3.1.2 乘法运算

1. 原码一位乘法运算

- 原码一位乘法的法则:
 - ①乘积的符号为被乘数的符号位与乘数的符号 位相异或;
 - ②乘积的绝对值为被乘数的绝对值与乘数的绝 对值之积。即

$$[X]_{\mathbb{R}} \times [Y]_{\mathbb{R}} = (X_0 \oplus Y_0)(|X| \times |Y|)$$

• 手工乘法运算

例 若[X]_原 = 0.1101,[Y]_原 = 1.1011,求两者之积。

解: 乘积的符号为0 ⊕1 =1

手算过程如下:

1101 × 1011 1101 部分积 1101 部分积 0000 部分积 1101 部分积 . 10001111

综上, 乘积的原码为: 1.10001111

• 机器实现思路: 部分积, 右移, 4位累加

*			I)				A	A ₀	Ao 操作	
7	0	0	0	0	0	1	0	1	1	$A_0 = 1, +X$	

		D					A	A ₀	操作
0 + 0	0	0	0	0	1	0	1	1	$A_0 = 1, +X$
0	1	1	0	1					

]	D				A	A ₀	操作
+ 0	0 1	0 1	0	0 1	1	0	1	1	$A_0 = 1, +X$
0	1 0	1 1	0 1	1 0	1	1	0	1	→右移一次

100			I)				A	A ₀	操作
107	0	0	0	0	0	1	0	1	1	$A_0 = 1, +X$
	+ 0	1	1	0	1					
	0	1	1	0	1					
	О	0	1	1	0	1	1	O	1	→右移一次
	+ 0	1	1	0	1					$A_0 = 1, +X$
	1	0	0	1	1	1	1	0	1	

			I)		,		A	A ₀	操作
	0	0	0	0	0	1	0	1	1	$A_0 = 1, +X$
+ ()	1	1	0	1					
()	1	1	0	1	ļ				V 2004
()	0	1	1	0	1	1	0	1	→右移一次
+ ()	1	1	0	1					$A_0 = 1, +X$
1	1	0	0	1	1	1	1	0	1	1.00
()	1	0	0	1	1	1	1	0	→右移一次

]	D				A	A ₀	操作
+ 0	0 1	0 1	0	0 1	1	0	1	1	$A_0 = 1, +X$
0 0 + 0	1 0 1	1 1 1	0 1 0	1 0 1	1	1	0	1	→右移一次 A ₀ =1, +X
1 0 0	0 1 0	0 0 0	1 0 0	1 1 0	1	1	0	1 0	→右移一次 A ₀ =0, +0
0	1	0	0	1	1	1	1	0	

		I)				A	A ₀	操作
0	0	0	0	0	1	0	1	1	$A_0 = 1, +X$
+ 0	1	1	0	1					
0	0	1	0	1 0	1	1	0	1	→右移一次
+ 0	1	1	0	1			25-51		$A_0 = 1, +X$
1 0 0	0 1 0	0 0 0	1 0 0	1 1 0	1	1	0	1	→右移一次 A ₀ =0, +0
0	1 0	0 1	0	1 0	1 1	1 1	1	0	→右移一次

			I)				A	A ₀	操作
+	0	0 1	0 1	0 0	0 1	1	0	1	1	$A_0=1, +X$
	0	1 0 1	1 1 1	0 1 0	1 0 1	1	1	0	1	→右移一次 A ₀ =1,+X
	1 0 0	0 1 0	0 0 0	1 0 0	1 1 0	1	1	0	1	→右移一次 A ₀ =0, +0
	0 0	1 0 1	0 1 1	0 0 0	1 0 1	1	1	1	0	→右移一次 A ₀ =1,+X
	1	0	0	0	1	1	1	1	1	

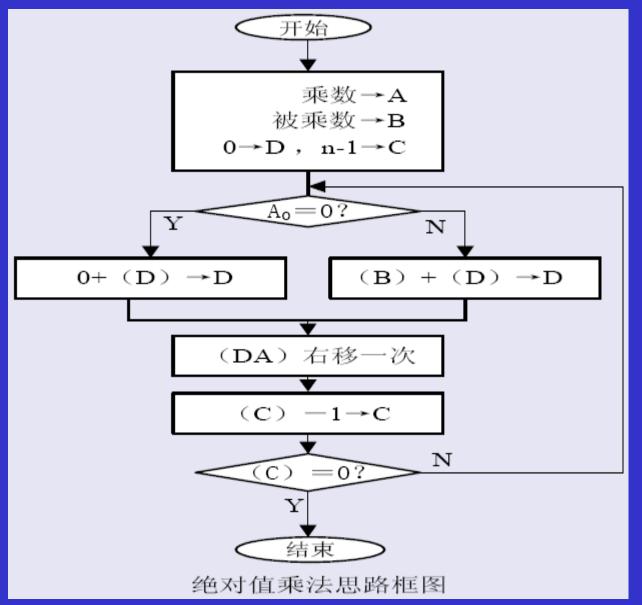
		I	D				A	A ₀	操作
+ 0 0	0 1	0	0	0	1	0	1	1	$A_0=1, +X$
0 0 + 0	1 0 1	1 1 1	0 1 0	1 0 1	1	1	0	1	→右移一次 A ₀ =1,+X
1 0 0	0 1 0	0 0 0	1 0 0	1 1 0	1	1	0	1	→右移一次 A ₀ =0, +0
0 0 + 0	1 0 1	0 1 1	0 0 0	1 0 1	1	1	1	0	→右移一次 A ₀ =1, +X
1 0	0 1	0 0	0	1 0	1 1	1 1	1 1	1	→右移一次

