(iv) – 48 – 23 = (– 48) + (– 23)

$$\Rightarrow \begin{array}{r} -48 \\ +(-23) \\ \hline -71 \end{array} \Rightarrow \begin{array}{rr} & 1\ 1\ 0\ 1\ 0\ 0\ 0\ 0 \\ (+) & 1\ 1\ 1\ 0\ 1\ 0\ 0\ 1 \\ \hline 1 & 1\ 0\ 1\ 1\ 1\ 0\ 0\ 1 \end{array} \Rightarrow -71$$

↑
Carry be ignored

From the above example, we observe the following:

(a) If the two operands are of the opposite sign, the result is to be obtained by the rule of subtraction using 2's complement (Sec. 2.6.1)
(b) If the two operands are of the same sign, the sign bit of the result (MSB) is to be compared with the sign bit of the operands. In case the sign bits are same, the result is correct and is in 2's complement form. If the sign bits are not same there is a problem of *overflow*, i.e. the result can not be accommodated using eight bits and the result is to be interpreted suitably. The result in this case will consist of nine bits, i.e. carry and eight bits, and the carry bit will give the sign of the number.

2.7 OCTAL NUMBER SYSTEM

The number system with base (or radix) eight is known as the *octal number system*. In this system, eight symbols, 0, 1, 2, 3, 4, 5, 6, and 7 are used to represent numbers. Similar to decimal and binary number systems, it is also a positional system and has, in general, two parts: integer and fractional, set apart by a radix (octal) point (.). Any number can be expressed in the form of Eq. (2.1) with $b = 8$ and $0 \le (d_i \text{ or } d_{-f}) \le 7$.The weights assigned to the various positions are given in Table 2.1. For example, $(6327.4051)_8$ is an octal number.

2.7.1 Octal-to-Decimal Conversion

Any octal number can be converted into its equivalent decimal number using the weights assigned to each octal digit position as given in Table 2.1.

Example 2.20

Convert $(6327.4051)_8$ into its equivalent decimal number.

Solution

Using the weights given in Table 2.1, we obtain

$$\begin{aligned}(6327.4051)_8 &= 6 \times 8^3 + 3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} \\ &\quad + 0 \times 8^{-2} + 5 \times 8^{-3} + 1 \times 8^{-4} \\ &= 3072 + 192 + 16 + 7 + \frac{4}{8} + 0 + \frac{5}{512} + \frac{1}{4096} \\ &= (3287.5100098)_{10}\end{aligned}$$

Thus, $(6327.4051)_8 = (3287.5100098)_{10}$

2.7.2 Decimal-to-Octal Conversion

The conversion from decimal to octal (base-10 to base-8) is similar to the conversion procedure for base-10 to base-2 conversion. The only difference is that number 8 is used in place of 2 for division in the case of integers and for multiplication in the case of fractional numbers.

Example 2.21

(a) Convert $(247)_{10}$ into octal
(b) Convert $(0.6875)_{10}$ into octal
(c) Convert $(3287.5100098)_{10}$ into octal

Solution

(a)

	Quotient	Remainder
$\frac{247}{8}$	30	7
$\frac{30}{8}$	3	6
$\frac{3}{8}$	0	3

3 6 7

Thus, $(247)_{10} = (367)_8$

(b)

$$0.6875 \times 8 = 5.5000 \rightarrow 5$$
$$0.5000 \times 8 = 4.0000 \rightarrow 4$$

Thus, $(0.6875)_{10} = (0.54)_8$

(c) *Integer part:*

	Quotient	Remainder
3287/8	410	7
410/8	51	2
51/8	6	3
6/8	0	6

6 3 2 7

Thus, $(3287)_{10} = (6327)_8$

Fractional part:

$$0.5100098 \times 8 = 4.0800784 \rightarrow 4$$
$$0.0800784 \times 8 = 0.6406272 \rightarrow 0$$
$$0.6406272 \times 8 = 5.1250176 \rightarrow 5$$
$$0.1250176 \times 8 = 1.0001408 \rightarrow 1$$

Thus $(0.5100098)_{10} \approx (0.4051)_8$
Therefore, $(3287.5100098)_{10} = (6327.4051)_8$

From the above examples we observe that the conversion for fractional numbers may not be exact. In general, an approximate equivalent can be determined by terminating the process of multiplication by eight at the desired point.

