UNIT-3
Understand the Number Systems

Number systems:

Number system contains a set of numbers that have common characteristics. In any number system; three aspects determine the value of each digit with in a number.

- ✓ The digit itself.
- ✓ The position of the digit in that number
- ✓ The base of the number system

3.1 various number systems used in digital computer:

The most important number systems for computer are:

- Decimal number system
- Binary number system
- Octal number system
- Hexadecimal number system

Radix(or) Base of a number:

The base of a number system is the number of different symbols or digits used in that system.

1)Decimal number system:

- ✓ The number system that we use in our day-to-day life is called the decimal number system.
- ✓ It is positional number system and is the mostly widely used system for representing numbers.
- ✓ In decimal number system the BASE is 10,because it uses 10 different symbols or digits(i.e.) 0,1,2,3,4,5,6,7,8,9 (0 to 9).

Example: The decimal number 1234(written as $(1234)_{10}$) represents the Polynomial.

POSITION	3	2	1	0
VALUE(WEIGHTS)	10^3	10^2	10^1	10^0
DECIMAL POINT	1	2	3	4

$=(1X10^3) + (2X10^2) + (3X10^1) + (4X10^0)$

$=(1X1000) + (2X100) + (3X10) + (4X1)$

$=1000 + 200 + 30 + 4$

$=1234$

$=(1234)_{10}$

2) Binary number system:

- ✓ Binary number system is a very important number system so far as the digital computers concerned, because computers work with binary numbers.
- ✓ Binary number system is the positional binary system.
- ✓ The BASE of Binary number system is 2 and it uses two digits (i.e.) 0 & 1.
- ✓ These symbols or digits are abbreviated as bits. Thus the word bit stands for either of the binary digits namely 0 or 1.

Example: the decimal equivalent of the binary number 11010 is

POSITION	4	3	2	1	0
VALUE(WEIGHTS)	2^4	2^3	2^2	2^1	2^0
BINARY POINT	1	1	0	1	0

$(11010)_2 = (1X2^4) + (1X2^3) + (0X2^2) + (1X2^1) + (0X2^0)$

$= (1X16) + (1X8) + (0X4) + (1X2) + (0X1)$

$=16 + 8 + 0 + 2 + 0$

$=26$

$(11010)_2 = (26)_{10}$

3) Octal number system:

- ✓ The base or radix of the octal number system is 8,because it uses 8 symbols or digits (i.e.) 0,1,2,3,4,5,6,7
- ✓ This system is also used in computer industry.
- ✓ This system was issued to provide a shorthand way to deal with long strings of 0's & 1's creates in binary.

Example: the decimal equivalent of the octal number 2056 is

POSITION	3	2	1	0
VALUE(WEIGHTS)	8^3	8^2	8^1	8^0
OCTAL POINT	2	0	5	6

$(2056)_8 = (2X8^3) + (0X8^2) + (5X8^1) + (6X8^0)$

$=2X512+0X64+5X8+6X1$

$=1024+0+40+6 =1070$

$(2056)_8 = (1070)_{10}$

4) Hexadecimal Number System:

- ✓ The base or radix of the octal number system is 16,because it uses 16 symbols or digits
- ✓ The first ten symbols are represented by digits 0 to 9 and the remaining six by the letters 'A' through 'F'.

  A=10, B=11, C=12, D=13, E=14, F=15.

- ✓ This number system is also used to provide a shorthand way to deal with long strings of 0's and 1's created in binary.

Example: Decimal equivalent of Hexadecimal number AF3D is

POSITION	3	2	1	0
VALUE(WEIGHTS)	16^3	16^2	16^1	16^0
HEXA-DECIMAL POINT	A	F	3	D

$(AF3D)_{16}=(AX16^{3)}+(FX16^2)+(3X16^1)+(DX16^0)$

$=(10X4096)+(15X256)+(3X16)+(13X1)$

$=40960+3840+48+13$

$=44861$

$(AF3D)_{16}=(44861)_{10}$

Comparison of the Binary, Octal, Decimal and Hexadecimal:

Attribute	Binary	Octal	Decimal	Hexadecimal
Base	2	8	10	16
Lower Digit	0	0	0	0
Highest Digit	1	7	9	F

IMPORTANT TABLE EQUIVALENT VALUES OF DIFFERENT SYSTEM:

Decimal System	Binary System	Octal System	Hexadecimal System
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	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
16	10000	20	10

3.2 Convert Decimal Number into Binary Number:

- The integer and fractional parts of the decimal number are dealt with separately.
- Divide the integer part of the given decimal number repeated by 2 and writing down the remainder after each division.
- The remainder are then taken in reverse order (bottom to top) to form the binary equivalent of the integer part of the decimal number.
- The Fractional part of the given decimal number is multiplied repeatedly by 2 each time recording carry.
- The integers (carry) taken from (top to bottom) to form the binary equivalent of fractional part to the decimal number.

EXAMPLES on Conversion of Decimal Number into Binary Number:

Example 1:

Convert decimal number125 into binary number

Answer:

It is integer part then Divide the given decimal number repeated by 2 and writing down the remainder after each division.

