Unit No 2:-

Binary System

2.2 Number System Conversation:-

2.2.1 Decimal to Binary and Binary to decimal.

Decimal to Binary:- To convert a decimal number to binary, we divide the the no. by 2 and take quotient and remainder. We continue dividing the quotient by 2 until we get quotient 0

example:- Convert 166_{10} (156 in decimal to binary).

2	156
2	78 - 0
2	39 - 0
2	19 - 1
2	9 - 1
2	4 - 1
2	2 - 0
	1 - 0

$156_{10} = 10011100_2$

Binary to decimal: To conversation of a number from binary number system to decimal number system is explained below with the help of an example.

example: Convert $(1000001)_2$ to decimal

$= 1\times2^6 + 0\times2^5 + 0\times2^4 + 0\times2^3 + 0\times2^2 + 0\times2^1 + 1\times0^0$

$= 64 + 1$

$= (64)_{10}$

2.2.2 Decimal to Hexadecimal and Hexa to decimal.

Decimal to Hexadecimal:- Hexadecimal number system has base 16, so for conversation of a number from decimal to hexadecimal, we divide the number by 16 and take both quotient and remainder, We continue dividing the quotient by 16 until the quotient becomes 0.

example:

16	69610
16	4350 – A
16	271 – E
16	16 – F
16	1 – 0

Hexadecimal to decimal:- The method for this conversation is same as converting from binary to decimal except the base value. Since hexadecimal has been 16, the "place value" correspond to the power of 16. To convert this multiply each place value by the corresponding power of 16.

example: Convert $(C921)_{16}$ to decimal.

$= C \times 16^3 + 9 \times 16^2 + 2 \times 16^1 + 1 \times 16^0$

$= 12 \times 16^3 + 9 \times 16^2 + 2 \times 16^1 + 1 \times 16^0$

$= 12 \times 4096 + 9 \times 256 + 2 \times 16 + 1 \times 1$

$= 49152 + 2304 + 32 + 1$

$= (51489)_{10}$

2.2.3 Hexadecimal to binary and Binary to Hexa:

Hexadecimal to binary: To convert Hexadecimal to binary, simply convert each hexadecimal digits binary values To find the four digit binary value,

Example:- Convert $(A23)_{16}$ ($A23$ in Hexa) to Binary.

In this number, there are three hexa decimal digits. Binary of each digit is given as

i) For A, the binary value is 1010

ii) For 2, the binary values is 0010

iii) For 3, the binary value is 0011

By combing all the binary values, we get 1010 0010 0011

So, $(A23)_{16} = (101000100011)_2$

Binary to Hexadecimal:- This conversion is also very easy with the help of Table 24. In the given binary number we start making groups four digits from right to left and replace every group with a hexa decimal digit.

example: Convert $(111010111)_2$ to de hexa decimal.

The groups in this binary number are

11010 1111

The left most group in blue colour has only 1 binary digit and by adding 0s we get:

0001 1010 1111

We replace each group with the respective hexadecimal and we get:

1AF

So, $(11010\ 1111)_2 = (1AF)_{16}$

2.3 Memory and DataStoreage:-

2.3.1 Memory: Computer memory is any physical device capable of storing data. There are two types of memory.

1- Volatile memory

2- Non- Volatile memory.

- **Volatile memory:** A device which holds data as long as it has power supply connected to it, is called volatile memory. for example Random access memory (RAM) which holds memory only as long as connected on it

- **Non-Volatile memory**:- A device which can hold data even if it is not connected to any power source is called non-volatile memory.
for example: Hard drivers, flash drivers. and memory card etc

2.3.2 Data representation in computer memory.

Digital computers store data in binary form. It means that the whether it is a text, picture, movie or same application, it is stored in computer's memory in the form of 0s and 1s
All the characters on the keyboard has an associated code in binary The code is called ASCII code of the character. ASCII stand for. **American standard Code for Information Interchange.**

2.3.3 Storage Device :- Any computing hard ware that is used for storing, porting and extracting data, is called a storage device. It can hold or store information both temporarily and permanently. It can also be internal and external to a computer. An external storage device is a a plug an play device Internal storage devices connected to same fixed slots.

example: RAM, Hard disk, CD, USB etc.

Difference b/w memory and Storage.

Memory	Storage.
Place where an application load its data during processing.	Usually the place where data is stored for long or short term.
Temporary storage device	Permanent storage device.
lesser in size	Greater in size.
High accessing speed	Low accessing speed.
It is called primary memory	It is called secondary memory.

2.4 Measurement of size of computer Memory:

The smallest amount of data to be stored in computer's memory is a 0 or 1. It is called bit. A collection of eight bits is called a byte.

Unit	Size
Bit	Smallest unit of data, can hold only value: 0 or 1
Byte	Groups of eight bits, enough space to store single ASCII character
kilo byte	1KB = 1,024 bytes
Mega byte	1MB = (1,024) KB or $(1024)^2$ bytes
Giga byte	1GB = 1024 MB or $(1024)^3$ bytes
Tera byte	1TB = 1024 GB or $(1024)^4$ bytes
Peta byte	1PB = 1,024 TB or $(1024)^5$ bytes.

