Chapter 3

Number Systems

We saw in the previous chapter that a computer stores data internally in a format that is not easily readable by human beings. This is the reason why input and output (I/O) interfaces are required. Every computer stores numbers, letters, and other special characters in coded form. Before going into the details of these codes, it is essential to have a basic understanding of number system. Hence, this chapter familiarizes you with the fundamentals of number system. It also introduces some commonly used number systems by computer professionals and relationship among them.

NON-POSITIONAL NUMBER SYSTEMS

Number systems are of two types – non-positional and positional. In early days, human beings counted on fingers. When counting beyond ten fingers, they used stones, pebbles, or sticks to indicate values. This method of counting uses an additive approach or *non-positional number system*. In this system, we have symbols such as I for 1, II for 2, III for 3, IIII for 4, IIIII for 5, etc. Each symbol represents the same value regardless of its position in a number, and to find the value of a number, one has to count the number of symbols present in the number. Since it is very difficult to perform arithmetic with such a number system, positional number systems were developed.

POSITIONAL NUMBER SYSTEMS

In a *positional number system*, there are only a few symbols called digits. These symbols represent different values, depending on the position they occupy in a number. The value of each digit in such a number is determined by three considerations:

1. The digit itself,
2. The position of the digit in the number, and
3. The base of the number system (where *base* is defined as the total number of digits available in the number system).

In our day-to-day life, we use *decimal number system*. In this system, base is equal to 10 because there are altogether ten symbols or digits (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). You know that in decimal number system, successive positions to the left of the decimal point represent units, tens, hundreds, thousands, etc. However, notice that each position represents a specific power of the base (10). For example, decimal number 2586 (written as 2586_{10}) consists of digit 6 in units position, 8 in tens position, 5 in hundreds position, and 2 in thousands position, and its value can be written as:

$$(2 \times 10^3) + (5 \times 10^2) + (8 \times 10^1) + (6 \times 10^0) = 2000 + 500 + 80 + 6 = 2586$$

Observe that the same digit signifies different values, depending on the position it occupies in the number. For example,

In 2586_{10} the digit 6 signifies $6 \times 10^0 = 6$
In 2568_{10} the digit 6 signifies $6 \times 10^1 = 60$
In 2658_{10} the digit 6 signifies $6 \times 10^2 = 600$
In 6258_{10} the digit 6 signifies $6 \times 10^3 = 6000$

Hence, we can represent any number by using the available digits and arranging them in various positions.

The principles that apply to decimal number system, also apply to any other positional number system. It is important to keep track of only the base of the number system in which we are working.

The value of the base in all positional number systems suggests the following characteristics:

1. The value of the base determines the total number of different symbols or digits available in the number system. The first of these choices is always zero.
2. The maximum value of a single digit is always equal to one less than the value of the base.

We describe below some positional number systems used in computer design and by computer professionals.

Binary Number System

Binary number system is like decimal number system, except that the base is 2, instead of 10. We can use only two symbols or digits (0 and 1) in this number system. Note that the largest single digit is 1 (one less than the base). Each position in a binary number represents a power of the base (2). Hence, in this system, the rightmost position is units (2^0) position, the second position from the right is 2's (2^1) position, and proceeding in this way, we have 4's (2^2) position, 8's (2^3) position, 16's (2^4) position, and so on. Therefore, decimal equivalent of binary number 10101 (written as 10101_2) is:

$$(1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = 16 + 0 + 4 + 0 + 1 = 21$$

In order to be specific about which system we are referring to, it is a common practice to indicate the base as a subscript. Hence, we write:

$$10101_2 = 21_{10}$$

The short form of "binary digit" is *bit*. Hence, a "bit" in computer terminology means either a 0 or 1. An n-bit number is a binary number consisting of 'n' bits. Figure 3.1 lists all 3-bit numbers along with their decimal equivalent. Remember that we have only two digits, 0 and 1, in binary number system and hence, binary equivalent of decimal number 2 has to be stated as 10 (read as one, zero). Another important point to note is that with 3 bits (positions), only 8 (2^3) different patterns of 0s and 1s are possible, and it may be seen from Figure 3.1 that a 3-bit number can have one of the 8 values in the range 0 to 7. In fact, any decimal number in the range 0 to 2^{n-1} can be represented in binary form as an n-bit number.

