Computer Science Illuminated

SECOND EDITION

Nell Dale • John Lewis

Chapter 03

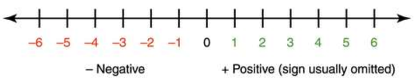
Data Representation I

Chapter Goals

- Representing negative values
- Representing fraction part
- Representing float

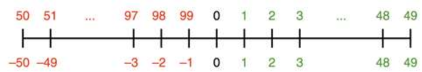
Representing Negative Values

- You have used the **signed-magnitude representation** of numbers since grade school. The sign represents the ordering, and the digits represent the magnitude of the number.



Representing Negative Values

- For example, if the maximum number of decimal digits we can represent is **two**, we can let 1 through 49 be the positive numbers 1 through 49 and let 50 through 99 represent the negative numbers -50 through -1.



Representing Negative Values

- To perform addition within this scheme, you just add the numbers together and discard any carry.

Sign-Magnitude	New Scheme
5 + – 6 – 1	5 + 94 99
– 4 + 6 2	96 + 6 2
– 2 + – 4 – 6	98 + 96 94

Representing Negative Values

- A-B=A+(-B). We can subtract one number from another by adding the negative of the second to the first.

Sign – Magnitude	New Scheme	Add Negative
–5 – 3 –8	95 – 3	95 + 97 92

Representing Negative Values

- Here is a formula that you can use to compute the negative representation

$$\text{Negative}(I) = 10^k - I, \text{ where } k \text{ is the number of digits}$$

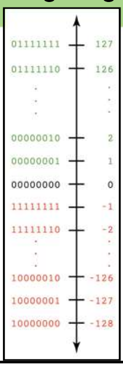
- This representation of negative numbers is called the **ten's complement**（补）.

Binary and Computers

- bit
  - A bit is the **basic unit of information** in computing and digital communications. A bit can have only one of two values, and may therefore be physically implemented with a two-state device. The most common representation of these values are 0 and 1.
- byte
  - 8 bits. (c types maybe int8_t, uint8_t, char)
- integer
  - A natural number, a negative number

Representing Negative Values

Two's Complement
To make it easier to look at long binary numbers, we make the number line vertical.



Negative (I) = $2^k - I$
where k the number of digits

A approach of two's complement

- Definition: ones' complement
  - Binary digit x,y satisfy x + y = 1. That x is ones' complement of y.
  - 1 is ones' complement of 0
  - 0 is ones' complement of 1
- two's complement

```
00010110  (y, equals decimal 22)

11101001  (ones' complement of y)
+      1
11101010  (the two's complement of y)
```


Representing Negative Values

- Addition and subtraction are accomplished the same way as in 10's complement arithmetic

```
-127   10000001
+ 1    00000001
-126   10000010
```

- Notice that with this representation, the leftmost bit in a negative number is always a 1.

Number Overflow（溢出）

- **Overflow** occurs when the value that we compute cannot fit into the number of bits we have allocated for the result. For example, if each value is stored using eight bits, adding 127 to 3 overflows.

```
  01111111
+ 00000011
  10000010
```

- Overflow is a classic example of the type of problems we encounter by mapping an infinite world onto a finite machine.

（计算机）如何判定溢出？

- 首先，你必须明白数的表示范围
  - 两位十进制数，[0..99]。如果补码表示负数，则表示[-50..49]
  - 8位二进制数，无符号整数表示[0..255]。有符号数[-128..127]
- 溢出就是超出了指定方法表示的数。
  - 例如 8位有符号数不能表示-129，怎么办？
- 二进制有符号数溢出的判定
  - 一个数不在 $[-2^{K-1}, 2^{K-1}-1]$范围内
  - 两个正数相加，结果是负数
  - 两个负数相加，结果是正数

3-13

History:溢出与阿丽亚娜五号

- 1996年6月4日，对于Ariane 5火箭的初次航行来说，这样一个错误产生了灾难性的后果。发射后仅仅37秒，火箭偏离它的飞行路径，解体并爆炸了。6亿美元付之一炬。
- 错误分析：
  - during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer. This resulted in an Operand Error.

3-14

课堂练习

Write out x, y, z values in binary codes
a)int8_t x = 111; int8_t y = -65; int8_t z = x + y;
b)int8_t x = -111; int8_t y = -65; int8_t z = x + y;
c)int8_t x = 130; int8_t y = -65; int8_t z = x + y;
d)int8_t x = 111; int8_t y = 65; int8_t z = x + y;
e)int8_t x = 0x3; int8_t y = 0x96; int8_t z = x + y;

3-15

Representing fraction part

- 公式：

$$d_n * R^{n-1} + d_{n-1} * R^{n-2} + \ldots + d_2 * R + d_1$$

- 移位操作：
  - 左移（<<）：X << k 表示 $X * R^k$
    - $(462)_{10}$ << 1 ＝ $(4620)_{10}$
    - $(101)_2$ << 2 ＝ $(10100)_2$
  - 右移（>>）：X >> k 表示 $X * R^{-k}$
    - $(462)_{10}$ >> 1 ＝ $(46\underline{2})_{10}$
    - $(101)_2$ >> 2 ＝ $(1\underline{01})_2$

小数部分

3-16

fraction part

- fraction part
  $(.d_1d_2\ldots d_k)_R$，R is base and k is position
- Positional notation formula for fraction part
  ？？？？

