计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第7次课：3.3.2 定点二位乘法

杜国栋

信息科学与工程学院计算机科学与工程系

gddu@ysu.edu.cn

燕山大學 YANSHAN UNIVERSITY

定点数的乘法 原码一位乘

例　　已知X=-0.1011，Y=0.1001，求$[X\times Y]_{原}$

解：　　$[X]_{原}$=1.1011，$[Y]_{原}$=0.1001

|X| = 0.1011，|Y| = 0.1001

按原码一位乘法运算规则，求[X×Y]原的数值部分。

|X|×|Y| = 0.01100011，　而Zs = Xs⊕Ys = 1⊕0 =1

最后求得$[X\times Y]_{原}$ = 1.01100011

部分积	乘数
0.0000 +）0.1011	1001
0.1011 0.0101 +）0.0000	1100
0.0101 0.0010 +）0.0000	1110
0.0010 0.0001 +）0.1011	0111
0.1100 0.0110 高位积	0011 低位积

定点数的乘法 补码一位乘（校正法）

已知 $[x]_{补}=x_0.x_1\cdots x_n$，$[y]_{补}=y_0.y_1\cdots y_n$，求$[x\cdot y]_{补}$？

1）被乘数 x 符号任意，乘数 y 符号为正

$$[x]_{补}=x_0.x_1x_2\cdots x_n=2+x=2^{n+1}+x\ (\mathrm{mod}\ 2)$$

$$[y]_{补}=0.y_1y_2\cdots y_n=y$$

则 $$[x]_{补}\cdot[y]_{补}=[x]_{补}\cdot y=(2^{n+1}+x)\cdot y=2^{n+1}\cdot y+xy$$

由于 $y=0.y_1y_2\cdots y_n=\sum_{i=1}^{n}y_i2^{-i}$，则 $2^{n+1}\cdot y=2\sum_{i=1}^{n}y_i2^{n-i}$，且 $\sum_{i=1}^{n}y_i2^{n-i}$ 是一个大于或等于 1 的正整数，根据模运算的性质，有 $2^{n+1}\cdot y=2(\mathrm{mod}\ 2)$，故

$$[x]_{补}\cdot[y]_{补}=2^{n+1}\cdot y+xy=2+xy=[x\cdot y]_{补}\quad(\mathrm{mod}\ 2)$$

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

定点数的乘法 补码一位乘（校正法）

已知 $[x]_{补}=x_0.x_1\cdots x_n$，$[y]_{补}=y_0.y_1\cdots y_n$，求$[x\cdot y]_{补}$？

2）被乘数 x 符号任意，乘数 y 符号为负

$$[x]_{补}=x_0.x_1x_2\cdots x_n$$

$$[y]_{补}=1.y_1y_2\cdots y_n=2+y\ (\mathrm{mod}\ 2)$$

则
$$y=[y]_{补}-2=1.y_1y_2\cdots y_n-2=0.y_1y_2\cdots y_n-1$$

$$x\cdot y=x(0.y_1y_2\cdots y_n-1)$$

$$=x(0.y_1y_2\cdots y_n)-x$$

故
$$[x\cdot y]_{补}=[x(0.y_1y_2\cdots y_n)]_{补}+[-x]_{补}$$

将上式 $0.y_1y_2\cdots y_n$ 视为一个正数，正好与上述情况相同。

则 $[x(0.y_1y_2\cdots y_n)]_{补}=[x]_{补}(0.y_1y_2\cdots y_n)$

所以
$$[x\cdot y]_{补}=[x]_{补}(0.y_1y_2\cdots y_n)+[-x]_{补}$$

$$[x\times y]_{补}=[x]_{补}(0.y_1y_2\cdots y_n)-[x]_{补}\times y_0$$

定点数的乘法 补码一位乘（校正法）

例题 1： 已知$[x]_{补}=1.0101$，$[y]_{补}=0.1101$，求$[x \cdot y]_{补}$?

定点数的乘法 补码一位乘（校正法）

例题 1：已知$[x]_{补}$=1.0101，$[y]_{补}$=0.1101，求$[x \cdot y]_{补}$?