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

Figure 3.1. 3-bit numbers with their decimal values.

Every computer stores numbers, letters, and other special characters in binary form. There are several occasions when computer professionals need to know the raw data contained in a computer's memory. A commonly used way of doing this is to print memory contents on a printer. This printout is called a *memory dump*. Memory dumps, which are in binary numbers, would have many pages of 0s and 1s. Working with these numbers would be very difficult and error prone for computer professionals. Hence, two number systems – octal and hexadecimal, are often used as shortcut notations for binary. These number systems and their relationship with binary number system are explained below.

Octal Number System

In octal number system, the base is 8. Hence, there are only eight symbols or digits: 0, 1, 2, 3, 4, 5, 6, and 7 (8 and 9 do not exist in this system). The largest single digit is 7 (one less than the base 8). Each position in an octal number represents a power of the base (8). Therefore, decimal equivalent of octal number 2057 (written as 2057_8) is:

$$(2 \times 8^3) + (0 \times 8^2) + (5 \times 8^1) + (7 \times 8^0) = 1024 + 0 + 40 + 7 = 1071$$

Hence, $2057_8 = 1071_{10}$

Observe that since there are only 8 digits in octal number system, 3 bits ($2^3 = 8$) are sufficient to represent any octal number in binary (see Figure 3.1).

Hexadecimal Number System

In hexadecimal number system, the base is 16. Hence, there are 16 symbols or digits. The first 10 digits are the same digits of decimal number system – 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. The remaining six digits are denoted by the symbols A, B, C, D, E, and F, representing decimal values 10, 11, 12, 13, 14, and 15, respectively. Hence, the largest single digit is F or 15 (one less than the base 16). Each position in hexadecimal number system represents a power of the base (16). Therefore, decimal equivalent of hexadecimal number 1AF (written as $1AF_{16}$) is:

$$(1 \times 16^2) + (A \times 16^1) + (F \times 16^0) = (1 \times 256) + (10 \times 16) + (15 \times 1) = 256 + 160 + 15 = 431$$

Hence, $1AF_{16} = 431_{10}$

Observe that since there are only 16 digits in hexadecimal number system, 4 bits ($2^4 = 16$) are sufficient to represent any hexadecimal number in binary.

CONVERTING FROM ONE NUMBER SYSTEM TO ANOTHER

Numbers expressed in decimal number system are much more meaningful to us, than are numbers expressed in any other number system. This is because we have been using decimal numbers in our day-to-day life, right from childhood. However, we can represent any number in one number system in any other number system. Because the input and final output values are to be in decimal, computer professionals are often required to convert numbers in other number systems to decimal and vice-versa. Many methods can be used to convert numbers from one base to another. A method of converting from another base to decimal, and a method of converting from decimal to another base are described below.

Converting from Another Base to Decimal

The following steps are used to convert a number in any other base to a base 10 (decimal) number:

Step 1: Determine the column (positional) value of each digit (this depends on the position of the digit and the base of the number system).

Step 2: Multiply the obtained column values (in Step 1) by the digits in the corresponding columns.

Step 3: Sum up the products calculated in Step 2. The total is the equivalent value in decimal.

Example 3.1.

$11001_2 = ?_{10}$

Solution:

Step 1: Determine column values

Column Number (from right)	Column Value
1	$2^0 = 1$
2	$2^1 = 2$
3	$2^2 = 4$
4	$2^3 = 8$
5	$2^4 = 16$

(Continued on next page ...)

Example 3.2.

$4706_8 = ?_{10}$

Solution:

Step 1: Determine column values

Column Number (from right)	Column Value
1	$8^0 = 1$
2	$8^1 = 8$
3	$8^2 = 64$
4	$8^3 = 512$

(Continued on next page ...)

