25.8. CONVERSIONS BETWEEN NUMBER SYSTEMS

A computer deals with numbers not in the way as we do. It uses binary number system to perform arithmetic other as well as operations dealing with numbers. In order to understand how a computer performs these operations, we require a proper understanding of conversion between various number systems. In the sections that follow, we learn how to convert a number in one system to that in another system.

25.8.1. Binary-to-Decimal Conversion

Binary numbers can be converted to decimal form quite easily. We have noted earlier that a binary number consists of bits each of which multiplies a different power of 2. For example, let us consider the binary number 11010.

Binary Number	:	1	1	0	1	0
Position Weights	:	2^4	2^3	2^2	2^1	2^0

The decimal equivalent of the binary number is the sum of all its bits multiplied by the weights of their respective positions. Thus

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

$$= 16 + 8 + 0 + 2 + 0$$

$$= 24 \leftarrow \text{Decimal equivalent}$$

From the above example we note that

(*i*) we have to take into account the weight of the position if there is 1 in a bit position and

(*ii*) we ignore the weight of the position if there is 0 in a bit position.

The procedure for converting a binary number into its decimal equivalent consists of the following 4-steps.

Step 1 : Write the binary number	– 1 0 1 1
Step 2 : Write the position weights beneath each bit.	– 2^3 2^2 2^1 2^0
	(or)
	8 4 2 1
Step 3 : Strike off weights corresponding to 0 in the binary number	– 8 4 2 1
Step 4 : Add the remaining weight values	– 8 + 2 + 1 = 11

∴ 11 is the decimal equivalent of the binary number 1011.

To convert a binary fraction into its decimal equivalent we multiply each digit in the fraction successively by $2^{-1}, 2^{-2}, 2^{-3}$....etc starting from the first digit after the binary point and then add the products. The value thus obtained gives the equivalent decimal fraction. The procedure is illustrated in the following example.

Example 25.14: Convert the binary number 10.101 to decimal sys-tem.

Step 1	–	1	0	.	1	0	1	← Binary number
Step 2	–	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}	← Position weights
Step 3	–	2	1	.	0.5	0.25	0.125	
Step 4	–	$2+0.5+0.125=2.625$						← Decimal equivalent

Thus 2.625 is the decimal equivalent of the binary number 10.101.

Usually, the number system corresponding to a particular number is indicated by writing the base of the number as a subscript to the number. Thus,

$$(1\ 0\ .\ 1\ 0\ 1)_2 = (2.625)_{10}$$

25.8.2. Decimal–to–Binary Conversion

Decimal-to-binary conversion is a relatively lengthy process. There is more than one method to perform the conversion.

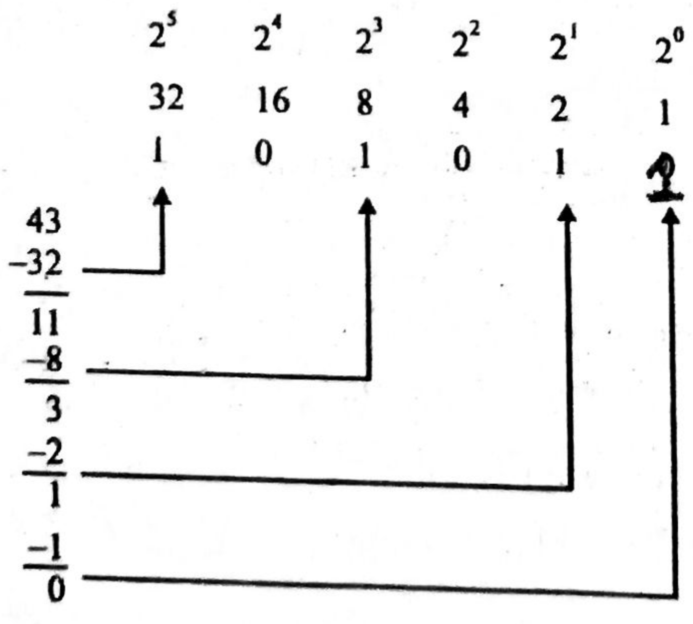
(a) Subtraction Method: This method is most suitable for whole numbers. The method consists of the following steps:

Step 1 : Write down the powers of 2.

Step 2 : Subtract each of the powers of 2 from the decimal number, starting from the power closest to the decimal number.