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

It is an integer part the remainder is taken from bottom to top is 1111101 Thus $(125)_{10} = (1111101)_2$

Example 2:

Convert decimal number 0.4375 into binary number

Answer:

It is a Fractional part then multiplies the given decimal number repeatedly by 2 each time recording carry.

Multiplication	product	carry
0.4375 x 2	= 0.8750	------- 0
0.8750 x 2	= 1.750	------- 1
0.750 x 2	= 1.50	------- 1
0.50 x 2	= 1.00	------- 1

It is an fractional part the carry is taken from top to bottom is 0111

Thus $(0.4375)_{10} = (0.0111)_2$

EXAMPLE 3: Convert the decimal number 43.375 to equivalent binary number

Answer: It consists of integer part (i.e.) 43 & fractional part (i.e.) 0.375

For integer part (i.e.) 43

It is integer part then Divide the given decimal number repeated by 2 and writing down the remainder after each Division.

Division		Remainder
2	43	
2	21	1
2	10	1
2	5	0
2	2	1
2	1	0
	0	1

It is an integer part the remainder is taken from bottom to top is 101011

$(43)_{10} = (101011)_2$

For fractional part (i.e.) 0.375

It is a Fractional part then multiplies the given decimal number repeatedly by 2 each time recording carry.

Multiplication	product	carry
0.375 x 2	= 0.750	------- 0
0.750 x 2	= 1.50	------- 1
0.50 x 2	= 1.00	------- 1

It is an fractional part the carry is taken from top to bottom is 011

Thus (0.375)10 = (0.011)2

The final answer is $(43.375)_{10} = (101011.011)_2$

Example: 4 convert the decimal number 282.0 to equivalent binary number.
Answer: It consists of integer part (i.e.) 282 & fractional part (i.e.) 0.0

For integer part (i.e.) 282

It is integer part then Divide the given decimal number repeated by 2 and writing down the remainder after each division.

Division		Remainder
2	282	
2	141	→ 0
2	70	→ 1
2	35	→ 0
2	17	→ 1
2	8	→ 1
2	4	→ 0
2	2	→ 0
2	1	→ 0
	0	→ 1

It is an integer part the remainder is taken from bottom to top is 100011010

$(282)_{10} = (100011010)_2$

for fractional part (i.e.) 0.0

It is a Fractional part then multiplies the given decimal number repeatedly by 2 each time recording carry.

Multiplication	product	carry
0.0 x 2	= 0.00	------- 0

It is an fractional part the carry is taken from top to bottom is 0

$(0.0)_{10} = (0)_2$

Final answer is: $(282.0)_{10} = (100011010.0)_2$

EXAMPLE 5:

Convert the decimal number 1024 to equivalent binary number.

Answer:

It is integer part then Divide the given decimal number repeated by 2 and writing down the remainder after each Division

Division		Remainder
2	1024	
2	512	→ 0
2	256	→ 0
2	128	→ 0
2	64	→ 0
2	32	→ 0
2	16	→ 0
2	8	→ 0
2	4	→ 0
2	2	→ 0
2	1	→ 0
	0	→ 1

It is an integer part the remainder is taken from bottom to top is

10000000000

$(1024)_{10} = (10000000000)_2$

EXAMPLE:6

Convert the decimal number 0.65 to equivalent binary number.

Answer:

It is a Fractional part then multiplies the given decimal number repeatedly by 2 each time recording carry.

Multiplication	product	carry
0.65 x 2	= 1.30 -------	1
0.30 x 2	= 0.60 -------	0
0.60 x 2	=1.20 -------	1
0.20 x 2	= 0.40 -------	0
0.40 x 2	= 0.80 -------	0

The fraction may never terminated so stopped the multiplication

It is a fractional part the carry is taken from top to bottom is 10100…

Thus $(0.65)_{10} = (10100…..)_2$

3.3 Convert Binary number into Decimal number:

- For Integer:

  We can convert a binary number to its decimal equivalent by multiplying each binary digit with its position weight and sum of all bits gives the decimal number.

- For Fraction: It can be done in two steps.

  Step-1: Multiplying the digits which lie after the decimal point by 2^{-1}, 2^{-2}, 2^{-3} …
  Step-2: Simplifying the resulting fractions and adding them to get decimal number.

EXAMPLE 1:

Convert the binary number 100101 to equivalent decimal number.

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	5	4	3	2	1	0
VALUE(WEIGHTS)	2^5	2^4	2^3	2^2	2^1	2^0
BINARY POINT	1	0	0	1	0	1

$(100101)_2 = (1x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (1x2^0)$

$=32+0+0+4+0+1$

$=37$

$(100101)_2 = (37)_{10}$

EXAMPLE 2:

Convert the binary number 0.111 to equivalent decimal number

Answer: It is a fractional part it can be done in two steps.

Step-1: Multiplying the digits which lie after the decimal point by 2^{-1}, 2^{-2}, 2^{-3}…..

Step-2: Simplifying the resulting fractions and adding them to get decimal number.

