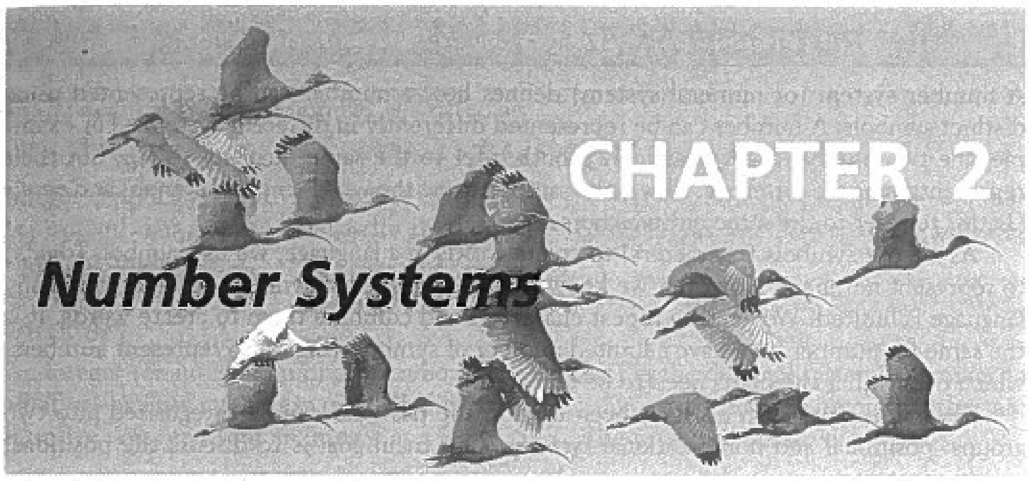
cording

mputer

mputer

ginated.

chapter.
ve their



CHAPTER 2 Number Systems

This chapter is a prelude to Chapters 3 and 4. In Chapter 3 we will show how data is stored inside the computer. In Chapter 4 we will show how logic and arithmetic operations are performed on data. This chapter is a preparation for understanding the contents of Chapter 3 and 4. Readers who know about number systems can skip this chapter and move on to Chapter 3 without loss of continuity. Note that the number systems discussed in this chapter are 'paper and pencil representations': we show how these numbers are stored in a computer in Chapter 3.

Objectives

After studying this chapter, the student should be able to:

- ❑ Understand the concept of number systems.
- ❑ Distinguish between nonpositional and positional number systems.
- ❑ Describe the decimal system (base 10).
- ❑ Describe the binary system (base 2).
- ❑ Describe the hexadecimal system (base 16).
- ❑ Describe the octal system (base 8).
- ❑ Convert a number in binary, octal, or hexadecimal to a number in the decimal system.
- ❑ Convert a number in the decimal system to a number in binary, octal, and hexadecimal.
- ❑ Convert a number in binary to octal and vice versa.
- ❑ Convert a number in binary to hexadecimal and *vice versa*.
- ❑ Find the number of digits needed in each system to represent a particular value.

2.1 INTRODUCTION

A **number system** (or numeral system) defines how a number can be represented using distinct symbols. A number can be represented differently in different systems. For example, the two numbers $(2A)_{16}$ and $(52)_8$ both refer to the same quantity, $(42)_{10}$, but their representations are different. This is the same as using the words *cheval* (French) and *equus* (Latin) to refer to the same entity, a horse.

As we use symbols (characters) to create words in a language, we use symbols (digits) to represent numbers. However, we know that the number of symbols (characters) in any language is limited. We need to repeat characters and combine them to create words. It is the same for numbers: we have a limited number of symbols (digits) to represent numbers, which means that the digits need to be repeated.

Several number systems have been used in the past and can be categorized into two groups: positional and non-positional systems. Our main goal is to discuss the positional number systems, but we also give examples of non-positional systems.

2.2 POSITIONAL NUMBER SYSTEMS

In a **positional number system**, the position a symbol occupies in the number determines the value it represents. In this system, a number represented as:

$$\pm\,(S_{k-1} \ldots S_2 S_1 S_0 . S_{-1} S_{-2} \ldots S_{-l})_b$$

has the value of:

$$n = \pm \quad S_{k-1} \times b^{k-1} + \ldots + S_1 \times b^1 + S_0 \times b^0 \qquad + \quad S_{-1} \times b^{-1} + S_{-2} \times b^{-2} + \ldots + S_{-l} \times b^{-l}$$

in which S is the set of symbols, b is the **base** (or **radix**), which is equal to the total number of the symbols in the set S, and S_k and S_l are symbols in the whole a and fraction parts of the number. Note that we have used an expression that can be extended from the right or from the left. In other words, the power of b can be 0 to $k - 1$ in one direction and -1 to $-l$ in the other direction. The terms with non-negative powers of b are related to the integral part of the number, while the terms with negative power of b are related to the fractional part of the number. The ± sign shows that the number can be either positive or negative. We will study several positional number systems in this chapter.

2.2.1 The decimal system (base 10)

The first positional number system we discuss in this chapter is the **decimal system**. The word **decimal** is derived from the Latin root **decem** (ten). In this system the base b = 10 and we use ten symbols to represent a number. The set of symbols is S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. As we know, the symbols in this system are often referred to as **decimal digits** or just digits. In this chapter, we use ± to show that a number can be positive or negative, but remember that these signs are not stored in computers—computers handle the sign differently, as we discuss in Chapter 3.

Computers store positive and negative numbers differently.

In the decimal system, a number is written as:

$$\pm\ (S_{k-1} \ldots S_2\, S_1\, S_0 .\ S_{-1}\, S_{-2} \ldots S_{-l})_{10}$$

but for simplicity, we often drop the parentheses, the base, and the plus sign (if the number is positive). For example, we write the number $+(552.23)_{10}$ as 552.23—the base and plus signs are implicit.

Integers

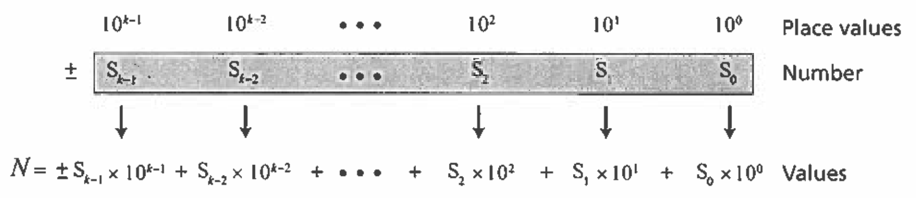
An **integer** (an integral number with no fractional part) in the decimal system is familiar to all of us—we use integers in our daily life. In fact, we have used them so much that they are intuitive. We represent an integer as $\pm S_{k-1} \ldots S_1\, S_0$. The value is calculated as:

$$N = \pm \quad S_{k-1} \times 10^{k-1} + S_{k-2} \times 10^{k-2} + \ldots + S_2 \times 10^2 + S_1 \times 10^1 + S_0 \times 10^0$$

in which S_k is a digit, $b = 10$ is the base, and k is the number of digits.