2.5 Boolean Algebra

2.5.1 Boolean Proposition:-

The proposition is sentence that can either be true or false.

example: "I play chess"

"I want to excel in mathematics"

But following sentence are not proposition

1. How are you?
2. Close the door.

2.5.2 Truth values: Every proposition takes one of two values true or false, and these values are called the truth values.

e.g:- Assume Q = "The sun rises in the west". Now assume another proposition R = "I have completed my homework."

2.5.3 Logical operators (And, or, Not)

Some time we assemble more than one proposition called a compound proposition. for example:

1) Today is Monday'
2) I am in School.

AND opearator (•):- If we use "AND" operator to connect two or more proposition, the the compound proposition is true only if all the connected proposition are true. It is denoted by dot (•) symbol. It means that P and Q may also be writtenas P.Q

OR operator :- We can also use "OR" operator to connect two or more proposition e.g "Today is Monday OR I am in school" In case of OR operator, the compound proposition is true if atleast one proposition is true. In other words the compound proposition of false only if all the proposition are false. OR operator can also be denoted by a Plus + symbol. It means that P OR Q may be also written as P+Q

Not operator :- The logical operator "Not" is not a connector but it is used to negate a proposition. For example, if P= "Today is Monday" then Not (P) means "Today is not Monday". So, with Not operator a True values becomes false and vice versa. Not operator can also be denoted by a "¬" symbol. It means that Not (P) may also be written as ¬P.

2.5.4 Truth table:- A truth table is used to check whether a proposition is True or false.

Truth table for AND operator:-

P	Q	P and Q
T	T	T
T	F	F
F	T	F
F	F	F

Truth table for OR operator:-

P	Q	P AND Q
T	T	T
T	F	T
F	T	T
F	F	F

Truth table for Not operator:-

P	Not (P)
T	F
F	T

Truth table for complex Boolean expression:

P	Not P)	Q	Not(P) AND(Q)
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F

2.5.5 LAWS of Boolean Algebra :-

The laws of Boolean Algebra helps to simplify complex Boolean Algebra expression.

- **Commutative Law:** This law states that the order of application of two separate proposition is not important. So,

a) A·B = B·A (The order in which two variables are ANDed and makes no difference)

b) A+B = B+A (The order in which two variables are ORed makes no difference)

A	B	A·B	B·A
F	F	F	F
F	T	F	F
T	F	F	F
T	T	T	T

- **Associative Law:-** This law is for several variable. According to this law there is no change in result if a grouping of expression is changed This law is the quite same in case of AND and OR operator.

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

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

A	B	C	A+B	B+C	(A+B)+C	A+(B+C
F	F	F	F	F	F	F
F	F	T	F	T	T	T
F	T	F	T	T	T	T
F	T	T	T	T	T	T
T	F	F	T	F	T	T
T	F	T	T	T	T	T
T	T	F	T	T	T	T
T	T	T	T	T	T	T

Truth table for complex Boolean expression

P	Not P)	Q	Not(P) AND(Q)
T	F	T	F
T	F	F	F
F	T	T	T
F	T	F	F

2.5.5 LAWS of Boolean Algebra :-

The laws of Boolean Algebra helps to simplify complex Boolean Algebra expression.

- **Commutative Law:** This law states that the order of application of two separate propsition is not important. So,

a) A·B = B·A (The order in which two variables are AND'ed and makes no difference)

b) A+B = B+A (The order in which two variables are OR'ed makes no difference)

A	B	A·B	B·A
F	F	F	F
F	T	F	F
T	F	F	F
T	T	T	T

- **Associative Law:-** This law is for several variable. According to this law there is no change in result if a grouping of expression is change. This law is the quite same in case of AND and OR operator.

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

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

A	B	C	A+B	B+C	(A+B)+C	A+(B+C)
F	F	F	F	F	F	F
F	F	T	F	T	T	T
F	T	F	T	T	T	T
F	T	T	T	T	T	T
T	F	F	T	F	T	T
T	F	T	T	T	T	T
T	T	F	T	T	T	T
T	T	T	T	T	T	T

- **Distributive Law:-**

The law is discussed in two ways i.e; "AND over OR" and "OR over AND".

a) $A \cdot (B + C) = (A \cdot B) + (A \cdot C)$ (AND over OR)

b) $A + (B \cdot C) = (A + B) \cdot (A + C)$ (OR over AND)

A	B	C	B+C	A·B	A·C	A·(B+C)	A·B+A·C
F	F	F	F	F	F	F	F
F	F	T	T	F	F	F	F
F	T	F	T	F	F	F	F
F	T	T	T	F	F	F	F
T	F	F	F	F	F	F	F
T	F	T	T	F	T	T	T
T	T	F	T	T	F	T	T
T	T	T	T	T	T	T	T

- **Identity Law:** If a variable is ORed with a false, the result is always equal to that variable. AND if a variable is ANDed with a true, the result is always equal to that variable.

a) A or False = A, A variable ORed with false is always equal to that variable

b) A AND True = A, A variable AND'ed with True is always equal to that variable.

2.5.6 Logical expressions:- We get a logical expression when some logical operator is applied to the Boolean proposition(s). For example, P and Q_1, $\neg(P \text{ OR } Q)$, P OR Q etc