1- Number Systems

In number system modern method of representing numbers symbolically is based on positional notations.

In this method, each number is represented by a string of symbols where each symbol is associated with a specific weight depending upon its positions. The total number of different symbols which are used in a particular number system is called the base or radix of the system and the weight of each position of a particular number is expressed as a power of the base. When a number is formed with the combination of the symbols, each symbol is then called a digit and the position of each symbol is referred to as the digit position.

Thus if a number system has symbols starting from 0, and the digits of the system are 0, 1, 2, ….. (r - 1) then the base or radix is r. If a number D of this system be represented by:

$D = d_{n-1}\ d_{n-2}\ \ldots\ d_i \ldots\ d_1\ d_0$

then the magnitude of this number is given by

$|D| = d_{n-1} r^{n-1} + d_{n-2} r^{n-2} + \ldots\ d_i r^i + \ldots\ d_1 r^1 + d_0 r^0$

$$\sum_{i=0}^{n-1} d_i r^i$$

Where each d_i ranges from 0 to r - 1, such that
$0 \le d_i \le r - 1,\ i = 0, 1, 2 \ldots (n - 1)$.

The digit at the extreme left has the highest positional value and is generally called the**Most Significant Digit**, or in short **MSD**.

similarly, the digit occupying the extreme right position has the least positional value and is referred to as the **Least Significant Digit** or**LSD**.

1- Decimal number

Decimal number system is the most common example of positional notational number system and all the arithmetical calculations undertaken by human being are carried out on the basis of this number system. In this system, the symbols used are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 and the base is 10. Thus the number

$d_{n-1}\ d_{n-2} \ldots d_1\ d_0$ means $d_{n-1} 10^{n-1} + d_{n-2} 10^{n-2} + \ldots + d_1 10^1 + d_0 10^0$

and this is an n-digit number. If the number be extended to the right of the decimal point, then the powers of the base will be negative starting from -1.

For example, the number 3528 has the magnitude

$3528 = 3 \times 10^3 + 5 \times 10^2 + 2 \times 10^1 + 8 \times 10^0$

and the number 26.57 has the magnitude

$26.57 = 2 \times 10 + 6 \times 10^0 + 5 \times 10^{-1} + 7 \times 10^{-2}$

2- Binary Number System

Binary number system uses two symbols 0 and 1 and its radix is 2. The symbols 0 and 1 are generally called **BITS** which is a contraction of the two words Binary digits.

An n-bit binary number of the form $\mathbf{a_{n-1}\, a_{n-2} \,\ldots\ldots\, a_1\, a_0}$
where each a_i ($i = 0, 1, \ldots.\ n - 1$) is either 0 or 1 has the magnitude.

$\mathbf{a_{n-1}\, 2^{n-1} + a_{n-2}\, 2^{n-2} + \ldots\ldots + a_1\, 2^1 + a_0 2^0.}$

For fractional binary numbers, the base has negative integral powers starting with -1 for the bit position just after the binary point.

The bit at the extreme left of a binary number has the highest positional value and is usually called the **Most Significant Bit** or **MSB**. Similarly, the bit occupying the extreme right position of a given binary number has the least positional value and is referred to as the **Least Significant Bit** or **LSB**.

To facilitate the distinction between different number systems, we generally use the respective radix as a subscript of the number. However the subscript will not be used when there is no scope of confusion.

a few examples on binary numbers and their decimal equivalents are given below:

examp1: $\mathbf{101101_2 = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0}$

$\mathbf{= 32 + 0 + 8 + 4 + 0 + 1}$

$\mathbf{= 45_{10}}$

The above results can be more clearly expressed in the following manner:

Binary Number	1	0	1	1	0	1	Decimal Number
Power of base	2^5	2^4	2^3	2^2	2^1	2^0	
Decimal equivalent	32	16	8	4	2	1	
Magnitude of each term	32	0	8	4	0	1	45

Exmple2:

Binary point

111.1011_2

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

$= 4 + 2 + 1 + .5 + 0 + .125 + .0625$

$= 7.6875_{10}$

The above results can be more clearly expressed in the following manner:

Binary Number	1	1	1	.	1	0	1	1	Decimal Number
Power of base	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}	
Decimal equivalent	4	2	1		.5	.25	.125	.0625	
Magnitude of each term	4	2	1		.5	0	.125	.0625	7.6875

Why binary numbers are used?