(... Continued from previous page)

Step 2: Multiply the column values by the corresponding column digits

16	8	4	2	1
x1	x1	x0	x0	x1
16	8	0	0	1

Step 3: Sum up the products

$16 + 8 + 0 + 0 + 1 = 25$

Hence, $11001_2 = 25_{10}$

(... Continued from previous page)

Step 2: Multiply the column values by the corresponding column digits

512	64	8	1
x4	x7	x0	x6
2048	448	0	6

Step 3: Sum up the products

$2048 + 448 + 0 + 6 = 2502$

Hence, $4706_8 = 2502_{10}$

Example 3.3.

$1AC_{16} = ?_{10}$

Solution:

$$1AC_{16} = 1 \times 16^2 + A \times 16^1 + C \times 16^0$$
$$= 1 \times 256 + 10 \times 16 + 12 \times 1$$
$$= 256 + 160 + 12$$
$$= 428_{10}$$

Example 3.4.

$4052_7 = ?_{10}$

Solution:

$$4052_7 = 4 \times 7^3 + 0 \times 7^2 + 5 \times 7^1 + 2 \times 7^0$$
$$= 4 \times 343 + 0 \times 49 + 5 \times 7 + 2 \times 1$$
$$= 1372 + 0 + 35 + 2$$
$$= 1409_{10}$$

Example 3.5.

$4052_6 = ?_{10}$

Solution:

$$4052_6 = 4 \times 6^3 + 0 \times 6^2 + 5 \times 6^1 + 2 \times 6^0$$
$$= 4 \times 216 + 0 \times 36 + 5 \times 6 + 2 \times 1$$
$$= 864 + 0 + 30 + 2$$
$$= 896_{10}$$

Comparing this result with the result obtained in Example 3.4, we find that although the digits (4052) are the same for both the numbers, their decimal equivalents are different. This is because the number in Example 3.4 is represented in base 7 number system, whereas the number in Example 3.5 is represented in base 6 number system.

Example 3.6.

Solution:

$$11001_4 = 1 \times 4^4 + 1 \times 4^3 + 0 \times 4^2 + 0 \times 4^1 + 1 \times 4^0$$
$$= 1 \times 256 + 1 \times 64 + 0 \times 16 + 0 \times 4 + 1 \times 1$$
$$= 256 + 64 + 0 + 0 + 1$$
$$= 321_{10}$$

Compare the result with the result obtained in Example 3.1.

Example 3.7.

$1AC_{13} = ?_{10}$

Solution:

$$1AC_{13} = 1 \times 13^2 + A \times 13^1 + C \times 13^0$$
$$= 1 \times 169 + 10 \times 13 + 12 \times 1$$
$$= 311_{10}$$

Compare the result with the result obtained in Example 3.3.

Converting from Decimal to Another Base (Division-Remainder Technique)

The following steps are used to convert a base 10 (decimal) number to a number in another base:

Step 1: Divide the decimal number by the value of the new base.

Step 2: Record the remainder from Step 1 as the rightmost digit (least significant digit) of the new base number.

Step 3: Divide the quotient of the previous division by the new base.

Step 4: Record the remainder from Step 3 as the next digit (to the left) of the new base number.

Repeat Steps 3 and 4, recording remainders from right to left, until the quotient becomes zero in Step 3. Note that the last remainder, thus obtained, will be the most significant digit of the new base number.

Example 3.8.

$25_{10} = ?_2$

Solution:

Steps 1 and 2: 25/2 = 12 and remainder 1
Steps 3 and 4: 12/2 = 6 and remainder 0
Steps 3 and 4: 6/2 = 3 and remainder 0
Steps 3 and 4: 3/2 = 1 and remainder 1
Steps 3 and 4: 1/2 = 0 and remainder 1

As mentioned in Steps 2 and 4, the remainders are now arranged in the reverse order, making the first remainder the least significant digit (LSD) and the last remainder the most significant digit (MSD).

Hence, $25_{10} = 11001_2$

Compare the result with the result obtained in Example 3.1.

Example 3.9.

$42_{10} = ?_2$

Solution:

2	42	Remainders
	21	0
	10	1
	5	0
	2	1
	1	0
	0	1

Hence, $42_{10} = 101010_2$

Example 3.10.

$952_{10} = ?_8$

Solution:

8	952	Remainders
	119	0
	14	7
	1	6
	0	1

Hence, $952_{10} = 1670_8$

Example 3.11.

