

本节内容

# 定点数 原码乘法运算

王道考研/CSKAOYAN.COM

1

雨声警告⚠



今天的雨  
下得跟依萍找她爸要钱那天一样大



今天的雨  
下的跟棋贵人被打死那天一样大

王道考研/CSKAOYAN.COM

2

本节总览

乘法运算

乘法运算的实现思想

原码的一位乘法

补码的一位乘法

王道考研/CSKAOYAN.COM


3

手算乘法（十进制）

r 进制:  $K_n K_{n-1} \dots K_2 K_1 K_0 K_{-1} K_{-2} \dots K_{-m}$


$= K_n \times r^n + K_{n-1} \times r^{n-1} + \dots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \dots + K_{-m} \times r^{-m}$

你怎么这个亚子



$$\begin{array}{r} 0.985 \\ \times 0.211 \\ \hline 985 \\ 985 \\ 1970 \\ \hline 0.207835 \end{array}$$

$$\begin{array}{r} 0.985 \\ \times 0.211 \\ \hline 0.000985 \\ 0.00985 \\ 0.1970 \\ \hline 0.207835 \end{array}$$



$$0.211 = 2 \times 10^{-1} + 1 \times 10^{-2} + 1 \times 10^{-3}$$
$$0.985 = 985 \times 10^{-3}$$
$$0.985 \times 0.211 = (985 \times 1 \times 10^{-6}) + (985 \times 1 \times 10^{-5}) + (985 \times 2 \times 10^{-4})$$

王道考研/CSKAOYAN.COM

4

王道考 研,cskaoyan.com

2

手算乘法（二进制）

$r$  进制:  $K_n K_{n-1} \dots K_2 K_1 K_0 K_{-1} K_{-2} \dots K_{-m}$   
 $= K_n \times r^n + K_{n-1} \times r^{n-1} + \dots + K_2 \times r^2 + K_1 \times r^1 + K_0 \times r^0 + K_{-1} \times r^{-1} + K_{-2} \times r^{-2} + \dots + K_{-m} \times r^{-m}$

$$\begin{array}{r} 0.1101 \\ \times 0.1011 \\ \hline 1101 \\ 1101 \\ 0000 \\ 1101 \\ \hline 0.10001111 \end{array}$$



$$\begin{array}{r} 0.1101 \\ \times 0.1011 \\ \hline 0.00001101 \\ 0.0001101 \\ 0.000000 \\ 0.01101 \\ \hline 0.10001111 \end{array}$$



- 考虑用机器实现:
- 实际数字有正负, 符号位如何处理?
  - 乘积的位数扩大一倍, 如何处理?
  - 4个位积都要保存下来最后统一相加?

用“移位”实现

(乘数)  $0.1011 = 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$   
(被乘数)  $0.1101 = 1101 \times 2^{-4}$   
 $0.1101 \times 0.1011 = (1101 \times 1 \times 2^{-8}) + (1101 \times 1 \times 2^{-7}) + (1101 \times 0 \times 2^{-6}) + (1101 \times 1 \times 2^{-5})$

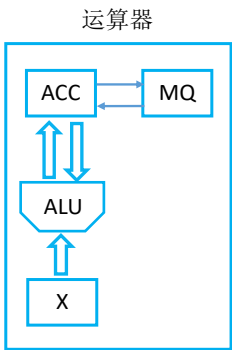
原码一位乘法

设机器字长为  $n+1=5$  位 (含1位符号位),  $[x]_{\text{原}} = 1.1101$ ,  $[y]_{\text{原}} = 0.1011$ , 采用原码一位乘法求  $x \cdot y$

符号位	数值位
-----	-----

符号单独处理: 符号位 =  $x_s \oplus y_s$   
数值位取绝对值进行乘法计算  
 $[x]_{\text{原}} = 0.1101$ ,  $[y]_{\text{原}} = 0.1011$

穿越：运算器的基本组成



运算器：用于实现算术运算（如：加减乘除）、逻辑运算（如：与或非）

- ACC: 累加器，用于存放操作数，或运算结果。
- MQ: 乘商寄存器，在乘、除运算时，用于存放操作数或运算结果。
- X: 通用的操作数寄存器，用于存放操作数
- ALU: 算术逻辑单元，通过内部复杂的电路实现算数运算、逻辑运算

