

第二章 数据的机器表示

2. 简单回答下列问题。(参考答案略)

- (1) 为什么计算机内部采用二进制表示信息？既然计算机内部所有信息都用二进制表示，为什么还要用到十六进制或八进制数？
- (2) 常用的定点数编码方式有哪几种？通常它们各自用来表示什么？
- (3) 为什么计算机中大多用补码表示带符号整数？
- (4) 在浮点数的基和位数一定的情况下，浮点数的表数范围和表数精度分别由什么决定？两者如何相互制约？
- (5) 为什么要对浮点数进行规格化？有哪两种规格化操作？
- (6) 为什么有些计算机中除了用二进制外还用 BCD 码来表示数值数据？
- (7) 为什么计算机处理汉字时会涉及到不同的编码（如，输入码、内码、字模码）？说明这些编码中哪些是用二进制编码，哪些不是用二进制编码，为什么？

4. 假定机器数为 8 位（1 位符号，7 位数值），写出下列各二进制数的原码和补码表示。

+0.1001, -0.1001, +1.0, -1.0, +0.010100, -0.010100, +0, -0

参考答案：

	原码	补码
+0.1001:	0.1001000	0.1001000
-0.1001:	1.1001000	1.0111000
+1.0:	溢出	溢出
-1.0:	溢出	1.0000000
+0.010100:	0.0101000	0.0101000
-0.010100:	1.0101000	1.1011000
+0:	0.0000000	0.0000000
-0:	1.0000000	0.0000000

5. 假定机器数为 8 位（1 位符号，7 位数值），写出下列各二进制数的补码和移码表示。

+1001, -1001, +1, -1, +10100, -10100, +0, -0

参考答案：

	移码	补码
+1001:	10001001	00001001
-1001:	01110111	11110111
+1:	10000001	00000001
-1:	01111111	11111111
+10100:	10010100	00010100
-10100:	01101100	11101100
+0:	10000000	00000000
-0:	10000000	00000000

6. 已知 $[x]_{\text{补}}$ ，求 x

- ① $[x]_{\text{补}} = 1.1100111$ ② $[x]_{\text{补}} = 10000000$
 ③ $[x]_{\text{补}} = 0.1010010$ ④ $[x]_{\text{补}} = 11010011$

参考答案：

- ① $[x]_{\text{补}} = 1.1100111$ $x = -0.0011001B$
 ② $[x]_{\text{补}} = 10000000$ $x = -10000000B = -128$
 ③ $[x]_{\text{补}} = 0.1010010$ $x = +0.101001B$
 ④ $[x]_{\text{补}} = 11010011$ $x = -0101101B$

7. 假定一台 32 位字长的机器中带符号整数用补码表示，寄存器 R1 和 R2 的内容分别为：R1：0000108BH，R2：8080108BH。不同指令对寄存器进行不同的操作，因而，不同指令执行时寄存器内容的真值不同。

请问：执行下列指令时，寄存器 R1 和 R2 的内容所对应的真值分别为多少？

- ① 无符号数加法指令
 ② 带符号整数乘法指令
 ③ 单精度浮点减法指令（IEEE754）

参考答案：

- | | R1: 0000108BH | R2: 8080108BH |
|---|--|--|
| ① | +108BH | +8080108BH |
| ② | +108B | -7F7FEF75H |
| ③ | 0000 0000 0000 0000 0001 0000 1000 1011
+0.002116H $\times 2^{-126}$ | 1000 0000 1000 0000 0001 0000 1000 1011
-1.002116H $\times 2^{-126}$ |

8. 假定机器 M 的字长为 32 位，用补码表示带符号数。下表第一列给出了在机器 M 上执行的关系表达式，请完成表中后面三栏内容的填写。

关系表达式	运算类型	结果	说明
-2147483648 == 2147483648U	无符号数	1	$10\dots = 10\dots 0$
-2147483648 < -2147483647	有符号数	1	$10\dots 00 (-2^{31}) < 10\dots 01 (-(2^{31}-1))$
-2147483648 < 2147483647U	无符号数	0*	$10\dots 0 (2^{31}) > 01\dots 1 (2^{31}-1)$
-2147483648 < 2147483647	有符号数	1	$10\dots 00 (-2^{31}) < 01\dots 1 (2^{31}-1)$
(unsigned)-2147483648 < -2147483647	无符号数	1	$10\dots 00(2^{31}) < 10\dots 01 (2^{31}+1)$
(unsigned)-2147483648 < 2147483647	无符号数	0*	$100\dots 0(2^{31}) > 01\dots 1 (2^{31}-1)$

9. 以下是一个 C 语言程序，用来计算一个数组 a 中每个元素的和。当参数 len 为 0 时，返回值 sum 应该是 0，但是在机器上执行时，却发生了存储器访问异常。请问是什么原因造成的，并说明程序应该如何修改。

```

1  floae sum_elements(float a[], unsigned len)
2  {
3      int i;
4      float result = 0;
5
6      for (i = 0; i <= len-1; i++)
7          result += a[i];

```

```

8      return result;
9  }

```

参考答案：

参数 `len` 的类型是 `unsigned`，所以，当 `len=0` 时，执行 `len-1` 的结果为 `11...1`，是最大可表示的无符号数，因而，任何无符号数都比它小，使得循环体被不断执行，引起数组元素的访问越界，发生存储器访问异常。

