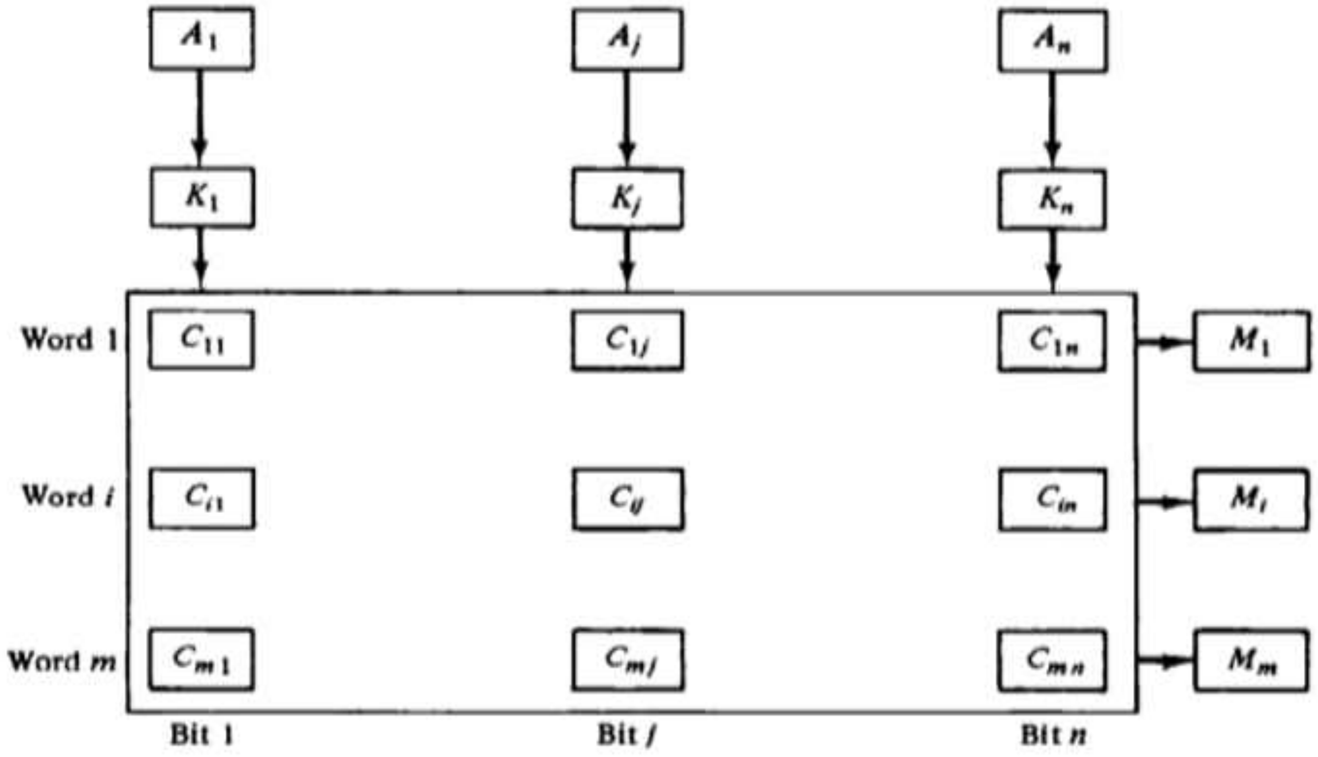




out put

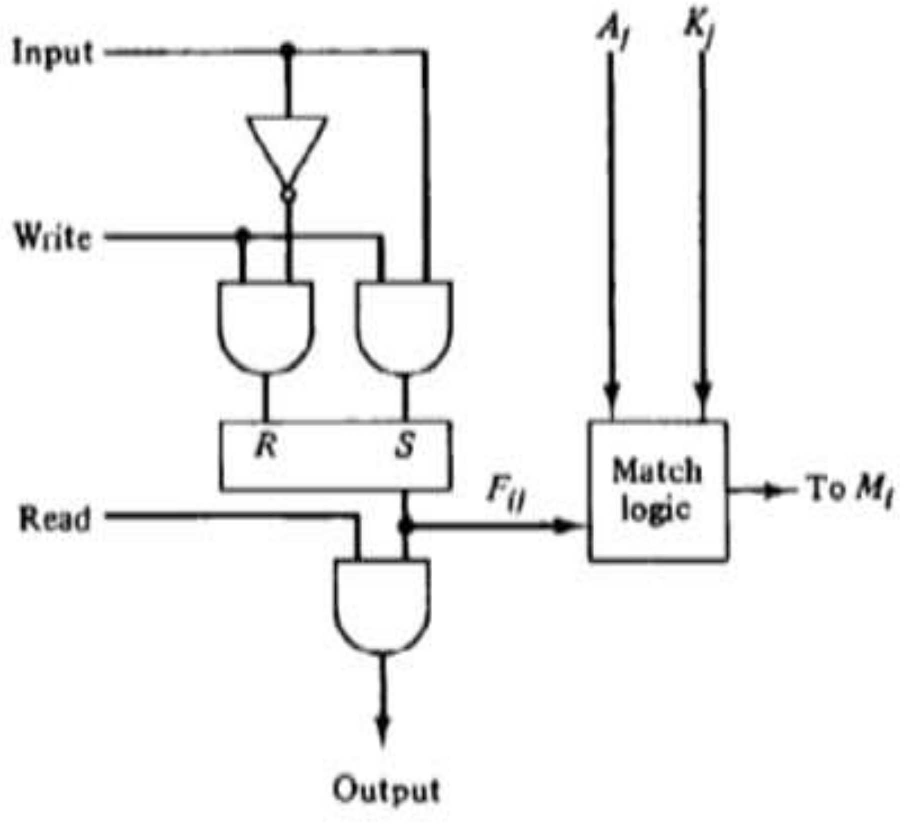
It consists of a memory array and logic for m words with n bits per word. The argument register A and key register K each have n bits, one for each bit of a word. The match register M has m bits, one for each memory word. Each word in memory is compared in parallel with the content of the argument register. The words that match the bits of the argument register set a corresponding bit in the match register. After the matching process, those bits in the match register that have been set bindicate the fact that their corresponding words have been matched. Reading is accomplished by a sequential access to memory for those words whose corresponding bits in the match register have been set.

Figure 12-7 Associative memory of m word, n cells per word.



The relation between the memory array and external registers in an associative memory is shown in Fig.12-7. The cells in the array are marked by the letter C with two subscripts. The first subscript gives the word number and second specifies the bit position in the word. Thus cell Cij is the cell for bit j in word i. A bit Aj in the argument register is compared with all the bits in column j of the array provided that kj =1.This is done for all columns j=1,2,….n. If a match occurs between all the unmasked bits of the argument and the bits in word I, the corresponding bit Mi in the match register is set to 1. If one or more unmasked bits of the argument and the word do not match, Mi is cleared to 0

Figure 12-8 One cell of associative memory.



It consists of flip-flop storage element Fij and the circuits for reading, writing, and matching the cell. The input bit is transferred into the storage cell during a write operation. The bit stored is read out during a read operation. The match logic compares the content of the storage cell with corresponding unmasked bit of the argument and provides an output for the decision logic that sets the bit in Mi.

Match Logic

The match logic for each word can be derived from the comparison algorithm for two binary numbers. First, neglect the key bits and compare the argument in A with the bits stored in the cells of the words. Word i is equal to the argument in A if Aj=F ijfor j=1,2,…..,n. Two bits are equal if they are both 1 or both 0. The equality of two bits can be expressed logically by the Boolean function xj=Aj Fij + Aj

'Fij ' where xj = 1 if the pair of bits in position j are equal;otherwise , xj =0. For a word i is equal to the argument in A we must have all xj variables equal to 1. This is the condition for setting the corresponding match bit Mi to 1. The Boolean function for this condition is

Mi = x1 x2 x3…… xn

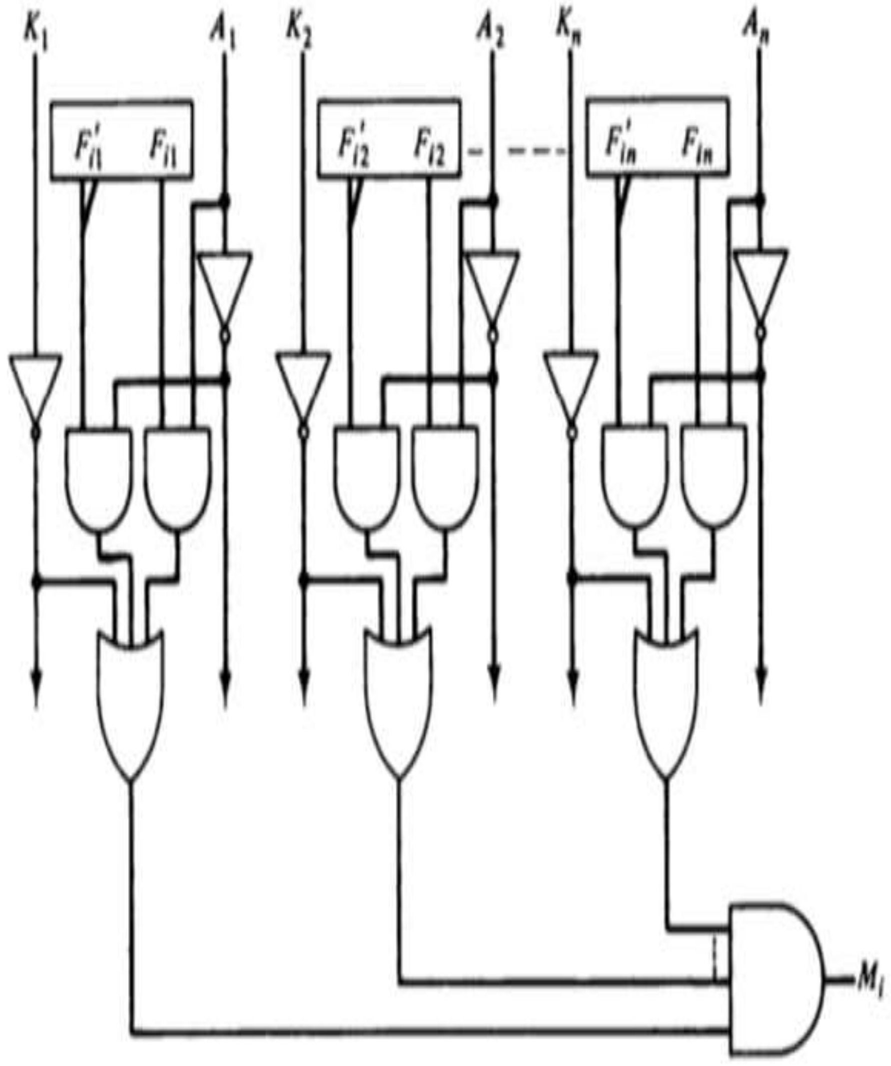


Figure 12-9 Match logic for one word of associative memory.

Each cell requires two AND gate and one OR gate. The inverters for A and K are needed once for each column and are used for all bits in the column. The output of all OR gates in the cells of the same word go to the input of a common AND gate to generate the match signal for Mi. Mi will be logic 1 if a match occurs and 0 if no match occur

UNIT 4

2 MARKS

1. What is memory?

A computer is an electronic device and that accepts data, processes on that data, and gives the desired output. It performs programmed computation with accuracy and speed. Or in other words, the computer takes data as input and stores the data/instructions in the memory (use them when required). After processes the

data, it converts into information. Finally, gives the output.

2. Types of hard Dis?

*PATA (Parallel ATA): An older interface standard for connecting hard drives to computers.

*SATA (Serial ATA): A more modern interface standard, commonly used for connecting HDDs and SSDs.

*SCSI (Small Computer System Interface): An older interface standard used for high-performance applications.

*SAS (Serial Attached SCSI): A more modern interface standard based on SCSI, used for high-performance applications.

3.What is mean by semiconductor ram memory?

A type of electronic memory known as semiconductor memory **stores digital data by making use of semiconductor materials**, most commonly silicon. Data is stored in binary format in this memory, with "1s" and "0s" representing electrical charges

4.Explain about Ram?

RAM, which stands for Random Access Memory, is the temporary storage space within your computer. It acts like your computer's short-term memory, holding data and instructions that the central processing unit (CPU) needs to access quickly for ongoing tasks. Unlike permanent storage devices like hard disk drives (HDDs) and solid-state drives (SSDs), RAM is volatile, meaning it loses all its data when you shut down your computer.

5.How the memory management system works?

The term memory can be defined as a collection of data in a specific format. It is used to store instructions and process data. The memory comprises a large array or group of words or bytes, each with its own location. The primary purpose of a computer system is to execute programs. These programs, along with the information they access, should be in the main memory during execution. The CPU fetches instructions from memory according to the value of the program counter.

6.How the memory is organized in a computer?

- Simultaneous Access Memory Organization: all the levels are directly connected to CPU.
- Hierarchical Access Memory Organization: all the levels are connected in hierarchical fashion.
- Five hierarchies based on the speed as well as use: registers, cache, main memory, magnetic discs, and magnetic tapes.

7.Explain about primary and secondary memory?

Secondary Memory (eg: SSD, HDD) can hold our files and programs for long term and our data is safe but less fast than primary.

8.What is mean by Rom?

Memory plays a crucial role in how devices operate, and one of the most important types is Read-Only Memory (ROM). Unlike RAM (Random Access Memory), which loses its data when the power is turned off, ROM is designed to store essential information permanently

9.What is mean by PCI express flash?.