$(0.111)_2 = (1x2^{-1}) + (1x2^{-2}) + (1x2^{-3})$

$= ½ + ¼ + 1/8$

$= \frac{4 + 2 +1}{8}$

$=7/8$

$=0.875$

$(0.111)_2 = (0.875)_{10}$

EXAMPLE 3:

Convert the binary number 101101.101 to decimal number.

Answer: It consists of integer part (i.e.) 1101 & fractional part (i.e.)0.101

For integer part (i.e.) 101101:

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	5	4	3	2	1	0
VALUE(WEIGHTS)	2^5	2^4	2^3	2^2	2^1	2^0
BINARY POINT	1	0	1	1	0	1

$(101101)_2 = (1X2^5)+(0X2^4)+(1X2^3)+(1X2^2)+(0X2^1)+(1X2^0)$

$=(1X32)+(0X16)+(1X8)+(1X4)+(0X2)+(1X1)$

$=32 + 0 + 8 + 4 + 0 + 1$

$=45$

$(101101)_2 = (45)_{10}$

For fractional part (i.e.) 0.101:

It is a fractional part it can be done in two steps.

Step-1: Multiplying the digits which lie after the decimal point by 2^{-1}, 2^{-2}, 2^{-3}…..

Step-2: Simplifying the resulting fractions and adding them to get decimal number

$(0.101)_2 = (1x2^{-1}) + (0x2^{-2}) + (1x2^{-3})$

$= ½ + 0 + 1/8$

$= \frac{4 + 0 + 1}{8}$

$= 5/8$

$= 0.625$

$(0.111)_2 = (0.625)_{10}$

The final answer is (101101.101)2 = (45.625)10

EXAMPLE 4: Convert the binary number 100101.0 to decimal number

Answer: It consists of integer part (i.e.) 100101 & fractional part (i.e.)0.0

- For integer part (i.e.)100101:

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	5	4	3	2	1	0
VALUE(WEIGHTS)	2^5	2^4	2^3	2^2	2^1	2^0
BINARY POINT	1	0	0	1	0	1

$(100101)_2 = (1x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (1x2^0)$

$= (1x32) + (0x16) + (0x8) + (1x4) + (0x2) + (1x1)$

$= 32 + 0 + 0 + 4 + 0 + 1$

$= 37$

$(100101)_2 = (37)_{10}$

- For fractional part (i.e.) 0.0

It is a fractional part it can be done in two steps.

Step-1: Multiplying the digits which lie after the decimal point by 2^{-1}, 2^{-2}, 2^{-3}…..

Step-2: Simplifying the resulting fractions and adding them to get decimal number

$(0.0)_2 = (0x2^{-1})$

$=0$

$(0.0)_2 = (0)_{10}$

The final answer is $(100101.0)_2 = (37.0)_{10}$

3.4 Convert binary number into hexadecimal number:

METHOD -1 for converting a binary number to its hexadecimal form first we convert binary number to decimal number and then convert decimal number to hexadecimal number.

METHOD-2

For INTEGER PART:

STEP-1: Divide the binary digits into groups of four (from RIGHT TO LEFT)

STEP-2: Convert each group of 4 binary digits to its hexadecimal equivalent

If any group does not have 4 bits it can be extended by adding 0's.

For integer part 0's can be added on left side of group

FOR FRACTIONAL PART:

STEP-1: Divide the binary digits into groups of four (from LEFT TO RIGHT)

STEP-2: Convert each group of 4 binary digits to its hexadecimal equivalent

If any group does not have 4 bits it can be extended by adding 0's

For fractional part 0's can be added on right side of group

EXAMPLE: Convert binary number 101110.100010 to hexadecimal number.

ANSWER:

METHOD-1

It consists of integer part (i.e.) 101110 & fractional part (i.e.) 0.100010

- **For integer part (i.e.) 101110:**

Binary to decimal: $(101110)_2 = (?)_{10}$

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	5	4	3	2	1	0
VALUE(WEIGHTS)	2^5	2^4	2^3	2^2	2^1	2^0
DIGITS	1	0	1	1	1	0

$(101101)_2 = (1x2^5) + (0x2^4) + (1x2^3) + (1x2^2) + (1x2^1) + (0x2^0)$

$= (1x32) + (0x16) + (1x8) + (1x4) + (1x2) + (0x1)$

$= 32 + 0 + 8 + 4 + 2 + 0$

$= 46$

$(101110)_2 = (46)_{10}$

Finally Decimal to hexadecimal: $(46)_{10} = (?)_{16}$

It is integer part then Divide the given decimal number repeated by 16 and writing down the remainder after each division.

It is an integer part the remainder is taken from bottom to top is 214

$(46)_{10} = (214)_{16}$

$(46)_{10} = (2E)_{16}$ (E=14)

Final answer for integer part $(101110)_2 = (46)_{10} = (2E)_{16}$

- **For fractional part (i.e.) 0.100010**

BINARY TO DECIMAL: $(0.100010)_2 = (?)_{10}$

It is a fractional part it can be done in two steps.

Step-1: Multiplying the digits which lie after the decimal point by $2^{-1}, 2^{-2}, 2^{-3}$…..

