课时2 定点/浮点数的表示与运算

考点	重要程度	占分	题型
1 定点数的表示与运算	★★★	0 ~ 3	选择、填空
2 浮点数的表示与运算	必考	6 ~ 10	选择、填空

2.1 定点数的表示与运算

1.定点数的表示

一.无符号数和有符号数的表示

在计算机中参与运算的机器数有两大类：无符号数和有符号数。

1）**无符号数.** 全部二进制位均为数值位，没有符号位，相当于数的绝对值。

若机器字长为8位，则数的表示范围为0～28-1，即0～255。

2）**有符号数.** 在机器中，数的“正”“负”号是无法识别的，有符号数用“0”表示“正”号，用“1”表示“负”号，二进制数的最高位为符号位.

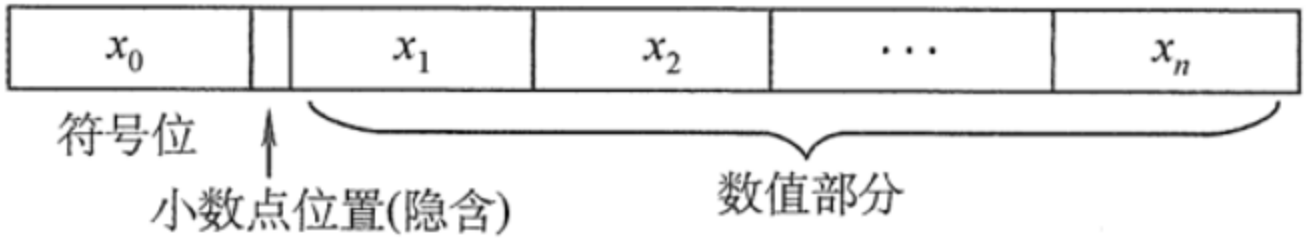
$[X]_{原}$ 表示原码，$[X]_{补}$ 表示补码，$[X]_{反}$ 表示反码，$[X]_{移}$ 表示移码。

1.定点数的表示

二.机器数的定点表示

1）定点小数

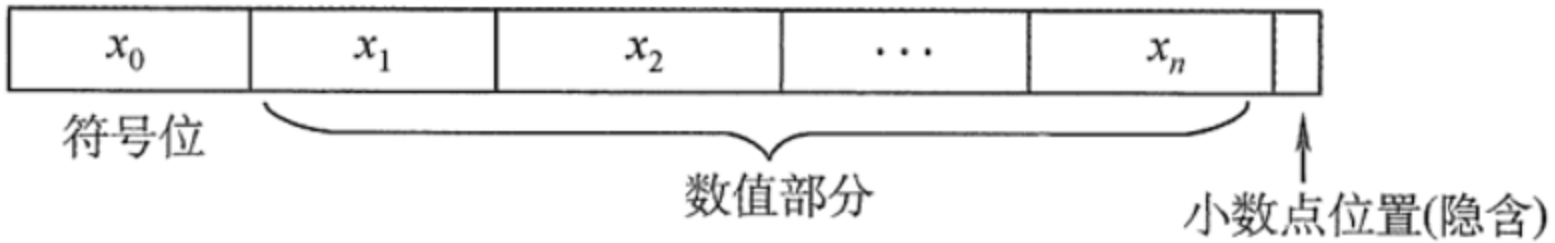
定点小数是纯小数，约定小数点位置在符号位之后、有效数值部分最高位之前



定点小数的格式

2）定点整数

定点整数是纯整数，约定小数点位置在有效数值部分最低位之后。



定点整数的格式

1.定点数的表示

三、原码、补码、反码、移码

（1）原码表示法:

用机器数的最高位表示该数的符号，其余的各位表示数的绝对值。

纯小数的原码定义

若X_1= +0.1101，X_2= -0.1101，字长为8位，则$[x_1]_{原}=0.1101000$，$[x_2]_{原}=1.1101000$.

若字长为n + 1，则原码小数的表示范围为 $-(1-2^{-n})\le X\le 1-2^{-n}$ (关于原点对称)

若X_1= +1110，X_2= -1110，字长为8位，则$[x_1]_{原}=0.0001110$，$[x_2]_{原}=1.0001110$.

