CHAPTER 19

Number Systems

19.1 THE DECIMAL SYSTEM

In everyday life we use a system based on decimal digits (0, 1, 2, 3, 4, 5, 6, 7, 8, 9) to represent numbers and refer to the system as the decimal system. Consider what the number 83 means. It means eight tens plus three:

$$83 = (8 \times 10) + 3$$

The number 4728 means four thousands, seven hundreds, two tens, plus eight:

$$4728 = (4 \times 1000) + (7 \times 100) + (2 \times 10) + 8$$

The decimal system is said to have a **base**, or **radix**, of 10. This means that each digit in the number is multiplied by 10 raised to a power corresponding to that digit's position:

$$\begin{aligned} 83 &= (8 \times 10^1) + (3 \times 10^0) \\ 4728 &= (4 \times 10^3) + (7 \times 10^2) + (2 \times 10^1) + (8 \times 10^0) \end{aligned}$$

The same principle holds for decimal fractions but negative powers of 10 are used. Thus, the decimal fraction 0.256 stands for 2 tenths plus 5 hundredths plus 6 thousandths:

$$0.256 = (2 \times 10^{-1}) + (5 \times 10^{-2}) + (6 \times 10^{-3})$$

A number with both an integer and fractional part has digits raised to both positive and negative powers of 10:

$$472.256 = (4 \times 10^2) + (7 \times 10^1) + (2 \times 10^0) + (2 \times 10^{-1}) + (5 \times 10^{-2}) + (6 \times 10^{-3})$$

In general, for the decimal representation of $X = \{\ldots d_2 d_1 d_0 . d_{-1} d_{-2} d_{-3} \ldots\}$, the value of X is

$$X = \sum_i (d_i \times 10^i) \quad \textbf{(19.1)}$$

19.2 THE BINARY SYSTEM

In the decimal system, 10 different digits are used to represent numbers with a base of 10. In the binary system, we have only two digits, 1 and 0. Thus, numbers in the binary system are represented to the base 2.

To avoid confusion, we will sometimes put a subscript on a number to indicate its base. For example, 83_{10} and 4728_{10} are numbers represented in decimal notation or, more briefly, decimal numbers. The digits 1 and 0 in binary notation have the same meaning as in decimal notation:

$$\begin{aligned} 0_2 &= 0_{10} \\ 1_2 &= 1_{10} \end{aligned}$$

To represent larger numbers, as with decimal notation, each digit in a binary number has a value depending on its position:

$$10_2 = (1 \times 2^1) + (0 \times 2^0) = 2_{10}$$
$$11_2 = (1 \times 2^1) + (1 \times 2^0) = 3_{10}$$
$$100_2 = (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 4_{10}$$

and so on. Again, fractional values are represented with negative powers of the radix:

$$1001.101 = 2^3 + 2^0 + 2^{-1} + 2^{-3} = 9.625_{10}$$

In general, for the binary representation of $Y = \{\ldots b_2 b_1 b_0 . b_{-1} b_{-2} b_{-3} \ldots\}$, the value of Y is

$$Y = \sum_i (b_i \times 2^i) \qquad \textbf{(19.2)}$$

19.3 CONVERTING BETWEEN BINARY AND DECIMAL

It is a simple matter to convert a number from binary notation to decimal notation. In fact, we showed several examples in the previous subsection. All that is required is to multiply each binary digit by the appropriate power of 2 and add the results.

To convert from decimal to binary, the integer and fractional parts are handled separately.

