国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第11卷第1期 1989年3月 Vol.11. No.1

软硬结合的迭代除法方案及其精度分析

张民选 李晓梅

（计算机系）

摘要 大型机采用的软硬结合的迭代除法方案具有很高的计算速度，但存在精度合理性问题。经过大量随机数试算验证和算法的误差分析证明：本文中提供的优选除法方案，提高了除法精度，解决了精度合理性问题。该除法方案装机运行后，提高了大型机的适应能力和运算速度，改善了处理效果。

关键词 迭代，除法器，优选法，精度，合理性，软硬件结合

分类号 TP313

1 大型机的迭代除法方案

除法是最常用的四则算术运算之一，除法运算速度对整个计算机系统的处理速度有着不可忽视的影响。在大型计算机上，一般采用多位除法或迭代除法方案，以便获得较高的除法运算速度。对于一种大型机，根据其线性流水线式多功能部件及向量运算可链接的特点，设计了独特的软硬结合的迭代除法方案，使得运算器很节省，除法速度很高，是一种成功的设计方案。

1.1 实现除法运算的有关指令

在一种大型机指令系统中，为除法运算提供了半精度倒数近似值和迭代乘$(2-xy)$两条专用指令以及浮点乘和舍入乘两条通用指令。上述四条指令分别由浮点倒数近似值部件和浮点乘法部件执行。浮点倒数近似值部件与浮点乘法部件可以链接成复合流水线功能部件。现将与除法有关的四条指令简介如下：

(1) 半精度倒数近似值指令：该指令由浮点倒数近似值部件执行。操作数与计算结果均为规格化浮点数。若y是x的倒数近似值，则有：

$$(1.0-(2^{-25}+2^{-26}-2^{-30}))\leqslant x\cdot y\leqslant(1.0+(2^{-25}+^{-27}))$$

倒数近似值指令的计算结果具有24位精度。

(2) 迭代乘$(2-xy)$指令：该指令由浮点乘法部件执行。指令要求两个源操作数的阶码之和必须是1，尾数是规格化数，才能保证结果的正确性。若$(2-x\cdot y)$的精确值为1.0，计算结果被处理成$(1.0-2^{-48})$。由于乘法器采用三位一乘为基的49位补码乘法算法

1987年11月3日收稿

和硬件截尾的设计方案，迭代乘指令的两个源操作数交换位置，计算结果可能出现差异。

(3) 浮点乘，舍入乘指令：浮点乘和舍入乘指令由浮点乘法部件执行。浮点乘指令，两个尾数字长为48位的源操作数相乘，乘积尾数为96位，截取其高48位做为计算结果，则计算结果的误差e_{MF}为：$0 \leqslant e_{MF} < 2^{-48}$.

对于舍入乘指令，两操作数相乘时，在乘法器金字塔的2^{-49}位无条件加"1"。当截取规格化乘积的高48位作为计算结果时，舍入的效果分为两种情况，其计算结果的误差e_{MR}分别为：

$-2^{-49} \leqslant e_{MR} < 2^{-49}$，当尾数乘积$\in [0.5, 1)$时（占61.37%）

$-2^{-48} \leqslant e_{MR} < 0$，　当尾数乘积$\in [0.25, 0.5)$时（占38.63%）

1.2 两类除法指令序列及其速度比较

根据上述四种指令和截尾乘法器的特点，有两类各22种除法指令序列可供选用。

(1) 先求出半精度商，然后再修正得到单精度商与函数$y = x_1 / x_2$. 设R_H为x_2的半精度倒数近似值，则求y的实际计算步骤是：

$R_H = 1/x_2$，　　半精度倒数近似值；

$y_H = R_H * x_1$，　　半精度商；

$\Delta C = 2 - R_H * x_2$，迭代修正因子；

$y = \Delta C * y_H$，　　单精度商。

以标量除法(S6=S1/S2)为例，相应的除法指令序列为：

```
S3  /HS2
S4  S3 * FS1
S5  S3 * IS2
S6  S5 * RS4
```

在其指令序列中，使用了倒数近似值、倒数迭代乘、浮点乘和舍入乘等四类指令。各种指令不同形式的组合，使该指令序列共有32种相异形式。

(2) 先求单精度倒数，再求单精度商与函数$y = x_1 / x_2$. 设R_H为x_2的半精度倒数近似值，求y的实际计算步骤为：

$P_H = 1/x_2$，　　半精度倒数近似值；

$\Delta C = 2 - x_2 R_H$，迭代修正因子；

$R_F = \Delta C * R_H$，单精度倒数近似值，

$y = R_F * x_1$，　　单精度商。

相应的指令序列（以S6=S1/S2为例）如下：

```
S3  /HS2
S4  S2 * IS3
S5  S4 * FS3
S6  S5 * RS1
```

各种指令的不同形式的组合，使该类除法方案也有32种指令序列。

(3) 两类除法指令序列的速度比较：对于标量除法（S/S型），两类除法指令序列有相同的计算速度；对于V/V型，S/V型向量除法，(1)类除法指令序列中的前两条指令

