Chapter-3

DATA REPRESENTATION

➢ Introduction

- In Digital Computer, data and instructions are stored in computer memory using binary code (or machine code) represented by **B**inary dig**IT**'s 1 and 0 called **BIT**'s.
- The data may contain digits, alphabets or special character, which are converted to bits, understandable by the computer.
- The number system uses well defined symbols called digits.
- Number systems are basically classified into two types. They are:
  - Non-positional number system
  - Positional number system

➢ Non-Positional Number System

- In olden days people use of this type of number system for simple calculations like additions and subtractions.
- The non-positional number system consists of different symbols that are used to represent numbers.
- Roman number system is an example of the non-positional number system i.e. I=1, V=5, X=10, L=50.
- This number system cannot be used effectively to perform arithmetic operations.

➢ Positional Number System

- This type of number system are:
  - Decimal number system
  - Binary number system
  - Octal number system
  - Hexadecimal number system
- The total number of digits present in any number system is called its **Base** or **Radix**.
- Every number is represented by a base (or radix) x, which represents x digits.
- The base is written after the number as subscript such as $512_{(10)}$.It is a Decimal number as its base is 10.

- To determine the quantity that the number represents, the number is multiplied by an integer power of x depending on the position it is located and then finds the sum of the weighted digits.
- **Example**: Consider a decimal number $512.45_{(10)}$ which can be represented in equivalent value as: $5x10^{2} + 1x10^{1} + 2x10^{0} + 4x10^{-1} + 5x10^{-2}$

➢ Decimal Number System

- It is the most widely used number system.
- The decimal number system consists of 10 digits from 0 to 9.
- It has 10 digits and hence its base or radix is 10.
- These digits can be used to represent any numeric value.
- Example: $123_{(10)}$, $456_{(10)}$, $7890_{(10)}$.
- Consider a decimal number **$542.76_{(10)}$** which can be represented in equivalent value as: $5x10^{2} + 4x10^{1} + 2x10^{0} + 7x10^{-1} + 6x10^{-2}$

	Hundreds	Tens	Units	One-tenth	One-hundredth
Weights	10^{2}	10^{1}	10^{0}	10^{-1}	10^{-2}
Digits	5	4	2	7	6
Values	500	40	2	0.7	0.06

➢ Binary Number System

- Digital computer represents all kinds of data and information in the binary system.
- Binary number system consists of two digits 0 (low voltage) and 1 (high voltage).
- Its base or radix is 2.
- Each digit or bit in binary number system can be 0 or 1.
- The positional values are expressed in power of 2.
- Example: $1011_{(2)}$, $111_{(2)}$, $100001_{(2)}$
- Consider a binary number **$11011.10_{(2)}$** which can be represented in equivalent value as: $1x2^{4} + 1x2^{3} + 0x2^{2} + 1x2^{1} + 1x2^{0} + 0x2^{-1} + 0x2^{-2}$

Weights	2^{4}	2^{3}	2^{2}	2^{1}	2^{0}	2^{-1}	2^{-2}
Digits	1	1	0	1	1	1	0
Values	16	8	4	2	1	0.5	0.25

- **Note**: In the binary number $11010_{(2)}$
- The left most bit 1 is the highest order bit. It is called as **Most Significant Bit (MSB)**.
- The right most bit 0 is the lower bit. It is called as **Least Significant Bit (LSB)**.

Octal Number System

- The octal number system has digits starting from 0 to 7.
- The base or radix of this system is 8.
- The positional values are expressed in power of 8.
- Any digit in this system is always less than 8.
- Example: $123_{(8)}$, $236_{(8)}$, $564_{(8)}$
- The number 6418 is not a valid octal number because 8 is not a valid digit.
- Consider a Octal number **$234.56_{(8)}$** which can be represented in equivalent value as: $2x8^{2} + 3x8^{1} + 4x8^{0} + 5x8^{-1} + 6x8^{-2}$

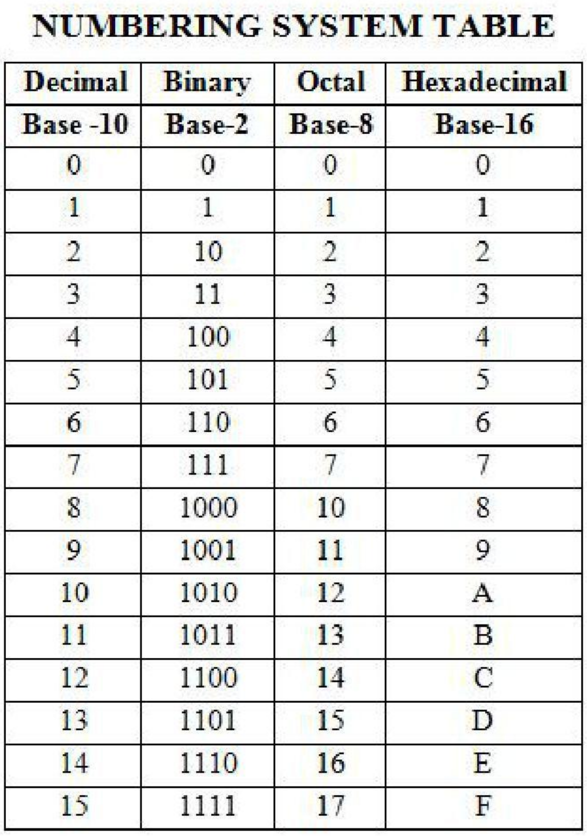
Weights	8^2	8^1	8^0	8^{-1}	8^{-2}
Digits	2	3	4	5	6
Values	64	8	1	0.125	0.015625

Hexadecimal Number System

- The hexadecimal number system consists of 16 digits from 0 to 9 and A to F.
- The letters A to F represent decimal numbers from 10 to 15.
- That is, ‘A’ represents 10, ‘B’ represents 11, ‘C’ represents 12, ‘D’ represents 13, ‘E’ represents 14 and ‘F’ represents 15.
- The base or radix of this number system is 16.
- Example: $A4_{(16)}$, $1AB_{(16)}$, $934_{(16)}$, $C_{(16)}$
- Consider a Hexadecimal number **$5AF.D_{(16)}$** which can be represented in equivalent value as: $5x16^{2} + Ax16^{1} + Fx16^{0} + Dx16^{-1}$

Weights	16^2	16^1	16^0	16^{-1}
Digits	5	A	F	D
Values	256	16	1	0.0625

Number System	Base	Symbol used
Binary	2	0, 1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F where A=10; B=11; C=12; D=13; E=14; F=15



NUMBERING SYSTEM TABLE

Decimal	Binary	Octal	Hexadecimal
Base -10	Base-2	Base-8	Base-16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

➢ Number System Conversions

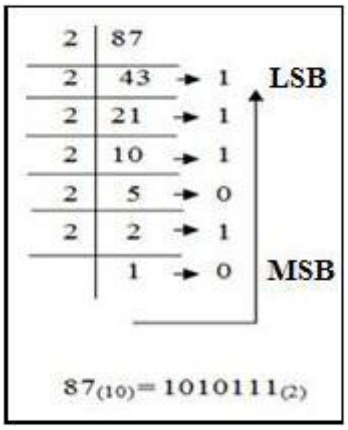
Conversion from Decimal to Binary:

1. Steps to convert decimal number to binary number:

- **Step 1**: Divide the given decimal number by 2.
- **Step 2**: Take the remainder and record it on the right side.
- **Step 3:** Repeat the Step 1 and Step 2 until the decimal number cannot be divided further.
- **Step 4**: The first remainder will be the LSB and the last remainder is the MSB. The equivalent binary number is then written from left to right i.e. from MSB to LSB.