 $M: |X| = 1101, |Y| = 1011, X_S \oplus Y_S = 0 \oplus 1 = 1$

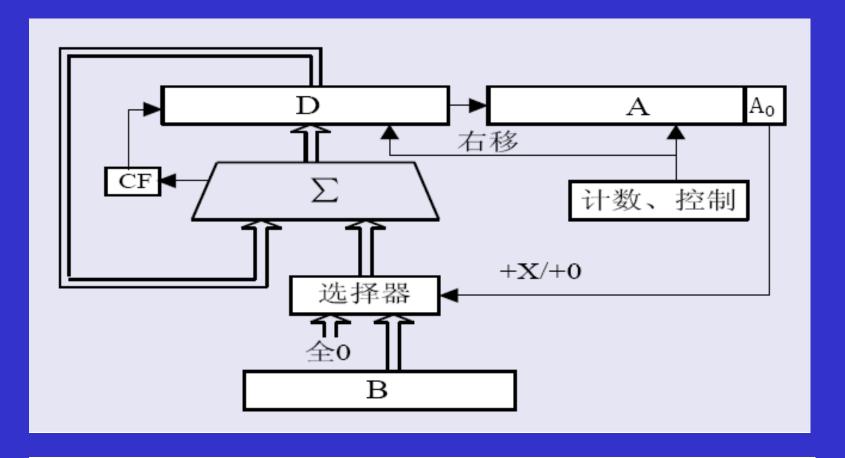
]	D				A	A ₀	操作
0 + 0	0	0	0	0	1	0	1	1	$A_0 = 1, +X$
0 0	1 0	1 1	0	1 0	1	1	0	1	→右移一次
+ 0	0	0	0	1	1	1	0	1/ /0	$A_0=1, +X$
0	1 0	0	0	1 0	1	1	1	/ ' 0	→右移一次 A ₀ =0, +0
0 + 0	1 0 1	0 1 1	0 0 0	1 0 1	1	1	1	0	→右移一次 A ₀ =1,+X
1 0	0	0	0 0	1 0	1/1	´ 1 1	1 1	1	→右移一次
拼接往	拼接符号后积为: <u>1</u> . <u>1 0 0 0 1 1 1 1</u>								

西安电子科技大学软工专业 2022年

• 实现流程



• 原码一位乘法器框图



D 部分积(初始清0), A 乘数, B 被乘数 D, A 乘积

• 原码二位乘法运算(了解):

Y_{i+1}	Y _i	C	操作
0	0	0	+0, 右移2次, C=0
0	0	1	+ X , 右移2次, C = 0
0	1	0	+ X , 右移2次, C = 0
0	1	1	+2 X ,右移2次,C = 0
1	0	0	+2 X , 右移2次, C = 0
1	0	1	- X , 右移2次, C = 1
1	1	0	- X , 右移2次, C = 1
1	1	1	+0, 右移2次, C=1

原码二位乘法的法则表

2. 补码一位乘法运算

布斯(Booth)法

假定被乘数X和乘数Y均为用补码表示的纯小数,分别为:

$$[X]_{ih} = X_0 . X_{-1}X_{-2}...X_{-(n-1)}$$

 $[Y]_{ih} = Y_0 . Y_{-1}Y_{-2}...Y_{-(n-1)}$ 附加位

其中 X_0 、 Y_0 是它们的符号位。

则布斯法补码一位乘法的算法公式为:

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

$$+ (Y_{-3} - Y_{-2})2^{-2} + \dots + (Y_{-(n-1)} - Y_{-(n-2)})2^{-(n-2)}$$

$$+ (0 - Y_{-(n-1)})2^{-(n-1)}]$$

$$Y' = (Y_{-1} - Y_0). (Y_{-2} - Y_{-1}) (Y_{-3} - Y_{-2}) \dots (Y_{-(n-1)} - Y_{-(n-2)})^{\frac{1}{2}} (0 - Y_{-(n-1)})$$

$$\mathbb{N}: [X \cdot Y]_{\frac{1}{2}h} = [X]_{\frac{1}{2}h} \cdot Y'$$

$\mathbf{Y_{i}}$ $\mathbf{Y_{i-1}}$	$\mathbf{Y}_{i-1} - \mathbf{Y}_{i}$	操作
0 0	0	+0,右移一次
0 1	1	+ [X] _补 ,右移一次
1 0	- 1	+ [-X] _补 ,右移一次
1 1	0	+0,右移一次

乘数相邻两位的操作规律

• 布斯法的运算法则:

- ①乘数与被乘数均用补码表示,连同符号位一起参加运算。
- ②乘数最低位后增加一个附加位(可用A₋₁表示),初始设定为0。
- ③从附加位开始,依据上表所示的操作规律, 完成运算。

解: $[X]_{i} = 00.1010$, $[Y]_{i} = 11.0011$,

符	号		I)			A	A			A-1	操作
0	0	0	0	0	0	1	0	0	1	1	0	

符	号		I)			A	4			A_{-1}	操作
О	0	О	О	0	0	1	О	0	1	1	0	
1	1	О	1	1	О		3					+ [-x] *
1	1	0	1	1	0							

符	号		I)			£	A			A-1	操 作
О	0	0	0	0	0	1	0	0	1	1	0	
1	1	0	1	1	0	ļ						+ [-x] *h
1	1	О	1	1	0							
1	1	1	О	1	1	0	1	О	О	1	1	右移一位

	符	号		I)			A	4			A_{-1}	操作
	0	0	О	О	О	0	1	О	О	1	1	0	
	1	1	О	1	1	О	L						+ [-x] *
•	1	1	О	1	1	0							
	1	1	1	О	1	1	0	1	О	О	1	1	右移一位
	О	О	О	О	О	О							+0
	1	1	1	0	1	1			,				

ぞ	子号		I)			Ā	4			A_{-1}	操作
0	О	0	0	0	0	1	0	0	1	1	0	
1	1	О	1	1	О	ļ	2					+ [-x] *
1	1	0	1	1	0							
1	1	1	О	1	1	О	1	О	О	1	1	右移一位
О	0	О	О	О	О							+0
1	1	1	О	1	1			7				
1	1	1	1	О	1	1	О	1	0	0	1	右移一位

解: $[X]_{\stackrel{?}{\uparrow}} = 00.1010$, $[Y]_{\stackrel{?}{\uparrow}} = 11.0011$,

•	符	号		I)			£	4			A_{-1}	操作
	0	0	0	О	0	О	1	0	0	1	1	0	
	1	1	О	1	1	О	ļ	2					+ [-x] *
	1	1	О	1	1	0							
	1	1	1	О	1	1	О	1	О	О	1	1	右移一位
	О	О	О	О	О	О							+0
	1	1	1	О	1	1			1				
	1	1	1	1	О	1	1	О	1	О	0	1	右移一位
	О	0	1	О	1	О							+ [x] *
	0	0	0	1	1	1			ļ,				