因为 y>0，所以利用公式$[x \cdot y]_{补} = [x]_{补} \cdot [y]_{补} = [x]_{补} \cdot (0.y_1 \cdots y_n)$可得结果，此时与原码一位乘类似。

考虑到运算部分积可能出现绝对值大于 1 的情况，注意此时并不是溢出，可通过后续右移解决，故部分积和被乘数取双符号位。

部分积	乘数($y_1y_2y_3y_4$)
00.0000 +11.0101 11.0101 11.1010	1101
11.1101	1110
+11.0101 11.0010 11.1001	0111
+11.0101 10.1110 11.0111	0011 0001
	最终结果：1.01110001

定点数的乘法 补码一位乘（校正法）

例题 2: 已知$[x]_{补}=0.1101$，$[y]_{补}=1.0101$，求$[x \cdot y]_{补}$?

定点数的乘法 补码一位乘（校正法）

例题 2：已知$[x]_{补}$=0.1101，$[y]_{补}$=1.0101，求$[x \cdot y]_{补}$?

因为 y<0，索引利用公式$[x \cdot y]_{补} = [x]_{补} \cdot (0.y_1 \ldots y_n) + [-x]_{补}$可得结果，此时与原码一位乘类似，只不过最后需要进行$[-x]_{补}$修正。

部分积	乘数($y_1y_2y_3y_4$)
00.0000 +00.1101 00.1101 00.0110	0101
00.0011	1010
+00.1101 01.0000 00.1000	0101
00.0100	0010 0001
+11.0011 修正 11.0111	
	最终结果：1.01110001

定点数的乘法 补码一位乘（Booth算法）

已知 $[x]_{补}=x_0.x_1\cdots x_n$，$[y]_{补}=y_0.y_1\cdots y_n$，求$[x\cdot y]_{补}$？

按补码乘法校正法规则，其基本算法可用一个统一的公式表示为

$$[x\cdot y]_{补}=[x]_{补}(0.y_1y_2\cdots y_n)-[x]_{补}\cdot y_0$$

当 $y_0=0$ 时，表示乘数 y 为正，无须校正，即

$$[x\cdot y]_{补}=[x]_{补}(0.y_1y_2\cdots y_n)$$

当 $y_0=1$ 时，表示乘数 y 为负，则

$$[x\cdot y]_{补}=[x]_{补}(0.y_1y_2\cdots y_n)-[x]_{补}$$

$$\begin{aligned}[x\cdot y]_{补}&=[x]_{补}(y_12^{-1}+y_22^{-2}+\cdots+y_n2^{-n})-[x]_{补}\cdot y_0\\&=[x]_{补}(-y_0+y_12^{-1}+y_22^{-2}+\cdots+y_n2^{-n})\\&=[x]_{补}[-y_0+(y_1-y_12^{-1})+(y_22^{-1}-y_22^{-2})+\cdots+(y_n2^{-(n-1)}-y_n2^{-n})]\\&=[x]_{补}[(y_1-y_0)+(y_2-y_1)2^{-1}+\cdots+(y_n-y_{n-1})2^{-(n-1)}+(0-y_n)2^{-n}]\\&=[x]_{补}[(y_1-y_0)+(y_2-y_1)2^{-1}+\cdots+(y_{n+1}-y_n)2^{-n}]\end{aligned}$$

其中，$y_{n+1}=0$。

定点数的乘法 补码一位乘（Booth算法）

已知 $[x]_{补} = x_0.x_1 \ldots x_n$，$[y]_{补} = y_0.y_1 \cdots y_n$，求 $[x \cdot y]_{补}$？

$$[x \cdot y]_{补} = [x]_{补}[(y_1 - y_0) + (y_2 - y_1)2^{-1} + \ldots + (y_n - y_{n-1})2^{-(n-1)} + (y_{n+1} - y_n)2^{-n}]$$

其中，$y_{n+1}=0$;

当 y>0 时，$y_0=0$；当 y<0 时，$y_0=1$;

$y_i y_{i+1}$	$y_{i+1} - y_i$	操作
00	0	新部分积=原部分积右移一位
01	1	新部分积=(原部分积加$[x]_{补}$之后，再右移一位)
10	-1	新部分积=(原部分积加$[-x]_{补}$之后，再右移一位)
11	0	新部分积=原部分积右移一位

定点数的乘法 补码一位乘（Booth算法）

例题 3：已知$[x]_{补}$=1.0101，$[y]_{补}$=1.0011，利用 Booth 算法求$[x \cdot y]_{补}$?

