NUMBER SYSTEMS

1.1Introduction

There are several number systems which we normally use, such as decimal, binary, octal, hexadecimal, etc. Amongst them we are most familiar with the decimal number system. These systems are classified according to the values of the base of the number system. The number system having the value of the base as 10 is called a decimal number system, whereas that with a base of 2 is called a binary number system. Likewise, the number systems having base 8 and 16 are called octal and hexadecimal number systems respectively.
With a decimal system we have 10 different digits, which are 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. But a binary system has only 2 different digits: 0 and 1. Hence, a binary number cannot have any digit other than 0 or 1. So to deal with a binary number system is quite easier than a decimal system. Now, in a digital world, we can think in binary nature, e.g., a light can be either off or on. There is no state in between these two. So we generally use the binary system when we deal with the digital world. Here comes the utility of a binary system. We can express everything in the world with the help of only two digits i.e., 0 and 1. For example, if we want to express 2510 in binary we may write 11001_2. The right most digit in a number system is called the 'Least Significant Bit' (LSB) or 'Least Significant Digit' (LSD). And the left most digit in a number system is called the 'Most Significant Bit' (MSB) or 'Most Significant Digit' (MSD). Now normally when we deal with different number systems we specify the base as the subscript to make it clear which number system is being used.
In an octal number system there are 8 digits 0, 1, 2, 3, 4, 5, 6, and 7. Hence, any octal number cannot have any digit greater than 7. Similarly, a hexadecimal number system has 16 digits 0 to 9 and the rest of the six digits are specified by letter symbols as A,B, C, D, E, and F. Here A, B, C, D, E, and F represent decimal 10, 11, 12, 13, 14, and 15 respectively. Octal and hexadecimal codes are useful to write assembly level language.

In general, we can express any number in any base or radix "X." Any number with base X, having n digits to the left and m digits to the right of the decimal point, can be expressed as:

$$a_n\ X^{n-1} + a_{n-1}\ X^{n-2} + a_{n-2}\ X^{n-3} + \ldots + a_2\ X^{1} + a_1\ X^{0} + b_1\ X^{-1} + b_2\ X^{-2} + \ldots + b_m\ X^{-m}$$

where an is the digit in the nth position. The coefficient an is termed as the MSD or Most Significant Digit and bm is termed as the LSD or the Least Significant Digit.

1.2 CONVERSION BETWEEN NUMBER SYSTEMS

It is often required to convert a number in a particular number system to any other number system, e.g., it may be required to convert a decimal number to binary or octal or hexadecimal. The reverse is also true, i.e., a binary number may be converted into decimal and so on.

1.2.1 Decimal-to-binary Conversion

Now to convert a number in decimal to a number in binary we have to divide the decimal number by 2 repeatedly, until the quotient of zero is obtained. This method of repeated division by 2 is called the 'double-dabble' method. The remainders are noted down for each of the division steps. Then the column of the remainder is read in reverse order i.e., from bottom to top order. We try to show the method with an example shown in Example 1.1.

Example 1.1. Convert $(26)_{10}$ into a binary number.

Solution:

Division	Quotient	Generated remainder
$\frac{26}{2}$	13	0
$\frac{13}{2}$	6	1
$\frac{6}{2}$	3	0
$\frac{3}{2}$	1	1
$\frac{1}{2}$	0	1

Hence the converted binary number is $(11010)_2$.

1.2.2 Decimal-to-octal Conversion

Similarly, to convert a number in decimal to a number in octal we have to divide the decimal number by 8 repeatedly, until the quotient of zero is obtained. This method of repeated division by 8 is called 'octal-dabble.' The remainders are noted down for each of the division steps. Then the column of the remainder is read from bottom to top order, just as in the case of the double-dabble method. We try to illustrate the method with an example shown in Example 1.2.

<u>Example 1.2.</u> Convert $(426)_{10}$ into an octal number.

Solution:

Division	Quotient	Generated remainder
$\frac{426}{8}$	53	2
$\frac{53}{8}$	6	5
$\frac{6}{8}$	0	6

Hence the converted octal number is $(652)_8$.

1.2.3 Decimal-to-hexadecimal Conversion

The same steps are repeated to convert a number in decimal to a number in hexadecimal. Only here we have to divide the decimal number by 16 repeatedly, until the quotient of zero is obtained. This method of repeated division by 16 is called 'hex-dabble.' The remainders are noted down for each of the division steps. Then the column of the remainder is read from bottom to top order as in the two previous cases. We try to discuss the method with an example shown in Example 1.3.

<u>Example 1.3.</u> Convert $(348)_{10}$ into a hexadecimal number.