	符	号		I)			A	4			A-1	操作
'	0	0	0	О	0	0	1	О	0	1	1	0	
	1	1	О	1	1	О	ļ						+ [-x] *
·	1	1	О	1	1	0							
	1	1	1	О	1	1	О	1	О	О	1	1	右移一位
	О	0	О	О	О	О							+0
,	1	1	1	О	1	1			3				
	1	1	1	1	О	1	1	О	1	О	0	1	右移一位
	О	О	1	О	1	О							+ [X] *h
	0	0	0	1	1	1			ļ,				
	О	О	О	О	1	1	1	1	0	1	О	0	右移一位

符	号		I)			£	4			$A_{\textbf{-1}}$	操作
0	0	0	О	0	0	1	0	О	1	1	0	
1	1	О	1	1	О	ļ	2					+ [-x] *
1	1	О	1	1	О							
1	1	1	О	1	1	О	1	О	О	1	1	右移一位
О	О	О	О	О	О							+0
1	1	1	О	1	1			1				
1	1	1	1	О	1	1	О	1	О	0	1	右移一位
О	О	1	О	1	О							+ [x] *
0	0	О	1	1	1			ļ,				
О	О	О	О	1	1	1	1	0	1	0	0	右移一位
О	О	О	О	О	О							+0
О	О	0	О	1	1					1		

解: $[X]_{\dot{\uparrow}} = 00.1010$, $[Y]_{\dot{\uparrow}} = 11.0011$,

•	符	号		I)			A	A.			A-1	操作
	О	О	О	О	О	О	1	О	О	1	1	0	
	1	1	О	1	1	О	L,						+ [-x] *h
	1	1	О	1	1	О							
	1	1	1	О	1	1	О	1	О	О	1	1	右移一位
	О	О	О	О	О	О							+0
	1	1	1	О	1	1			1				
	1	1	1	1	О	1	1	О	1	О	0	1	右移一位
	О	О	1	О	1	О							+ [x] *
	О	0	О	1	1	1							
	О	О	О	О	1	1	1	1	0	1	O	0	右移一位
	О	О	О	О	О	О							+0
	0	О	0	О	1	1							
	О	О	О	О	О	1	1	1	1	О	1	О	右移一位

解: $[X]_{i} = 00.1010$, $[Y]_{i} = 11.0011$,

-													
	符	号		I)			£	4			A_{-1}	操作
	0	0	О	О	0	0	1	О	0	1	1	0	
	1	1	0	1	1	0							+ [-x] *
	1	1	О	1	1	0							
	1	1	1	О	1	1	О	1	О	О	1	1	右移一位
	О	О	О	О	О	О							+0
	1	1	1	О	1	1			,				
	1	1	1	1	О	1	1	О	1	О	0	1	右移一位
	0	О	1	О	1	О							+ [X] *h
	0	0	О	1	1	1			ļ,				
	О	О	О	О	1	1	1	1	0	1	O	0	右移一位
	0	О	О	О	О	О							+0
	0	0	0	0	1	1			l.		1		
	0	О	О	О	О	1	1	1	1	О	1	О	右移一位
	1	1	О	1	1	О							+ [-x] *
	1	1	0	1	1	1					l ₁		

解: $[X]_{i} = 00.1010$, $[Y]_{i} = 11.0011$,

符	号		I)			1	4			A_{-1}	操作
О	0	О	О	0	0	1	0	0	1	1	0	
1	1	О	1	1	О	L	3					+ [-x] *
1	1	О	1	1	О							
1	1	1	О	1	1	0	1	О	О	1	1	右移一位
О	О	О	О	О	О							+0
1	1	1	О	1	1		Ī	7				
1	1	1	1	О	1	1	О	1	О	0	1	右移一位
О	О	1	О	1	О							+ [x] *
О	0	О	1	1	1			ļ,				
О	О	О	О	1	1	1	1	0	1	0	0	右移一位
О	О	О	О	О	О							+0
О	0	О	О	1	1			l.				
О	О	О	О	О	1	1	1	1	О	1	О	右移一位
1	1	О	1	1	О							+ [-x] *
1	1	О	1	1	1							
1	1	1	О	1	1	1	1	1	1	0		右移一位

解: $[X]_{i} = 00.1010$, $[Y]_{i} = 11.0011$,

符	号		I)			£	4			A_{-1}	操作
0	0	О	О	О	О	1	О	О	1	1	0	
1	1	О	1	1	О	L	3					+ [-x] *
1	1	О	1	1	О							
1	1	1	О	1	1	0	1	О	О	1	1	右移一位
О	О	О	О	О	О							+0
1	1	1	О	1	1		<u></u>	7				
1	1	1	1	О	1	1	О	1	О	0	1	右移一位
О	О	1	О	1	О							+ [X] *h
0	0	О	1	1	1			l				
О	О	О	О	1	1	1	1	0	1	0	0	右移一位
О	О	О	О	О	О							+0
0	0	О	О	1	1							
О	О	О	О	О	1	1	1	1	О	1	О	右移一位
1	1	0	1	1	О							+ [-x] *
1	1	О	1	1	1					l		
1	1	1	О	1	1	1	1	1	1	0		右移一位

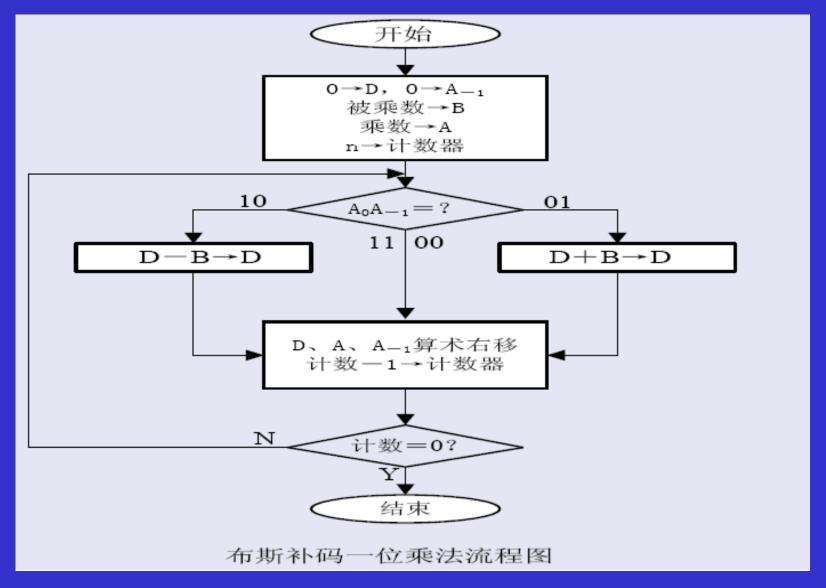
解: $[X]_{\dot{\uparrow}} = 00.1010$, $[Y]_{\dot{\uparrow}} = 11.0011$,

