

计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第10次课：习题课

杜国栋

信息科学与工程学院计算机科学与工程系

gddu@ysu.edu.cn



燕山大学
YANSHAN UNIVERSITY



课程目标

- 复习带符号数的二进制数据的表示方法;
- 复习定点数加减法运算;
- 复习定点数原码一位乘、补码一位乘和原码两位乘;
- 复习浮点数的加减法运算;





码值为FFH

若为真值127，则为 [填空1] 码；若为真值-127，则为 [填空2] 码；

若表示-1，则为 [填空3] 码；若表示-0，则为 [填空4] 码；





码值为FFH

若为真值127，则为（ ）码；若为真值-127，则为（ ）码；
若表示-1，则为（ ）码；若表示-0，则为（ ）码；

FFH 1111 1111 移码 原码 补码 反码





将下列十进制数表示成浮点规格化数，阶码3位用补码表示，尾数9位用补码表示。

(1) $27/64$

(2) $-27/64$





例 6: 将下列 十进制数表示成浮点规格化数, 阶码 3 位, 用补码表示, 尾数 9 位, 用补码表示。

(1) $27/64$ (2) $-27/64$

解: $27/64 = 11011 \times 2^{-6} = 0.11011 \times 2^{-1}$, 阶码为: 111 尾数为: 110110000

$-27/64 = -0.11011 \times 2^{-1}$, -0.11011 的补码为: 1.00101, 阶码为: 111 尾数为 001010000





写出下列数据规格化浮点数的编码

(1) +111000 (1位符号位, 阶码5位移码, 尾数10位补码)

(2) -10101 (1位符号位, 阶码5位移码, 尾数10位补码)





例 7: 写出下列数据规格化浮点数的编码

(1) +111000 (设 1 位符号位, 阶码 5 位移码, 尾数 10 位补码)

(2) -10101 (设 1 位符号位, 阶码 5 位移码, 尾数 10 位补码)

解: (1) $+111000 = 0.111000 \times 2^6$, 6 的移码为 10110

尾数的补码为 1110000000, 因此编码为: 0101101110000000

(2) $-10101 = -0.10101 \times 2^5$, 5 的移码为 10101

尾数的补码为 0101100000, 因此编码为: 1101010101100000





某机器字长32位，采用定点小数表示，符号位1位，尾数31位，则可表示的最大正小数为（ ），最小负小数为（ ）

A. $+(2^{31}-1)$

B. $-(1-2^{-32})$

C. $+(1-2^{-31}) \approx +1$

D. $-(1-2^{-31}) \approx -1$



例 8: 某机器字长 32 位, 采用定点小数表示, 符号位 1 位, 尾数 31 位, 则可表示的最大正小数为 (), 最小负小数为 ()

A $+(2^{31}-1)$ B $-(1-2^{-32})$ C $+(1-2^{-31}) \approx +1$ D $-(1-2^{-31}) \approx -1$

解: 最大正小数为: $+(1-2^{-31})$

最小负小数为: $-(1-2^{-31})$





- 有一个字长为32位的浮点数，符号位1位，阶码8位，用移码表示；位数23位，用补码表示。请写出：

(1) 非规格化数能表示的最大正数、最小正数、最大负数和最小负数

(2) 规格化数能表示的最大正数、最小正数、最大负数和最小负数

(1) 非规格化数: $[x = (-1)^s * (0.M) * 2^E]$

1	最大正数 (原/补码)	0 11111111 111...1	真值为 $+(1-2^{-23}) * (2^{127})$
2			
3	最小正数 (原/补码)	0 00000000 000...1	真值为 $+(2^{-23}) * (2^{128}) = +2^{-151}$
4			
5	最大负数 (原码)	1 00000000 000...1	真值为 $-(2^{-23}) * (2^{128}) = -2^{-151}$
6	最大负数 (补码)	1 00000000 111...1	
7			
8	最小负数 (原码)	1 11111111 111...1	真值为 $-(1-2^{-23}) * (2^{127})$
9	最小负数 (补码)	1 11111111 000...0	真值为 -2^{127}

(2) 规格化数:

1	最大正数 (原/补码)	0 11111111 111...1	真值为 $+(1-2^{-23}) * 2^{127}$
2			
3	最小正数 (原/补码)	0 00000000 100...0	真值为 $+(2^{-1}) * (2^{128}) = 2^{129}$
4			
5	最大负数 (补码)	1 00000000 011...1	真值为 $-(2^{-23} + 2^{-1}) * (2^{128})$
6	最大负数 (原码)	1 00000000 100...1	
7			
8	最小负数 (补码)	1 11111111 000...0	真值为 -2^{127}
9	最小负数 (原码)	1 11111111 111...1	真值为 $-(1-2^{-23}) * 2^{127}$