Solution:

Division	Quotient	Generated remainder
$\frac{348}{16}$	21	12
$\frac{21}{16}$	1	5
$\frac{1}{16}$	0	1

Hence the converted hexadecimal number is $(15C)_{16}$.

1.2.4 Binary-to-decimal Conversion

Now we discuss the reverse method, *i.e.*, the method of conversion of binary, octal, or hexadecimal numbers to decimal numbers. Now we have to keep in mind that each of the binary, octal, or hexadecimal number system is a positional number system, *i.e.*, each of the digits in the number systems discussed above has a positional weight as in the case of the decimal system. We illustrate the process with the help of examples.

Example 1.4. Convert $(10110)_2$ into a decimal number.

Solution. The binary number given is 1 0 1 1 0

Positional weights *4 3 2 1 0*

The positional weights for each of the digits are written in italics below each digit. Hence the decimal equivalent number is given as:

$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

$= 16 + 0 + 4 + 2 + 0$

$= (22)_{10}$.

Hence we find that here, for the sake of conversion, we have to multiply each bit with its positional weights depending on the base of the number system.

1.2.5 Octal-to-decimal Conversion

Example 1.5. Convert 3462_8 into a decimal number.

Solution. The octal number given is 3 4 6 2

Positional weights 3 2 1 0

The positional weights for each of the digits are written in italics below each digit. Hence the decimal equivalent number is given as:

$3 \times 8^3 + 4 \times 8^2 + 6 \times 8^1 + 2 \times 8^0$

$= 1536 + 256 + 48 + 2$

$= (1842)_{10}$.

1.2.6 Hexadecimal-to-decimal Conversion

Example 1.6. *Convert $42AD_{16}$ into a decimal number.*

Solution. The hexadecimal number given is 4 2 A D

Positional weights *3 2 1 0*

The positional weights for each of the digits are written in italics below each digit. Hence the decimal equivalent number is given as:

$4 \times 16^3 + 2 \times 16^2 + 10 \times 16^1 + 13 \times 16^0$

$= 16384 + 512 + 160 + 13$

$= (17069)_{10}$.

1.2.7 Fractional Conversion

So far we have dealt with the conversion of integer numbers only. Now if the number contains the fractional part we have to deal in a different way when converting the number from a different number system (*i.e.*, binary, octal, or hexadecimal) to a decimal number system or vice versa. We illustrate this with examples.

Example 1.7. Convert 1010.011_2 into a decimal number.

Solution. The binary number given is 1 0 1 0. 0 1 1

Positional weights *3 2 1 0 -1-2-3*

The positional weights for each of the digits are written in italics below each digit.

Hence the decimal equivalent number is given as:

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

$= 8 + 0 + 2 + 0 + 0 + 0.25 + 0.125$

$= (10.375)_{10}$.

Example 1.8. *Convert 362.35_8 into a decimal number.*

Solution. The octal number given is 3 6 2. 3 5

Positional weights *2 1 0 -1-2*

The positional weights for each of the digits are written in italics below each digit.

Hence the decimal equivalent number is given as:

$3 \times 8^2 + 6 \times 8^1 + 2 \times 8^0 + 3 \times 8^{-1} + 5 \times 8^{-2}$

$= 192 + 48 + 2 + 0.375 + 0.078125$

$= (242.453125)_{10}$.

Example 1.9. *Convert $42A.12_{16}$ into a decimal number.*

Solution. The hexadecimal number given is 4 2 A. 1 2

Positional weights *2 1 0 -1-2*

The positional weights for each of the digits are written in italics below each digit.

Hence the decimal equivalent number is given as:

$4 \times 16^2 + 2 \times 16^1 + 10 \times 16^0 + 1 \times 16^{-1} + 1 \times 16^{-2}$

$= 1024 + 32 + 10 + 0.0625 + 0.00390625$

$= (1066.06640625)_{10}$.

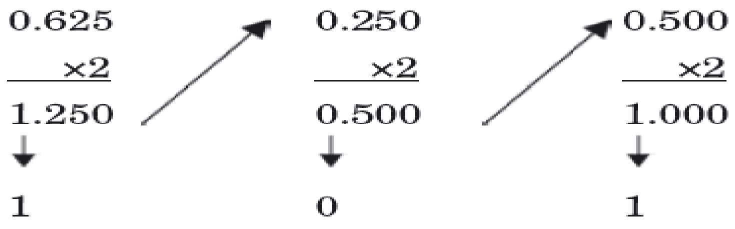
Example 1.10. Convert *25.625_{10}* into a binary number.

Solution.

Division	Quotient	Generated remainder
$\frac{25}{2}$	12	1
$\frac{12}{2}$	6	0
$\frac{6}{2}$	3	0
$\frac{3}{2}$	1	1
$\frac{1}{2}$	0	1

Therefore, $(25)_{10} = (11001)_2$.