	加	减	乘	除
Accumulator	ACC	被加数、和	乘积高位	被除数、余数
Multiple-Quotient Register	MQ		乘数、乘积低位	商
Arithmetic and Logic Unit	X	加数	被乘数	除数

王道考研/CSKAOYAN.COM

7

原码一位乘法

设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{原} = 1.1101$ ， $[y]_{原} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$

符号位	数值位
-----	-----

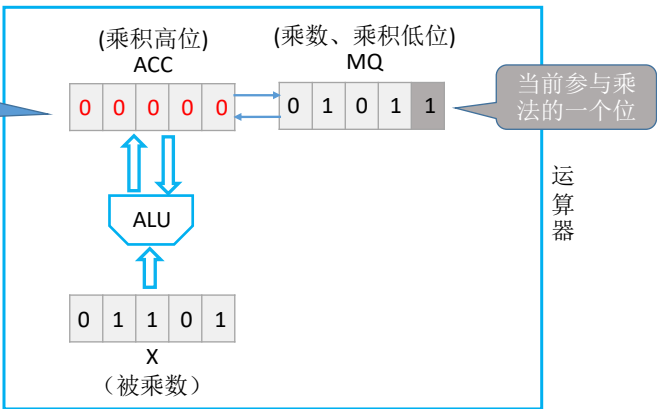
符号单独处理：符号位 =  $x_s \oplus y_s$   
数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复n次