$428_{10} = ?_{16}$

Solution:

		Remainders in hexadecimal
16	428	
	26	12 = C
	1	10 = A
	0	1 = 1

Hence, $428_{10} = 1AC_{16}$

Compare the result with the result obtained in Example 3.3.

Example 3.12.

$100_{10} = ?_5$

Solution:

5	100	Remainders
	20	0
	4	0
	0	4

Hence, $100_{10} = 400_5$

Example 3.13.

$100_{10} = ?_4$

Solution:

		Remainders
4	100	
	25	0
	6	1
	1	2
	0	1

Hence, $100_{10} = 1210_4$

Compare the result with the result obtained in Example 3.12.

Example 3.14.

$1715_{10} = ?_{12}$

Solution:

		Remainders in base 12
12	1715	
	142	11 = B
	11	10 = A
	0	11 = B

Hence, $1715_{10} = BAB_{12}$

Converting from a Base Other Than 10 to Another Base Other Than 10

The following steps are used to convert a number in a base other than 10, to a number in another base other than 10:

Step 1: Convert the original number to a base 10 (decimal) number.

Step 2: Convert the decimal number obtained in Step 1 to the new base number.

Example 3.15.

$545_6 = ?_4$

Solution:

Step 1: Convert from base 6 to base 10

$$\begin{aligned} 545 &= 5 \times 6^2 + 4 \times 6^1 + 5 \times 6^0 \\ &= 5 \times 36 + 4 \times 6 + 5 \times 1 \\ &= 180 + 24 + 5 \\ &= 209_{10} \end{aligned}$$

(Continued on next page ...)

Example 3.16.

$101110_2 = ?_8$

Solution:

Step 1: Convert 101110_2 to base 10

$$\begin{aligned} 101110_2 &= 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + \\ &\quad 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\ &= 32 + 0 + 8 + 4 + 2 + 0 \\ &= 46_{10} \end{aligned}$$

(Continued on next page ...)

(... Continued from previous page)

Step 2: Convert 209_{10} to base 4

4	209	Remainders
	52	1
	13	0
	3	1
	0	3

$209_{10} = 3101_4$

Therefore, $545_6 = 209_{10} = 3101_4$

Hence, $545_6 = 3101_4$

(... Continued from previous page)

Step 2: Convert 46_{10} to base 8

8	46	Remainders
	5	6
	0	5

$46_{10} = 56_8$

Therefore, $101110_2 = 46_{10} = 56_8$

Hence, $101110_2 = 56_8$

Example 3.17.

$11010011_2 = ?_{16}$

Solution:

Step 1: Convert 11010011_2 to base 10

$$
\begin{aligned}
11010011_2 &= 1 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\
&= 1 \times 128 + 1 \times 64 + 0 \times 32 + 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 \\
&= 128 + 64 + 0 + 16 + 0 + 0 + 2 + 1 \\
&= 211_{10}
\end{aligned}
$$

Step 2: Convert 211_{10} to base 16

16	211	Remainders
	13	3 = 3 in Hexadecimal
	0	13 = D in Hexadecimal

Therefore, $11010011_2 = 211_{10} = D3_{16}$

Hence, $11010011_2 = D3_{16}$

Example 3.16 illustrates the method of converting a binary number to an octal number. Similarly, Example 3.17 shows how to convert a binary number to a hexadecimal number. However, these are lengthy procedures and we can use shortcut methods when we desire such conversions. These shortcut methods are described below.

Shortcut Method for Binary to Octal Conversion

The following steps are used in this method:

Step 1: Divide the binary digits into groups of three (starting from the right).

Step 2: Convert each group of three binary digits to one octal digit. Since there are only 8 digits (0 to 7) in octal number system (refer to Figure 3.1), 3 bits ($2^3 = 8$) are sufficient to represent any octal number in binary. Moreover, since decimal digits 0 to 7 are equal to the octal digits 0 to 7, we use binary to decimal conversion method in this step.

Example 3.18.

$101110_2 = ?_8$

Solution:

Step 1: Divide the binary digits into groups of 3, starting from the right (LSD).