3-17

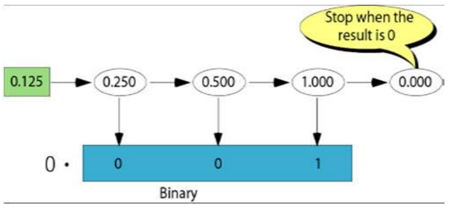
转换实数到二进制

- 计算：$(0.75)_{10} = (?)_2$
  - $(0.75)_{10} * 2 = (1.5)_{10} = (1)_2 + (0.5)_{10}$
  - $(1)_2 * 2 = 1 << 1 = (10)_2$
  - $(0.5)_{10} * 2 = (1)_{10} = (1)_2$；
  - $(11)_2 >> 2 = (11)_2 = (0.11)_2$

3-18

转换实数到二进制

- Converting the integer part
- Converting the fraction part



3-19

转换实数到二进制

- 2.55

2.55 1.1 0.2 0.4 0.8 1.6 1.2 0.4 ……

10. 1 0 0 0 1 1 0 ……

3-20

转换实数到二进制

- 用八进制将十进制小数快速转为二进制

0.15 1.2 1.6 4.8 6.4 3.2 ……

0. 1 1 4 6 3 ……

0. 001 001 100 110 011 ……

3-21

问题与思考

- float a = (float)0.15;
- float b = (float)0.45 / 3;

- 问 （a==b）成立吗？ 为什么？
- 如何判断 a 与 b 相等？

3-22

课堂练习

- $(0.125)_{10} = (?)_2$
- $(3.65)_{10} = (?)_2$
- $(2.3)_8 = (?)_{10}$
- $(2.3)_8 = (?)_2$
- $(1011.0101)_2=(?)_8$

3-23

Representing Real Numbers

- Real numbers have a whole part and a fractional part. For example 104.32, 0.999999, 357.0, and 3.14159.
  - the digits represent values according to their position, and
  - those position values are relative to the base.
- The positions to the right of the decimal point are the tenths position (10^{-1} or one tenth), the hundredths position (10^{-2} or one hundredth), etc.

3-24

Representing Real Numbers

- In binary, the same rules apply but the base value is 2. Since we are not working in base 10, the decimal point is referred to as a **radix point（小数点）**.
- The positions to the right of the radix point in binary are the halves position (2^{-1} or one half), the quarters position (2^{-2} or one quarter), etc.

3-25

Representing Real Numbers

- A real value in base 10 can be defined by the following formula.

$$sign * mantissa * 10^{exp}$$

- The representation is called **floating point** because the number of digits is fixed but the radix point floats.
- Mantissa（小数部分）

3-26

Representing Real Numbers

- **Scientific notation** A form of floating-point representation in which the decimal point is kept to the right of the leftmost digit.

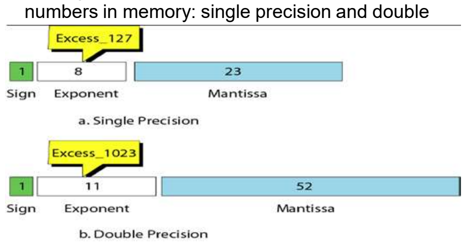
  For example, 12001.32708 would be written as 1.200132708E+4 in scientific notation.

- Likewise, a binary floating –point value is defined by the following formula:
  sign * mantissa * 2^{exp}

3-27

IEEE STANDARDS 754－2008

- The institute of electrical and electronics engineers (IEEE)has defined three standards for storing floating-point number; two are used to store numbers in memory: single precision and double



3-28

Example: Excess_127

- \+ 2^6 * 1.01000111001

- The sign is positive.
- The Excess_127 representation of the exponent is 133(127+6). In binary, this is 10000101
- The mantissa is 01000111001.You add extra 0s on the right to make it 23 bits.
- The number in memory is stored as:
  – 0 10000101 01000111001000000000000

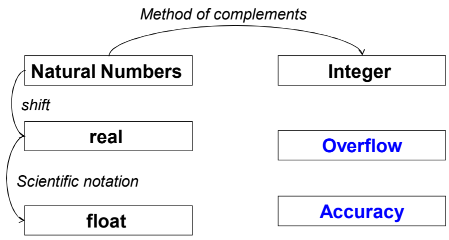
3-29

课堂练习

Write out variables in binary and hexadecimal codes
1) float x = 70;
2) float y = -1.125;

3-30

Summary



作业（part 1 of 2）

Write out variables x, y and z in binary code
1) int8_t x = 67; int8_t y = -7; int8_t z = y - x;
2) int8_t x = 0xd3;
3) uint8_t = 0xd3;
4) int8_t x = 127; int8_t y = -7; int8_t z = y – x;

5) float x = 1.125;
6) float x = 23.0;
7) float x = 0.45;

上述代码中，哪些出现溢出错误，哪些出现精度误差。

作业（part 2 of 2）

使用维基百科，解释以下概念。
1)Method of complements
2)Byte
3)Integer (computer science)
4)Floating point

仔细阅读" Method of complements"的内容，你将注意到nines' complement in the decimal 和 ones' complement in binary 等概念.
1)请证明：二进制的负数（two's complement of X）等于 X 的 ones' complement ＋ 1（即，X每位求反加1）
2)Int8_t x = - 017; 请用8进制描述变量 x。在c中017即$(017)_8$

阅读维基百科" Two's complement"的内容，特别是Sign extension小节内容。
1)C程序：int8_t x = -0x1f; int y = x; 请用16进制描述变量 x 和 y，并说明 int y = x 的计算过程。
2)请用数学证明，为什么可以这么计算。

阅读维基百科" Floating point"的内容，
1) NaN 是什么？