0.1101  
× 0.1011  
-----  
01101  
01101  
00000  
01101  
-----  
0.10001111

当前位=1，则ACC加上被乘数  
当前位=0，则ACC加上 0

在正式进行乘法之前，ACC置0  
 $00000+01101=01101$

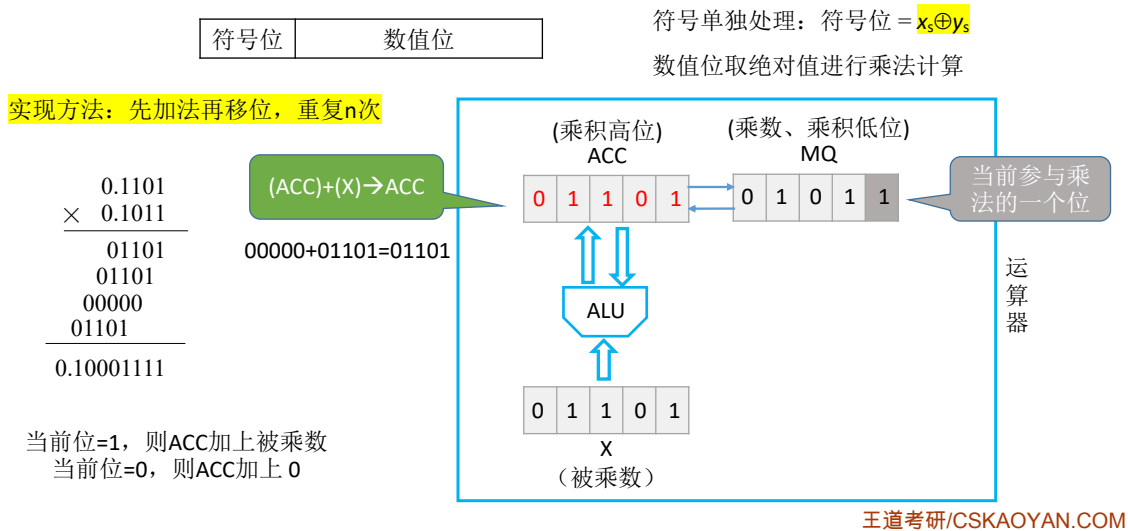


王道考研/CSKAOYAN.COM

8

## 原码一位乘法

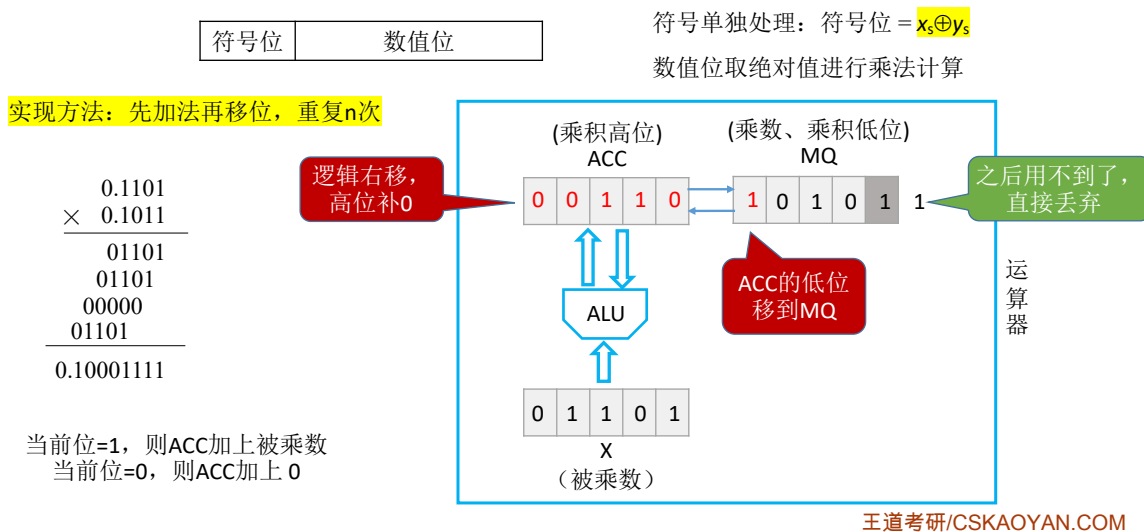
设机器字长为  $n+1=5$  位 (含 1 位符号位),  $[x]_{\text{原}} = 1.1101$ ,  $[y]_{\text{原}} = 0.1011$ , 采用原码一位乘法求  $x \cdot y$



9

## 原码一位乘法

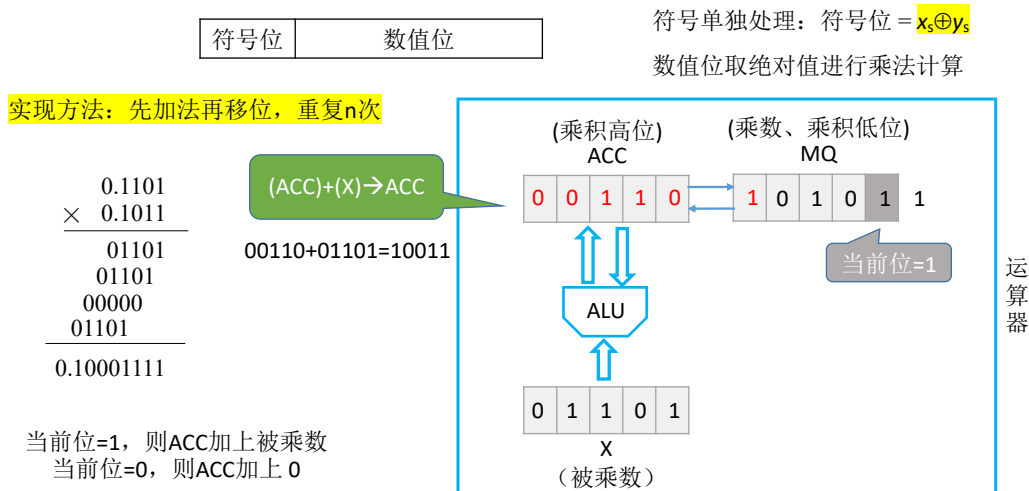
设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{\text{原}} = 1.1101$ ， $[y]_{\text{原}} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$



10

## 原码一位乘法

设机器字长为  $n+1=5$  位 (含 1 位符号位),  $[x]_{\text{原}} = 1.1101$ ,  $[y]_{\text{原}} = 0.1011$ , 采用原码一位乘法求  $x \cdot y$

