UNIT-3

DATA REPESENTATION

- Number system: -

  → Positional number.

  ↓

  Decimal, binary, octal, hexadecimal.

  → Non positional number.

- Number: -number is a symbol used for making, majoring and counting.

- Number system are basically of two types:-

  1. non positional number
  2. positional number
     1) Non positional number: - The no which has not any position and basis is called non positional number.

        Ex: -

        I. For 1
        II. For 2
        III. For 3

     2) Positional number: - the no which as position or basis is called positional number.

Ex: - 2585

$=2*10^3+5*10^2+8*10^1+5*10^0$

=2*1000+5*100+8*10+5*1
=2000+500+80+5
=2585

- Positional number system is divided in to four types: -
  i. Decimal number system (10)
  ii. Binary number system (2)
  iii. Octal number system (8)
  iv. Hexa decimal system (16)

i. Decimal number system: -

- The number system which we used in our day to day life is called decimal number system.
- The number which has base 10 is known as decimal number.
- Such as: 0,1,2,3,4,5,6,7,8,9.

Ex: - 234,536

ii. Binary number system: -

- This is exactly decimal number system accept that bace is 2 insteed of (1,0).
- It has only two symbols or digit 0, 1.

Ex: -

- $(1101)_2=(13)_{10}$

$=1*2^3+1*2^2 0*2^1+1*2^0$
$=8+4+0+1$
$=13$

- Bit: - binary digit is obtained refer to by the common abbreviation.

Hence bits in computer technology mean either 0 or 1.

- A binary system of n bit is called a n bit number.
  - List of 3 bit binary:

Binary Decimal

0000	0
0001	1
0010	2

0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

❖ Note: - Decimal number in the fix number of 0 to 2^n-can be represented form as n bit number.

- Byte: - Byte is the fixed numbers of absent bit that represent a particular system normally a byte consist of eight bit.

1 byte = 8 bit

1 kilobyte = 2^{10}byte(1024 bit)

Megabyte = 2^{20}byte(1024*1024)

Gigabyte =2^{30}byte(1024*1024*1024)

iii. Octal number system: -

In the octal number system basic are 8. hence these are only eight symbol or digit : - 0,1,2,3,4,5,6,7.

Ex: - $(2057)_8= (14)_8$

iv. Hexa decimal system: -

In the hexa decimal number is basis is 16. hence are 16 symbol digit: 0,1,2,3,4,5,6,7,8,9,10(A),11(B),12(C),13(D),14(E),15(F).

Ex: -32D, C23, A23 etc.