Example: To convert the decimal number $87_{(10)}$ to binary.

- So 87 decimal is written as 1010111 in binary.
- It can be written as $87_{(10)} = 1010111_{(2)}$



2. Steps to convert decimal fraction number to binary number:

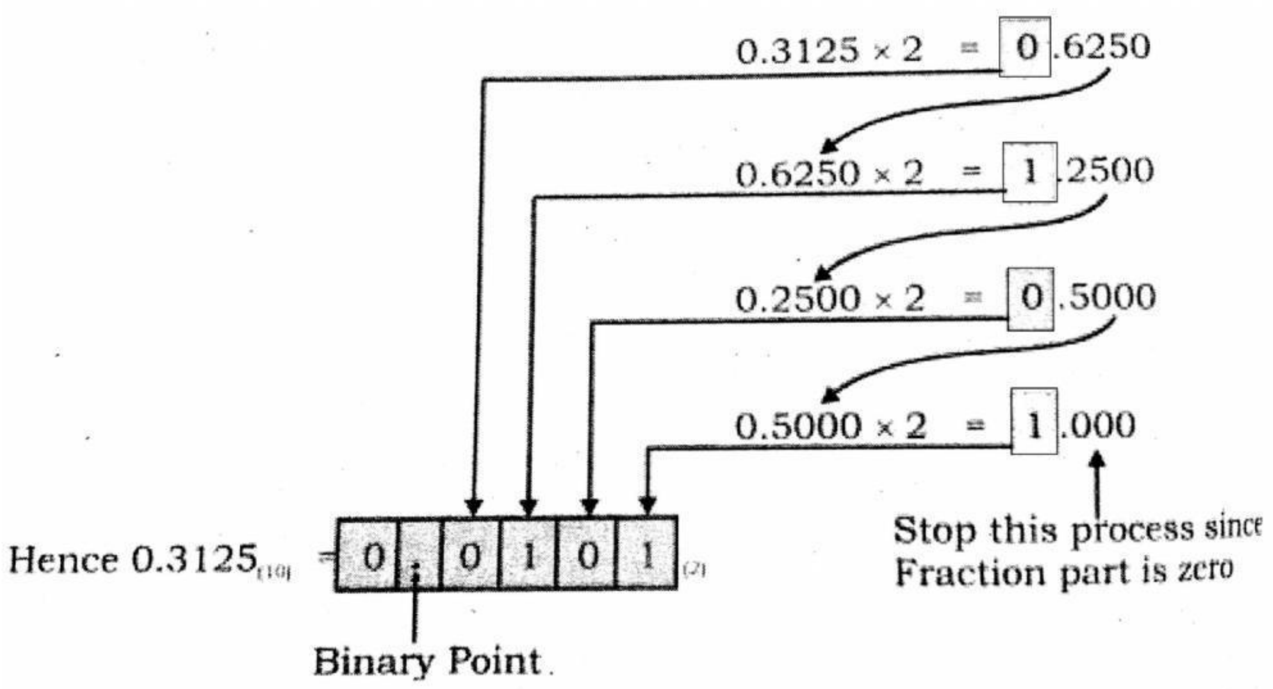
- **Step 1**: Multiply the given decimal fraction number by 2.
- **Step 2**: Note the carry and the product.
- **Step 3:** Repeat the Step 1 and Step 2 until the decimal number cannot be divided further.
- **Step 4**: The first carry will be the MSB and the last carry is the LSB. The equivalent binary fraction number is written from MSB to LSB.

Example 1: To convert the decimal number $0.3125_{(10)}$ to binary.

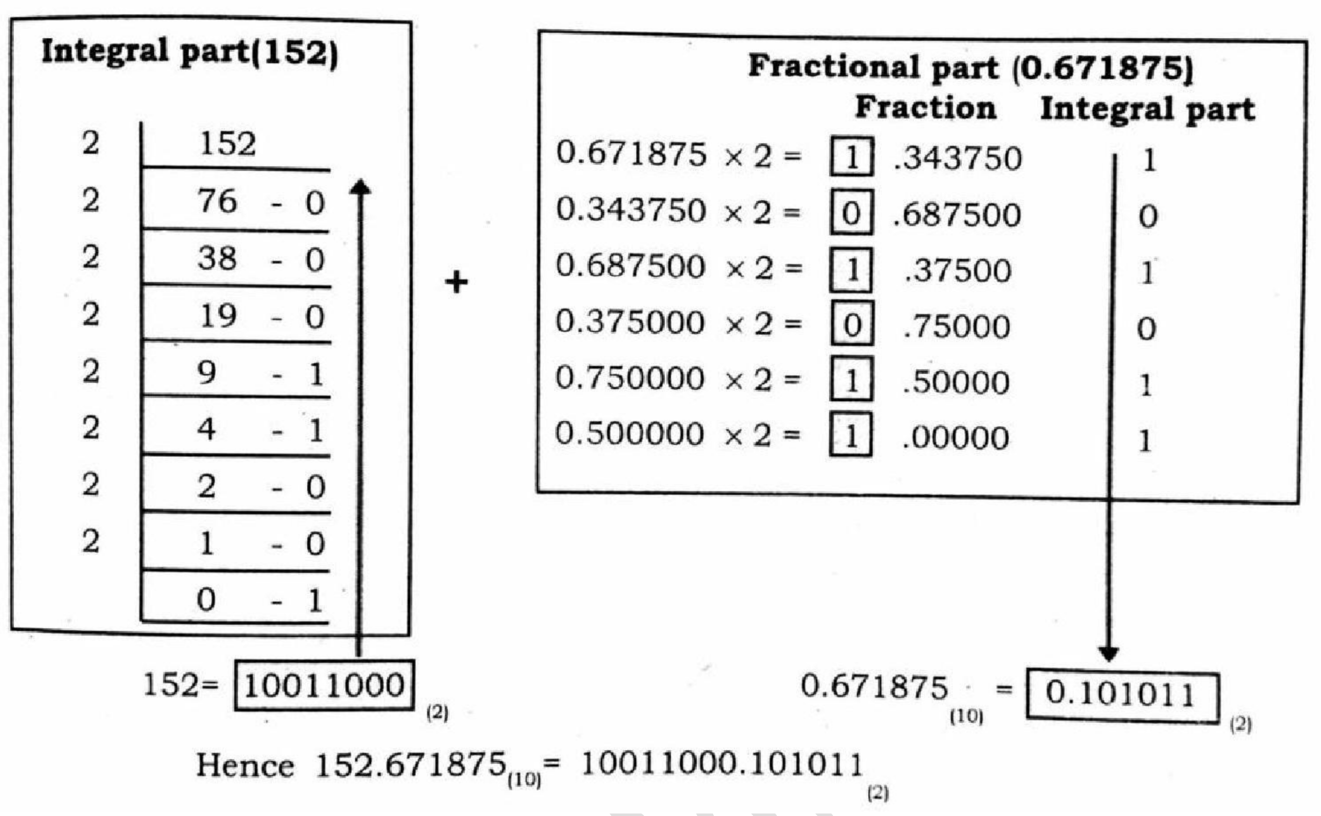
Multiply by 2	Carry	Product
0.3125 x 2	0 (MSB)	0.625
0.625 x 2	1	0.25
0.25 x 2	0	0.50
0.50 x 2	1 (LSB)	0.00
0.00		

- Therefore, $0.3125_{(10)} = 0.0101_{(2)}$

OR



Example 2: To convert the decimal number $152.671875_{(10)}$ to binary.