Step-2: Simplifying the resulting fractions and adding them to get decimal number

$(0.100010)_2 = (1x2^{-1}) + (0x2^{-2}) + (0x2^{-3}) + (0x2^{-4}) + (1x2^{-5}) + (0x2^{-6})$

$= ½ + 0 + 0 + 0 + 1/32 + 0$

$= \frac{16+1}{32}$

$= 17/32$

$= 0.53125$

$(0.100010)_2 = (0.53125)_{10}$

FINALLY DECIMAL TO HEXADECIMAL: $(0.53125)_{10} = (?)_{16}$

It is a Fractional part then multiplies the given decimal number repeatedly by 16 each time recording carry.

Multiplication		product		carry
0.53125 x 16	=	8.5	-------	8
0.5 x 16	=	8.0	-------	8
0.0 x 16	=	0.00	-------	0

It is a fractional part the carry is taken from top to bottom is 880

Final answer for fractional part $(0.53125)_{10} = (0.880)_{16}$

$(101110.100010)_2 = (46.53125)_{10} = (2E.880)_{16}$

METHOD-2

FOR INTEGER PART:

STEP-1: Divide the binary digits into groups of four (from RIGHT TO LEFT)

STEP-2: Convert each group of 4 binary digits to its hexadecimal equivalent

If any group does not have 4 bits it can be extended by adding 0's

For integer part 0's can be added on left side of group

FOR FRACTIONAL PART:

STEP-1: Divide the binary digits into groups of four (from LEFT TO RIGHT)

STEP-2: Convert each group of 4 binary digits to its hexadecimal equivalent

If any group does not have 4 bits it can be extended by adding 0's

For fractional part 0's can be added on right side of group

101110.100010 = (0010) (1110). (1000) (1000)
= (2) (E) . (8) (8)

$(101110.100010)_2 = (2E.88)_{16}$

3.5 CONVERT HEXADECIMAL TO BINARY NUMBER

Method-1: Given hexadecimal first convert into decimal and then convert it into the Binary number.

Method-2: To convert a hexadecimal to its binary form each digit of the given should be converted In to the 4 bit binary.

EXAMPLE: convert the hexadecimal number FB2.1CE8 to binary number.

Answer: method-1

First convert the given hexadecimal into decimal and then convert it into the Binary number.

It consists of integer part (i.e.) FB2 & fractional part (i.e.) 0.1CE8

- **For integer part (i.e.) FB2**

FIRST HEXADECIMAL to decimal: $(FB2)_{16} = (?)_{10}$

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	2	1	0
VALUE(WEIGHTS)	16^2	16^1	16^0
DIGITS	F	B	2

$(FB2)_{16} = (Fx16^2) + (Bx16^1) + (2x16^0)$

$= (Fx256) + (Bx16) + (2x1)$

$= (15x256) + (11x16) + (2x1)$ (F=15, B=11)

$= 3840 + 176 + 2$

$= 4018$

$(FB2)_{16} = (4018)_{10}$

- **Finally decimal to binary**

It is integer part then Divide the given decimal number repeated by 2 and writing down the remainder after each Division

It is an integer part the remainder is taken from bottom to top is 111110110010

$(4018)_{10} = (111110110010)_2$

FINAL ANSWER FOR INTEGER PART: $(FB2)_{16} = (4018)_{10} = (111110110010)_2$

- **FOR FRACTIONAL PART (i.e.) 0.1CE8:**

HEXA DECIMAL TO DECIMAL NUMBER

For fractional part it can be done in two steps.

Step-1: Multiplying the digits which lie after the decimal point by 16^{-1}, 16^{-2}, 16^{-3}…..

Step-2: Simplifying the resulting fractions and adding them to get decimal number.

$(0.1CE8)_2 = (1x16^{-1}) + (Cx16^{-2}) + (Ex16^{-3}) + (8x16^{-4})$

$= 1/16 + C/256 + E/4096 + 8/65536$

$= 1/16 + 12/256 + 14/4096 + 8/65536$ (C=12, E=14)

$= \frac{4096 + 3072 + 224 + 8}{65536}$

$= 7400/65536$

$= 0.1129$

$(0.1CE8)_2 = (0.1129)_{10}$

FINALLY DECIMAL TO BINARY:

It is a Fractional part then multiplies the given decimal number repeatedly by 2 each time recording carry.

Multiplication	product	carry
0.1129 x 2	= 0.2258	------- 0
0.2258 x 2	= 0.4516	------- 0
0.4516 x 2	= 0.9032	------- 0

0.9032 x 2 = 1.8064 ------- 1
0.8064 x 2 = 1.6128 ------- 1

- The fraction may never terminated so stopped the multiplication

It is a fractional part the carry is taken from top to bottom is 00011…

Thus $(0.1129)_{10} = (0.00011\ldots\ldots)_2$

FOR fractional part: $(0.1CE8)_{16} = (0.1129)_{10} = (0.00011\ldots\ldots)_2$

FINAL ANSWER: $(FB2.1CE8)_{16} = (4018.1129)_{10} = (111110110010.00011\ldots\ldots)_2$

