CHAPTER 2
Number Systems

2.1

Objectives

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

- ❑ Understand the concept of number systems.
- ❑ Distinguish between non-positional and positional number systems.
- ❑ Describe the decimal, binary, hexadecimal, and octal system.
- ❑ 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.

2.2

2-1 INTRODUCTION

A **number 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.

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.

2.3

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:

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

has the value of:

To be added later

in which S is the set of symbols, b is the base (or **radix**).

2.4

2.2.1 The decimal system (base 10)

The word decimal is derived from the Latin root **decem** (ten). In this system the **base b = 10** and we use ten symbols

$$S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

The symbols in this system are often referred to as **decimal digits** or just **digits**.

Integers

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

Figure 2.1 **Place values for an integer in the 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

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

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	=	− (	7×1000	+	5×100	+	0×10	+	8×1	) Values

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

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

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

2.9

2.2.2 The binary system (base 2)

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 = {1, 2}. The symbols in this system are often referred to as **binary digits** or **bits** (binary digit).

2.10

Integers

We can represent an integer as:

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

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

	2^{k-1}	2^{k-2}	$\bullet\bullet\bullet$	2^2	2^1	2^0	Place values
$\pm$	S_{k-1}	S_{k-2}	$\bullet\bullet\bullet$	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	$+\ 1 \times 2^3$	$+\ 0 \times 2^2$	$+\ 0 \times 2^1$	$+\ 1 \times 2^0$	Decimal

The equivalent decimal number is N = 16 + 8 + 0 + 0 + 1 = 25.

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:

	Integral part	.	Fractional part
$R = \pm$	$S_{k-1} \times 2^{k-1} + \ldots + S_1 \times 2^1 + S_0 \times 2^0$	+	$S_{-1} \times 2^{-1} + \ldots + S_{-l} \times 2^{-l}$

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

The value in decimal system is R = 4 + 0 + 1 + 0.5 + 0.25 = 5.75

2.13

The hexadecimal system (base 16)

The word **hexadecimal** is derived from the Greek root **hex** (six) and the Latin root **decem** (ten). In this system the **base b = 16** and we use sixteen symbols to represent a number. The set of symbols is

$$\boxed{S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}}$$

Note that the symbols A, B, C, D, E, F are equivalent to 10, 11, 12, 13, 14, and 15 respectively. The symbols in this system are often referred to as **hexadecimal digits**.

2.14

Integers

We can represent an integer as:

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

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

	16^{k-1}	16^{k-2}	$\cdots$	16^2	16^1	16^0	Place values
$\pm$	S_{k-1}	S_{k-2}	$\cdots$	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

The equivalent decimal number is N = 512 + 160 + 14 = 686.

2.2.4 The octal system (base 8)

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

Integers

We can represent an integer as:

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

Figure 2.3 **Place values for an integer in the octal system**

	8^{k-1}	8^{k-2}	$\cdots$	8^2	8^1	8^0	Place values
$\pm$	S_{k-1}	S_{k-2}	$\cdots$	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.

2.19

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	$(A2C.A1)_{16}$

2.20

Table 2.2 shows how the number 0 to 15 is represented in different systems.

Table 2.2 Comparison of numbers in the four systems

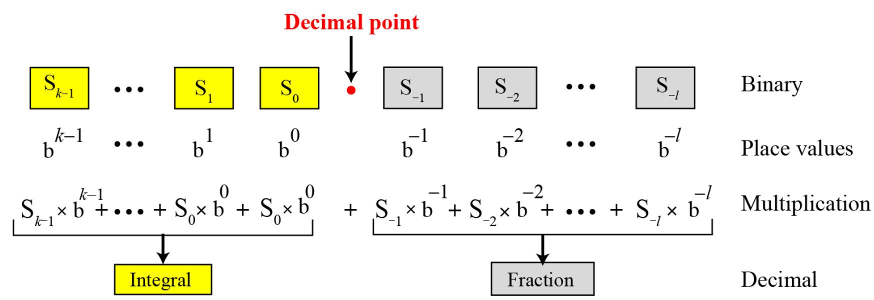
Decimal	*Binary*	*Octal*	*Hexadecimal*
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

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

Figure 2.5 **Converting other bases to 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).

2.25

Example 2.10

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

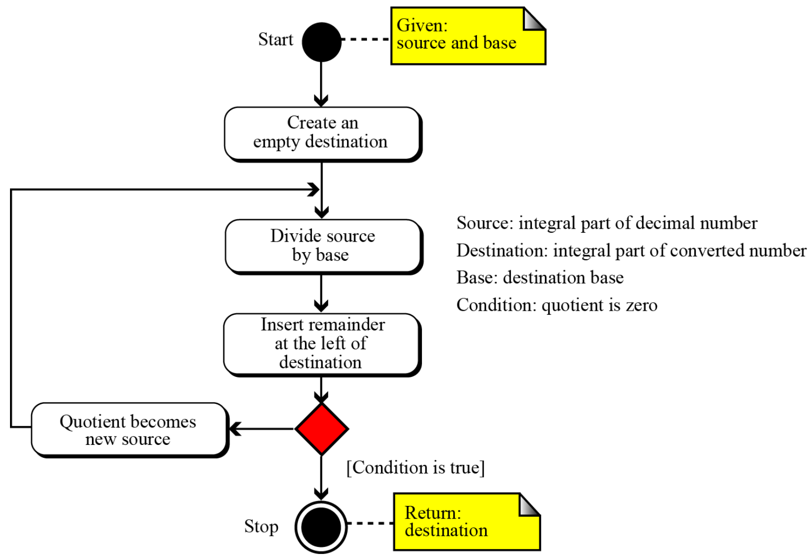
Octal	2		3	•	1		7
Place values	8^1		8^0		8^{-1}		8^{-2}
Partial result	16	+	3	+	0.125	+	0.109
Decimal: 19.234							