若字长为n + 1，则原码整数的表示范围为 $-(1-2^{-n})\le X\le 1-2^{-n}$ (关于原点对称)

注意：真值零的原码表示有正零和负零两种形式，即$[+0]_{原}$= 00000和$[-0]_{原}$= 10000。

1.定点数的表示

(2)补码表示法

正数：与原码相同

负数：符号位不变，数值位取反加1

若X_1= +0.1001，X_2= -0.0110，字长为8位,则$[x_1]_{补}=0.1001000$ $[x_2]_{补}$1.1010000。

若字长为n + 1，则补码的表示范围为-1≤x≤1-2-n（比原码多表示-1）。

纯整数的补码定义

若X_1= +1010，X_2= -1101，字长为8位，则$[x_1]_{补}$= 0,0001010，$[x_2]_{补}$= 1,1110011。

若字长为n + 1，则补码的表示范围为-2n≤x≤2n-1（比原码多表示-2n）。

注意：真值零的补码表示是唯一的。即$[+0]_{补}=[-0]_{补}=0.0000$

1.定点数的表示

（3）反码表示法

正数：与原码相同

负数：符号位不变，数值位取反

反码通常用来作为由原码求补码或由补码求原码的中间过渡。

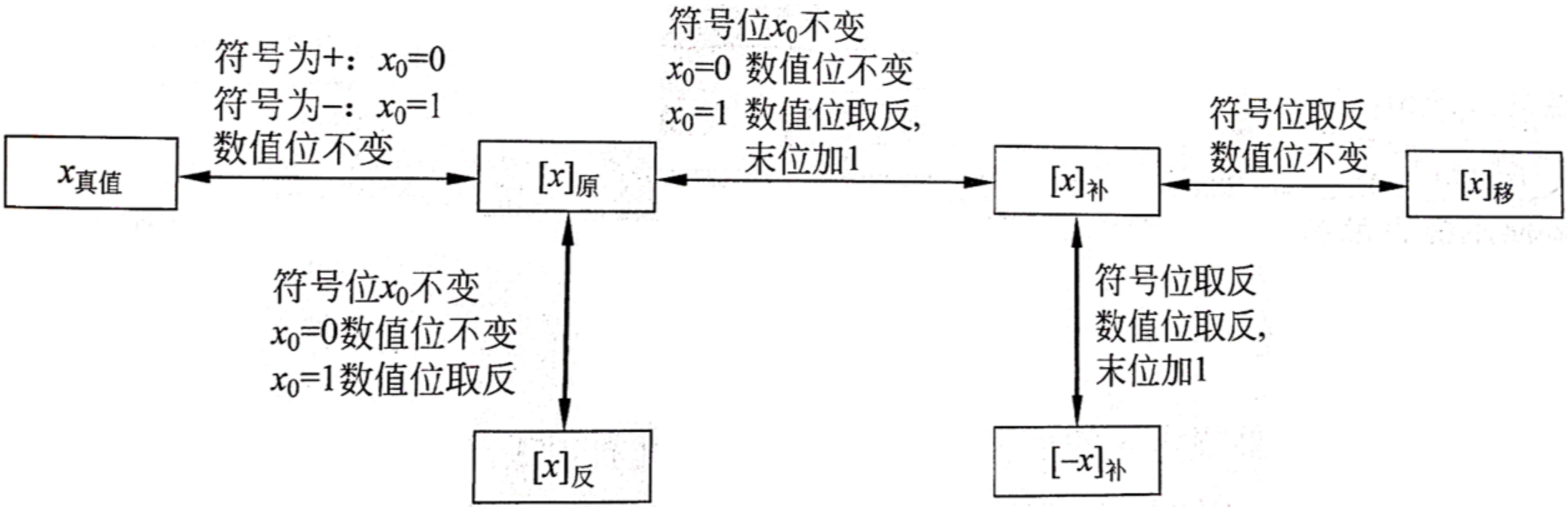
若X_1= +0.0110，X_2= -0.0110，字长为8位，则$[x_1]_{反}$= 0.0110000，$[x_2]_{反}$ = 1.1001111。

若字长为n + 1，则反码的表示范围为$-(1-2^{-n}) \le x \le 1-2^{-n}$（关于原点对称）。

注意：真值零的反码表示不唯一 $[+0]_{反} = 0.0000$；$[-0]_{反}$= 1.1111。

1.定点数的表示

真值、原码、补码、反码的转换规律，如图所示.



