

## 补码一位乘法

理解了原码一位乘法后，我们继续从数学原理出发，推导补码乘法

为了得到补码一位乘法的规律，我们先从补码和真值的转换公式开始讨论

## 补码与真值的转换公式

设 $[x]_{\text{补}} = x_0.x_1x_2\dots x_n$ , 当 $x \geq 0$ 时,

当 $x < 0$ 时

所以

故得出  $x =$

## 补码的右移

正数补码右移一位，相当于乘 $1/2$ ，  
负数用补码表示时，右移一位还相当于乘 $1/2$ 么

设 $[x]_{\text{补}} = x_0.x_1x_2\dots x_n$ , 因为

$x =$

所以

写成补码形式，即得

## 补码乘法规则

设被乘数 $[x]_{\text{补}} = x_0.x_1x_2\dots x_n$ , 乘数 $[y]_{\text{补}} = y_0.y_1y_2\dots y_n$ 均为任意符号

则有补码乘法算式

$$[x*y]_{\text{补}} =$$

## 证明如下

(1) 被乘数 $x$ 符号任意，乘数 $y$ 符号为正

根据补码定义

$$[x]_{\text{补}} =$$

$$[y]_{\text{补}} =$$

所以

$$[x]_{\text{补}} * [y]_{\text{补}} =$$

其中  $(y_1y_2\dots y_n)$  是大于0的正整数，根据模运算性质有

$$2(y_1y_2\dots y_n) =$$

所以

$$[x]_{\text{补}} * [y]_{\text{补}} =$$

即

$$[x*y]_{\text{补}} =$$

(2) 被乘数 $x$ 符号任意，乘数 $y$ 符号为负

$$[x]_{\text{补}} =$$

$$[y]_{\text{补}} =$$

由此

$$y =$$

所以

$$x * y =$$

$$[x * y]_{\text{补}} =$$

所以

$$[x * y]_{\text{补}} =$$

(3) 被乘数 $x$ 符号任意，乘数 $y$ 符号为负

将以上 (1) (2) 两种情况综合起来，即得补码乘法的统一算式

$$[x * y]_{\text{补}} =$$

为了推出串行逻辑实现的分步算法，将上式展开加以变换

$$[x * y]_{\text{补}} =$$

写成递推公式如下：

$$[z_0]_{\text{补}} =$$

因此，我们不难得到补码乘法，本质上要进行如下 $n+1$ 次加法操作和 $n$ 次移位操作

对比 原码是 $y_i \times x$

1.加法  $(y(n+1)-y_n) \times [x]_{\text{补}}$

1. $y(n+1)-y_n=0$ ,则加0

2. $y(n+1)-y_n=1$ ,则加 $[x]_{\text{补}}$

3. $y(n+1)-y_n=-1$ ,则减 $[x]_{\text{补}}$ , 等价于加 $[-x]_{\text{补}}$

这样的操作  
总共 $n+1$ 次!