某机器字长32位，采用定点整数表示，符号位为1位，尾数31位，则可表示的最大正整数为()，最小负整数为()

A. $+(2^{31}-1)$

B. $-(1-2^{-32})$

C. $+(2^{-30}-1)$

D. $-(2^{-31}-1)$





例 9: 某机器字长 32 位, 采用定点整数表示, 符号位为 1 位, 尾数 31 位, 则可表示的最大正整数为, 最小负整数为: ↵

A $+(2^{31}-1)$ B $-(1-2^{-32})$ C $+(2^{-30}-1)$ D $-(2^{-31}-1)$ ↵

解: 最大正整数为: $+(2^{31}-1)$ ↵

最小负整数为: $-(2^{31}-1)$ ↵





32位浮点数格式中，符号位为1位，阶码8位，尾数23位，则它所能表示的最大规格化正数为（ ），最小负数为（ ）

A. $+(1-2^{-23}) \times 2^{127}$

B. $-(1-2^{-23}) \times 2^{127}$

C. $+(2^{-23}-1) \times 2^{127}$

D. $-(2^{-23}-1) \times 2^{127}$





例 9：32 位浮点数格式中，符号位为 1 位，阶码 8 位，尾数 23 位，则它所能表示的最大规格化正数为（），最小负数为（）

解：最大规格化正数为： $+(1-2^{-23}) \times 2^{127}$

最小负数为： $-(1-2^{-23}) \times 2^{127}$





已知 x 和 y ，用变形补码计算 $x-y$ 和 $x+y$ ，并指出结果是否溢出。

$$(1) \quad x=27/32 \quad y=31/32$$

$$(2) \quad x = 13/16 \quad y=-11/16$$





例 1: 已知 x 和 y , 用变形补码计算 $x-y$ 和 $x+y$, 并指出结果是否溢出。

$$(1) x=27/32 \quad y=31/32 \quad (2) x=13/16 \quad y=-11/16$$

解: (1) $x = 27/32 = 11011 * 2^{-5} = 0.11011$

$$y = 31/32 = 11111 * 2^{-5} = 0.11111$$

$$[x]_{\text{补}} = 0.11011 \quad [y]_{\text{补}} = 0.11111 \quad [-y]_{\text{补}} = 1.00001$$

$$[x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}} = 00.11011 + 00.11111 = 01.11010$$

双符号为 01, 表示产生溢出

$$[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}} = 00.11011 + 11.00001 = 11.11100$$

双符号位为 11, 无溢出

$$(2) x = 13/16 = 1101 * 2^{-4} = 0.1101$$

$$y = -11/16 = -1011 * 2^{-4} = -0.1011$$

$$[x]_{\text{补}} = 0.1101 \quad [y]_{\text{补}} = 1.0101 \quad [-y]_{\text{补}} = 0.1011$$

$$[x+y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}} = 00.1101 + 11.0101 = 00.0010$$

双符号为 00, 无溢出

$$[x-y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}} = 00.1101 + 00.1011 = 01.1000$$

双符号位为 01, 溢出





将十进制整数 $x=18$, $y=15$ 转为6位移码, 其中1位为符号位, 5位为数值位, 求 $x-y$ 和 $x+y$ 的移码, 并指出结果是否溢出?





例 2: 将下列十进制整数转为 6 位移码, 其中 1 位为符号位, 5 位为数值位, 求 $x-y$ 和 $x+y$ 的移码, 并指出结果是否溢出? ↵

(1) $x=18$ $y=15$ ↵

解: (1) $x=18$ $[x]_{\text{原}}=010010$ $[x]_{\text{移}}=110010$ ↵

$y=15$ $[y]_{\text{原}}=001111$ $[y]_{\text{补}}=001111$ $[-y]_{\text{补}}=110001$ $[y]_{\text{移}}=101111$ ↵

$[x-y]_{\text{移}}=[x]_{\text{移}}+[-y]_{\text{补}}=1110010+1110001=1100011$ 未溢出 ↵

$[x+y]_{\text{移}}=[x]_{\text{移}}+[y]_{\text{补}}=1110010+0001111=0000001$ 未溢出 ↵



$$x=2^{01} * 0.1101 \quad y=2^{11} * (-0.1011), \quad \text{求 } x+y$$





例 3: $x=2^{01}*0.1101$ $y=2^{11}*(-0.1011)$, 求 $x+y$

解: 对阶 $\Delta E = [E_x]_{\text{补}} + [-E_y]_{\text{补}} = 0001 + 1101 = 1110 < 0$ $\Delta E = -2$

$$X = 2^{11} * 0.001101$$

尾数相加 $[M_x]_{\text{补}} + [M_y]_{\text{补}} = 00.001101 + 11.0101 = 11.100001$

规格化 左规一位 $[M_x]_{\text{补}} + [M_y]_{\text{补}} = 11.00001$ $E = 0011 + 1111 = 0010$

$$x+y = 2^{10} * (-0.11111)$$





已知 $[x]_{\text{补}} = 1.1011000$ $[y]_{\text{补}} = 1.1011000$ 使用变形补码计算 $2[x]_{\text{补}} + 1/2[y]_{\text{补}}$ 并判断结果有无溢出。





