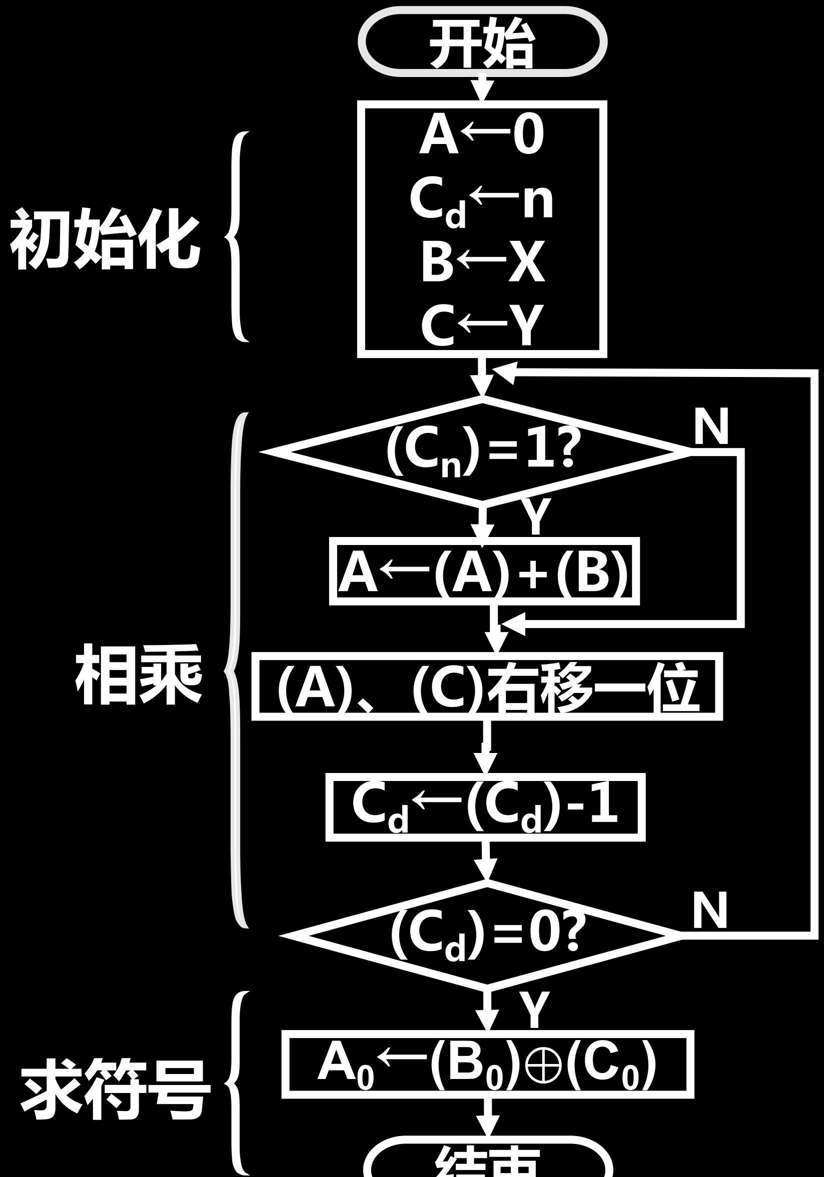
定点数乘除法

大连理工大学 赖晓晨

乘法运算

1. 定点数一位乘法
   (1) 定点原码一位乘法
   (2) 定点补码一位乘法
      - 校正法
      - 布斯法（booth）
2. 定点数二位乘法
   (1) 定点原码两位乘
   (2) 定点补码两位乘