Integral part(152)

2	152
2	76 - 0
2	38 - 0
2	19 - 0
2	9 - 1
2	4 - 1
2	2 - 0
2	1 - 0
	0 - 1

$152 = \boxed{10011000}_{(2)}$

+

Fractional part (0.671875)

	Fraction	Integral part
0.671875 × 2 = $\boxed{1}$	.343750	1
0.343750 × 2 = $\boxed{0}$	.687500	0
0.687500 × 2 = $\boxed{1}$	.37500	1
0.375000 × 2 = $\boxed{0}$	.75000	0
0.750000 × 2 = $\boxed{1}$	.50000	1
0.500000 × 2 = $\boxed{1}$	.00000	1

$0.671875_{(10)} = \boxed{0.101011}_{(2)}$

Hence $152.671875_{(10)} = 10011000.101011_{(2)}$

3. Steps to convert binary number to decimal number

- **Step 1**: Start at the rightmost bit.
- **Step 2**: Take that bit and multiply by 2^n, when n is the current position beginning at 0 and increasing by 1 each time. This represents a power of two.
- **Step 3**: Then, add all the products.
- **Step 4**: After addition, the resultant is equal to the decimal value of the binary number.

Example 1: To convert the binary number $1010111_{(2)}$ to decimal.

$$\begin{aligned}
&= 1x2^6 + 0x2^5 + 1x2^4 + 0x2^3 + 1x2^2 + 1x2^1 + 1x2^0 \\
&= 1x64 + 0x32 + 1x16 + 0x8 + 1x4 + 1x2 + 1x1 \\
&= 64 + 16 + 4 + 2 + 1 \\
&= 87
\end{aligned}$$

- Therefore, $1010111_{(2)} = 87_{(10)}$

Example 2: To convert the binary number $11011.101_{(2)}$ to decimal.

$$= 1\text{x}2^4 + 1\text{x}2^3 + 0\text{x}2^2 + 1\text{x}2^1 + 1\text{x}2^0 + 1\text{x}2^{-1} + 0\text{x}2^{-2} + 1\text{x}2^{-3}$$

= 1x16 + 1x8 + 0x4+ 1x2 + 1x1 + 1x0.5+ 0x0.25+ 1x0.125

= 16 + 8 + 2 + 1 + 0.5 + 0.125

$= 27.625_{(10)}$

OR

Weights	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
Digits	1	1	0	1	1	1	0	1
Values	16	8	4	2	1	0.5	0.25	0.125

- Therefore, $11011.101_{(2)} = 27.625_{(10)}$

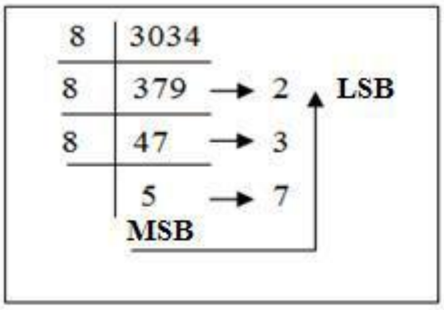
Conversion from Decimal to Octal

1. Steps to convert decimal number to octal number

- **Step 1:** Divide the given decimal number by 8.
- **Step 2:** Take the remainder and record it on the side.
- **Step 3:** Repeat the Step 1 and Step 2 until the decimal number cannot be divided further.
- **Step 4:** The first remainder will be the LSB and the last remainder is the MSB. The equivalent octal number is then written from left to right i.e. from MSB to LSB.

Example 1: To convert the decimal number $3034_{(10)}$ to octal number.

- So 3034 decimal is written as 5732 in octal.
- It can be written as $3034_{(10)} = 5732_{(8)}$



- **Note:** If the number is less than 8 the octal number is same as decimal number.

Example 2: To convert the decimal number $0.3125_{(10)}$ to octal number.

0.3125 x 8	**=**	**2.5000**	**2**
0.5000 x 8	**=**	**4.0000**	**4**

- Therefore, $0.3125_{(10)} = 0.24_{(8)}$

2. Steps to convert octal number to decimal number

- **Step 1:** Start at the rightmost bit.
- **Step 2:** Take that bit and multiply by 8^n, when n is the current position beginning at 0 and increasing by 1 each time. This represents the power of 8.
- **Step 3:** Then, add all the products.
- **Step 4:** After addition, the resultant is equal to the decimal value of the octal number.

Example 1: To convert the octal or base-8 number $5732_{(8)}$ to decimal

$$\boxed{\begin{aligned} &= 5x8^3 + 7x8^2 + 3x8^1 + 2x8^0 \\ &= 5x512 + 7x64 + 3x8 + 2x1 \\ &= 2560 + 448 + 24 + 2 \\ &= 3034 \end{aligned}}$$

- Therefore, $5732_{(8)} = 3034_{(10)}$

Example 2: To convert the octal number $234.56_{(8)}$ to decimal number.

$$= 2x8^2 + 3x8^1 + 4x8^0 + 5x8^{-1} + 6x8^{-2}$$
$$= 2x64 + 3x8 + 4x1 + 5x0.125 + 6x0.015625$$
$$= 128 + 24 + 4 + 0.625 + 0.09375$$
$$= 156.71875_{(10)}$$

OR

Weights	8^2	8^1	8^0	8^{-1}	8^{-2}
Digits	2	3	4	5	6
Values	64	8	1	0.125	0.015625

- Therefore, $234.56_{(8)} = 156.71875_{(10)}$

Conversion from Decimal to Hexadecimal

1. Steps to convert decimal number to hexadecimal number

- **Step 1**: Divide the decimal number by 16.
- **Step 2**: Take the remainder and record it on the side.
- **Step 3:** Repeat the Step 1 and Step 2 until the decimal number cannot be divided further.

- **Step 4:** The first remainder will be the LSB and the last remainder is the MSB. The equivalent hexadecimal number is then written from left to right i.e. from MSB to LSB.

Example To convert the decimal number $16242_{(10)}$ to hexadecimal

16	16242	
16	1015	$\rightarrow$ 2 **LSB**
16	63	$\rightarrow$ 7
	3 **MSB**	$\rightarrow$ 15 (i.e. F)

- So 16242 decimal is written as 3F72 in hexadecimal.
- It can be written as $16242_{(10)} = 3F72_{(16)}$
- **Note**: If the number is less than 16 the hexadecimal number is same as decimal number.

2. Steps to convert hexadecimal number to decimal number

- Step 1: Start at the rightmost bit.
- Step 2: Take that bit and multiply by 16^n, where n is the current position beginning at 0 and increasing by 1 each time. This represents a power of 16.
- Step 3: Then, add all the products.
- Step 4: After addition, the resultant is equal to the decimal value of the hexadecimal number.

