

221900175-毛九骏-第二章

2. 简单回答下列问题。
- (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,为什么还要学习十六进制表示？
- (2) 常用的定点数编码方式有几种？通常它们各自用来表示什么信息？
- (3) 为什么现代计算机中大多用补码表示带符号整数？
- (4) 在浮点数的基数和总位数一定的情况下,浮点数的表示范围和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) 为什么计算机处理汉字时会涉及不同的编码(如输入码、内码、字模码)？说明这些编码中哪些用二进制编码,哪些不用二进制编码。为什么？

- (1) 原因： 1. 制造两个稳定态的物理器件较为容易；
2. 二进制编码、计数、运算逻辑简单；
3. 与逻辑命题相对应，便于逻辑运算，方便使用逻辑电路实现算术运算。

学习十六进制原因：

- 1.既是二进制的简便表示又便于阅读和书写；
2. 与二进制对应简单转换容易
3. 可以显示的缩短长度

(2)

定点数编码方式	表示的信息
原码表示	计算机中使用原码表示定点小数
补码表示	带符号整数
变形补码	判断运算中是否有溢出
反码表示	计算机中很少使用
移码表示	表示“阶”，一般表示浮点数的阶

- (3) 为什么使用补码：
- 实现带符号整数的加减计算的统一

(4) 浮点数的基数和总位数一定的情况下

	由什么决定
范围	偏移常量 & 阶码的位数
精度	尾数 的位数

制约：由于基数和总位数一定，

1. 精度越大，所需尾数的位数越多，阶码的位数越少，范围越窄；
2. 范围越宽，所需的阶码位数越多，尾码的位数越少，精度越小。

(5)

为什么要规格化	尽量多的得到有效数位，使浮点数有唯一表示
两种规格化的操作	左规 (尾数左移,阶减一) & 右规 (尾数右移,阶加一)

(6)

为什么处理汉字会涉及不同编码	
输入码（外码）	不使用二进制

	是使用键盘上的某些键组合对应一个汉字
内码	如GB2312 采用二进制
字模码	使用二进制点阵记录后 转为二进制编码

3. 实现下列各数的转换。

$$(1) (25.8125)_{10} = (?)_2 = (?)_8 = (?)_{16}$$

$$(2) (101101.011)_2 = (?)_{10} = (?)_8 = (?)_{16}$$

$$(3) (4E.C)_{16} = (?)_{10} = (?)_2$$

$$\begin{aligned} (1) \quad 25.8125 &= (32 - 1) - 6 + 0.5 + 0.25 + 0.0625 \\ &= 1 \ 1111B - 00110B + 0.1B + 0.01B + 0.0001B \\ &= (1 \ 1001.1101)_2 \\ &= (31.64)_8 \\ &= (19.D)_{16} \end{aligned}$$

$$\begin{aligned} (2) \quad 10 \ 1101.011_2 &= (55.3)_8 \\ &= (2D.6)_{16} \\ &= 32 + 13 + 0.25 + 0.125 \\ &= (45.375)_{10} \end{aligned}$$

$$\begin{aligned} (3) \quad (4E.C)_{16} &= 64 + 14 + 12 * (1/16) \\ &= (78.75)_{10} \\ &= (0100 \ 1110.11)_2 \end{aligned}$$

5. 假定机器数为 8 位(1 位符号,7 位数值),写出下列各二进制整数的补码和移码(偏置常数为 128)表示。

$$+1001, -1001, +1, -1, +10100, -10100, +0, -0 \quad 128 = 1000 \ 0000$$

二进制数	补码	移码
+1001	0000 1001	1000 1001
-1001	1111 0111	0111 0111
+1	0000 0001	1000 0001
-1	1111 1111	0111 1111
+1 0100	0001 0100	1001 0100
-1 0100	1110 1100	0110 1100
+0	0000 0000	1000 0000
-0	0000 0000	1000 0000