Integers

For the integer part, recall that in binary notation, an integer represented by

$$b_{m-1} b_{m-2} \ldots b_2 b_1 b_0 \qquad b_i = 0 \text{ or } 1$$

has the value

$$(b_{m-1} \times 2^{m-1}) + (b_{m-2} \times 2^{m-2}) + \cdots + (b_1 \times 2^1) + b_0$$

Suppose it is required to convert a decimal integer N into binary form. If we divide N by 2, in the decimal system, and obtain a quotient N_1 and a remainder R_0, we may write

$$N = 2 \times N_1 + R_0 \qquad R_0 = 0 \text{ or } 1$$

Next, we divide the quotient N_1 by 2. Assume that the new quotient is N_2 and the new remainder R_1. Then

$$N_1 = 2 \times N_2 + R_1 \qquad R_1 = 0 \text{ or } 1$$

so that

$$N = 2(2N_2 + R_1) + R_0 = (N_2 \times 2^2) + (R_1 \times 2^1) + R_0$$

If next

$$N_2 = 2N_3 + R_2$$

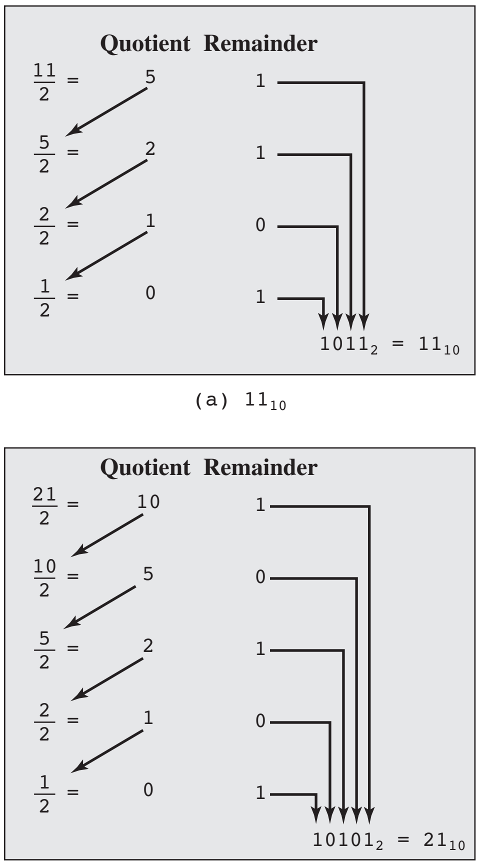
we have

$$N = (N_3 \times 2^3) + (R_2 \times 2^2) + (R_1 \times 2^1) + R_0$$

Because $N > N_1 > N_2 \ldots$, continuing this sequence will eventually produce a quotient $N_{m-1} = 1$ (except for the decimal integers 0 and 1, whose binary equivalents are 0 and 1, respectively) and a remainder R_{m-2}, which is 0 or 1. Then

$$N = (1 \times 2^{m-1}) + (R_{m-2} \times 2^{m-2}) + \cdots + (R_2 \times 2^2) + (R_1 \times 2^1) + R_0$$

which is the binary form of N. Hence, we convert from base 10 to base 2 by repeated divisions by 2. The remainders and the final quotient, 1, give us, in order of increasing significance, the binary digits of N. Figure 19.1 shows two examples.



(b) 21_{10}

Figure 19.1 Examples of Converting from Decimal Notation to Binary Notation for Integers

Fractions

For the fractional part, recall that in binary notation, a number with a value between 0 and 1 is represented by

$$0.b_{-1}b_{-2}b_{-3}\ldots \qquad b_i = 0 \text{ or } 1$$

and has the value

$$(b_{-1} \times 2^{-1}) + (b_{-2} \times 2^{-2}) + (b_{-3} \times 2^{-3})\cdots$$

This can be rewritten as

$$2^{-1} \times (b_{-1} + 2^{-1} \times (b_{-2} + 2^{-1} \times (b_{-3} + \cdots$$

This expression suggests a technique for conversion. Suppose we want to convert the number F $(0 < F < 1)$ from decimal to binary notation. We know that F can be expressed in the form

$$F = 2^{-1} \times (b_{-1} + 2^{-1} \times (b_{-2} + 2^{-1} \times (b_{-3} + \cdots$$

