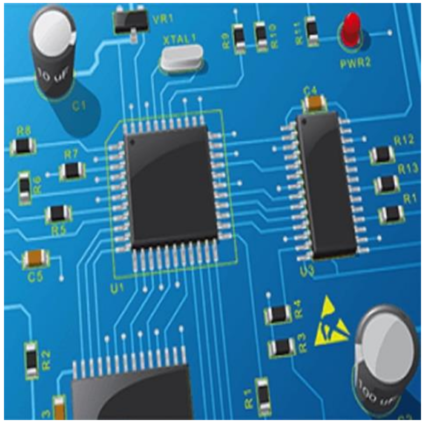
CSCI 341: Computer Organization

Spring 2026

Dr. Qi Han



Topics for Module 1

- Part 1: Course Logistics
- **Part 2: Review number representation**
  - ➔ Fixed point numbers
    - Canvas: number system conversion references
    - Textbook section 2.4
    - Worksheet on integer representation
  - Floating point numbers
    - Textbook Section 3.5 (up to “Floating Point Addition) and the Elaboration part at the end of the section
    - Worksheet on floating point number representation

Integer Representation

- **bits vs. numbers**
- **Different Bases**
  - Binary (base 2):
    - Example: 0000 0001 0010 0011 0100 0101
    - MSB, LSB
    - Odd vs. even numbers in binary
    - Numbers that are multiples of 4 in binary: end with 00
    - Add 0 to the end of of a number = multiplying the number by 2
  - Hexadecimal (base 16)
  - Decimal (base 10)

Signed Integers

- Two's Complement (2C) Representation
- MSB is sign bit
  - 1 for negative numbers
  - 0 for non-negative numbers
- Given an n-bit signed binary integer:
  - $X_{n-1}X_{n-2}\ldots X_1X_0$

$$x = -x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \cdots + x_1 2^1 + x_0 2^0$$

- Non-negative numbers have the same unsigned and 2s-complement representation

2C Signed Integers

- **Example**

  $1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1100_2$
  $= -1 \times 2^{31} + 1 \times 2^{30} + \ldots + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$
  $= -2{,}147{,}483{,}648 + 2{,}147{,}483{,}644 = -4_{10}$

- **Range: $[-2^{n-1}, 2^{n-1}-1]$**
  - $-(-2^{n-1})$ can't be represented
- **Some specific numbers**
  - 0: 0000 0000 … 0000
  - –1: 1111 1111 … 1111
  - smallest: 1000 0000 … 0000
  - largest: 0111 1111 … 1111

2C Signed Integers in RISC-V

RISC-V uses 32 bits to represent an integer

$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_{two} = 0_{ten}$
$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_{two} = +\ 1_{ten}$
$0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_{two} = +\ 2_{ten}$
...
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_{two} = +\ 2{,}147{,}483{,}646_{ten}$
$0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{two} = +\ 2{,}147{,}483{,}647_{ten}$ *maxint*
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000_{two} = -\ 2{,}147{,}483{,}648_{ten}$ *minint*
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0001_{two} = -\ 2{,}147{,}483{,}647_{ten}$
$1000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0010_{two} = -\ 2{,}147{,}483{,}646_{ten}$
...
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1101_{two} = -\ 3_{ten}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1110_{two} = -\ 2_{ten}$
$1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111_{two} = -\ 1_{ten}$

Range: $[-2^{31}, 2^{31}-1]$

Number: $(b^{31}x(-2^{31})) + (b^{30}x2^{30}) + \ldots + (b^{0}x2^{0})$

Signed Negation

- **If x is a binary number, how to get –x in binary?**
  - Complement (i.e., 1 → 0, 0 → 1) and add 1
- **Why?**

$$x + \overline{x} = 1111...111_2 = -1$$

$$\overline{x} + 1 = -x$$

- **Example**
  - 2_{10} = 0000 0000 … 0010_2
  - How to represent -2 in binary?
  - -2_{10} = 1111 1111 … 1101_2 + 1 = 1111 1111 … 1110_2

Sign Extension

- **The operation of increasing the number of bits of a binary number while preserving the number's sign (positive/negative) and value.**
  - Make copies of MSB and place that in the left of the word
  - Positive 2C numbers have an infinite number of 0s on the left
  - Negative 2C numbers have an infinite number of 1s on the left
- **Example 1**
  - Sign extend $0011\ 1100_2$ to 16 bits: $0000\ 0000\ 0011\ 1100_2$
  - Sign extend $1001\ 1001_2$ to 16 bits: $1111\ 1111\ 1001\ 1001_2$
- **Example 2**
  - $1011_{two} = -8 + 2 + 1 = -5$
  - $11011_{two} = -16+8+2+1 = -5$
  - $111011_{two} = -32+16+8+2+1 = -5$
- **The same way as leading zeroes do not affect values of a positive number, leading 1s do not affect values of a negative number**

Hexadecimal

- An easier way to express binary numbers
- The letters that stand for hexadecimal numbers above 9 can be upper or lower case – both are used
- Note: one hex digit