而[-X]_补 = 11.0110。

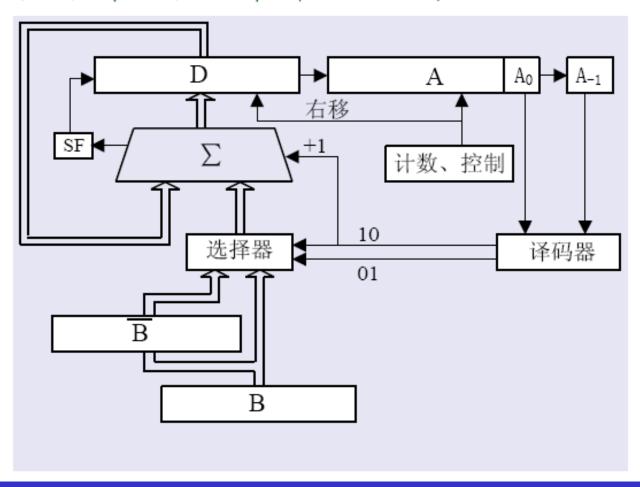
	符	号	D				A					A_{-1}	操 作		
	О	0	О	О	О	0	1	О	О	1	1	0			
	1	1	О	1	1	О	L	,					+ [-x] *		
	1	1	О	1	1	0									
	1	1	1	О	1	1	0	1	О	О	1	1	右移一位		
	О	О	О	О	О	О							+0		
	1	1	1	0	1	1		<u></u>	,						
	1	1	1	1	О	1	1	0	1	О	0	1	右移一位		
	О	О	1	О	1	О							+ [x] *h		
	О	0	О	1	1	1			ļ,						
	О	О	О	О	1	1	1	1	0	1	0	0	右移一位		
	О	О	О	О	О	О							+0		
	О	0	О	О	1	1			l.						
	О	О	О	О	О	1	1	1	1	О	1	О	右移一位		
	1	1	О	1	1	О							+ [-x] *		
	1	1	О	1	1	1					ļ				
	1	1	1	О	1	1	1	1	1	1	0		右移一位		

 $[X \cdot Y] *= 1.011111110$

•运算过程(课后学习)



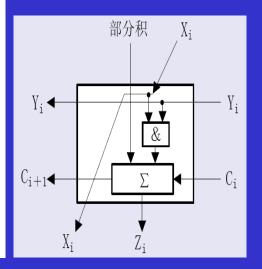
- 补码乘法器框图 (课后学习)
 - 补码一位乘法器(布斯法)框图



阵列乘法器 (课后学习)

阵列乘法器(原码一位乘)
 设X=X₃X₂X₁X₀, Y=Y₃Y₂Y₁Y₀, 计算X·Y=?

			Х3	X_2	X_1	X_0
		X	Y ₃	Y_2	Y_1	Y_0
			X ₃ Y ₀	X ₂ Y ₀	X_1Y_0	X ₀ Y ₀
		$X_3 Y_1$	$X_2 Y_1$	X_1Y_1	$X_0 Y_1$	
	X ₃ Y ₂	$X_2 Y_2$	X_1Y_2	$X_0 Y_2$		
X3 Y3	$X_2 Y_3$	X_1Y_3	$X_0 Y_3$			
Z ₆	Z_5	Z4	\mathbb{Z}_3	Z_2	Z_1	Z ₀



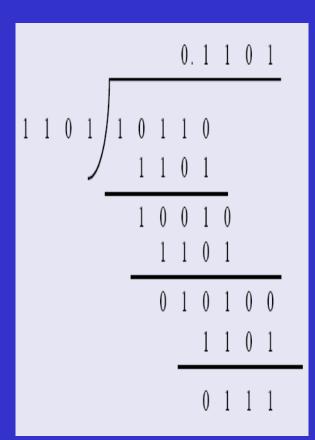
3.1.3 除法运算

1. 原码除法运算

- 原码除法的法则
- ①除数≠0;定点纯小数时,|被除数|<|除数|; 定点纯整数时,|被除数|>|除数|。
- ②与原码乘法类似的是原码除法商的符号和商的值也是分别处理的。
- ③商的符号等于被除数的符号与除数的符号相异或,商的值就等于被除数的绝对值除以除数的绝对值。
- ④最后将商的符号与商的值拼接在一起就得到原码除 法的商。

• 手工除法过程

解:由题意可知,假定被除数X和除数Y均为正数。 未来商的符号也为正。



• 恢复余数法

- 符号位单独处理。
- 对于定点纯小数,被除数左移一位, 减除数,

若够减, 上商为1; 若不够减,上商为0,同时加除数 (恢复余数)。

- 余数左移一位,减除数,若够减,上商为1;
 若不够减,上商为0,同时加除数(恢复余数)。
- 重复上述过程直到除尽或精度达到要求。
- 拼接商符得到商,即可获得除法的结果。

若被除数X = -0.10001011, 除数Y = 0.1110,

试利用原码恢复余数法求 商及余数。

解:该例满足

|X| < |Y|,

且 | Y | ≠0

已知[X]_原 = 1.10001011,

 $[Y]_{\text{R}} = 0.1110$.

商符 = 1 ⊕ 0 = 1;

符	号		ì	披除	数	(余	数)			商	操作
0	0	1	0	0	0	1	0	1	1	0	

若被除数X = -0.10001011, 除数Y = 0.1110,

试利用原码恢复余数法求 商及余数。

解:该例满足

|X| < |Y|,

且 | Y | ≠0

已知[X]_原 = 1.10001011,

 $[Y]_{\text{R}} = 0.1110$.

