对"右移一位"和"双符号位"的说明：

③ **规格化并判溢出。**

$[X_M+Y_M]_{补}$ 的结果为 00011000，需要将尾数左移一位，同时阶码减 1，没有溢出，得：

$X+Y$：0　00010　110000　　即 $+0.110000\times 2^{+0010}$

$[X_M-Y_M]_{补}$ 的结果为 01010010，需要将尾数右移一位，同时阶码加 1，没有溢出，得：

$X-Y$：0　00100　101001　　即 $+0.101001\times 2^{+0100}$

很多人对于书本【例10.13】"规格化并判溢出"中的右移一位存有疑惑（如上图红线标注所示），其很大一部分原因在于对"双符号位"的作用和意义缺少理解。

一、先理解双符号位最大的作用（如果你已经很了解了，请直接看"**二**"）

1.解决补码加减的溢出问题。即使溢出了，最高符号位（**如"10"的1表示负，或者"01"的0表示正**）仍旧可以正确表示运算结果的正负性。

2.可以立马看出运算的结果是否发生了溢出：

符号位是**"00"或"11"=没有溢出**，符号位后面的部分就是所有的数值了。

符号位若是"01"，可以看出结果**是正数**，而且**溢出了一位**，这也说明**符号位后面的部分不能代表完整的数值**（因为有溢出呀，肯定有个高位的"1"进位到了符号位，这个怎么能忘呢？）。

符号位是"10"的时候，结果**是负数**，而且**溢出了一位**，**符号位后面的部分不能代表完整的数值。**

二、再看图片中的例子

$[X_M+Y_M]_{补}$ 的结果为 00011000，需要将尾数左移一位，同时阶码减 1，没有溢出，得：

$X+Y$：0　00010　110000　　即 $+0.110000\times 2^{+0010}$

> 第一种情况很好理解，"00"说明没有溢出，后面的"011000"就是代表了所有的数值大小，很明显开头不是"1"，需要左移成"110000"。

$[X_M-Y_M]_{补}$ 的结果为 01010010，需要将尾数右移一位，同时阶码加 1，没有溢出，得：

$X-Y$：0　00100　101001　　即 $+0.101001\times 2^{+0100}$

> 解析：对于第二种情况，我们已经知道到"01"表示有溢出，那么显然后面的"010010"不是所有数值大小，正确的理解应该是要想到"1010010"才是我们算出来的结果！只不过最高位的"1"已经超出了我们所规定的数值部分的位数，被进位到符号位去了，这也是为什么符号位不是"00"而是"01"。这也是"01"在移位方向的判断中对我们唯一的帮助，知道前面还有个高位的"1"后就不用再去搭理什么"双符号位"了。接下来，只需要明白我们的运算结果的绝对值是大于1的，需要右移一位，并且在右移后的最高位补"1"，即从"010010"->"101001"，并且阶码大小+1就行了。
> 至于双符号位的"01"怎么办？回答当然是没啥特别的用处了，知道X-Y的"数符"应该是"0"（双符号位能看出结果的正负性）后就可以完全扔掉了。

> 再举个例子，设X的双符号位的补码是"11 0000"，Y的双符号位的补码是"11 0001"。X+Y="10 0001"，结果的符号位"10"说明结果是负的并且有溢出。那么"0001"就需要右移一位，并且需要在右移后的尾数的最高位补上溢出的"1"，即"0001"->"1000"（以截断法为例），数符根据"10"可以知道应该是"1"，最后阶码大小+1。

总结，在判断移位方向的时候，双符号位用于提示我们是否有运算溢出，只要有溢出，一律采用右移，并且在 在最高位补“1”。没有运算溢出时，则根据尾数具体情况来选择移动方向。

完