Positional Numbering System

Hui Chen [a]

[a]CUNY Brooklyn College, Brooklyn, NY, USA

September 6, 2023

Outline

Acknowledgement

The content of most slides come from the authors of the textbook:

Null, Linda, & Lobur, Julia (2018). The essentials of computer organization and architecture (5th ed.). Jones & Bartlett Learning.

Table of Contents

Lesson Objectives

Students are expected to be able to

1. *describe the fundamentals of numerical data representation and manipulation in digital computers*;
2. *convert between various radix systems*;
3. convert and perform arithmetic in signed integer representations;
4. explain how errors can occur in computations because of overflow and truncation;
5. express floating numbers in floating-point representation;
6. recognize the most popular character codes; and
7. describe the concepts of error detecting and correcting codes.

Table of Contents

Data: Bit

A bit is the most basic unit of information in a computer.

- It can be expressed as a state of "on" or "off" in a digital circuit,
- alternatively, "high" or "low" voltage, or
- ...

We use 0 and 1 to indicate these two states.

Data: Byte

A byte is a group of bits, almost no exception, a group of 8 bits.

- A byte is the smallest possible addressable unit of computer storage.
- The term "addressable" means that a particular byte can be retrieved according to its location in memory.

A group of four bits is called a nibble. Bytes, therefore, consist of two nibbles: a "high-order nibble" and a "low-order" nibble.

Data: Word

A word is a contiguous group of bytes.

- Words can be any number of bits or bytes.
- Word sizes of 16, 32, or 64 bits are most common.

Byte-addressable vs. word-addressable systems

- In a word-addressable system, a word is the smallest addressable unit of storage.

Table of Contents

Positional Numbering System

- Bytes store numbers using the position of each bit to represent a power of 2.
- The binary system is also called the base-2 system.
- Our decimal system is the base-10 system. It uses powers of 10 for each position in a number.
- Any integer quantity can be represented exactly using any base (or radix).

Examples of Decimal Number

Given a decimal number 3415, a number in powers of 10 is

$$3415 = 3 \times 10^3 + 4 \times 10^2 + 1 \times 10^1 + 5 \times 10^0 \quad (1)$$

Given a decimal number 5143.89, a number in power of 10 is

$$5143.89 = 5 \times 10^3 + 1 \times 10^2 + 4 \times 10^1 + 3 \times 10^0 + 8 \times 10^{-1} + 9 \times 10^{-2} \quad (2)$$

Which digit is the most significant digit, which the least significant?

Examples of Binary Number

Given a binary number 11011, a number in powers of 2 is

$$11011 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^1 + 1 \times 2^0 \tag{3}$$

Given a decimal number 11011.11, a number in power of 2 is

$$11011.11 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} \tag{4}$$

Which digit (bit) is the most significant digit (bit), which the least significant?

Table of Contents

Importance of Base-2 Number System

Binary numbers are the basis for all data representation in computers because digital circuits store binary numbers.

The knowledge of it helps understand operations of computer systems and instruction set architectures

Conversion Integers between Bases

Every integer value (whole numbers) can be represented exactly using any radix system.

Introduce two methods of radix conversion

- Subtraction method
- Division method

Subtraction Method: Base-10 to Base-2

Let's look at an example: convert the decimal number 190 to base 2:

$$
\begin{array}{r}
190 \\
-128 = 2^7 \times 1 \\
\hline
62 \\
-0 = 2^6 \times 0 \\
\hline
62 \\
-32 = 2^5 \times 1 \\
\hline
30 \\
-16 = 2^4 \times 1 \\
\hline
\end{array}
\qquad
\begin{array}{r}
14 \\
-8 = 2^3 \times 1 \\
\hline
6 \\
-4 = 2^2 \times 1 \\
\hline
2 \\
-2 = 2^1 \times 1 \\
\hline
0 \\
-0 = 2^0 \times 0 \\
\hline
0
\end{array}
$$

Subtraction Method: Base-10 to Base-2: Result

Let's look at an example: convert the decimal number 190 to base 2:

position	7	6	5	4	3	2	1	0
	1	0	1	1	1	1	1	0

Subtraction Method: Base-10 to Base-2: Steps

Let's look at an example: convert the decimal number 190 to base 2:

1. By a quick enumeration, we know that $128 = 2^7 < 190 < 2^8 = 256$, the result shall be less than (8+1) bits wide.
2. Do subtraction: $190 - 1 \times 2^7 = 190 - 128 = 62$, so we have a bit 1 at position 7
3. By another quick enumeration, we know that $32 = 2^5 < 64 < 2^6 = 64$, 62 shall be less than (6+1) bits wide.
4. Do subtraction: $62 - 2^5 = 62 - 32 = 30$, so we have a bit 1 at position 5.
5. Repeating the subtraction ...

$190_{10} = 10111110_2$

Subtraction Method: Base-10 to Base-3

Let's look at an example: convert the decimal number 190 to base 3:

$$
\begin{array}{l}
190 \\
-162 = 3^4 \times 2 \\
\hline
28 \\
-27 = 3^3 \times 1 \\
\hline
1 \\
-0 = 3^2 \times 0 \\
\hline
1 \\
-0 = 3^1 \times 0 \\
\hline
1 \\
1 = 3^0 \times 1 \\
\hline
\end{array}
$$

Subtraction Method: Base-10 to Base-3: Result

Let's look at an example: convert the decimal number 190 to base 3:

position	4	3	2	1	0
	2	1	0	0	1

Division Method: Base-10 to Base-2

Let's look at an example: convert the decimal number 190 to base 2:

The method is to compute the quotient and the remainder until the quotient becomes 0

Division Method: Base-10 to Base-2

```
2 |  190   0
  -----
 2 |  95   1
   ----
 2 |  47   1
   ----
 2 |  23   1
   ----
 2 |  11   1
   ----
  2 |  5   1
    ----
  2 |  2   0
    ----
  2 |  1   1
    ----
      0
```

Division Method: Base-10 to Base-3

Let's look at an example: convert the decimal number 190 to base 3:

```
3 | 190   1
  -----
3 |  63   0
  ----
3 |  21   0
  ----
3 |   7   1
  ----
3 |   2   2
  ----
      0
```

Converting Fractional Numbers

Fractional values can only be approximated in all base systems.