只要将 `len` 声明为 `int` 型，或循环的测试条件改为 `i<len`。

11. 某机器字长为 16 位，下列几种情况所能表示的数的范围是什么？

- ① 无符号整数
- ② 原码定点小数
- ③ 补码定点小数
- ④ 补码定点整数
- ⑤ 下述格式的浮点数（基为 2，移码的偏置常数为 128）

参考答案：

- ① 无符号整数： $0 - 2^{16} - 1$
- ② 原码定点小数： $-(1-2^{-15}) - +(1-2^{-15})$
- ③ 补码定点小数： $-1 - +(1-2^{-15})$
- ④ 补码定点整数： $-32768 - +32767$
- ⑤ 浮点数： 负数： $-(1-2^{-7}) \times 2^{+127} - 2^{-7} \times 2^{-128}$ 即： -2^{-135}
 正数： $+2^{-135} - (1-2^{-7}) \times 2^{+127}$

13. 假定一个变量的值为 4098，分别用 32 位 2-补码和 IEEE754 单精度浮点格式表示该变量（结果用十六进制表示），并说明哪段二进制序列在两种表示中完全相同，为什么一定相同。

参考答案：

$4098 = +1\ 0000\ 0000\ 0010B = +1.0000\ 0000\ 001 \times 2^{12}$

32 位 2-补码形式为：0000 0000 0000 0000 0001 0000 0000 0010 （00001002H）

IEEE754 单精度格式为：0 10001011 0000 0000 0010 0000 0000 000 （45801000H）

红色部分为相同序列，因为不管数整数还是小数，其有效数字应该一致！

14. 假定一个变量的值为 -2147483647，分别用 32 位 2-补码和 IEEE754 单精度浮点格式表示该变量（结果用十六进制表示），并说明哪种表示的值完全精确，哪种表示的是近似值。

参考答案：

$-2147483647 = -111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111B$

$= -1.11\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111 \times 2^{30}$

32 位 2-补码形式为：1000 0000 0000 0000 0000 0000 0000 0001 （80000001H）

IEEE754 单精度格式为：1 10011101 1111 1111 1111 1111 1111 111 （CEFFFFFFH）

32 位 2-补码形式能表示精确的值，而浮点数表示的是近似值，低位被截断

17. 已知下列字符编码：A=100 0001，a=110 0001，0=011 0000，求 E、e、f、7、G、Z、5 的 7 位 ASCII 码和第一位前加入奇校验位后的 8 位编码。（略）
18. 假定在一个程序中定义了变量 x、y 和 i，其中，x 和 y 是 float 型变量（用 IEEE754 单精度浮点数表示），i 是 16 位 short 型变量（用补码表示）。程序执行到某一时刻，x=-0.125、y=7.5、i=100，它们都被写到了主存（按字节编址），其地址分别是 100，108 和 112。请分别画出在大端机和小端机上变量 x、y 和 i 在内存的存放情况。

参考答案：

$$-0.125 = -0.001\text{B} = -1.0 \times 2^{-3}$$

x 在机器内部的机器数为：1 01111100 00...0 (BE000000H)

$$7.5 = +111.1\text{B} = +1.111 \times 2^2$$

y 在机器内部的机器数为：0 10000001 11100...0 (40F00000H)

$$100 = 64 + 32 + 4 = 1100100\text{B}$$

i 在机器内部表示的机器数为：0000 0000 0110 0100 (0064H)

	大端机	小端机
地址	内容	内容
100	BEH	00H
101	00H	00H
102	00H	00H
103	00H	BEH
108	40H	00H
109	F0H	00H
110	00H	F0H
111	00H	40H
112	00H	64H
113	64H	00H

19. 写出 16 位数据字的 SEC 码。假定数据字为 0101 0001 0100 0110，说明 SEC 码如何正确检测数据位 5 的错误。

参考答案：

码字排列如下：

M16 M15 M14 M13 M12 P5 M11 M10 M9 M8 M7 M6 M5 P4 M4 M3 M2 P3 M1 P2 P1

得到如下的分组校验码：

$$P1 = M1 \oplus M2 \oplus M3 \oplus M4 \oplus M5 \oplus M7 \oplus M9 \oplus M11 \oplus M12 \oplus M14 \oplus M16$$

$$P2 = M1 \oplus M3 \oplus M4 \oplus M6 \oplus M7 \oplus M10 \oplus M11 \oplus M13 \oplus M14$$

$$P3 = M2 \oplus M3 \oplus M4 \oplus M8 \oplus M9 \oplus M10 \oplus M11 \oplus M15 \oplus M16$$

$$P4 = M5 \oplus M6 \oplus M7 \oplus M8 \oplus M9 \oplus M10 \oplus M11$$

$$P5 = M12 \oplus M13 \oplus M14 \oplus M15 \oplus M16$$

当数据字为 0101 0001 0100 0110 时，得到

$$P5P4P3P2P1=00011$$

第五位数据出错时，数据字变为：0101 0001 0101 0110，得到：

$$P5'P4'P3'P2'P1'=01010$$

所以， 00011 和 01010 按位异或，得到故障字为：01001，说明码字第 9 位出错，即 M5 错。

20. 假设要传送的数据信息为：100011，若约定的生成多项式为： $G(x)=x^3+1$ ，则校验码为 111。

假定在接收端接收到的数据信息为 100010，说明如何正确检测其错误，写出检测过程。

参考答案：用 100010 111 除以 1001，得到余数为 001，不为 0，说明有错。