Another way to show an integer in a number system is to use **place values**, which are powers of 10 (10^0, 10^1, … 10^{k-1}) for decimal numbers. Figure 2.1 shows an integer in the decimal system using place values.

Figure 2.1 *Place value for an integer in decimal system*


10^{k-1} 10^{k-2} • • • 10^2 10^1 10^0 Place values
$\pm$ S_{k-1} S_{k-2} • • • S_2 S_1 S_0 Number
$N = \pm S_{k-1} \times 10^{k-1} + S_{k-2} \times 10^{k-2} + \bullet\bullet\bullet + S_2 \times 10^2 + S_1 \times 10^1 + S_0 \times 10^0$ Values

Example 2.1

The following shows the place values for the integer +224 in the decimal system.

			10^2		10^1		10^0	Place values
			2		2		4	Number
N	=	+	2×10^2	+	2×10^1	+	4×10^0	Values

Note that the digit 2 in position 1 has the value 20, but the same digit in position 2 has the value 200. Also note that we normally drop the plus sign, but it is implicit.

Example 2.2

The following shows the place values for the decimal number –7508. We have used 1, 10, 100, and 1000 instead of powers of 10.

		1000		100		10		1	Place values
		7		5		0		8	Number
$N =$	$-$	$(7 \times 1000$	$+$	5×100	$+$	0×10	$+$	$8 \times 1)$	Place values

Maximum value

Sometimes we need to know the maximum value of a decimal integer that can be represented by k digits. The answer is $N_{max} = 10^k - 1$. For example, if $k = 5$, then the maximum value is $N_{max} = 10^5 - 1 = 99999$.

Reals

A **real** (a number with a fractional part) in the decimal system is also familiar. For example, we use this system to show dollars and cents ($23.40). We can represent a real as $\pm\ S^{k-1} \ldots S_1 S_0 \bullet S_{-1} \ldots\ S_{-l}$. The value is calculated as:

	Integral part		Fractional part
$R = \pm$	$S_{k-1} \times 10^{k-1} + \ldots + S_1 \times 10^1 + S_0 \times 10^0$	$+$	$S_{-1} \times 10^{-1} + \ldots + S_{-l} \times 10^{-l}$

in which S_i is a digit, b = 10 is the base, k is the number of digits in the integral part, and l is the number of digits in the fractional part. The decimal point we use in our representation separates the fractional part from the integral part.

Example 2.3

The following shows the place values for the real number +24.13.

	10^1		10^0		10^{-1}		10^{-2}	Place values
	2		4	•	1		3	Number
$R = +$	2×10	$+$	4×1	$+$	1×0.1	$+$	3×0.01	Values

2.2.2 The binary system (base 2)

The second positional number system we discuss in this chapter is the **binary system**. The word **binary** is derived from the Latin root **bini** (or two by two). In this system the base b = 2 and we use only two symbols, S = {0, 1}. The symbols in this system are often referred to as **binary digits** or **bits** (binary digit). As we will see in Chapter 3, data and programs are stored in the computer using binary patterns, a string of bits. This is because the computer is made of electronic switches that can have only two states, on and off. The bit 1 represents one of these two states and the bit 0 the other.

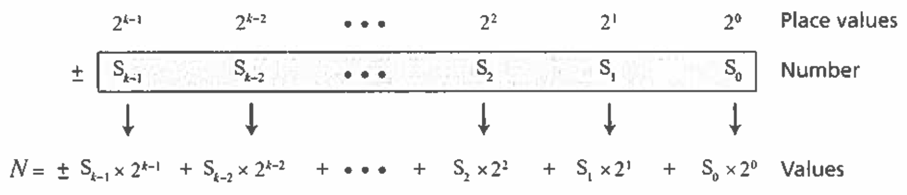
Integers

We can represent an integer as $\pm\ (S_{k-1} \ldots S_1\ S_0)_2$. The value is calculated as:

$$N = \pm\ S_{k-1} \times 2^{k-1} + S_{k-2} \times 2^{k-2} + \ldots + S_2 \times 2^2 + S_1 \times 2^1 + S_0 \times 2^0$$

in which S_i is a digit, $b = 2$ is the base, and k is the number of bits. Another way to show a binary number is to use place values ($2^0, 2^1, \ldots 2^{k-1}$). **Figure 2.2 shows a number in the binary number system using place values:**

Figure 2.2 *Place values in an integer in the binary system*



2^{k-1} 2^{k-2} • • • 2^2 2^1 2^0 Place values
$\pm$ S_{k-1} S_{k-2} • • • S_2 S_1 S_0 Number
$N = \pm\ S_{k-1} \times 2^{k-1} + S_{k-2} \times 2^{k-2} + \bullet\bullet\bullet + S_2 \times 2^2 + S_1 \times 2^1 + S_0 \times 2^0$ Values

Example 2.4

The following shows that the number $(11001)_2$ in binary is the same as 25 in decimal. The subscript 2 shows that the base is 2.

	2^4		2^3		2^2		2^1		2^0	Place values
	1		1		0		0		1	Number
$N =$	1×2^4	+	1×2^3	+	0×2^2	+	0×2^1	+	1×2^0	Decimal

Note that the equivalent decimal number is $N = 16 + 8 + 0 + 0 + 1 = 25$.

Maximum value

The maximum value of a binary integer with k digits is $N_{max} = 2^k - 1$. For example, if $k = 5$, then the maximum value is $N_{max} = 2^5 - 1 = 31$.

Reals

A real—a number with an optional fractional part—in the binary system can be made of k bits on the left and l bits on the right, $\pm\ (S^{k-1} \ldots S_1\ S_0 \bullet S_{-1} \ldots S_{-l})_2$. The value can be calculated as:

$$\begin{array}{ccccc} & \text{Integral part} & \bullet & \text{Fractional part} \\ R = \pm & S_{k-1} \times 2^{k-1} + \ldots + S_1 \times 2^1 + S_0 \times 2^0 & + & S_{-1} \times 2^{-1} + \ldots + S_{-l} \times 2^{-l} \end{array}$$

in which S_i is a bit, $b = 2$ is the base, k is the number of bits to the left, and l is the number of bits to the right of the decimal point. Note that k starts from 0, but l starts from -1. The highest power is $k - 1$ and the lowest power is $-l$.

Example 2.5

The following shows that the number $(101.11)_2$ in binary is equal to the number 5.75 in decimal.