2.7.3 Octal-to-Binary Conversion

Octal numbers can be converted into equivalent binary numbers by replacing each octal digit by its 3-bit equivalent binary. Table 2.6 gives octal numbers and their binary equivalents for decimal numbers 0 to 15.

Table 2.6 ***Binary and Decimal Equivalents of Octal Numbers***

Octal	Decimal	Binary
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111
10	8	001000
11	9	001001
12	10	001010
13	11	001011
14	12	001100
15	13	001101
16	14	001110
17	15	001111

Example 2.22

Convert $(736)_8$ into an equivalent binary number.

Solution

From Table 2.6, we observe the binary equivalents of 7, 3 and 6 as 111, 011, and 110, respectively. Therefore, $(736)_8 = (111\ 011\ 110)_2$.

2.7.4 Binary-to-Octal Conversion

Binary numbers can be converted into equivalent octal numbers by making groups of three bits starting from LSB and moving towards MSB for integer part of the number and then replacing each group of three bits by its octal representation. For fractional part, the groupings of three bits are made starting from the binary point.

Example 2.23

Convert $(1001110)_2$ to its octal equivalent.

Solution

$$\begin{aligned}(1001110)_2 &= (\underline{00}1\ 001\ 110)_2\\ &= (1\ 1\ 6)_8\\ &= (116)_8\end{aligned}$$

Example 2.24

Convert $(0.10100110)_2$ to its equivalent octal number.

Solution

$$\begin{aligned}(0.10100110)_2 &= (0.101\ 001\ 10\underline{0})_2\\ &= (0.5\ 1\ 4)_8\\ &= (0.514)_8\end{aligned}$$

Example 2.25

Convert the following binary numbers to octal numbers

(a) 11001110001.000101111001
(b) 1011011110.11001010011
(c) 111110001.10011001101

Solution

(a) 011 001 110 001.000 101 111 001 = $(3161.0571)_8$
(b) 001 011 011 110.110 010 100 110 = $(1336.6246)_8$
(c) 111 110 001.100 110 011 010 = $(761.4632)_8$

From the above examples we observe that in forming the 3-bit groupings, 0's may be required to complete the first (most significant digit) group in the integer part and the last (least significant digit) group in the fractional part.

2.7.5 Octal Arithmetic

Octal arithmetic rules are similar to the decimal or binary arithmetic. Normally, we are not interested in performing octal arithmetic operations using octal representation of numbers. This number system is normally used to enter long strings of binary data into a digital system like a microcomputer. This makes the task of entering binary data in a microcomputer easier. Arithmetic operations can be performed by converting the octal numbers to binary numbers and then using the rules of binary arithmetic.

Example 2.26

Add $(23)_8$ and $(67)_8$.

Solution

$$\begin{array}{rcl} 23 & = & 010011 \\ (+)67 & = & 110111 \\ \hline (112)_8 & = & 1001\,010 \end{array}$$

Example 2.27

Subtract (a) $(37)_8$ from $(53)_8$
(b) $(75)_8$ from $(26)_8$

Solution

Using 8-bit representation,

(a)
$$\begin{array}{rcll} (53)_8 & = & 00101011 & \\ -(37)_8 & = & (+)\ 11100001 & \text{Two's complement of } (37)_8 \\ \hline (14)_8 & = & 100001100 & \end{array}$$

Discard carry ⟶ ↑

(b)
$$\begin{array}{rcll} (26)_8 & = & 00010110 & \\ -(75)_8 & = & (+)\ 11000011 & \text{Two's complement of } (75)_8 \\ \hline -(47)_8 & & 11011001 & \text{Two's complement of result} \end{array}$$

Two's complement of 11011001 = 00 100 111 = $(47)_8$

Multiplication and division can also be performed using the binary representation of octal numbers and then making use of multiplication and division rules of binary numbers.

2.7.6 Applications of Octal Number System

In digital systems, binary numbers are required to be entered and certain results or status signals are required to be displayed. It is highly inconvenient to handle long strings of binary numbers. It may cause errors also. Therefore, octal numbers are used for entering the binary data and displaying certain informations. Therefore, the knowledge of octal number system is very important for the efficient use of microprocessors and other digital circuits. For example, the binary number 011111110 can easily be remembered as 376 and can be