6. 已知 $[x]_{\text{补}}$, 求 x 。

$$(1) [x]_{\text{补}} = 1.1100111$$

$$(2) [x]_{\text{补}} = 10000000$$

$$(3) [x]_{\text{补}} = 0.1010010$$

$$(4) [x]_{\text{补}} = 11010011$$

$$(1) x = (-0.0011001)_2$$

$$(2) x = (-1000\ 0000)_2 = -128$$

$$(3) x = (0.101001)_2$$

$$(4) x = (-10\ 1101)_2 = -45$$

7. 假定一台 32 位字长的机器中带符号整数用补码表示,浮点数用 IEEE 754 标准表示,寄存器 R1 和 R2 的内容分别为 R1: 0000 108BH, R2: 8080 108BH。不同指令对寄存器内容进行不同的操作,因而,不同指令执行时寄存器内容对应的真值不同。假定执行下列运算指令时,操作数为寄存器 R1 和 R2 的内容,则 R1 和 R2 中操作数的真值分别为多少?

- (1) 无符号整数加法指令。
- (2) 带符号整数乘法指令。
- (3) 单精度浮点数减法指令。

	R_1	R_2
无符号整数加法	$11 + 8 \times 16 + 1 \times 16^3$ $= 4235 = 108B\ H$	$11 + 8 \times 16 + 16^3 + 8 \times 16^5 + 8 \times 16^7$ $= 21\ 5587\ 6491 = 8080\ 108B\ H$
带符号整数乘法	$4235 = 108B\ H$	$-(7F7F\ EF74\ H)$
单精度浮点数减法	$2^{-137} * (1 + 2^{-5} + 2^{-9} + 2^{-12} + 2^{-13})$	$-2^{-126} - 2^{-137} * (1 + 2^{-5} + 2^{-9} + 2^{-12} + 2^{-13})$

$$\begin{aligned} &0\ 0000\ 0000\ 000\ 0000\ 0001\ 0000\ 1000\ 101 \\ &\rightarrow 2^{-126} * 0.000\ 0000\ 0001\ 0000\ 1000\ 1011 = 2^{-137} * 1.0000\ 1000\ 1011 = \\ &2^{-137} * (1 + 2^{-5} + 2^{-9} + 2^{-12} + 2^{-13}) \end{aligned}$$

$$\begin{aligned} &1\ 0000\ 0001\ 000\ 0000\ 0001\ 0000\ 1000\ 101 \\ &\rightarrow -2^{1-127} * 1.000\ 0000\ 0001\ 0000\ 1000\ 1011 = -2^{-126} - 2^{-137} * 1.0000\ 1000\ 1011 = \\ &-2^{-126} - 2^{-137} * (1 + 2^{-5} + 2^{-9} + 2^{-12} + 2^{-13}) \end{aligned}$$

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。请问这是是什么原因造成的,并说明程序应该如何修改。

```
1 float 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
9     return result;
10 }
```

函数形参中 len 的类型为 `unsigned` 即 无符号整数,使用原码表示。

当 $len = 0$ 时, $len - 1 = 11 \dots 1111$ 也就是最大的无符号整数,任何无符号整数类型的数都比其小,

又由于 i 一直为 `int` 类型,即使强制转换为无符号数, i 也将一直小于 $len - 1$

故而 循环体一直进行,故而存储器访问异常

10. 设某浮点数格式为

数符	阶码	尾数
1位	5位移码	6位补码数值部分

其中,移码的偏置常数为 16,补码采用一位符号位,基数为 4。

(1) 用这种格式表示下列十进制数: $+1.75$, $+19$, $-1/8$ 。

(2) 写出该格式浮点数的表示范围,并与 12 位定点补码整数和定点补码小数表示范围比较。

(1)

$$(1.75)_{10} = (1.11)_2 = (0.0111)_2 \times 4^1 \quad 010001011100$$

(2) 写出该格式浮点数的表示范围,并与 12 位定点补码整数和定点补码小数表示范围比较。

(1)

$$\begin{aligned}
 (1.75)_{10} &= (1.11)_2 = (0.111)_2 \times 4^1 & 0 \ 1000 \ 1 \ 0 \ 11 \ 10 \ 0 \\
 (19)_{10} &= (10011)_2 = (0.10011)_2 \times 4^3 & 0 \ 10011 \ 0 \ 10011 \\
 (-1/8)_{10} &= (-0.001)_2 = (-0.1)_2 \times 4^{-1} & 1 \ 01111 \ 100 \ 000
 \end{aligned}$$