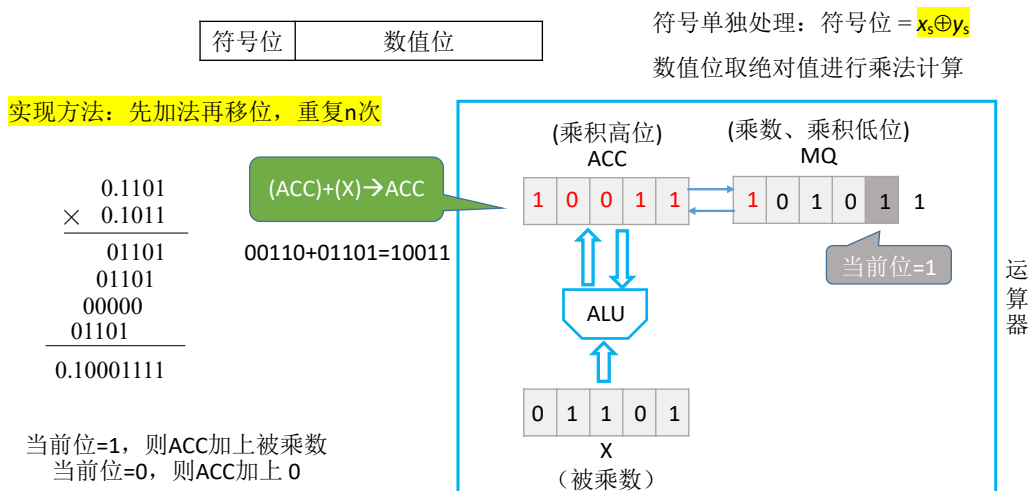


王道考研/CSKAOYAN.COM

11

## 原码一位乘法

设机器字长为  $n+1=5$  位 (含 1 位符号位),  $[x]_{\text{原}} = 1.1101$ ,  $[y]_{\text{原}} = 0.1011$ , 采用原码一位乘法求  $x \cdot y$



王道考研/CSKAOYAN.COM

12

### 原码一位乘法

设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{\text{原}} = 1.1101$ ， $[y]_{\text{原}} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$

符号位	数值位

符号单独处理：符号位 =  $x_s \oplus y_s$   
数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复  $n$  次

0.1101

×

0.1011

01101

01101

00000

01101

0.10001111

当前位=1，则ACC加上被乘数  
当前位=0，则ACC加上 0

(乘积高位)  
ACC

01001

(乘数、乘积低位)  
MQ

1101011

逻辑右移，高位补0

红色部分可称为“部分积”

ALU

01101

X  
(被乘数)

运算器

王道考研/CSKAOYAN.COM

13

### 原码一位乘法

设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{\text{原}} = 1.1101$ ， $[y]_{\text{原}} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$

符号位	数值位

符号单独处理：符号位 =  $x_s \oplus y_s$   
数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复  $n$  次

0.1101

×

0.1011

01101

01101

00000

01101

0.10001111

当前位=1，则ACC加上被乘数  
当前位=0，则ACC加上 0

(乘积高位)  
ACC

01001

(乘数、乘积低位)  
MQ

1101011

(ACC)+0→ACC

当前位=0

ALU

01101

X  
(被乘数)

运算器

王道考研/CSKAOYAN.COM

14

王道考 研,cskaoyan.com

7

## 原码一位乘法

设机器字长为  $n+1=5$  位 (含 1 位符号位),  $[x]_{\text{原}} = 1.1101$ ,  $[y]_{\text{原}} = 0.1011$ , 采用原码一位乘法求  $x \cdot y$

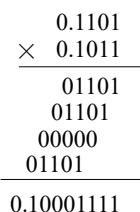
符号位	数值位
-----	-----

符号单独处理：符号位 =  $x_s \oplus y_s$

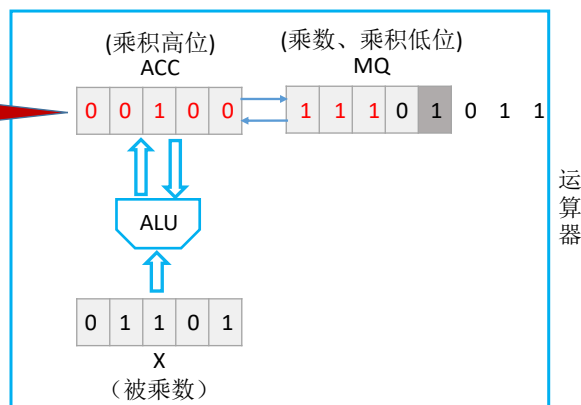
数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复n次

逻辑右移，  
高位补0



当前位=1，则ACC加上被乘数  
当前位=0，则ACC加上 0



王道考研/CSKAOYAN.COM

15

## 原码一位乘法

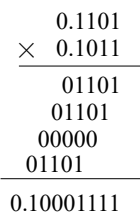
设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{\text{原}} = 1.1101$ ， $[y]_{\text{原}} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$

