UNIT-1: NUMBER SYSTEMS

1.1:Decimal,Binary, Octal And Hexa-Decimal Number System And Their Interconversion.

1.2: *Binary And Hexadecimal Addition, Subtraction And Multiplication*

1.3: *1's And 2's Compliment Methods Of Addition/ Subtraction*

OBJECTIVES

After going through this unit, you will be able to

- understand the decimal, binary, octal and hexadecimal number systems
- convert from one number system into another
- apply arithmetic operations to binary numbers

.

INTRODUCTION

The binary number system and digital codes are fundamental to computers. In this chapter, the binary number system and its relationship to other systems such as decimal, hexadecimal, and octal are introduced. Arithmetic operations with binary numbers are discussed to provide a basis for understanding how computers and many other types of digital systems work.

NUMBER SYSTEMS

A number system relates quantities and symbols. The base or radix of a number system represents the number of digits or basic symbols in that particular number system. In decimal system the base is 10, because of use the numbers 0, 1, 2,3,4,5,6,7,8 and 9.

Binary Number System

A binary number system is a code that uses only two basic symbols. The digits can be any two distinct characters, but it should be 0 or 1. The binary equivalent for some decimal numbers are given below

decimal	0	1	2	3	4	5	6	7	8	9	10	11
binary	0	1	10	11	100	101	110	111	1000	1001	1010	1011

Each digit in a binary number has a value or weight. The LSB has a value of 1. The second from the right has a value of 2, the next 4 , etc.,

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

Binary to decimal conversion:

$(1001)_2 = X_{10}$

$1001 = 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0$
$= 8+0+0+1$
$(1001)_2 = (9)_{10}$

Fractions:

For fractions the weights of the digit positions are written from right of the binary point and weights are given as follows.

2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}

E.g.:

$(0.0110)_2 = X_{10}$
$= 0x2^{-1} + 1x2^{-2} + 1x2^{-3} + 0x2^{-4}$
$= 0\ x0.5 + 1x0.25 + 1x0.125 + 0x0.0625$
$= (0.375)_{10}$

E.g.:

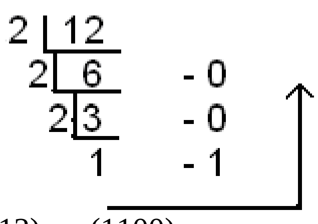
$(1011.101)_2 = X_{10}$
$= 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 + 1x2^{-1} + 0x2^{-2} + 1x2^{-3}$
$= 8 + 0 + 2 + 1 + 0.5 + 0 + 0.125$
$= (11.625)_{10}$

Decimal to binary conversion:
(Double Dabble method)

In this method the decimal number is divided by 2 progressively and the remainder is written after each division. Then the remainders are taken in the reverse order to form the binary number.

E.g.:

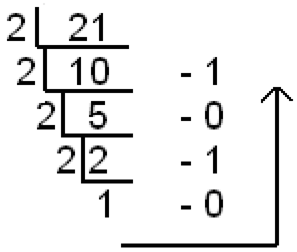
$(12)_{10} = X_2$



$(12)_{10} = (1100)_2$

E.g.:

$(21)_2 = X_2$



$(21)_2 = (10101)_2$

Fractions:

The fraction is multiplied by 2 and the carry in the integer position is written after each multiplication. Then they are written in the forward order to get the corresponding binary equivalent.

E.g.:

$(0.4375)_{10} = X_2$

2 x 0.4375 = 0.8750 => 0
2 x 0.8750 = 1.750 => 1
2 x 0. 750 = 1.5 => 1
2 x 0.5 = 1.0 => 1

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

Octal Number System

Octal number system has a base of 8 i.e., it has eight basic symbols. First eight decimal digits 0, 1,2,3,4,5,6,7 are used in this system.

Octal to decimal conversion:

In the octal number system each digit corresponds to the powers of 8. The weight of digital position in octal number is as follows

8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}	8^{-3}

To convert from octal to decimal multiply each octal digit by its weight and add the resulting products.

E.g.:

$(48)_8 = X_{10}$

$48 = 4 \times 8^1 + 7 \times 8^0$
$= 32 + 7$
$= 39$

$(48)_8 = (39)_{10}$

E.g.:

$(22.34)_8 = X_{10}$

$22.34 = 2 \times 8^1 + 2 \times 8^0 + 3 \times 8^{-1} + 4 \times 8^{-2}$
$=16 + 2 + 3 \times 1/8 + 4 \times 1/64$
$= (18.4375)$

$(22.34)_8 = (18.4375)_{10}$

Decimal to octal conversion:

Here the number is divided by 8 progressively and each time the remainder is written and finally the remainders are written in the reverse order to form the octal number. If the number has a fraction part, that part is multiplied by 8 and carry in the integer part is taken. Finally the carries are taken in the forward order.

