Course Title: 103 Introductions to Computers

UNIT-3: Number System

3.1. Introduction of Decimal, Binary, Octal and Hexadecimal number Systems.
3.2 Conversion of Decimal to Binary and Binary to Decimal
3.3 Binary addition & subtraction
3.4 ASCII and ANSI character code

What is a number system?

A Number system is a method of showing numbers by writing, which is a mathematical way of representing the numbers of a given set, by using the numbers or symbols in a mathematical manner. The writing system for denoting numbers using digits or symbols in a logical manner is defined as a Number system. The numeral system Represents a useful set of numbers, reflects the arithmetic and algebraic structure of a number, and provides standard representation. The digits from 0 to 9 can be used to form all the numbers. With these digits, anyone can create infinite numbers. For example, 156,3907, 3456, 1298, 784859 etc.

There are two types of Number System:

1. **Non Positional Number System –**

- ➔ In this, each symbol represents the same value, regardless of its position inthe number. Symbols are used to find out the value of a particular number.
- ➔ Use symbols such as I for 1, II for 2, III for 3, IIII for 4, IIIII for 5, etc
- ➔ Each symbol represents the same value regardless of its position in the number
- ➔ The symbols are simply added to find out the value of a particular number
- ➔ It is difficult to perform arithmetic with such a number system

2. **Positional Number System –**

- ➔ The number system having a few symbols representing different values,depending on the position they occupy in the number is called positional number system.
- ➔ The radix or base of the number system is the total number of digits allowedby the number system.
- ➔ Use only a few symbols called digits
- ➔ These symbols represent different values depending on the position they occupy in the number
- ➔ The value of each digit is determined by:

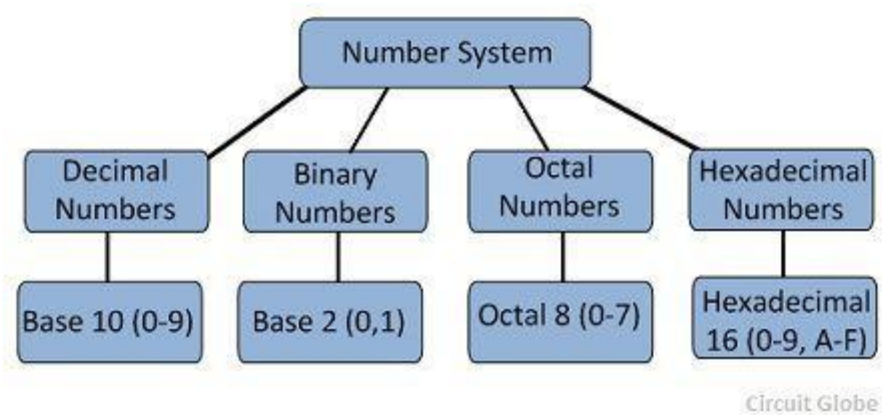
  1.The digit itself
  2. The position of the digit in the number
  3. The base of the number system
  (base = total number of digits in the number system)

- ➔ The maximum value of a single digit is always equal to one less than the value of the base

3.1. Introduction of Decimal, Binary, Octal and Hexadecimal number Systems.

- **According to Radix/Base, there are four type of number systems :**

1. Binary Number System
2. Octal Number System
3. Decimal Number System
4. Hexadecimal Number System



1. Binary Number System

- Base of binary number system is 2.
- 0 and 1 are the symbols used in binary number system.
- 1 is the largest single digit number.
- A single symbol of binary number system (0 or 1) is called bit.
- Aggregation of 8 bits is called byte.
- Aggregation of 4 bits is called nibble or half byte or half octet.
- Also called base 2 number system.
- Digital systems handle information using electronic components such as transistors etc. all of which can represent only two states or conditions on (1)or off(0) .
- The highest decimal number that can be represented by n bits binary numberis 2n-1.
- Computer stores numbers, letters and other characters in binary form.
- E.g. (10101)2

2. Octal Number System

- Base of octal number system is 8. ,
- 0 to 7 are the symbols used in octal number system.