Answer: method-2

To convert a hexadecimal to its binary form each digit of the given hexadecimal should be converted to its 4 bit binary equivalent & combine into a single binary number

FOR INTEGER PART(i.e.) FB2	**FOR FRACTIONAL PART(i.e.) 0.1CE8**
$(F)_{16} = (15)_{10} = (1111)_2$ $(B)_{16} = (11)_{10} = (1011)_2$ $(2)_{16} = (2)_{10} = (0010)_2$ All the source taken from table $(FB2)_{16} = (111110110010)_2$	$(1)_{16} = (1)_{10} = (0001)_2$ $(C)_{16} = (12)_{10} = (1010)_2$ $(E)_{16} = (14)_{10} = (1110)_2$ $(8)_{16} = (8)_{10} = (1000)_2$ All the source taken from table $(0.1CE8)_{16} = (0.0001101011101000)_2$
FINAL ANSWER: $(FB2.1CE8)_{16} = (111110110010.0001101011101000)_2$	

Coding schemes Introduction

A Computer accepts and processes data in binary form. The process of converting data into binary form is called encoding of data. So we use some codes to represent characters in terms of 0'sand 1's.

The most widely used codes are as follows:

1. Binary coded Decimal (BCD).
2. American Standard code for Information Interchange (ASCII).
3. Extended Binary Coded Decimal Interchange Code (EBCDIC).

1. Binary Coded Decimal (BCD): The BCD is the simplest binary code to represent a decimal number.

In BCD code each decimal digit is represented by four binary bits.

Decimal Digit	BCD Equivalent 8421
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

3.6 Explain the ASCII coding Scheme

The ASCII (**American Standard Code for Information Interchange**) is pronounced as "ask-ee".

- It was developed by American National Standards Institute **(ANSI).**
- It is very widely used code in small computers. The first ASCII code was 7-bit code but was later modified to an 8-bit code. It is a standard code to represent **alphanumeric data.**
  - **ASCII-7** is a 7-bit standard allows 2^7 combinations.128 unique symbols are represented using ASCII-7. ASCII-7 has been modified by IBM to ASCII-8.
  - **ASCII-8** is an extended version of ASCII-7. ASCII-8 is an 8-bit standard allows 2^8 combinations. 256 unique symbols are represented using ASCII-8

ASCII Printing Characters Chart

Decimal	Hex	Oct	Character	Decimal	Hex	Oct	Character	Decimal	Hex	Oct	Character
32	20	040	space	64	40	100	@	96	60	140	`
33	21	041	!	65	41	101	A	97	61	141	a
34	22	042	"	66	42	102	B	98	62	142	b
35	23	043	#	67	43	103	C	99	63	143	c
36	24	044	$	68	44	104	D	100	64	144	d
37	25	045	%	69	45	105	E	101	65	145	e
38	26	046	&	70	46	106	F	102	66	146	f
39	27	047	'	71	47	107	G	103	67	147	g
40	28	050	(	72	48	110	H	104	68	150	h
41	29	051	)	73	49	111	I	105	69	151	i
42	2A	052	*	74	4A	112	J	106	6A	152	j
43	2B	053	+	75	4B	113	K	107	6B	153	k
44	2C	054	,	76	4C	114	L	108	6C	154	l
45	2D	055	-	77	4D	115	M	109	6D	155	m
46	2E	056	.	78	4E	116	N	110	6E	156	n
47	2F	057	/	79	4F	117	O	111	6F	157	o
48	30	060	0	80	50	120	P	112	70	160	p
49	31	061	1	81	51	121	Q	113	71	161	q
50	32	062	2	82	52	122	R	114	72	162	r
51	33	063	3	83	53	123	S	115	73	163	s
52	34	064	4	84	54	124	T	116	74	164	t
53	35	065	5	85	55	125	U	117	75	165	u
54	36	066	6	86	56	126	V	118	76	166	v
55	37	067	7	87	57	127	W	119	77	167	w
56	38	070	8	88	58	130	X	120	78	170	x
57	39	071	9	89	59	131	Y	121	79	171	y
58	3A	072	:	90	5A	132	Z	122	7A	172	z
59	3B	073	;	91	5B	133	[	123	7B	173	{
60	3C	074	<	92	5C	134	\	124	7C	174	|
61	3D	075	=	93	5D	135	]	125	7D	175	}
62	3E	076	>	94	5E	136	^	126	7E	176	~
63	3F	077	?	95	5F	137	_	127	7F	177	DEL

We can check the value of any ASCII code by just holding down the ALT key and typing the ASCII code
Example: when we hold down the ALT key and type 66 from the keyboard, then the character B appears On the screen.

3.7 Explain the EBCDIC coding Scheme

EBCDIC is pronounced as "ebb-seedick".

- It is extended form of BCD code which is 8 bit. It is used extensively in large computers. **Extended Binary Coded Decimal Interchange Code** (EBCDIC) uses 8 bits to represent a symbol in the data.
- It is an **8 bit code** developed by **IBM** in 1964. It supports 256 characters. EBCDIC uses more or less the same characters as ASCII, but different code points. It has non-sequence letter format.
- EBCDIC codes are mainly used in the mainframe computers
- EBCDIC allows 2^8 = 256 combinations of bits.
- It represents alphabets, decimal digits.