$0_{16} = 0_{10} = 0000_2$	$4_{16} = 4_{10} = 0100_2$	$8_{16} = 8_{10} = 1000_2$	$C_{16} = 12_{10} = 1100_2$
$1_{16} = 1_{10} = 0001_2$	$5_{16} = 5_{10} = 0101_2$	$9_{16} = 9_{10} = 1001_2$	$D_{16} = 13_{10} = 1101_2$
$2_{16} = 2_{10} = 0010_2$	$6_{16} = 6_{10} = 0110_2$	$A_{16} = 10_{10} = 1010_2$	$E_{16} = 14_{10} = 1110_2$
$3_{16} = 3_{10} = 0011_2$	$7_{16} = 7_{10} = 0111_2$	$B_{16} = 11_{10} = 1011_2$	$F_{16} = 15_{10} = 1111_2$

Number base conversion

- **Binary ↔ Hex**
  - Signs are already part of the number
- **Binary/Hex → Decimal**
  - Signed vs. unsigned
- **Decimal → Binary/Hex**
  - Signed vs. unsigned

Binary → Hexadecimal

- Separate into 4-bit groups, starting from the right

$$\text{Converting: } \underbrace{0111}_{7_{16}}\ \underbrace{1000}_{8_{16}}\ \underbrace{1010}_{A_{16}}\ \underbrace{0101}_{5_{16}}\ \underbrace{1010}_{A_{16}}\ \underbrace{1111}_{F_{16}}\ \underbrace{1011}_{B_{16}}\ \underbrace{1110}_{E_{16}}$$

- Or, 0111 1000 1010 0101 1010 1111 1011 1110

= 0x 78A5AFBE

- The “0x” prefix before a number signifies “hexadecimal.”

Hexadecimal → Binary

- **Convert each hex digit into an equivalent 4-bit binary number**
- **Example**

  0x2375 = 0010 0011 0111 0101

Unsigned Binary Integers → Decimal

- **Given an n-bit binary number: $X_{n-1}X_{n-2}\ldots X_1X_0$**
- **$X = X_{n-1} \times 2^{n-1} + X_{n-2} \times 2^{n-2} + \ldots X_1 \times 2^1 + X_0 \times 2^0$**

- **Example**

  $0000\ 0000 \ldots 1011_2$

  $= 0 + 0 + 1 \times 2^3 + 0 \times 2^2 + 1 x 2^1 + 1 x 2^0$

  $= 0 + 0 + \ldots + 8 + 0 + 2 + 1$

  $= 11_{10}$

- **Range: $[0, 2^n-1]$**
  - $1111\ldots1_{two} = 1+2^1+2^2+..+2^{n-1} = (1-2^n)/(1-2) = 2^n-1$

Review: Sum of Geometric Series

- **A geometric sequence/series: {a, ar, ar^2, ar^3, ... }**

where:

  - a is the first term, and
  - r is the factor between the terms (called the "common ratio")

- **Sum of a geometric series:** $a + ar + ar^2 + ... + ar^{(n-1)}$

$$\sum_{k=0}^{n-1}(ar^k) = a\left(\frac{1-r^n}{1-r}\right)$$

- **$1111...1_{two} = 1+2^1+2^2+..+2^{n-1} = (1-2^n)/(1-2) = 2^n-1$**

Signed Binary Integers → Decimal

- Given an n-bit **negative** binary number: $X_{n-1}X_{n-2}\ldots X_1X_0$
- Method 1
  - $X = - X_{n-1} \times 2^{n-1} + X_{n-2} \times 2^{n-2} + \ldots X_1 \times 2^1 + X_0 \times 2^0$
  - Note the negative sign in the first item above

- Method 2 (for negative binary X)
  - Use signed negation to get the absolute value (-X)
  - Convert the absolute value to decimal
  - Add the negative sign

Unsigned Hexadecimal $\rightarrow$ Decimal

- Given an n-digit hexadecimal number X: $X_{n-1}X_{n-2}\ldots X_1X_0$

$$X = X_{n-1} \times 16^{n-1} + X_{n-2} \times 16^{n-2} + \ldots X_1 \times 16^1 + X_0 \times 16^0$$

- Example

23_{16} or 0x23

$= 2 \times 16^1 + 3 \times 16^0$

$= 35_{10}$

Negative Hex → Decimal

- **Method 1 (the way the computer does it)**
  - Convert the original number to binary
  - Use signed negation to get the magnitude of the number
  - Convert the resulting number to decimal
  - Add a negative sign in front of the decimal
- **Method 2**
  - Convert the original number to decimal as if the number is unsigned
  - Subtract 16^n from the resulting number, where n is the number of hex digits used to represent the number
  - Given an n-digit hexadecimal number X: $X_{n-1}X_{n-2}\ldots X_1X_0$

$$X = X_{n-1} \times 16^{n-1} + X_{n-2} \times 16^{n-2} + \ldots X_1 \times 16^1 + X_0 \times 16^0 - 16^n$$