Example 1: To convert the Hexadecimal or base-16 number 3F72 to a decimal number.

$$\begin{aligned}
&= 3x16^3 + Fx16^2 + 7x16^1 + 2x16^0 \\
&= 3x4096 + 15x256 + 7x16 + 2x1 \\
&= 12288 + 3840 + 112 + 2 \\
&= 16242
\end{aligned}$$

Therefore, $3F72_{(16)} = 16242_{(10)}$

Example 2: To convert the hexadecimal number $5AF.D_{(16)}$ to decimal number.

$$\begin{aligned}
&= 5x16^2 + 10x16^1 + 15x16^0 + 13x16^{-1} \\
&= 5x256 + 10x16 + 15x1 + 13x0.0625 \\
&= 1280 + 160 + 15 + 0.8125 \\
&= 1455.8125_{(10)}
\end{aligned}$$

OR

Weights	16^2	16^1	16^0	16^{-1}
Digits	5	A	F	D
Values	256	16	1	0.0625

- Therefore, $5AF.D_{(16)} = 1455.8125_{(10)}$

Conversion from Binary to Octal

Steps to convert Binary to octal

- Take a binary number in groups of 3 and use the appropriate octal digit in its place.
- Begin at the rightmost 3 bits. If we are not able to form a group of three, insert 0s to the left until we get all groups of 3 bits each.
- Write the octal equivalent of each group. Repeat the steps until all groups have been converted.

Example 1: Consider the binary number $1010111_{(2)}$

$$\underbrace{1}_{1}\qquad\underbrace{010}_{2}\qquad\underbrace{111}_{7}$$

Therefore, $1010111_{(2)} = 127_{(8)}$

Example 2: Consider the binary number $0.110111_{(2)}$

$$\underbrace{0}_{0}\qquad\underbrace{110}_{6}\qquad\underbrace{111}_{7}$$

Therefore, $0.110111_{(2)} = 0.67_{(8)}$

Example 3: Consider the binary number $1101.10111_{(2)}$

$$\underbrace{\mathbf{00}1}_{1}\qquad\underbrace{101}_{5}\qquad\underbrace{101}_{5}\qquad\underbrace{11\mathbf{0}}_{6}$$

Therefore, $1101.10111_{(2)} = 15.56_{(8)}$

Note: To make group of 3 bits, for whole numbers, it may be necessary to add a 0's to the left of MSB and when representing fractions, it may be necessary to add a 0's to right of LSB.

Conversion from Octal to Binary

Steps to convert octal to binary

- **Step 1**: Take the each digit from octal number
- **Step 2**: Convert each digit to 3-bit binary number. (Each octal digit is represented by a three-bit binary number as shown in Numbering System Table)

Octal digit	0	1	2	3	4	5	6	7
Binary Equivalent	000	001	010	011	100	101	110	111

Example 1: Consider the octal number $456_{(8)}$ into binary

4 → 100

5 → 101

6 → 110

Therefore, $456_{(8)} = 100101110_{(2)}$

Example 2: Consider the octal number $73.16_{(8)}$ into binary

7 → 100

3 → 101

1 → 001

6 → 110

Therefore, $73.16_{(8)} = 100101.001110_{(2)}$

Conversion from Binary to Hexadecimal

Steps to convert Binary to Hexadecimal

- Take a binary number in groups of 4 and use the appropriate hexadecimal digit in its place.
- Begin at the rightmost 4 bits. If we are not able to form a group of four, insert 0s to the left until we get all groups of 4 bits each.
- Write the hexadecimal equivalent of each group. Repeat the steps until all groups have been converted.

Example 1: Consider the binary number $1011001_{(2)}$

$$\underbrace{0101}_{5} \qquad \underbrace{1001}_{9}$$

Therefore, $1011001_{(2)} = 59_{(16)}$

Example 2: Consider the binary number $0.11010111_{(2)}$

$$\underbrace{0}_{0} \qquad \underbrace{1101}_{D} \qquad \underbrace{0111}_{7}$$

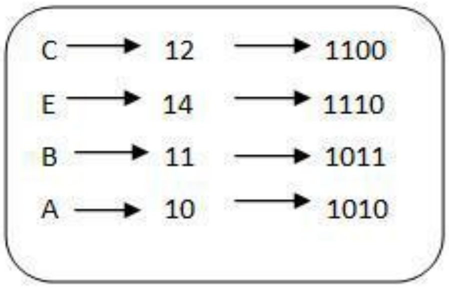
Therefore, $0.110111_{(2)} = 0.D7_{(16)}$

Conversion from Hexadecimal to Binary

Steps to convert hexadecimal to binary

- **Step 1**: Take the each digit from hexadecimal number
- **Step 2**: Convert each digit to 4-bit binary number. (Each hexadecimal digit is represented by a four-bit binary number as shown in Numbering System Table)

Example: Consider the hexadecimal number CEBA $_{(16)}$



$$\begin{tikzcd}
C \arrow[r] & 12 \arrow[r] & 1100 \\
E \arrow[r] & 14 \arrow[r] & 1110 \\
B \arrow[r] & 11 \arrow[r] & 1011 \\
A \arrow[r] & 10 \arrow[r] & 1010
\end{tikzcd}$$

Therefore, CEBA $_{(16)}$ = 1100 1110 1011 1010 $_{(2)}$

Conversion from Octal to Hexadecimal

Steps to convert Octal to Hexadecimal

Using Binary system, we can easily convert octal numbers to hexadecimal numbers and vice-versa

- **Step 1**: write the binary equivalent of each octal digit.
- **Step 2**: Regroup them into 4 bits from the right side with zeros added, if necessary.
- **Step 3**: Convert each group into its equivalent hexadecimal digit.

Example: Consider the octal number 274 $_{(8)}$

2 → **010**

7 → **111**

4 → **100**

Therefore, 274 $_{(8)}$ = 010 111 100 $_{(2)}$

Group the bits into group of 4 bits as **0** <u>1011 1100</u>

$$\underbrace{0}_{0} \qquad \underbrace{1011}_{B} \qquad \underbrace{1100}_{C}$$

Therefore, 274 $_{(8)}$ = BC $_{(8)}$

Conversion from Hexad ecimal to Octal

Steps to convert Hexadecima l to Octal

- **Step 1**: write the binary equivalent of each hexadecimal digit.
- **Step 2**: Regroup them into 3 bits from the right side with zeros added, i f necessary.
- **Step 3**: Convert each grou p into its equivalent octal digit.