BELOW IS A ASCII TABLE

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Below is a EBCDIC TABLE

Dec	Hx	Oct	Char	
0	0	000	nul	(Null)
1	1	001	soh	(Start of Heading)
2	2	002	stx	(Start of Text)
3	3	003	etx	(End of Text)
4	4	004	pf	(Punch Off)
5	5	005	ht	(Horizontal Tab)
6	6	006	lc	(Lower Case)
7	7	007	del	(Delete)
8	8	010	ge	
9	9	011	rlf	
10	a	012	smm	(Start of Manual Message)
11	b	013	vt	(Vertical Tab)
12	c	014	ff	(Form Feed)
13	d	015	cr	(Carriage Return)
14	e	016	so	(Shift Out)
15	f	017	si	(Shift in)
16	10	020	dle	(Data Link Escape)
17	11	021	dc1	(Device Control 1)
18	12	022	dc2	dc2 (Device Control 2)
19	13	023	tm	(Tape Mark)
20	14	024	res	(Restore)
21	15	025	nl	(New Line)
22	16	026	bs	(Backspace)
23	17	027	il	(Idle)
24	18	030	can	(Cancel)
25	19	031	em	(End of Medium)
26	1a	032	cc	(Cursor Control)
27	1b	033	cu1	(Customer Use 1)
28	1c	034	ifs	(Interchange File Separator)
29	1d	035	igs	(Interchange Group Separator)
30	1e	036	irs	(Interchange Record
31	1f	037	ius	(Interchange Unit Separator)
32	20	040	ds	(Digit Select)
33	21	041	sos	(Start of Significance)
34	22	042	fs	(Field Separator)
35	23	043		
36	24	044	byp	(Bypass)
37	25	045	lf	(Line Feed)
38	26	046	etb	(End of Transmision Block)
39	27	047	esc	(Escape)
40	28	050		
41	29	051		
42	2a	052	sm	(Set Mode)
43	2b	053	cu2	(Customer Use 2)
44	2c	054		
45	2d	055	enq	(Enquiry)
46	2e	056	ack	(Acknowledge)
47	2f	057	bel	(Bell)
48	30	060		
49	31	061		
50	32	062	syn	(Synchronous Idle)
51	33	063		
52	34	064	pn	(Punch On)
53	35	065	rs	(Reader Stop)
54	36	066	uc	(Upper Case)
55	37	067	eot	(End of Transmission)
56	38	070		
57	39	071		
58	3a	072		
59	3b	073	cu3	(Customer Use 3)
60	3c	074	dc4	(Device Control 4)
61	3d	075	nak	(Negative Acknowledge)
62	3e	076		
63	3f	077	sub	(Substitute)
64	40	100	Sp	(Space)

Dec	Hx	Oct	Char
65	41	101	
66	42	102	
67	43	103	
68	44	104	
69	45	105	
70	46	106	
71	47	107	
72	48	110	
73	49	111	
74	4a	112	¢
75	4b	113	.
76	4c	114	>
77	4d	115	(
78	4e	116	+
79	4f	117	|
80	50	120	&
81	51	121	
82	52	122	
83	53	123	
84	54	124	
85	55	125	
86	56	126	
87	57	127	
88	58	130	
89	59	131	
90	5a	132	!
91	5b	133	$
92	5c	134	*
93	5d	135	)
94	5e	136	;
95	5f	137	
96	60	140	-
97	61	141	/
98	62	142	
99	63	143	
100	64	144	
101	65	145	
102	66	146	
103	67	147	
104	68	150	
105	69	151	
106	6a	152	|
107	6b	153	,
108	6c	154	%
109	6d	155	
110	6e	156	<
111	6f	157	?
112	70	160	
113	71	161	
114	72	162	
115	73	163	
116	74	164	
117	75	165	
118	76	166	
119	77	167	
120	78	170	
121	79	171	`
122	7a	172	:
123	7b	173	#
124	7c	174	@
125	7d	175	'
126	7e	176	=
127	7f	177	"
128	80	200	
129	81	201	a

Dec	Hx	Oct	Char
130	82	202	b
131	83	203	c
132	84	204	d
133	85	205	e
134	86	206	f
135	87	207	g
136	88	210	h
137	89	211	i
138	8a	212	
139	8b	213	
140	8c	214	
141	8d	215	
142	8e	216	
143	8f	217	
144	90	220	
145	91	221	j
146	92	222	k
147	93	223	l
148	94	224	m
149	95	225	n
150	96	226	o
151	97	227	p
152	98	230	q
153	99	231	r
154	9a	232	
155	9b	233	
156	9c	234	
157	9d	235	
158	9e	236	
159	9f	237	
160	a0	240	
161	a1	241	~
162	a2	242	s
163	a3	243	t
164	a4	244	u
165	a5	245	v
166	a6	246	w
167	a7	247	x
168	a8	250	y
169	a9	251	z
170	aa	252	
171	ab	253	
172	ac	254	
173	ad	255	
174	ae	256	
175	af	257	
176	b0	260	
177	b1	261	
178	b2	262	
179	b3	263	
180	b4	264	
181	b5	265	
182	b6	266	
183	b7	267	
184	b8	270	
185	b9	271	
186	ba	272	
187	bb	273	
188	bc	274	
189	bd	275	
190	be	276	
191	bf	277	
192	c0	300	{
193	c1	301	A
194	c2	302	B

