Chapter 2
Number Systems
Outlines

- Introduction
- Positional Number Systems
  - Base 10, 2, 8, 16.
- Nonpositional Number Systems
  - Roman Numerals
Objectives

After studying this chapter, the student should understand:

- The concept of number systems.
- Non-positional and positional number systems.
- Decimal, Binary, Hexadecimal and Octal system.
- Convert a number among binary, octal, hexadecimal, and decimal systems.
- Find the number of digits needed in each system to represent a particular value.
1-1 Introduction
The Definition of Number System

- A number can be represented using distinct symbols and 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.
- Two groups
  - positional and non-positional systems
1-2 Position Number Systems
Overview

- In a **positional number system**, a number represented as:
  \[ \pm \left( S_{k-1} \ldots S_2 S_1 S_0 \cdot S_{-1} S_{-2} \ldots S_{-l} \right)_b \]
  has the value of:

  \[ n = \pm \sum_{i=-l}^{k-1} S_i \times b^i \]

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

The Base includes Base10(Decimal), Base2(Binary), Base16(Hexadecimal), or Base8(octal).
The decimal system (base 10)

- The word decimal is derived from the Latin root *decem* (ten).
  - base $b = 10$, and
  - 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.
  - Integer examples
  - Real examples
$N = \pm \sum_{k=0}^{k-1} S_k \times 10^k$

Figure 2.1  Place values for an integer in the decimal system
Base 10 – Integers (2)

- Example 2.1 shows the place values for the integer +224 in the decimal system

\[ N = 2 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 \]

- Example 2.2 shows the place values for the decimal number −7508

\[ N = -(7 \times 1000 + 5 \times 100 + 0 \times 10 + 8 \times 1) \]
Base 10 – Reals

- A real – a number with a fractional part

\[
R = \pm \left( S_{k-1} \times 10^{k-1} + \ldots + S_1 \times 10^1 + S_0 \times 10^0 \right) + \left( S_{-1} \times 10^{-1} + \ldots + S_{-l} \times 10^{-l} \right)
\]

- Example 2.3 shows the place values for the real number +24.13.

\[
R = + \left( 2 \times 10 \right) + \left( 4 \times 1 \right) + \left( 1 \times 0.1 \right) + \left( 3 \times 0.01 \right)
\]

- Place values
- Number
- Values
The binary system (base 2)

- The word binary is derived from the Latin root *bini* (or two by two).
  - base \( b = 2 \), and
  - two symbols, \( S = \{0, 1\} \)

- The symbols in this system are often referred to as binary digits or bits (binary digit).
  - Integer examples
  - Real examples
Base 2 – Integers (1)

- We can represent an Integer as:

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

\[ \begin{array}{c}
2^{k-1} & 2^{k-2} & \ldots & 2^2 & 2^1 & 2^0 \\
S_{k-1} & S_{k-2} & \ldots & S_2 & S_1 & S_0 \\
\downarrow & \downarrow & \ldots & \downarrow & \downarrow & \downarrow \\
N = \pm S_{k-1} \times 2^{k-1} + S_{k-2} \times 2^{k-2} + \ldots + S_2 \times 2^2 + S_1 \times 2^1 + S_0 \times 2^0 \\
\end{array} \]

**Figure 2.2** Place values for an integer in the binary system
Example 2.4 shows that the number $(11001)_2$ in binary is the same as 25 in decimal. The subscript 2 shows that the base is 2.

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

\[
N = 2^4 \times 1 + 2^3 \times 1 + 2^2 \times 0 + 2^1 \times 0 + 2^0 \times 1 
\]
Example 2.5 shows that the number \((101.11)_2\) in binary is equal to the number 5.75 in decimal.

The equivalent decimal number is \(R = 4 + 0 + 1 + 0.5 + 0.25 = 5.75\).
The hexadecimal system (base 16)

- The word hexadecimal is derived from the Greek root **hex** (six) and the Latin root **decem** (ten).
  - **base** $b = 16$, and
  - sixteen symbols,
    - $S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$
- Symbol 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.
Base 16 – Integers (1)

- 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
Example 2.6 shows that the number \((2AE)_{16}\) in hexadecimal:

\[
N = 2 \times 16^2 + 10 \times 16^1 + 14 \times 16^0
\]

The equivalent decimal number is \(N = 512 + 160 + 14 = 686\).
The octal system (base 8)

- The word octal is derived from the Latin root *octo* (eight).
  - base $b = 8$, and
  - Eight symbols, $S = \{0, 1, 2, 3, 4, 5, 6, 7\}$
- Place values for an integer in the octal system
### Base 8 – Integers (1)