Example: Consider the hexad ecimal number FADE $_{(16)}$

F	→	**1111**
A	→	**1010**
D	→	**1101**
E	→	**1110**

Therefore, FADE $_{(16)}$ = 1111 1010 1101 1110 $_{(2)}$

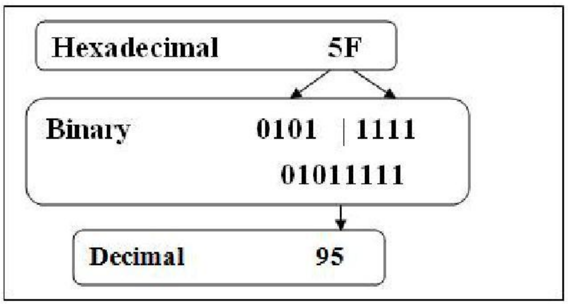
Group the bits into group of 3 bits from LSB as $\underline{\textbf{00}1}\ \ \underline{111}\ \underline{101}\ \underline{011}\ \ \underline{011}\ \underline{110}$

$$\underbrace{001}_{1}\qquad\underbrace{111}_{7}\qquad\underbrace{101}_{5}\qquad\underbrace{011}_{3}\qquad\underbrace{011}_{3}\qquad\underbrace{110}_{6}$$

Therefore, FADE $_{(16)}$= 175336 $_{(8)}$

Note

Shortcut method to convert hexadecimal to decimal number



Binary Arithmetic

- It involves addition, subtraction, multiplication and division operations.
- Binary arithmetic is much sim pler to learn because system deals with only two digit 0's and 1's.
- When binary arithmetic operations are performed on binary numbers, the results are also 0's and 1's.

Binary Addition

- The addition of two binary numbers is performed in same manner as the addition of decimal number.
- The basic rules of binary addition are:

Addend1	Addend2	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- It adds only two bits and gives sum and carry. If a carry is generated, it should be carried over to the addition of next two bits.
- **Example 1**: Add the following number

a) 4 and 3

Decimal	Binary
4	1 0 1
+ 3	+ 0 1 1
7	1 1 1

b) 9 and 5

Decimal	Binary
9	1 0 0 1
+ 5	+ 1 0 1
14	1 1 1 0

- **Example 2**: Add 75 and 18 in binary number.

Convert this decimal number into binary i.e.

2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1
1	0	0	1	0	1	1

75= 64 + 8 + 2 + 1 = 1001011

18= 16 + 2 = 10010

Carry	→	1
Append 1	→	1 0 0 1 0 1 1
Append 2	→	1 0 0 1 0
Sum	→	1 0 1 1 1 0 1

- **Example 3**: Add binary number 1011.011 and 1101.111

Carry	→	1 1 1 1 1 1
Append 1	→	1 0 1 1 . 0 1 1
Append 2	→	1 1 0 1 . 1 1 1
Sum	→	1 1 0 0 1 . 0 1 0

Exercise:
1) Add 10101 and 11011
2) Add 1011101 and 1100111

➢ Binary Subtraction

- This operation is same as the one performed in the decimal system.
- This operation is consists of two steps:
  - o Determine whether it is necessary for us to borrow. If the subtrahend (the lower digit) is larger than the minuend (the upper digit), it is necessary to borrow from the column to the left. In binary two is borrowed.
  - o Subtract the lower value from the upper value
- The basic rules of binary subtraction are:

Minuend	Subtrahend	Difference	Barrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- When we subtract 1 from 0, it is necessary to borrow 1 from the next left column i.e. from the next higher order position.
- **Example 1**: Subtract the following number

  a) 10 from 14 b) 9 from 29 c) 3 from 5

Decimal	Binary	Decimal	Binary	Decimal	Binary
14	1 1 1 0	29	1 1 1 0 1	5	1 0 1
- 10	- 1 0 1 0	- 09	- 0 1 0 0 1	- 3	- 0 1 1
4	0 1 0 0	20	1 0 1 0 0	2	0 1 0

- **Example 2**: Add 75 and -18 in binary number.

  75 = 64 + 8 + 2 + 1 = 1001011

  18 = 16 + 2 = 10010

Borrow	→		
Minuend	→	1 0 0 1 0 1 1	75
Subtrahend	→	1 0 0 1 0	- 18
Difference	→	0 1 1 1 0 0 1	57

- **Example 3**: Subtract binary number 0110100 from 1011000

Binary		Decimal
2 2	← borrow	
$\cancel{1}011\cancel{1}000_{(2)}$		88_{10}
$0110100_{(2)}$		52_{10}
$0100100_{(2)}$		36_{10}

Exercise:
1) Subtract 01110 from 10101
2) Subtract 25 from 35

➢ Representation of signed Integers

- The digital computer handle both positive and negative integer.
- It means, is required for representing the sign of the number (- or +), but cannot use the sign (-) to denote the negative number or sign (+) to denote the positive number.
- So, this is done by adding the leftmost bit to the number called **sign bit**.
- A positive number has a sign bit 0, while the negative has a sign bit 1.
- It is also called as **fixed point representation**.
- A negative signed integer can be represented in one of the following:
    1) Sign and magnitude method
    2) One's complement method
    3) Two's complement method

➢ Sign and magnitude method

- An integer containing a sign bit followed by magnitude bits are called **sign-magnitude integer**.
- In this method, first bit (MSB) is considered as a sign bit and the remaining bits stand for magnitude.
- Here positive number starts with 0 and negative number starts with 1.
- **Example**: Consider a decimal number $25_{(10)}$

2	25	
2	12	→1
2	6	→0
2	3	→0
	1	→1

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1
1	1	0	0	1

- So the binary number is $11001_{(2)}$. If we take the size of the word is 1 byte(8 bits), then the number 25 will be represented as

0	0	0	1	1	0	0	1

Sign Bit Magnitude = 25

- Suppose, if the number is -25, and then it will be represented as:

1	0	0	1	1	0	0	1

Sign Bit Magnitude = 25

➢ 1’s Complement representation

- This is the simplest method of representing negative binary number.
- The 1’s complement of a binary number is obtained by changing each 0 to 1 and each 1 to 0.
- In other words, change each bit in the number to its complement.
- **Example 1**: Find the 1’s complement of 101000.

  Original binary number → 1 0 1 0 0 0

  ↓ ↓ ↓ ↓ ↓ ↓

  Find 1’s Complement → 0 1 0 1 1 1

  Thus 1’s complement of 101000 is 010111.

- **Example 2**: Find the 1’s complement of 1010111.

  Original binary number → 1 0 1 0 1 1 1

  ↓ ↓ ↓ ↓ ↓ ↓ ↓

  Find 1’s Complement → 0 1 0 1 0 0 0

  Thus 1’s complement of 101000 is 010111.

➢ 2’s Complement representation

- The 2’s complement of a binary number is obtained by taking 1’s complement of the number and adding 1 to the Least Significant Bit (LSB) position.
- The general procedure to find 2’s complement is given by:

2’s Complement = 1’s Complement + 1

- **Example 1**: Find the 2’s complement of 101000.

  Original binary number → 1 0 1 0 0 0

  ↓ ↓ ↓ ↓ ↓ ↓

  Find 1’s Complement → 0 1 0 1 1 1

  Add 1 to LSB + 1

  Hence 2’s Complement of 101000 is → 0 1 1 0 0 0

- **Example 2**: Find the 1’s and 2’s complement of 1011101.

  Original binary number → 1 0 1 1 1 0 1

  1’s Complement → 0 1 0 0 0 1 0

  + 1

  Hence 2’s Complement is → 0 1 0 0 0 1 1

Note:

Negation: It is the operation of converting a positive number to its negative equivalent or a negative number to its positive equivalent. Negation is performed by performing 2's complement system.