It may be observed from the discussions of the preceding section that the use of a base smaller than 10 requires more positions to represent a given decimal number. As for example, the binary number 10101 requires 5 bit positions to represent the decimal number 21 which requires two positions for its decimal representation. This is a major disadvantage of the binary number system. In spite of this fact, all the modern digital computers have been basically designed on the basis of binary number system.

Why this bias to binary number?

There are several reasons for this.

The first and foremost reason is that electronic components, as a natural coincidence, operate in a binary mode. A switch is either open/off (called 0 state) or closed/on (called 1 state); a transistor is either not conducting (0 state) or is conducting (1 state).

- **This two-state nature of the electronic components can be easily expresses with the help of binary numbers.**

- **The second reason is that computer circuits have to handle only two bits instead of 10 digits of the decimal system. This simplifies the design of the machine, reduces the cost and improves the reliability.**

- Lastly, binary number system is used because all the operations that can be done in the decimal system can also be done with a binary number of radix 2.

3- Octal Number System

Octal number system has a base or radix 8. Eight different symbols, namely **0, 1, 2, 3, 4, 5, 6, 7** are used to represent octal numbers. Conversion of octal numbers to their decimal equivalents can be accomplished by using the same rule which was followed to convert binary numbers to decimal numbers, except that we now have a radix 8 instead of 2. Thus the octal number 273 has the decimal equivalent.

$\mathbf{273_8}$

$= 2 \times 8^2 + 7 \times 8^1 + 3 \times 8^0$

$= 128 + 56 + 3$

$= \mathbf{187_{10}}$

4- Hexa -decimal Number System

The hexa-decimal number system has a radix or base 16. It requires 16 symbols to represent a number in this system. The symbols are **0 to 9, A, B, C, D, E, F** where the symbols **A, B, C, D, E, F** represent the decimal numbers **10, 11, 12, 13, 14, 15** respectively.

Example : Convert $B6A_{16}$ to its decimal equivalent.

Solution:

$B6A_{16}$

$= 11 \times 16^2 + 6 \times 16^1 + 10 \times 16^0$

$= 2816 + 96 + 10$

$= 2922_{10}$ **Therefore, $B6A_{16} = 2922_{10}$**

Conversion Of Numbers

Conversion of numbers from one system to another becomes necessary to understand the process and the logic of the operations of a computer system. It is not very difficult to convert numbers from one base to another.

There are several traditional methods of converting the numbers from binary to decimal conversion. We shall discuss here the two most commonly used methods, namely: "**Expansion or value box Method**" **and** "**Multiplication and Division Method**"

Note :We will first discuss about the conversion of binary numbers to their decimal equivalents , after that applying these methods to octal and hexa-decimal number respectively.

(i) Expansion Method:

In expansion method the conversion of binary numbers to their decimal equivalents are shown with the help of the examples.

1. Convert the decimal numbers to their binary equivalents:

Exp1: 256

Solution: 256

256	128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0	0

Since the given number 256 appears in the first row, we put 1 in the slot below 256 and fill all the other slots to the right of this slot with zeros.

Thus, 256_{10} = 100000000_2

exp2: 77

Solution:

77

64	32	16	8	4	2	1
1	0	0	1	1	0	1

The given number is less than 128 but greater than 64. We therefore put 1 in the slot corresponding to 64 in the first row. Next, we subtract 64 from 77 and get 13 as remainder.

This remainder is less than 16 and greater than 8. So we put 1 in the slot corresponding to 8 and subtract 8 from 13. This gives 13 - 8 = 5. This remainder is greater than 4 and less than 8.

Hence we put 1 in the slot corresponding to 4 and subtracting 4 from 5 we get 1. Now, 1 is present in the right hand most slot of the first row. We, therefore, put 1 in the corresponding slot and fill all other slots with zeros.

Thus, 77_{10} = 1001101_2.

2. Conversion of decimal fractions to binary fractions

Conversion of decimal fractions to binary fractions may also be accomplished by using similar method. Let us observe the procedure with the help of the following example:

Exp1 : Convert 0.675_{10} to its binary equivalent.

Solution:

1	.5	.25	.125	0.625	.03125
	1	0	1	0	1

Subtract .5 from the given number to get .675 - .5 = .175 and place 1 in the slot corresponding to .5 of the first row.