Dec	Hx	Oct	Char
195	c3	303	C
196	c4	304	D
197	c5	305	E
198	c6	306	F
199	c7	307	G
200	c8	310	H
201	c9	311	I
202	ca	312	
203	cb	313	
204	cc	314	
205	cd	315	
206	ce	316	
207	cf	317	
208	d0	320	}
209	d1	321	J
210	d2	322	K
211	d3	323	L
212	d4	324	M
213	d5	325	N
214	d6	326	O
215	d7	327	P
216	d8	330	Q
217	d9	331	R
218	da	332	
219	db	333	
220	dc	334	
221	dd	335	
222	de	336	
223	df	337	
224	e0	340	\
225	e1	341	
226	e2	342	S
227	e3	343	T
228	e4	344	U
229	e5	345	V
230	e6	346	W
231	e7	347	X
232	e8	350	Y
233	e9	351	Z
234	ea	352	
235	eb	353	
236	ec	354	
237	ed	355	
238	ee	356	
239	eF	357	
240	f0	360	0
241	f1	361	1
242	f2	362	2
243	f3	363	3
244	f4	364	4
245	f5	365	5
246	f6	366	6
247	f7	367	7
248	f8	370	8
249	f9	371	9
250	fa	372	|
251	fb	373	
252	fc	374	
253	fd	375	
254	fe	376	
255	ff	377	eo

Source: www.pubblinet.com

Other Systems

CONVERT OCTAL NUMBER INTO BINARY NUMBER:

Method-1: First we convert octal number into decimal and then decimal into binary.

Method-2: convert each digit of the given octal number into its 3-bit binary equivalent

Example: Convert the octal number 562 to equivalent binary number

Answer:

Method-1

First convert the given octal number into decimal and then convert it into the Binary number.

FIRST octal to decimal: $(562)_8 = (?)_{10}$

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	2	1	0
VALUE(WEIGHTS)	8^2	8^1	8^0
DIGITS	5	6	2

$(2AB)_8 = (5x8^2) + (6x8^1) + (2x8^0)$
$= (5x64) + (6x8) + (2x1)$
$= (320) + (48) + (2)$
$= 370$
$(2AB)_8 = (370)_{10}$

- **Finally decimal to binary:** $(370)_{10} = (?)_2$

  It is integer part then Divide the given decimal number repeated by 2 and writing down the Remainder after each Division

Division		Remainder
2	370	
2	185	0
2	92	1
2	46	0
2	23	0
2	11	1
2	5	1
2	2	1
2	1	0
	0	1

It is an integer part the remainder is taken from bottom to top is 101110010

$(370)_{10} = (101110010)_2$

FINAL ANSWER: $(562)_8 = (370)_{10} = (101110010)_2$

METHOD-2

Convert each digit of the given octal number into its 3-bit binary equivalent

$(5)_8 = (5)_{10} = (101)_2$
$(6)_8 = (6)_{10} = (110)_2$
$(2)_8 = (2)_{10} = (010)_2$

$(562)_8 = (101110010)_2$

CONVERT BINARY NUMBER INTO OCTAL NUMBER:

Method-1: To convert into octal number we need to
Convert binary into decimal and then convert decimal to octal number.

Method-2: FOR INTEGER PART:

STEP-1: Divide the binary digits into groups (from RIGHT TO LEFT)
STEP-2: Convert each group of 3 binary digits to its octal equivalent
If any group does not have 3 bits it can be extended by adding 0's
For integer part 0's can be added on left side of group

FOR FRACTIONAL PART:

STEP-1: Divide the binary digits into groups (from LEFT TO RIGHT)
STEP-2: Convert each group of 3 binary digits to its octal equivalent
If any group does not have 3 bits it can be extended by adding 0's
For fractional part 0's can be added on right side of group

EXAMPLE 1: Convert the binary number 1001110.0 to octal number

Answer:

Method-1

For converting a binary number to its octal form first we convert binary number to decimal number and then Convert decimal number to octal number.
It consists of integer part (i.e.) 1001110 & fractional part (i.e.) 0.0

- **For integer part (i.e.) 1001110**

Binary to decimal: $(1001110)_2 = (?)_{10}$

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	6	5	4	3	2	1	0
VALUE(WEIGHTS)	2^6	2^5	2^4	2^3	2^2	2^1	2^0
DIGITS	1	0	0	1	1	1	0

$(1001110)_2 = (1x2^6) + (0x2^5) + (0x2^4) + (1x2^3) + (1x2^2) + (1x2^1) + (0x2^0)$
$= (1x64) + (0x32) + (0x16) + (1x8) + (1x4) + (1x2) + (0x1)$
$= 64 + 0 + 0 + 8 + 4 + 2 + 0$
$= 78$
$(1001110)_2 = (78)_{10}$

Finally Decimal to octal number $(78)_{10} = (?)_8$

It is integer part then Divide the given decimal number repeated by 8 and writing down the remainder after each division.