- **Example 1**: Consider the number +12. Its binary representation is $01100_{(2)}$. Find the 2's complement of +12.

```
Original binary number    →    0 1 1 0 0
                               ↓ ↓ ↓ ↓ ↓
Find 1's Complement       →    1 0 0 1 1
Add 1 to LSB                   +       1
                               ---------
2's Complement            →    1 0 1 0 0
```

Clearly, this is a negative number since the sign bit is 1. Actually, 10100 represent $-12_{(10)}$ which is the negative equivalent of the number $12_{(10)}$

- **Example 2**: Consider the number -12. Its binary representation is $10100_{(2)}$. Find the 2's complement of -12.

```
Original binary number    →    1 0 1 0 0
                               ↓ ↓ ↓ ↓ ↓
Find 1's Complement       →    0 1 0 1 1
Add 1 to LSB                   +       1
                               ---------
2's Complement            →    0 1 1 0 0
```

Clearly, this is a positive number since the sign bit is 0. Actually, 01100 represent $12_{(10)}$ which is the negative equivalent of the number $-12_{(10)}$

➢ Subtraction of Binary Number using Complement

- Most of the computers perform subtraction using complemented number. This is less expensive because the same addition circuit is used for subtraction with slight changes in the circuit.
- In the binary number system we can perform subtraction operation using two methods of complements:
    - Subtraction using 1's Complement
    - Subtraction using 2's Complement

➢ Subtraction using 1's Complement

- **Case 1**: Subtracting a smaller number from a larger number (Minuend is greater than Subtrahend)
  - **Step 1**: Find the 1's complement of the subtrahend.
  - **Step 2**: Add this to the minuend.
  - **Step 3**: Carry is generated, this carry is called as the end around carry
  - **Step 4**: Add the end around carry back to the LSB to get the final difference.

- **Example 1**: Subtract 15 from 23 using 1's complement.

	Decimal	Binary
Minuend	23	10111
Subtrahend	-15	01111

1's complement of subtrahend is 10000

Minuend	→	1 0 1 1 1
1's Complement of subtrahend	→	+ 1 0 0 0 0
End around carry	→	**1** 1 0 1 1 1
Add end around carry	→	+ 1
Difference is	→	1 0 0 0

- **Case 2**: Subtracting a larger number from a smaller number (Minuend is less than Subtrahend)
  - **Step 1**: Find the 1's complement of the subtrahend.
  - **Step 2**: Add this to the minuend.
  - **Step 3**: There will be no carry, Re complement the answer to get the difference

- **Example 1**: Subtract 52 from 25 using 1's complement.

	Decimal	Binary
Minuend	25	011001
Subtrahend	-52	110100

1's complement of subtrahend is 001011

Minuend	→	0 1 1 0 0 1
1's Complement of subtrahend	→	+ 0 0 1 0 1 1
	→	1 0 0 1 0 0
Since there is no carry		↓ ↓ ↓ ↓ ↓ ↓
take 1's complement and	→	0 1 1 0 1 1
attach a negative sign		Hence, the result = - 011011 i.e. - 27

Subtraction using 2’s Complement

- **Case 1**: Subtracting a smaller number from a larger number (Minuend is greater than Subtrahend)
  - **Step 1**: Find the 2’s complement of the subtrahend.
  - **Step 2**: Add this to the minuend.
  - **Step 3**: Carry is generated, Discard the carry and the remaining bits give the difference.

- **Example 1**: Subtract 09 from 17 using 2’s complement.

	Decimal	Binary
Minuend	17	10001
Subtrahend	-09	01001

```
1’s complement of subtrahend (9) is 1 0 1 1 0  ┐
Add 1 to LSB                        +       1  ├  STEP 1
2’s complement of 9 is              1 0 1 1 1  ┘

Minuend                      →      1 00 0 1   ┐
2’s Complement of subtrahend →    + 1 0 1 1 1  ┘  STEP 2
End around carry             →    1 0 10 0 0
                                  └──→ Discard the carry }   STEP 3
Difference is                →      10 0 0
```

Hence, the result = 1000 i.e. 8

- **Case 2**: Subtracting a larger number from a smaller number (Minuend is less than Subtrahend)
  - **Step 1**: Find the 2’s complement of the subtrahend.
  - **Step 2**: Add this to the minuend.
  - **Step 3**: There will be no carry, hence take the 2’s complement of the answer and place a negative sign in front.

- **Example 1**: Subtract 47 from 26 using 2’s complement.

	Decimal	Binary
Minuend	26	011010
Subtrahend	-47	101111

```
1’s complement of subtrahend (47) is 0 1 0 0 0 0  ┐
Add 1 to LSB                         +          1  ├  STEP 1
2’s complement of 9 is               0 1 0 0 0 1  ┘
```

$$\left.\begin{array}{lcr} \text{Minuend} & \rightarrow & 0\ 1\ 1\ 0\ 1\ 0 \\ \text{2's Complement of subtrahend} & \rightarrow & +0\ 1\ 0\ 0\ 0\ 1 \end{array}\right\} \textbf{STEP 2}$$

$$\begin{array}{r} \hline 1\ 0\ 1\ 0\ 1\ 1 \end{array}$$

$$\left.\begin{array}{lcr} \text{1's Complement of answer} & \rightarrow & 0\ 1\ 0\ 1\ 0\ 0 \\ \text{2's Complement of answer} & \rightarrow & 0\ 1\ 0\ 1\ 0\ 0 \\ & & +\qquad\quad 1 \end{array}\right\} \textbf{STEP 3}$$

$$\begin{array}{r} \hline 0\ 1\ 0\ 1\ 0\ 1 \end{array}$$

Hence, the result = 10101 i.e. 21

➢ Computer Codes

- Computer code helps us to represent characters in a coded form in the memory of the computer.
- These codes represent specific formats which are used to record data.
- Some of the commonly used computer codes are:
  - Binary Coded Decimal (BCD)
  - Extended Binary Coded Decimal Interchange Code (EBCDIC)
  - American Standard Code for Information Interchange (ASCII)
  - Excess-3 Code.

➢ BCD code (or Weighted BCD Code or 8421 Code)

- BCD stands for **Binary Coded Decimal.**
- It is one of the early computer codes.
- In this coding system, the bits are given from left to right, the weights 8,4,2,1 respectively.
- The BCD equivalent of each decimal digit is shown in table.

Decimal Digit	BCD Equivalent	Decimal Digit	BCD Equivalent
0	0000	5	0101
1	0001	6	0110
2	0010	7	0111
3	0011	8	1000
4	0100	9	1001

- **Example**: Convert the decimal number 537 into BCD.

$$\underbrace{5}_{0101}\qquad\underbrace{3}_{0110}\qquad\underbrace{7}_{0111}$$

Hence, the BCD of 537 is 010101100111

- In 4-bit BCD only **2^4=16** configurations are possible which is insufficient to represent the various characters.
- Hence 6-bit BCD code was developed by adding two zone positions with which it is possible to represent **2^6=64** characters.

➢ Excess-3 BCD code (or XS-3 Code)

- The Excess-3 BCD code is a non-weighted code used to express decimal number**.**
- The name Excess-3 code derived from the fact that each binary code is the corresponding **BCD code plus $0011_{(2)}$**(i.e. Decimal 3).
- This code is used in some old computers.
- The following table gives the Excess-3 code equivalent of decimal (0-9).

Decimal Number	BCD Number	Adding 3 to BCD	Excess-3 Equivalent
0	0000	11	0011
1	0001	11	0100
2	0010	11	0101
3	0011	11	0110
4	0100	11	0111
5	0101	11	1000
6	0110	11	1001
7	0111	11	1010
8	1000	11	1011
9	1001	11	1100

- **Example**: Find Excess-3(XS3) representation of decimal number 537.

