

【计算机组成原理】定点乘法运算之原码一位乘法

$x * y = z$

讨论已知x和y的情况下，怎么通过原码一位乘法方法得出z~~

首先说下**运算规则**~

1. z的符号位通过x和y的符号位进行异或运算得到~（这个很好理解哒，负负得正，正正得正，正负得负嘛~所以把符号位异或得到的结果就是乘法运算后应该的结果咯~）

2. 所以就不用讨论x和y的符号位啦，z除了符号之外的其他部分由x的绝对值乘以y的绝对值得到~

1、2两点总结一下就是说：**被乘数和乘数均取绝对值参加运算，符号位单独考虑**~

我们手工进行乘法运算的时候，是通过y从右往左每一位都和x相乘，（乘完一次就往前缩进一个数位）然后把结果相加得到的~机器也是这样运算哒~不过机器为了节约空间，毕竟按照手算的方法那样两个n位相乘最后可能会需要2n的长度空间才能得到结果，计算机是采用把每次用y的一位和x相乘的结果（叫做部分积）累加后右移一位，再处理y当前位的下一位的~

3. 我们把被乘数x先取双符号，而且让部分积初始值为0，并且长度和被乘数x相同（就是添0让长度相同的意思啦~）

计算机只有0和1，所以处理乘法的时候运算法则远没有99乘法表那么复杂，运算规则为：

4. 从y的最后一位开始（一直到第一位）分别与x相乘：

- 当y的当前位为1，则部分积加上x的绝对值
- 当y的当前位为0，则部分积加上0

5. 右移一位，在前面加0。不断处理y的每一位，知道y的所有位都处理过为止~~

可能有点晕，**举个栗子**~

比如 $x = 0.1101$ ， $y = 0.1011$

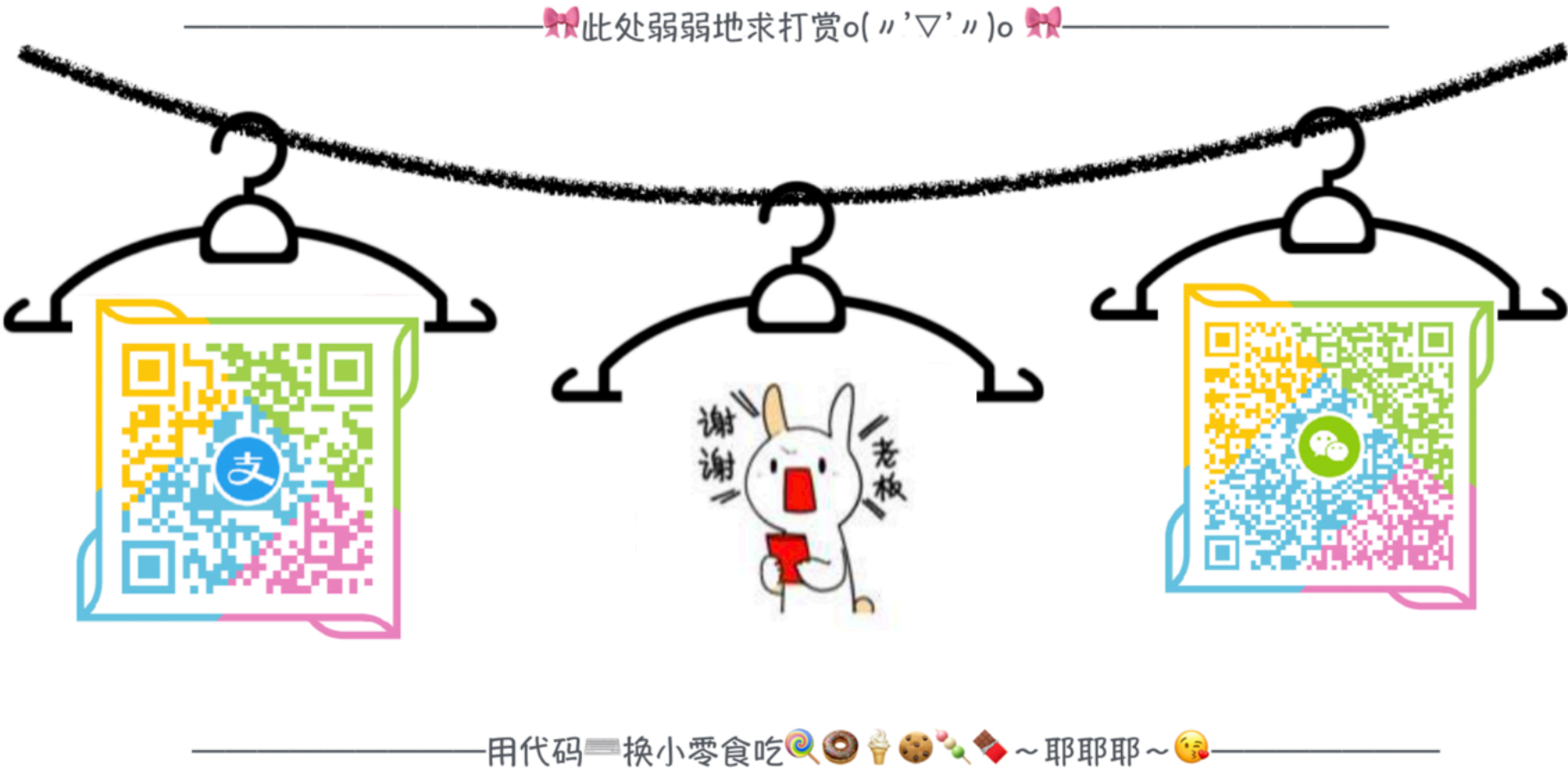
1. 先把部分积设为初始值0（长度扩展到和x相同），即
00.0000
2. y的最后一位是1，所以要加上x的绝对值：
 $00.0000 + 00.1101 = 00.1101$
3. **右移1位**，前面补0，变成了00.01101
4. 好啦，下面处理y的倒数第二位，还是1，继续加x的绝对值：
 $00.01101 + 00.1101 = 01.00111$
5. 别忘记**右移1位**，前面补0，这样就变成了00.100111
6. y倒数第3位是0，只要加0就好了（加0的结果还是本身啊。），所以还是：
00.100111
7. 既然加了0，别忘记**右移一位**哦，所以变成了00.0100111
8. y还剩最后一个位（也就是第一位）没处理啦，第一位是1哦，那就加上x的绝对值：
 $00.0100111 + 00.1101 = 01.0001111$
9. 别忘记**右移一位**！所以**最后结果是 00.10001111**~~~

嗯好啦，这就是最后结果~~不过双符号位就变成一个0就好咯，也就是最后结果为0.10001111~~~

♥[点击这里 -> 订阅《PAT | 蓝桥 | LeetCode学习路径 & 刷题经验》by 柳婳](#)

♥[点击这里 -> 订阅《从放弃C语言到使用C++刷算法的简明教程》by 柳婳](#)

♥[点击这里 -> 订阅PAT甲级乙级、蓝桥杯、GPLT天梯赛、LeetCode题解离线版](#)



用代码 换 小零食吃 🍪🍩🍫🍬🍭🍮🍯🍡 ~ 耶耶耶~ 🥳

上—
【软件项目管理】资源平衡最好用于非关键路径活动

下一
【计算机组成原理】定点乘法运算之补码一位乘法（Booth算法）