乘法运算的控制流程



部分积循环迭代

$$z_0 = 0$$

$$z_1 = (z_0 + XY_n)2^{-1}$$

$$z_2 = (z_1 + XY_{n-1})2^{-1}$$

$$\cdots$$

$$z_{i+1} = (z_i + XY_{n-i})2^{-1}$$

$$\cdots$$

$$z_n = (z_{n-1} + XY_1)2^{-1}$$

移位- 相加

例题1

已知 $x = -0.1110$ $y = 0.1101$ 求 $[x \cdot y]_{原}$

	部分积	乘数	说明
	0.0000	1101	部分积 初态 $z_0 = 0$
	+0.1110		$+ x^*$
	0.1110		
逻辑右移	0.0111	0110	$\rightarrow$ 1，得 z_1
	+0.0000		$+ 0$
	0.0111	0	
逻辑右移	0.0011	1011	$\rightarrow$ 1，得 z_2
	+0.1110		$+ x^*$
	1.0001	10	
逻辑右移	0.1000	1101	$\rightarrow$ 1，得 z_3
	+0.1110		$+ x^*$
	1.0110	110	
逻辑右移	0.1011	0110	$\rightarrow$ 1，得 z_4

结果分析

1. 乘积的符号位 $x_0 \oplus y_0 = 1 \oplus 0 = 1$

2. 数值部分按绝对值相乘

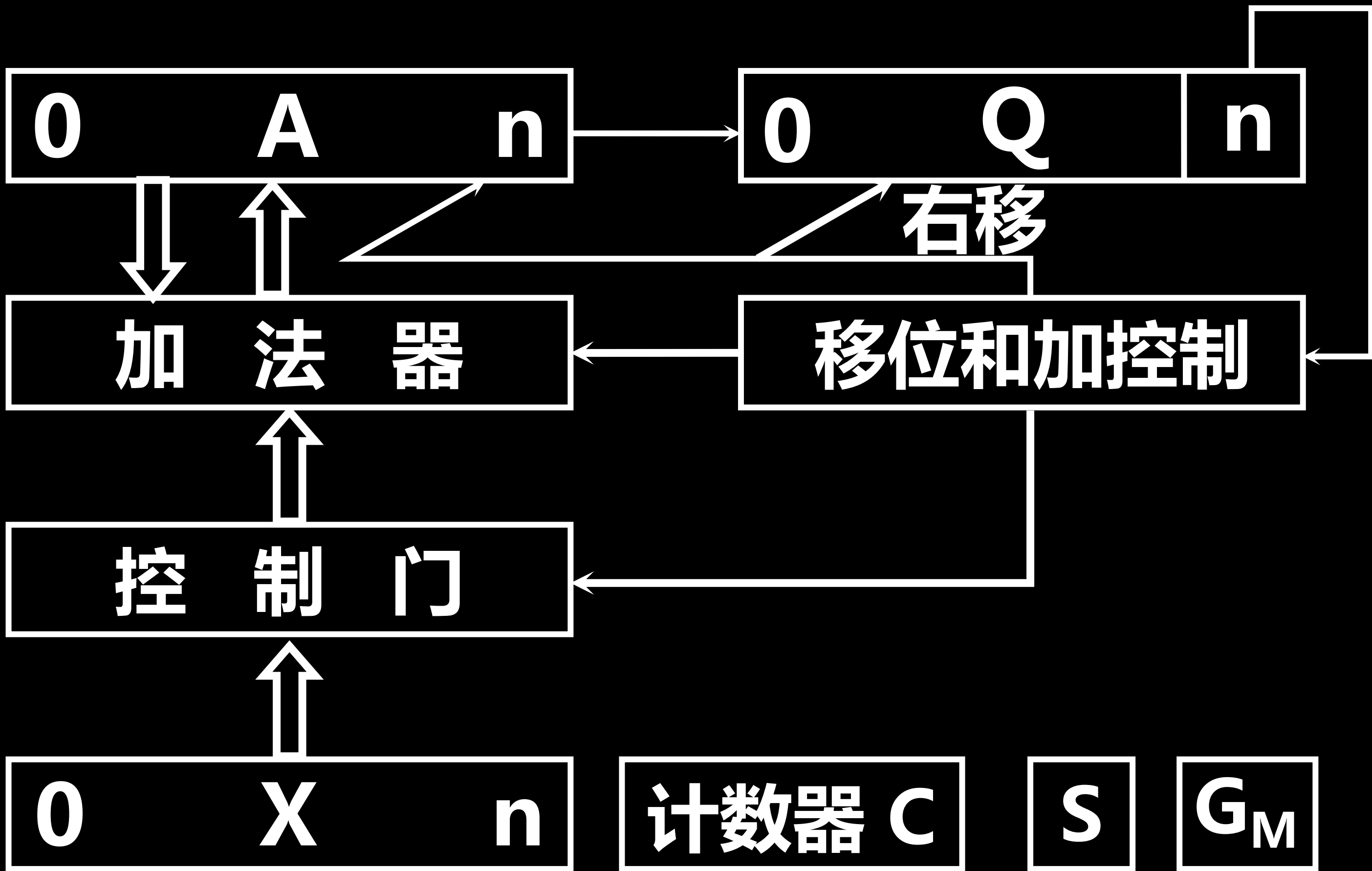
$$x^* \bullet y^* = 0.10110110$$

则 $[x \bullet y]_{原} = 1.10110110$

特点：

- 用移位的次数判断乘法是否结束
- 绝对值运算
- 逻辑移位

原码一位乘的硬件配置



A、X、Q 均 n+1 位　移位和加受末位乘数控制

定点补码一位乘法预备知识1：

补码与真值的转换关系

设 $[X]_{补}=X_0 . X_1X_2 \ldots X_n$

$$X = -X_0 + \sum_{i=1}^{n} X_i 2^{-i} = -X_0+0.X_1X_2\ldots X_n$$

