Computer Organization & Architecture

Chapter 9 Review – Integer Representation

Zhang Yang 张杨
cszyang@scut.edu.cn

Autumn 2025

Content of this lecture

- Signed Integers
- Signed-Magnitude Representation
- Signed Two's Complement Representation
- Converting between Different Bit Lengths
- Summary
- Exercise

Signed Integers

- An *n*-bit integer

  $B = b_{n-1} \ldots b_1 b_0$

  - Where $b_i = 0$ or 1 for $0 \le i \le n-1$
  - Represents an unsigned integer value $0 \sim 2^n-1$

    $V(B) = b_{n-1} * 2^{n-1} + \ldots + b_1 * 2^1 + b_0 * 2^0$

- Need to represent both positive and negative numbers
  - Signed-Magnitude representation
  - Signed One's Complement representation (not included here)
  - Signed Two's Complement representation

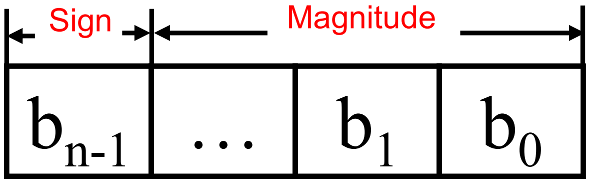
Signed-Magnitude Representation (1)

- For an *n*-bit integer
  - The sign part is a 1-bit value that is 0 for positive numbers, and 1 for negative numbers.
  - The magnitude part is an *(n-1)*-bit value that holds the absolute value of the number.

Signed-Magnitude Representation (2)

- An n-bit integer

  $$B = b_{n-1} \ldots b_1 b_0$$



  - $b_{n-1} = 0$, B is a positive number
  - $b_{n-1} = 1$, B is a negative number
- Example
  - $+18_{10} = \underline{0}0010010$
  - $-18_{10} = \underline{1}0010010$
  - $+0_{10} = \underline{0}0000000$
  - $-0_{10} = \underline{1}0000000$

Signed-Magnitude Representation (3)

- Representation Range
  - In general, if an n-bit sequence of binary digits $b_{n-1} \ldots b_1 b_0$ is interpreted as an signed integer B, its value is

$$V(B) = \begin{cases} \sum_{i=0}^{n-2} 2^i b_i & \text{if } b_{n-1} = 0 \\ -\sum_{i=0}^{n-2} 2^i b_i & \text{if } b_{n-1} = 1 \end{cases}$$

$$-(2^{n-1} - 1) \leq V(B) \leq 2^{n-1} - 1$$

Signed-Magnitude Representation (4)

- Drawbacks
  - Addition and subtraction require a consideration of both the signs of the numbers and their relative magnitudes to carry out the required operation.
  - There are two representations of 0 .
    - $+0 = \underline{0}0000000$
    - $-0 = \underline{1}0000000$

Signed Two's Complement Representation (1)

- For an *n*-bit integer
  - The sign part is a 1-bit value that is 0 for positive numbers, and 1 for negative numbers.
  - The magnitude part is an *(n-1)*-bit value.
    - Positive numbers: equivalent to the magnitude part of a signed-magnitude integer.
    - Negative numbers: represented as the bitwise complement of its absolute value +1.

Signed Two's Complement Representation (2)

- Example
  - +18 = 00010010
  - - 18 = 11101110
  - + 0 = 00000000
  - - 0 = 00000000

Signed Two's Complement Representation (3)

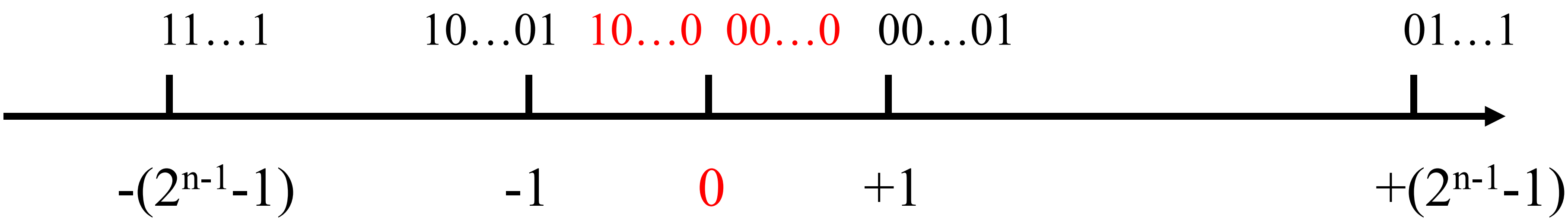
- Representation Range
  - The general case $B = b_{n-1} \ldots b_1 b_0$

$$
V(B) = \begin{cases} \sum_{i=0}^{n-2} 2^i b_i & B \geq 0 \\ -2^{n-1} b_{n-1} + \sum_{i=0}^{n-2} 2^i b_i & B < 0 \end{cases}
$$