商符 = 1 ⊕ 0 = 1;

符	号		Ì	披除	数	(余	数)			商	操作
0	0	1	0	0	0	1	0	1	1	0	
0		0		0	1	0	1	1	0		左移一位

若被除数X = -0.10001011, 除数Y = 0.1110, 试利用原码恢复余数法求 商及余数。

商及余数。
解:该例满足
X < Y ,
且 Y ≠0
已知[X] _原 = 1.10001011,
[Y] _原 = 0.1110。
商符 = 1 ⊕ 0 = 1;
绝对值相除见表。

符	号		Ì	披除	数	(余	数)			商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0 0 1	1	0	1	1	0		左移一位
1	1	0	0	1	0						- Y
0	0	0	0	1	1	0	1	1	0	1	够减,商为1

若被除数X = -0.10001011, 除数Y = 0.1110, 试利用原码恢复余数法求 商及余数。

解:该例满足 |X|<|Y|, 且|Y|≠0 已知[X]_原 = 1.10001011, [Y]_原 = 0.1110。 商符 = 1 ⊕ 0 = 1;

符	号		Ì	披除	数	(余	数)			商	操作
0 0 1	0 1 1	1 0 0	0	0 0 1	1	1	0	1	0	0	左移一位 - Y
0	0	0	0	1	1 0	0		1	0	1	够减,商为1 左移一位

若被除数X = -0.10001011, 除数Y = 0.1110, 试利用原码恢复余数法求 商及余数。 解:该例满足

| X | < | Y | , 且 | Y | ≠0 已知[X]_原 = 1.10001011, [Y]_原 = 0.1110。

商符 = 1 ⊕ 0 = 1; 绝对值相除见表。

符	号		Ì	披除	数	(余	数)			商	操作
0 0 1	0 1 1	1 0 0	0 0 0	0 0 1	0 1 0	1	0	1	0	0	左移一位 - Y
0 0 1	0 0 1	0 0 0	0 1 0	1 1 1	1 0 0	0	1	0	0	1	够减,商为1 左移一位 Y
1	1	1	0	0	0	1	1	0	1	0	不够减,商为0

若被除数X = -0.10001011, 除数Y=0.1110, 试利用原码恢复余数法求 商及余数。 解:该例满足 |X| < |Y|且 | Y | ≠0 已知[X]_原 = 1.10001011, $[Y]_{\text{R}} = 0.1110$. 商符 = 1 ⊕ 0 = 1;

符	号		Ì	披除	数	(余	数)			商	操作
0 0 1	0 1 1	1 0 0	0 0 0	0 0 1	0 1 0	1	0	1	0	0	左移一位 - Y
0 0 1	0 0 1	0 0 0	0 1 0	1 1 1	1 0 0	0	1	0	0	1	够减,商为1 左移一位 Y
1 0	1	1	0 1	0 1	0 0	1	1	0	1	0	不够减,商为0 + Y

若被除数X = -0.10001011, 除数Y=0.1110, 试利用原码恢复余数法求 商及余数。 解:该例满足 |X| < |Y|且 | Y | ≠0 已知[X]_原 = 1.10001011, $[Y]_{\text{R}} = 0.1110$. 商符 = 1 ⊕ 0 = 1;

符	号		Ì	披除	数	(余	数)			商	操作
0 0 1	0 1 1	1 0 0	0 0 0	0 0 1	0 1 0	1	0	1	0	0	左移一位 Y
0 0 1	0 0 1	0 0 0	0 1 0	1 1 1	1 0 0	0 1	1	0	0	1	够减,商为1 左移一位 Y
1 0	1 0	1 1	0 1	0 1	0	1	1	0	1	0	不够减, 商为0 + Y
0	0	0	1	1	0	1	1	0	1	0	恢复余数

若被除数X = -0.10001011, 除数Y = 0.1110, 试利用原码恢复余数法求 商及余数。 解:该例满足

| X | < | Y | , 且 | Y | ≠0 已知[X]_原 = 1.10001011, [Y]_原 = 0.1110。 商符 = 1 ⊕ 0 = 1;

符	号		Ì	披除	数	(余	数)			商	操作
0	0	1	0	0	0	1	0	1	1	0	
0	1	0	0	0	1	0	1	1	0		左移一位
1	1	0	0	1	0						- Y
0	0	0	0	1	1	0	1	1	0	1	够减,商为1
0_	0	0	1	1	0	1	_1	0	1		左移一位
1	1	0	0	1	0						- Y
1	1	1	0	0	0	1	1	0	1	0	不够减,商为0
0	0	1	1	1	0						+ Y
0	0	0	1	1	0	1	1	0	1	0	恢复余数
0	0	1	1	0	1	1	0	1	0		左移一位

若被除数X = -0.10001011, 除数Y = 0.1110, 试利用原码恢复余数法求 商及余数。

解:该例满足 |X|<|Y|, 且|Y|≠0 已知[X]_原 = 1.10001011, [Y]_原 = 0.1110。

> 商符 = 1 ⊕ 0 = 1; 绝对值相除见表。

符号			Ì	披除	数	(余	数)			商	操作
0 (- 1	1	0	0	0	1	0	1	1	0	左移一位
1 1	- 1	0	0	1	0	Ü	1	1	V		- Y
0 ()	0	0	1	1	0	1	1	0	1	够减,商为1 左移一位
1 1		0	0	1	0			Š	1		- Y
1 1		1	0	0	0	1	1	0	1	0	不够减,商为0 + Y
0 (+	0	1	1	0	1	1	0	1	0	恢复余数
0 (1	1	0	1	1	0	1	0		左移一位 - Y
1 1		1	1	1	1	1	0	1	0	0	不够减,商为0

除数Y=0.1110, 试利用原码恢复余数法求 商及余数。 解:该例满足 |X|<|Y|, 且|Y|≠0 已知[X]_原=1.10001011, [Y]_原=0.1110。 商符=1⊕0=1;