E.g.:

$(19.11)_{10} = X_8$

$$\begin{array}{r|l} 8 & 19 \\ \hline & 2 - 3 \end{array}$$

0.11 x 8 = 0.88 => 0
0.88 x 8 = 7.04 => 7
0.04 x 8 = 0.32 => 0
0.32 x 8 = 2.56 => 2
0.56 x 8 = 4.48 => 4

$(19.11)_{10} = (23.07024)_8$

Octal to binary conversion:

Since the base of octal number is 8, i.e., the third power of 2, each octal number is converted into its equivalent binary digit of length three.

E.g.:

$(57.127)_8 = X_2$

```
 5     7  .  1     2     7
101   111 . 001   010   111
```

$(57.127)_8 = (101111001010111)_2$

Binary to octal:

The given binary number is grouped into a group of 3 bits, starting at the octal point and each group is converted into its octal equivalent.

E.g.:

$(1101101.11101)_2 = X_8$

```
001 101 101.111 010
 1   5   5 . 7   2
```

$(1101101.11101)_2 = (155.72)_8$

Hexadecimal Number System:

The hexadecimal number system has a base of 16. It has 16 symbols from 0 through 9 and A through F.

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

Binary to hexadecimal:
The binary number is grouped into bits of 4 from the binary point then the corresponding hexadecimal equivalent is written.

E.g.:

$(100101110\ .\ 11011)_2 = X_{16}$

0001 0010 1110 . 1101 1000
1 2 E . D 8

$(100101110\ .\ 11011)_2 = (12E\ .\ D8)_{16}$

Hexadecimal to binary:

Since the base of hexadecimal number is 16, i.e., the fourth power of 2, each hexadecimal number is converted into its equivalent binary digit of length four.

E.g.:

$(5D.\ 2A)_{16} = X_2$

5 D . 2 A
0101 1101 . 0010 1010

$(5D.\ 2A)_{16} = (01011101.00101010)_2$

Decimal to hexadecimal:

The decimal number is divided by 16 and carries are taken after each division and then written in the reverse order. The fractional part is multiplied by 16 and carry is taken in the forward order.

E.g.:

$(2479.859))_{10} = X_{16}$

16 | 2479
16 | 154 - 15(F)
9 - 10(A)

16 x 0.859 = 13.744 => 13 (D)

16 x 0.744 = 11.904 => 11 (B)
16 x 0.904 = 14.464 => 14 (E)
16 x 0.464 = 7.424 => 7
16 x 0.424 = 6.784 => 6

$(2479.859)_{10} = (9AF.DBE76)_{16}$

Hexadecimal to decimal:

Each digit of the hexadecimal number is multiplied by its weight and then added.

E.g.:

$(81.21)_{16} = X_{10}$
$= 8 \times 16^1 + 1 \times 16^0 + 2 \times 16^{-1} + 1 \times 16^{-2}$
$= 8 \times 16 + 1 \times 1 + 2/16 + 1/16^2$
$= (129.1289)_{10}$

$(81.21)_{16} = (129.1289)_{10}$

Binary Arithmetic

Binary Addition:
To perform the binary addition we have to follow the binary table given below.
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 => plus a carry-over of 1

Carry-overs are performed in the same manner as in decimal arithmetic. Since 1 is the largest digit in the binary system, any sum greater than 1 requires that a digit be considered over.

E.g.:

111	1010	11.01
110	1101	101.11
1001	10111	1001.00

Binary Subtraction:

To perform the binary subtraction the following binary subtraction table should be followed.

0 – 0 = 0

1 – 0 = 1
1 – 1 = 0
0 – 1 = 1 with a borrow of 1 is equivalent to 10 – 1 = 1

E.g.:

111
010
101

E.g.:

110.01
100.10
001.11

1's complement:

To obtain 1's complement of a binary number each bit of the number is subtracted from 1.

E.g.:

Binary number	1's Complement
0101	1010
1001	0110
1101	0010
0001	1110

Thus 1's complement of a binary number is the number that results when we change each 0 to a 1 and each 1 to a 0.

1's complement subtraction:

Instead of subtracting the second number from the first, the 1's complement of the second number is added to the first number. The last carry which is said to be a END AROUND CARRY, is added to get the final result.