Now the number .175 is less than .25 and greater than .125. So, we put 1 in the slot corresponding to the number .125 of the first row and subtract .125 from .175 to get .175 - .125 = .05. The remainder .05 is less than .0625 but greater than .03125.

Hence we put 1 in the slot corresponding to 0.3125 and the subtraction given .05 - .03125 = .01875 and continue the process. The other slots are then filled with zeros.

Thus, $.675_{10} = (.10101...)_2$

Note:

It should be noted that the conversion of decimal fractions to binary fractions may not be exact and the process is to be continued until there is no remainder or the remainder is less than the order of accuracy desired.

(ii) Multiplication and Division Method

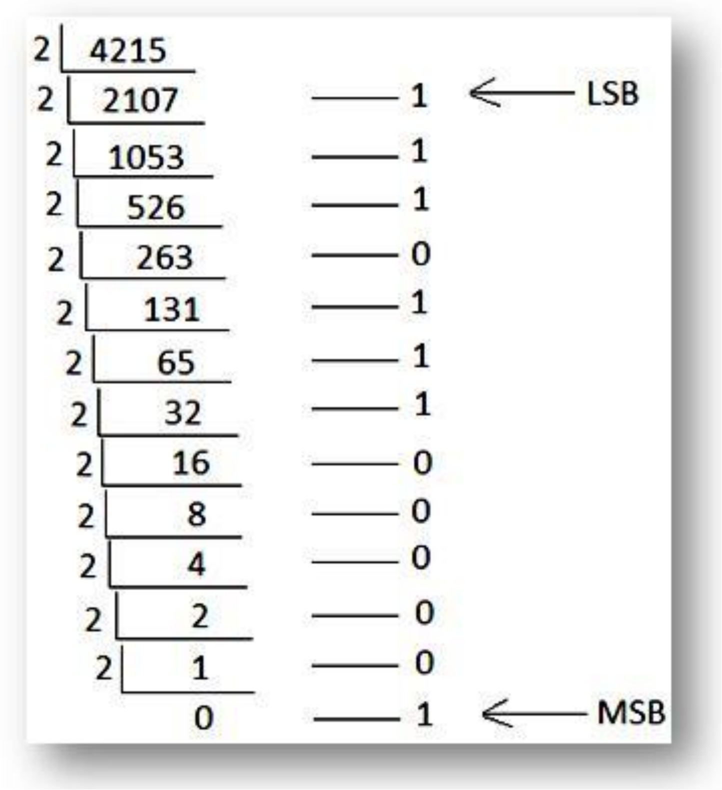
The expiation or value box method of converting numbers from decimal to binary is laborious and time consuming and is suitable for small numbers when it can be performed mentally. It is advisable not to use it for large numbers. The conversion of large numbers may be conveniently done by multiplication and division method which is described below.

To effect the conversion of positive integers of the decimal system to binary numbers the decimal number is repeatedly divided by the base of the binary number system, i.e., by 2. The division is to be carried until the quotient is zero and the remainder of each division is recorded on the right. The binary equivalent of the decimal number is then obtained by writing down the successive remainders. The first remainder is the least significant bit and the last one is the most significant bit of the binary number. Thus the binary equivalent is written from the bottom upwards.

We explain conversion of numbers using multiplication and division method with the help of following example.

Exp1: Convert 4215_{10} to its binary equivalent

Solution:



Therefore, $4215_{10} = 1000001110111_2$

The conversion of decimal fractions to binary fractions is accomplished by multiplying repeatedly the decimal fraction by the base 2 of the binary number. The integral part after each multiplication is either 0 or 1. The equivalent binary fraction is obtained by writing the integral parts of each product to the right of the binary point in the same sequence. If the fractional part of the product becomes exactly zero at a certain stage, then the binary fraction is finite, otherwise, the fraction is non-terminating and then we find the binary fraction upto the desired degree of accuracy. We explain the process with the help of the following examples.