1.定点数的表示

（4）移码表示法

移码常用来表示浮点数的阶码。它只能表示整数。

移码就是在真值X上加上一个常数（偏置值），通常这个常数取2^n相当于X在数轴上向正方向偏移了若干单位。移码定义为

$$[x]_{移}=2^n+x(2^n>x\geq -2^n，\text{其中机器字长为}n+1)$$

若正数X_1= +10101，X_2= -10101，字长为8位，则

$[x_1]_{移}=2^7+10101=1,0010101$；$[x_2]_{移}=2^7+(-10101)=0,1101011$；

1.定点数的表示

移码具有以下特点：

(1)移码中零表示唯一，$[+0]_{移}=2^n+0=[-0]_{移}=2^n-0=100\cdots0$（n个"0"）

(2)移码全0时，对应真值的最小值-2^n；移码全1时，对应真值的最大值2^n-1

(3)移码保持了数据原有的大小顺序，移码大真值就大，移码小真值就小。

2.定点数的运算

一.定点数的移位运算

(1) 算术移位的对象是有符号数，在移位过程中符号位保持不变。

(2) 逻辑移位将操作数视为无符号数，不管是左移还是右移，都添0。

不同机器数算术位移后的空位添补规则

	码制	添补代码
正数	原码、补码、反码	0
负数	原码	0
	补码	左移添0
		右移添1
	反码	1

2.定点数的运算

二.补码定点数加减法运算

补码加减运算的特点如下（设机器字长为n + 1）。

1）参与运算的两个操作数均用补码表示

2）按二进制运算规则运算，逢二进一。

3）符号位与数值位按同样规则一起参与运算，符号位运算产生的进位要丢掉，结果的符号位由运算得出。

2.定点数的运算

4）补码加减运算依据下面的公式进行

$$[A+B]_{补}=[A]_{补}+[B]_{补}$$

$$[A-B]_{补}=[A]_{补}+[-B]_{补}$$

即，若做加法，则两数的补码直接相加；

若做减法，则将被减数与减数的机器负数相加。

5）补码运算的结果亦为补码。

【题1】 设机器字长为8位（含1位符号位），A = 15，B = 24，求 $[A+B]_{补}$ 和 $[A-B]_{补}$

解 由 A=+15=+0001111，　B=+24=+0011000；

得 $[A]_{补}=00001111$，　$[B]_{补}=00011000$

可知 $[-B]_{补}=11101000$

所以 $[A+B]_{补}=00001111+00011000=00100111$，

其符号位为0，对应真值为 + 39

$[A-B]_{补}=[A]_{补}+[-B]_{补}=00001111+11101000=11110111$，

其符号位为1，对应真值为-9

2.定点数的运算

三.溢出判断

（1）采用一位符号位

参加操作的两个数符号相同，结果又与原操作数符号不同，则表示结果溢出.

（2）采用双符号位

① $Ss_1Ss_2 = 00$：表示结果为正数，无溢出。

② $Ss_1Ss_2 = 00$：表示结果为正数，溢出。

③ $Ss_1Ss_2 = 00$：表示结果为负数，溢出。

④ $Ss_1Ss_2 = 00$：表示结果为负数，无溢出。

3.数据的存储和排列

大墙方式

	0800H	0801H	0802H	0803H	
…	01H	23H	45H	67H	…

小墙方式

	0800H	0801H	0802H	0803H	
…	67H	45H	23H	01H	…

大端方式按从最高有效字节到最低有效字节的顺序存储数据，即最高有效字节存放在前面；

小端方式按从最低有效字节到最高有效字节的顺序存储数据，即最低有效字节存放在前面.

