计算机组成原理

PRINCIPLES OF COMPUTER ORGANIZATION

第6次课：3 3.1 定点一位乘法

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

厚 德 · 博 学 · 求 是

课程目标

- 掌握定点原码一位乘;
- 熟悉补码一位乘;
- 了解乘法器的结构。

定点数的乘法　原码一位乘

设 A=0.1101，B=0.1011，求 $A\times B$。

笔算时，乘积的符号由心算而得，正正为正。其数值部分的运算如下：

$$
\begin{array}{rll}
 & 0.1101 & \\
\times & 0.1011 & \\
\hline
 & 1101 & A*2^0 \quad A\text{ 不移位} \\
 & 1101 & A*2^1 \quad A\text{ 左移 1 位} \\
 & 0000 & 0*2^2 \quad 0\text{ 左移 2 位} \\
 & 1101 & A*2^3 \quad A\text{ 左移 3 位} \\
\hline
 & 0.10001111 &
\end{array}
$$

显然，这里包含着被乘数的多次左移，以及 4 个位积的相加运算。

两大问题：

① 将 4 个位积一次相加，机器很难实现；

② 乘积位数增长一倍(需要 2N 位的加法器)，将造成存储空间和运算时间浪费。

定点数的乘法 原码一位乘

设 A=0.1101，B=0.1011，求 $A\times B$。

$$
\begin{aligned}
A\times B &= A\times 0.1011 \\
&= A\times 0.1+ A\times 0.00+ A\times 0.001+ A\times 0.0001 \\
&= A\times 0.1+ A\times 0.00+0.001(A+0.1 A) \\
&= A\times 0.1+ 0.01(0A+0.1(A+0.1 A)) \\
&=0.1(A+0.1(0A+0.1(A+0.1 A))) \\
&=2^{-1}(A+2^{-1} (0A+2^{-1} (A+2^{-1}A))) \\
&=2^{-1}(A+2^{-1} (0A+2^{-1} (A+2^{-1}(A+0))))
\end{aligned}
$$

$$
\begin{cases}
z_0 = 0 \\
z_1 = 2^{-1}(x^* y_4 + z_0) \\
z_2 = 2^{-1}(x^* y_3 + z_1) \\
z_3 = 2^{-1}(x^* y_2 + z_2) \\
z_4 = 2^{-1}(x^* y_1 + z_3)
\end{cases}
\qquad \text{其中，} \quad y_0.y_1y_2y_3y_4 = 0.1011
$$

由此可得，两个数相乘的过程，可视为加法和移位（相当于右移）两种运算，这对于计算机而言，非常容易实现。

定点数的乘法　原码一位乘

部分积	乘数($y_1y_2y_3y_4$)
0.0000 +0.1101 0.1101 0.0110	1011
+0.1101 1.0011 0.1001	1101
+0.0000 0.1001 0.0100	1110
+0.1101 1.0001 0.1000	1111 1111
	最终结果：0.10001111

上述运算步骤可归纳如下：

① 乘法运算可用移位和加法来实现，两个 4 位数相乘，总共需要进行 4 次加法和 4 次移位。

② 由乘数的末位决定被乘数是否与原部分积相加，然后右移一位，形成新的部分积。与此同时，乘数右移一位，空出的最高位存放新的部分积的最低位。

③ 每次做加法时，被乘数仅仅与部分积的高位进行相加，其低位被移至乘数所空出高位位置。

定点数的乘法　原码一位乘

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

定点数的乘法　　*原码一位乘*

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

$[x]_{原}=1.1110$

$[y]_{原}=1.1101$

$x_0=1$，$y_0=1$

$x^*=0.1110$， $y^*=0.1101$

按照原码一位乘的公式 $[x \cdot y]_{原}=(x_0 \oplus y_0).(0.x_1 \ldots x_n)(0.y_1 \ldots y_n)$ 可得数值部分为：

部分积	乘数($y_1y_2y_3y_4$)
0.0000 +0.1110 0.1110 0.0111	1101
+0.0000 0.0111 0.0011	0110
+0.1110 1.0001 0.1000	1011
+0.1110 1.0110 0.1011	1101 0110
最终结果：	$[x^* \cdot y^*]_{原}=0.10110110$ $[x \cdot y]_{原}=0.10110110$ $x \cdot y=+0.10110110$

定点数的乘法 原码一位乘

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

定点数的乘法　　*原码一位乘*

例3-6　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. 0 0 0 0 +) 0. 1 0 1 1	1 0 0 1
0. 1 0 1 1 0. 0 1 0 1 +) 0. 0 0 0 0	1 1 0 0
0. 0 1 0 1 0. 0 0 1 0 +) 0. 0 0 0 0	1 1 1 0
0. 0 0 1 0 0. 0 0 0 1 +) 0. 1 0 1 1	0 1 1 1
0. 1 1 0 0 0. 0 1 1 0 高位积	0 0 1 1 低位积

定点数的乘法

补码一位乘（校正法）

$$\begin{aligned}
&[z_0]_{补}=0\\
&[z_1]_{补}=2^{-1}(y_n[x]_{补}+[z_0]_{补})\\
&[z_2]_{补}=2^{-1}(y_{n-1}[x]_{补}+[z_1]_{补})\\
&\quad\vdots\\
&[z_i]_{补}=2^{-1}(y_{n-i+1}[x]_{补}+[z_{i-1}]_{补})\\
&\quad\vdots\\
&[x\cdot y]_{补}=[z_n]_{补}=2^{-1}(y_1[x]_{补}+[z_{n-1}]_{补})
\end{aligned}$$

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

1）被乘数 x 符号任意，乘数 y 符号为正

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

$$[y]_{补}=0.y_1y_2\cdots y_n=y$$

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 $2^{n+1}\cdot y=2(\bmod 2)$，故

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

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

对照原码乘法式(6.9)和式(6.10)可见，当乘数 y 为正数时，不管被乘数 x 符号如何，都可按原码乘法的规则运算，即

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$[x\cdot y]_{补}=[x]_{补}\cdot[y]_{补}=[x]_{补}\cdot(0.y_1...y_n)$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

总结

- 原码一位乘;
- 补码一位乘（校正法）；
- 补码一位乘（Booth算法）。

课后习题：P68　3.17　3.18

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