定点数的乘法 补码一位乘（Booth算法）

例题 3：已知$[x]_{补}$=1.0101，$[y]_{补}$=1.0011，利用 Booth 算法求$[x \cdot y]_{补}$?

由 $[x \cdot y]_{补} = [x]_{补}[(y_1 - y_0) + (y_2 - y_1)2^{-1} + \ldots + (y_n - y_{n-1})2^{-(n-1)} + (y_{n+1} - y_n)2^{-n}]$ 可得。 $y_0 = 1$， $y_5 = 0$(附加位)

部分积	乘数 $y_0 \ldots y_n$（补码，单符号）	附加位 y_{n+1}
00.0000 +00.1011 00.1011 00.0101	10011 11001	0 1
00.0010	11100	1
+11.0101 11.0111 11.1011 11.1101	11110	0
+00.1011 00.1000	11111	0
		最终结果：0.10001111

定点数的乘法 补码一位乘（Booth算法）

例：x=13，y=-10　求x·y=？

定点数的乘法 补码一位乘（Booth算法）

例：x=13, y=-10　求x·y=?

$[x]_{补}$=001101，$[y]_{补}$=10110, $[-x]_{补}$=110011

部分积	乘数 $y_n\ y_{n+1}$	说明
0 0 0 0 0 0	1 0 1 1 0 0	y_{n+1}=0
+ 0 0 0 0 0 0		y_ny_{n+1}=00, 加0
0 0 0 0 0 0		
→ 0 0 0 0 0 0	0 1 0 1 1 0	右移一位
+ 1 1 0 0 1 1		y_ny_{n+1}=10, 加$[-x]_{补}$
1 1 0 0 1 1		
→ 1 1 1 0 0 1	1 0 1 0 1 1	右移一位
+ 0 0 0 0 0 0		y_ny_{n+1}=11, 加0
1 1.1 0 0 1		
→ 1 1.1 1 0 0	1 1 0 1 0 1	右移一位
+ 0 0 1 1 0 1		y_ny_{n+1}=01, 加$[x]_{补}$
0 0 1 0 0 1		
→ 0 0 0 1 0 0	1 1 1 0 1 0	右移一位
+ 1 1 0 0 1 1		y_ny_{n+1}=10, 加$[-x]_{补}$
1 1 0 1 1 1	1 1 1 0 1 0	最后一位不移位