当X<0时：

$$[X]_{补} = 1.X_1X_2\ldots X_n = 2+X$$

$$X = [X]_{补}-2 = 1.X_1X_2\ldots X_n - 2$$

$$= -1 + 0.X_1X_2\ldots X_n$$

定点补码一位乘法预备知识2：

补码右移

- 连同符号位将数右移一位，并保持符号位不变，相当于乘1/2。

$$[X]_{补} = X_0.X_1X_2...X_n$$

$$[(1/2)*X]_{补} = X_0.X_0X_1X_2...X_n$$

- 以5个二进制位为例，一个符号位，四个数值位：　[-6] = 1,1010

右移一位，符号位不变：

1,1101 = [-3] = [-6/2]

定点补码一位乘

- 补码一位乘用到的公式

设 $[X]_{补} = X_0 . X_1X_2 \ldots X_n$

$[Y]_{补} = Y_0 . Y_1Y_2 \ldots Y_n$

$$[X*Y]_{补} = [X]_{补}\left(-Y_0 + \sum_{i=1}^{n} Y_i 2^{-i}\right)$$

$$\sum_{i=1}^{n} Y_i 2^{-i} = (0 . Y_1Y_2 \ldots Y_n)$$

校正法实现补码一位乘

$$[X*Y]_补 = [X]_补 \left(-Y_0 + \sum_{i=1}^{n} Y_i 2^{-i}\right)$$

$$[X*Y]_补 = [X]_补 * \sum_{i=1}^{n} Y_i\, 2^{-i} - [X]_补 * Y_0$$

当y为正时，不校正；

$$[X \cdot Y]_补 = [X]_补 \cdot Y = [X]_补 (0\,.\,Y_1 Y_2 \ldots Y_n)$$

当y为负时，须校正；

$$[X \cdot Y]_补 = [X]_补 (0\,.\,Y_1 Y_2 \ldots Y_n) + [-X]_补$$

校正法实现补码一位乘

由Y的补码的最低位乘X的补码，然后右移一位，再加Y的补码的次低位乘X的补码，然后接着右移一位，再加。。。直到Y补码的最高数据位乘X补码，最后右移一位，得到XY积的补码