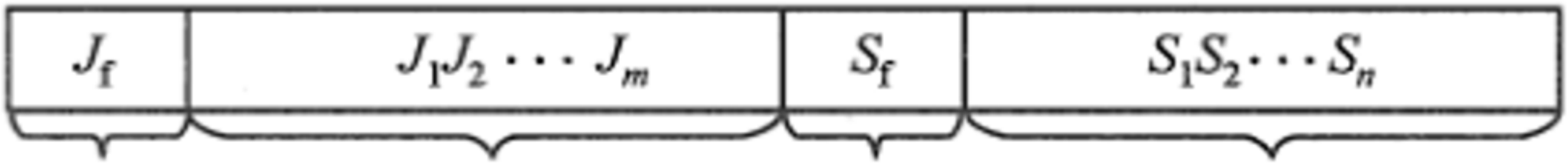
2.2 浮点数的表示与运算

1.浮点数的表示

一.浮点数的表示格式

E和M都是有符号的定点数，E称为阶码，M称为尾数。

可见浮点数由阶码和尾数两部分组成，如图所示:



浮点数的一般格式

1.浮点数的表示

二.规格化浮点数

为了提高运算的精度，需要充分地利用尾数的有效数位，通常采取浮点数规格化形式，即规定尾数的最高数位必须是一个有效值。

左规：当浮点数运算的结果为非规格化时,要进行规格化处理，将尾数算术左移一位、阶码减1（基数为2时）的方法称为左规，左归可能要进行多次。

右规：当浮点数运算的结果尾数出现溢出（双符号位为01或10）时，将尾数算术右移一位、阶码加1（基数为2时）的方法称为右规。需要右归时，只需进行一次。

1.浮点数的表示

1）原码规格化后.

正数为0.1xx…x的形式，

负数为1.1xx…x的形式.

2）补码规格化后.

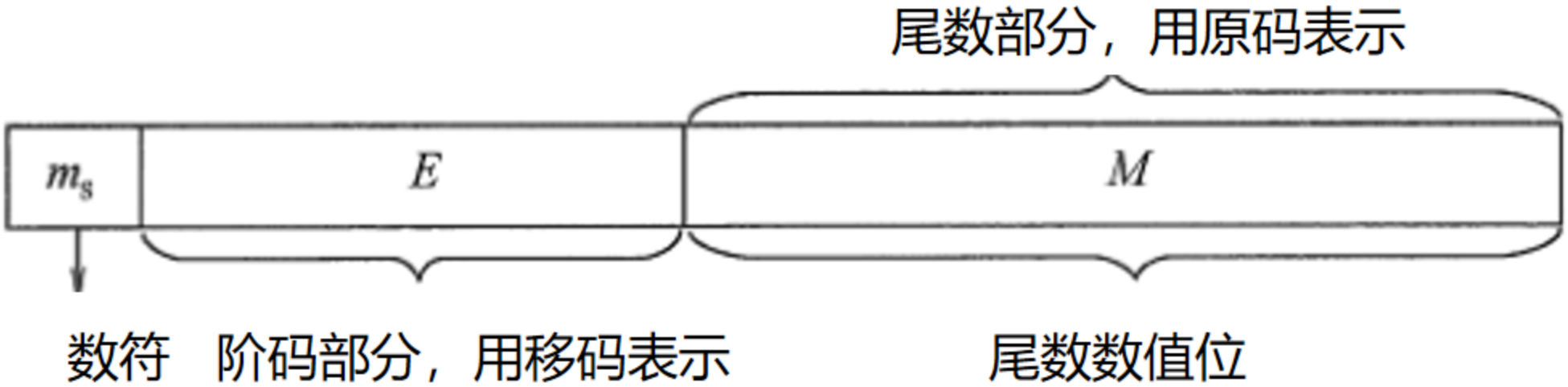
正数为0.1xx…x的形式，尾数的表示范围为1 / 2≤M≤（1-2^{-n} ）、

负数为1.0xx…x的形式，其最大值表示为1.01…1，最小值表示为1.00…0.

1.浮点数的表示

三 . IEEE 754标准

按照IEEE 754标准，常用的浮点数的格式如图所示:



1.浮点数的表示

IEEE 754标准规定常用的浮点数格式有短浮点数（单精度、float型）、长浮点数（双精度、double型）、临时浮点数，见表。

IEEE 754 浮点数的格式

类型	数符	阶码	尾数数值	总位数	偏置值	
					十六进制	十进制
短浮点数	1	8	23	32	7FH	127
长浮点数	1	11	52	64	3FFH	1023
临时浮点数	1	15	64	80	3FFFH	16383