	2^2		2^1		2^0		2^{-1}		2^{-2}	Place values
	1		0		1	•	1		1	Number
$R =$	1×2^2	+	0×2^1	+	1×2^0	+	1×2^{-1}	+	1×2^{-2}	Values

Note that the value in the decimal system is $R = 4 + 0 + 1 + 0.5 + 0.25 = 5.75$.

2.2.3 The hexadecimal system (base 16)

Although the binary system is used to store data in computers, it is not convenient for representation of numbers outside the computer, as a number in binary notation is much longer than the corresponding number in decimal notation. However, the decimal system does not show what is stored in computer as binary directly—there is no obvious relationship between the number of bits in binary and the number of decimal digits. Conversion from one to the other is not fast, as we will see shortly.

To overcome this problem, two positional systems were devised: hexadecimal and octal. We first discuss the **hexadecimal system**, which is more common. The word **hexadecimal** is derived from the Greek root **hex** (six) and the Latin root **decem** (ten). To be consistent with decimal and binary, it should really have been called **sexadecimal**, from the Latin roots **sex** and **decem**. In this system the base b = 16 and we use 16 symbols to represent a number. The set of symbols is S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}. Note that the symbols A, B, C, D, E, F (uppercase or lowercase) are equivalent to 10, 11, 12, 13, 14, and 15 respectively. The symbols in this system are often referred to as **hexadecimal digits**.

Integers

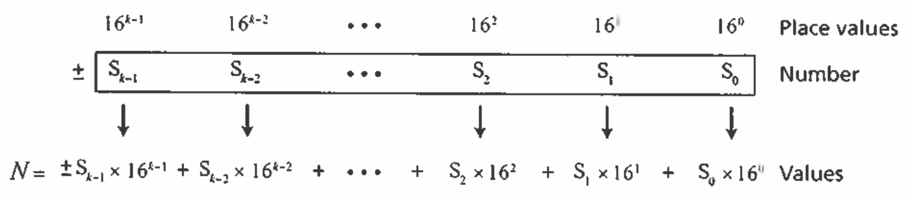
We can represent an integer as $\pm\ S_{k-1} \ldots S_1\ S_0$. The value is calculated as:

$$N = \pm\ S_{k-1} \times 16^{k-1} + S_{k-2} \times 16^{k-2} + \ldots + S_2 \times 16^2 + S_1 \times 16^1 + S_0 \times 16^0$$

in which S_i is a digit, b = 16 is the base, and k is the number of digits.

Another way to show a hexadecimal number is to use place values $(16^0, 16^1, \ldots 16^{k-1})$. Figure 2.3 shows a number in the hexadecimal number system using place values.

Figure 2.3 *Place values in an integer in the hexadecimal system*



16^{k-1} 16^{k-2} ••• 16^2 16^1 16^0 Place values
$\pm$ S_{k-1} S_{k-2} ••• S_2 S_1 S_0 Number
$N = \pm S_{k-1} \times 16^{k-1} + S_{k-2} \times 16^{k-2} + \cdots + S_2 \times 16^2 + S_1 \times 16^1 + S_0 \times 16^0$ Values

Example 2.6

The following shows that the number $(2AE)_{16}$ in hexadecimal is equivalent to 686 in decimal.

	16^2		16^1		16^0	Place values
	2		A		E	Number
$N =$	2×16^2	+	10×16^1	+	14×16^0	Values

Maximum value

The maximum value of a hexadecimal integer with k digits is $N_{max} = 16^k - 1$. For example, if $k = 5$, then the maximum value is $N_{max} = 16^5 - 1 = 1\,048\,575$.

Reals

Although a real number can be also represented in the hexadecimal system, it is not very common.

2.2.4 The octal system (base 8)

The second system that was devised to show the equivalent of the binary system outside the computer is the **octal system**. The word **octal** is derived from the Latin root **octo** (eight). In this system the base b = 8 and we use eight symbols to represent a number. The set of symbols is S = {0, 1, 2, 3, 4, 5, 6, 7}. The symbols in this system are often referred to as octal digits.

Integers

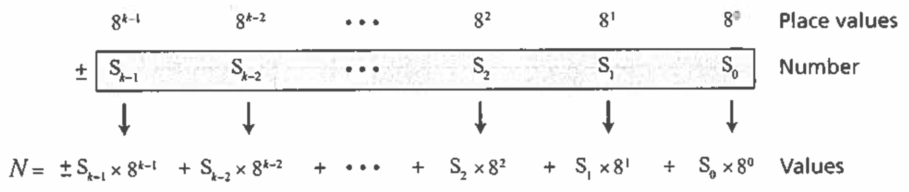
We can represent an integer as $\pm\ S_{k-1} \ldots S_1\ S_0$. The value is calculated as:

$$N = \pm\, S_{k-1} \times 8^{k-1} + S_{k-2} \times 8^{k-2} + \ldots + S_2 \times 8^2 + S_1 \times 8^1 + S_0 \times 8^0$$

in which S_i is a digit, b = 8 is the base, and k is the number of digits.

Another way to show an octal number is to use place values ($8^0, 8^1, \ldots 8^{k-1}$). Figure 2.4 shows a number in the octal number system using place values.

Figure 2.4 *Place values in an integer in the octal system*


8^{k-1} 8^{k-2} ... 8^2 8^1 8^0
Place values
$\pm$ S_{k-1} S_{k-2} ... S_2 S_1 S_0
Number
$N = \pm S_{k-1} \times 8^{k-1} + S_{k-2} \times 8^{k-2} + \cdots + S_2 \times 8^2 + S_1 \times 8^1 + S_0 \times 8^0$
Values

Example 2.7

The following shows that the number $(1256)_8$ in octal is the same as 686 in decimal.

	8^3		8^2		8^1		8^0	Place values
	1		2		5		6	Number
N =	1×8^3	+	2×8^2	+	5×8^1	+	6×8^0	Values

Note that the decimal number is $N = 512 + 128 + 40 + 6 = 686$.

Maximum Value

The maximum value of an octal integer with k digits is $N_{max} = 8^k - 1$. For example, if $k = 5$, then the maximum value is $N_{max} = 8^5 - 1 = 32767$.

Reals

Although a real number can be also represented in the octal system, it is not very common.

2.2.5 Summary of the four positional systems

Table 2.1 shows a summary of the four positional number systems discussed in this chapter.

Table 2.1 Summary of the four positional number systems

System	*Base*	*Symbols*	*Examples*
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	2345.56
Binary	2	0, 1	$(1001.11)_2$
Octal	8	0, 1, 2, 3, 4, 5, 6, 7	$(156.23)_8$
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$(A2C.A1)_{16}$

Table 2.2 shows how the number 15 is represented with two digits in decimal, four digits in binary, two digits in octal, and only one digit in hexadecimal. The hexadecimal representation is definitely the shortest.