- Unlike integer values, fractions do not necessarily have exact representations under all radices.
- Example: the quantity $\frac{1}{4}$ is exactly representable in the binary and decimal systems, but is not in the ternary (base 3) numbering system.

Fractional Numbers

- Fractional decimal values have nonzero digits to the right of the decimal point.
- Fractional values of other radix systems have nonzero digits to the right of the radix point.
- Numerals to the right of a radix point represent negative powers of the radix

Fractional Numbers: Examples

$$0.47_{10} = 4 \times 10^{-1} + 7 \times 10^{-2} \tag{5}$$

$$0.11_2 = 1 \times 2^{-1} + 1 \times 2^{-2} \tag{6}$$

$$= \frac{1}{2} + \frac{1}{4} \tag{7}$$

$$= \frac{3}{4} = 0.75_{10} \tag{8}$$

Converting Fractional Numbers Between Bases

Like integers, there are two methods:

- Subtraction method
- Multiplication method

Converting Fractional Numbers: Subtraction Method

Identical to the subtraction method for whole numbers except

1. that we subtract negative powers of the radix instead, and
2. that we always start with the largest value first, n^{-1}, where n is our radix, and work our way along using larger negative exponents.

Subtraction Method: Base-10 to Base-2

Let's look at an example: convert the decimal number 0.8125 to base 2:

$$
\begin{array}{l}
0.8125 \\
-0.5000 = 2^{-1} \times 1 \\
\hline
0.3125 \\
-0.2500 = 2^{-2} \times 1 \\
\hline
0.0625 \\
-0.0000 = 2^{-3} \times 0 \\
\hline
0.0625 \\
-0.0625 = 2^{-4} \times 1 \\
\hline
0.0000
\end{array}
$$

The result:

position	binary point	1	2	3	4
	.	1	1	0	1

Converting Fractional Numbers: Multiplication Method

Repeatedly multiply the fraction by the radix (base).

- Ignoring the value in the units place at each step, continue multiplying each fractional part by the radix.
- Stop when the fractional part becomes 0, or
- stop when the desired accuracy is reached since the number can only be approximated.

Subtraction Method: Base-10 to Base-2

Let's look at an example: convert the decimal number 0.8125 to base 2:

$$
\begin{array}{r}
0.8125 \\
.8125 \\
\times 2 \\
\hline
1.6250 \\
.6250 \\
\times 2 \\
\hline
1.2500 \\
.2500 \\
\times 2 \\
\hline
0.5000 \\
.5000 \\
\times 2 \\
\hline
1.0000
\end{array}
$$

Subtraction Method: Base-10 to Base-2

The result:

position	binary point	1	2	3	4
	.	1	1	0	1

Converting Fractional Numbers

The two methods work with any base.

How about convert the decimal number 0.8125 to base 3?

Base 2, 8, and 16

- The binary numbering system is the most important radix system for digital computers.
- However, it is difficult to read long strings of binary numbers—and even a modestly-sized decimal number becomes a very long binary number.
  For example: $11010100011011_2 = 13595_10$
- For compactness and ease of reading, binary values are usually expressed using the hexadecimal, or base-16, numbering system.
- It is not as common, but base-8 or the octal numbering system is also used.

Hexadecimal Numbers

The hexadecimal numbering system uses the numerals 0 through 9 and the letters A through F.

$$12_{10} = C_{16}$$
$$26_{10} = 1A_{16}$$

Hexadecimal Numerals

Hexadecimal Numeral	Binary Value	Decimal Value
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Converting between Binary and Hexadecimal Numbers

- It is easy to convert between base 16 and base 2, because $16 = 2^4$.
- Thus, to convert from binary to hexadecimal, all we need to do is group the binary digits into groups of four.
- A group of four binary digits is called a hextet.
- If the number of bits is not a multiple of 4, pad on the left with zeros.

Converting between Binary and Hexadecimal Numbers: Example

Using groups of hextets, the binary number 11010100011011_2 $(= 13595_{10})$ in hexadecimal is:

```
0011 0101 0001 1011
   3    5    1    B
```

Octal Numbers

The octal numbering system (base 8) uses the numerals 0 through 7.

$$7_{10} = 7_8$$
$$9_{10} = 11_8$$

Octal was very useful when computers used six-bit words.

Octal Numerals

Octal Numeral	Binary Value	Decimal Value
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Converting between Binary and Octal Numbers

- It is easy to convert between base 8 and base 2, because $8 = 2^3$.
- Thus, to convert from binary to octal, all we need to do is group the binary digits into groups of three.
- A group of four binary digits is called an octet.
- If the number of bits is not a multiple of 3, pad on the left with zeros.

Converting between Binary and Octal Numbers: Example

Using groups of octets, the binary number 11010100011011_2 $(= 13595_{10})$ in octal is:

```
011 010 100 011 011
  3   2   4   3   3
```

Summary and Q&A

You are expected to be able to

1. *describe the fundamentals of numerical data representation and manipulation in digital computers*;
2. *convert between various radix systems*;

Any questions on:

- Bit and byte
- Positional Numbering System
- Converting between Bases
  - Converting Integers (Whole Numbers)
  - Converting Fractional Numbers
  - Hexadecimal and Octal Numbers