- 7 is the largest single digit number used in octal number system.
- Each octal digit can be represented using 3 binary digits.
- Uses eight digits 0,1,2,3,4,5,6 and 7.
- Also called base 8 number system.
- E.g. (12570)8

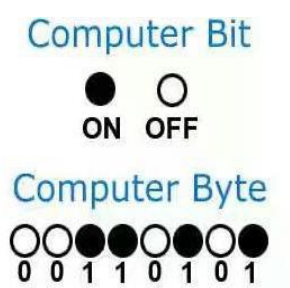
3. Decimal Number System

- Base of decimal number system is 10.
- 0 to 9 are the symbols used in decimal number system.
- 9 Is the largest single digit number used in decimal number system,
- Use ten digits 0,1,2,3,4,5,6,7,8,9.
- Also called base 10 number system.
- Used in day-to-day life.
- E.g. (3925)10

4. Hexadecimal Number System

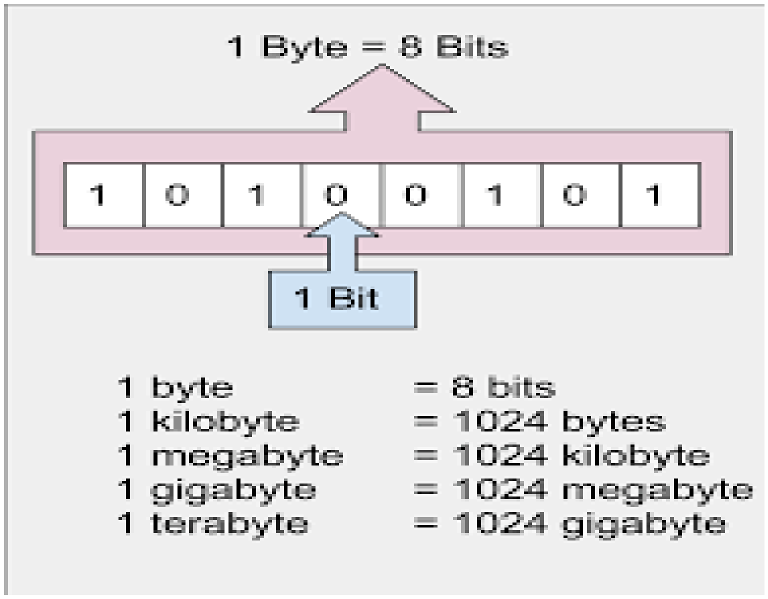
- Base of hexadecimal number system is 16.
- 0 to 9, A, B, C, D, E, and F are the symbols used in hexadecimal number system.
- F is the largest single digit number used in hexadecimal number system. IM
- Each hexadecimal digit can be represented using 4 binary digits.
- A nibble can be represented by a single hexadecimal digit also called hex digit.
- Use 16 digits 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.
- Also called base 16 number system.
- E.g. (19FDE)16

<u>Bits and Bytes –</u>



- The smallest unit of data in a computer is called Bit (Binary Digit).
- A bit has a single binary value i.e. 0 or 1.

- There are eight bits in a byte.
- A single character would use one byte of memory.
- An alphanumeric character is stored as 1 byte. For example, to store the letter 'R' uses 1 byte, which is stored by the computer as 8 bits '01010010'.
- Bit – A bit is a value of either a 1 or 0.
- Nibble – A nibble is 4 bits.



Need of Binary Number System

Digital computers use digital circuits. Digital circuits can handle only two states or conditions on /high (1) or off/low (0). Binary number system is ideally suited for representing two possible states and it greatly simplifies the internal circuit design of computers, resulting in less expensive and more reliable circuits.

Need of Decimal Number System

Human beings are more familiar with decimal number system. They can easily read, understand and manipulate decimal number system.

Need of Hexadecimal and Octal Number System

Octal and hexadecimal number systems are great ways to concisely represent a bit pattern. Each octal digit is exactly equivalent to 3 bits, and each hexadecimal digit is exactly equivalent to 4 bits. They are easily convertible to and from binary.

HEXADECIMAL	DECIMAL	OCTAL	BINARY
0	0	0	0000
1	1	1	0001
2	2	2	0010
3	3	3	0011
4	4	4	0100
5	5	5	0101
6	6	6	0110
7	7	7	0111
8	8	10	1000
9	9	11	1001
A	10	12	1010
B	11	13	1011
C	12	14	1100
D	13	15	1101
E	14	16	1110
F	15	17	1111

3.2 Conversion of Decimal to Binary and Binary to Decimal

❖ **Decimal to binary –**

- Divide the decimal number to be converted by 2.
- Get the remainder from step 1 as the rightmost digit (Least Significant Digit(LSD)) of new base number.
- Divide the quotient of the previous divide by 2.
- Record the remainder form step 3 as the next digit(to the left) of the new basenumber.
- Repeat step 3 and 4, getting remainders from right to left, until the quotientbecomes zero in step 3.
- The Last remainder obtained will be the Most Significant Digit (MSD) of the newbase number. At last remainders have to be arranged in reverse order.

