ITM1010
Computer and Communication Technologies

Lecture #2
Part I: Introduction to Computer Technologies
Introduction to Numbering Systems
Decimal Numbering System

- The decimal numbering system is a positional-weighted numbering system. This means that each digital position has a specific weight (value). For example, 0.1, 1, 10, 100 all contain a 1, but each 1 has a different place value.

- The decimal system uses ten different basic symbols: 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. Each of these symbols is called a digit. The base is 10.

- The value of a decimal number is determined by positional weight, the sum-of-weights method.
The Sum-of-Weights Method

Example: the number 42,100.00

\[
\begin{align*}
10^4 & \quad 10^3 & \quad 10^2 & \quad 10^1 & \quad 10^0 & \quad 10^{-1} & \quad 10^{-2} \\
4 & \quad 2 & \quad 1 & \quad 0 & \quad 0 & \quad . & \quad 0 & \quad 0 \\
\end{align*}
\]

\[
(4 \times 10^4) + (2 \times 10^3) + (1 \times 10^2) + (0 \times 10^1) + (0 \times 10^0) + \ldots + (0 \times 10^{-1}) + (0 \times 10^{-2})
\]

\[
40,000 + 2000 + 100 + 0 + 0 + 0 + 0 + 0
\]

\[
= 42,100.00
\]
Why Binary Systems (Digital)

- **Information Integrity**: Better noise immunity

- **Information Manipulation**: The binary numbering system offers a simple means of representing and processing information in both hardware and software.

![Diagram of digital signal, noise, signal on tape, processor, and output signal]
Sources of Digital Information

- Analog signal
- Digital samples
- Encoded data
- Analog signal

ADC → CD recorder → CD player
# Binary Numbering System

## Example:

\[
1011.11_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}
\]

\[
= 8 + 0 + 2 + 1 + 0.5 + 0.25
\]

\[
= 11.75_{(10)}
\]
Binary Notation

Binary Digit is called BIT (Binary Digit)

Possible representations:

- 1 0
- high low
- true false
- 0 volt +5 volt

LSB – Least Significant Bit

Bit change with the least effect (on the right)

MSB – Most Significant Bit

Bit change with the most effect (on the left)
Binary & Decimal Conversion

- **Binary-to-Decimal Conversion**
  Sum-of-weights method. Example
  
  \[ 1011.11_{(2)} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 11.75_{(10)} \]

- **Decimal-to-Binary Conversion**
  Radix division for integer & Radix multiplication for fractional:
  
  Example: \( 92.875 \)
  
  \[ \begin{array}{cccccccc}
  1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
  1 & 2 & 5 & 11 & 23 & 46 & 92 \\
  \hline
  2 & 2 & 2 & 2 & 2 & 2 & 2
  \end{array} \]
  
  \[ 0.875 \times 2 = 1.75 \Rightarrow 1 \text{ MSB} \]
  
  \[ 0.75 \times 2 = 1.5 \Rightarrow 1 \]
  
  \[ 0.5 \times 2 = 1 \Rightarrow 1 \text{ LSB} \]
  
  \[ 92.875_{(10)} = 1011100.111_{(2)} \]
Class exercise

Convert the following decimal number to its binary equivalent.

12.375 (10)
Binary Arithmetic

- **Addition**: Rules: $0 + 0 = 0$, $1 + 0 = 1$, $1 + 1 = 10$
  
  $1101 + 1001 = 10110$, in decimal $\Rightarrow 13_{(10)} + 9_{(10)} = 22_{(10)}$

- **Subtraction**
  
  - Rules: $0 - 0 = 0$, $1 - 0 = 1$, $1 - 1 = 0$, & $10 - 1 = 1$
  
  $10110 - 1001 = 1101$, in decimal $\Rightarrow 22_{(10)} - 9_{(10)} = 13_{(10)}$

- **Multiplication**
  
  $1101 \times 101 = 1000001$, in decimal $\Rightarrow 13_{10} \times 5_{10} = 65_{10}$

- **Division**
  
  $110111 \div 101 = 1011$, in decimal $\Rightarrow 55_{10} \div 5_{10} = 11_{10}$
Complement representations

The process of subtraction can be accomplished by adding a negative number to a positive number.

Complement number representations are designed for this purpose.
- Ones complement
- Twos complement

Multiplication & division can be accomplished by repetitive addition and subtraction respectively.

Many computers perform all basic mathematical operations almost entirely with adder circuits.
Ones Complement

The 1s complement of a binary number of a binary number is derived by subtracting each bit in the number to be complemented from 1.

- e.g. 1s complement of 1100 is 0011

\[
\begin{array}{cccc}
1 & 1 & 1 & 1 \\
-1 & 1 & 0 & 0 \\
\hline
0 & 0 & 1 & 1 \quad (1s\ comp)
\end{array}
\]
Subtraction in 1s Complement

\[
\begin{align*}
14_{(10)} - 2_{(10)} &= 12_{(10)} \\
\text{Carry} &= 1 \\
\text{→ positive result} \\
7_{(10)} - 9_{(10)} &= -2_{(10)}
\end{align*}
\]

**Example 1:**

\[
\begin{align*}
&\quad \begin{array}{c}
0111 \\
-1001
\end{array} \\
\text{Step 1:} &\quad -1001 = 0110_{(1s\, comp)} \\
\text{Step 2:} &\quad \begin{array}{c}
\phantom{0}0111 \\
+0110
\end{array} \\
\text{Step 3:} &\quad 1101 \quad (\text{Difference in 1s complement form}) \\
&\quad 1101 = -0010
\end{align*}
\]

**Example 2:**

\[
\begin{align*}
&\quad \begin{array}{c}
1110 \\
-0010
\end{array} \\
\text{Step 1:} &\quad -0010 = 1101_{(1s\, comp)} \\
\text{Step 2:} &\quad \begin{array}{c}
\phantom{0}1110 \\
+1101_{(1s\, comp)}
\end{array} \\
\text{Step 3:} &\quad \begin{array}{c}
\phantom{0}1101 \\
\uparrow
\end{array} 1011 \\
\text{→ positive result}
\end{align*}
\]
2s Complement

Twos (2s) Complement = 1s complement + 1.

11_{(10)} - 5_{(10)} = 6_{(10)}

Carry = 1 → positive result

8_{(10)} - 12_{(10)} = -4_{(10)}
Class Exercise

Subtract the following binary number using 2s complement.

\[
\begin{array}{c}
0 & 1 & 0 & 1 \\
- & 0 & 1 & 0 & 0 \\
\hline
0 & 0 & 1 & 1 \\
\end{array}
\quad
\begin{array}{c}
0 & 0 & 1 & 1 \\
- & 0 & 1 & 0 & 1 \\
\hline
0 & 1 & 0 & 1 \\
\end{array}
\]
Sign Bit

A single bit, usually the leftmost bit, may be used to distinguish positive and negative numbers. The meaning of the sign bit can be fixed arbitrarily. But normally,

sign bit

0 - positive number
1 - negative number

e.g. \(-5_{(10)} = 1101\)

\(+5_{(10)} = 0101\)

– Note: the magnitude of a number is represented by the lower three bits
The leftmost bit still indicates sign.

- In two’s complement representation, two numbers can be added simply as two positive numbers e.g. $6 + (-2) = 4$
- Overflow occurs whenever the sum of two positive numbers yields a negative result or when two negative numbers are summed and the result is positive.