Fractional Part:



i.e., $(0.625)_{10} = (0.101)_2$

Therefore, $(25.625)_{10} = (11001.101)_2$

1.2.8 Conversion from a Binary to Octal Number and Vice Versa

We know that the maximum digit in an octal number system is 7, which can be represented as 111_2 in a binary system. Hence, starting from the LSB, we group three digits at a time and replace them by the decimal equivalent of those groups and we get the final octal number.

Example 1.11. Convert 101101010_2 into an equivalent octal number.

Solution. The binary number given is 101101010

Starting with LSB and grouping 3 bits 101 101 010

Octal equivalent 5 5 2

Hence the octal equivalent number is $(552)_8$.

Example 1.12. Convert 1011110_2 into an equivalent octal number.

Solution. The binary number given is 1011110

Starting with LSB and grouping 3 bits 001 011 110

Octal equivalent 1 3 6

Hence the octal equivalent number is $(136)_8$.

Since at the time of grouping the three digits in Example 1.14 starting from the LSB, we find that the third group cannot be completed, since only one 1 is left out in the third group, so we complete the group by adding two 0s in the MSB side. This is called left padding of the number with 0. Now if the number has a fractional part then there will be two different classes of groups—one for the integer part starting from the left of the decimal point and proceeding toward the left and the second one starting from the right of the decimal point and proceeding toward the right. If, for the second class, any 1 is left out, we complete the group by adding two 0s on the right side. This is called right-padding.

Example 1.13. Convert 1101.0111_2 into an equivalent octal number.

Solution. The binary number given is 1101.0111

Grouping 3 bits 001 101. 011 100

Octal equivalent: 1 5 3 4

Hence the octal number is $(15.34)_8$.

Now if the octal number is given and you're asked to convert it into its binary equivalent, then each octal digit is converted into a 3-bit-equivalent binary number and combining all those digits we get the final binary equivalent.

Example 1.14. Convert 235_8 into an equivalent binary number.

Solution. The octal number given is 2 3 5

3-bit binary equivalent 010 011 101

Hence the binary number is $(010011101)_2$.

Example 1.15. Convert 47.321_8 into an equivalent binary number.

Solution. The octal number given is 4 7 3 2 1

3-bit binary equivalent 100 111 011 010 001

Hence the binary number is $(100111.011010001)_2$.

1.2.9 Conversion from a Binary to Hexadecimal Number and Vice Versa

We know that the maximum digit in a hexadecimal system is 15, which can be represented by 1111_2 in a binary system. Hence, starting from the LSB, we group four digits at a time and replace them with the hexadecimal equivalent of those groups and we get the final hexadecimal number.

Example 1.16. Convert 11010110_2 into an equivalent hexadecimal number.

Solution. The binary number given is 11010110

Starting with LSB and grouping 4 bits 1101 0110

Hexadecimal equivalent D 6

Hence the hexadecimal equivalent number is $(D6)_{16}$.

Example 1.17. Convert 110011110_2 into an equivalent hexadecimal number.

Solution. The binary number given is 110011110

Starting with LSB and grouping 4 bits 0001 1001 1110

Hexadecimal equivalent 1 9 E

Hence the hexadecimal equivalent number is $(19E)_{16}$.

Since at the time of grouping of four digits starting from the LSB, in Example 1.19 we find that the third group cannot be completed, since only one 1 is left out, so we complete the group by adding three 0s to the MSB side. Now if the number has a fractional part, as in the case of octal numbers, then there will be two different classes of groups—one for the integer part starting from the left of the decimal point and proceeding toward the left and the second one starting from the right of the decimal point and proceeding toward the right. If, for the second class, any uncompleted group is left out, we complete the group by adding 0s on the right side.

Example 1.18. Convert 111011.011_2 into an equivalent hexadecimal number.

Solution. The binary number given is 111011.011

Grouping 4 bits 0011 1011. 0110

Hexadecimal equivalent 3 B 6

Hence the hexadecimal equivalent number is $(3B.6)_{16}$.

Now if the hexadecimal number is given and you're asked to convert it into its binary equivalent, then each hexadecimal digit is converted into a 4-bit-equivalent binary number and by combining all those digits we get the final binary equivalent.

Example 1.19. *Convert* $29C_{16}$ *into an equivalent binary number.*

Solution. The hexadecimal number given is 2 9 C

4-bit binary equivalent 0010 1001 1100

Hence the equivalent binary number is $(001010011100)_2$.

Example 1.20. *Convert* $9E.AF2_{16}$ *into an equivalent binary number.*

Solution. The hexadecimal number given is 9 E A F 2

4-bit binary equivalent 1001 1110 1010 1111 0010

Hence the equivalent binary number is $(10011110.101011110010)_2$.