Decimal to Binary Example:

EXAMPLE 1:

$(78)_{10} = (?)_2$

		remainder
2	78	
2	39	0
2	19	1
2	9	1
2	4	1
2	2	0
	1	0

Solution:

answer is $(1 0 0 1 1 1 0)_2$

Ex

Convert $(68)_{10}$ to Binary

2	68	0
2	34	0
2	17	1
2	8	0
2	4	0
2	2	0
2	1	1
	0	

$(68)_{10} = (1000100)_2$

Convert 125_{10} to Binary

2	125	1
2	62	0
2	31	1
2	15	1
2	7	1
2	3	1
2	1	1
	0	

$(125)_{10} = (1111101)_2$

❖ <u>Decimal to binary with fraction part to binary –</u>

- Begin with fractional part of decimal and multiply by 2. The whole number part of the result is the first binary digit to the right of the point.

Ex

.625 x 2 = 1.25, the first binary digit to the right of the point is a 1.

- Next disregard the whole number part of the previous result (the 1 in this case) and multiply by 2 for fractional part (0.25 in this case) again. The whole number part (0) of this new result (0.50) is the second binary digit to the right of the paint.
- Continue this process until you get a zero as fractional part or until an infinite repeating pattern is recognized (if zero in fractional part does not come, minimum repeat process for six times).

EXAMPLE

$(25.625)_{10} = (?)_2$

		remainder
2	25	
2	12	1
2	6	0
2	3	0
	1	1

Solution:

integer part,

fractional part,

0.625 * 2 = 1.25 integer 1

0.25 * 2 = 0.5 integer 0

0.5 * 2 = 1 integer 1

So, $(0.101)_2$

Now, answer will be $(11001.101)_2$

Convert $(124.24)_{10}$ to Binary

2	124	0
2	62	0
2	31	1
2	15	1
2	7	1
2	3	1
2	1	1
	0	

0.24 x 2 = 0.48	0
0.48 x 2 = 0.96	0
0.96 x 2 = 1.92	1
0.92 x 2 = 1.84	1
0.84 x 2 = 1.68	1
0.68 x 2 = 1.36	1

$(124.24)_{10}= (1111100.001111)_2$

Convert $(120.25)_{10}$ to Binary

2	120	0
2	60	0
2	30	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	

0.25 x 2 = 0.50	0
0.50 x 2 = 1.00	1

$120.25_{10}= (1111000.01)_2$

❖ <u>Binary to Decimal:</u>

Binary number system is like decimal number system, except that the base is 2, instead of 10. We can use only two symbols or digits (0 and 1) in this system. Notethat the largest digit is 1 (One less the base).

1. Determine the positional value of each digit.

Ex: (101)2 Positional value of underlined one is 2^2=1*4=4

2. Multiply the positional value (in step 1) by the digits in the corresponding columns.
3. Sum up the products calculated in step 2. Total Is equivalent value in decimal.

EXAMPLE:

1. $(101000)_2 = (?)_{10}$
$= 1*(2)^5 + (0*2)^4 + (1*2)^3 + (0*2)^2 + (0*2)^1 + (0*2)^0$
$= 32+0+8+0+0+0$
$= 32+8$
$= (40)_{10}$

2. $(101111)_2 = (?)_{10}$
$= 1*(2)^5 + 0*(2)^4 + 1*(2)^3 + 1*(2)^2 + 1*(2)^1 + 1*(2)^0$
$= 32+0+8+4+2+1$
$= 32+8+4+2+1$
$= (47)_{10}$

3. $(10011101)_2 = (?)_{10}$
$= (1*2)^7 + (0*2)^6 + (0*2)^5 + (1*2)^4 + (1*2)^3 + (1*2)^2 + (0*2)^1 + (1*2)^0$
$=128+0+0+16+8+4+0+1$
$=128+16+8+4+1$
$= (157)_{10}$

4. $(10010110)_2 = (?)_{10}$
$= (1*2)^7 + (0*2)^6 + (0*2)^5 + (1*2)^4 + (0*2)^3 + (1*2)^2 + (1*2)^1 + (0*2)^0$
$=128+0+0+16+0+4+2+0$
$=128+16+4+2$
= (150)10

Convert $(10101)_2$ to Decimal

$(10101)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 16+0+4+0+1$

$= (21)_{10}$

Convert $(101101.001)_2$ to Decimal

$(101101.001)_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

$= 32 + 0 + 8 + 4 + 0 + 1 + 0 + 0 + 1 \times 0.125$

$= (45.125)_{10}$

$(101101.001)_2 = (45.125)_{10}$

Convert $(110001.101)_2$ to Decimal

$(110001.101)_2 = 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

$= 32 + 16 + 0 + 0 + 0 + 1 + 1 \times 0.5 + 0 + 1 \times 0.125$

$= (49.625)_{10}$

3.3 Binary addition & subtraction

Binary Addition –

- There are four rules for binary addition.