- We can represent an Integer as:

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

\[
\begin{array}{ccccccc}
8^{k-1} & 8^{k-2} & \cdots & 8^2 & 8^1 & 8^0 \\
\pm & S_{k-1} & S_{k-2} & \cdots & S_2 & S_1 & S_0 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
N = \pm S_{k-1} \times 8^{k-1} + S_{k-2} \times 8^{k-2} + \cdots + S_2 \times 8^2 + S_1 \times 8^1 + S_0 \times 8^0 & \text{Values}
\end{array}
\]

**Figure 2.4** Place values for an integer in the octal system
Example 2.7 shows that the number \((1256)_8\) in octal is the same as 686 in decimal.

\[
N = 1 \times 8^3 + 2 \times 8^2 + 5 \times 8^1 + 6 \times 8^0
\]

The decimal number is \(N = 512 + 128 + 40 + 6 = 686\).
# Summary of the Base 10/2/8/16 Positional Systems (1)

## Table 2.1 Summary of the Four Positional Number Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Base</th>
<th>Symbols</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal</td>
<td>10</td>
<td>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</td>
<td>2345.56</td>
</tr>
<tr>
<td>Binary</td>
<td>2</td>
<td>0, 1</td>
<td>(1001.11)₂</td>
</tr>
<tr>
<td>Octal</td>
<td>8</td>
<td>0, 1, 2, 3, 4, 5, 6, 7</td>
<td>(156.23)₈</td>
</tr>
<tr>
<td>Hexadecimal</td>
<td>16</td>
<td>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F</td>
<td>(A2C.A1)₁₆</td>
</tr>
</tbody>
</table>
Summary (2)

The number 0 to 15 is represented in different systems

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Octal</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>101</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>110</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>111</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>11</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>12</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>1011</td>
<td>13</td>
<td>B</td>
</tr>
<tr>
<td>12</td>
<td>1100</td>
<td>14</td>
<td>C</td>
</tr>
<tr>
<td>13</td>
<td>1101</td>
<td>15</td>
<td>D</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
<td>16</td>
<td>E</td>
</tr>
<tr>
<td>15</td>
<td>1111</td>
<td>17</td>
<td>F</td>
</tr>
</tbody>
</table>
Conversion

- The decimal system is more familiar than the other systems
- To convert a number in one system to the equivalent number in another system
  - Any base – Decimal
  - Binary – Hexadecimal
  - Binary – Octal
  - Octal – Hexadecimal
Any base to decimal conversion (1)

- Converting other bases to decimal (Fig. 2.5)
Any base to decimal conversion (2)

- Example 2.8 shows how to convert the binary number \((110.11)_2\) to decimal: \((110.11)_2 = 6.75\).

<table>
<thead>
<tr>
<th>Binary</th>
<th>Place values</th>
<th>Partial results</th>
<th>Decimal: 6.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2^2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2^1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2^0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>•</td>
<td>2^-1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2^-2</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>
Example 2.9 shows how to convert the hexadecimal number \((1A.23)_{16}\) to decimal.

<table>
<thead>
<tr>
<th>Hexadecimal Place values</th>
<th>1</th>
<th>A</th>
<th>•</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place values</td>
<td>16&lt;sup&gt;1&lt;/sup&gt;</td>
<td>16&lt;sup&gt;0&lt;/sup&gt;</td>
<td>16&lt;sup&gt;-1&lt;/sup&gt;</td>
<td>16&lt;sup&gt;-2&lt;/sup&gt;</td>
<td></td>
</tr>
<tr>
<td>Partial result</td>
<td>16</td>
<td>10</td>
<td>0.125</td>
<td>0.012</td>
<td></td>
</tr>
<tr>
<td>Decimal:</td>
<td>26.137</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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).
Any base to decimal conversion (4)

- Example 2.10 shows how to convert \((23.17)_8\) to decimal.

<table>
<thead>
<tr>
<th>Octal Place values</th>
<th>Partial result</th>
<th>Decimal: 19.234</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 (8^1)</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>3 (8^0)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>\cdot 1 (8^{-1})</td>
<td>0.125</td>
<td></td>
</tr>
<tr>
<td>7 (8^{-2})</td>
<td>0.109</td>
<td></td>
</tr>
</tbody>
</table>

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 conversion

- Two procedures for converting a decimal number to its equivalent in any base.
  - Converting the integral part
  - Converting the fractional part
Converting the integral part (1)

UML’s state diagram

Start

Create an empty destination

Divide source by base

Insert remainder at the left of destination

Quotient becomes new source

[Condition is true]

Stop

Given: source and base

Source: integral part of decimal number
Destination: integral part of converted number
Base: destination base
Condition: quotient is zero

Return: destination
Converting the integral part (2)

- The Figure shows the destination is made with each repetition. (process manually)

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

- Divide by \( b \)
- \( Q \): Quotients
- \( R \): Remainders
- \( S \): Source
- \( D \): Destination
- \( D_i \): Destination digit
Converting the integral part (3)

- Example 2.11 shows how to convert 35 in decimal to binary. The result is $35 = (100011)_2$.