Table 2.2 Comparison of numbers in the four systems

Decimal	*Binary*	*Octal*	*Hexadecimal*
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5

Table 2.2 (Continued)

Decimal	*Binary*	*Octal*	*Hexadecimal*
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

$(\ )_8 \Rightarrow (\ )_{10}$
octal to decimal

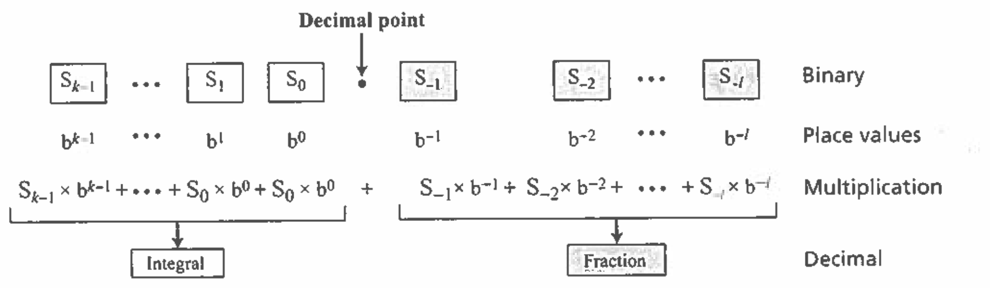
2.2.6 Conversion

We need to know how to convert a number in one system to the equivalent number in another system. Since the decimal system is more familiar than the other systems, we first show how to covert from any base to decimal. Then we show how to convert from decimal to any base. Finally, we show how we can easily convert from binary to hexadecimal or octal and *vice versa*.

Any base to decimal conversion

This type of conversion is easy and fast. We multiply each digit with its place value in the source system and add the results to get the number in the decimal system. Figure 2.5 shows the idea.

Figure 2.5 *Converting other bases to decimal*


Decimal point
S_{k-1} ••• S_1 S_0 • S_{-1} S_{-2} ••• S_{-l}
Binary
b^{k-1} ••• b^1 b^0 b^{-1} b^{-2} ••• b^{-l}
Place values
$S_{k-1} \times b^{k-1} + \cdots + S_0 \times b^0 + S_0 \times b^0$ + $S_{-1} \times b^{-1} + S_{-2} \times b^{-2} + \cdots + S_{-l} \times b^{-l}$
Multiplication
Integral
Fraction
Decimal

Example 2.8

The following shows how to convert the binary number $(110.11)_2$ to decimal: $(110.11)_2 = 6.75$.

Binary	1		1		0	•	1		1
Place values	2^2		2^1		2^0		2^{-1}		2^{-2}
Partial results	4	+	2	+	0	+	0.5	+	0.25
Decimal: 6.75									

Example 2.9

The following shows how to convert the hexadecimal number $(1A.23)_{16}$ to decimal.

Hexadecimal	1		A	•	2		3
Place values	16^1		16^0		16^{-1}		16^{-2}
Partial result	16	+	10	+	0.125	+	0.012
Decimal: 26.137							

Note that the result in the decimal notation is not exact, because $3 \times 16^{-2} = 0.01171875$. We have rounded this value to three digits (0.012). In other words, $(1A.23)_{16} \approx 26.137$. When we convert a number in decimal to hexadecimal, we need to specify how many digits we allow to the right of the decimal point.

Example 2.10

The following shows how to convert $(23.17)_8$ to decimal.

Octal	2		3	•	1		7
Place values	8^1		8^0		8^{-1}		8^{-2}
Partial result	16	+	3	+	0.125	+	0.109
Decimal: 19.234							

This means that $(23.17)_8 \approx 19.234$ in decimal. Again, we have rounded up $7 \times 8^{-2} = 0.109375$.

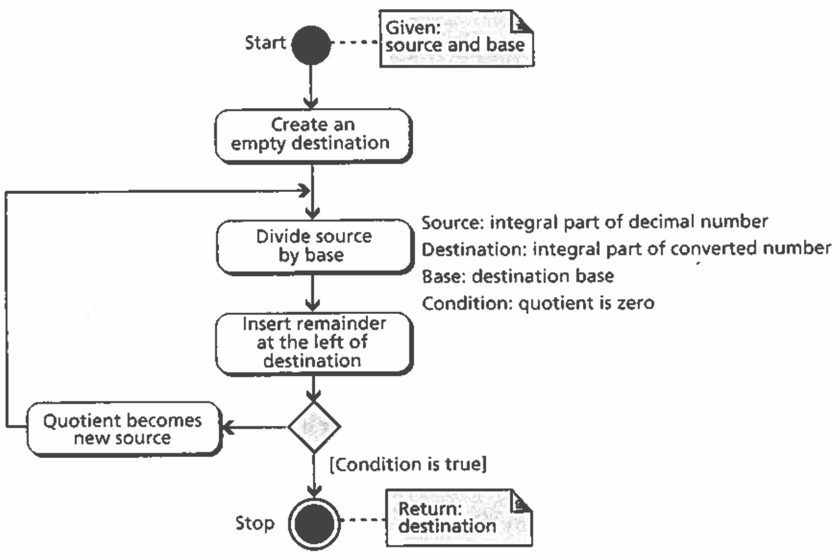
Decimal to any base

We can convert a decimal number to its equivalent in any base. We need two procedures, one for the integral part and one for the fractional part.

Converting the integral part

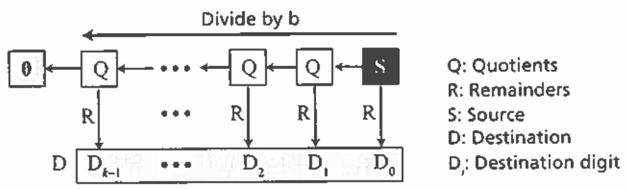
The integral part can be converted using repetitive division. Figure 2.6 shows the UML diagram for the process. We use UML diagrams through the book. For those readers not familiar with UML diagrams, please see Appendix B.

Figure 2.6 *Algorithm to convert the integral part*



We call the integral part of the decimal number the *source* and the integral part of the converted number the *destination*. We first create an empty destination. We then repetitively divide the source to get the quotient and the remainder. The remainder is inserted to the left of the destination. The quotient becomes a new source. Figure 2.7 shows the how the destination is made with each repetition.

Figure 2.7 *Converting the integral part*



We use Figure 1.7 below to illustrate the process manually with some examples.

Example 2.11

The following shows how to convert 35 in decimal to binary. We start with the number in decimal, we move to the left while continuously finding the quotients and the remainder of division by 2. The result is 35 = $(100011)_2$.