Input A	Input B	Sum (S) A+B	Carry (C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Example:

0011010 + 001100 = 00100110

11	carry
0011010	$= 26_{10}$
+0001100	$= 12_{10}$
0100110	$= 38_{10}$

Add $(1100111)_2$ to $(110011)_2$

Carry	1	1			1	1	1	
		1	1	0	0	1	1	1
	+		1	1	0	0	1	1
	1	0	0	1	1	0	1	0

Addition of 3 ones (1+1+1) can be performed by 1+1=10, 10+1=11. So, 1+1+1=11 (1 carry 1)

Add $(11101.11)_2$ to $(11011.10)_2$

Carry	1	1	1	1	1	1			
		1	1	1	0	1	.	1	1
	+	1	1	0	1	1	.	1	0
	1	1	1	0	0	1		0	1

<u>Binary Subtraction –</u>

- There are four rules for binary subtraction.

Input A	Input B	Subtract (S) A-B	Borrow (B)
0	0	0	0
0	1	0	1
1	0	1	0
1	1	0	0

Examples:

0011010 - 001100 = 00001110

$$
\begin{array}{rl}
1\ 1 & \text{borrow} \\
0\ 0\ \cancel{1}\ \cancel{1}\ 0\ 1\ 0 & = 26_{10} \\
-\,0\ 0\ 0\ 1\ 1\ 0\ 0 & = 12_{10} \\
\hline
0\ 0\ 0\ 1\ 1\ 1\ 0 & = 14_{10}
\end{array}
$$

Subtract $(1001)_2$ from $(10110)_2$

Borrow	0	10		0	10	← Decimal value 2
	~~1~~	0	1	~~1~~	0	
-		1	0	0	1	
	0	1	1	0	1	

Subtract $(110.0011)_2$ from $(111.0100)_2$

							1		
Borrow						0	~~10~~	10	
	1	1	1	.	0	~~1~~	0	0	
-	1	1	0	.	0	0	1	1	
	0	0	1	.	0	0	0	1	

3.4 ASCII and ANSI character code

- ➔ ASCII (American Standard Code for Information Interchange) –is a 7-bit character set. It let's You represent up to 128 characters, number 0 to127
- ➔ It is a standard data-transmission code that is used by personal computers to represent letters, numbers, punctuation marks, and control characters.
- ➔ ASCII code is popular in data communications, used to represent data inmicrocomputers and frequently found in larger computers.
- ➔ ANSI (American National Standards Institute) published ASCII in 1963.
- ➔ 0 to 31 & 127 is control character which can not display like enter key, backspace key, etc.

Character	ASCII value in Decimal
0 – 9	48 – 57
A – Z	65 – 90
a – z	97 – 122

ANSI character code:

- The generic term ANSI (American National Standards Institute) is the primary organization of US which is involved in the development of technology standards in the United States.
- Computer standards from ANSI include the American Standard Code for Information Interchange (ASCII) and the Small Computer System Interface (SCSI).
- ANSI Character Set, also known as Windows Code Page, is an 8-bit character set used by Microsoft Windows 95 and Windows 98 that lets yourepresent up to 256 characters (numbered 0 through 255).
- The ASCII (American Standard Code for Information Exchange) character set is a subset of the ANSI (American National Standards Institute) characterset with characters numbered 32 through 126, each representing a displayable character. Some ANSI character codes cannot be displayed by Windows 95 or Windows 98 applications and are generally displayed as solid blocks on the output device.
- The 256-character limit of ANSI supports only a few international characters

Differences

- When ASCII was created, it only used 7 bits for a total maximum combinationof 128 characters. It was created for the English language, and it proved good enough to hold all the letters, numbers, special characters and symbols, as wellas non-printed characters.
- In ANSI, 8 bits are used; increasing the maximum number of characters to be represented up to 256. This is expanded even further because of how ANSI uses code pages with different character sets. There are a number of ANSI codepages that are meant for other languages like Japanese, Chinese, and many others. The application processing the file just needs to know which code page is in use in order to decipher the files properly.
- ANSI has more characters than ASCII
- ASCII uses 7 bits while ANSI uses 8
- ASCII characters are fixed to the code points while ANSI code points mayrepresent different characters
- ASCII is more straightforward to use than ANSI
- ASCII works with Unicode while ANSI compatibility is very limited