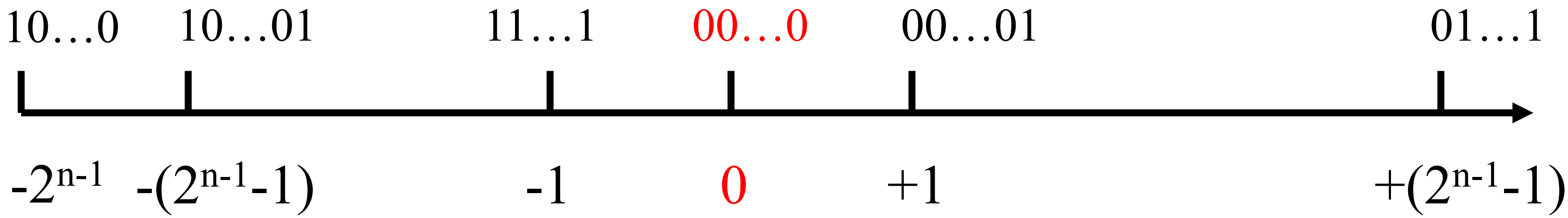
$= -2^{n-1} b_{n-1} + \sum_{i=0}^{n-2} 2^i b_i$ (for both positive and negative numbers)

$$
-2^{n-1} \leq V(B) \leq 2^{n-1} - 1
$$

Comparison



signed-magnitude



signed two’s complement

Conclusions

- In two systems, the leftmost bit is 0 for positive numbers and 1 for negative numbers.
- Positive values have identical representations in all systems, but negative values have different representations.
- The Sign and Magnitude system is the simplest representation, but it is also the most awkward for addition and subtraction operations.
- The 2’s-complement system is the most efficient method for performing addition and subtraction operations.

Converting between Different Bit Lengths (1)

- Sign and Magnitude Numbers
  - Move the sign bit to the new left-most position and fill in with zeros.
  - Example
    - ＋18 = <u>0</u>0010010 (8 bits)
    - ＋18 = <u>0</u>000000000010010 (16 bits)
    - －18 = <u>1</u>0010010 (8 bits)
    - －18 = <u>1</u>000000000010010 (16 bits)

Converting between Different Bit Lengths (2)

- Converting between Different Bit Lengths (ctd.)
  - Signed Two's Complement Numbers
    - Example
      - +18 = 00010010 (8 bits)
      - +18 = 0000000000010010 (16 bits)
      - −18 = 11101110 (8 bits)
      - ~~−18~~ = 1000000001101110 (16 bits)

      -32568
    - Sign Extension
      - Move the sign bit to the new left-most position and fill in with copies of the sign bit. For positive numbers, fill in with zeros, and for negative numbers, fill in with ones.

Converting between Different Bit Lengths (3)

- Example
  - －18 = 11101110 (8 bits)
  - －18 = 1111111111101110 (16 bits)

Summary (1)

- 知识点：Integer Representation
  - Signed-magnitude
  - Signed two’s complement
- 掌握程度
  - 给定一个真值（用正负号来分别表示正数、负数，这样的数称真值），正确转换出原码、补码。
  - 负数原码转换成真值
  - 负数补码转换成真值

Summary (2)

- 掌握程度
  - 给定一个真值和机器码制，会计算表示范围。
  - 负数原码和补码互相转换：符号不变，数值部分求反加1。

Exercise (1)

- 1. _______ is the most efficient method for performing addition and subtraction operations.
  - A. Signed-Magnitude Representation
  - B. Signed 1’s Complement Representation
  - C. <u>Signed 2’s Complement Representation</u>
  - D. None of the above

Exercise (2)

- 2. The range of an 8-bit signed 2's complement integer is ________.
  - A. [-256,+256]
  - B. [-256,+255]
  - C. [-128,+128]
  - D. [-128,+127]

Exercise (3)

- 3. Given the 8-bit binary number: 10011101. What decimal number does this represent if the computer uses signed-magnitude representation?
  - A. +29
  - B. - 29
  - C. +99
  - D. - 99

Exercise (4)

- 4. Given the 8-bit binary number: 10011101. What decimal number does this represent if the computer uses signed two's complement representation?
  - A. +29
  - B. - 29
  - C. +99
  - D. - 99