5	3	7	→	Decimal Digit
0101	0110	0111	→	8421 BCD Code
0011	0011	0011	→	Add 3
1000	1010	0110	→	Excess-3 Code

➢ EBCDIC

- It stand for Extended Binary Coded Decimal Interchange Code.
- This was developed by IBM.
- It uses an 8-bit code and hence possible to represent 256 different characters or bit combinations.

- EBCDIC is used on most computers and computer equipment today.
- It is a coding method generally used by larger computers (mainframes) to present letters, numbers or other symbols in a binary language the computer can understand.
- EBCDIC is an 8-bit code; therefore, it is divided into two 4-bit groups, where each 4-bit can be represented as 1 hexadecimal digit.

➢ ASCII

- It stands for the **American Standard Code for Information Interchange**.
- It is a 7-bit code, which is possible to represent 2^7=128 characters.
- It is used in most microcomputers and minicomputers and in mainframes.
- The ASCII code (Pronounced *ask-ee*) is of two types – ASCII-7 and ASCII-8.
- ASCII-7 is 7-bit code for representing English characters as numbers, with each letter assigned a number from 0 to 127.
- **Example**: The ASCII code for uppercase M is 77.

****

CHAPTER 3 – DATA REPRESENTATION BLUE PRINT				
VSA (1 marks)	SA (2 marks)	LA (3 Marks)	Essay (5 Marks)	Total
-	-	01 Question	01 Question	08 Marks

CHAPTER 3

DATA REPRESENTATION

Review Questions

Very short answer type questions:

1. Define the base of the number system.
   Ans: The total number of elements present in the number system is called base of the number system.

2. What is the expansion of BIT?
   Ans: BIT means Binary DigIT.

3. Define MSB.
   Ans: MSB stands for Most Significant Bit.

4. Define LSB.
   Ans: LSB stands for Least Significant Bit.

5. What is the weight of the LSB of an 8-bit number?
   Ans: $2^0 = 1$

6. What is the weight of the MSB of an 16-bit number?
   Ans: 32768

7. What does BCD stand for?
   Ans: BCD stands for Binary Coded Decimal.

8. What is the expansion of ASCII?
   Ans: ASCII stands for American Standard Code for Information Interchange.

9. What is the expansion of EBCDIC?
   Ans: EBCDIC stands for Extended Binary Coded Decimal Interchange Code.

10. What is binary system?
    Ans: The number system which has base 2 is called binary system.

11. What is octal system?
    Ans: The number system which has base 8 is called octal system.

12. What is hexa-decimal system?
    Ans: The number system which has base 16 is called hexa-decimal system.

13. How are negative numbers represented?
    Ans: The negative number represented in binary system by adding extra digit 1 to the front of binary number.

14. Write 1's complement of 11010111(2).
    Ans: 11010111(2) = 00101000.

15. Write 2's complement of 11011011(2).
Ans: 11011011(2) = 00100100 + 1 = 00100101.

<u>Short answer type questions:</u>

1. Specify the rule for representing number using positional notation in any number system.
   Ans: In positional notation every number is represented by base x, which represents x digits. To represent the quantity of the number, it is multiplied by an integer power of x.

2. Mention the different types of positional number system.
   Ans: The different types of positional number systems are:
   - Decimal number system
   - Binary number system
   - Octal number system
   - Hexa-decimal number system

3. Explain the need of binary system in computers.
   Ans: the binary system contains 0 and 1, the computer only understands binary language i.e machine level language , so binary system is required in computers.

4. What is the importance of hexa-decimal system?
   Ans: the memory address is an hexa decimal number. The computer stores and retrieves data from the memory using its address, which can be represented using hexa decimal numbers.

5. What is 1's complement? Give an example>
   Ans: The 1's complement of binary number can be obtained by changing 1 to 0 or 0 to 1.
   Example: 25 = 11001= 00110.

6. What is 2's complement? Give an example.
   Ans: The 2's complement of binary number can be obtained by finding 1's complement of the number and adding 1 to LSB.
   Example: 39 = 100111 = 011000 + 1 = 011001

7. What are computer codes? Give an example.
   Ans: It is a set of symbols for representing characters.
   Example: ASCII codes.

8. Mention the different types of number systems.
   Ans: The different types are:
   - Positional number system
   - Non-positional number system.

9. What is the use of binary number system over decimal number system?
   Ans: since computer can directly understand the binary numbers as compared to decimal

numbers which has to be converted to binary to be used in the system.

10. Convert $97.188_{(10)}$ to binary.

Ans:

```
2 | 97                              0.188 *2=0.376
  |-----                            0.376 *2=0.752
2 | 48  --------  1                 0.752 *2=1.504
  |-----                            0.504 *2=1.008
2 | 24  ---------  0
2 | 12  ---------  0
2 | 6   ---------  0
2 | 3   --------   0
     1  ---------  1
```

$97.188_{(10)} = 1100001.0011_{(2)}$

11. Convert $728.45_{(10)}$ to binary.

Ans:

```
2 | 728 ----- 0
2 | 364 ----- 0        2 | 5 --------- 1       0.45 *2=0.9
2 | 182 ----- 0        2 | 2 -------- 1            0.9 *2=1.8
2 | 91  ----- 0             1 -------- 0        0.8 *2=1.6
2 | 45  ------ 1                                 0.6 *2=1.2
2 | 22  ----- 1
2 | 11  ----- 0
```

$728.45_{(10)} = 10110110000.0111_{(2)}$

12. Convert $1101111.101_{(2)}$ to decimal

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

$1 * 64 + 1 * 32 + 0 * 16 + 1 * 8 + 1 * 4 + 1*2 + 1*1 + 1*0.5 + 0*0.25 + 1 * 0.125$

$64 + 32 + 0 + 8 + 4 + 2 + 1 + 0.5 + 0 + 0.125$

111.625

13. Convert $2835_{(16)}$ to decimal.

Ans: $2*16^3 + 8*16^2 + 3*16^1 + 5*16^0$

2*4096+8*256+48+5

8192+2048+48+5

$10293_{(10)}$

14. Convert $789_{(10)}$ to octal.

Ans:

```
8 |  789
8 |   98    --- 5
8 |   12    --- 2
       1    ---  4
```

$789_{(10)} = 1425_{(8)}$

15. Convert 4563 in octal to binary.

Ans: 4563

4 = 100

5 = 101

6 = 110

3 = 011

4563 = $100101110011_{(2)}$

16. Convert $BED_{(16)}$ to binary.

Ans: B = 11 = 1011

E = 14 = 1110

D = 13 = 1101

BED = $101111101101_{(2)}$

17. Convert 1101.01101 in binary to octal.

Ans: $15.32_{(8)}$

18. Convert A492.B in hexadecimal to decimal.

Ans: $10*16^3+4*16^{2+}9*16^1+2*16^0.\ 11*16^{-1}$

40960+1024+144+2 . 68

$42130.68_{(10)}$

19. Convert $512.45_{(10)}$ to hexadecimal.

Ans:

```
16 | 512                         0.45 *16 = 7.2
16 |  32   --- 0                 0.2 * 16 = 3.2
   |   2   --- 0
```

$512.45_{(10)} = 200.73_{(16)}$

20. Convert $11011110_{(2)}$ to hexadecimal.

Ans: 1101 1110

13 14

D E

11011110 = $DE_{(16)}$

21. Convert $6A9.ABC_{(16)}$ to binary.

Ans:

011010101001.101010111100

22. Add 1010101 and 1010111.

Ans:

```
  1 0 1 0 1 0 1
  1 0 1 0 1 1 1
---------------
1 1 1 0 1 1 1 0
```

Long answer type question:

1. Give the Radix of:

   Ans: (a) Decimal system = Base 10

   (b) Binary system = Base 2

   (c) Octal system = Base 8

   (d) Hexa-decimal = Base 16

2. Explain 1's and 2's complement with example.

   Ans: ***1's complement:*** The negative binary number is formed by subtracting the value of each bit in the word from 1. This changes the value of each bit from 0 to 1 and 1 to 0.