- Example 2.12 shows how to convert 126 in decimal to octal. The result is $126 = (176)_8$. 
Example 2.13 shows how to convert 126 in decimal to hexadecimal. The result is $126 = (7E)_{16}$
Converting the fractional part (1)

UML’s State diagram

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
Converting the fractional part (2)

- The Figure shows the destination is made with each repetition. (process manually)

Figure 2.9
Converting the fractional part of a number in decimal to other bases
Converting the fractional part (3)

- Example 2.14 converts the decimal number 0.625 to binary.

<table>
<thead>
<tr>
<th>Decimal</th>
<th>0.625</th>
<th>→</th>
<th>0.25</th>
<th>→</th>
<th>0.50</th>
<th>→</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>□ 1 □</td>
<td></td>
<td>□ 0 □</td>
<td></td>
<td>□ 1 □</td>
<td></td>
<td>□ 0 □</td>
</tr>
</tbody>
</table>

Since the number 0.625 = (0.101)₂ has no integral part, the example shows how the fractional part is calculated.
Converting the fractional part (4)

- Example 2.15 shows how to convert 0.634 to octal using a maximum of four digits. The result is \(0.634 = (0.5044)_8\).

  \[
  \begin{array}{cccccc}
  \text{Decimal} & 0.634 & \rightarrow & 0.072 & \rightarrow & 0.576 & \rightarrow & 0.608 & \rightarrow & 0.864 \\
  \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
  \text{Octal} & 5 & 0 & 4 & 4 & \\
  \end{array}
  \]

- Example 2.16 shows how to convert 178.6 in decimal to hexadecimal using only one digit to the right of the decimal point. The result is \(178.6 = (B2.9)_{16}\).

  \[
  \begin{array}{cccccc}
  \text{Decimal} & 0 & \leftarrow & 11 & \leftarrow & \text{178} & 0.6 & \rightarrow & 0.6 \\
  \downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
  \text{Hexadecimal} & B & 2 & \cdot & 9 & \\
  \end{array}
  \]
Converting the fractional part (5)

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

<table>
<thead>
<tr>
<th>Place values</th>
<th>$2^7$</th>
<th>$2^6$</th>
<th>$2^5$</th>
<th>$2^4$</th>
<th>$2^3$</th>
<th>$2^2$</th>
<th>$2^1$</th>
<th>$2^0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal equivalent</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

For Example:

Decimal 165 = \[128\, \, +\, \, 0\, \, +\, \, 32\, \, +\, \, 0\, \, +\, \, 0\, \, +\, \, 0\, \, +\, \, 4\, \, +\, \, 0\, \, +\, \, 1\]

Binary \[1\, \, 0\, \, 1\, \, 0\, \, 0\, \, 1\, \, 0\, \, 1\]
Converting the fractional part (6)

- A method can be used to convert a decimal fraction to binary when the denominator is a power of two:
- Convert $27/64$ to binary: The answer is $(0.011011)_2$

<table>
<thead>
<tr>
<th>Place values</th>
<th>$2^{-1}$</th>
<th>$2^{-2}$</th>
<th>$2^{-3}$</th>
<th>$2^{-4}$</th>
<th>$2^{-5}$</th>
<th>$2^{-6}$</th>
<th>$2^{-7}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal equivalent</td>
<td>$1/2$</td>
<td>$1/4$</td>
<td>$1/8$</td>
<td>$1/16$</td>
<td>$1/32$</td>
<td>$1/64$</td>
<td>$1/128$</td>
</tr>
</tbody>
</table>

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

\[
0 \ 1 \ 1 \ 0 \ 1 \ 1
\]
Binary-hexadecimal conversion (1)

- A relationship between the two bases: four bits in binary is one digit in hexadecimal

**Figure 2.10**
Binary to hexadecimal and hexadecimal to binary conversion
Example 2.19 shows the hexadecimal equivalent of the binary number \((110011100010)_2\).

\[ \begin{array}{ccc}
100 & 1110 & 0010 \\
\end{array} \]

- First, arranging the binary number in 4-bit patterns
- Using the equivalent of each pattern shown in Table 2.2 on page 25
- Then, changing the number to hexadecimal: \((4E2)_{16}\).
Binary-hexadecimal conversion (3)

- Example 2.20: What is the binary equivalent of $(24C)_{16}$?
- Each hexadecimal digit is converted to 4-bit patterns. The result is $(001001001100)_2$.