(2)

	值
最大正数M+	$(0.111 \ 111)_2 \times 4^{15} = 0.984375 \times 4^{15} \approx 2^{30}$
最小正数m+	$(0.000 \ 001)_2 \times 4^{-16} = 2^{-6} \times 4^{-16} = 2^{-38}$
0	0
最大负数M-	-最小正数 $= -2^{-38}$
最小负数m-	$-(1.000 \ 000)_2 \times 4^{15} = -1.000 \times 4^{15} = -2^{30}$

范围为: 从 $m_- \sim M_+$ 不连续

名称	范围
12位 定点补码整数	$-2^{11} \sim 2^{11} - 1$ 的整数
12位 定点补码小数	$-1.0 \sim 1 - 2^{-11}$ 不连续

12. 以 IEEE 754 单精度浮点数格式表示下列十进制数。

$+1.75, +19, -1/8, 258$

13. 设一个变量的值为 4098, 要求分别用 32 位补码整数和 IEEE 754 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, 为什么会相同?

$$\begin{aligned}
 1.75 &= (1.11)_2 = 1.11 \times 2^{127-127} \rightarrow 0 \ 0111 \ 1111 \ 110 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \\
 &= 3FE0 \ 0000 \ H
 \end{aligned}$$

$$\begin{aligned}
 19 &= (1 \ 0011)_2 = 1.0011 \times 2^{131-127} \rightarrow 0 \ 1000 \ 0011 \ 001 \ 1000 \ 0000 \ 0000 \ 0000 \ 0000 \\
 &= 4198 \ 0000 \ H
 \end{aligned}$$

$$\begin{aligned}
 -\frac{1}{8} &= -2^{-3} = -1 \times 2^{127-3-127} \rightarrow 1 \ 0111 \ 1100 \ 000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \\
 &= BE00 \ 0000 \ H
 \end{aligned}$$

$$\begin{aligned}
 258 &= (1 \ 0000 \ 0010)_2 = (1.0000 \ 0010)_2 \times 2^{8+127-127} = 0 \ 1000 \ 0111 \ 000 \ 0001 \ 0000 \ 0000 \ 0000 \ 0000 \\
 &= 4381 \ 0000 \ H
 \end{aligned}$$

$$\begin{aligned}
 4098 &= 2^{12} + 2 \\
 &= (\text{int})(0000 \ 0000 \ 0000 \ 0000 \ 0001 \ 0000 \ 0000 \ 0010) = (\text{int})(00001002 \ H) \\
 &= (1.0 + 2^{-11}) \times 2^{127+12-127} \\
 &= (\text{float}) \ 0 \ 1000 \ 1011 \ 000 \ 0000 \ 0001 \ 0000 \ 0000 \ 0000 \\
 &= (\text{float}) 4580 \ 1000 \ H
 \end{aligned}$$

黄色片段是一致的,

相同的原因: 都是除去隐藏位以外的有效部分

17. 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 IEEE 754 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器和小端机器上变量

17. 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 IEEE 754 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器和小端机器上变量 x 、 y 和 i 在内存的存放位置。

$$\begin{aligned}
 x &= -0.125 = \text{BE00 0000 H} \\
 y &= 7.5 = (111.1)_2 = 1.111 \text{ B} * 2^{127+2-127} = 0\ 1000\ 0001\ 111\ 0000\ 0000\ 0000\ 0000\ 0000 \\
 &= 40\text{F0 0000 H} \\
 i &= 64 + 32 + 4 = 0000\ 0000\ 0110\ 0100 \\
 &= 0064 \text{ H}
 \end{aligned}$$

地址	大端	小端
100	BE H	00 H
101	00 H	00 H
102	00 H	00 H
103	00 H	BE H
104	/	/
105	/	/
106	/	/
107	/	/
108	40 H	00 H
109	F0 H	00 H
110	00 H	F0 H
111	00 H	40 H
112	00 H	64 H
113	64 H	00 H
114	/	/
115	/	/
...