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

2.26

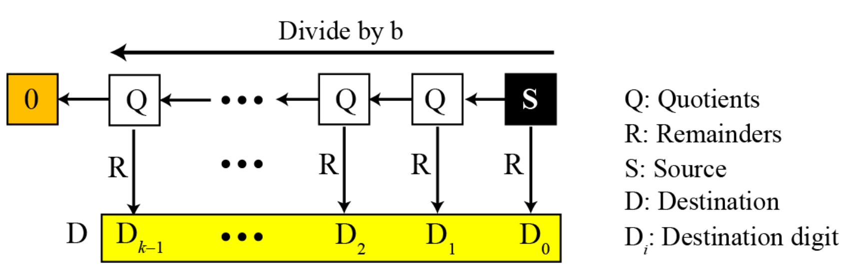
Decimal to any base

Figure 2.6 **Converting other bases to decimal (integral part)**



2.27

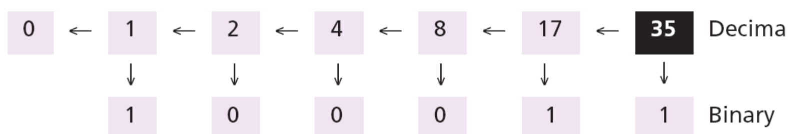
Figure 2.7 **Converting the integral part in decimal to other bases**



2.28

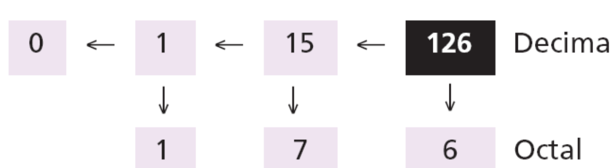
Example 2.11

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 = $(100011)_2$.



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 = $(176)_8$.



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 $126 = (7E)_{16}$

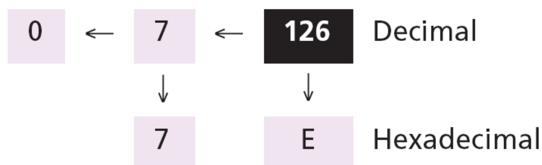


Figure 2.8 Converting fractional part in decimal to other bases

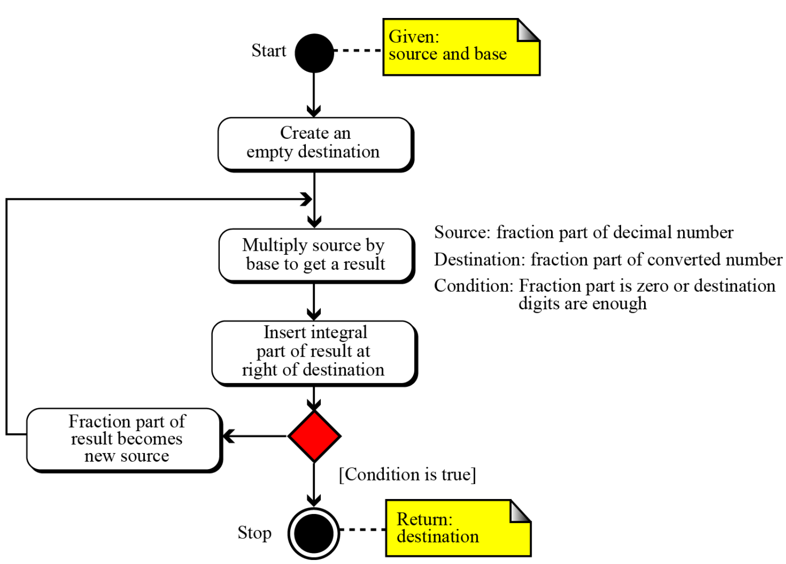
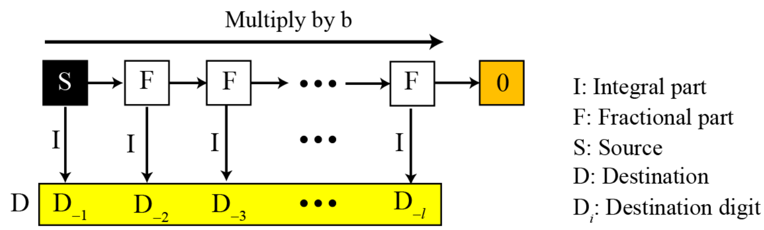


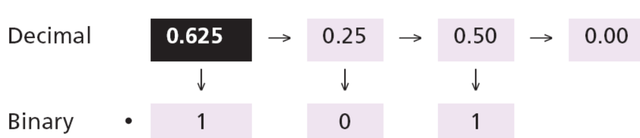
Figure 2.9 Converting fractional part



Note:
The fraction may never become zero.
Stop when enough digits have been created.