1.2.10 Conversion from an Octal to Hexadecimal Number and Vice Versa

Conversion from octal to hexadecimal and vice versa is sometimes required. To convert an octal number into a hexadecimal number the following steps are to be followed:

(*i*) First convert the octal number to its binary equivalent (as already discussed above).

(*ii*) Then form groups of 4 bits, starting from the LSB.

(*iii*) Then write the equivalent hexadecimal number for each group of 4 bits.

Similarly, for converting a hexadecimal number into an octal number the following steps are to be followed:

(*i*) First convert the hexadecimal number to its binary equivalent.

(*ii*) Then form groups of 3 bits, starting from the LSB.

(*iii*) Then write the equivalent octal number for each group of 3 bits.

Example 1.21. Convert the following hexadecimal numbers into equivalent octal numbers.

(a) A72E *(b) 4.BF85*

Solution:

(*a*) Given hexadecimal number is A 7 2 E

Binary equivalent is 1010 0111 0010 1110

= 1010011100101110

Forming groups of 3 bits from the LSB	001	010	011	100	101	110
Octal equivalent	1	2	3	4	5	6

Hence the octal equivalent of $(A72E)_{16}$ is $(123456)_8$.

(*b*) Given hexadecimal number is 4 B F 8 5

Binary equivalent is 0100 1011 1111 1000 0101

= 0100.1011111110000101

Forming groups of 3 bits	100.	101	111	111	000	010	100
Octal equivalent	4	5	7	7	0	2	4

Hence the octal equivalent of $(4.BF85)_{16}$ is $(4.577024)_{8}$.

Example 1.22. Convert $(247)_8$ into an equivalent hexadecimal number.

Solution. Given octal number is 2 4 7

Binary equivalent is 010 100 111

= 010100111

Forming groups of 4 bits from the LSB 1010 0111

Hexadecimal equivalent A 7

Hence the hexadecimal equivalent of $(247)_8$ is $(A7)_{16}$.

Example 1.23. Convert $(36.532)_8$ into an equivalent hexadecimal number.

Solution. Given octal number is 3 6 5 3 2

Binary equivalent is 011 110 101 011 010

=011110.101011010

Forming groups of 4 bits 0001 1110. 1010 1101

Hexadecimal equivalent 1 E. A D

Hence the hexadecimal equivalent of $(36.532)_8$ is $(1E.AD)_{16}$.

1.3 Binary Arithmetic

1.3.1 Binary Addition

The four basic rules for adding binary digits (bits) are as follows:

0+0=0 Sum of 0 with a carry 0

0+1=1 Sum of 1 with a carry 0

1+0=1 Sum of 1 with a carry 0

1+1=1 0 Sum of 0 with a carry 1

Examples:

$$\begin{array}{r} 110 \\ +\ 100 \\ \hline 1010 \end{array} \qquad \begin{array}{r} 6 \\ +\ 4 \\ \hline 10 \end{array} \qquad \begin{array}{r} 111 \\ +\ 011 \\ \hline 1010 \end{array} \qquad \begin{array}{r} 7 \\ +\ 3 \\ \hline 10 \end{array} \qquad \begin{array}{r} 1111 \\ +\ 1100 \\ \hline 11011 \end{array} \qquad \begin{array}{r} 15 \\ +\ 12 \\ \hline 27 \end{array}$$

1.3.2 Binary Subtraction

The four basic rules for subtracting are as follows:

0-0=0

1-1=0

1-0=1

0-1=1 0-1 with a borrow of 1

Examples:

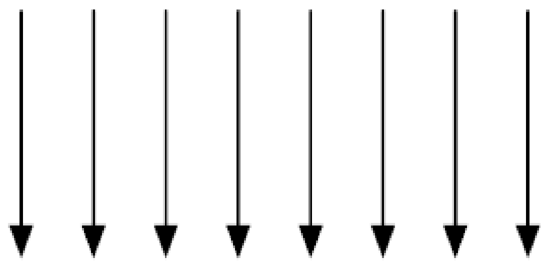
11	3	11	3	101
- 01	- 1	- 10	- 2	- 011
10	2	01	1	010

5	110	6	101101	45
- 3	- 101	- 5	- 001110	- 14
2	001	1	011111	31

1.4 1's And 2's Complement of Binary Number

The 1's complement and the 2's complement of binary number are important because they permit the representation of negative numbers.

Binary Number **1 0 1 1 0 0 1 0**



1'sComplement **0 1 0 0 1 1 0 1**

2's Complement of a binary number is found by adding 1 to the LSB of the 1's Complement.
2's Complement= (1's Complement) +1

```
Binary number          10110010
1'scomplement          01001101
Add 1               +         1
---------------------------------------
2's complement         01001110
```