0	←	1	←	2	←	4	←	8	←	17	←	35	Decimal
		↓		↓		↓		↓		↓		↓	
		1		0		0		0		1		1	Binary

Example 2.12

The following shows how to convert 126 in decimal to its equivalent in the octal system. We move to the right while continuously finding the quotients and the remainder of division by 8. The result is 126 = $(176)_8$.

$$\begin{tikzcd}
0 & 1 \arrow[l] \arrow[d] & 15 \arrow[l] \arrow[d] & 126 \arrow[l] \arrow[d] & \text{Decimal} \\
 & 1 & 7 & 6 & \text{Octal}
\end{tikzcd}$$

Example 2.13

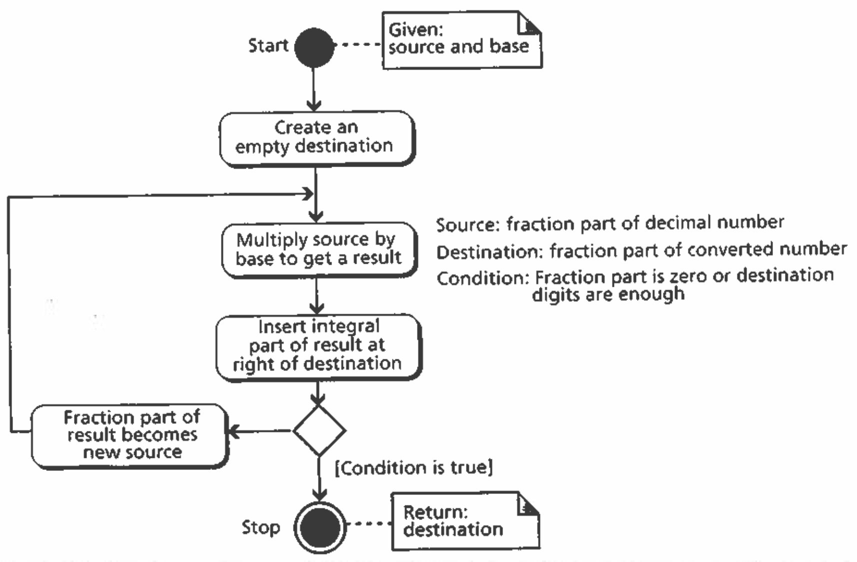
The following shows how we convert 126 in decimal to its equivalent in the hexadecimal system. We move to the right while continuously finding the quotients and the remainder of division by 16. The result is 126 = $(7E)_{16}$

$$\begin{tikzcd}
0 & 7 \arrow[l] \arrow[d] & 126 \arrow[l] \arrow[d] & \text{Decimal} \\
 & 7 & E & \text{Hexadecimal}
\end{tikzcd}$$

Converting the fractional part

The fractional part can be converted using repetitive multiplication. We call the fractional part of the decimal number the *source* and the fractional part of the converted number the *destination*. We first create an empty destination. We then repetitively multiply the source to get the result. The integral part of the result is inserted to the right of the destination, while the fractional part becomes the new source. Figure 2.8 shows the UML diagram for the process. Figure 2.9 shows how the destination is made in each repetition. We use the figure to illustrate the process manually with some examples.

Figure 2.8 *Algorithm to convert the fractional part*

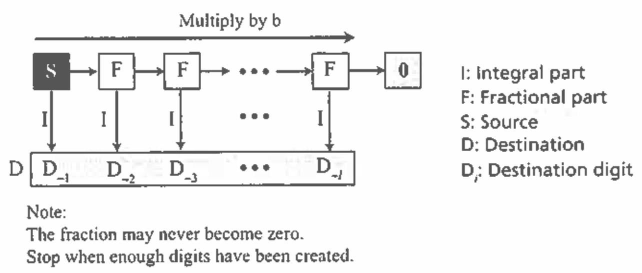

Start
Given: source and base
Create an empty destination
Multiply source by base to get a result
Source: fraction part of decimal number
Destination: fraction part of converted number
Condition: Fraction part is zero or destination digits are enough
Insert integral part of result at right of destination
Fraction part of result becomes new source
[Condition is true]
Stop
Return: destination

system.
division

decimal
nainder

ıctional
ber the
source
ination,
ram for
use the

Figure 2.9 *Converting the fractional part*



Example 2.14

Convert the decimal number 0.625 to binary.

Solution

Since the number 0.625 has no integral part, the example shows how the fractional part is calculated. The base here is 2. Write the decimal number at the left corner. Multiply the number continuously by 2 and record the integral and fractional part of the result. The fractional part moves to the right, and the integral part is recorded under each operation. Stop when the fractional part is 0 or there are enough bits. The result is $(0.101)_2$.

Decimal		0.625	→	0.25	→	0.50	→	0.00
		↓		↓		↓		
Binary	•	1		0		1		

Example 2.15

The following shows how to convert 0.634 to octal using a maximum of four digits. The result is $0.634 = (0.5044)_8$. Note that we multiple by 8 (base octal).

Decimal		0.634	→	0.072	→	0.576	→	0.608	→	0.864
		↓		↓		↓		↓		
Octal	•	5		0		4		4		

Example 2.16

The following shows how to convert 178.6 in decimal to hexadecimal using only one digit to the right of the decimal point. The result is $178.6 = (B2.9)_{16}$ Note that we divide or multiple by 16 (base hexadecimal).

Decimal	0	←	11	←	178	•	0.6	→	0.6
			↓		↓		↓		
Hexadecimal			B		2	•	9		

Example 2.17

An alternative method for converting a small decimal integer (usually less than 256) to binary is to break the number as the sum of numbers that are equivalent to the binary place values shows:

Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal equivalent	128	64	32	16	8	4	2	1

Using this table, we can convert 165 to binary $(10100101)_2$ as shown below:

Decimal 165 =	128	+	0	+	32	+	0	+	0	+	4	+	0	+	1
Binary	1		0		1		0		0		1		0		1

Example 2.18

A similar method can be used to convert a decimal fraction to binary when the denominator is a power of two:

Place values	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Decimal equivalent	½	¼	⅛	$1/16$	$1/32$	$1/64$	$1/128$

Using this table, we convert $27/64$ to binary $(0.011011)_2$ as shown below:

Decimal = $27/64$	$16/64$	+	$8/64$	+	$2/64$	+	$1/64$
	¼	+	⅛	+	$1/32$	+	$1/64$

Aligning these fraction according to decimal equivalent values. Note that since ½ and $1/16$ are missing, we replace them with 0s.