绝对值相除见表。

若被除数X = -0.10001011,

符	号		Ì	披除	数	(余	数)			商	操作
0	0	1	0	0	0	1	0	1	1	0	Lord D
0	1	0	0	0	1	0	1	1	0		左移一位 - Y
1	1	0	0	1	0						1
0	0	0	0	1	1	0	1	1	0	1	够减,商为1
0	0	0	_1_	1	0	1	1	0	1		左移一位
1	1	0	0	1	0						- Y
1	1	1	0	0	0	1	1	0	1	0	不够减,商为0
0	0	1	1	1	0						+ Y
0	0	0	1	1	0	1	1	0	1	0	恢复余数
0	0	1	1	0	1	1	0	1	0		左移一位
1	1	0	0	1	0						- Y
1	1	1	1	1	1	1	0	1	0	0	不够减,商为0
0	0	1	1	1	0						+ Y
0	0	1	1	0	1	1	0	1	0	0	恢复余数

若被除数X = -0.10001011, 除数Y = 0.1110, 试利用原码恢复余数法求 商及余数。

解:该例满足 |X|<|Y|, 且|Y|≠0 已知[X]_原 = 1.10001011, [Y]_原 = 0.1110。 商符=1⊕0=1; 绝对值相除见表。

符号	被除数 (余数)	商操作
0 0	1 0 0 0 1 0 1 <u>1</u>	_0
0 1	0 0 0 1 0 1 1 0	左移一位
1 1	0 0 1 0	Y
0 0	0 0 1 1 0 1 1 0	1 够减,商为1
0 0	0 1 1 0 1 1 0 1	左移一位
1 1	0 0 1 0	
1 1	1 0 0 0 1 1 0 1	0 不够减, 商为0
0 0	1 1 1 0	+ Y
0 0	0 1 1 0 1 1 0 1	0 恢复余数
0 0	1 1 0 1 1 0 1 0	左移一位
1 1	0 0 1 0	
1 1	1 1 1 1 1 0 1 0	0 不够减, 商为0
0 0	1 1 1 0	+ Y
0 0	1 1 0 1 1 0 1 0	0 恢复余数
0 1	1 0 1 1 0 1 0 0	左移一位

若被除数X = -0.10001011, 除数Y = 0.1110, 试利用原码恢复余数法求 商及余数。

解:该例满足 |X|<|Y|, 且|Y|≠0 已知[X]_原 = 1.10001011, [Y]_原 = 0.1110。 商符=1⊕0=1; 绝对值相除见表。

符号	를		Ì	披除	数	(余	数)			商	操作
0 0 1	0 1 1	1 0 0	0 0 0	0 0 1	0 1 0	1 0	0	1	0	0	左移一位 Y
0 0 1	0 0 1	0 0 0	0 1 0	1 1 1	1 0 0	0 1	1 1	0	0	1	够减,商为1 左移一位
1 0	1	1	0 1	0 1	0	1	1	0	1	0	不够减,商为0 + Y
0 0 1	0 0 1	0 1 0	1 1 0	1 0 1	0 1 0	1	0	0	1	0	恢复余数 左移一位
1	1	1 1	1 1	1 1	1	1	0	1	0	0	不够减,商为0 + Y
0 0 1	0 1 1	1 1 0	1 0 0	0 1 1	1 1 0	1 0	0	1 0	0	0	恢复余数 左移一位 Y
0	0	1	1	0	1	0	1	0	0	1	够减,商为1

若被除数X = -0.10001011, 除数Y = 0.1110,

试利用原码恢复余数法求 商及余数。

解:该例满足

|X| < |Y|,

且 | Y | ≠0

已知[X]_原 = 1.10001011,

[Y]_原 = 0.1110。

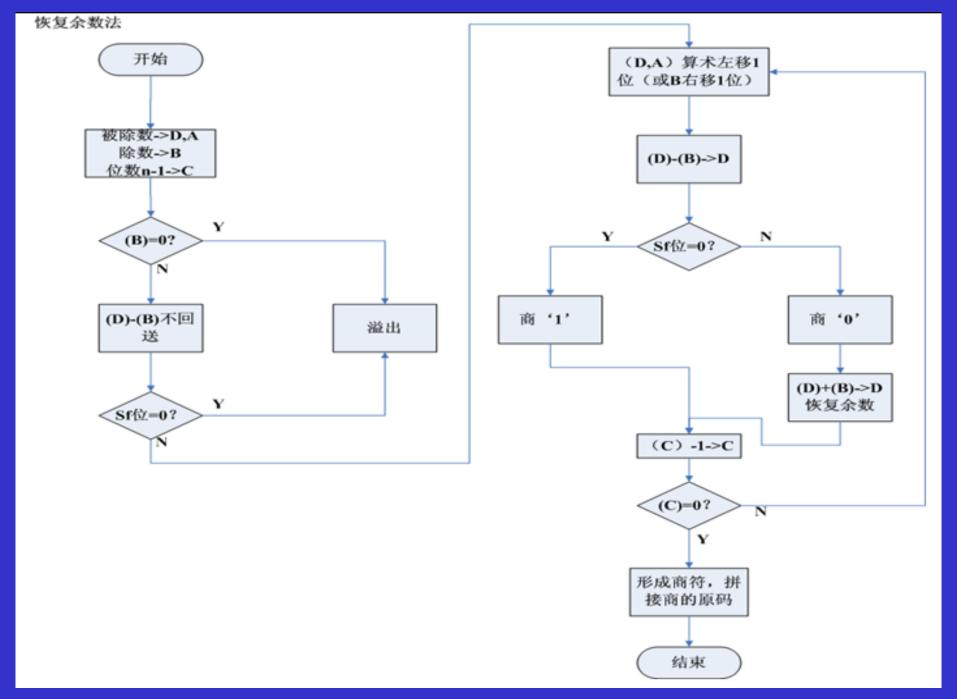
商符 = 1 ⊕ 0 = <u>1</u>;

绝对值相除见表。

商 = <u>1</u>.1001 <

余数 = 1.1101 × 2⁻⁴

符号	被除数 (余数)	商操作
0 0	1 0 0 0 1 0 1 <u>1</u>	_ 0
0 1	0 0 0 1 0 1 1 0	左移一位
1 1	0 0 1 0	- Y
0 0	0 0 1 1 0 1 1 0	1 够减,商为1
0 0	0 1 1 0 1 1 0 1	左移一位
1 1	0 0 1 0	
1 1	1 0 0 0 1 1 0 1	0 不够减, 商为0
0 0	1 1 1 0	+ Y
0 0 0 0 1 1	0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 0 0 0 1 0	0 恢复余数 左移一位 - Y
1 1	1 1 1 1 1 0 1 0	0 不够减, 商为0
0 0	1 1 1 0	+ Y
0 0	1 1 0 1 1 0 1 0	0 恢复余数
0 1	1 0 1 1 0 1 0 0	左移一位
1 1	0 0 1 0	Y
0 0	1 1 0 1 0 1 0 0	1 够减,商为1