当y为正时，不校正；

$$[X\cdot Y]_{补}=[X]_{补}\cdot Y=[X]_{补}(0.Y_1Y_2\ldots Y_n)$$

当y为负时，须校正；

$$[X\cdot Y]_{补}=[X]_{补}(0.Y_1Y_2\ldots Y_n)+[-X]_{补}$$

例　设 X=-0.1101，Y=+0.1011 求 $[X\cdot Y]_{补}$

设 X=-0.1101，Y=-0.1011　求 $[X\cdot Y]_{补}$

例题2

	部分积P	乘数Y
	0 0. 0 0 0 0	1 0 1 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 0 1 1	
右移1位→	1 1. 1 0 0 1	1 1 0 1
+[X]	1 1. 0 0 1 1	
	1 0. 1 1 0 0	
右移1位→	1 1. 0 1 1 0	0 1 1 0
+0	0 0. 0 0 0 0	
	1 1. 0 1 1 0	
右移1位→	1 1. 1 0 1 1	0 0 1 1
+[X]	1 1. 0 0 1 1	
	1 0. 1 1 1 0	
右移1位→	1 1. 0 1 1 1	0 0 0 1
	乘积高位	乘积低位

例: 设X= - 0.1101，Y=0.1011，求X·Y。

$[X]_{补}=11.0011$

$[Y]_{补}=00.1011$

[X·Y]补=1.01110001

X·Y= -0.10001111

例题2

	部分积P	乘数Y
	0 0. 0 0 0 0	1 0 1 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 0 1 1	
右移1位→	1 1. 1 0 0 1	1 1 0 1
+[X]	1 1. 0 0 1 1	
	1 0. 1 1 0 0	
右移1位→	1 1. 0 1 1 0	0 1 1 0
+0	0 0. 0 0 0 0	
	1 1. 0 1 1 0	
右移1位→	1 1. 1 0 1 1	0 0 1 1
+[X]	1 1. 0 0 1 1	
	1 0. 1 1 1 0	
右移1位→	1 1. 0 1 1 1	0 0 0 1
	乘积高位	乘积低位

为什么用两个符号位？

Y。

$[X]_{补}=11.0011$

$[Y]_{补}=00.1011$

$[X\cdot Y]_{补}=1.01110001$

$X\cdot Y=-0.10001111$

例题2

例: 设X= - 0.1101，Y=0.1011，求X·Y。

	部分积P	乘数Y
	0 0. 0 0 0 0	1 0 1 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 0 1 1	
右移1位→	1 1. 1 0 0 1	1 1 0 1
+[X]	1 1. 0 0 1 1	
	1 0. 1 1 0 0	
右移1位→	1 1. 0	0
+0	0 0. 0 0 0	
	1 1. 0 1 1 0	
右移1位→	1 1. 1 0 1 1	
+[X]	1 1. 0 0 1 1	
	1 0. 1 1 1 0	
右移1位→	1 1. 0 1 1 1	0
	乘积高位	乘积低位