If we multiply F by 2, we obtain,

$$2 \times F = b_{-1} + 2^{-1} \times (b_{-2} + 2^{-1} \times (b_{-3} + \cdots$$

From this equation, we see that the integer part of $(2 \times F)$, which must be either 0 or 1 because $0 < F < 1$, is simply b_{-1}. So we can say $(2 \times F) = b_{-1} + F_1$, where $0 < F_1 < 1$ and where

$$F_1 = 2^{-1} \times (b_{-2} + 2^{-1} \times (b_{-3} + 2^{-1} \times (b_{-4} + \cdots$$

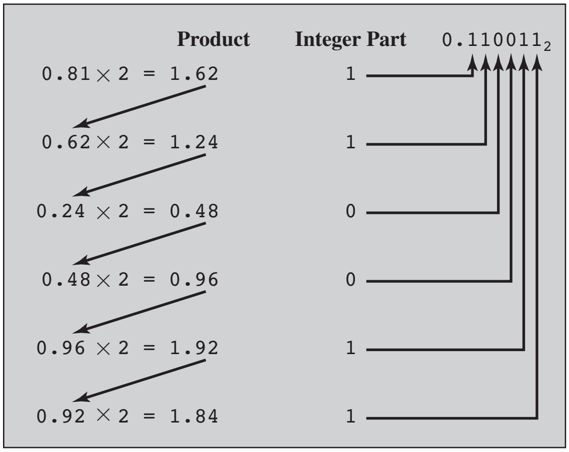
To find b_{-2}, we repeat the process. Therefore, the conversion algorithm involves repeated multiplication by 2. At each step, the fractional part of the number from the previous step is multiplied by 2. The digit to the left of the decimal point in the product will be 0 or 1 and contributes to the binary representation, starting with the most significant digit. The fractional part of the product is used as the multiplicand in the next step. Figure 19.2 shows two examples.

This process is not necessarily exact; that is, a decimal fraction with a finite number of digits may require a binary fraction with an infinite number of digits. In such cases, the conversion algorithm is usually halted after a prespecified number of steps, depending on the desired accuracy.

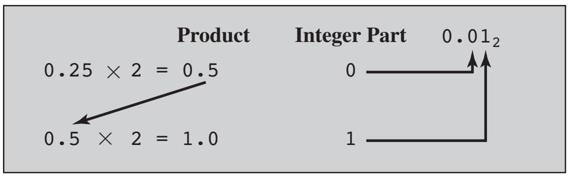
19.4 HEXADECIMAL NOTATION

Because of the inherent binary nature of digital computer components, all forms of data within computers are represented by various binary codes. However, no matter how convenient the binary system is for computers, it is exceedingly cumbersome for human beings. Consequently, most computer professionals who must spend time working with the actual raw data in the computer prefer a more compact notation.

What notation to use? One possibility is the decimal notation. This is certainly more compact than binary notation, but it is awkward because of the tediousness of converting between base 2 and base 10.



(a) $0.81_{10} = 0.110011_2$ (approximately)



(b) $0.25_{10} = 0.01_2$ (exactly)

Figure 19.2 Examples of Converting from Decimal Notation to Binary Notation for Fractions

Instead, a notation known as hexadecimal has been adopted. Binary digits are grouped into sets of four. Each possible combination of four binary digits is given a symbol, as follows:

0000 = 0	0100 = 4	1000 = 8	1100 = C
0001 = 1	0101 = 5	1001 = 9	1101 = D
0010 = 2	0110 = 6	1010 = A	1110 = E
0011 = 3	0111 = 7	1011 = B	1111 = F

Because 16 symbols are used, the notation is called **hexadecimal**, and the 16 symbols are the **hexadecimal digits**.