PCIe, or Peripheral Component Interconnect Express, is a standard for connecting a computer's motherboard with peripherals such as graphics cards, sound cards, and solid-state drives. A PCIe card plugs into a corresponding slot on the motherboard, with types ranging from x1 to x16, indicating the number of data lanes available. More lanes mean higher data transfer rates, akin to more lanes on a highway enabling faster traffic flow. The standard is maintained by the Peripheral Component Interconnect Special Interest Group (PCI-SIG).

10.How many types of memories and what are they?

There are **four general types of memories**:

- Sensory memory
- Short-term memory
- Working memory
- Long-term memory

UNIT-5:- INPUT OUTPUT ORGANIZATION

PART-A (10 MARKS)

1. Explain about input devices with example and diagram?

Sol:- The electromagnetic devices that accept data or a set of instructions from the outside world and then translate that data into machine-readable and understandable form are known as input devices. Computer input devices serve as an interface between the outside world and the computer for proper communication. When the users enter data using various input devices, the data can be saved in computer memory for further processing and preparation. Using the output devices, the intended and calculated results can be acquired when the processing and handling are completed. An input device transmits data to a computer and allows you to communicate with it and control it.

Different Types of Input Devices

Various types of Input Devices are,

Keyboard

For entering data into a computer, the keyboard is the most common and commonly used input device. It contains various keys for entering letters, numbers, and characters. Although there are some additional keys for completing various activities, the keyboard layout is identical to that of a standard typewriter. It is generally available in two different sizes 84 keys or 101/102 keys and for Windows and the Internet, it is also available with 104 keys or 108 keys. It is connected to a computer system with the help of a USB or a Bluetooth device.

The keys on the keyboard are

- **Numeric Keys:** These keys are used to enter numeric data and move the cursor. It is typically made up of 17 keys.
- **Keyboard Shortcuts:** These keys include the letter keys (A-Z) and the number keys (09).
- **Control Keys:** The pointer and the screen are controlled by these keys. It comes with four directional arrow keys. Control keys include Home, End, Insert, Alternate(Alt), Delete, Control(Ctrl), and Escape.
- **Special Keys:** Enter, Shift, Caps Lock, NumLk, Tab, and Print Screen are some of the special function keys on the keyboard.
- **Function Keys:** The 12 keys from F1 to F12 are on the topmost row of the keyboard.

Generally, the keyboard is of three types:

- QWERTY Keyboard
- AZERTY Keyboard
- DVORAK Keyboard

Characteristics of Keyboard

- The keyboard has various functions keys for a different purpose
- Instead of using the mouse, we can utilize the arrow keys on the keyboard to do the same purpose as the mouse.
- The main keyboard, cursor keys, numeric keypad, and function keys are the four primary components of a keyboard.

- Keyboards are more affordable.

backspace
tab
return
caps lock
shift
shift
new
pg up
pg dn
alt gr
send
special
new

Mouse

The mouse is the most used pointing device. While clicking and dragging, the mouse moves a little cursor across the screen. If you let off of the mouse, the cursor will come to a halt. You must move the mouse for the computer to move; it will not move on its own. As a result, it’s a device that accepts input. Or we can say that a mouse is an input device that allows you to control the coordinates and movement of the on-screen cursor/pointer by moving the mouse on a flat surface. The left mouse button can be used to pick or move items, while the right mouse button displays additional menus when clicked. It was invented in 1963 by Douglas C. Engelbart.

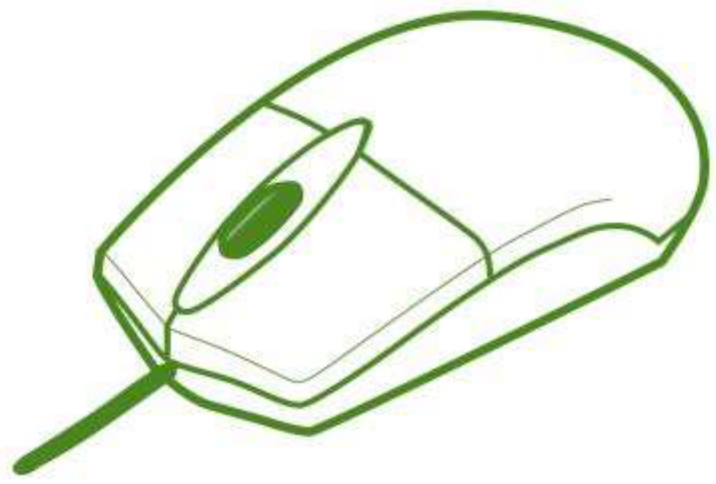
Generally, the mouse is of four types

- Trackball Mouse
- Mechanical Mouse
- Optical Mouse
- Wireless Mouse

Characteristics of Mouse

- A mouse is used to move the cursor on the screen in the desired direction.
- A mouse allows users to choose files, folders, or multiple files or text or, all at once.
- Hover over any object with the mouse pointer.

- A mouse can be used to open a file, folder, etc. You must first move your pointer to a file, folder, and then double-click on it to open or execute.



Joystick

A pointing device used to move the cursor around the screen is the joystick. Both the bottom and top ends of the stick have a spherical ball affixed to them. A socket contains the lower spherical ball. You can adjust the joystick in all directions. Trackballs became quite popular in laptops and PCs since they fit neatly inside the case and take up less room when in use. They are more precise and long-lasting than a mouse, which is why they are still utilized. It is invented by C.B.Mirick.

Characteristics of Joystick

- It's utilized to regulate the cursor's position across a display screen.
- It's utilized in computer games to move the characters and symbols around.
- It commonly features one or more push buttons, the condition of which can be controlled by the computer as well.

Light Pen

A light pen is a pointing device that has the appearance of a pen. It can be used to draw on the monitor screen or to pick a menu item. In a small tube, a photocell and an optical system are housed. The photocell sensor element determines the screen location and sends a signal to the CPU when the tip of a light pen is moved across a monitor screen while the pen button is pressed.

Characteristics of Light Pen

- When drawing graphics, a light pen comes in very handy.
- Objects on the display screen are selected with a light pen.

Scanner

A scanner is a type of input device that works in the same way as a photocopier. It's used when there's data on paper that needs to be transferred to the computer's hard disc for further processing. The scanner collects images from the source and translates them to a digital version that can be saved on the hard disks. These graphics can be changed before they are printed.

Generally, the scanner is of five types:

- Flatbed Scanner
- Handheld Scanner
- Sheetfed Scanner
- Drum Scanner
- Photo Scanner

Characteristics of Scanner

- You can scan film negatives via a scanner if there is a transparent media adaptor.
- A scanner may also scan low-quality or non-standard-weight paper.
- The scanners are adaptable, allowing you to scan a wide range of items regardless of their size. You can scan small items as well as large documents if you can locate them.

OCR

OCR stands for Optical Character Recognition in its full form. OCR is a computer reading technique that reads numbers, characters, and symbols. OCR is a technique for recognizing text in documents that have been scanned into digital form. Optical character recognition (OCR) refers to a device that reads printed text. Character by character, OCR scans the text, converts it to a machine-readable code, and saves it into the memory of the system. OCR also functions as a scanner, scanning documents, photos, images, and handwritten text and storing the information in memory, which may then be compared to previously stored data.

Characteristics of OCR

- The technology offers a complete solution for form processing and document capture.
- It has capabilities for defining shapes, scanning, image pre-processing, and identification.

Barcode Reader

A bar code reader is a device that reads bar-coded data (data that is represented by light and dark lines). To label things, number books, and so on, bar-coded data is often utilized. It could be a standalone scanner or a component of one. A barcode reader is a device that reads barcodes and extracts data from them. The code bar is used to read the bar code printed on any goods. By impacting light beams on barcode lines, a barcode reader identifies existing data in barcodes.

Characteristics of Barcode Reader

- When a card is inserted, auto-start barcode scanners begin scanning immediately.
- Reading indicators give the user confirms that the card has been swiped correctly.
- It’s simple to use, simply hold your phone up to the code and scan it.

Web Camera

A webcam is an input device since it records a video image of the scene in front of it. It can either be incorporated inside the computer (for example, a laptop) or connected via USB. A webcam is a small digital video camera that is connected to a computer. Because it can capture pictures and record video, it's also known as a web camera.

Characteristics of Web Camera

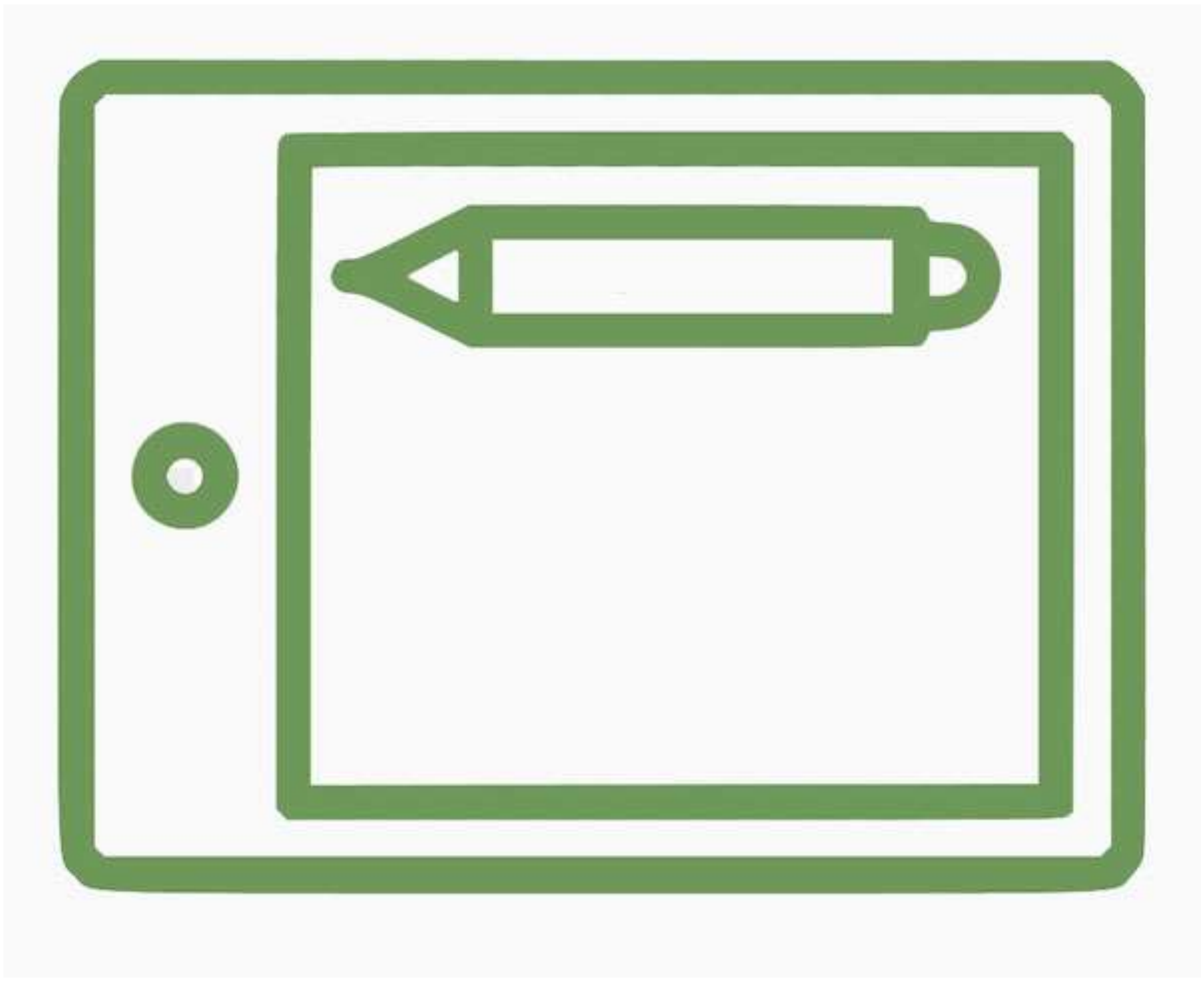
- Webcams are used to allow individuals to see one other while chatting online. This is formally referred to as 'teleconferencing'.
- Because webcams can take a picture only if movement is detected in the scene in front of them, they are commonly utilized in burglar alarms and other security systems.
- Hundreds of webcams can be found all around the world, each pointing to a fascinating scene such as the exterior view of a facility in the Arctic or Niagara Falls. The webcam is connected to a computer that regularly sends an image to an internet server. After that, people connect to the server to view the most recent image.

Graphic Tablets

A graphics tablet, also known as a digitizing tablet, is a computer input device that allows users to draw drawings and graphics by hand, much like they would with a pencil and paper. A graphics tablet is a flat surface on which the user can draw a picture with the help of an attached stylus, which is a pen-like drawing device.

Characteristics of a Graphics Tablets

- The graphics tablet is a pressure-sensitive tablet that is controlled by a pen.
- Drawing, writing, inserting, etc. can be done with the pen.
- It provides more precision and the ability to monitor (than a touchscreen).