用补码运算，“加”后的和可正可负，高位符号位代表了实际极性，所以可以防止因溢出而丢掉正确的符号。

例题2

部分积P　　　　乘数Y

0 0. 0 0 0 0　　1 0 1 1 1

+[X] 1 1. 0 0 1 1

1 1. 0 0 1 1

右移1位→ 1 1. 1 0 0 1　　1 1 0 1

+[X] 1 1. 0 0 1 1

1 0. 1 1 0 0

右移1位→ 1 1. 0 1 1 0　　0 1 1 0

+0 0 0. 0 0 0 0

1 1. 0 1 1 0

右移1位→ 1 1. 1 0 1 1　　0 0 1 1

+[X] 1 1. 0 0 1 1

1 0. 1 1 1 0

右移1位→ 1 1. 0 1 1 1　　0 0 0 1

乘积高位　　乘积低位

1101，

X·Y。

011

$[Y]_{补}=00.1011$

$[X\cdot Y]_{补}=1.01110001$

$X\cdot Y= -0.10001111$

布斯（Booth）法补码一位乘

$$[X \cdot Y]_{补} = [X]_{补} \cdot (-Y_0 + \sum_{i=1}^{n} Y_i \cdot 2^{-i})$$

$$= [x](-Y_0 + Y_1 2^{-1} + Y_2 2^{-2} + \ldots + Y_n 2^{-n})$$

$$= [x][-Y_0 + (Y_1 - Y_1 2^{-1}) + (Y_2 2^{-1} - Y_2 2^{-2})$$

$$+ \ldots + (Y_n 2^{-(n-1)} - Y_n 2^{-n})]$$

$$= [x][(Y_1 - Y_0) + (Y_2 - Y_1) 2^{-1} + \ldots + (Y_n - Y_{n-1}) 2^{-(n-1)}$$

$$+ (0 - Y_n) 2^{-n}]$$

补充一项 $Y_{n+1}=0$，则上式可写为：

$$[X \cdot Y] = [X] \cdot \sum_{i=0}^{n} (Y_{i+1} - Y_i) 2^{-i}$$

Booth法控制流程

$$[X \cdot Y] = [X] \cdot \sum_{i=0}^{n} \left(Y_{i+1} - Y_i\right) 2^{-i}$$

按机器执行顺序求出每一步的部分积。

$[z_0]_补 = 0$

$[z_1]_补 = \{ [z_0]_补 + (Y_{n+1} - Y_n)[X]_补 \} 2^{-1}$　$Y_{n+1} = 0$

$[z_2]_补 = \{ [z_1]_补 + (Y_n - Y_{n-1})[X]_补 \} 2^{-1}$

$[z_i]_补 = \{ [z_{i-1}]_补 + (Y_{n-i+2} - Y_{n-i+1})[X]_补 \} 2^{-1}$

$[z_n]_补 = \{ [z_{n-1}]_补 + (Y_2 - Y_1)[X]_补 \} 2^{-1}$

$[z_{n+1}]_补 = \{ [z_n]_补 + (Y_1 - Y_0)[X]_补 \} = [X \cdot Y]_补$

Booth法控制流程

具体步骤：

- 乘数附加位置，0，即 $Y_{n+1} = 0$
- 分析[X]系数可能性，$(Y_{i+1} - Y_i)$可能有不同的组合

Y_{i+1}	Y_i	操作
0/1	0/1	P_i+0, 右移1位
1	0	$P_i+[X]_{补}$， 右移1位
0	1	$P_i+[-X]_{补}$，右移1位

- 加法重复n+1 步， 最后一次不移位，得 [X·Y] 补

例题3

	部分积P	乘数Y
	0 0. 0 0 0 0	0.1 0 1 1 0
+[-X]	0 0. 1 1 0 1	
	0 0. 1 1 0 1	
右移1位→	0 0. 0 1 1 0	1 0 1 0 1 1
+0	0 0. 0 0 0 0	
	0 0. 0 1 1 0	
右移1位→	0 0. 0 0 1 1	0 1 0 1 0 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 1 1 0	
右移1位→	1 1. 1 0 1 1	0 0 1 0 1 0
+[-X]	0 0. 1 1 0 1	
	0 0. 1 0 0 0	
右移1位→	0 0. 0 1 0 0	0 0 0 1 0 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 1 1 1	0 0 0 1 0
	乘积高位	乘积低位

例：设X= - 0.1101，Y=0.1011，求X·Y。

$[X]_{补}=11.0011$

$[Y]_{补}=00.1011$

$[-X]_{补}=00.1101$

Y_{i+1} Y_i	
00/11	P_i +0
10	$P_i + [X]_{补}$
01	$P_i + [-X]_{补}$

例题3

	部分积P	乘数Y
	0 0. 0 0 0 0	0.1 0 1 1 0
+[-X]	0 0. 1 1 0 1	
	0 0. 1 1 0 1	
右移1位→	0 0. 0 1 1 0	1 0 1 0 1 1
+0	0 0. 0 0 0 0	
	0 0. 0 1 1 0	
右移1位→	0 0. 0 0 1 1	
+[X]	1 1. 0 0 1 1	
	1 1. 0 1 1 0	
右移1位→	1 1. 1 0 1 1	0 0 1 0 1 0
+[-X]	0 0. 1 1 0 1	
	0 0. 1 0 0 0	
右移1位→	0 0. 0 1 0 0	0 0 0 1 0 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 1 1 1	0 0 0 1 0
	乘积高位	乘积低位