Division		Remainder
8	78	
8	9	→ 6
8	1	→ 1
	0	→ 1

It is an integer part the remainder is taken from bottom to top is 116
$(78)_{10} = (116)_8$
Final answer for integer part $(1001110)_2 = (78)_{10} = (116)_8$

- **For fractional part (i.e.) 0.0**

Binary to decimal: $(0.0)_2 = (?)_{10}$

It is a fractional part it can be done in two steps.

Step-1: Multiplying the digits which lie after the decimal point by $2^{-1}, 2^{-2}, 2^{-3}$.....
Step-2: Simplifying the resulting fractions and adding them to get decimal number.

$(0.0)_2 = (0x2^{-1})$
$= 0$
$(0.0)_2 = (0.0)_{10}$

FINALLY DECIMAL TO OCTAL:

It is a Fractional part then multiplies the given decimal number repeatedly by 8 each time recording carry.

Multiplication		product		carry
0.0 x 8	=	0.0	-------	0

It is a fractional part the carry is taken from top to bottom is 0

Thus $(0.0)_{10} = (0.0)_8$

Final answer for fractional part $(0.0)_2 = (0.0)_{10} = (0.0)_8$

THE FINAL ANSWER IS $(1001110.0)_2 = (78.0)_{10} = (116.0)_8$

METHOD-2

(1001110.0) = (1) (001) (110). (0)
= (001) (001) (110). (000)
= (1) (1) (6) . (0)

$(1001110.0)_2 = (116.0)_8$

CONVERT HEXADECIMAL INTO OCTAL NUMBER:

EXAMPLE 1: What is the result of the conversion $56.34_{(16)}=?_{(8)}$

ANSWER: first we have to convert the hexadecimal into decimal number & then decimal number into octal

It consists of integer part (i.e.) 56 & fractional part (i.e.) 0.34

- **For integer part (i.e.) 56**

FIRST HEXADECIMAL to decimal: $(56)_{16}= (?)_{10}$

It is an integer part so we can multiply each binary digit with its position weight and sum of all bits gives the decimal number.

POSITION	1	0
VALUE(WEIGHTS)	16^1	16^0
DIGITS	5	6

$(56)_{16} = (5 \times 16^1) + (6 \times 16^0)$
$= (5 \times 16) + (6 \times 1)$
$= 80 + 6$
$= 86$

$(56)_{16} = (86)_{10}$

Finally DECIMAL to OCTAL: $(86)_{10} = (?)_8$

It is integer part then Divide the given decimal number repeated by 8 and writing down the remainder after each Division

Division		Remainder
8	86	
8	10	→ 6
8	1	→ 2
	0	→ 1

It is an integer part the remainder is taken from bottom to top is 126

$(86)_{10} = (126)_8$

FINAL ANSWER FOR INTEGER PART: $(56)_{16} = (86)_{10} = (126)_8$

- **FOR FRACTIONAL PART (i.e.) 0.34:**

FIRST HEXA DECIMAL TO DECIMAL NUMBER: $(0.34)_{16} = (?)_{10}$

For fractional part it can be done in two steps.

Step-1: Multiplying the digits which lie after the decimal point by 16^{-1}, 16^{-2}, 16^{-3}…..

Step-2: Simplifying the resulting fractions and adding them to get decimal number.

$(0.34)_{16} = (3 \times 16^{-1}) + (4 \times 16^{-2})$

= 3/16 + 4/256

= 3X16 + 4

256

= 48 + 4

52

= 52/256

= 0.203125

$(0.34)_{16} = (0.203125)_{10}$

FINALLY DECIMAL TO OCTAL: $(0.203125)_{10} = (?)_8$

It is a Fractional part then multiplies the given decimal number repeatedly by 8 each time recording carry.

Multiplication	product	carry
0.203125 x 8	= 1.625	------- 1
0.625 x 8	= 5	------- 5

It is an fractional part the carry is taken from top to bottom is 15

Thus $(0.203125)_{10} = (0.15)_8$

FINAL ANSWER FOR fractional part: $(0.34)_{16} = (0.203125)_{10} = (0.15)_8$

FINAL ANSWER: $(56.34)_{16} = (86.203125)_{10} = (126.15)_8$

Concept of a byte and word

Bit: It is the smallest unit of information used in a computer system. It can have either the value 0 or1
Nibble: It is a combination of 4 bits. Example: 0101
Byte: Byte is a group of 8 bits, Derived from words “by eight”. Example: 11001100, 00011100
Word: combination of 16 bits (or) Group of 2 bytes. Example: 1111111100000000
Double Word: 32 bits.
KB (KILO BYTE): Used to represent 1024 bytes of information.
MB (MEGA BYTE): Used to represent 1024 KILO bytes of information.
Likewise GB, TB, PB, EB, ZB, YB