$[x\cdot y]_{补}$=101111110　　$x\cdot y = -01000\ 0010=(-82)_{16}=(-130)_{10}$

定点数的乘法 原码二位乘

被乘数X和乘数Y为用原码表示的纯小数

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

$[Y]_{原}=Y_0 . Y_{-1}Y_{-2}\cdots Y_{-(n-1)}$

X_0、Y_0为符号位

两位乘数位有四种组合：

$Y_{i+1}Y_i=00$ 对应+0

$Y_{i+1}Y_i=01$ 对应+|X|

$Y_{i+1}Y_i=10$ 对应+2|X|

$Y_{i+1}Y_i=11$ 对应+3|X|

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

定点数的乘法 原码二位乘

【例】

设X＝＋0.100111，Y＝－0.100111，利用原码求积。

定点数的乘法 原码二位乘

【例】

设X＝＋0.100111，Y＝－0.100111，利用原码求积。

【解】

$[X]_{原}=0.100111$

$[Y]_{原}=1.100111$

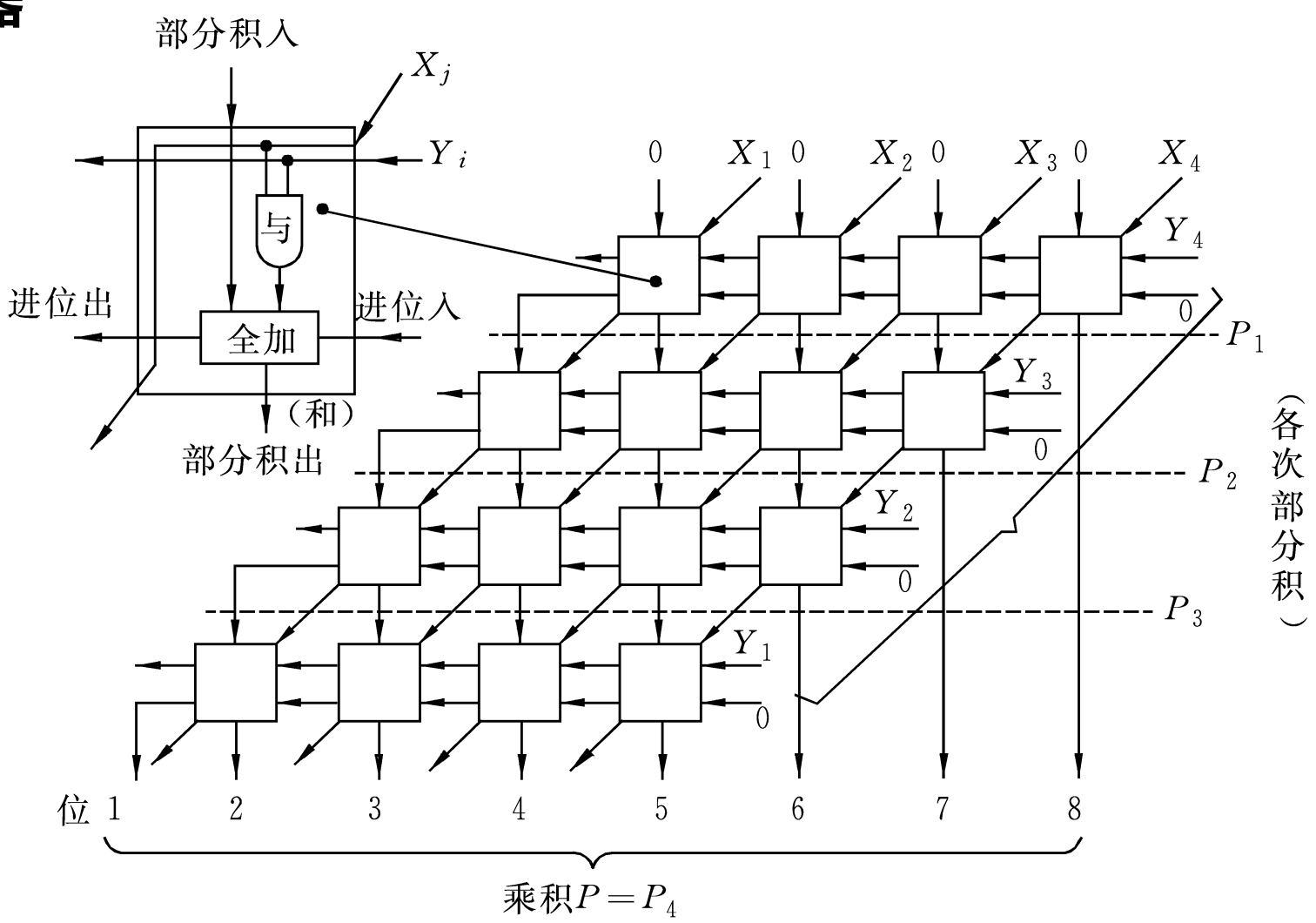
$[-|X|]_{补}=1.011001$

原码二位乘法的运算过程

符号位	D	A	操作
0 0 0	0 0 0 0 0 0	1 0 0 1 1 1	C=0
1 1 1	0 1 1 0 0 1		－X
1 1 1	0 1 1 0 0 1		C=1
1 1 1	1 1 0 1 1 0	0 1 1 0 0 1	→右移二位
0 0 1	0 0 1 1 1 0		+2X
0 0 1	0 0 0 1 0 0		C=0
0 0 0	0 1 0 0 0 1	0 0 0 1 1 0	→右移二位
0 0 1	0 0 1 1 1 0		+2X
0 0 1	0 1 1 1 1 1		C=0
0 0 0	0 1 0 1 1 1	1 1 0 0 0 1	→右移二位

乘积的符号为：$0 \oplus 1=1$，$[X \cdot Y]_{原}=1.010111110001$

阵列乘法器


部分积入
X_j
Y_i
与
进位出
全加
进位入
(和)
部分积出
0 X_1 0 X_2 0 X_3 0 X_4
Y_4
0
P_1
Y_3
0
P_2
Y_2
0
P_3
Y_1
0
(各次部分积)
位 1 2 3 4 5 6 7 8
乘积$P=P_4$

有问题欢迎随时跟我讨论

办公地点：西校区信息馆423

邮　　箱：gddu@ysu.edu.cn