例：设X= - 0.1101，Y=0.1011，求X·Y。

[X·Y]补=1.01110001

X·Y=-0.10001111

Y_{i+1} Y_i	
00/11	P_i +0
10	P_i + $[X]_{补}$
01	P_i+ $[-X]_{补}$

例题3

	部分积P	乘数Y
	0 0. 0 0 0 0	0.1 0 1 1 0
+[-X]	0 0. 1 1 0 1	
	0 0. 1 1 0 1	
右移1位→	0 0. 0 1 1 0	1 0 1 0 1 1
+0	0 0. 0 0 0 0	
	0 0. 0 1 1 0	
右移1位→	0 0. 0 0 1 1	0 1 0 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 1 1 0	
右移1位→	1 1. 1 0 1 1	0 0 1
+[-X]	0 0. 1 1 0 1	
	0 0. 1 0 0 0	
右移1位→	0 0. 0 1 0 0	0 0 0 1 0 1
+[X]	1 1. 0 0 1 1	
	1 1. 0 1 1 1	0 0 0 1 0
	乘积高位	乘积低位

例：设X=－0.1101，Y=0.1011，求X·Y。

为什么Y的符号位也参与运算？
位也参与运算？
而且一共加了5
次，且最后一次
加不移位？

00/11	P_i +0
10	P_i + $[X]_{补}$
01	P_i+ $[-X]_{补}$

布斯（Booth）法

布斯(Booth)法

$$[X\cdot Y]_{补} = [x][(Y_1 - Y_0) + (Y_2 - Y_1)2^{-1} + \ldots + (Y_n - Y_{n-1})2^{-(n-1)} + (0 - Y_n)2^{-n}]$$

N+1项相加，最高项权为1

除法分类

1. 定点原码一位除
    - 恢复余数法
    - 加减交替法
2. 定点补码一位除

定点除法：被除数绝对值小于除数
被除数和除数均不为0
原码除法符号用异或求得

加减交替法

1. 是恢复余数除法的一种修正：
   - 当求得的差值为正时，商上1， 余数左移一位减除数，新的余数为：$R_i = 2R_{i-1} - Y$。
   - 当求得的差值为负时，商上0，恢复余数，左移一位减除数，新余数为：$R_i = 2(R_{i-1} + Y) - Y = 2R_{i-1} + Y$。
2. 商为1时，求下一轮余数的方法为左移余数减除数，商为0时，左移余数并且加除数。
3. 若最后一次上商为0，需恢复余数，即$R_n + Y$。