Decimal $27/64$ =	0	+	¼	+	⅛	+	0	+	$1/32$	+	$1/64$
Binary	0		1		1		0		1		1

Number of digits

We often need to know the number of digits before converting a number from decimal to other bases. In a positional number system with base b, we can always find the number of digits of an integer using the relation $k = \lceil \log_b N \rceil$, in which $\lceil x \rceil$ means the smallest integer greater than or equal to x (it is also called the **ceiling** of x), and N is the decimal value of the integer. For example, we can find the required number of bits in the decimal number 234 in all four systems as follows:

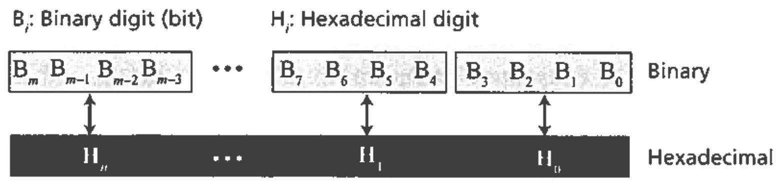
a. In decimal: $k_d = \lceil \log_{10} 234 \rceil = \lceil 2.37 \rceil = 3$, which is obvious.
b. In binary: $k_b = \lceil \log_2 234 \rceil = \lceil 7.8 \rceil = 8$. This is true because 234 = $(11101010)_2$
c. In octal: $k_o = \lceil \log_8 234 \rceil = \lceil 2.62 \rceil = 3$. This is true because 234 = $(352)_8$
d. In hexadecimal $k_h = \lceil \log_{16} 234 \rceil = \lceil 1.96 \rceil = 2$. This is true because 234 = $(EA)_{16}$

See Appendix G for information on how to calculate $\log_b N$ if your calculator does not include logs to any base.

Binary-hexadecimal conversion

We can easy change a number from binary to hexadecimal and *vice verse*. The reason is that there is a relationship between the two bases: four bits in binary is one digit in hexadecimal. Figure 2.10 shows how this conversion can be done.

Figure 2.10 *Binary to hexadecimal and hexadecimal to binary*



Example 2.19

Show the hexadecimal equivalent of the binary number $(10011100010)_2$.

Solution

We first arrange the binary number in 4-bit patterns: 100 1110 0010. Note that the leftmost pattern can have one to four bits. We then use the equivalent of each pattern shown in Table 2.2 on page 24–25 to change the number to hexadecimal: $(4E2)_{16}$.

Example 2.20

What is the binary equivalent of $(24C)_{16}$?

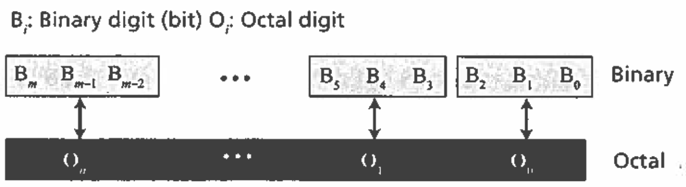
Solution

Each hexadecimal digit is converted to 4-bit patterns: $2 \rightarrow 0010$, $4 \rightarrow 0100$, and $C \rightarrow 1100$. The result is $(001001001100)_2$.

Binary-octal conversion

We can easy convert a number from binary to octal and vice verse. The reason is that there is an interesting relationship between the two bases: three bits is one octal digit. Figure 2.11 shows how this conversion can be done.

Figure 2.11 Binary to hexadecimal conversion



Example 2.21

Show the octal equivalent of the binary number $(101110010)_2$.

Solution

Each group of three bits is translated into one octal digit. The equivalent of each 3-bit group is shown in Table 2.2 on page 24–25. The result is $(562)_8$.

Example 2.22

What is the binary equivalent of for $(24)_8$?

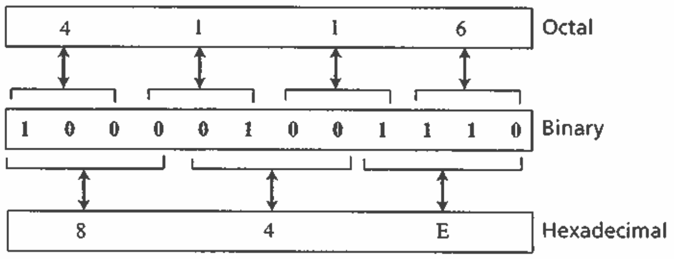
Solution

Write each octal digit as its equivalent bit pattern to get $(010100)_2$.

Octal-hexadecimal conversion

It is not difficult to convert a number in octal to hexadecimal or *vice versa*. We can use the binary system as the intermediate system. Figure 2.12 shows an example.

Figure 2.12 *Octal to hexadecimal and hexadecimal to octal conversion*



The following illustrates the process:

- To convert from octal to hexadecimal, we first convert the number in the octal system to binary. We then rearrange the bits in groups of four bits to find the hexadecimal equivalent.
- To convert from hexadecimal to octal, we first convert the number in the hexadecimal system to binary. We then rearrange the bits in groups of three to find the octal equivalent.

Number of digits

In conversion from one base to another, we often need to know the minimum number of digits we need in the destination system if we know the maximum number of digits in the source system. For example, if we know that we use at most six decimal digits in the source system, we want to know the minimum number of binary digits we need in the destination system. In general, assume that we are using k digits in base b_1 system. The maximum number we can represent in the source system is $b_1^k - 1$. The maximum number we can have in the destination system is $b_2^x - 1$. Therefore, $b_2^x - 1 \geq b_1^k - 1$. This means $b_2^x \geq b_1^k$, which means:

$$x \geq k \times (\log b_1 / \log b_2) \qquad \text{or} \qquad x = \lceil k \times (\log b_1 / \log b_2) \rceil$$

Example 2.23

Find the minimum number of binary digits required to store decimal integers with a maximum of six digits.

Solution

$k = 6$, $b_1 = 10$, and $b_2 = 2$. Then $x = \lceil k \times (\log b_1 / \log b_2) \rceil = \lceil 6 \times (1 / 0.30103) \rceil = 20$. The largest six-digit decimal number is 999 999 and the largest 20-bit binary number is 1 048 575. Note that the largest number that can be represented by a 19-bit number is 524287, which is smaller than 999 999. We definitely need 20 bits.

2.3 NON-POSITIONAL NUMBER SYSTEMS

Although nonpositional number systems are not used in computers, we give a short review here for comparison with positional number systems. A **nonpositional number system** still uses a limited number of symbols in which each symbol has a value. However, the position a symbol occupies in the number normally bears no relation to its value—the value of each symbol is fixed. To find the value of a number, we add the value of all symbols present in the representation. In this system, a number is represented as:

$$S_{k-1} \ldots S_2 S_1 S_0 \bullet S_{-1} S_{-2} \ldots S_{-l}$$

and has the value of:

$$n = \pm \overbrace{S_{k-1} + \ldots + S_1 + S_0}^{\text{Integral part}} + \overbrace{S_{-1} + S_{-2} + \ldots + S_{-l}}^{\text{Fractional part}}$$

There are some exception to the addition rule we just mentioned, as shown in Example 2.24.