• 加减交替法

假定第i次余数减除数(用B表示)得的余数为R,

当R≥0时,商1,应左移一位,即 2R, 然后(第i+1次)减去除数,就是 2R -B

当R<0时, 商0, 应恢复余数, 即 R+B, 再左移一位, 即 2(R+B), 然后(第i+1次)减去除数,就是 2(R+B)-B = 2R +B

- 即当第i次余数减除数的余数R<0时,不用加除数 来恢复余数,而只是将其左移一次,变为2R,到 第i+1次运算时将其加除数,也就是2R+B。
- 加减交替法的运算法则
 - ①若余数R≥0,则商上1,左移一次,减除数;
 - ②若余数R<0,则商上0,左移一次,加除数。

•	符	号		À	波除	数	(余	数)		商	操作	
	0	0	1	0	0	0	1	0	1	1	0	

符	号		剂	波除	数	(余	数)			商	操作
0	0	1	0	0	0	1	0	1	1	О	
0	1	0	0	0	1	O	1	1	0		左移一位

符	号		À	波除	数	(余	数)			商	操作
0	0	1	0	0	0	1	0	1	1	0	TA 12
О	1	0	0	0	1	0	1	1	0		左移一位
1	1	О	O	1	O						- Y

1	符	号		À	波除	数	(余	数)		商	操作	
)	0	1	0	0	0	1	0	1	1	0	
	C	1	0	0	0 1	1	0	1	1	0		左移一位
1	1	1	О	0	1	0						- Y
)	0	0	0	1	1	0	1	1	0	1	R≥0,商为1

符	号		À	波除	数	(余	数)			商	操作
0				O		1	0	1	1	О	
0	1	0	0	О	1	0	1	1	0		左移一位
_ 1	1	О	0	1	О						$ \mid$ Y \mid
0	0	О	0	1	1	0	1	1	0	1	R≥0,商为1
0	0	0	1	1	0		1		1		左移一位

	符	号		À	波除	数	(余	数)			商	操作
	0 0 1	0 1 1	1 0 0	0 0 0			1 0	0	1 1	0	0	左移一位 一丨Y丨
_	0 0 1	0 0 1	0 0 0	0 1 0	1 1 1	1 0 0	0	1	0	0	1	R≥0,商为1 左移一位 - Y

符	号		À	波除	数	(余	数)			商	操作
0	O	1	O	O	O	1	O	1	1	О	
О	1	О	0	O	1	O	1	1	0		左移一位
1	1	О	О	1	O						- Y
0	О	О	О	1	1	0	1	1	0	1	R≥0,商为1
О	0	0	1	1	0	1	1	0	1		左移一位
_ 1	1	О	О	1	О						- Y
1	1	1	0	0	0	1	1	0	1	0	R<0,商为0

符	号		À	波除	数	(余	数)			商	操作
0	0	1	0	0	О	1	0	1	1	0	1
О	1	O	0	O	1	0	1	1	0		左移一位
_ 1	1	О	О	1	О						- Y
0	O	О	О	1	1	0	1	1	0	1	R≥0,商为1
О	O	0	1	1	O	1	1	0	1		左移一位
1	1	О	О	1	О						- Y
1	1	1	О	0	О	1	1	0	1	О	R<0,商为0
1	1	О	O	O	1		0	1	О		左移一位

符号	被除数(余数)	商操作
0 0 0 1 1 1	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	0 左移一位
0 0 0 0 1 1	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	1 R≥0,商为1 左移一位 Y
1 1 1 1 0 0	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	0 R<0,商为0 左移一位 + Y

符	号		À	波除	数	(余	数)			商	操作
0 0 1	0 1 1	1 0 0	0 0 0	0 0 1	0 1 0	1 0	0	1	0	0	左移一位
0 0 1	0 0 1	0 0 0	0 1 0	1 1 1	1 0 0	0	1	0	0	1	R≥0,商为1 左移一位
1 1 0	1 1 0	1 0 1	0 0 1	0 0 1	0 1 0	1 1	1 0	0	1	0	R<0,商为0 左移一位 + Y
1	1	1	1	1	1	1	0	1	0	0	R<0,商为0

符号		被除数(余数)									操作
0	0	1	О	О	0	1	0	1	1	О	
0	1	0	0	0	1	O	1	1	0		左移一位
_ 1	1	О	О	1	О						- Y
0	0	О	0	1	1	0	1	1	0	1	R≥0,商为1
0	O	0	1	1	0	1	1	0	1		左移一位
1	1	О	О	1	О						- Y
1	1	1	0	0	О	1	1	0	1	0	R<0,商为0
1	1	0	0	0	1	1	0	1	0		左移一位
0	O	1	1	1	0						+ Y
1	1	1	1	1	1	1	0	1	0	О	R<0,商为0
1	1	1	1	1	1	0	1	O	0		左移一位

符号		被除数(余数)									操作
0	0	1	О	О	O	1	О	1	1	О	
0	1	0	0	0	1	0	1	1	0		左移一位
_ 1	1	О	О	1	О						- Y
0	0	О	О	1	1	0	1	1	0	1	R≥0,商为1
0	O	0	1	1	0	1	1	0	1		左移一位
_ 1	1	О	О	1	О						- Y
1	1	1	0	0	0	1	1	0	1	0	R<0,商为0
1	1	0	0	0	1	1	0	1	0		左移一位
0	О	1	1	1	О						+ Y
1	1	1	1	1	1	1	0	1	0	О	R<0,商为0
1	1	1	1	1	1	0	1	0	0		左移一位
0	О	1	1	1	О						+ Y