101 110

Step 2: Convert each group into one digit of octal (use binary-to-decimal conversion method).

$$101_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5_8$$

$$110_2 = 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 4 + 2 + 0 = 6_8$$

Hence, $101110_2 = 56_8$

Compare the result with the result obtained in Example 3.16.

Example 3.19.

$1101010_2 = ?_8$

Solution:

$1101010_2 =$ 001 101 010 (Group of 3 digits from the right)

$= 152_8$ (Convert each group to an octal digit)

Hence, $1101010_2 = 152_8$

Shortcut Method for Octal to Binary Conversion

The following steps are used in this method:

Step 1: Convert each octal digit to a 3 digit binary number (the octal digits may be treated as decimal numbers for this conversion).

Step 2: Combine all the resulting binary groups (of 3 digits each) into a single binary number.

Example 3.20.

$562_8 = ?_2$

Solution:

Step 1: Convert each octal digit to 3 binary digits.

$5_8 = 101_2$
$6_8 = 110_2$
$2_8 = 010_2$

Step 2: Combine the binary groups.

$$562_8 = \underset{5}{\underline{101}} \quad \underset{6}{\underline{110}} \quad \underset{2}{\underline{010}}$$

Hence, $562_8 = 101110010_2$

Example 3.21.

$6751_8 = ?_2$

Solution:

$$6751_8 = \underset{6}{\underline{110}} \quad \underset{7}{\underline{111}} \quad \underset{5}{\underline{101}} \quad \underset{1}{\underline{001}}$$

$$= 110111101001_2$$

Hence, $6751_8 = 110111101001_2$

Shortcut Method for Binary to Hexadecimal Conversion

The following steps are used in this method:

Step 1: Divide the binary digits into groups of four (starting from the right).

Step 2: Convert each group of four binary digits to one hexadecimal digit. Remember that hexadecimal digits 0 to 9 are equal to decimal digits 0 to 9 and hexadecimal digits A to F are equal to decimal values 10 to 15. Hence, for this step, we use binary to decimal conversion procedure and represent decimal values 10 to 15 as hexadecimal A to F.

Example 3.22.

$11010011_2 = ?_{16}$

Solution:

Step 1: Divide the binary digits into groups of 4, starting from the right (LSD).

1101 0011

Step 2: Convert each group of 4 binary digits to 1 hexadecimal digit

$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
$= 8 + 4 + 0 + 1$
$= 13_{10}$
$= D_{16}$

$0011_2 = 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$
$= 0 + 0 + 2 + 1$
$= 3_{16}$

Hence, $11010011_2 = D3_{16}$

Compare the result with the result of Example 3.17.

Example 3.23.

$10110101100_2 = ?_{16}$

Solution:

$10110101100_2 =$ 0101 1010 1100 (Group 4 digits from the right)

$= 5AC$ (Convert each group to a hexadecimal digit)

Hence, $10110101100_2 = 5AC_{16}$

Shortcut Method for Hexadecimal to Binary Conversion

The following steps are used in this method:

Step 1: Convert decimal equivalent of each hexadecimal digit to 4 binary digits.

Step 2: Combine all resulting binary groups (of 4 digits each) into a single binary number.

Example 3.24.

$2AB_{16} = ?_2$

Solution:

Step 1: Convert decimal equivalent of each hexadecimal digit to 4 binary digits.

$2_{16} = 2_{10} = 0010_2$
$A_{16} = 10_{10} = 1010_2$
$B_{16} = 11_{10} = 1011_2$

Step 2: Combine the binary groups.

$$2AB_{16} = \underset{2}{\underline{0010}}\ \underset{A}{\underline{1010}}\ \underset{B}{\underline{1011}}$$

Hence, $2AB_{16} = 001010101011_2$

Example 3.25.