Example 2.24

The **Roman number system** is a good example of a nonpositional number system. This system was invented by the Romans and was used until the sixteenth-century in Europe. Roman numerals are still used in sports events, clock dials, and other applications. This number system has a set of symbols S = {I, V, X, L, C, D, M}. The values of each symbol is shown in Table 2.3

Table 2.3 Values of symbols in the Roman number system

Symbol	*I*	*V*	*X*	*L*	*C*	*D*	*M*
Value	1	5	10	50	100	500	1000

To find the value of a number, we need to add the value of symbols subject to specific rules:

1. When a symbol with a smaller value is placed after a symbol having an equal or larger value, the values are added.
2. When a symbol with a smaller value is placed before a symbol having a larger value, the smaller value is subtracted from the larger one.
3. A symbol S_1 cannot come before another symbol S_2 if $S_1 \leq 10 \times S_2$. For example, I or V cannot come before C.
4. For large numbers a bar is placed above any of the six symbols (all symbols except I) to express multiplication by 1000. For example, $\overline{V}$ = 5000 and $\overline{M}$ = 1000000.
5. Although Romans used the word **nulla** (nothing) to convey the concept of zero, the Roman numerals lack a zero digit in their system.

The following shows some Roman numbers and their values.

III	→	1 + 1 + 1	=	3
IV	→	5 − 1	=	4
VIII	→	5 + 1 + 1 + 1	=	8
XVIII	→	10 + 5 + 1 + 1 + 1	=	18
XIX	→	10 + (10 −1)	=	19
LXXII	→	50 + 10 + 10 + 1 + 1	=	72
CI	→	100 + 1	=	101
MMVII	→	1000 + 1000 + 5 + 1 + 1	=	2007
MDC	→	1000 + 500 + 100	=	1600

2.4 END-CHAPTER MATERIALS

2.4.1 Recommended reading

For more details about the subjects discussed in this chapter, the following books are recommended:

- ❑ Stalling, W. *Computer Organization and Architecture*, Upper Saddle River, NJ: Prentice Hall, 2000
- ❑ Mano, M. *Computer System Architecture*, Upper Saddle River, NJ: Prentice Hall, 1993
- ❑ Null, L. and Lobur, J. *Computer Organization and Architecture*, Sudbury, MA: Jones and Bartlett, 2003
- ❑ Brown, S. and Vranesic, Z. *Fundamentals of Digital Logic with Verilog Design*, New York: McGraw-Hill, 2003

c rules:
r larger
lue, the
ıle, I or
pt I) to
ːro, the

oks are
I: Pren-
ll, 1993
ι: Jones
n, New

2.4.2 Key terms

base 18	binary digit 20
binary system 20	bit 20
decimal digit 18	decimal system 18
hexadecimal digit 22	hexadecimal system 22
integer 19	nonpositional number system 33
number system 18	octal digit 23
octal system 23	place value 19
positional number system 18	radix 18
real 20	Roman number system 33

2.4.3 Summary

- A number system (or numeral system) is a system that uses distinct symbols to represent a number. In a positional number system, the position a symbol occupies in the number determines the value it represents. Each position has a place value associated with it. A nonpositional number system uses a limited number of symbols in which each symbol has a value. However, the position a symbol occupies in the number normally bears no relation to its value: the value of each symbol is normally fixed.
- In the decimal system, the base b = 10 and we use ten symbols to represent numbers. The symbols in this system are often referred to as **decimal digits** or just **digits**. In the binary system, the base b = 2 and we use only two symbols to represent numbers. The symbols in this system are often referred to as **binary digits** or **bits**. In a hexadecimal system, the base = 16 and we use 16 symbols to represent numbers. The symbols in this system are often referred to as **hexadecimal digits**. In an octal system, the base b = 8 and we use eight symbols to represent numbers. The symbols in this system are often referred to as octal digits.
- We can convert a number in any system to decimal. We multiply each digit with its place value in the source system and add the result to get the number in the decimal system. We can convert a decimal number to its equivalent in any base using two different procedure, one for the integral part and one for the fractional part. The integral part needs repeated division and the fraction part needs repeated multiplication.
- Conversion from the binary system to the hexadecimal system and from the hexadecimal system to the binary system is very easy, because four bits in the binary system are represented as one digit in the hexadecimal system.
- Conversion from the binary system to the octal system and from the octal system to the binary system is very easy, because three bits in the binary system is represented as one digit in the octal system.

2.5 PRACTICE SET

2.5.1 Quizzes

A set of interactive quizzes for this chapter can be found on the book website. It is strongly recommended that the student take the quizzes to check his/her understanding of the materials before continuing with the practice set.

2.5.2 Review questions

Q2-1. Define a number system.

Q2-2. Distinguish between positional and nonpositional number systems.

Q2-3. Define the base or radix in a positional number system. What is the relationship between a base and the number of symbols in a positional number system.

Q2-4. Explain the decimal system. Why is it called *decimal*? What is the base in this system?

Q2-5. Explain the binary system. Why is it called *binary*? What is the base in this system?

Q2-6. Explain the octal system. Why is it called *octal*? What is the base in this system?

Q2-7. Explain the hexadecimal system. Why is it called *hexadecimal*? What is the base in this system?

Q2-8. Why is it easy to convert from binary to hexadecimal and vice versa?

Q2-9. How many bits in the binary system are represented by one digit in the hexadecimal system?

Q2-10. How bits in the binary system are represented by one digit in the octal system?

2.5.3 Problems

P2-1. Convert the following binary numbers to decimal without using a calculator, showing your work:

a. $(01101)_2$ c. $(011110.01)_2$
b. $(1011000)_2$ d. $(111111.111)_2$

P2-2. Convert the following hexadecimal numbers to decimal without using a calculator, showing your work:

a. $(AB2)_{16}$ c. $(ABB)_{16}$
b. $(123)_{16}$ d. $(35E.E1)_{16}$

P2-3. Convert the following octal numbers to decimal without using a calculator, showing your work:

a. $(237)_8$ c. $(617.7)_8$
b. $(2731)_8$ d. $(21.11)_8$

P2-4. Convert the following decimal numbers to binary without using a calculator, showing your work:

a. 1234 c. 124.02
b. 88 d. 14.56

P2-5. Convert the following decimal numbers to octal without using a calculator, showing your work:

a. 1156
b. 99
c. 11.4
d. 72.8

P2-6. Convert the following decimal numbers to hexadecimal without using a calculator, showing your work:

a. 567
b. 1411
c. 12.13
d. 16

P2-7. Convert the following octal numbers to hexadecimal without using a calculator, showing your work:

a. $(514)_8$
b. $(411)_8$
c. $(13.7)_8$
d. $(1256)_8$

P2-8. Convert the following hexadecimal numbers to octal without using a calculator, showing your work:

a. $(51A)_{16}$
b. $(4E1)_{16}$
c. $(BB.C)_{16}$
d. $(ABC.D)_{16}$

P2-9. Convert the following binary numbers to octal without using a calculator, showing your work:

a. $(01101)_2$
b. $(1011000)_2$
c. $(011110.01)_2$
d. $(111111.111)_2$

P2-10. Convert the following binary numbers to hexadecimal without using a calculator, showing your work:

a. $(01101)_2$
b. $(1011000)_2$
c. $(011110.01)_2$
d. $(111111.111)_2$

P2-11. Convert the following decimal numbers to binary using the alternative method discussed in Example 2.17, showing your work:

a. 121
b. 78
c. 255
d. 214

P2-12. Change the following decimal numbers into binary using the alternative method discussed in Example 2.18, showing your work:

a. $3\frac{5}{8}$
b. $12\frac{3}{32}$
c. $4\frac{13}{64}$
d. $12\frac{5}{128}$

P2-13. In a positional number system with base b, the largest integer number that can be represented using k digits is $b^k - 1$. Find the largest number in each of the following systems with *six* digits.

a. Binary
b. Decimal
c. Hexadecimal
d. Octal

P2-14. Without converting, find the minimum number of digits needed in the destination system for each of the following cases:

a. Five-digit decimal number converted to binary.
b. Four-digit decimal converted to octal.
c. Seven-digit decimal converted to hexadecimal.

P2-15. Without converting, find the minimum number of digits needed in the destination system for each of the following cases:

a. 5-bit binary number converted to decimal.

b. Three-digit octal number converted to decimal.

c. Three-digit hexadecimal converted to decimal.

P2-16. The following table shows how to rewrite a fraction so the denominator is a power of two (1, 4, 8, 16, and so on).

Original	*New*	*Original*	*New*
0.5	$\frac{1}{2}$	0.25	$\frac{1}{4}$
0.125	$\frac{1}{8}$	0.0625	$\frac{1}{16}$
0.03125	$\frac{1}{32}$	0.015625	$\frac{1}{64}$

However, sometimes we need a combination of entries to find the appropriate fraction. For example, 0.625 is not in the table, but we know that 0.625 is 0.5 + 0.125. This means that 0.625 can be written as $\frac{1}{2} + \frac{1}{8}$, or $\frac{5}{8}$.
Change the following decimal fractions to a fraction with a power of 2.

a. 0.1875 c. 0.40625

b. 0.640625 d. 0.375

P2-17. Using the results of the previous problem, change the following decimal numbers to binary numbers.

a. 7.1875 c. 11.40625

b. 12.640625 d. 0.375

P2-18. Find the maximum value of an integer in each of the following cases:

a. b = 10, k = 10 c. b = 8, k = 8

b. b = 2, k = 12 d. b = 16, k = 7

P2-19. Find the minimum number of required bits to store the following integers:

a. less than 1000

b. less than 100000

c. less than 64

d. less than 256

P2-20. A number less than b^k can be represented using k digits in base b. Show the number of digits needed in each of the following cases.

a. Integers less than 2^{14} in binary

b. Integers less than 10^8 in decimal

c. Integers less than 8^{13} in octal

d. Integers less than 16^4 in hexadecimal

P2-21. A common base used on the Internet is b = 256. We need 256 symbols to represent a number in this system. Instead of creating this large number of symbols, the designers of this system have used decimal numbers to represent a symbol: 0 to 255. In other words, the set of symbols is S = {0, 1, 2, 3, ..., 255}. A number in this system is always in the format $S_1.S_2.S_3.S_4$ with four symbols and three dots that separate the symbols. The system is used to define Internet addresses (see Chapter 6). An example of an address in this system is 10.200.14.72, which is equivalent to $10 \times 256^3 + 200 \times 256^2 + 14 \times 256^1 + 72 \times 256^0 = 180883016$ in decimal. This number system is called *dotted decimal notation*. Find the decimal value of each of the following Internet addresses:

a. 17.234.34.14 c. 110.14.56.78
b. 14.56.234.56 d. 24.56.13.11

P2-22. Internet addresses described in the previous problem are also represented as patterns of bits. In this case, 32 bits are used to represent an address, eight bits for each symbol in dotted decimal notation. For example, the address 10.200.14.72 can also be represented as 00001010 11001000 00001110 01001000. Show the bit representation of the following Internet addresses:

a. 17.234.34.14 c. 110.14.56.78
b. 14.56.234.56 d. 24.56.13.11

P2-23. Write the decimal equivalent of the following Roman numbers:

a. XV c. VLIII
b. XXVII d. MCLVII

P2-24. Convert the following decimal numbers to Roman numbers:

a. 17 c. 82
b. 38 d. 999

P2-25. Find which of the following Roman numerals are not valid:

a. MMIM c. CVC
b. MIC d. VX

P2-26. Mayan civilization invented a positional vigesimal (base 20) numeral system, called the *Mayan numeral system*. They use base 20 probably because they used both their fingers and toes for counting. This system has 20 symbols that are constructed from three simpler symbols. The advanced feature of the system is that it has a symbol for zero, which is a shell. The other two symbols are a circle (or a pebble) for one and a horizontal bar (or a stick) for five. To represent a number greater than nineteen, numerals are written vertically. Search the Internet to answer the following: what are the decimal numbers 12, 123, 452, and 1256 in the Mayan numeral system.

P2-27. *Babylonian* civilization is credited for developing the first positional numeral system, called the *Babylonian numeral system*. They inherited the Sumerian and Akkadian numeral system and developed it into positional sexagesimal system (base 60). This base is still used today for times and angles. For example, one hour is 60 minutes and one minute is 60 seconds: similarly, one degree is 60 minutes and one minute is 60 seconds. As a positional system with base b requires b symbols (digits), we expect a positional sexagesimal system to require 60 symbols.

However, the Babylonians did not have a symbol for zero, and produced the other 59 symbols by stacking two symbols, those for one and ten. Search the Internet to answer the following questions:

a. Express the following decimal numbers in Babylonian numerals: 11291, 3646, 3582.

b. Mention problems that might arise from not having a symbol for 0. Find how the Babylonian numeral system addresses the problem.

2.5.4 Applets

We have created some Java applets to simulate some concepts discussed in each chapter. It is strongly recommended that the students active these applets to improve their understanding of the materials discussed in each chapter.