符号位	数值位
-----	-----

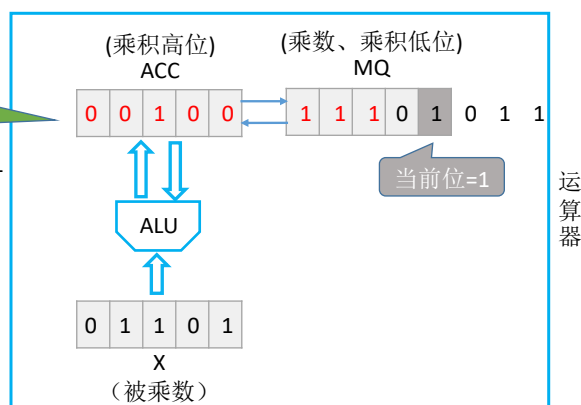
符号单独处理：符号位 =  $x_s \oplus y_s$

数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复n次

$$(ACC) + (X) \rightarrow ACC$$
$$00100 + 01101 = 10001$$


当前位=1, 则ACC加上被乘数  
当前位=0, 则ACC加上 0



王道考研/CSKAOYAN.COM

16



### 原码一位乘法

设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{\text{原}} = 1.1101$ ， $[y]_{\text{原}} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$

符号位	数值位

符号单独处理：符号位 =  $x_s \oplus y_s$   
数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复  $n$  次

$$\begin{array}{r} 0.1101 \\ \times 0.1011 \\ \hline 01101 \\ 01101 \\ 00000 \\ 01101 \\ \hline 0.10001111 \end{array}$$

(ACC)+(X)→ACC  
00100+01101=10001

当前位=1，则ACC加上被乘数  
当前位=0，则ACC加上 0

(乘积高位)  
ACC

(乘数、乘积低位)  
MQ

1 0 0 0 1

1 1 1 0 1 0 1 1

当前位=1

运算器

ALU

0 1 1 0 1

X  
(被乘数)

王道考研/CSKAOYAN.COM

17

### 原码一位乘法

设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{\text{原}} = 1.1101$ ， $[y]_{\text{原}} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$

符号位	数值位

符号单独处理：符号位 =  $x_s \oplus y_s$   
数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复  $n$  次

$$\begin{array}{r} 0.1101 \\ \times 0.1011 \\ \hline 01101 \\ 01101 \\ 00000 \\ 01101 \\ \hline 0.10001111 \end{array}$$

逻辑右移，高位补0

小数点隐含位置

乘数的符号位不用参与运算

你品 你细品

当前位=1，则ACC加上被乘数  
当前位=0，则ACC加上 0

(乘积高位)  
ACC

(乘数、乘积低位)  
MQ

0 1 0 0 0

1 1 1 1 0 1 0 1 1

乘数的符号位不用参与运算

运算器

ALU

0 1 1 0 1

X  
(被乘数)

王道考研/CSKAOYAN.COM

18

王道考 研,cskaoyan.com

9

原码一位乘法

设机器字长为  $n+1=5$  位（含1位符号位）， $[x]_{\text{原}} = 1.1101$ ， $[y]_{\text{原}} = 0.1011$ ，采用原码一位乘法求  $x \cdot y$

符号位	数值位
-----	-----

符号单独处理：符号位 =  $x_s \oplus y_s$

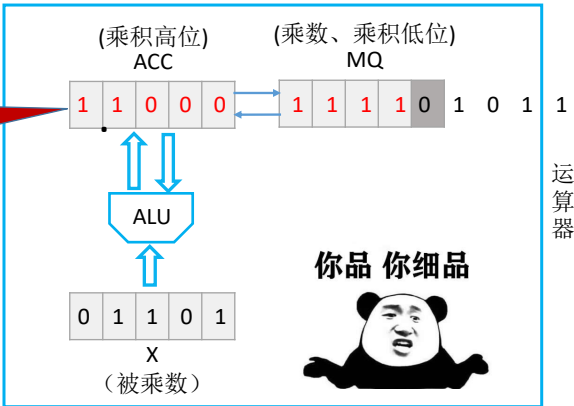
数值位取绝对值进行乘法计算

实现方法：先加法再移位，重复  $n$  次

0.1101  
× 0.1011  
-----  
01101  
01101  
00000  
01101  
-----  
0.10001111

当前位=1，则ACC加上被乘数  
当前位=0，则ACC加上 0

修改符号  
位  $x_s \oplus y_s = 1$



王道考研/CSKAOYAN.COM

19

原码一位乘法（手算模拟）

设机器字长为5位（含1位符号位， $n=4$ ）， $x = -0.1101$ ， $y = +0.1011$ ，采用原码一位乘法求  $x \cdot y$



解： $|x| = 00.1101$ ， $|y| = 00.1011$ ，原码一位乘法的求解过程如下。

Tips:

- 乘数的符号位不参与运算，可以省略
- 原码一位乘可以只用单符号位
- 答题时最终结果最好写为原码机器数

原码一位乘法：机器字长  $n+1$ ，数值部分占  $n$  位

符号位通过异或确定；数值部分通过被乘数和乘数绝对值的  $n$  轮加法、移位完成  
根据当前乘数中参与运算的位确定 (ACC) 加什么。若当前运算位=1，则  $(ACC) + [|x|]_{\text{原}}$ ；  
若=0，则  $(ACC) + 0$ 。

每轮加法后ACC、MQ的内容统一逻辑右移

符号位  $P_s = x_s \oplus y_s = 1 \oplus 0 = 1$ ，得  $x \cdot y = -0.10001111$ 。

ACC (高位部分积)	MQ (低位部分积/乘数)	说明
00.0000	1011 丢失位	起始情况
+ x  00.1101		$C_4=1$ ，则+ x
00.1101		
右移 00.0110	1101 1	右移部分积和乘数
+ x  00.1101		$C_4=1$ ，则+ x
01.0011		
右移 00.1001	1110 11	右移部分积和乘数
+0 00.0000		$C_4=0$ ，则+0
00.1001		
右移 00.0100	1111 011	右移部分积和乘数
+ x  00.1101		$C_4=1$ ，则+ x
01.0001		
右移 00.1000	1111 1011	右移部分积和乘数
		乘数全部移出
		结果的绝对值部分

王道考研/CSKAOYAN.COM

20

本节内容

定点数  
补码乘法运算

王道考研/CSKAOYAN.COM

21

补码一位乘法

设机器字长为5位（含1位符号位， $n=4$ ）， $x = -0.1101$ ， $y = +0.1011$ ，采用Booth算法求 $x \cdot y$   
 $[x]_{补} = 1.0011$ ， $[-x]_{补} = 0.1101$ ， $[y]_{补} = 0.1011$

原码一位乘法：  
进行  $n$  轮加法、移位

每次加法可能  $+0$ 、 $+ [|x|]_{原}$

每次移位是“逻辑右移”

符号位不参与运算

根据当前MQ中的最低位来确定加什么

MQ中最低位 = 1时， $(ACC) + [|x|]_{原}$   
MQ中最低位 = 0时， $(ACC) + 0$

补码一位乘法：  
进行  $n$  轮加法、移位，最后再多来一次加法

每次加法可能  $+0$ 、 $+ [x]_{补}$ 、 $+ [-x]_{补}$

每次移位是“补码的算数右移”

符号位参与运算

根据当前MQ中的最低位、辅助位来确定加什么

辅助位 - MQ中最低位 = 1时， $(ACC) + [x]_{补}$   
辅助位 - MQ中最低位 = 0时， $(ACC) + 0$   
辅助位 - MQ中最低位 = -1时， $(ACC) + [-x]_{补}$

王道考研/CSKAOYAN.COM

22

### 补码一位乘法

设机器字长为5位（含1位符号位， $n=4$ ）， $x = -0.1101$ ， $y = +0.1011$ ，采用Booth算法求 $x \cdot y$   
 $[x]_{补} = 1.0011$ ， $[-x]_{补} = 0.1101$ ， $[y]_{补} = 0.1011$

运算器

(乘积高位) ACC

0 0 0 0 0

(乘数、乘积低位) MQ

0 1 0 1 1 0

ALU

X

1 1 0 0 1 1

(被乘数)

所有寄存器都统一用 $n+2$ 位，因此采用双符号位补码运算

辅助位初始为0。每次右移会使MQ的最低位顶替原本的辅助位（事实上MQ共 $n+2$ 位）

会有辅助电路实现 $[-x]_{补}$ 的转换

补码一位乘法：  
进行 $n$ 轮加法、移位，最后再多来一次加法

每次加法可能 $+0$ 、 $+ [x]_{补}$ 、 $+ [-x]_{补}$

每次移位是“补码的算数右移”

