**第三章作业**

1. 假定某数采用IEEE 754单精度浮点数格式表示为C820 0000H，则该数的值是（ C ）。

A. (–1.01)10×217 B. (–1.01) 10×2144

C. (–1.25) 10×217 D. (–1.25) 10×2144

1. 假定变量i、f的数据类型分别是int、float。已知i=12345，f=1.2345e3，则在一个32位机器中执行下列表达式时，结果为“假”的是（ C ）。

A. i= =(int)(float)i B. i= =(int)(double)i

C. f= =(float)(int)f D. f= =(float)(double)f

1. 某计算机字长为8位，其CPU中有一个8位加法器。已知带符号整数x=–69，y=–38，现要在该加法器中完成x+y的运算，则该加法器的两个输入端信息和输入的低位进位信息分别为( DN )。

A. 1011 1011、1101 1010、0 B. 1011 1011、1101 1010、1

C. 1011 1011、0010 0101、0 D. 1011 1011、0010 0101、1

1. 某8位计算机中，假定x和y是两个带符号整数变量，用补码表示，x=63，y= –31，则x+y的机器数及其相应的溢出标志OF分别是( B )。

A. 1FH、0 B. 20H、0 C. 1FH、1 D. 20H、

1. 若两个float型变量（用IEEE 754单精度浮点格式表示）x和y的机器数分别表示为x=40E8 0000H，y=C204 0000H，则在计算x+y时，第一步对阶操作的结果[ΔE]补为( B )。

A. 0000 0111 B. 0000 0011 C. 1111 1011 D. 1111 1101

1. 某字长为8位的计算机中，x和y为无符号整数，已知x=68，y=80，x和y分别存放在寄存器A和B中。请回答下列问题（要求最终用十六进制表示二进制序列）。
2. 寄存器A和B中的内容分别是什么？

A: 44H B: 50H

1. 若x和y相加后的结果存放在寄存器C中，则寄存器C中的内容是什么？运算结果是否正确？加法器最高位的进位Cout是什么？零标志ZF和进位标志CF各是什么？

C: 94H

正确

Cout = 0

ZF = 0

CF = 0

1. 若x和y相减后的结果存放在寄存器D中，则寄存器D中的内容是什么？运算结果是否正确？加法器最高位的进位Cout是什么？零标志ZF和借位标志CF各是什么？

D: F4H

错误

Cout = 0

ZF = 0

CF = 1

1. 无符号整数加/减运算时，加法器最高位进位Cout的含义是什么？它与进/借位标志CF的关系是什么？

Cout表示是否进位或借位，以此判断结果是否溢出

加法时CF = Cout, 减法时 CF = ~ Cout (即Cout取反)

1. 无符号整数一般用来表示什么信息？为什么通常不对无符号整数的运算结果判断溢出？

表示浮点数的阶码，对一些事物的编号，地址信息

无符号整数结果溢出相当于取模运算，溢出即表示循环，没有必要判断溢出，同时这样能节省成本

1. 考虑以下C语言程序代码：

int func1 (unsigned word)

{

return (int) (( word <<24) >> 24);

}

int func2 (unsigned word)

{

return ( (int) word <<24 ) >> 24;

}

假设在一个32位机器上执行这些函数，sizeof (int)=4，。说明函数func1和func2的功能，并填写下表。

func1: 取传入参数的后8位，并将其以int值输出，即求传入参数后8位表示的无符号整数值

func2: 取传入参数的后8位，并将其符号扩展成32位，即求传入参数后8位表示的整数值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| W | | func1(w) | | func2(w) | |
| 机器数 | 值 | 机器数 | 值 | 机器数 | 值 |
| 0000 007FH | 127 | 0000 007FH | 127 | 0000 007FH | 127 |
| 0000 0080H | 128 | 0000 0080H | 128 | FFFF FF80H | -128 |
| 0000 00FFH | 255 | 0000 00FFH | 255 | FFFF FFFFH | -1 |
| 0000 0100H | 256 | 0000 0000H | 0 | 0000 0000H | 0 |

1. 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= –130、y=7.25、i=130，它们都被写到了主存（按字节编址），其地址分别是&x，&y和&i。请分别给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。

x: C301 0000H y: 40E8 0000H i: 0000 0082H

大端机器：x: C3 01 00 00 y: 40 E8 00 00 i: 00 00 00 82

小端机器：x: 00 00 01 C3 y: 00 00 E8 40 i: 82 00 00 00

1. 以下是函数fpower2的C语言源程序，它用于计算2*x*的浮点数表示，其中调用了函数u2f，u2f用于将一个无符号整数表示的0/1序列作为float类型返回。请填写fpower2函数中的空白部分，以使其能正确计算结果。

1 float fpower2(int x)

2 {

3 unsigned exp, frac, u;

4

5 if (x< -149 ) { /\* 值太小，返回0.0 \*/

6 exp = 0 ;

7 frac = 0 ;

8 } else if (x< -126 ) { /\* 返回非规格化结果 \*/

9 exp = 0 ;

10 frac = 1<<(x+150) ;

11 } else if (x< 128 ) { /\* 返回规格化结果 \*/

12 exp = x+127 ;

13 frac = 0 ;

14 } else { /\* 值太大，返回+∞ \*/

15 exp = 255 ;

16 frac = 0 ;

17 }

18 u = exp << 23 | frac;

19 return u2f(u);

1. }

10. 采用IEEE 754单精度浮点数格式计算下列表达式的值。

（1）0.75+(– 65.25） （2）0.75–(– 65.25）

(1) 0.75 = 1.1 \* 2-1 -65.25 = - 1.00000101 \* 26

对阶：0.75 = 0.00000011 \* 26

求和：-1.0000001 \* 26

所以和为 C281 0000H

(2)0.75 – (- 65.25) = 0.75 + 65.25

0.75 = 1.1 \* 2-1 65.25 = 1.00000101 \* 26

对阶：0.75 = 0.00000011 \* 26

求和：1.0000100 \* 26

所以和为： 8284 0000H