Exp2. Convert the following decimal numbers to their binary equivalents:

(a) 0.375

Solution:

Decimal Numbers to Binary Number Conversion Table		
Multiplication	**Integer**	**Fraction**
0.375 × 2 = 0.75	**0**	**.75**
0.75 × 2 = 1.5	**1**	**.5**
.5 × 2 = 1.0	**1**	**0**

Therefore, $0.375_{10} = 0.011_2$

(b) 0.435

Solution:

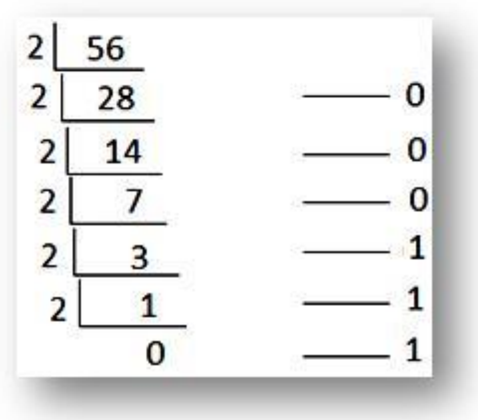
Decimal Numbers to Binary Number Conversion Table		
Multiplication	**Integer**	**Fraction**
0.435 × 2 = 0.87	**0**	**.87**
0.87 × 2 = 1.74	**1**	**.74**
.74 × 2 = 1.48	**1**	**.48**
.48 × 2 = 0.96	**0**	**.96**
.96 × 2 = 1.92	**1**	**.92**

Therefore, $0.435_{10} = (0.01101...)_2$

☞**Note :** Fox mixed number, we will have to separate the number into its integral and fractional parts and find the binary equivalent of each part independently. Finally, we add the two parts to get the binary equivalent of the given number.

Exp3.: Convert $(56.75)_{10}$ to its binary equivalent.

Solution:
At first we find the binary equivalent of 56.



Therefore, $56_{10} = 111000_2$
The binary equivalent of 0.75 is obtained below:

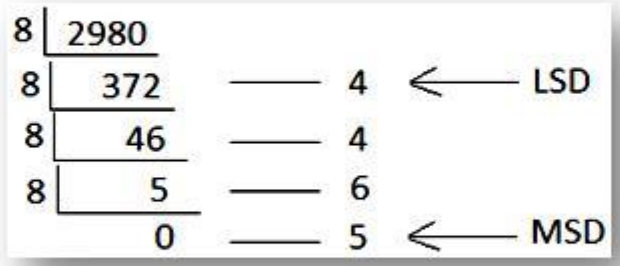
Decimal Numbers to Binary Number Conversion Table		
Multiplication	**Integer**	**Fraction**
0.75 × 2 = 1.5	**1**	**.5**
0.5 × 2 = 1.0	**1**	**0**

Therefore, $0.75_{10} = 0.11_{10}$
Hence $56.75_{10} = 111000.11_{10}$

Exp3 : Convert the decimal numbers to their octal equivalents:

(a) 2980

Solution:



Hence $2980_{10} = 5644_8$

(b) 0.685

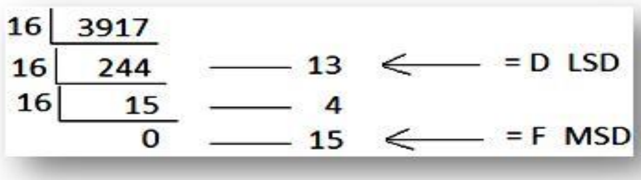
Solution:

Decimal Numbers to Binary Number Conversion Table		
Multiplication	**Integer**	**Fraction**
0.685 × 8 = 5.480	**5**	**.48**
0.48 × 8 = 3.84	**3**	**.84**
.84 × 8 = 6.72	**6**	**.72**
.72 × 8 = 5.76	**5**	**.76**

4- conversion of hexa-decimal numbers to their decimal equivalents

The conversion of hexa-decimal numbers to their decimal equivalents is straight-forward and follows the same rules as that of octal or binary to decimal. Similarly, conversion of decimal to hexa-decimal may be worked out with the help of division or multiplication, as the case may be, by the radix 16.

Exp 4: Convert 3917_{10} to its hexa-decimal equivalent
Solution:



Therefore, 3917_{10} = $F4D_{16}$

Conversion of Binary Numbers to Octal or Hexa - decimal Numbers

Conversion of binary numbers to octal or hexa-decimal numbers and vice-versa may be accomplished very easily.