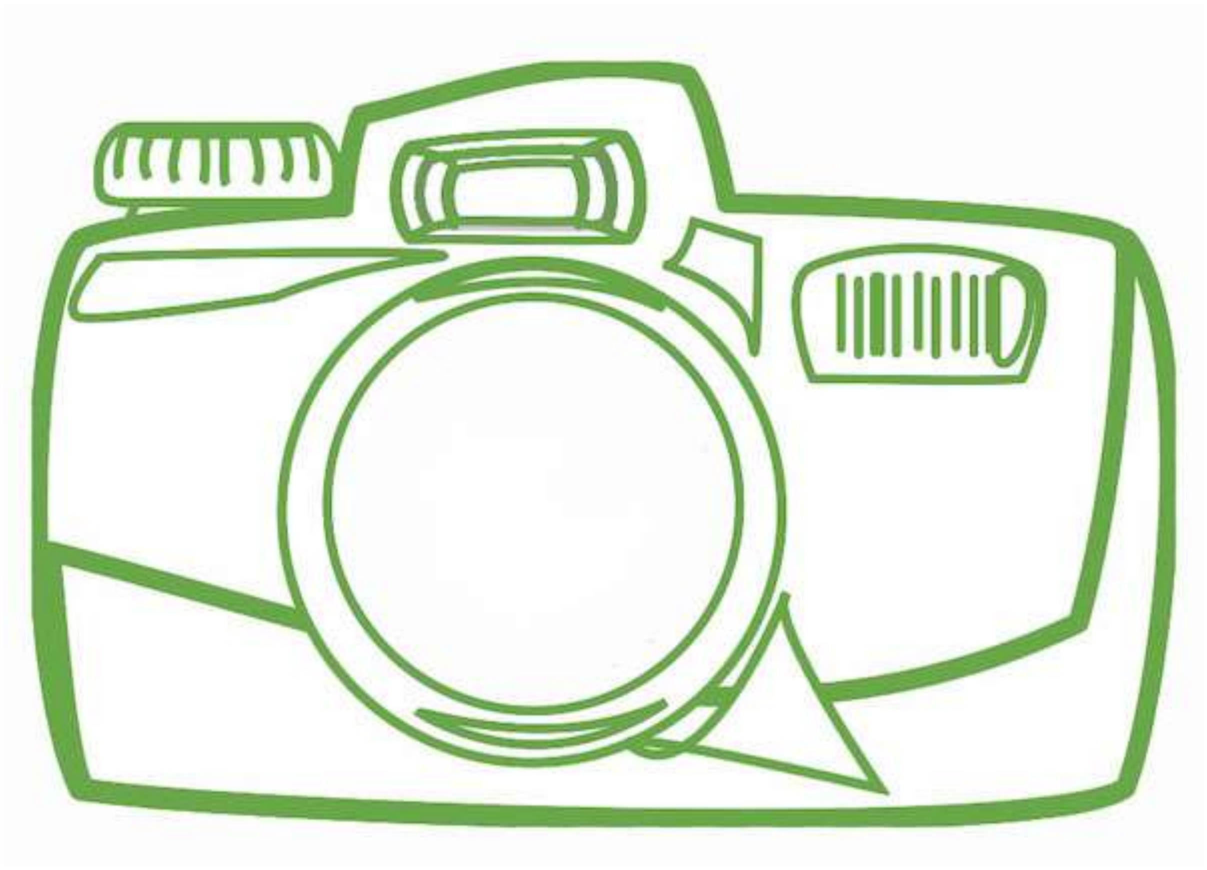


Digital Cameras

Digital camera is a device that takes photographs as input. Images are saved on memory cards as data. It comes with an LCD display that allows users to view and review photographs. A digital camera contains photosensors that record the light that enters into the camera lens. So, when the light strikes the photosensors, they return the electric current and this electric current is used to create images.

Characteristics of Digital Camera

- Users can immediately examine images and movies on the LCD screen.
- All the photos can be stored in the storage device.
- Users can select and choose the images they want to develop.
- Easily portable & takes less space.



Touchscreen

A touchscreen is a type of input device that allows users to interact with a digital display by directly touching the screen's surface. It enables the user to perform various actions, such as selecting options, typing on a virtual keyboard, drawing, or manipulating objects, by physically touching the screen.

Characteristics of Touchscreen

- With a touchscreen, users can directly interact with the content displayed on the screen, eliminating the need for additional input devices like a mouse or keyboard.
- Touchscreens use various technologies to detect and respond to touch inputs, including capacitive, resistive, infrared, and surface acoustic wave (SAW) technologies.
- Many modern touchscreens support multi-touch gestures, allowing users to use multiple fingers or gestures for more advanced interactions, such as pinch-to-zoom and rotating objects.

- Touchscreens often support various gestures, including tapping, swiping, pinching, and rotating, to control and manipulate on-screen elements.

2. Extreme briefly about output devices with diagram and example?

Sol:- Any peripheral that accepts data from a computer and prints, projects, or reproduces it is known as an output device. The output may be audio, video, hard copy – printed paper, etc. Output devices convert the computer data to human understandable form. We give input to the computer using input devices and the computer performs operations on the data and displays the output to the user using the output device.

How Does an Output Device Work?

In order to show the output, an output device uses a signal it receives from the computer to accomplish a task. An output device's fundamental operation is listed below as an illustration.

- If you enter 'Hi Geeks' on a computer keyboard (input device), the computer receives the signal.
- Once the input has been processed by the computer, an output device—a monitor—is signalled.

- The display (output) of the 'Hi Geeks' on the screen occurs once the monitor receives the signal.
- Another example of an output device is a printer, which might print that 'Hi Geeks' if it were supported.

  If the computer was working and had no output device attached, you could still type 'Hi Geeks' on the keyboard and it would still be processed. Without an output device, you couldn't see what happened or verify the input, though.

  Different Types of Output Devices

  The various output devices are as below:

  1. Monitor

  A computer's principal output device is a monitor, often known as a visual display unit (VDU). It displays the processed data like text, images, videos, audios, etc. It makes images by arranging microscopic dots in a rectangular pattern, known as pixels. The sharpness of an image is determined by the number of pixels. There are two types of monitor viewing screens:

- **Cathode-Ray Tube (CRT):** This type of monitor is based on a cathode ray tube. In which the cathode ray tube generates a beam of electrons with the help of electron guns they strike on the inner surface of phosphorescent of the screen to generate images. The CRT monitor holds millions of phosphorus dotes in three different colors, i.e., red, blue, and green. These dots glow when the beam struck on them and create an image. The main parts of the CRT monitor are the electron gun, fluorescent screen, glass envelope, deflection plate assembly, and base.
- **Display on a Flat Panel Monitor with a Cathode-Ray Tube (CRT):** A flat-panel display is a type of video display with less volume, weight, and power consumption than a CRT. They can be put on the wrist or hang on the wall. Calculators, video games, monitors, laptop, and graphical displays all use flat-panel displays.

- **Plasma Monitor:** It is also a flat panel display but it is based on plasma display technology. In a plasma monitor, a small cell is present in between two glass surfaces and these cells contain a solution of noble gases and mercury. So when the electricity supply on the gas present in the cell converts into plasma and produces UV light that creates an image. It is much better than an LCD monitor. The resolution of this monitor is also high up to 1920 x 1920. It has a good contrast ratio, high refresh rate, etc.

Characteristics of Monitor:

- **Resolution pixels:** Pixels are the smallest element of any image
- **Size:** The size of the monitor is diagonal measurement of a desktop screen is typically 14 to 25 inches.
- **Refresh Rate:** Total number of times per second that an image on a display is repainted or refreshed.

Monitor

2. Printer

Printers are information output devices that allow you to print data on paper. Or in other words, it is an output device that creates a hard copy of the processed data or information. Printers are divided into two categories:

- **Impact Printer:** In impact printers, characters are printed on the ribbon, which is then smashed on the paper. Or we can say that such type of printer uses a print head or hammer to print the data on the paper. Here to print the paper the hammer or print head strikes an ink ribbon against the paper and the character starts printing. Some of the types of impact printers are:

  - Dot matrix printer
  - Daisy wheel printer
  - Line printer
  - Chain printer

- **Impact printers have the following characteristics:**
  - Extremely low consumable costs.
  - Fairly noisy
  - It’s perfect for large-scale printing because of its inexpensive cost.
  - Physical contact with the paper is required to form an image.
- **Non-Impact Printers:** Non-impact printers print characters without the use of a ribbon. These printers are often known as page printers because they print a full page at a time. Some of the types of non-impact printers are:
  - Laser printer
  - Inkjet printer
- **Non-impact printers have the following characteristics:**
  - Quicker.
  - They don’t produce much noise.
  - Superior quality.
  - Supports a wide range of fonts and character sizes.

Printer

3. Plotter

A plotter is a device that prints high-quality graphics in a variety of color formats. It works in a similar way to a printer, although it has more advanced features. It is used to print large maps, architectural drawings, large-format printing, and create pictures, 3D postcards, advertising signs, charts, and various designs of the internal structure of building machines, as well as create pictures, 3D postcards, advertising signs, charts, and various designs of the internal structure of building machines.

Characteristics of Plotter:

- Large size prints can be taken via plotters.
- It is slow and expensive.

Plotter

4. Projector

A projector is a device that allows users to project their output onto a large area, such as a screen or a wall. It can be used to project the output of a computer and other devices onto a screen. It magnifies texts, photos, and movies using light and lenses. As a result, it's an excellent output device for giving presentations or teaching big groups of people.

Characteristics of Projector:

- They are lightweight, and one person can easily take them out of the box, connect them, and hang an image on the wall.
- Projectors can be the most cost-effective option for large-screen video in your home.
- A small projector mounted on a back shelf or bookcase, or mounted on the ceiling, takes up no area on the floor. It is barely visible when it is not in use.

Projector

5. Speakers

Speakers are connected to computers to allow sound to be output. For the working of speakers, sound cards are required. From simple two-speaker output devices to surround-sound multi-channel sets, speakers come in a variety of shapes and sizes. They take audio input from the computer's sound card and output sound waves as audio output.

Characteristics of Speakers:

- Speakers are available in a wide range of qualities and prices.
- Small, plastic computer speakers with low sound quality are often included with computer systems.

Speaker

6. Headphones

To hear the sound, use earbuds with your computer, laptop, or smartphone. It enables you to hear the sound without causing any inconvenience to others. To translate electronic signals into sounds without causing inconvenience to others. They can be wired or wireless and can be connected to computers, laptops, mobile phones, etc. They are connected with the devices via Bluetooth.

Characteristics of Headphones:

- Stereo phones and headsets are other names for them.
- Earphones or earbuds are the names for the in-ear variants.
- The term headset denotes a combination of headphones and a microphone used for two-way communication, such as using a telephone.

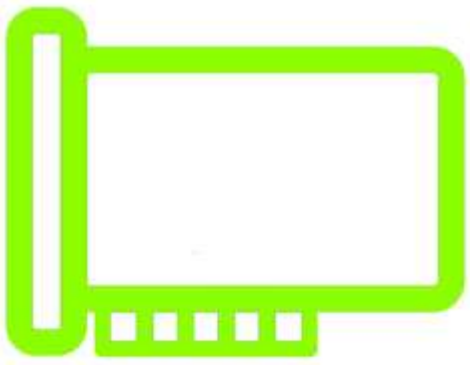
Heaphone

7. Sound Card

Sound cards are computer output devices that are inserted into the computer. A sound card, either external or internal, is required to produce sound on any computer (built-in). An external sound card enables for better overall sound generation and is required for wide and clear sound recording, as well as sound without noise and interference.

Characteristics of Sound Card:

- To listen speakers or headphones, to play games, watch movies, listen to music, or use audio and video conferencing, we use an internal sound card.
- Frequency is a sound card parameter that represents the number of signals the card processes per unit of time. The frequency is expressed in hertz. The frequency of most sound cards is 96 or 192 kHz.
- Synthesizers and a variety of electronic musical instruments, such as drums and keyboards, can be connected to your computer using a sound card with standard musical instrument digital interface (MIDI) connections.



Sound Card

8. Video Card

An extension card via which a computer can transfer graphical data to a video display device like a TV, or monitor. It processes photos and video, as well as other functions that the CPU generally does. As they have a good processing capability and video RAM, Gamers utilize video cards.

Characteristics of Video Card:

- Heat sinks are required for video cards with high performance as they generate a lot of heat.
- Also known as graphics card and require software installation in addition to the hardware.
- When working with huge files, video cards supply a significant quantity of video-only memory that frees up CPU resources, allowing the system to run more effectively.

Video Card

9. Speech Synthesizer

A speech synthesizer is a computerized device that takes in data, interprets it, and generates audible words. It might be a computer card, a box connected by a cable, or software that works with the computer's sound card.

Characteristics of speech synthesizer:

- Any text, predetermined input can be translated into audible speech.

- For people who are unable to talk or have impaired vision, it can provide digital verbal communication.
- It takes in data, interprets it, and generates sound output.

Speech Synthesizer

10. GPS

The Global Positioning System (GPS) is a radio-based satellite navigation system that uses radio signals to pinpoint a specific position. The sender sends a radio signal to satellites, which collect data such as time, location, speed, and other variables and deliver it to the reception computer for analysis. Because this processed data can be evaluated to obtain information, it is considered as an output device.

Characteristics of GPS:

- GPS satellites constantly communicate their position and time.
- Solar storms, high storm cover, and other factors impair GPS equipment.
- The Global Positioning System (GPS) is based on the mathematical idea of trilateration.
- The GPS works independently of telephonic or internet reception and does not need the user to send any data, however, to improve accuracy both technologies can be used.

GPS

3. Explain about accessing input and output devices?

Sol:- **Accessing I/O**

• The important components of any computer system are CPU, memory and I/O devices (peripherals). The CPU fetches instructions (opcodes and operands/data) from memory, processes them and stores results in memory. The other components of the computer system (I/O devices) may be loosely called the Input/Output system.

• The main function of I/O system is to transfer information between CPU or memory and the outside world.

• The important point to be noted here is, I/O devices (peripherals) cannot be connected directly to the system bus. The reasons are discussed here.

1. A variety of peripherals with different methods of operation are available. So it would be impractical to incorporate the necessary logic within the CPU to control a range of devices.

2. The data transfer rate of peripherals is often much slower than that of the memory or CPU. So it is impractical to use the high speed system bus to communicate directly with the peripherals.

3. Generally, the peripherals used in a computer system have different data formats and word lengths than that of CPU used in it.