$ABC_{16} = ?_2$

Solution:

$$ABC_{16} = \underset{A}{\underline{1010}}\ \underset{B}{\underline{1011}}\ \underset{C}{\underline{1100}}$$

$$= 101010111100_2$$

Hence, $ABC_{16} = 101010111100_2$

Figure 3.2 summarizes the relationship among decimal, hexadecimal, binary, and octal number systems. Note that the maximum value for a single digit of octal (7) is equal to the maximum value of three digits of binary. The value range of one digit of octal duplicates the value range of three digits of binary. If we substitute octal digits for binary digits, the substitution is on a one-to-three basis. Hence, computers that print octal numbers instead of binary, while taking memory dump, save one-third of printing space and time.

Similarly, note that the maximum value of one digit in hexadecimal is equal to the maximum value of four digits in binary. Hence, the value range of one digit of hexadecimal is equivalent to the value range of four digits of binary. Therefore, hexadecimal shortcut notation is a one-to-four reduction in space and time required for memory dump.

Decimal	Hexadecimal	Binary	Octal
0	0	0	0
1	1	1	1
2	2	10	2
3	3	11	3
4	4	100	4
5	5	101	5
6	6	110	6
7	7	111	7
8	8	1000	10
9	9	1001	11
10	A	1010	12
11	B	1011	13
12	C	1100	14
13	D	1101	15
14	E	1110	16
15	F	1111	17

FRACTIONAL NUMBERS

In binary number system, fractional numbers are formed in the same way as in decimal number system. For example, in decimal number system,

$0.235 = (2 \times 10^{-1}) + (3 \times 10^{-2}) + (5 \times 10^{-3})$ and $68.53 = (6 \times 10^{1}) + (8 \times 10^{0}) + (5 \times 10^{-1}) + (3 \times 10^{-2})$

Similarly, in binary number system,

$0.101 = (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3})$ and $10.01 = (1 \times 2^{1}) + (0 \times 2^{0}) + (0 \times 2^{-1}) + (1 \times 2^{-2})$

Hence, binary point serves the same purpose as decimal point. Some positional values in binary number system are given below.

Binary Point

Position	4	3	2	1	0	.	-1	-2	-3	-4
Position Value	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}
Quantity Represented	16	8	4	2	1		½	$^1/_4$	$^1/_8$	$^1/_{16}$

In general, a number in a number system with base *b* would be written as:

$$a_n a_{n-1} \ldots a_0 . a_{-1} a_{-2} \ldots a_{-m}$$

and would be interpreted to mean

$$a_n \times b^n + a_{n-1} \times b^{n-1} + \ldots + a_0 \times b^0 + a_{-1} \times b^{-1} + a_{-2} \times b^{-2} + \ldots + a_{-m} \times b^{-m}$$

The symbols a_n, a_{n-1}, ..., a_{-m} used in the above representation should be one of the *b* symbols allowed in the number system.

Hence, as per the above mentioned general rule,

$46.32_8 = (4 \times 8^1) + (6 \times 8^0) + (3 \times 8^{-1}) + (2 \times 8^{-2})$ and $5A.3C_{16} = (5 \times 16^1) + (A \times 16^0) + (3 \times 16^{-1}) + (C \times 16^{-2})$

Example 3.26.

Find decimal equivalent of the binary number 110.101

Solution:

$$\begin{aligned} 110.101_2 &= 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} \\ &= 4 + 2 + 0 + 0.5 + 0 + 0.125 \\ &= 6 + 0.5 + 0.125 \\ &= 6.625_{10} \end{aligned}$$

Example 3.27.

Find decimal equivalent of the octal number 127.54

Solution:

$$\begin{aligned} 127.54_8 &= 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 5 \times 8^{-1} + 4 \times 8^{-2} \\ &= 64 + 16 + 7 + 5/8 + 4/64 \\ &= 87 + 0.625 + 0.0625 \\ &= 87.6875_{10} \end{aligned}$$

Example 3.28

Find decimal equivalent of the hexadecimal number 2B.C4

Solution:

$$
\begin{aligned}
2B.C4_{16} &= 2 \times 16^1 + B \times 16^0 + C \times 16^{-1} + 4 \times 16^{-2} \\
&= 32 + B + C/16 + 4/256 \\
&= 32 + 11 + 12/16 + 4/256 \\
&= 43 + 0.75 + 0.015625 \\
&= 43.765652_{10}
\end{aligned}
$$