Since a string of 3 bits can have 8 different permutations, it follows that each 3-bit string is uniquely represented by one octal digit. Similarly, since a string of 4 bits has 16 different permutations each 4 bit string represents a hexa-decimal digit uniquely. The table below gives the decimal numbers 0 to 15 and their binary, octal and hexa-decimal equivalents and also the corresponding 3-bit and 4-bit strings.

Conversion of binary numbers to octal or hexa-decimal numbers and vice versa:

Decimal	Binary	Octal	3-bit String	Hexa-decimal	4-bit String
0	0	0	000	0	0000
1	1	1	001	1	0001
2	10	2	010	2	0010
3	11	3	011	3	0011
4	100	4	100	4	0100
5	101	5	101	5	0101
6	110	6	110	6	0110
7	111	7	111	7	0111
8	1000	10	-	8	1000
9	1001	11	-	9	1001
10	1010	12	-	A	1010
11	1011	13	-	B	1011
12	1100	14	-	C	1100
13	1101	15	-	D	1101
14	1110	16	-	E	1110
15	1111	17	-	F	1111

Thus to convert a binary number to its octal equivalent we arrange the bits into groups of 3 starting at the binary point and move towards the MSB. We then replace each group by the corresponding octal digit. If the number of bits is not a multiple of 3, we add necessary number of zeros to the left of MSB. For binary fractions, we have to work towards the right of the binary point and follow the same procedure. Similarly, for conversion of octal numbers to binary numbers, we have to replace each octal digit by its 3-bit binary equivalent.

The same procedure is to be adopted in the case of hexa-decimal numbers and vice versa by converting the given numbers to binary numbers first with the help of above procedure and then converting these binary numbers to hexa-decimal numbers. Conversion to decimal may also be accomplished by the same procedure.

The following examples explain the converting method :

1. Convert the following to octal numbers:

(a) 1110101110_2

Solution: $001\underline{110}101\underline{110} = 001\ 110\ 101\ 110 = 1656_8$

Hence the required octal equivalent is 1656.

(b) 111101.01101_2

Solution: $111\underline{101}.011\underline{010}_2 = 75.32_8$

Hence the required octal equivalent is 75.32.

2. Convert the following to their binary equivalents:

(a) 1573_8

Solution: $1573_8 = 001\ 101\ 111\ 011 = 1101111011_2$

Hence the required binary number is 1101111011.

(b) 64.175_8

Solution: $64.175_8 = 110\ 100\ .\ 001\ 111\ 101 = 110100.001111101_2$

Hence the required binary number is 110100.001111101.

3. Convert the following to hexa-decimal numbers:

(a) 1111101101_2

Solution: $\underline{0011}1110\underline{1101} = 0011\ 1110\ 1101 = 3ED_{16}$

Therefore, $11\ 1110\ 1101_2 = 3ED_{16}$

(b) 11110.01011_2

Solution:
$1\underline{1110}.01011_2 = 0001\ 1110\ .\ 0101\ 1000 = 1E.58_{16}$

Therefore, $11110.01011_2 = 1E.58_{16}$

4. Convert the following to binary equivalents:

(a) $A748_{16}$

Solution: $A748_{16} = 1010\ 0111\ 0100\ 1000 = 1010011101001000_2$

Hence the required binary equivalent is 1010011101001000.

(b) $BA2.23C_{16}$

Solution: $BA2.23C_{16} = 1011\ 1010\ 0010\ .\ 0010\ 0011\ 1100_2$

= 101110100010.0010001111

Hence the required binary equivalent is 101110100010 . 0010001111

5. Convert 1573_8 to hexa-decimal

Solution: 1573_8 = 001101111011 = 0011 0111 1011 $37B_{16}$

Hence $1573_8 = 37B_{16}$

6. Convert $A748_{16}$ to octal equivalents.

Solution: $A748_{16}$ = 1010 0111 0100 1000= 001 010 011 101 001 000

= 123510_8

Therefore, $A748_{16} = 123510_8$

7. Convert the following to decimal numbers:

(a) 725_8

Solution: 725_8 = 111010101= 256 + 128 + 64 + 16 + 4 + 1= 469_{10}

Therefore, $725_8 = 469_{10}$

(b) $D9F_{16}$

Solution: $D9F_{16}$ = 1101 1001 1111= 110110011111

= 2048 + 1024 + 256 + 128 + 16 + 8 + 4 + 2 + 1 = 3487_{10}

Therefore, $D9F_{16} = 3487_{10}$