• So to overcome all these difficulties, it is necessary to use a module in between system bus and peripherals, called I/O module or I/O system, or I/O interface.

The functions performed by an I/O interface are:

1. Handle data transfer between much slower peripherals and CPU or memory.

2. Handle data transfer between CPU or memory and peripherals having different data formats and word lengths.

3. Match signal levels of different I/O protocols with computer signal levels.

4. Provides necessary driving capabilities - sinking and sourcing currents.

Requirements of I/O System

• The I/O system if nothing but the hardware required to connect an I/O device to the bus. It is also called I/O interface. The major requirements of an I/O interface are :

1. Control and timing

2. Processor communication

3. Device communication

4. Data buffering

5. Error detection

• The important blocks necessary in any I/O interface are shown in Fig. 8.6.1.

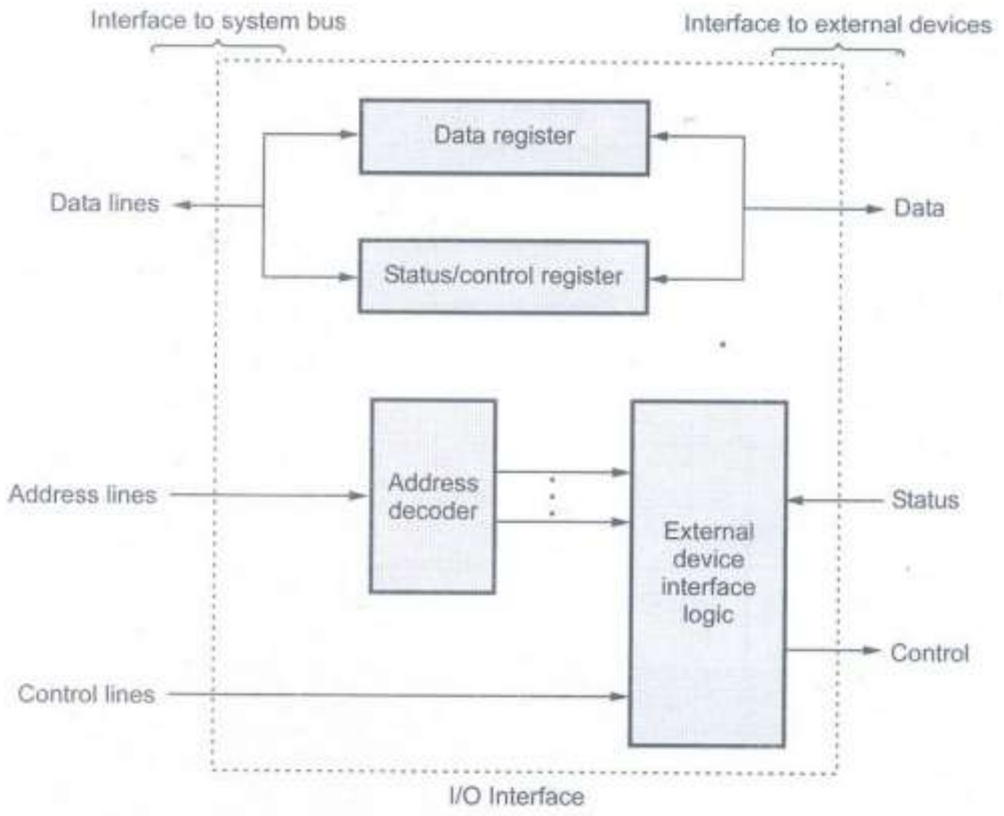


Fig. 8.6.1 Block diagram of I/O interface

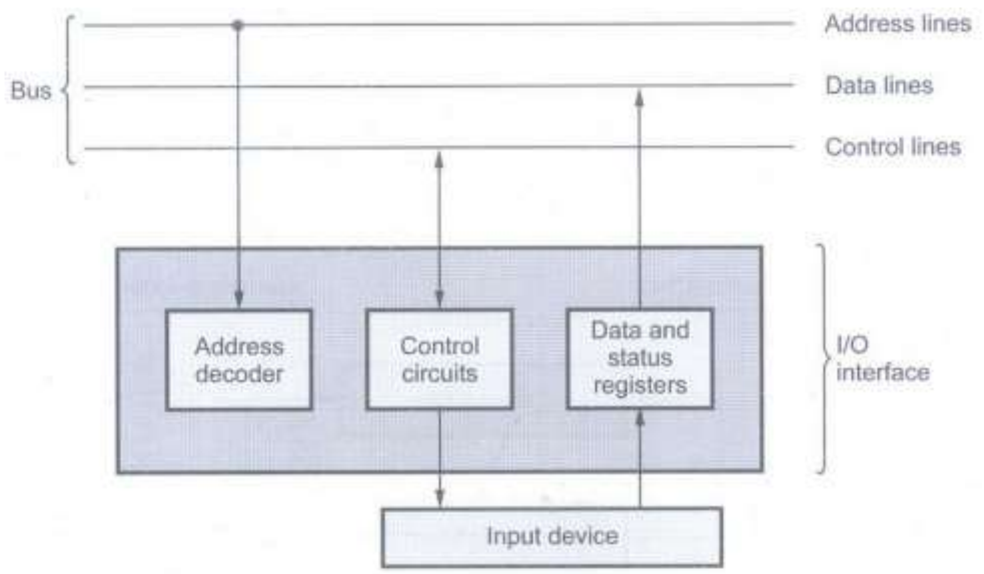
• As shown in the Fig. 8.6.1, I/O interface consists of data register, status/control register, address decoder and external device interface logic.

• The data register holds the data being transferred to or from the processor.

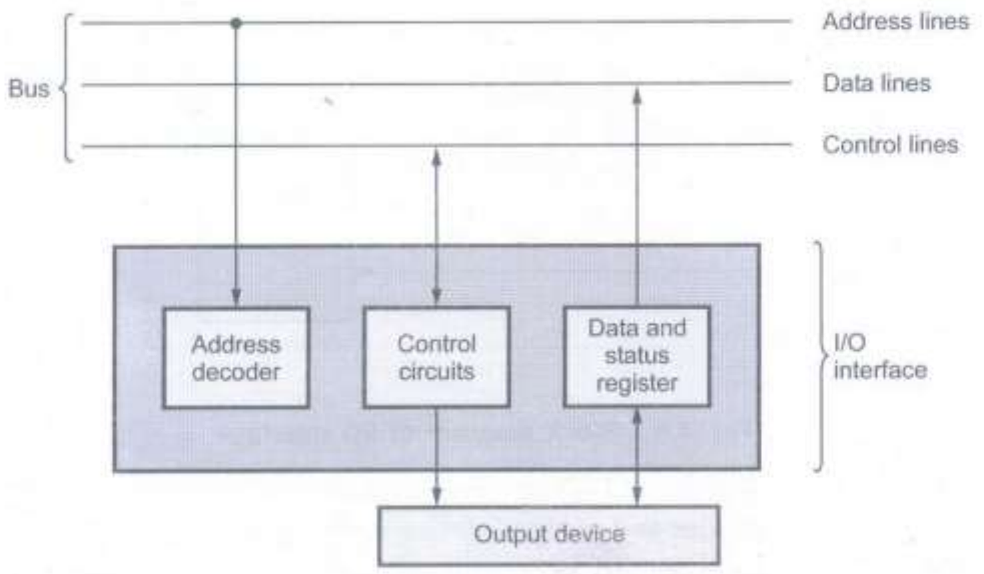
• The status/control register contains information relevant to the operation of the I/O device. Both data and status/control registers are connected to the data bus.

• Address lines drive the address decoder. The address decoder enables the device to recognize its address when address appears on the address lines.

• The external device interface logic accepts inputs from address decoder, processor control lines and status signal from the I/O device and generates control signals to control the direction and speed of data transfer between processor and I/O devices.



(a) I/O interface for input device



(b) I/O interface for output device

Fig. 8.6.2

• The Fig. 8.6.2 shows the I/O interface for input device and output device. Here, for simplicity block schematic of I/O interface is shown instead of detail connections.

• The address decoder enables the device when its address appears on the address lines.

• The data register holds the data being transferred to or from the processor.

• The status register contains information relevant to the operation of the I/O device.

• Both the data and status registers are assigned with unique addresses and they are connected to the data bus.

I/O Interfacing Techniques

I/O devices can be interfaced to a computer system I/O in two ways, which are called interfacing techniques,

• Memory mapped I/O

• I/O mapped I/O

Memory mapped I/O

• In this technique, the total memory address space is partitioned and part of this space is devoted to I/O addressing as shown in Fig. 8.6.3.

• When this technique is used, a memory reference instruction that causes data to be fetched from or stored at address specified, automatically becomes an I/O instruction if that address is made the address of an I/O port.

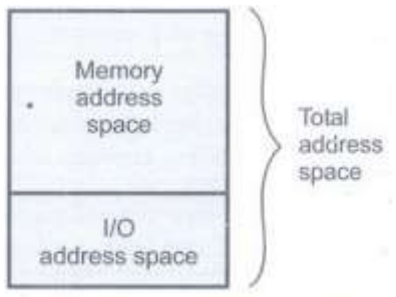


Fig. 8.6.3 Address space

Advantage

• The usual memory related instructions are used for I/O related operations. The special I/O instructions are not required.

Disadvantage

• The memory address space is reduced.

I/O mapped I/O

• If we do not want to reduce the memory address space, we allot a different I/O address space, apart from total memory space which is called I/O mapped I/O technique as shown in Fig. 8.6.4.

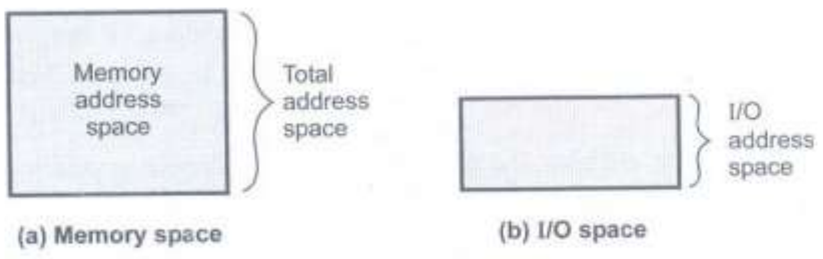


Fig. 8.6.4 Address space

Advantage

• The advantage is that the full memory address space is available.

Disadvantage

• The memory related instructions do not work. Therefore, processor can only use this mode if it has special instructions for I/O related operations such as I/O read, I/O write.

Memory Mapped I/O, I/O Mapped I/O Comparison

Sr. No.	Memory mapped I/O	I/O mapped I/O
1.	Memory and I/O share the entire address range of processor.	Processor provides separate address range for memory and I/O devices.
2.	Usually, processor provides more address lines for accessing memory. Therefore more decoding is required control signals.	Usually, processor provides less address lines for accessing I/O. Therefore, less decoding is required.
3.	Memory control signals are used to control read and write I/O operations.	I/O control signals are used to control read and write I/O operations.

Sr. No.	Memory bus	I/O bus
1.	Memory address bus shares entire address range.	I/O bus shares only I/O address range.
2.	Memory address bus width is greater than I/O address bus width	I/O address bus width is smaller than memory address bus width.
3.	Memory bus includes data bus, address bus and control signals to access memory.	I/O bus includes data bus, address bus and control signals to access I/O.

Types of Data Transfer Techniques

• In I/O data transfer, the system requires the transfer of data between external circuitry and the processor. Different ways of I/O data transfer are:

1. Program controlled I/O or polling control.

2. Interrupt program controlled I/O or interrupt driven I/O.

3. Hardware controlled I/O.

4. I/O controlled by handshake signals.

Program controlled I/O or polling control

• In program controlled I/O, the transfer of data is completely under the control of the processor program. This means that the data transfer takes place only when an I/O transfer instructions executed. In most of the cases it is necessary to check whether the device is ready for data transfer or not. To check this, processor polls the status bit associated with the I/O device.

Interrupt program controlled I/O or interrupt driven I/O

• In interrupt program controlled approach, when a peripheral is ready to transfer data, it sends an interrupt signal to the processor. This indicates that the I/O data transfer is initiated by the external I/O device.

• When interrupted, the processor stops the execution of the program and transfers the program control to an interrupt service routine.

• This interrupt service routine performs the data transfer.

• After the data transfer, it returns control to the main program at the point it was interrupted.

Hardware controlled I/O

• To increase the speed of data transfer between processors memory and I/O, the hardware controlled I/O is used. It is commonly referred to as Direct Memory Access (DMA). The hardware which controls this data transfer is commonly known as DMA controller.

• The DMA controller sends a HOLD signal to the processor to initiate data transfer. In response to HOLD signal, processor releases its data, address and control buses to the DMA controller. Then the data transfer is controlled at high speed by the DMA controller without the intervention of the processor.

• After data transfer, DMA controller sends low on the HOLD pin, which gives the control of data, address, and control buses back to the processor.

• This type of data transfer is used for large data transfers.

I/O Control by handshake signals

• The handshake signals are used to ensure the readiness of the I/O device and to synchronize the timing of the data transfer. In this data transfer, the status of handshaking signals are checked between the processor and an I/O device and when both are ready, the actual data is transferred.

4. Explain about interrupts processor with examples?

Sol:- The interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high-priority process requiring interruption of the current working process. In I/O devices one of the bus control lines is dedicated for this purpose and is called the *Interrupt Service Routine (ISR).*

When a device raises an interrupt at let's say process i,e., the processor first completes the execution of instruction i. Then it loads the Program Counter (PC) with the address of the first instruction of the ISR. Before loading the Program Counter with the address, the address of the interrupted instruction is moved to a temporary location. Therefore, after handling the interrupt the processor can continue with process i+1.