Points to Remember

1. Number systems are of two types – non-positional and positional.
2. In a *non-positional number system*, each symbol represents the same value regardless of its position in the number and to find the value of a number, one has to count the number of symbols present in the number. It is very difficult to perform arithmetic with such a number system.
3. In a *positional number system*, there are only a few symbols called digits. These symbols represent different values depending on the position they occupy in a number. The value of each digit in such a number is determined by three considerations:
   - The digit itself,
   - The position of the digit in the number, and
   - The base of the number system (where base is defined as the total number of digits available in the number system).
4. In our day-to-day life, we use the *decimal number system*. In this system, base is equal to 10 because there are altogether ten symbols or digits (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9).
5. Some positional number systems that are used in computer design and by computer professionals are *binary* (having base 2), *octal* (having base 8), and *hexadecimal* (having base 16).
6. Figure 3.2 shows the relationship among decimal, hexadecimal, binary, and octal number systems.
7. Readers must know the techniques for following types of conversions to convert numbers from one base to another:
   - Converting from another base to decimal
   - Converting from decimal to another base
   - Converting from a base other than 10 to another base other than 10
   - Shortcut method for binary to octal conversion
   - Shortcut method for octal to binary conversion
   - Shortcut method for binary to hexadecimal conversion
   - Shortcut method for hexadecimal to binary conversion
8. *Fractional numbers* are formed in the same way as in decimal number system. In general, a number in a number system with base *b* would be written as:

   $a_n a_{n-1} \ldots a_0 . a_{-1} a_{-2} \ldots a_{-m}$

   and would be interpreted to mean

   $a_n \times b^n + a_{n-1} \times b^{n-1} + \ldots + a_0 \times b^0 + a_{-1} \times b^{-1} + a_{-2} \times b^{-2} + \ldots + a_{-m} \times b^{-m}$

   The symbols a_n, a_{n-1}, ..., a_{-m} used in the above representation should be one of the *b* symbols allowed in the number system.

Questions

1. Differentiate between positional and non-positional number systems? Write examples of both types of number systems.
2. What is base of a number system? Write examples to illustrate the role of base in positional number systems.

3. What is the value of base for decimal, hexadecimal, binary, and octal number systems?

4. Write an example for octal number system to show that the same digit may signify different values depending on the position it occupies in the number.

5. What will be the total number of different symbols or digits and the maximum value of a single digit for the following number systems:
   (a) Number system with base 5
   (b) Number system with base 20
   (c) Number system with base 9
   (d) Number system with base 12

6. What is a 'bit' in computer terminology? How many different patterns of bits are possible with
   (a) 6 bits (b) 7 bits (c) 8 bits

7. Explain the meaning of the term "memory dump"

8. Why do we use octal and/or hexadecimal number systems as shortcut notations?

9. Find decimal equivalent of following binary numbers:
   (a) 1101011
   (b) 11010
   (c) 10110011
   (d) 11011101
   (e) 1110101
   (f) 1000
   (g) 10110001100
   (h) 110001
   (i) 1010101100
   (j) 111

10. Find octal equivalent of the binary numbers of Question 9.

11. Find hexadecimal equivalent of the binary numbers of Question 9.

12. Convert the following numbers to decimal numbers:
   (a) 110110_2
   (b) 2573_6
   (c) $2A3B_{16}$
   (d) 1234_9

13. Convert the following decimal numbers to binary numbers:
   (a) 435_{10}
   (b) 1694_{10}
   (c) 32_{10}
   (d) 135_{10}

14. Convert the decimal numbers of Question 13 to octal numbers.

15. Convert the decimal numbers of Question 13 to hexadecimal numbers.

16. Carry out the following conversions:
   (a) $125_6 = ?_4$ (b) $24_9 = ?_3$ (c) $ABC_{16} = ?_8$

17. Convert the following numbers to their binary equivalent:
   (a) $2AC_{16}$
   (b) FAB_{16}
   (c) 2614_8
   (d) 562_8

18. Find decimal equivalent of following numbers:
   (a) 111.01_2
   (b) 1001.011_2
   (c) 247.65_8
   (d) $A2B.D4_{16}$