1.浮点数的表示

阶码是以移码形式存储的。对于短浮点数，偏置值为127；

IEEE 754标准中，规格化的短浮点数的真值为 $(-1)^S \times 1.M \times 2^{E-127}$

格式	最小值	最大值
单精度	$E=1, M=0$ $1.0\times 2^{1-127}=2^{-126}$	$E=254, M=.111\cdots,$ $1.111\cdots 1\times 2^{254-127}=2^{127}\times(2-2^{-23})$
双精度	$E=1, M=0$ $1.0\times 2^{1-1023}=2^{-1022}$	$E=2046, M=.1111\cdots,$ $1.111\cdots 1\times 2^{2046-1023}=2^{1023}\times(2-2^{-52})$

IEEE 754浮点数的范围

1.浮点数的表示

四．定点、浮点表示的区别

（1）数值的表示范围

若定点数和浮点数的字长相同，

则浮点表示法所能表示的数值范围将远远大于 定点表示法。

（2）精度

精度是指一个数所含有效数值位的位数。对于字长相同的定点数和浮点数来说，浮点数虽然扩大了数的表示范围，但精度降低了。

四．定点、浮点表示的区别

(3) 数的运算

浮点数包括阶码和尾数两部分，运算时要做尾数的运算，

也做阶码的运算，而且运算结果要求规格化，所以浮点运算比定点运算复杂。

(4) 溢出问题

在定点运算中，当运算结果超出数的表示范围时，发生溢出；

在浮点运算中，运算结果超出尾数表示范围却不一定溢出，

只有规格化后阶码超出所能表示的范围时，才发生溢出。

2.浮点数的加减运算

浮点数运算的特点是阶码运算和尾数运算分开进行。浮点数的加减运算一律采用补码。

浮点数加减运算步骤

一．对阶

对阶的目的是使两个操作数的小数点位置对齐，即使得两个数的阶码相等。为此，先求阶差，然后以小阶向大阶看齐的原则，将阶码小的尾数右移一位（基数为2），阶加1，直到两个数的阶码相等为止。尾数右移时，舍弃掉有效位会产生误差。

二． 尾数求和

将对阶后的尾数按定点数加（减）运算规则运算.

浮点数加减运算步骤

三．规格化

以双符号位为例，当尾数大于0时，其补码规格化形式为 [S]补 = 00.1xx···x

当尾数小于0时，其补码规格化形式为 [S]补 = 11.0xx···x

当尾数的最高数值位与符号位不同时，即为规格化形式。规格化分为左规与右规两种。

1）**左规：** 当尾数出现00.0xx···x或11.1xx····x时，需左规，即尾数左移1位，和的阶码减1，直到尾数为00.1xx···x或11.0xx···x。

2）**右规：** 当尾数求和结果溢出（如尾数为 10．xx···x或01．xx···x）时，需右规，即尾数右移一位，和的阶码加1。

浮点数加减运算步骤

四．舍入

在对阶和右规的过程中，可能会将尾数低位丢失，引起误差。常见的舍入方法“0”舍“1”入法和恒置“1”法.

“0”舍“1”入法： 类似于十进制数运算中的“四舍五入”法，

即在尾数右移时，被移去的最高数值位为0，则舍去；

被移去的最高数值位为1时，则在尾数的末位加1。

这样做可能会使尾数又溢出，此时需再做一次右规。

恒置“1”法： 尾数右移时，不论丢掉的最高数值位是“1”还是“0”，

都使右移后的尾数末位恒置“1”。

这种方法同样有使尾数变大和变小的两种可能。

浮点数加减运算步骤

五．溢出判断

在浮点数规格化中已指出，当尾数之和（差）出现01．xx…x或10．xx…x时，并不表示溢出，只能将此数右规后，再根据阶码来判断浮点数运算结果是否溢出。

【题5】已知X = -0.875x21，Y = 0.625x22，设浮点数格式为阶符1位，阶码2位，数符1位，尾数3位，通过补码求出Z = X-Y的二进制浮点数规格化结果是（ B ）

A. 1011011　　B.0111011　　C. 1001011　　D.以上都不对

解析：将X = -0.875x21和Y = 0.625x22写成7位浮点数形式，

有X = 001 1001和Y = 0100101（前半部分为阶符、阶码，后半部分为数符、数码），

对阶之后，X = 0101100，对阶后尾数做减法，结果需要进行右规，

最终结果Z = 0111011。

注意：尾数为01．XXX或10．XXX时在浮点数中不算真正的溢出，

此时只需右移一位阶码加1即可