Step 3 : If the power can be subtracted, write a binary 1 in the bit position corresponding to that power of 2. If the power cannot be subtracted, write a binary 0.

Example 25.15: Conversion of $(43)_{10}$



Therefore $(43)_{10} = (1\,0\,1\,0\,1\,1)_2$

(b) Double-Dabble method or Two-part method : This method is more suitable for fractional numbers because we need not to know the negative powers of 2 to perform the conversion. In this method we convert the whole part of the number first and then the decimal part next. The whole

number part is converted by progressively dividing by 2. Each time there is a remainder of one, we write down a binary 1 and each time the remainder is zero, we write 0.

Next, we convert the fractional part of repeatedly multiplying by 2 until the fractional part exactly equals zero or until a sufficient number of bits have been obtained. The method thus consists of the following steps

Step 1 : Convert the whole part of the decimal number by repeatedly dividing by 2.

Step 2 : Each time, if there is no remainder, write down a 0. If there is a remainder write a 1.

Step 3 : Write remainders in reverse order to form the binary number

Step 4 : Multiply the fractional part repeatedly by 2.

Step 5 : Record each time a carry in the integer position .

Step 6 : Continue the process till the fractional part is zero or enough bits have been obtained.

Example 25.16: Convert $(43.3125)_{10}$ to binary system.

The whole part of the number is 43

	Division	Quotient	Remainder	
Step 1 :	$\frac{43}{2}=$	21	1	Least significant bit (LSB) ↑
	$\frac{21}{2}=$	10	1	
	$\frac{10}{2}=$	5	0	
	$\frac{5}{2}=$	2	1	
	$\frac{2}{2}=$	1	0	
	$\frac{1}{2}=$	0	1	Most significant bit (MSB)

$(43)_{10} = (1\,0\,1\,0\,1\,1)_2$

The fractional part of the decimal number is 0.3125

Multiplication	Whole number part	
0.3125 × 2 = 0.6250	0	(MSB) ↓
0.6250 × 2 = (1).2500	1	
.2500 × 2 = 0.5000	0	
0.5000 × 2 = (1).0000	1	(LSB)

∴ $(0.3125)_{10} = (0\,.\,0\,1\,0\,1)_2$

Thus, $(43.3125)_{10} = (1\,0\,1\,0\,1\,1\,.\,0101)_2$

25.8.3. Octal-to-Decimal Conversion

An octal-to-decimal conversion is done in the same manner as a binary-to-decimal conversion; that is by adding up the position weights. The procedure consists of the following four steps.

Step 1 : Write the octal number	—	1	0	3	3
Step 2 : Write the position weights beneath each digit.	—	8^3	8^2	8^1	8^0
	—	512	64	8	1
Step 3 : Srike off weights corresponding to 0 in the octal number.	—	512	64	8	1
Step 4 : Add the remaining weight values.	—	$(512 \times 1) + (3 \times 8) + (3 \times 1)$			
		$= 512 + 24 + 3 = 539$			

$\therefore (1\,0\,3\,3)_8 = (539)_{10}$

25.8.4. Decimal-to-Octal Conversion

To convert a decimal number to octal, a method similar to double-dabble is used. The technique is called octal-dabble. We repeatedly divide the octal number by 8 writing down the remainders after each division. By taking the remainders in reverse order, we obtain the octal number. In case of decimal fractions, they are multiplied repeatedly by 8 and the carry digits are written in the integer position.

The following example illustrates the method.

Example 25.17 Convert $(375.23)_{10}$ into octal form.

$\frac{375}{8} =$	46	7 ↑
$\frac{46}{8} =$	5	6
$\frac{5}{8} =$	0	5

$\therefore (375)_{10} = (567)_8$

$0.23 \times 8 = 1.84$	0.84	Carry 1 ↓
$0.84 \times 8 = 6.72$	0.72	Carry 6
$0.72 \times 8 = 5.76$	0.76	Carry 5
etc..	etc..	etc..

$\therefore (0.23)_{10} = (0.165)_8$

Thus, $(375.23)_{10} = (567.165)_8$

25.8.5. Octal-to-Binary Conversion

The base of octal number system is 8 and that of binary system is 2. Since $8 = 2^3$, there is a direct correlation between 3-bit groups in a binary number and the octal digits. It means each octal digit can be replaced by a 3-bit binary equivalent.