A sequence of hexadecimal digits can be thought of as representing an integer in base 16 (Table 19.1). Thus,

$$\begin{aligned} 2C_{16} &= (2_{16} \times 16^1) + (C_{16} \times 16^0) \\ &= (2_{10} \times 16^1) + (12_{10} \times 16^0) = 44 \end{aligned}$$

Hexadecimal notation is used not only for representing integers. It is also used as a concise notation for representing any sequence of binary digits, whether they

Table 19.1 Decimal, Binary, and Hexadecimal

Decimal (base 10)	Binary (base 2)	Hexadecimal (base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	0001 0000	10
17	0001 0001	11
18	0001 0010	12
31	0001 0000	1F
100	0110 0100	64
255	1111 0000	FF
256	0001 0000 0000	100

represent text, numbers, or some other type of data. The reasons for using hexadecimal notation are

1. It is more compact than binary notation.
2. In most computers, binary data occupy some multiple of 4 bits, and hence some multiple of a single hexadecimal digit.
3. It is extremely easy to convert between binary and hexadecimal.

As an example of the last point, consider the binary string 110111100001. This is equivalent to

$$\begin{array}{ccccc} 1101 & 1110 & 0001 & = & \mathrm{DE1}_{16} \\ \mathrm{D} & \mathrm{E} & 1 & & \end{array}$$

This process is performed so naturally that an experienced programmer can mentally convert visual representations of binary data to their hexadecimal equivalent without written effort

19.5 KEY TERMS AND PROBLEMS

Key Terms

base binary decimal	fraction hexadecimal	integer radix

Problems

19.1 Convert the following binary numbers to their decimal equivalents:
a. 001100 **b.** 000011 **c.** 011100 **d.** 111100 **e.** 101010

19.2 Convert the following binary numbers to their decimal equivalents:
a. 11100.011 **b.** 110011.10011 **c.** 1010101010.1

19.3 Convert the following decimal numbers to their binary equivalents:
a. 64 **b.** 100 **c.** 111 **d.** 145 **e.** 255

19.4 Convert the following decimal numbers to their binary equivalents:
a. 34.75 **b.** 25.25 **c.** 27.1875

19.5 Express the following octal numbers in hexadecimal notation:
a. 12 **b.** 5655 **c.** 2550276 **d.** 76545336 **e.** 3726755

19.6 Convert the following hexadecimal numbers to their decimal equivalents:
a. C **b.** 9F **c.** D52 **d.** 67E **e.** ABCD

19.7 Convert the following hexadecimal numbers to their decimal equivalents:
a. F.4 **b.** D3.E **c.** 1111.1 **d.** 888.8 **e.** EBA.C

19.8 Convert the following decimal numbers to their hexadecimal equivalents:
a. 16 **b.** 80 **c.** 2560 **d.** 3000 **e.** 62,500

19.9 Convert the following decimal numbers to their hexadecimal equivalents:
a. 204.125 **b.** 255.875 **c.** 631.25 **d.** 10000.00390625

19.10 Convert the following hexadecimal numbers to their binary equivalents:
a. E **b.** 1C **c.** A64 **d.** 1F.C **e.** 239.4

19.11 Convert the following binary numbers to their hexadecimal equivalents:
a. 1001.1111 **b.** 110101.011001 **c.** 10100111.111011

19.12 Prove that every real number with a terminating binary representation (finite number of digits to the right of the binary point) also has a terminating decimal representation (finite number of digits to the right of the decimal point).

19.13 Equations (19.1) and (19.2) define the representation of numbers in base 10 and base 2, respectively. In general, for the representation in base g of $X = \{\ldots x_2 x_1 x_0 \cdot x_{-1} x_{-2} x_{-3} \ldots\}$, the value of X is

$$X = \sum_i (x_i \times g^i)$$

Thus, 65 in base 7 is $(6 \times 7^1) + (5 \times 7^0) = 47$.

Count from one to 20_{10} in the following bases:
a. 8 **b.** 6 **c.** 5 **d.** 3

19.14 Perform the indicated base conversions:
a. 54_8 to base 5
b. 312_4 to base 7
c. 520_6 to base 7
d. 12212_3 to base 9

19.15 What generalizations can you draw about converting a number from one base to a power of that base, e.g., from base 3 to base 9 (3^2) or from base 2 to base 4 (2^2) or base 8 (2^3)?