While the processor is handling the interrupts, it must inform the device that its request has been recognized so that it stops sending the interrupt request signal. Also, saving the registers so that the interrupted process can be restored in the future, increases the delay between the time an interrupt is received and the start of the execution of the ISR. This is called Interrupt Latency.

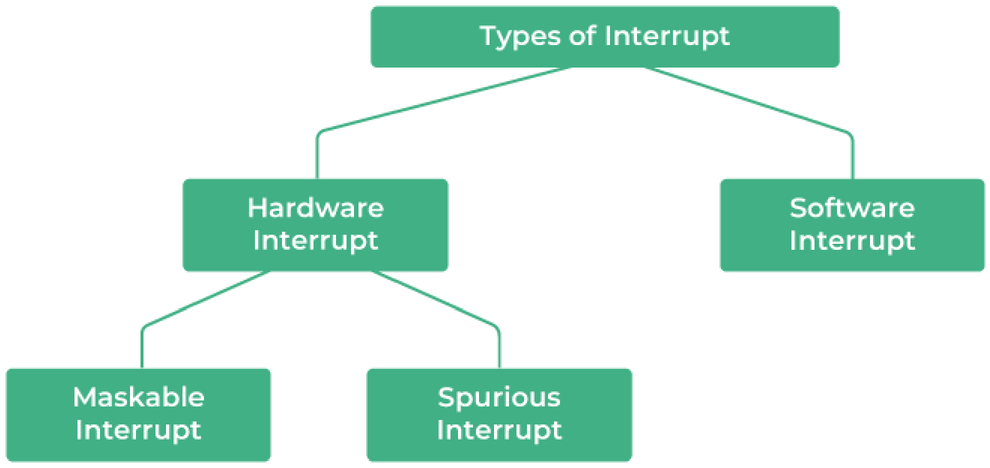
Types of Interrupt

Event-related software or hardware can trigger the issuance of interrupt signals. These fall into one of two categories: software interrupts or hardware interrupts.

1. Software Interrupts

A sort of interrupt called a software interrupt is one that is produced by software or a system as opposed to hardware. Traps and exceptions are other names for software interruptions. They serve as a signal for the operating system or a system service to carry out a certain function or respond to an error condition. Generally, software interrupts occur as a result of specific instructions being used or exceptions in the operation. In our system, software interrupts often occur when system calls are made. In contrast to the fork() system call, which also generates a software interrupt, division by zero throws an exception that results in the software interrupt.

A particular instruction known as an "interrupt instruction" is used to create software interrupts. When the interrupt instruction is used, the processor stops what it is doing and switches over to a particular interrupt handler code. The interrupt handler routine completes the required work or handles any errors before handing back control to the interrupted application.



Types of Interrupt

2. Hardware Interrupts

In a hardware interrupt, all the devices are connected to the Interrupt Request Line. A single request line is used for all the n devices. To request an interrupt, a device closes its associated switch. When a device requests an interrupt, the value of INTR is the logical OR of the requests from individual devices.

Hardware interrupts are further divided into two types of interrupt

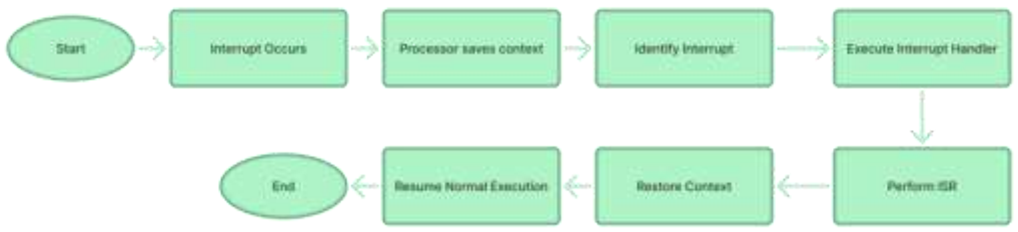
- **Maskable Interrupt:** Hardware interrupts can be selectively enabled and disabled thanks to an inbuilt interrupt mask register that is commonly found in processors. A bit in the mask register corresponds to each interrupt signal; on some systems, the interrupt is enabled when the bit is set and disabled when the bit is clear, but on other systems, the interrupt is deactivated when the bit is set.
- **Spurious Interrupt:** A hardware interrupt for which there is no source is known as a spurious interrupt. This phenomenon might also be referred to as phantom or ghost interrupts. When a wired-OR interrupt circuit is connected to a level-sensitive processor input, spurious interruptions are typically an issue. When a system performs badly, it could be challenging to locate these interruptions.

Sequences of Events Involved in Handling an IRQ(Interrupt Request)

- Devices raise an IRQ.
- The processor interrupts the program currently being executed.
- The device is informed that its request has been recognized and the device deactivates the request signal.
- The requested action is performed.
- An interrupt is enabled and the interrupted program is resumed.

Flowchart of Interrupt Handling Mechanism

The Image below depicts the flowchart of interrupt handling mechanism



Interrupt Handling Mechanism

- Step 1:- Any time that an interrupt is raised, it may either be an I/O interrupt or a system interrupt.
- Step 2:- The current state comprising registers and the program counter is then stored in order to conserve the state of the process.
- Step 3:- The current interrupt and its handler is identified through the interrupt vector table in the processor.
- Step 4:- This control now shifts to the interrupt handler, which is a function located in the kernel space.
- Step 5:- Specific tasks are performed by Interrupt Service Routine (ISR) which are essential to manage interrupt.
- Step 6:- The status from the previous session is retrieved so as to build on the process from that point.
- Step 7:- The control is then shifted back to the other process that was pending and the normal process continues.

Managing Multiple Devices

When more than one device raises an interrupt request signal, then additional information is needed to decide which device to be considered first. The following methods are used to decide which device to select: Polling, Vectored Interrupts, and Interrupt Nesting. These are explained below.

- **Polling:** In polling, the first device encountered with the IRQ bit set is the device that is to be serviced first. Appropriate ISR is called to service the same. It is easy to implement but a lot of time is wasted by interrogating the IRQ bit of all devices.

- **Vectored Interrupts:** In vectored interrupts, a device requesting an interrupt identifies itself directly by sending a special code to the processor over the bus. This enables the processor to identify the device that generated the interrupt. The special code can be the starting address of the ISR or where the ISR is located in memory and is called the interrupt vector.
- **Interrupt Nesting:** In this method, the I/O device is organized in a priority structure. Therefore, an interrupt request from a higher-priority device is recognized whereas a request from a lower-priority device is not. The processor accepts interrupts only from devices/processes having priority.

  Processors' priority is encoded in a few bits of PS (Process Status register). It can be changed by program instructions that are written into the PS. The processor is in supervised mode only while executing OS routines. It switches to user mode before executing application programs.

  Interrupt Priority Schemes

  Interrupt priority schemes are used in microprocessors and microcontrollers to manage multiple interrupt requests (IRQs). These schemes ensure that more urgent tasks are processed before less important ones, making them essential for real-time systems and efficient interrupt handling.

  Types of Interrupt Priority Schemes

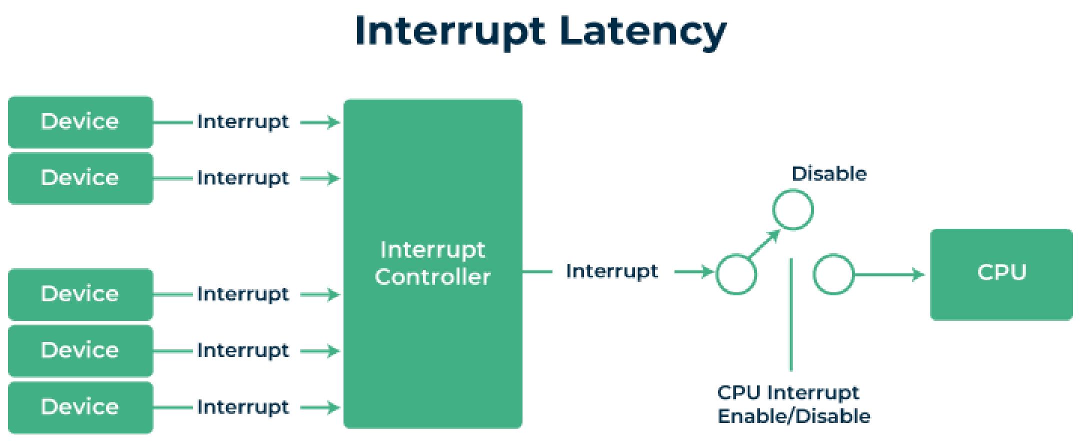
- **Fixed Priority Scheme:** In this scheme, each interrupt has a predetermined priority level. The interrupt with the highest priority is handled first. If two interrupts occur simultaneously, the one with the higher priority is serviced. **Example**: Interrupt A (priority 1) is serviced before Interrupt B (priority 2).
- **Dynamic Priority Scheme**: In a dynamic priority scheme, the priority of an interrupt can change based on system conditions. This helps prioritize real-time or critical tasks over others. **Example**: A system may increase the

priority of a sensor interrupt based on its importance at a particular moment.

- **Vectored Interrupt Scheme**: Here, each interrupt has a specific memory address (vector). The processor jumps to this address to handle the interrupt, and the interrupt with the highest priority is processed first. **Example**: The system uses memory addresses to quickly handle the most urgent interrupts.
- **Priority Masking**: This scheme allows lower-priority interrupts to be temporarily disabled, ensuring that high-priority interrupts are handled immediately without delay. **Example**: If Interrupt A is more critical than Interrupt B, Interrupt B may be masked until Interrupt A is processed.
- **Round-Robin Priority Scheme**: In this scheme, interrupts are processed in a cyclic order, ensuring each interrupt gets handled fairly, especially when all interrupts have the same priority. **Example**:Interrupts A, B, and C are handled in a round-robin manner.

What is Interrupt Latency?

The amount of time between the generation of an interrupt and its handling is known as interrupt latency. The number of created interrupts, the number of enabled interruptions, the number of interrupts that may be handled, and the time required to handle each interrupt all affect interrupt latency. When the device generating the interrupt needs a specific length of time to generate the interrupt, interrupt latency is required. For instance, if a printer is printing paper, the computer needs to stop the printing program and wait for the document to finish printing. The interrupt latency is the amount of time the computer has to stop the program from operating.



Interrupt Latency

How CPU React when Interrupt Occurs?

- **Interrupt Detection:** The CPU continuously video displays unit interrupt lines or alerts from diverse resources, consisting of hardware gadgets or software program commands, to hit upon interrupt requests.
- **Interrupt Acknowledgment:** Upon detecting an interrupt request, the CPU acknowledges the interrupt using sending an acknowledgment sign to the interrupting device or software program.
- **Interrupt Handling:** The CPU identifies the form of interrupt primarily based on its supply, together with a hardware interrupt from a device or a software interrupt from a training. It then seems the cope with the corresponding interrupt handler habitual within the interrupt vector desk.
- **Context Saving:** Before moving manipulate to the interrupt handler ordinary, the CPU saves the present-day execution context, inclusive of the program counter (PC), processor state, and any applicable sign-in contents, onto the stack or in the devoted garage.
- **Transfer Control:** The CPU transfers manipulation to the interrupt handler ordinary with the aid of placing this system counter (PC) to the address of the handler habitual retrieved from the interrupt vector desk.
- **Interrupt Servicing:** The interrupt handler habitual executes to carrier the interrupt. It plays responsibilities to interrupt, such as reading facts from a

device, processing enter/output operations, or coping with a software program request.

Triggering Methods

Every interrupt signal input is intended to be activated by a certain signal edge (level change) or a logic signal level. Level-sensitive inputs make constant requests for processor attention as long as they are treated with a specific logic level (high or low). Edge-sensitive inputs are responsive to signal edges; a service request will latch on to a specific (rising or falling) edge. When the interrupt handler runs, the CPU resets the latch.

- **Level-Trigger:** The interrupt signal must be held at its specific active logic level (high or low) to request a level-triggered interrupt. A level-triggered interrupt is triggered when a device drives the signal to the active level and maintains it there. When the CPU instructs it to do so, usually after the device has been serviced, it denies the signal.
- **Edge-Trigger:** An interrupt that is caused by a level change on the interrupt line—either a rising or lowering edge—is known as an edge-triggered interrupt (low to high). A pulse is driven onto the line and released to its inactive state by a device that wishes to indicate an interrupt. It can be necessary to use specialized hardware to detect the pulse if polled I/O is unable to pick it up due to its short duration.

Benefits of Interrupt

- **Real-time Responsiveness:** Interrupts permit a system to reply promptly to outside events or signals, permitting real-time processing.
- **Efficient Resource usage:** Interrupt-driven structures are more efficient than system that depend on busy-waiting or polling strategies. Instead of continuously checking for the incidence of event, interrupts permit the processor to remain idle until an event occurs, conserving processing energy and lowering energy intake.
- **Multitasking and Concurrency:** Interrupts allow multitasking with the aid of allowing a processor to address multiple tasks concurrently.

- **Improved system Throughput:** By coping with occasions asynchronously, interrupts allow a device to overlap computation with I/O operations or other responsibilities, maximizing system throughput and universal overall performance.

Conclusion

- The events known as interrupts alert the CPU to handle the request.
- Both software and hardware can create interruptions.
- Maskable and non-maskable interrupts are the two categories of hardware interrupts.
- Generally speaking, exceptions and special instructions like fork() are what trigger software interrupts.
- After completing the service request and handling the interruption, the CPU continues the process where it left off.

5. Explain about PCI Express?

Sol:- **PCIe** stands for **Peripheral Component Interconnect express**. It is an interface standard that is used to connect high-speed components. PCIe is available in a different physical configuration which includes x1, x4, x8, x16, x32. The motherboard has a number of PCIe slots to connect different components such as GPU(or video cards or graphics cards), WI-FI cards, SSD (Solid-state drive). Different motherboards have different types of PCIe slots.