Example 25.18: Convert $(257)_8$ to binary

2 5 7

010 101 111

$\therefore (257)_8 = (010101111)_2$

Example 25.19: Convert $(34.56)_8$ to binary

3 4 5 6

011 100 101 110

$\therefore (34.56)_8 = (011\ 100\ .\ 101\ 110)_2$

The relationship of octal to binary digits is shown in Table 3.

Table 3. Octal and Binary Equivalents

Octal Digit	*Binary Bits*
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

25.8.6. Binary-to-Octal Conversion:

The binary-to-octal conversion is a reverse process of octal-to-binary conversion. The binary bits are partitioned into groups of 3-bits and each group is replaced by its octal equivalent.

Example 25.20: Convert $(1011.01101)_2$ to octal

The whole part of the binary number 1011 can be separated into two 3-bits groups by adding two zero bits to its left. Thus,

1011 = 001 011

Similarly, the fractional part can be separated by adding one zero bit to its right. Thus,

01101 = 011 010

Now

001 011 . 011 010

1 3 . 3 2

$\therefore (1011.01101)_2 = (13.32)_8$

25.8.7. Hexadecimal-to-Decimal Conversion

Hexadecimal number system has a base 16. The position of each digit has a weight equal to a power of 16. The decimal equivalent of the hexadecimal number is the sum of all digits multiplied by the weights of their respective positions. Consider the hexadecimal number $(473)_{16}$.

	4	7	3
Position Weights:	16^2	16^1	16^0
	256	16	1

$$= (4 \times 256) + (7 \times 16) + (3 \times 1)$$
$$= 1024 + 112 + 3$$
$$= 1139$$

$\therefore \quad (473)_{16} = (1139)_{10}$

Example 25.21: Convert $(A\,C\,5)_{16}$ to decimal.

$$\begin{matrix} A & C & 5 \\ 16^2 & 16^1 & 16^0 \end{matrix}$$
$$= (10 \times 256) + (12 \times 16) + (5 \times 1)$$
$$= 2560 + 192 + 5$$
$$= 2757$$

$\therefore \quad (A\,C\,5)_{16} = (2757)_{10}$

25.8.8. Decimal-to-Hexadecimal Conversion

We employ hexa-dabble method to convert a decimal number to hexadecimal number. We repeatedly divide the whole part of number by 16 and write the remainders in reverse order to form the whole part of hexadecimal number.

Remainders greater than 9 are expressed as equivalent hexadecimal digits A to F.

The fractional part is multiplied by 16 and the carry is written in the integer position. These carries are written in normal order to obtain the fractional part.

Example 25.22: Convert $(379.54)_{10}$ to hexadecimal

$$379 \div 16 = 23 \qquad 11 \rightarrow B \uparrow$$
$$23 \div 16 = 1 \qquad 7$$
$$1 \div 16 = 0 \qquad 1$$

$\therefore \quad (379)_{10} = (17B)_{16}$

$$0.54 \times 16 = 8.64 \qquad \text{Carry } 8 \downarrow$$
$$0.64 \times 16 = 10.24 \qquad \text{carry } 10 \rightarrow A$$

$\therefore \quad (0.54)_{10} = (8A)_{16}$

Thus, $(379.54)_{10} = (17\,B.8A)_{16}$

25.8.9. Hexadecimal-to-Binary Conversion

The hexadecimal number system has a base $16 = 2^4$. We write the 4-bit binary equivalent of each hexadecimal digit.

Example 25.23: Convert $(7\,A\,B)_{16}$ to binary.

7	A	B
0111	1010	1011

$\therefore (7\,A\,B)_{16} = (0111\;1010\;1011)_2$

Note that 4-bit binary equivalent must be used to replace each hexadecimal digit.

25.8.10. Binary-to-Hexadecimal Conversion

To convert a binary number to hexadecimal equivalent we partition the binary number into 4-bit groups and each group of 4 is replaced by its hexadecimal equivalent.

Example 25.24: Convert $(1\ 0\ 1\ 1\ 1\ 0\ 1)_2$ to hexadecimal.

Step 1 : 0101 1101

Step 2 : 5 D

$\therefore (1011101)_2 = (5D)_{16}$