   Example: Consider the binary number 101000

   101000

   Ans: 010111

   2's complement: The 2's complement of a binary number is obtained by adding 1 to the 1's complement of the binary number.

   Example: Consider the binary number 101000

   101000

   Ans: 010111 + 1 = 011000

3. Subtract 36 from 83 using 2's complement.

   Ans: 83 – 36

   83 – 1010011 – Minuend

   36 – 0100100 – Subtrahend

   Step 1: Take 2's complement of subtrahend.

   0100100

   1011011 + 1 = 1011011

   Step 2: Add with minuend

   1011011 + 1010011 = (1)0101110

   Step 3: If we get carry, discard the carry…

   ~~(1)~~ 0101110 = 0101110

   Ans: 0101110

4. Using 1's complement method, solve $54_{(10)} – 87_{(10)}$

   Ans: 54 – 0110110 – Minuend

   87 – 1010111 – Subtrahend

   Step 1: Take 1's complement of subtrahend.

   1010111

   0101000

   Step 2: Add with minuend

   0101000 + 0110110 = 1011110

   Step 3: No carry, discard the carry take 1's complement of the subtrahend and add –^{ve} symbol.

   1011110 = - 0100001

   Ans: - 0100001

5. Using 2's complement method, solve $73_{(10)} – 25_{(10)}$

   Ans: 73 – 1001001 – Minuend

25 – 0110011 – Subtrahend

Step 1: Take 2’s complement of subtrahend.
0110011
1001100 + 1 = 1001101
Step 2: Add with minuend
1001101 + 1001001 = 10010110
Step 3: If we get carry, discard the carry…
$\cancel{(1)}$ 0010110 = 0010110
Ans: 0010110

6. Add: $64_{(10)} + 35_{(10)}$ using binary addition.
Ans: 64 = 1000000
35 = 0100011
1000000 + 0100011 = 1100011

Essay type questions:

1. Find $11001001.1011_{(2)} = (?)_{(8)} = (?)_{(16)}$
Ans: $11001001.1011_{(2)}$ = $311.54_{(8)}$ = $C9.B_{(16)}$

$$\underbrace{0\,1\,1}_{3}\;\underbrace{0\,0\,1}_{1}\;\underbrace{0\,0\,1}_{1}\,.\,\underbrace{1\,0\,1}_{5}\;\underbrace{1\,0\,0}_{4}$$

$$\underbrace{1\,1\,0\,0}_{C}\;\underbrace{1\,0\,0\,1}_{9}\,.\,\underbrace{1\,0\,1\,1}_{B}$$

2. Find $FADE_{(16)} = (?)_{(8)} = (?)_{(10)}$
Ans: F = 15 – 1111
A = 10 – 1010
D = 13 – 1101
E = 14 – 1110
$FADE_{(16)} = 1111101011011110_{(2)}$

001 111 101 011 011 110
1 7 5 3 3 $6_{(8)}$

$1*8^5+7*8^4+5*8^3+3*8^2+3*8^1+6*8^0$
1*32760+7*4096+5*512+3*64+3*8+6*1
32760+28672+2560+192+24+6
$64214_{(10)}$

3. Explain different types of computer codes.
Ans: Types of computer codes are:
- ***BCD (Binary Coded Decimal):*** BCD stands for Binary Coded Decimal or 8421 coding.In a 4 – bit BCD there are 2^4 = 16 combinations and it is used to represent 2^6 = 64 characters.
- ***Excess 3 BCD Code:*** This is obtained from BCD code by adding 3 (0011) to the BCD number.
  Example:

5
0101 + 0011 = 1000

- ***High version of Excess 3 BCD:***
    1. ***EBCDIC:*** Extended Binary Coded Decimal Interchange Code [2^8 =256 characters]. It supports all the 256 characters.
- ***ASCII:*** American Standard Code for Information Interchange. It's a 7 bit coding system which supports 128 characters. It is used to represent Alpha numeric.
  Example: ASCII values
  A= 65   B = 92
  a = 92   b = 93

4. Evaluate: $BEAD_{(16)} = (?)_{(10)} = (?)_{(2)} = (?)_{(8)}$
   Ans:
   $11*16^3+14*16^2+10*16^1+13*16^0$
   11*4096+14*256+10*16+13*1
   45056+3584+160+13
   $48813_{(10)}$
   $48813_{(10)} = 10111110101011_{(2)}$

   010 111 110 101 011
   2 7 6 5 3
   $27653_{(8)}$
5. Subtract: $25_{(10)} – 14_{(10)}$ using 1's and 2's complement.
   Ans: 25 = 11001 – Minuend
   14 = 01110 – Subtrahend

1's complement:

Step 1: Take 1's complement of the subtrahend.
01110
10001
Step 2: Add with minuend.
10001 + 11001 = 101010
Step 3: No carry, take 1's complement of the result and add $–^{ve}$ symbol.
101010 = - 010101
Ans: -010101

2's complement:

Step 1: Take 2's complement of the subtrahend.
01110
10001 + 1 = 10010
Step 2: Add with minuend.
10010 + 11001 = 101011
Step 3: No carry, take 2's complement of the result and add $–^{ve}$ symbol.

101011 + 1 = - 010101

Ans: -010101

6. Write a brief note on computer codes. (E.Q.3)

7. Write a note on ASCII code.

   Ans: ***ASCII:*** American Standard Code for Information Interchange. It's a 7 bit coding system which supports 128 characters. It is used to represent Alpha numeric.

   Example: ASCII values

   A= 65   B = 92

   a = 92   b = 93

Assignment:

I. Solve the following

   a. Convert Decimal to Binary, Octal and Hexa-decimal
      1. 97
      2. 879
      3. 256
      4. 930
      5. 606

   b. Convert Binary to decimal
      1010001, 1000110, 1110001, 1010101, 1001

   c. Convert Octal to decimal
      456, 157, 324, 64, 101

   d. Convert Hexa decimal to decimal
      56A, E214, ACD, BCA, BEAD, DEAF

   e. Convert the following
      BACD = -------(2) ----------(8) -------------(10)
      DEAF = -------(2) ----------(8) -------------(10)
      FEED = -------(2) ----------(8) -------------(10)

II. Find the 1's Complement of the following
    34, 132, 345, 675, 44, 66

III. Find the 2's Complement of the following
     34, 132, 345, 675, 44, 66