Generation of PCIe

Till now six generations of PCIe have been introduced in the market i.e PCIe 1.0, PCIe 2.0, PCIe 3.0, PCIe 4.0, PCIe 5.0, PCIe 6.0 out of these only first four have been debuted in the market. PCIe 4.0 was first introduced in 2019 by AMD Ryzen 3000-series CPUs.

History

Arapaho Work Group (AWG), initially consisted of Intel engineers, later expanded to include industry partners, draw this standard. First PCIe was named as High-Speed Interconnect (HSI), then renamed to 3GIO (3rd generation I/O) and finally renamed to PCIe.

- **PCIe 1.0a:** It was introduced by PCI-SIG in year 2003. It has a per-lane data rate of 250 MB/s and a transfer rate of 2.5 Giga transfers per second (GT/s).
- **PCIe 1.1:** It was introduced by PCI-SIG in year 2005. It has more clarification and improvement over PCIe 1.0a but per-lane data rate and transfer rate was unchanged.
- **PCIe 2.0:** It was introduced by PCI-SIG in year 2007. It doubled the per-lane data rate and transfer rate compared to PCIe 1.0. It has a per-lane data rate of 500MB/s instead of 250 MB/s and a transfer rate of 5GT/s instead of 2.5 Giga transfers per second (GT/s). PCIe 2.0 slots provides backward compatibility with PCIe 1.x cards.

- **PCIe 3.0:** It was introduced by PCI-SIG in November 2010, after multiple delays. In August 2007, PCI-SIG announced backward compatible with existing PCI Express implementations and a bit rate of 8 Giga transfers per second (GT/s) for PCI Express 3.0. PCI-SIG also announced, a delay in release until Q2 2010 for the final specification for PCI Express 3.0. A number of optimizations for enhanced signaling and data integrity, including transmitter and receiver equalization, PLL improvements, clock data recovery, and channel enhancements for currently supported topologies were added in PCI Express 3.0 specification It has a per-lane data rate of 984.6MB/s instead of 500MB/s (as in PCIe 2.0) and a transfer rate of 8GT/s instead of 5 GT/s (as in PCIe 2.0).
- **PCIe 4.0:** It was introduced by PCI-SIG on November 29, 2011. It doubled the per-lane data rate and transfer rate compared to PCIe 3.0. It has a per-lane data rate of 1969MB/s instead of 984.6MB/s (as in PCIe 3.0) and a transfer rate of 16GT/s instead of 8 GT/s (as in PCIe 3.0). PCIe 4.0 provides full backward and forward compatibility.
- **PCIe 5.0:** PCI Express 5.0 preliminary specification was introduces by PCI-SIG in JUNE, 2017. In a 16-lane configuration Bandwidth was expected to increase to 32 GT/s, yielding 63 GB/s in each direction. The draft was expected to be standardized in 2019. Final PCI-Express 5.0 specification was introduced by PCI-SIG On 29 May 2019. The mass production for PCIe 5.0 is planned to start in 2020.
- **PCIe 6.0:** PCI-SIG announced the development of PCI Express 6.0 specification On June 18, 2019. In a 16-lane configuration bandwidth is expected to increase to 64 GT/s, yielding 128 GB/s in each direction. It has a target release date of 2021. 4-level pulse-amplitude modulation (PAM-4) with low-latency forward error correction (FEC) in place of non-return-to-zero (NRZ) modulation is used in this new standard. Forward error correction is used to increase data integrity and PAM-4 is used as line code so that two bits are transferred per transfer which was not provided in the earlier version. It has 64 GT/s data transfer rate (raw bit rate) and up to 256 GB/s via x16 configuration.

Generation Comparison:

	Bandwidth	Gigatransfer	Frequency	Encoding
PCIe 1.0	8 GB/s	2.5 GT/s	2.5 GHz	8b/10b
PCIe 2.0	16 GB/s	5 GT/s	5.0 GHz	8b/10b
PCIe 3.0	32 GB/s	8 GT/s	8.0 GHz	128b/130b
PCIe 4.0	64 GB/s	16 GT/s	16.0 GHz	128b/130b
PCIe 5.0	128 GB/s	32 GT/s	32.0 GHz	128b/130b

6. Explain about types of USB and its users?

Sol:- **Universal Serial Bus (USB)**

USB was designed to standardize the connection of peripherals like pointing devices, keyboards, digital images and video cameras. But some devices such as printers, portable media players, disk drives, and network adaptors to personal computers used USB to communicate and to supply electric power. It is commonplace to many devices and has largely replaced interfaces such as serial ports and parallel ports. USB connectors have replaced other types of battery chargers for portable devices with themselves.

What is a Universal Serial Bus(USB)?

Universal Serial Bus (USB) is an industry standard that establishes specifications for connectors, cables, and protocols for communication, connection, and power supply between personal computers and their peripheral devices. There have been 3 generations of USB specifications:

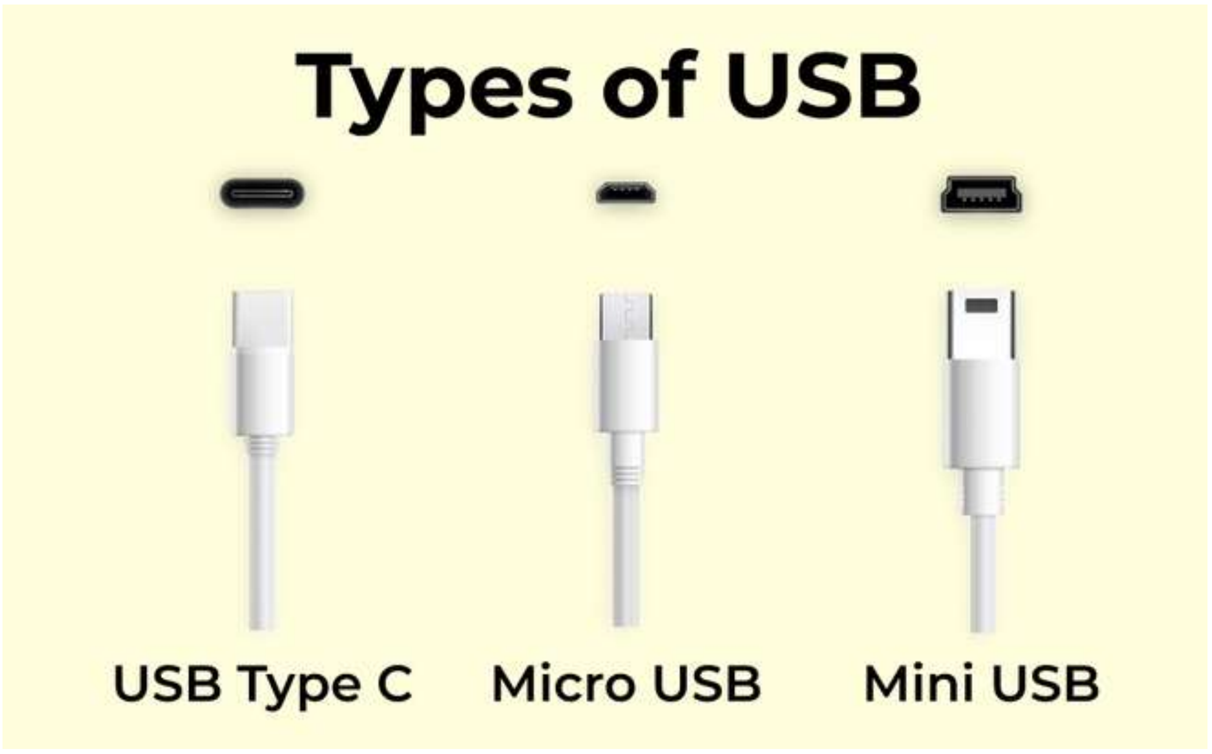
- USB 1.x
- USB 2.0
- USB 3.x

The first USB was formulated in the mid-1990s. USB 1.1 was announced in 1995 and released in 1996. It was too popular and grab the market till about the year 2000. In the duration of USB 1.1 Intel announced a USB host controller and Philips announced USB audio for isochronous communication with consumer electronics devices.

In April of 2000, USB 2.0 was announced. USB 2.0 has multiple updates and additions. The USB Implementer Forum (USB IF) currently maintains the USB standard and it was released in 1996.

USB Connector Types

USB connectors have different shapes and sizes. Most of the USB connectors are the standard USB, Mini-USB, and Micro-USB, which have two or more variations of connectors. Information on each type are shown below.



Types of USB

Mini USB

Mini USB is available in three different types A type, B type, and AB type. It is used with computer peripherals and digital cameras. The most common kind of interface is this one, that is referred to as mini B. Micro USB and USB-C cables basically take the place of mini USB on the latest devices. It uses coaxial cable to transmit data and power between two devices. it applies to mobile hard drives, digital cameras, and MP3 players. One end of a micro USB cable has a much smaller quadrilateral hub, and the other end has a regular USB hub with a flat head. It can be easily plugged into mobile devices. Although the tiny USB is mainly designed for, it can also be used to transfer data between computers having at least one USB port for charging device.

Micro USB

A reduced version of the USB (Universal Serial Bus), the micro-USB. It was created for connecting small and mobile devices including digital cameras, smartphones, GPS components, MP3 players, and photo printers and was first announced in 2007 as a replacement for mini USB.

The three different types of Micro-USB are Micro A, Micro B, and Micro USB 3. The connector size for the type Micro-A and Micro-B is 6.85 x 1.8 mm, while the Micro-A connector has a larger maximum overmild size. Because it has more pins on the side for twice as many wires than micro B, USB 3 micro is more comparable to micro B yet has faster speed. Micro USB and normal USB versions are both plug-and-play and hot-swappable is still widely used with electronic devices.

USB Type-C

A USB Type-C port is a relatively new type of connector that may be found on the majority of contemporary newer Android smartphones and other USB-connected devices. Data and power are delivered to computing machines using it. In contrast to traditional USB connections, USB-C cables can be connected into devices in either direction, including upside down.

USB Transfer Speeds

Since it is an external bus standard, USB 1.0 can accommodate up to 127 peripheral devices and data transfer rates of 12 Mbps.

The USB 2.0 standard, commonly referred to as high-speed USB, was created in 2001 by Philips, Lucent, Microsoft, Hewlett-Packard, Intel, NEC, and Compaq. It can support a transfer rate of 60 megabytes per second or more up to 480 Mbps.

USB 3.0, generally known as SuperSpeed USB 3.0, was made accessible for the first time by Buffalo Technology in November 2009. The enhanced functionality and speed of USB 3.0 contributed to advancements in power management, improved bandwidth capacity, and USB 2.0 technology.

Up to 5.0 gigabits per second (Gbps), or 640 megabytes per second, can be supported. After the release of USB 3.1, its name was changed to USB 3.1 Gen1 for manufacturing considerations. With the release of their Dell XPS and Inspiron computer series in April 2011, Dell began to roll out USB 3.0 connections.

The most recent version of the USB protocol commonly known as SuperSpeed, that was made available until July 31, 2013, is USB 3.1. It can support transfer rates of up to 10 Gbps. Recently, USB 3.0 and 3.1 revisions are used by different devices to improve speed and performance.

Advantages of USB

The Universal Serial Bus was designed to simplify and improve the interface between personal computers and peripheral devices when compared with previously existing standard or ad-hoc proprietary interfaces.

1. The USB interface is self-configuring. This means that the user need not adjust settings on the device and interface for speed or data format, or configure interrupts, input/output addresses, or direct memory access channels.
2. USB connectors are standardized at the host, so any peripheral can use any available receptacle. USB takes full advantage of the additional processing power that can be economically put into peripheral devices so that they can manage themselves. USB devices mostly do not have user-adjustable interface settings.
3. The USB interface is hot pluggable or plug and plays, meaning devices can be exchanged without rebooting the host computer. Small devices can be powered directly from the USB interface thus removing extra power supply cables.
4. The USB interface defines protocols for improving reliability over previous interfaces and recovery from common errors.
5. Installation of a device relying on the USB standard minimal operator action is required.

Disadvantages of USB

1. USB cables are limited in length.
2. USB has a strict tree topology and master-slave protocol for addressing peripheral devices. Peripheral devices cannot interact with one another except via the host, and two hosts cannot communicate over their USB ports directly.

3. Some very high-speed peripheral devices require sustained speeds not available in the USB standard.
4. For a product developer, the use of USB requires the implementation of a complex protocol and implies an intelligent controller in the peripheral device.
5. Use of the USB logos on the product requires annual fees and membership in the organization.

Comparison of USB 1.x, USB 2.0, and USB 3.x

Specification	USB 1.x	USB 2.0	USB 3.x
Release Year	1996	2000	2008
Data Transfer Rate	Low Speed: 1.5 Mbps, Full Speed: 12 Mbps	High Speed: 480 Mbps	SuperSpeed: 5 Gbps, SuperSpeed+ (SS+): 10 Gbps
Power Delivery	5V, 500mA (2.5W)	5V, 500mA (2.5W) for USB 2.0, 5V, 900mA (4.5W) for USB 3.x	5V, 900mA (4.5W) for USB 3.x, 20V, 5A (100W) for USB 3.1
Connector Types	Type A, Type B, Mini-USB, Micro-USB	Same as USB 1.x, plus Type C	Same as USB 2.0

Specification	USB 1.x	USB 2.0	USB 3.x
Cable Length Limit	5 meters (16.4 feet)	Same as USB 1.x	Same as USB 2.0
Backward Compatibility	Yes	Yes	Yes

Note: USB 3.x includes USB 3.0, USB 3.1, and USB 3.2. The USB 3.2 standard also includes two additional transfer modes: SuperSpeed+ (SS+) and SuperSpeed+ (SS++) which can transfer data at up to 20Gbps and 40Gbps respectively.