符号位参与运算

根据当前MQ中的最低位、辅助位 来确定加什么

辅助位 - MQ中最低位 = 1时， $(ACC) + [x]_{补}$   
辅助位 - MQ中最低位 = 0时， $(ACC) + 0$   
辅助位 - MQ中最低位 = -1时， $(ACC) + [-x]_{补}$

王道考研/CSKAOYAN.COM

23

### 补码一位乘法（手算模拟）

设机器字长为5位（含1位符号位， $n=4$ ）， $x = -0.1101$ ， $y = +0.1011$ ，采用Booth算法求 $x \cdot y$   
 $[x]_{补} = 1.0011$ ， $[-x]_{补} = 0.1101$ ， $[y]_{补} = 0.1011$

	(高位部分积)	(低位部分积/乘数)	说明
	00.0000	0.1011 0 丢失位	起始情况
正数算数右移	$+ [-x]_{补}$ 00.1101		$Y_4 Y_5 = 10$ ， $Y_5 - Y_4 = -1$ ，则 $+ [-x]_{补}$
	00.1101		
右移	00.0110	----- 10.101 10	右移部分积和乘数
$+0$	00.0000		$Y_4 Y_5 = 11$ ， $Y_5 - Y_4 = 0$ ，则 $+0$
	00.0110		
右移	00.0011	----- 010.10 110	右移部分积和乘数
负数算数右移	$+ [x]_{补}$ 11.0011		$Y_4 Y_5 = 01$ ， $Y_5 - Y_4 = 1$ ，则 $+ [x]_{补}$
	11.0110		
右移	11.1011	----- 0010.1 0110	右移部分积和乘数
	11.1011		
右移	11.0100	----- 00010.1 0110	右移部分积和乘数
$+ [-x]_{补}$ 00.1101			$Y_4 Y_5 = 10$ ， $Y_5 - Y_4 = -1$ ，则 $+ [-x]_{补}$
	00.1000		
最后多一次加法	右移 00.0100	----- 00010.1 0110	右移部分积和乘数
	$+ [x]_{补}$ 11.0011		$Y_4 Y_5 = 01$ ， $Y_5 - Y_4 = 1$ ，则 $+ [x]_{补}$
	11.0111		
		构成 $[x \cdot y]_{补}$	

辅助位

原符号位

正数算数右移

负数算数右移

最后多一次加法

$n$ 轮加法、算数右移，加法规则如下：  
辅助位 - MQ中最低位 = 1时， $(ACC) + [x]_{补}$   
辅助位 - MQ中最低位 = 0时， $(ACC) + 0$   
辅助位 - MQ中最低位 = -1时， $(ACC) + [-x]_{补}$

补码的算数右移：  
符号位不动，数值位右移，正数右移补0，负数右移补1（符号位是啥就补啥）

注：一般来说，Booth算法的被乘数、部分积采用双符号位补码

$[x \cdot y]_{补} = 11.01110001$   
即 $x \cdot y = -0.10001111$

王道考研/CSKAOYAN.COM

24

王道考 研/CSKAOYAN.COM

12

## 知识点回顾

部分积、被乘数、乘数都可采用双符号位原码，也可用单符号位原码（手算时乘数的符号位可不写）

**原码一位乘法：**

符号位通过异或确定，数值位由被乘数和乘数的绝对值进行  $n$  轮加法、移位

每次加法可能  $+0$ 、 $+|x|_{\text{原}}$

每次移位是“逻辑右移”

乘数的符号位不参与运算

MQ中最低位 = 1时， $(ACC)+|x|_{\text{原}}$   
MQ中最低位 = 0时， $(ACC)+0$



朋友，过两招？

部分积、被乘数采用双符号位补码；乘数采用单符号位补码，并在末位添个0

**补码一位乘法（Booth算法）：**

符号位、数值位都是由被乘数和乘数进行  $n$  轮加法、移位，最后再多来一次加法

每次加法可能  $+0$ 、 $+x_{\text{补}}$ 、 $+[-x]_{\text{补}}$

每次移位是“补码的算数右移”

乘数的符号位参与运算

辅助位 - MQ中“最低位” = 1时， $(ACC)+x_{\text{补}}$   
辅助位 - MQ中“最低位” = 0时， $(ACC)+0$   
辅助位 - MQ中“最低位” = -1时， $(ACC)+[-x]_{\text{补}}$

王道考研/CSKAOYAN.COM