Q. Conversion of decimal to binary number

$(41)_{10} = (101001)_2$

a	r
41/2=20	1
20/2=10	0
10/2=5	0
5/2=2	1
2/2=1	0
1/2=0	1

Q. conversion of binary to decimal

$(10011)_2 = (?)_{10}$

$1*2^4+0*2^3=0*2^2+1*2^1+1*2^0$

$=16+0+0+2+1$

$=19$

` Q. Conversion of decimal to octal

$(98)_{10}= (?)_8$

98/8=12   2

12/8=1 4

1/8=0 1

$(142)_8$ Ans…

Q. conversion of octal to decimal

$(142)_8 = (?)_{10}$

$1*8^2+4*8^1+2*8^0$

64+32+2=98ans…

Q. conversion of decimal to hexadecimal

$(8256)_{10} = (?)_{16}$

8256/16=516 0

516/16=32 4

32/16=2 0

2/16=0 2

$(2040)_{16}$ ans…

Q. conversion of hexadecimal to decimal

$(2040)_{16} = (?)_{10}$

$2*16^3+0*16^2+4*16^1+0*16^0$

8192+64

$= (8256)_{10}$

Q. conversion of binary to octal

$(110111)_2 = (?)_8$

binary to decimal

32	16	8	4	2	1
1	1	0	1	1	1

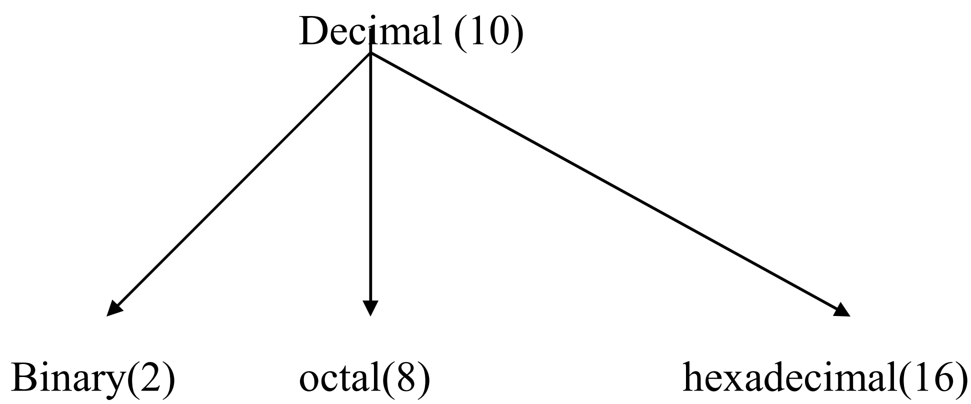
=55

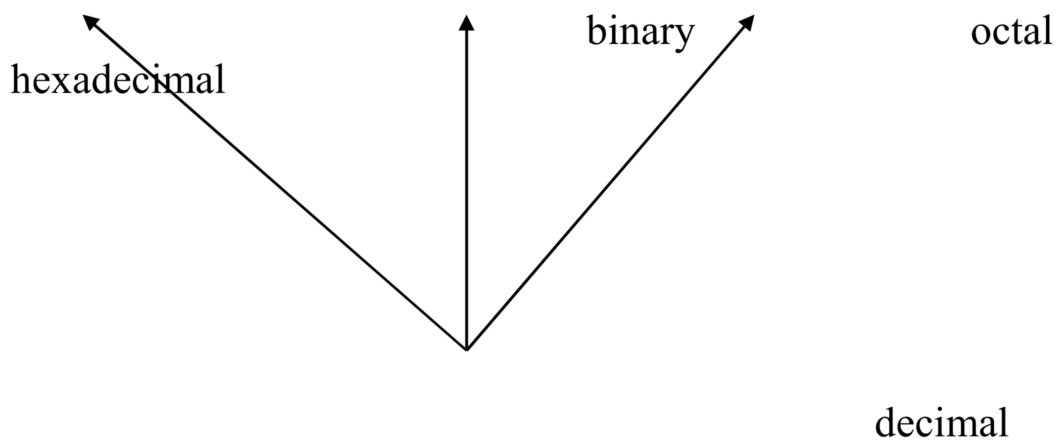
55/8=6 7

6/8= 0 6

$= (6\ 7)_8$

❖ Note: -

- 



- RULE:- the base of octal number is 8 but 8 is equal (=) to 2^3 there for to convert a binary number to octal number make three byte ,binary group of the binary number.

Ex: -

Right to left

Left to right

Q. $(101101101101.110101101101)_2 = (?)_8$

101 101 101 101. 110 101 101 101

5 5 5 5 . 6 5 5 5

$(5555.6555)_8$

Q.<u>conversion of octal number fraction to binary fraction</u>.

$(367422.110523)_{\mathbf{8}} = (?)_2$

3 6 7 4 2 1 1. 1 0 5 2 3
011 110 111 100 010 001 001. 001 000 101 010 011

=01111011110001000 1001.001000101010011

- Rule: -the base of the fractional hexadecimal number 16 but $16=2^4$ three for to convert a binary number to hexadecimal to make four byte binary group of the binary number.

Q. conversion of hexadecimal fraction to binary fraction.
$(D3FA48.18CD86)_{16}$
$(1101\ 0011\ 1111\ 1010\ 0100\ 1000.0001\ 1011\ 1100\ 011\ 000\ 0110)_2$

- Binary arithmetic

I. Addition
II. Subtraction
III. Multiplication
IV. Division

I. Addition

Binary addition
0+0=0
1+0=0
0+1=0
1+1=0 pulse a carry to next high column
101+100=1001, 1101+1101=11010 1011+1111=11010,
1111+1111=11110,
110110101+111111111=110110100, 1001+0101=1110,
1001+0101=1110.

II. Subtraction

Binary subtraction

0-0=0

1-0=1

1-1=0

0-1=1 with borrow from next column.

110-101=001, 1001-111=010, 1000-111=001, 101010-11111=01011, 1110-0101=1001, 1010-0101=0101.

III. Multiplication

Binary multiplication

0*0=0 Ex: - 1111*101=1001011

1*1=1 11111*1111=111010001

0*1=0

1*0=0

IV. Division

Binary division

0/1=0

1/1=1

Ex: - 100001/11=1011

100001/100=1000.01

➢ Complement

- It is a method.
- It is a rules or pedicure.
- Compliments are used in digital computer for simplifying the subtraction and for logical calculation.
- There are two types of complements for each base=r
  - i (r-1)'s complements
  - ii R's complements

i (r-1)'s complements:-given a number N is base, digit n. (r-1)'s complement of n is define as $(r^n-1)-N$.

ii r's complement: - given a number N is base 'r' and digit n,r's complements N defined as :

$(r^n-N)= (r^n-1)-N+1=(r-1)$'s complement + 1

- In the case of decimal number base is equal to 10.
- There are component.
  - i 9's complement
  - ii 10's complement

i 9's complement: - 9's complement defined as $(10^n-1)-N$ where_

n=digit.

N=number.