7. Explain about direct memory access?

Sol:- **Direct Memory Access (DMA) Controller in Computer Architecture**

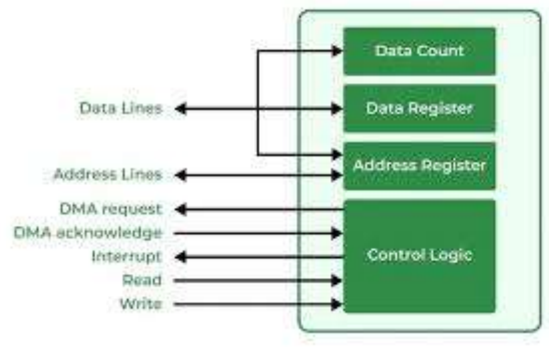
In modern computer systems, transferring data between input/output devices and memory can be a slow process if the CPU is required to manage every step. To address this, a Direct Memory Access (DMA) Controller is utilized. A Direct Memory Access (DMA) Controller solves this by allowing I/O devices to transfer data directly to memory, reducing CPU involvement. This increases system efficiency and speeds up data transfers, freeing the CPU to focus on other tasks. DMA controller needs the same old circuits of an interface to communicate with the CPU and Input/Output devices.

What is a DMA Controller?

Direct Memory Access (DMA) uses hardware for accessing the memory, that hardware is called a DMA Controller. It has the work of transferring the data between Input Output devices and main memory with very less interaction with the processor. The direct Memory Access Controller is a control unit, which has the work of transferring data.

DMA Controller in Computer Architecture

DMA Controller is a type of control unit that works as an interface for the data bus and the I/O Devices. As mentioned, DMA Controller has the work of transferring the data without the intervention of the processors, processors can control the data transfer. DMA Controller also contains an address unit, which generates the address and selects an I/O device for the transfer of data. Here we are showing the block diagram of the DMA Controller.



Block Diagram of DMA Controller

Types of Direct Memory Access (DMA)

There are four popular types of DMA.

- Single-Ended DMA
- Dual-Ended DMA
- Arbitrated-Ended DMA
- Interleaved DMA

Single-Ended DMA: Single-Ended DMA Controllers operate by reading and writing from a single memory address. They are the simplest DMA.

Dual-Ended DMA: Dual-Ended DMA controllers can read and write from two memory addresses. Dual-ended DMA is more advanced than single-ended DMA.

Arbitrated-Ended DMA: Arbitrated-Ended DMA works by reading and writing to several memory addresses. It is more advanced than Dual-Ended DMA.

Interleaved DMA: Interleaved DMA are those DMA that read from one memory address and write from another memory address.

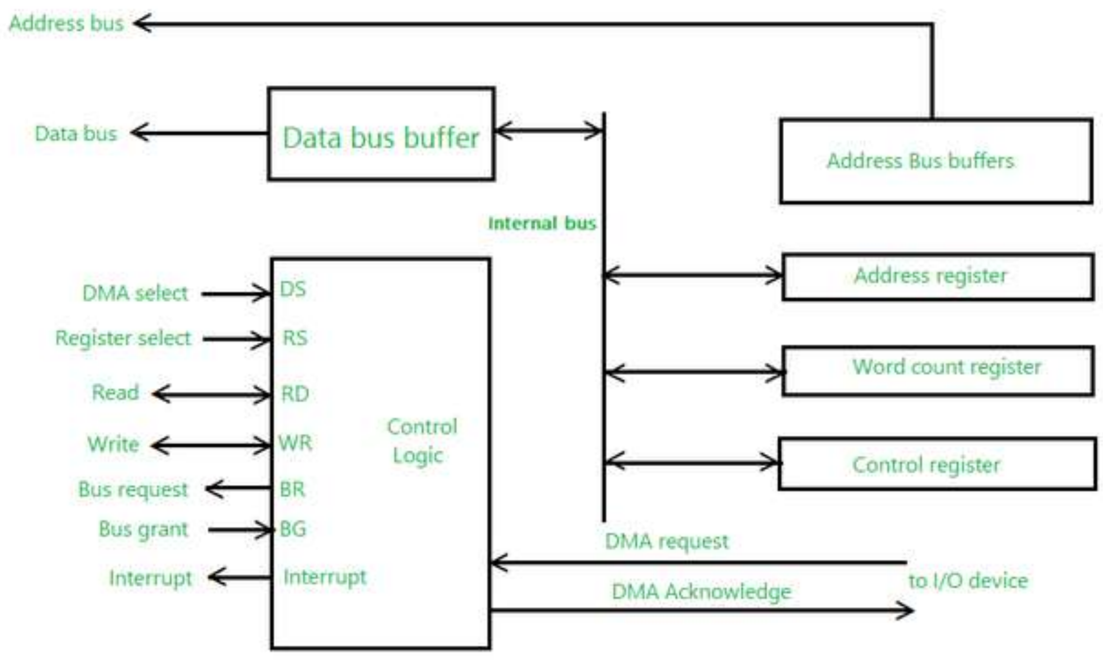
Working of DMA Controller

The DMA controller registers have three registers as follows.

- **Address register –** It contains the address to specify the desired location in memory.
- **Word count register –** It contains the number of words to be transferred.
- **Control register –** It specifies the transfer mode.

Note: All registers in the DMA appear to the CPU as I/O interface registers. Therefore, the CPU can both read and write into the DMA registers under program control via the data bus.

The figure below shows the block diagram of the DMA controller. The unit communicates with the CPU through the data bus and control lines. Through the use of the address bus and allowing the DMA and RS register to select inputs, the register within the DMA is chosen by the CPU. RD and WR are two-way inputs. When BG (bus grant) input is 0, the CPU can communicate with DMA registers. When BG (bus grant) input is 1, the CPU has relinquished the buses and DMA can communicate directly with the memory.



Working Diagram of DMA Controller

Explanation: The CPU initializes the DMA by sending the given information through the data bus.

- The starting address of the memory block where the data is available (to read) or where data are to be stored (to write).
- It also sends word count which is the number of words in the memory block to be read or written.
- Control to define the mode of transfer such as read or write.
- A control to begin the DMA transfer

Modes of Data Transfer in DMA

There are 3 modes of data transfer in DMA that are described below.

- **Burst Mode:** In Burst Mode, buses are handed over to the CPU by the DMA if the whole data is completely transferred, not before that.
- **Cycle Stealing Mode:** In Cycle Stealing Mode, buses are handed over to the CPU by the DMA after the transfer of each byte. Continuous request for bus control is generated by this Data Transfer Mode. It works more easily for higher-priority tasks.
- **Transparent Mode:** Transparent Mode in DMA does not require any bus in the transfer of the data as it works when the CPU is executing the transaction.

What is 8237 DMA Controller?

8237 DMA Controller is a type of DMA Controller which has a flexible number of channels but generally works on 4 Input-Output channels. In these present channels, the channel has to be given the highest priority to be decided by the Priority Encoder. Each channel in the 8237 DMA Controller has to be programmed separately.

What is 8257 DMA Controller?

8257 DMA Controller is a type of DMA Controller, that when a single Intel 8212 I/O device is paired with it, becomes 4 channel DMA Controller. In

8257 DMA Controller, the highest priority channel is acknowledged. It contains two 16-bit registers, one is DMA Address Register and the other one is Terminal Count Register.

Advantages of DMA Controller

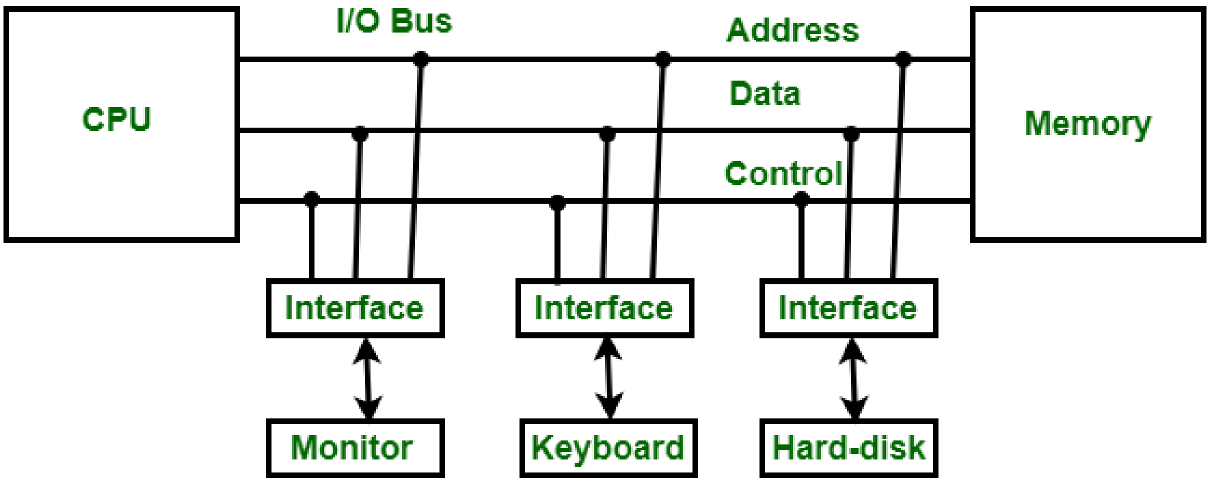
- Data Memory Access speeds up memory operations and data transfer.
- CPU is not involved while transferring data.
- DMA requires very few clock cycles while transferring data.
- DMA distributes workload very appropriately.
- DMA helps the CPU in decreasing its load.

Disadvantages of DMA Controller

- Direct Memory Access is a costly operation because of additional operations.
- DMA suffers from Cache-Coherence Problems.
- DMA Controller increases the overall cost of the system.
- DMA Controller increases the complexity of the software.

8. Explain about organization of input and output devices with diagram?

Sol:- Input-Output Interface is used as a method which helps in transferring of information between the internal storage devices i.e. memory and the external peripheral device . A peripheral device is that which provide input and output for the computer, it is also called Input-Output devices. For Example: A keyboard and mouse provide Input to the computer are called input devices while a monitor and printer that provide output to the computer are called output devices. Just like the external hard-drives, there is also availability of some peripheral devices which are able to provide both input and output.



Input-Output Interface

In micro-computer base system, the only purpose of peripheral devices is just to provide **special communication links** for the interfacing them with the CPU. To resolve the differences between peripheral devices and CPU, there is a special need for communication links.

The major differences are as follows:

1. The nature of peripheral devices is electromagnetic and electro-mechanical. The nature of the CPU is electronic. There is a lot of difference in the mode of operation of both peripheral devices and CPU.
2. There is also a synchronization mechanism because the data transfer rate of peripheral devices are slow than CPU.
3. In peripheral devices, data code and formats are differ from the format in the CPU and memory.
4. The operating mode of peripheral devices are different and each may be controlled so as not to disturb the operation of other peripheral devices connected to CPU.

There is a special need of the additional hardware to resolve the differences between CPU and peripheral devices to supervise and synchronize all input and output devices.

Functions of Input-Output Interface:

1. It is used to synchronize the operating speed of CPU with respect to input-output devices.

2. It selects the input-output device which is appropriate for the interpretation of the input-output signal.
3. It is capable of providing signals like control and timing signals.
4. In this data buffering can be possible through data bus.
5. There are various error detectors.
6. It converts serial data into parallel data and vice-versa.
7. It also convert digital data into analog signal and vice-versa.

9. Explain about USB-A, USB-B and USB-C?

Sol:- USB, short for Universal Serial Bus, is an industry standard for data communication within a limited distance. A USB port is a cable, connector, and protocols connection interface on computers, peripherals, and other electronic devices.

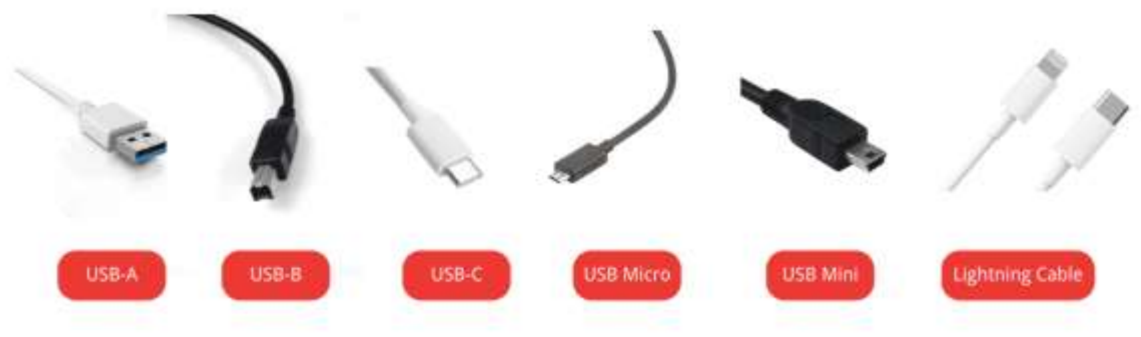
USB was originally designed for computer peripherals: keyboards, mice, external disk drives, printers, scanners, cameras, and the like. More recently, USB has become a versatile connector for audio and video devices like speakers, microphones, monitors, and webcams.

A USB port enables USB devices to be connected to each other directly or via a USB cable (on most occasions) to transfer data or supply power.

Except for data transmission, USB's charging capability mean that it can also be used solely for charging.

What types of USB ports are there?

USB ports come in different shapes and designs. Originally, there were just two USB types, USB-A and USB-B. Now, USB-C is joining the game and now changing everything.



