Numeral System and Its Importance

HONG@IS.NAIST.JP

ROOM: B405
Contents

➢ Number System Introduction
  ✓ Number systems used by human
  ✓ Number systems used by computer

➢ Number System Conversion

➢ Signed Number Representation

➢ Fractional Number Representation
  ✓ Fixed point number
  ✓ Floating point number

➢ Size of Number: Complexity vs. Performance
Number systems used by human

- Number systems is a writing system for expressing number
- Number system is necessary for solving mathematics problem

Simple natural number: 1,2,3,…,10

https://www.youtube.com/watch?v=cy-8lPVKLIo
Number systems used by human

- For business activities → Whole number and Natural number are developed, ex: 0, 1, 2, 305, etc.
To share the foods ➔ Fractional number have been developed, ex: 1/5, 1/2, etc.
Number systems used by human

- For building house, pyramid, complex architecture, etc.
- Science discovery activities: discover the universe, stars, moon, etc.
- Irrational number, real number, complex number have developed.
Number systems used by human

Natural Number

Whole number

Integer

Rational Number

a, b: real numbers
j: imaginary unit (j^2 = -1)

Real Number

Complex Number

Imaginary Number

a + jb

1 + j2, 1.54 – j0.3, etc

Fractional number

1/2, 0.35, 5, 3/5, etc

1/2, 0.35, 5, 3/5, etc

Integer

Whole number

Natural Number

\( \pi = 3.141592... \)

\( \sqrt{2} = 1.41421... \)
Number systems used by Computer

- Electronic circuit or computer operates based on the on-off switch of electric current. ⇒ The basic number system of computer is BINARY number.

- One binary number is called as a BIT

- A long chain of BITs may be difficult to remembered for computer engineers ⇒ OCTAL and HEXA-DECIMAL number systems are more convenient.

Binary: 100111010110
Octal: 4726
\[(100111010110)_2 = (4726)_8\]

Binary: 100111010110
Hexa-decimal: 9B6
\[(100111010110)_2 = (9B6)_{16}\]
For computer users, who do not familiar with computer architecture → Binary, Octal, and Hexa-decimal are difficult for them

→ Decimal System (base-10) is used.

SUMMARY:

<table>
<thead>
<tr>
<th>System</th>
<th>Base</th>
<th>Digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>2</td>
<td>0, 1</td>
</tr>
<tr>
<td>Octal</td>
<td>8</td>
<td>0, 1, 2, 3, 4, 5, 6, 7</td>
</tr>
<tr>
<td>Hexa-decimal</td>
<td>16</td>
<td>0, 1, 2, ..., 9, A, B, C, D, E, F</td>
</tr>
<tr>
<td>Decimal</td>
<td>10</td>
<td>0, 1, 2, ..., 9</td>
</tr>
</tbody>
</table>
Contents

- Number System Introduction
  - Number systems used by human
  - Number systems used by computer

- Number System Conversion

- Signed Number Representation

- Fractional Number Representation
  - Fixed point number
  - Floating point number

- Size of Number: Complexity vs. Performance
Number System Representation

- A number has two parts: Integer & Fractional parts
- Position notation of number
  
  \[ N = (a_{n-1} \cdots a_0 \cdot a_{-1} \cdots a_{-m})_r \]
  
  Example: \( (13.25)_{10} \rightarrow (r = 10, a_1 = 1, a_0 = 3, a_{-1} = 2, a_{-2} = 5) \)
  
  \( (1001.1101)_2 \), etc.

- Polynomial notation of number:
  
  \[ N = \sum_{i=-m}^{n-1} a_i r^i = a_{n-1} \times r^{n-1} + \cdots + a_{-m} \times r^{-m} \]

  Example: \( N = (13.25)_{10} \) can be written as:
  
  \[ N = 1 \times 10^1 + 3 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2} \]
  
  \( N = (101.1101)_2 \) can be rewritten as:
  
  \[ N = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \]
  
  \[ N = 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \]
Exercise 4-1

Write the polynomial notation of the following numbers:

N = (101.1101)_2 \rightarrow N = ?

N = (765.02)_8 \rightarrow N = ?