Example 2.14

Convert the decimal number 0.625 to binary.



Since the number 0.625 = $(0.101)_2$ has no integral part, the example shows how the fractional part is calculated.

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

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

2.37

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	1/2	1/4	1/8	1/16	1/32	1/64	1/128

Decimal = 27/64	16/64	+	8/64	+	2/64	+	1/64
	1/4	+	1/8	+	1/32	+	1/64

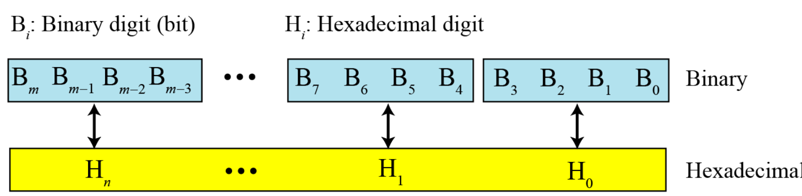
Decimal 27/64 =	0	+	1/4	+	1/8	+	0	+	1/32	+	1/64
Binary	0		1		1		0		1		1

The answer is then $(0.011011)_2$

2.38

Binary-hexadecimal conversion

Figure 2.10 **Binary to hexadecimal and hexadecimal to binary conversion**



2.39

Example 2.19

Show the hexadecimal equivalent of the binary number $(110011100010)_2$.

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 25 to change the number to hexadecimal: (4E2)16.

2.40

Example 2.20

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

Solution

Each hexadecimal digit is converted to 4-bit patterns:

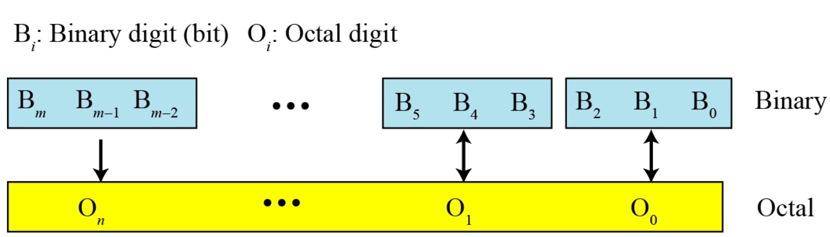
$$2 \rightarrow 0010,\ 4 \rightarrow 0100,\ \text{and } C \rightarrow 1100$$

The result is $(001001001100)_2$.

2.41

Binary-octal conversion

Figure 2.10 **Binary to octal and octal to binary conversion**



2.42

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 25.

$$\mathbf{101 \quad 110 \quad 010}$$

The result is $(562)_8$.

2.43

Example 2.22

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

Solution

Write each octal digit as its equivalent bit pattern to get

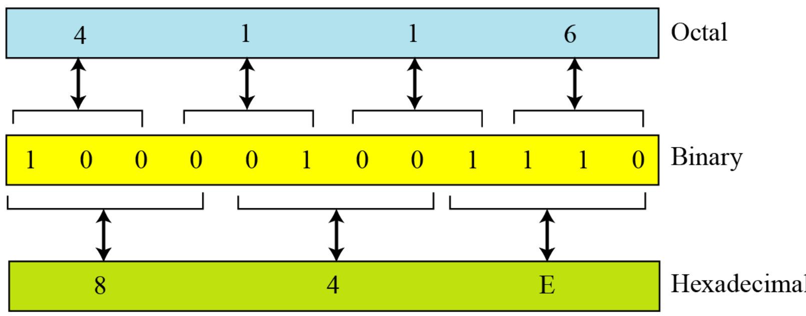
$$\mathbf{2 \rightarrow 010 \;\; \text{and} \;\; 4 \rightarrow 100}$$

The result is $(010100)_2$.

2.44

Octal-hexadecimal conversion

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



2.45

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

$$\boldsymbol{x = \lceil k \times (\log b_1 / \log b_2) \rceil = \lceil 6 \times (1 / 0.30103) \rceil = 20.}$$

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 twenty bits.

2.46

2-3 NONPOSITIONAL NUMBER SYSTEMS

Although **non-positional number systems** are not used in computers, we give a short review here for comparison with positional number systems. A non-positional 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.

2.47

In this system, a number is represented as:

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

and has the value of:

$$n = \pm \underbrace{S_{k-1} + \ldots + S_1 + S_0}_{\textit{Integral part}} + \underbrace{S_{-1} + S_{-2} + \ldots + S_{-l}}_{\textit{Fractional part}}$$

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

2.48

Example 2.24

Roman numerals are a good example of a non-positional number system. 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 (See the textbook).

2.49

Example 2.24 (Continued)

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

2.50