例题1

例：设X=0.1011，Y=0.1101，求X/Y。

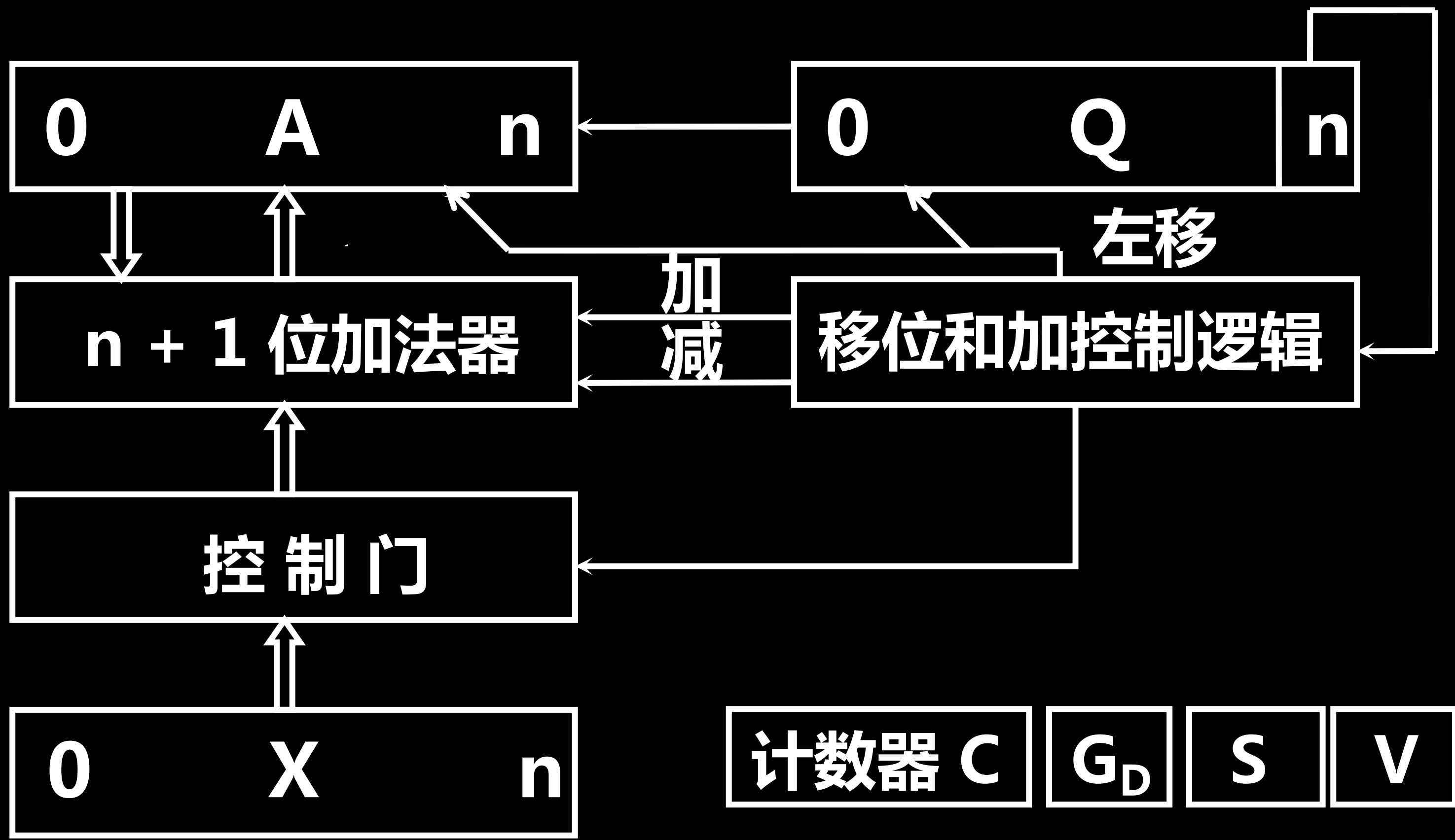
$[-Y]_{补}$ =11.0011

移 n 次，加 n+1 次

	被除数（余数）	商	说明
	0 0 1 0 1 1	0 0 0 0 0	开始
+[-Y]	1 1 0 0 1 1		-Y即+[-Y]
	1 1 1 1 1 0	0 0 0 0 0	不够减，商0
	1 1 1 1 0 0	0 0 0 0	左移
+Y	0 0 1 1 0 1		+Y
	0 0 1 0 0 1	0 0 0 0 1	够减，商1
	0 1 0 0 1 0	0 0 0 1	左移
+[-Y]	1 1 0 0 1 1		减Y
	0 0 0 1 0 1	0 0 0 1 1	够减，商1
	0 0 1 0 1 0	0 0 1 1	左移
+[-Y]	1 1 0 0 1 1		
	1 1 1 1 0 1	0 0 1 1 0	不够减， 商0
	1 1 1 0 1 0	0 1 1 0	左移
+Y	0 0 1 1 0 1		
	0 0 0 1 1 1	0 1 1 0 1	够减 ，商1

商：X/Y=0.1101　　余数：$0.0111*2^{-4}$

原码加减交替除法硬件配置



A、X、Q 均 n +1位

用 Q_n 控制加减交替

定点补码一位除法

1. 如果被除数与除数同号，用被除数减去除数；如异号，用被除数加除数。如果余数与除数同号，商为1，如果异号，商为0。该商即符号位。
2. 求商的数值部分：如上次商为1，则余数左移一位后减去除数；如上次商为0，则余数左移一位加除数。求商方法同1。如此重复执行n-1次。
3. 商的最后一位可采用恒置1法，或按2的方法再求一步，得到商的第n位。除不尽时，如商为负，要在商的最低位加1，商为正不需加1。

计算机组织与结构

大连理工大学 赖晓晨