N = (25.5)_{10} \rightarrow N = ?
Exercise 4-1: Answer

\( i = 2 \ 1 \ 0 \ -1 \ -2 \ -3 \ -4 \)

\[ N = (101.1101)_2 \Rightarrow N = 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} \]

Or, \[ N = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} + 0 \times 2^{-3} + 1 \times 2^{-4} \]

\( i = 2 \ 1 \ 0 \ -1 \ -2 \)

\[ N = (765.02)_8 \Rightarrow N = 7 \times 8^2 + 6 \times 8^1 + 5 \times 8^0 + 0 \times 8^{-1} + 2 \times 8^{-2} \]

\( i = 1 \ 0 \ -1 \)

\[ N = (25.5)_{10} \Rightarrow N = 2 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1} \]
Number conversions

- Convert from base-B to base-10 (B can be 2, 8, 16):
- Convert from base-10 to base-B
  - Convert the **Integer** part: Radix **Divide** method
  - Convert the **Fractional** part: Radix **Multiplier** method
Number conversions

Convert from base-B to base-10 (B can be 2, 8, 16):

Two steps:
- Step 1: Write the polynomial notation of base-B number
- Step 2: Calculate the polynomial using base-10 arithmetic

Example-1: \((101.1)_2 \rightarrow (??)_{10}\)

Step 1: \((101.1)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}\)

Step 2: \((101.1)_2 = (4)_{10} + (0)_{10} + (1)_{10} + (0.5)_{10} = (5.5)_{10} \rightarrow (101.1)_2 = (5.5)_{10}\)

Example-2: \((9e)_{16} \rightarrow (??)_{10}\)

Step 1: \((9e)_{16} = 9 \times 16^1 + 14 \times 16^0\)

Step 2: \((9e)_{16} = (144)_{10} + (14)_{10} = (158)_{10} \rightarrow (9e)_{16} = (158)_{10}\)
Exercise 4-2

Convert the following numbers into base-10:

✓ $(11011011)_2 = (??)_{10}$
✓ $(25)_8 = (??)_{10}$
Exercise 4-2: Answer

\[(11011011)_2 = (?,?)_{10}\]

\[\rightarrow (11011011)_2 = 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^0\]

\[= 128 + 64 + 16 + 8 + 2 + 1\]

\[= (219)_{10}\]

\[(25)_8 = (?,?)_{10}\]

\[\rightarrow (25)_8 = 2 \times 8^1 + 5 \times 8^0 = 16 + 5 = (21)_{10}\]
Number conversions

- Convert from base-10 to base-B

\[(N)_{10} = (A)_B\]

1. Convert the integer part:
   use Radix Divide Method

\[(13)_{10} = (?)_2 \quad (B=2)\]

| Q(0) = 13 | Q(2)/2 = 3/2 = 1, remainder: 1 |
| Q(0)/2 = 13/2 = 6, remainder: 1 | \rightarrow a(3) = 1 |
| \rightarrow a(0) = 1 | \rightarrow Q(3) = 1 |
| \rightarrow Q(1) = 6 | |
| Q(1)/2 = 6/2 = 3, remainder: 0 | Q(3)/2 = 1/2 = 0, remainder: 1 |
| \rightarrow a(0) = 0 | \rightarrow a(4) = 1 |
| \rightarrow Q(2) = 3 | \rightarrow Q(4) = 0 \rightarrow completed! |

\rightarrow (13)_{10} = (1101)_{10}
Number conversions

Convert from base-10 to base-\(B\)

\((N)_{10} = (A)_{B}\)

2. Convert the fraction part:
Use Radix Multiply method

\((0.6875)_{10} = (?)_2\) \((B=2)\)

| \(F(0) = 0.6875\) | \(0.6875 \times 2 = 1.375 \Rightarrow a(0) = 1\) |
| \(F(1) = 0.375\) | \(0.375 \times 2 = 0.75 \Rightarrow a(1) = 0\) |
| \(F(2) = 0.75\) | \(0.75 \times 2 = 1.5 \Rightarrow a(2) = 1\) |
| \(F(3) = 0.5\) | \(0.5 \times 2 = 1.0 \Rightarrow a(3) = 1\) |
| \(F(4) = 0 \Rightarrow completed\) | \(\Rightarrow (0.6875)_{10} = (0.1011)_2\) |
Number conversions

Quick method to convert from binary to base-2\(^x\) (octal \(x = 3\), hexa \(x = 4\)):

- **Step-1**: Divide binary number into sets of \(x\) digits, add leading zeros as needed.