  - $0xC9 = 12 \times 16^1 + 9 \times 16^0 - 16^2 = -55_{10}$

Unsigned Integer Decimal → Binary

- **The Process : *Repeated Division by 2***
  a) Divide the *Decimal Number* by 2; the remainder is the LSB of *Binary Number* .
  b) If the quotation is zero, the conversion is complete; else repeat step (a) using the quotation as the Decimal Number. The new remainder is the next least significant bit of the *Binary Number.*

Example:

Convert the decimal number 6_{10} into its binary equivalent.

$$2\,\overline{)\,6}^{\;3} \quad r = 0 \leftarrow \text{Least Significant Bit}$$

$$2\,\overline{)\,3}^{\;1} \quad r = 1$$

$$2\,\overline{)\,1}^{\;0} \quad r = 1 \leftarrow \text{Most Significant Bit}$$

$$\boxed{\therefore\; 6_{10} = 110_2}$$

Unsigned Integer Decimal → Binary : Example #1

Example:

Convert the decimal number 26_{10} into its binary equivalent.

Solution:

$$2\overline{)26}^{\,13} \quad r = 0 \leftarrow \text{LSB}$$

$$2\overline{)13}^{\,6} \quad r = 1$$

$$2\overline{)6}^{\,3} \quad r = 0$$

$$2\overline{)3}^{\,1} \quad r = 1$$

$$2\overline{)1}^{\,0} \quad r = 1 \leftarrow \text{MSB}$$

$$\boxed{\therefore\ 26_{10} = 11010_2}$$

Unsigned Integer Decimal → Binary : Example #2

Example:

Convert the decimal number 41_{10} into its binary equivalent.

Solution:

$$2\,\overline{)\,41}^{\;20} \qquad r = 1 \leftarrow \text{LSB}$$

$$2\,\overline{)\,20}^{\;10} \qquad r = 0$$

$$2\,\overline{)\,10}^{\;5} \qquad r = 0$$

$$2\,\overline{)\,5}^{\;2} \qquad r = 1$$

$$2\,\overline{)\,2}^{\;1} \qquad r = 0$$

$$2\,\overline{)\,1}^{\;0} \qquad r = 1 \leftarrow \text{MSB}$$

$$\boxed{\therefore\ 41_{10} = 101001_2}$$

Negative Decimal → Binary

- **Convert the absolute value of the original number to binary**
- **Use signed negation to get the binary of the original number**

Unsigned Integer Decimal → Hexadecimal

- **The Process : *Repeated Division by 16***
  a) Divide the *Decimal Number* by 16; the remainder is the rightmost digit of *hexadecimal number* .
  b) If the quotation is zero, the conversion is complete; else repeat step (a) using the quotation as the Decimal Number. The new remainder is the next rightmost digit of the *hex number.*

Example:

Convert the decimal number 382_{10} into its hexadecimal equivalent.

382/16 = 23 R 14 (=E)
23/16 = 1 R 7
1/16 = 0 R 1

In reverse order, the hexadecimal number is 17E

Negative Decimal → Hex

1. **Convert the absolute value of the original number to binary**
2. **Find two's complement of that binary using signed negation**
3. **Convert the resulting binary to hex**
4. **Example: convert -45 to hex**
   1. $|-45| = 45 = 0010\ 1101_2$
   2. $0010\ 1101_2 \rightarrow 1101\ 0011_2$
   3. 0xD3

Fun Base Number Problems

- **Conversion between any pair of bases in which one base in a perfect power of the other**

- **Example: convert 2671_9 to base 3 without first converting to base 10**
  - $9 = 3^2$
  - Each digit in base 9 corresponds to two digits in base 3
  - 2 20 21 01

Self-check Exercises 1

1. Convert integer binary 11010001 to hex

2. Convert 8-bit unsigned binary 11001100 to decimal

3. Convert 8-bit unsigned number 0xCD to decimal and binary

4. Convert 65 to hexadecimal

5. Convert 34 to binary

6. What is the largest possible 16-bit unsigned binary number?

Keys to Self-check Exercises 1

1. **Convert integer binary 11010001 to hex**
   - 0x D1
2. **Convert 8-bit unsigned binary 11001100 to decimal**
   - 204
3. **Convert 8-bit unsigned number 0xCD to decimal and binary**
   - 205, 11001101
4. **Convert 65 to hexadecimal**
   - 0x41
5. **Convert 34 to binary**
   - 100010
6. **What is the largest possible 16-bit unsigned binary number?**
   - 1111111111111111 = 2^16 – 1 = 65535

Self-check Exercises 2

1. Convert the 8-bit, 2's complement 10110101 to decimal
2. Convert decimal $(-1)_{10}$ to 9-bit 2's complement binary
3. Convert 8-bit 2's complement binary 11001101 to decimal and hex
4. What is the largest possible 16-bit signed binary number in 2's complement?

Keys to Self-check Exercises 2

1. **Convert the 8-bit, 2's complement 10110101 to decimal**

   -75

2. **Convert decimal $(-1)_{10}$ to 9-bit 2's complement binary**

   111111111

3. **Convert 8-bit 2's complement binary 11001101 to decimal and hex**

   -51, 0xCD

4. **What is the largest possible 16-bit signed binary number in 2's complement?**

   0111111111111111 = 2^15 -1 = 32767