E.g.:

```
7 -     111 -----------------> 111 +
3       011 1's complement 100
4                           1011 +
                            └→1
                           100 ❼ result
```

If there is no carry in the 1's complement subtraction, it indicates that the result is a negative and number will be in its 1's complement form. So complement it to get the final result.

.:

8 -	1000 -----------------> 1000 +		
10	1010 1's complement 0101		
4		1101 1's complement	- 0010 result

The following points should be noted down when we do 1's complement subtraction.

1. Write the first number (minuend) as such.
2. Write the 1's complement of second number(subtrahend)
3. Add the two numbers.
4. The carry that arises from the addition is said to be "end around carry".
5. End-around carry should be added with the sum to get the result.
6. If there is no end around carry find out the 1's complement of the sum and put a negative sign before the result as the result is negative.

2's Complement:

2's complement results when we add '1' to 1's complement of the given number i.e., 2's complement =1's complement + 1

Binary Number	1's complement	2's complement
1010	0101	0110
0101	1010	1011
1001	0110	0111
0001	1110	1111

2's Complement Subtraction:

Steps:

1. Write the first number as such
2. Write down the 2's complement of the second number.
3. Add the two numbers.
4. If there is a carry, discard it and the remaining part (sum) will be the result (positive).
5. If there is no carry, find out the 2's complement of the sum and put negative sign before the result as the result is negative.

E.g.:

```
1) 10 -      1010  -----------------> 1010 +
    8        1000  2's complement  1000
    2                             10010

                                   0010 ➐ result

2)  5 -      0101  -----------------> 0101 +
   12        1100  2's complement  0100
    4                              1001  2's complement  – 0111 ➐
                                                          result
```

Binary multiplication:

The table for binary multiplication is given below

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

E.g.:

1011 x 110

```
      1011 x
       110
      0000
     1011
    1011
   1000010
```

E.g.:

101.01 x 11.01

```
      101.01 x
       11.01
      101 01
     00000
    10101
   10101
  10001.0001
```

Binary division:
The table for binary division is as follows.
0 ÷ 1 = 0
1 ÷ 1 = 1

As in the decimal system division by zero is meaning less.

.:

1) 1100 ÷ 11

```
     100
11 | 1100
   | 11
     0
```

2) 1001 ÷ 10

```
     100.1
10 | 1001
   | 10
     0010
       10
        0
```

BCD Addition

Binary Coded Decimal(BCD) is a way to express each of the decimal digits with a binary code. There are only ten code groups in the BCD system. The 8421 code is a type of BCD code. In BCD each decimal digit , 0 through 9 is represented by a binary code of four bits. The designation of 8421 indicates the binary weights of the four bits (2^3,2^2,2^1,2^0). The largest 4-bit code is 1001. The numbers 1010, 1011, 1100, 1101, 1110, and 1111 are called forbidden numbers. The following table represents the decimal and 8421 equivalent numbers.

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

8421 Addition:
In 8421 addition, if there is a carry or if it results in a forbidden group, then 0110(6) should be added in order to bring the result to the 8421 mode again.

E.g.:

```
  8  +        1000 +
  7           0111
 15           1111
            + 0110
         0001 0101
```

E.g.:

```
 18  +       0001 1000 +
  2          0000 0010
 20          0001 1010
           + 0000 0110
             0010 0000
```

Alphanumeric code

Computers, printers and the other devices must process both alphabetic and numeric information. Serial coding systems have been developed to represent alphanumeric information as a series of 1's and 0's. The characters to be coded are alphabets(26), numerals (10) and special characters such as +,-, /,*, $ etc,

In order to code a character, string of binary digits is used. In order to ensure uniformity in coding, two standard codes have been used.

1. ASCII: American Standard Code for Information Interchange.
2. EBCDIC: Extended Binary Coded Decimal Interchange Code. It is an 8 bit code.

ASCII is 7-bit code of the form X_6, X_5, X_4, X_3, X_2, X_1, X_0 and is used to code two types of information. One type is the printable character such as alphabets, digits and special characters. The other type is known as control characters which represent the coded information to control the operation of the digital computer and are not printed.

CHECK YOUR PROGRESS 1

1. $2 \times 10^1 + 8 \times 10^0$ is equal to
(a) 10 (b) 280 (c) 2.8 (d) 28

2. The binary number 1101 is equal to the decimal number
(a) 13 (b) 49 (c) 11 (d) 3

3. The decimal 17 is equal to the binary number
(a) 10010 (b) 11000 (c) 10001 (d) 01001

4. The sum of 11010 + 01111 equals
(a) 101001 (b) 101010 (c) 110101 (d) 101000