Ex: - $(2356)_{10}$

9's complement

N=2356

N=4

R=10

9's complement= $(r^n-1)-N$

$= (10^4-1)-2356$

$= (10000-1)-2356$

$=9999-2356$

$=7643$ ans…

ii 10's complement=(r^n-N)

$= (10^4-2356)$

$=10000-2356$

$=7644$ ans…

- In the case of binary number:-
  - i 1's complement
  - ii 2's complement

i. 1's complement; - 1's complement as defined as $(2^n-1)-N$.

n=digit
N=number
r=base or radix

ii. 2's complement: -2's complement as defined as(2^n-N) where
N=digit
N=number

Ex: - $(1010)_2$
N=1010
n=4
r=2

1's complement= $(r^n-1)-N$
$= (2^4-1)1010$
$= (16-1)-1010$
$=15-1010$
$=1111-1010$
$=0101$

2's complement= (r^n-N)
$= (2^4-1010)$
$= (16-1010)$
$= (16-10)$
$=6$
$=0110$

- <u>BCD</u>

- BCD stands for binary coded decimal.
- The binary coded decimal (BCD) is one of the early computer codes.
- It is based on the idea of converting each digit of a decimal number in to its binary equivalents the entire decimal value in to a pure binary form.
- It converted each digit in-4-bit initially.
- That means it converted 2^4=16 character.

- This makes conversion process easier.
- The BCD equivalent of each decimal digit each in the figure.

➢ <u>4-bit BCD</u>

Decimal no	BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000

Ex: - binary conversion (BC) –

$(42)_{10} = (101010)_2$

But conversion it in BCD

$(42)_{10} = (01000010)_{BCD}$

- 4-bit BCD code represent with adding zone position of 2-byte=2^6=64 character.
- 6-bit BCD represented 64 characters such as 26 letter alphabet, 28 other character or 10 numbers.

➢ <u>6-bit BCD</u>

Character	Zone	BCD
A I	11 11	0001 1001
J R	10 10	0001 1001

S Z	01 01	0010 1001
L 9 O	00 00 00	0001 1001 0000

- Character BCD table

CHARACTER	ZONE	BCD
A	11	0001
B	11	0010
C	11	0011
D	11	0100
E	11	0101
F	11	0110
G	11	0111
H	11	1000
I	11	1001
J	10	0001
K	10	0010
L	10	0011
M	10	0100
N	10	0101
O	10	0110
P	10	0111
Q	10	1000
R	10	1001
S	01	0010
T	01	0011
U	01	0100
V	01	0101
W	01	0110
X	01	0111

Y	01	1000
Z	01	1001

- Number BCD table

Number zone BCD

1	00	0001
2	00	0010
3	00	0011
4	00	0100
5	00	0101
6	00	0110
7	00	0111
8	00	1000
9	00	1001
10	00	0000

- EBCDIC

- EBCDIC stands for extended binary coded decimal interchange code.
- The major problem of BCD is that only 64(2^6) different character in it.
- This is not sufficient for providing decimal number (10), lower case letter (28) and upper case letter (28) and other special character (28).
- Hence BCD code extended 6-bit code to 8-bit code.
- The added 2-bit are used at additional zone bit expending the zone to four byte this is called decimal interchange code.
- In this case 2^8=256 different character can be represented.

- 8-BIT EBCDIC

Character zone EBCDIC

A I	1100 1100	0001 1001
J	1101	0001

R	1101	1001
S Z	1110 1110	0010 1001
O 9	1111 1111	0000 1001

- <u>NUMBER EBCDIC TABLE</u>

NUMBER	ZONE	EBCDIC
0	1111	0000
1	1111	0001
2	1111	0010
3	1111	0011
4	1111	0100
5	1111	0101
6	1111	0110
7	1111	0111
8	1111	1000
9	1111	1001

- <u>Character EBCDIC Table</u>

CHARACTER	ZONE	EBCDIC
A	1100	0001
B	11OO	0010
C	1100	0011
D	1100	0100
E	1100	0101
F	1100	0110
G	1100	0111
H	1100	1000
I	1100	1001
J	1101	0001
K	1101	0010

L	1101	0011
M	1101	0100
N	1101	0101
O	1101	0110
P	1101	0111
Q	1101	1000
R	1101	1001
S	1110	0010
T	1110	0011
U	1110	0100
V	1110	0101
W	1110	0110
X	1110	0111
Y	1110	1000
Z	1110	1001

- ASCII

- ASCII stands for American standard code for information interchange.
- It is widely by several computer manufacturers as there computer interchange code.
- This code is popular in data communication and it used almost exclusively to represent data internally in micro computer.
- ASCII is of two type ASCII-7 and ASCII-8.

  - ASCII-7

7-bit code which allow 128(2^7) different character. The first 3-bit are indicates zone bit and last 4-bit indicates the digit.

  - ASCII-8

8-bit code which allows 256(2^8) different character. The first 4-bit are used as zone bit and last 4-bit indicates as digits as digit. It is extend version of ASCII-7.

- Table of ASCII-7

Character	Zone	Digit
O 9	011 011	0000 1001
A O	100 100	0001 1111
P Z	101 101	0000 1010

- Table of ASCII- 8

Character	Zone	Digit
O 9	0101 0101	0000 1001
A O	0101 0101	0001 1111
P Z	1010 1010	0000 1010