$2 \rightarrow 0010$, $4 \rightarrow 0100$, and $C \rightarrow 1100$
Binary-octal conversion (1)

- A relationship between the two bases: three bits in binary is one octal digit

$B_i$: Binary digit (bit)  $O_i$: Octal digit

Figure 2.10  Binary to octal and octal to binary conversion
Binary-octal conversion (2)

- Example 2.21 shows the octal equivalent of the binary number \((101110010)_2\).
  - Each group of three bits is translated into one octal digit (Table 2.2). The result is \((562)_8\).

\[
\begin{align*}
101 & \quad 110 & \quad 010 \\
\end{align*}
\]

- Example 2.22: What is the binary equivalent of for \((24)_8\)?
  - Write each octal digit as its equivalent bit pattern to get. The result is \((010100)_2\).

\[
2 \rightarrow 010 \quad \text{and} \quad 4 \rightarrow 100
\]
Octal-hexadecimal conversion (1)

- Using the binary system as the intermediate system.

![Diagram](image)

Figure 2.12
Octal to hexadecimal and hexadecimal to octal conversion
Octal-hexadecimal conversion (2)

- Example 2.23 – to find the minimum number of binary digits required to store decimal integers with a maximum of six digits.
  - \( k = 6, \ b_1 = 10, \) and \( b_2 = 2. \) Then

\[
x = \left\lceil k \times \left( \log_{b_1} \left( \frac{1}{\log_{b_2} \frac{1}{b_1}} \right) \right) \right\rceil = \left\lceil 6 \times \left( 1 / 0.30103 \right) \right\rceil = 20.
\]

- The largest six-digit decimal number is 999,999.
- The largest 20-bit binary number is 1,048,575.
- The largest 19-bit number is 524,287, which is smaller than 999,999.
- We definitely need twenty bits.
1-3 Nonpositional Number Systems
Overview (1)

- Non-positional number systems are not used in computers.
- A non-positional number system still uses a limited number of symbols in which each symbol has a value.
- We give a short review for comparison with positional number systems.
Overview (2)

- A number is represented as:

\[ s_{k-1} \ldots s_2 s_1 s_0 \cdot s_{-1} s_{-2} \ldots s_{-l} \]

and has the value of:

\[
\begin{align*}
&n = \pm \quad \text{Integral part} \\
&\quad s_{k-1} + \ldots + s_1 + s_0 \\
&\quad + \quad \text{Fractional part} \\
&\quad s_{-1} + s_{-2} + \ldots + s_{-l}
\end{align*}
\]

Some exceptions to the addition rule, as shown in Example 2.24 (Roman Numerals).
Roman numerals

- Roman numerals is a non-positional number system
  - The set of symbols, $S = \{I, V, X, L, C, D, M\}$.
  - Table 2.3 shows the values of each symbol

<table>
<thead>
<tr>
<th>Symbol</th>
<th>I</th>
<th>V</th>
<th>X</th>
<th>L</th>
<th>C</th>
<th>D</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>50</td>
<td>100</td>
<td>500</td>
<td>1000</td>
</tr>
</tbody>
</table>

- To find the value of a number, we need to add the value of symbols subject to specific rules (See Page 34).
<table>
<thead>
<tr>
<th>Roman Numeral</th>
<th>Conversion</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>III</td>
<td>1 + 1 + 1</td>
<td>3</td>
</tr>
<tr>
<td>IV</td>
<td>5 − 1</td>
<td>4</td>
</tr>
<tr>
<td>VIII</td>
<td>5 + 1 + 1 + 1</td>
<td>8</td>
</tr>
<tr>
<td>XVIII</td>
<td>10 + 5 + 1 + 1 + 1</td>
<td>18</td>
</tr>
<tr>
<td>XIX</td>
<td>10 + (10 − 1)</td>
<td>19</td>
</tr>
<tr>
<td>LXXII</td>
<td>50 + 10 + 10 + 1 + 1</td>
<td>72</td>
</tr>
<tr>
<td>CI</td>
<td>100 + 1</td>
<td>101</td>
</tr>
<tr>
<td>MMVII</td>
<td>1000 + 1000 + 5 + 1 + 1</td>
<td>2007</td>
</tr>
<tr>
<td>MDC</td>
<td>1000 + 500 + 100</td>
<td>1600</td>
</tr>
</tbody>
</table>