Ex: 1011100 \(\rightarrow\) **001 011 100** (octal case)

\(\rightarrow\) **0101 1100** (hexadecimal case)

- **Step-2**: Use the below tables to convert each \(x\)-digits to a single octal/hex digit

**Examples:**

\((001 011 100)_2 \rightarrow (134)_8\)
\((0101 1100)_2 \rightarrow (5c)_{16}\)

<table>
<thead>
<tr>
<th>Binary</th>
<th>Octal</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>011</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>110</td>
<td>6</td>
</tr>
<tr>
<td>111</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>A (or a)</td>
</tr>
<tr>
<td>1011</td>
<td>B (or b)</td>
</tr>
<tr>
<td>1100</td>
<td>C (or c)</td>
</tr>
<tr>
<td>1101</td>
<td>D (or d)</td>
</tr>
<tr>
<td>1110</td>
<td>E (or e)</td>
</tr>
<tr>
<td>1111</td>
<td>F (or f)</td>
</tr>
</tbody>
</table>
Number conversions

Quick method to convert from base-$2^x$ to binary (octal $x = 3$, hexa $x = 4$):

- **Step-1:** Use below table to convert each octal/hexa digit into $x$-digit of binary.
- **Step-2:** Delete the left-most zero digits

Examples:
- $(35)_8 \rightarrow (011 101)_2$
- $(011 101)_2 \rightarrow (11101)_2$
- $(2F)_{16} \rightarrow (0010 1111)_2$
- $(0010 1111)_2 \rightarrow (101111)_2$

<table>
<thead>
<tr>
<th>Binary</th>
<th>Octal</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>2</td>
</tr>
<tr>
<td>011</td>
<td>3</td>
</tr>
<tr>
<td>100</td>
<td>4</td>
</tr>
<tr>
<td>101</td>
<td>5</td>
</tr>
<tr>
<td>110</td>
<td>6</td>
</tr>
<tr>
<td>111</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Binary</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
</tr>
<tr>
<td>0001</td>
<td>1</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
</tr>
<tr>
<td>0011</td>
<td>3</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
</tr>
<tr>
<td>0110</td>
<td>6</td>
</tr>
<tr>
<td>0111</td>
<td>7</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
</tr>
<tr>
<td>1001</td>
<td>9</td>
</tr>
<tr>
<td>1010</td>
<td>A (or a)</td>
</tr>
<tr>
<td>1011</td>
<td>B (or b)</td>
</tr>
<tr>
<td>1100</td>
<td>C (or c)</td>
</tr>
<tr>
<td>1101</td>
<td>D (or d)</td>
</tr>
<tr>
<td>1110</td>
<td>E (or e)</td>
</tr>
<tr>
<td>1111</td>
<td>F (or f)</td>
</tr>
</tbody>
</table>
Exercise 4-3

- Convert the following numbers
  - $(58)_{16} = (?)_2$
  - $(1011010)_2 = (?)_{16} = (?)_8$
Exercise 4-3: Answer

- \((58)_{16} = (0101 1000)_{2}\)
- \((0101 1010)_{2} = (5a)_{16}\)
- \((001 011 010)_{2} = (132)_{8}\)
Contents

➢ Number System Introduction
  ✓ Number systems used by human
  ✓ Number systems used by computer

➢ Number System Conversion

➢ Signed Number Representation

➢ Fractional Number Representation
  ✓ Fixed point number
  ✓ Floating point number

➢ Size of Number: Complexity vs. Performance
How to present signed number?

- In mathematics, negative number is represented by minus “-” sign.

- In computer hardware, numbers are represented as sequence of bits (0, 1) → how to represent negative number?

- Signed Number Representation Methods:
  - Signed magnitude representation
  - Ones’ complement
  - Two’s complement
Signed number Representation
- Signed magnitude representation-

Signed magnitude representation:

✓ Sign bit: commonly represented by the most significant bit (MSB) (0 mean positive, 1 means negative).