例 4: 已知 $[x]_{\text{补}}=1.1011000$ $[y]_{\text{补}}=1.1011000$ 使用变形补码计算 $2[x]_{\text{补}}+1/2[y]_{\text{补}}$ 并计算有无溢出。

解: $[x]_{\text{补}}=1.1011000$ $[y]_{\text{补}}=1.1011000$

$$2[x]_{\text{补}}=1.011000 \quad 1/2[y]_{\text{补}}=1.1101100$$

$$2[x]_{\text{补}}+1/2[y]_{\text{补}}=1.011000+1.1101100=11.0011100 \quad \text{无溢出}$$





$2[x]_{\text{补}} = 1.0101001$, $1/2[y]_{\text{原}} = 1.01011000$, 用变形补码计算 $[x]_{\text{补}} + [y]_{\text{补}}$ 并判断结果是否有溢出





例 5: $2[x]_{\text{补}} = 1.0101001$ $1/2[y]_{\text{原}} = 1.01011000$, 用变形补码计算 $[x]_{\text{补}} + [y]_{\text{补}}$ 并判断是否有溢出。

解: $2[x]_{\text{补}} = 1.0101001$ $[x]_{\text{补}} = 1.10101001$

$1/2[y]_{\text{原}} = 1.01011000$ $[y]_{\text{原}} = 1.10110000$ $[y]_{\text{补}} = 1.01010000$

$[x]_{\text{补}} + [y]_{\text{补}} = 11.10101001$

$+ 11.01010000$

$= 11.11111001$ 无溢出



假定4个整数用8位补码分别表示 $Y1=FEH$,
 $F2=F2H$, $F3=90H$, $F4=F8H$ 。若将运算结果放在
一个8位寄存器中, 则下列运算会发生溢出的是 ()

A $Y1*Y2$

B $Y2*Y3$

C $Y1*Y4$

D $Y2*Y4$





例 6: 假定 4 个整数用 8 位补码分别表示 $Y1=FEH$, $F2=F2H$, $F3=90H$, $F4=F8H$ 。
若将运算结果放在一个 8 位寄存器中, 则下列运算会发生溢出的是 ()

A $Y1*Y2$ B $Y2*Y3$ C $Y1*Y4$ D $Y2*Y4$

解: 8 位补码表示真值的范围为 **-128~127**

$Y1=FEH$ 补码 11111110 原码 10000010 真值 -2

$F2=F2H$ 补码 11110010 原码 10001110 真值 -14

$F3=90H$ 补码 10010000 原码 11110000 真值 -112

$F4=F8H$ 补码 11111000 原码 10001000 真值 -8

$Y1 * Y2 = (-2) * (-14) = 28$, 没有溢出。

$Y2 * Y3 = (-14) * (-112) = 1568 > 127$, 溢出了。

$Y1 * Y4 = (-2) * (-8) = 16$, 没有溢出。

$Y2 * Y4 = (-14) * (-8) = 112$, 没有溢出。





设浮点数的阶码和尾数均采用补码表示，
且尾数分别为5位和7位（均含2位符号
位），若有两个数 $X = 2^7 * 29/32$ ， $Y =$
 $2^5 * 5/8$ ，则用浮点加法计算 $X + Y =$





例 7: 设浮点数的阶码和尾数均采用补码表示, 且尾数分别为 5 位和 7 位 (均含 2 位符号位), 若有两个数 $X = 2^7 * 29/32$, $Y = 2^5 * 5/8$, 则用浮点加法计算 $X+Y=$

解: X 的浮点数格式为: 00,111; 00,11101 (分号前为阶码, 分号为尾数)

Y 的浮点格式为: 00,101; 00,10100

对阶: 阶码 111 比 101 大, 所以将 Y 的阶码加 2 变成 111, 尾数右移两位 00101

尾数相加: $00,11101 + 00,00101 = 01,00010$ 结果符号位 01, 所以需要右移

规格化 将尾数右移 1 位, 变成 10001, 阶码加 1 (从 00,111 变成 01,000)

所以 $X+Y = 01,000; 00,10001$

判断是否溢出: 阶码符号位为 01, 发生溢出



有问题欢迎随时跟我讨论

办公地点：西校区信息馆423

邮 箱：gddu@ysu.edu.cn