2.乘 $2^{(-1)}$  → 等价于右移一位

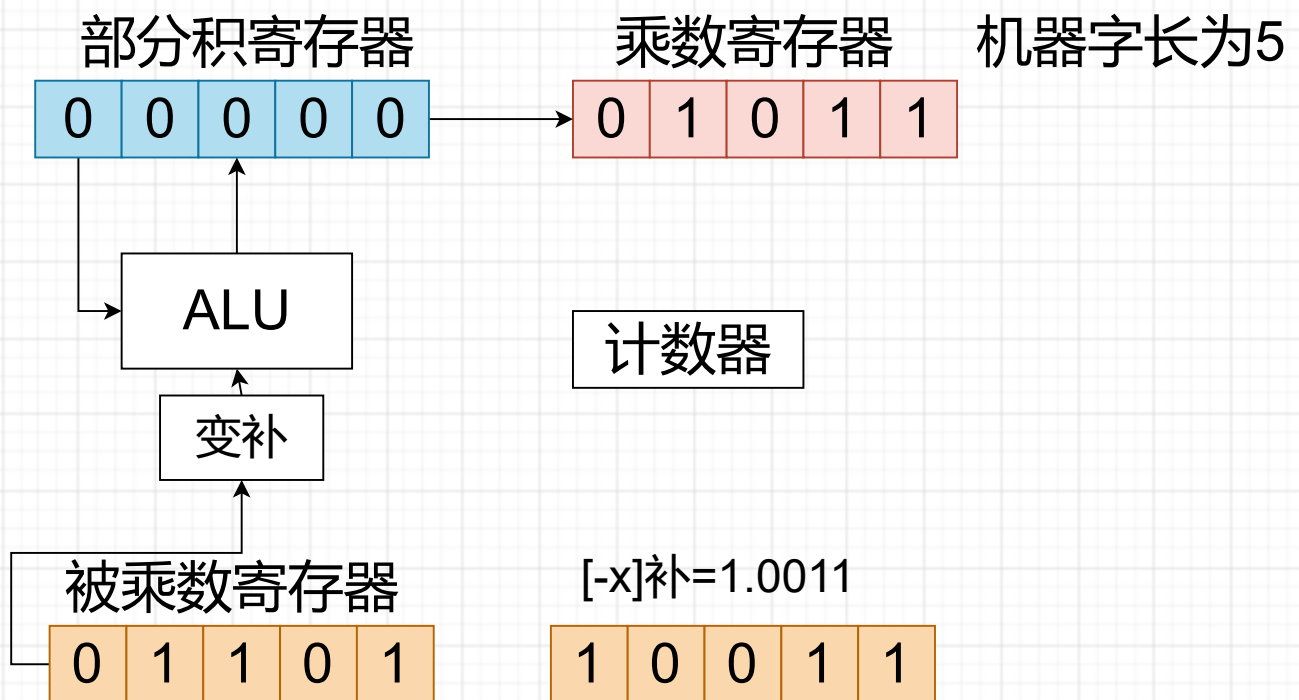
右移操作总共 $n$ 次!

注意，补码乘法中，包含符号位，因此每次移位一定是**算术右移**

懂了原理之后，我们来看看计算机是如何进行补码乘法运算的

$x=0.1101$

$y=0.1011$

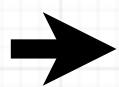


手算模拟

部分积	乘数	说明
0 0 0 0 0	0 1 0 1 1 0	$z_0=0, y_5-y_4=-1$
+ 1 0 0 1 1		加法 $(y_{i+1}-y_i) \times [x]_{\text{补}}$

1 0 0 1 1

0 1 0 1 1 0



1 1 0 0 1

1 0 1 0 1 1

算术右移 (乘 $2^{-1}$ )  
 $z1\downarrow, y4-y3=0$

+

0 0 0 0 0

0

加法  $(y(i+1)-y_i) * [x]_{\text{补}}$

1 1 0 0 1

1 0 1 0 1 1



1 1 1 0 0

1 1 0 1 0 1

算术右移 (乘 $2^{-1}$ )  
 $z2\downarrow, y3-y2=1$

+

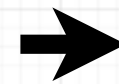
0 1 1 0 1

0 0

加法  $(y(i+1)-y_i) * [x]_{\text{补}}$

0 1 0 0 1

1 1 0 1 0 1



0 0 1 0 0

1 1 1 0 1 0

算术右移 (乘 $2^{-1}$ )  
 $z3\downarrow, y2-y1=-1$

+

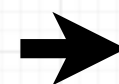
1 0 0 1 1

0 0 0

加法  $(y(i+1)-y_i) * [x]_{\text{补}}$

1 0 1 1 1

1 1 1 0 1 0



1 1 0 1 1

1 1 1 1 0 1

算术右移 (乘 $2^{-1}$ )  
 $z4\downarrow, y1-y0=1$

+

0 1 1 0 1

0 0 0 0

加法  $(y(i+1)-y_i) * [x]_{\text{补}}$

0 1 0 0 0

1 1 1 1 0 1

最后一步不移位

总结, 定点小数 $n+1$ 位补码乘法 (包括符号位),  
进行 $n+1$ 次加法,  $n$ 次右移, 得到 $2n+1$ 位结果

口诀: 蒜头补肾454或者蒜头补乘454

蒜头 (ou) : 算术右(ou)移, 符号位不变, 正数补0, 负数补1

454表示, 4位尾数, 5次加法, 4次右移

里昂学长

接下来我们再进行进一步改进手算模拟, 使用双符号位

$$x=0.1101$$

$$y=0.1011$$