✓ Magnitude: represented by the remained bits. It shows the absolute value of the number.

Problem: Zero value can be represented as 0 0000 or 1 0000

Example: \((0101)_2 = (5)_{10} ; (1101)_2 = (-5)_{10}\)
Signed number Representation
- Ones’ Complement -

- The ones’ complement form of a negative binary number is the bitwise NOT of its positive number. Ex: \((5)_{10} = (0101)_2 \rightarrow (-5)_{10} = (1010)_2\)
- If using \(N\) bits to represent a number \(\rightarrow\) its range will be \(-2^{N-1} - 1\) to \(2^{N-1} - 1\)
- Steps to add two numbers:
  - Do conventional binary addition
  - Add an end-around carry
- Problems:
  - Multiple zeros (0000 or 1111)
  - End-around carry

Example:

<table>
<thead>
<tr>
<th>binary</th>
<th>decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111110</td>
<td>-1</td>
</tr>
<tr>
<td>+ 00000010</td>
<td>+2</td>
</tr>
<tr>
<td>1 00000000</td>
<td>0</td>
</tr>
<tr>
<td>00000001</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{array}{ccc}
0001 & 0110 & 22 \\
+ 0000 & 0011 & 3 \\
\hline
0001 & 1001 & 25 \\
\end{array}
\]
Signed number Representation
- Two’s Complement (1/2) -

- To avoid the end-around carry and multiple zeros problem, the two’s complement has been introduced and widely used.

- The negative number is done by inverting all the bits and adding 1 to the result.

- There is only one zero: 00000

- Method to find the negation of a number in two’s complement

  - Method 1:
    - Invert all the bits
    - Add the result to 1

  - Example: Find \((-3)_{10} = (?)_{2}\) using 4 bits
    - Invert: 0011 (3) → 1100
    - Add 1: 1100 + 1 = 1101 (-3)

  - Method 2:
    - Find the first ‘1’ (start from right)
    - Invert all the bits in the left of that ‘1’.

  - Example: Find \((-3)_{10} = (?)_{2}\)
    - Find the first ‘1’: 0011 (3)
    - Invert all the left bits: 0011 → 1101 (-3)
Signed number Representation
- Two’s Complement (2/2) -

- Add two number in two’s complement format:
  - Perform the binary adding
  - Ignore the last carry bit.

- For example: Perform the following additions using 4 bits

\[
\begin{array}{c}
1010 \text{ (-6)} \\
+0101 \text{ (5)} \\
\hline
1111 \text{ (-1)}
\end{array}
\]

\[
\begin{array}{c}
0011 \text{ (3)} \\
+1110 \text{ (-2)} \\
\hline
\text{X0001 (1)}
\end{array}
\]
Exercise 4-4

- Find the signed magnitude representation of the following numbers (using 5 bits):
  - if \((7)_{10} = (00111)_2\) \(\Rightarrow\) \((-7)_{10} = (?)_2\)
  - if \((10)_{10} = (01010)_2\) \(\Rightarrow\) \((-10)_{10} = (?)_2\)

- Find the ones’ complement of the following numbers (using 5 bits)
  - \((-7)_{10} = (?)_2\)
  - \((-10)_{10} = (?)_2\)

- Find the negation of the following two’s complement numbers:
  - \(0111010\) \(\Rightarrow\) \(1010011\) \(\Rightarrow\) \(110111010\)
Exercise 4-4: Answer

- Find the signed magnitude representation of the following numbers (using 5 bits):
  ✓ if \((7)_{10} = (00111)_2\) → \((-7)_{10} = (10111)_2\)
  ✓ if \((10)_{10} = (01010)_2\) → \((-10)_{10} = (11010)_2\)

- Find the ones’ complement of the following numbers (using 5 bits)
  ✓ \((-7)_{10} = (11000)_2\) \((-10)_{10} = (10101)_2\)

- Find the negation of the following two’s complement numbers:
  ✓ \(0111010\) \(1010011\) \(110111010\)
  → \(1000110\) \(0101101\) \(001000110\)
Contents

- Number System Introduction
  - Number systems used by human
  - Number systems used by computer

- Number System Conversion

- Signed Number Representation

- Fractional Number Representation
  - Fixed point number
  - Floating point number