解: [X]_原 = 1.10001011, [Y]_原 = 0.1110, 商符 = 1 ⊕ 0 = 1。

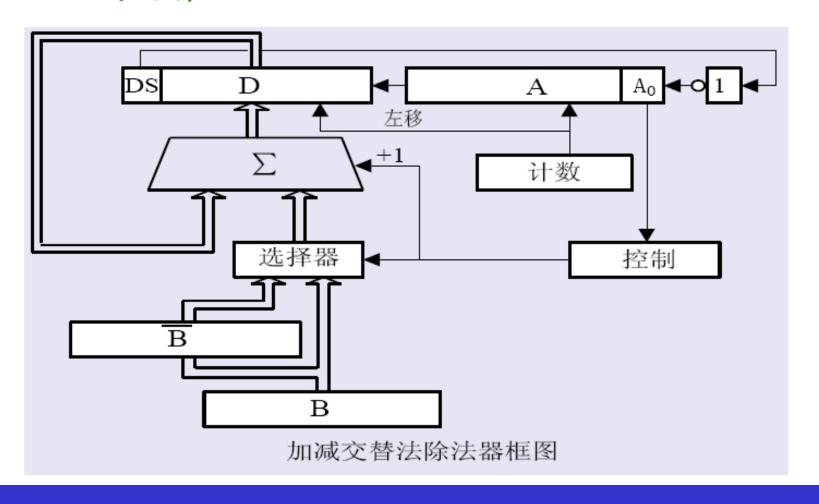
符号		被除数(余数)									操作
0	O	1	O	O	O	1	0	1	1	О	
0	1	0	O	O	1	O	1	1	0		左移一位
1	1	О	О	1	0						- Y
0	O	О	O	1	1	0	1	1	0	1	R≥0,商为1
0	O	0	1	1	O	1	1	0	1		左移一位
_ 1	1	О	О	1	0						- Y
1	1	1	0	0	0	1	1	0	1	О	R<0,商为0
1	1	0	O	0	1	1	0	1	0		左移一位
0	О	1	1	1	О						+ Y
1	1	1	1	1	1	1	0	1	0	O	R<0,商为0
1	1	1	1	1	1	0	1	0	0		左移一位
О	О	1	1	1	О						+ Y
0	0	1	1	0	1	0	1	0	0	1	R≥0,商为1

商 = 1.1001 余数 = ?

问题:

当加减交替法运算结束肘,如果末位商 为0,这时的余数是错误的「负数」。 如何获得正确的余数?

• 加减交替法除法器



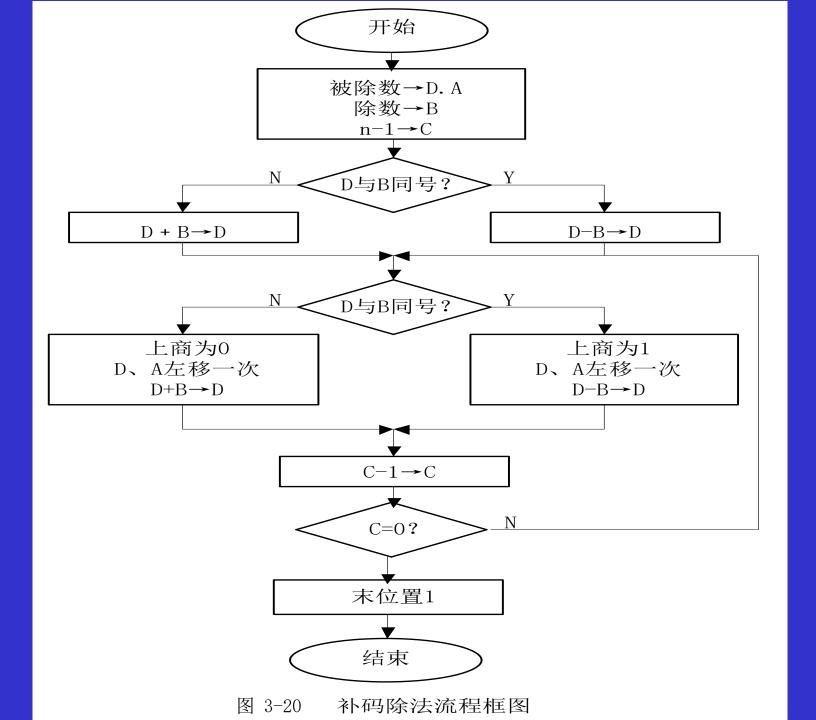
2. 补码除法运算(课后学习)

补码加减交替除法运算规则: (推导过程略)

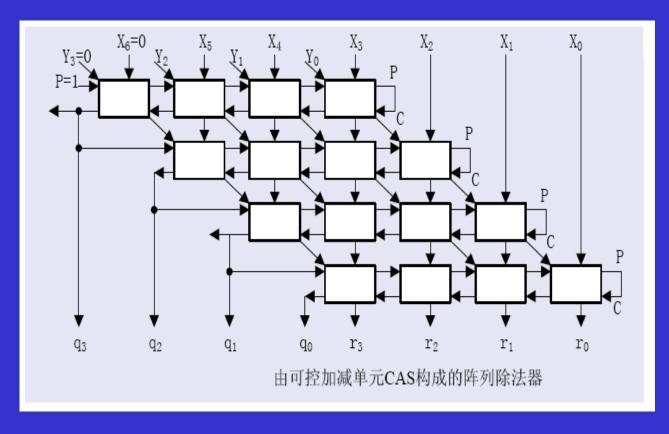
- (1)被除数[X]_补与除数[Y]_补同号,商为正,做减运算, 若余数[R₁]_补与[Y]_补同号,则溢出; 被除数[X]_补与除数[Y]_补异号,商为负,做加运算, 若余数[R₁]_补与[Y]_补异号,则溢出。
- (2) 若余数[R_i]_补与除数[Y]_补同号,上商'1',左移一位, 减除数;

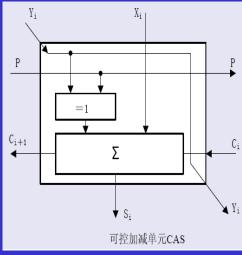
若余数[R_i]_补与除数[Y]_补异号,上商'0',左移一位, 加除数。

- (3) 重复步骤(2),连同符号位在内,共做n-1次(n 为字长),末位采用恒置'1'法。
- (4) 商的符号位与数值位均在运算中产生。



阵列除法器 (课后学习)





本章作业-2

第20 (1) 题

第22 (1) 题 (其中,补码加减交替法不用计算余数)