USB-A

USB-A is the most commonly known USB type, mainly used for wired mice and keyboards and USB sticks. It's the cable with that one wider end. Only one, as the connector is not rotationally symmetrical and both ends are different, corresponding to a different type of port.

USB-B

USB-B port isn't as wide as a USB A port, and it also has a tiny rectangular hole in the middle. It is usually not used on modern computers. But it is commonly found on printers, routers, and scanners.

USB-C

USB-C port has a flatter and smaller male port. The hole in the middle of the port where the small connector pins will fit into is a small and flat oval hole. It is the new standard for replacing USB-A and USB-B, and it is designed better for thinner devices, like MacBook, iMac, and mobile phones.

USB Mini

USB Mini are further divided into two variants: USB Mini A and USB Mini B. These are smaller counterparts of Type A and Type B USB connectors. You will likely find USB Mini in portable cameras, game controllers, and some old mobile phones.

USB Micro

Both USB-A and USB-B have micro versions. Micro USB is a very common USB connector you will find in many smartphones these days. However, with the advent of USB Type C, Micro USBs are slowly getting phased out in newer models of high-end smartphones. But Micro USB is still widely used in budget smartphones and other electronic devices worldwide.

Lightning Cable

Modern Apple devices, like iPhones and iPads, often have an entirely different kind of USB connector – the Lightning cable. The cable has a thin, rectangular connector on one end and a Type C connector on the other end. Lightning cables are reversible, i.e., they can be plugged in either way without worrying about which is the right side up.

What is the difference between USB-A, USB-B, and USB-C?

USB-A, B, or C only refers to the physical design (or shape) of the ports and connectors. USB-A is in a flat and rectangular shape. USB-B comes in a variety of designs, and the standard one is a bit squarer. Moreover, USB-C has a more compact, rectangular shape with rounded corners.

Whether it's a USB-A, B, or C, their data transfer speed are determined by USB versions. All three types of USB ports can run USB versions of 1.1, 2.0, 3.0, or 3.1. The higher the USB version is, the faster the speed it brings.

	USB-A	**USB-B**	**USB-C**
Types	USB Type A, USB Mini A, USB Micro A	USB Type B, USB Mini B, USB Micro B	/
Shape	Rectangular	Multiple designs (mostly square)	Smaller r with rour corners
Connection	Host and connector	Connector	Host and
Backward Compatible	No	Yes	Yes

	USB-A	USB-B	USB-C
Reversible	No	No	Yes
Versions	1.1, 2.0, 3.0, 3.1	1.1, 2.0, 3.0, 3.1	1.1, 2.0, 3
Version Speed	1.1 - top speed of 12Mbps, 2.0 – top speed of 480Mbps, 3.0 – speed of 5Gbps, 3.1 – top speed of 10Gbps		
Devices	Almost all desktops, laptops, and other computers Most tablets Smart TVs Game consoles Flash drives or memory sticks DVD and Blue-Ray players Peripherals like keyboards and computer mice	Printers Scanners Projectors Other devices that connect to your computer	Some of t smartpho Some of t laptops Nintendo game con USBC Hul Laptop & Smartpho

USB not working, what should you do?

Have your PC or laptop USB ports stopped working? Here are five of the most common ways to troubleshoot and fix a USB port that is not working properly.

- **Try a different USB port.**

  Unplug the device and plug it into a different USB port.

- **Replace the USB cable.**

  If it's old or has been roughly handled, it might not work right anymore. Change the cable and see if that resolves the issue.

- **Clean the USB port.**

Use a very thin, non-metallic tool like a toothpick to carefully clear the port. Don’t use blasts of canned, high-pressure air.

- **Restart the computer.**

  Turning it off and then back on again can flush out any corrupt data and reset device drivers automatically.

- **Reset SMC if you are using a Mac.**

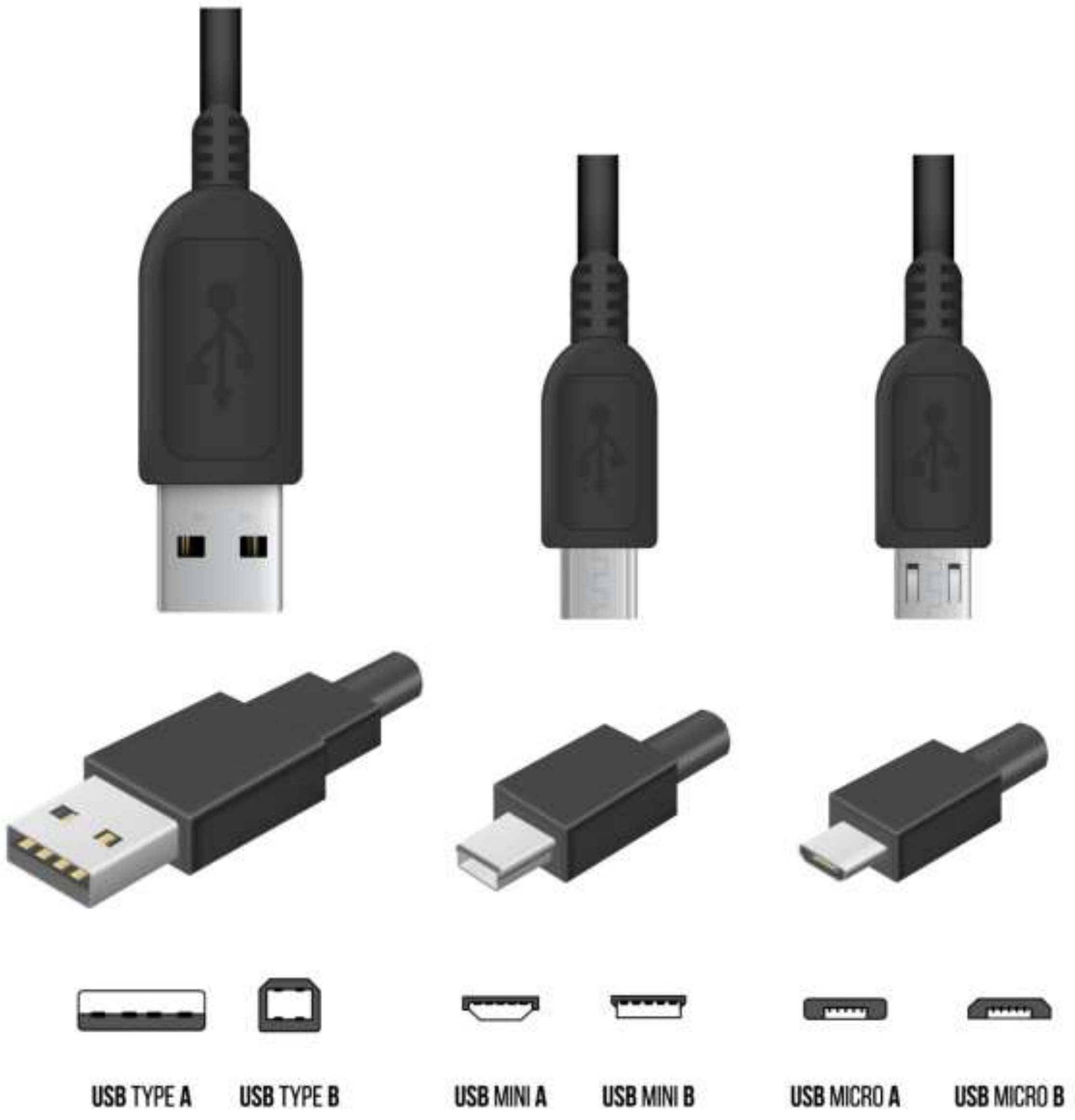
  Or open the device manager to scan for hardware changes and disable and re-enable the USB controller if you have a Windows PC.

10. Explain about micro-USB and mini-USB?

Sol:- For a technology that has "universal" as the first part of its name, there sure are a lot of different types of USB connectors. (Look, we didn't name the technology, OK?) Purchase the wrong type of connector, and you might be dealing with some headaches when you can't connect your devices.

While our offerings at USB Memory Direct are focused on awesome custom flash drives with the most common USB connectors, there are definitely many more styles of USB device connectors out there. In fact, some of those styles have been quite important in the history of the devices we all use today.

In this five-minute primer, we'll talk about the question of mini USB vs. micro USB. (Yeah, they're different things!) Plus, we'll throw in some important information about the technology that's going to replace both.



What Is a Mini USB Connector?

A mini USB connector is a much smaller, five-pin version of a standard USB connector, with two crimped sides that give it a trapezoid shape. (Reach back to your memory of geometry class for that one!) These connectors are also known today as mini-B connectors.

Mini USB was introduced in the early 2000s, making it one of the first USB connector variants. It first showed up on devices like mp3 players and point-and-shoot digital cameras. (Remember those?)

The mini-B and its fully extinct cousin, the mini-A, provided an important advantage: They allowed both charging and data transfer between a device like a phone or camera and a computer. Back in the day, that was a revolutionary concept and a hugely useful technology for a world that was just beginning to love mobile devices.

Only a small handful of devices use mini USB connectors today. The technology isn't quite obsolete, but it's rare enough that you might have to order the cables online rather than grabbing them off the shelf from a big-box retailer. It's important mostly as a step in the history of USB cables.

MINI USB CONNECTOR

What Is a Micro USB Connector?

In 2007, several years after the mini USB connector made its debut, the micro USB connector -- also known as the micro-B connector -- arrived. Its sleek five-pin shape is basically a slimmer version of the mini-B connector, built with the intended purpose of standardizing the connectors for Android mobile devices.

The effort at standardization paid off, too -- micro USB has been the standard connector for the charging ports on Android phones and other devices for many years. They offer charging speeds and data transfer capabilities comparable to or better than mini USB, and devices and accessories with micro USB connectors are still reasonably common.

Unfortunately, micro USB connectors have had their share of issues, too. For example, a micro USB cable designed for USB 2.0 might not be compatible with a USB 3.0 device. The ports and connectors are also notoriously easy to break, as anyone who's ever accidentally bent their Android charger slightly out of shape can attest.

However, these issues might become moot points soon. That's because manufacturers who previously used micro-USB connectors have begun to phase them out in favor of USB-C connectors. Next, let's look at what we can expect from that exciting new technology.

MICRO USB CONNECTOR

PART-B (2 MARKS)

1. What are the input devices?

Sol:-

- **Keyboard**
- **Mouse**
- **Microphone**
- **Scanner**
- **Webcam**
- **Touchscreen**
- **Joystick**
- **Trackball**
- **Light Pen**
- **Barcode Reader**

2. What are the output devices?

Sol:-

- Monitors
- Graphic Plotter
- Printer
- Speakers
- Headphones
- Projector
- GPS

3. Types of USB?

Sol:-

- USB-C
- USB-A
- USB B
- Mini USB
- Micro USB
- USB 2.0
- USB 3.1
- USB 1.0
- USB 3.2
- USB 3.0
- USB Micro B
- USB Mini B
- USB 4.0

- Thunderbolt 3
- Type
- USB 3.0 cables
- USB4
- Identify Usb Versions
- Micro Usb-b
- Type B
- USB 2.0 cables
- USB adapters
- USB connector types
- USB docks

4. How is direct memory access works?

Sol:- **Working:**

- The **DMA controller** facilitates the transfer of data between memory and peripherals without involving the CPU for each individual data operation, as mentioned in the article.
- The **DMA Select** and **DMA Request** initiate the process when a peripheral wants to transfer data, similar to how DMA allows peripherals to operate independently of the CPU.
- **Address Bus** and **Data Bus** handle the flow of data and memory addresses during the transfer, improving system efficiency by bypassing the CPU.
- The **Registers** (Address Register, Word Count Register, Control Register) store the necessary information to control the transfer, as described in the article, where DMA controls the movement of data between the device and memory.

- The **Interrupt** is triggered once the transfer is completed, similar to how the CPU is notified in the article that DMA operations have been finished.

5. What is the mission understandable language?

Sol:- The "mission understandable language" you're likely referring to is machine language, the fundamental language computers understand, consisting of binary code (0s and 1s).

6. What is the purpose of compiler and interpreter in a computer?

Sol:- Both compilers and interpreters translate high-level programming languages into machine code, but they do so differently: compilers translate the entire code at once before execution, while interpreters translate and execute code line by line.

Compiler:

- **Purpose:** Translates the entire source code (written in a high-level language) into machine code (or an intermediate representation) before the program is executed.
- **Process:**
    - Takes the entire program as input.
    - Analyzes the code for errors and generates an executable file (e.g., .exe).
    - This executable file can then be run directly by the computer.
- **Examples:** C, C++, Java.

Interpreter:

- **Purpose:** Translates and executes source code line by line during program execution.
- **Process:**
    - Reads one line of code, translates it, and executes it immediately.

    - Does not generate an independent executable file.
    - Requires the source code to be present during execution.
- **Examples:** Python, Ruby.

7. Define interface circuits?

Sol:- Interface circuits are electronic circuits designed to connect and enable communication between different devices or systems, ensuring compatibility and facilitating data transfer or control signals. They act as a bridge, converting signals and formats to match the requirements of the connected devices.

8. What are the input devices and which device converts the human understanding language into machine language?

Sol:- Input devices like keyboards, mice, joysticks, and microphones allow users to interact with a computer, while a translator program (compiler or interpreter) converts human-readable code into machine language.

9. Which are the latest input and output devices?

Sol:-

Latest input devices are:-

1. Advanced touchscreens
2. Gesture recognition technologies
3. VR controllers

Latest output devices are:-

1. High-resolution displays
2. Advanced audio systems
3. 3D printing technology

10. Which converts the machine language into human understandable language?

Sol:- A disassembler converts machine code (the language computers understand) into a more human-readable assembly language, which can then be further analyzed or translated.