- Size of Number: Complexity vs. Performance
Fixed Point Number

**Fixed-point** number represents a **real number** having a **fixed number** of digits after the radix point

For examples: \( m = 3 \) bits, \( f = 4 \) bits, \( t = 7 \) bits

\[
(0101100)_2 = (010.1100)_2 = (2.75)_{10}
\]
Floating Point Number

- Floating point number is a **formula** representation of a real number in trade-off between **Range** and **Precision**

For example: \( 1452 \times 10^{-3} = 1.452 \quad 1452 \times 10^{-2} = 14.52 \)

- It means that the **radix point can “float”**.
- In computer, only base 2 can be used.

\[
\text{significand} \times \text{base}^{-\text{exponent}}
\]

For example: \( a = 3, b = 4, t = 8 \rightarrow (10100011)_2 = -3 \times 2^{-2} = (-0.75)_{10} \)

For example: \( 1452 \times 10^{-3} = 1.452 \)

- It means that the **radix point can “float”**.
- In computer, only base 2 can be used.
Exercise 4-5

Find the equivalent **decimal** values of the following unsigned fixed point numbers (m = 4, f = 4, t = 8):

✓ (10011100)₂ = (?)₁₀

Find the equivalent **decimal** values of the following signed fixed point numbers (m = 3, f = 4, t = 8):

✓ (10011100)₂ = (?)₁₀

Find the equivalent **decimal** values of the following floating point numbers (a = 3, b = 4, t = 8):

✓ (00110101)₂ = (?)₁₀
Exercise 4-5: Answer

- Find the equivalent decimal values of the following unsigned fixed point numbers \((m = 4, f = 4, t = 8)\):
  - \((1001.1100)_2 = (9.75)_{10}\)

- Find the equivalent decimal values of the following signed fixed point numbers \((m = 3, f = 4, t = 8)\):
  - \((1 001.1100)_2 = (-1.75)_{10}\)

- Find the equivalent decimal values of the following floating point numbers \((a = 3, b = 4, t = 8)\):
  - \((0 011 0101)_2 = (+5 \times 2^{-3})_{10} = (0.625)\)
Contents

- Number System Introduction
  - Number systems used by human
  - Number systems used by computer

- Number System Conversion

- Signed Number Representation

- Fractional Number Representation
  - Fixed point number
  - Floating point number

- Size of Number: Complexity vs. Performance
Size of Number: Complexity vs. Performance

- Memory space: each binary bit is represented by a FlipFlop.
- Combinational circuit: the number of bits needs to be equal to the circuit size.
- To reduce memory size & circuit size, the number of bits needs to be reduced accordingly.

Examples:

- If \( m = 4, f = 4 \), then \( 10.69 = (1010.1011)_2 = (10.6875)_{10} \) with error = 0.025.
- If \( m = 4, f = 2 \), then \( 10.69 = (1010.10)_2 = (10.5)_{10} \) with error = 0.19.
- If \( m = 3, f = 4 \), then \( 10.69 = (010.1011)_2 = (2.6875)_{10} \) and the result overflows.

1-bit adder

4-bit adder
Size of Number: Complexity vs. Performance

Transmitter (Tx)
sender / talker

Receiver (Rx)
listener

Transmitter

- PSDU gen
- Scrambler
- BCC Encoder
- OFDM sym split
- Interleave
- Mapper
- Subcar Alloca
- IFFT
- GI Insert
- Sym concat

Receiver

- BER?
- Viterbi Decod
- OFDM sym concat
- De-Interleave
- Demapper
- Pilot Data Extract
- FFT
- GI Remover
- OFDM sym split

✓ N = ceil(8xDAT_LEN+22): number of OFDM symbols
✓ Each OFDM symbol has 64 subcarriers (48 data subcarriers + 4 pilot subcarriers + 12 null subcarriers)
Sent 10,000 bits $\rightarrow$ 100 bit error

Sent 10,000 bits $\rightarrow$ 1 bit error
Contents

- Number System Introduction
  - Number systems used by human
  - Number systems used by computer

- Number System Conversion

- Signed Number Representation

- Fractional Number Representation
  - Fixed point number
  - Floating point number

- Size of Number: Complexity vs. Performance