可以链接执行，因而计算速度比(2)类除法指令序列快30%；对于V/S型向量除法，(1)类除法指令序列的四条指令中有两条向量指令和两条标量指令，而(2)中除法指令序列由一条向量指令，三条标量指令组成，因而(2)比(1)的计算速度快；除数为常数的情况下，编译系统为了加快目标程序的计算速度，通常在编译时就求出该常数的倒数，然后在目标代码中用一条乘法指令实现该除法，实现时采用了(2)类除法指令序列。

2 除法方案的精度分析

2.1 除法运算精度的判定标准

从除法的精度合理性考虑，对于$y=x_1/x_2$，当$x_1\geqslant x_2$时，应有$y\geqslant 1.0$；当$x_1\leqslant x_2$时，应有$y\leqslant 1.0$，特别是当$x_1=x_2$时，应有$y=1.0$。因此，可用$(1-x/x)$的计算结果来度量除法运算的精度。$(1-x/x)$的计算结果，实际上反映的是除数倒数的相对误差。若计算有$(1-x/x)=0$，则说明单精度倒数的相对误差小于2^{-48}（机器浮点数的尾数字长48位），即单精度除法运算具有47位精度，且能够满足除法的精度合理性要求。

用$(1-x/x)$的计算结果来检验除法运算的精度时，应采用实际使用的除法指令序列进行分析，这样可以排除因运算误差不同而产生的影响。

2.2 第一种除法指令序列的精度分析

设$\varepsilon(x)$为$(1-x/x)$的计算误差，根据计算步骤和考虑到机器运算误差的引入，则有$\varepsilon(x)$：

$$\begin{aligned}\varepsilon(x)&=|1-[(xR_H+\Delta_1)(2-xR_H+\Delta_2)+\Delta_3]|\\&=|(1-(xR_H)(2-xR_H))-((2-xR_H)\Delta_1+xR_H\Delta_2+\Delta_1\Delta_2)-\Delta_3|\\&=|\varepsilon_1+\varepsilon_2+\varepsilon_3|\end{aligned}$$

式中：$\varepsilon_1=1-(xR_H)(2-xR_H)$，为迭代除法的理论误差；

$\varepsilon_2=-((2-xR_H)\Delta_2+xR_H\Delta_2+\Delta_1\Delta_2)$，为求半精度商和迭代因子两条指令引进的计算误差；

$\varepsilon_3=-\Delta_3$，为求单精度商最后一条乘法指令引进的计算误差。

下面分三种情况进行讨论：

(1) 当$xR_H=1.0$时，有：

$\varepsilon_1=\varepsilon_2=2^{-48}$，$\varepsilon_3=2^{-48}$

则 $\varepsilon(x)=0$。

(2) 当$xR_H<1.0$时，有：

$(1.0-(2^{-25}+2^{-26}-2^{-30}))\leqslant xR_H<1.0$；$0<\varepsilon_1<(2^{-49}+2^{-52}-2^{-54})$

$-2^{-49}<\Delta_1\leqslant 0$；$\Delta_2=-\Delta_1$

$0\leqslant\varepsilon_2<(2^{-72}+2^{-73})$；$-(2^{-49}+2^{-52})\leqslant\varepsilon_3<(2^{-49}-2^{-52})$。

亦有：$\varepsilon(x)<2^{-48}$。

(3) 当$xR_H>1.0$时，有：

$1.0<xR_H\leqslant(1.0+(2^{-25}+2^{-27}))$；$0<\varepsilon_1\leqslant(2^{-50}+2^{-51}+2^{-54})$

$-(2^{-72}+2^{-73})<\varepsilon_2\leqslant 0$；$-(2^{-49}+2^{-52})\leqslant\varepsilon_3<2^{-49}-2^{-52}$

此种情况仍有：$\varepsilon(x)<2^{-48}$。

综合(1)～(3)种情况，得到$(1-x/x)$的计算误差$\varepsilon(x)$小于2^{-48}.用该除法指令序列求半精度商时采用浮点乘，求单精度商时采用舍入乘，使x/x的计算结果严格等于1.0，满足除法的精度合理性要求。

2.3 第二种除法指令序列的精度分析

设迭代乘截断误差为$\Delta 1$，半精度倒数修正或单精度倒数之乘法的截断误差为Δ_2，倒数与被除数相乘的截断误差为Δ_3，用$\varepsilon(x)$表示$(1-x/x)$的计算误差，则有：

$$\begin{aligned}\varepsilon(x)&=|1-[x((2-xR_H+\Delta_1)R_H+\Delta_2)+\Delta_3]|\\&=|(1-x(2-xR_H)R_H)+(-xR_H\Delta_1)+(-x\Delta_2-\Delta_3)|\\&=|\varepsilon_1+\varepsilon_2+\varepsilon_3|\end{aligned}\tag{2.2}$$

式中：$0\leqslant\varepsilon_1<(2^{-49}+2^{-52}-2^{-54})$；

$$(1.0-(2^{-25}+2^{-26}-2^{-30}))\leqslant xR_H\leqslant(1.0+(2^{-25}+2^{-27}))$$；

$0\leqslant\Delta_1<2^{-48}$；　$-(2^{-48}+2^{-73}+2^{-75})<\varepsilon_2\leqslant 0$；　$\varepsilon_3=-x\Delta_2-\Delta_3$.

由于舍入乘与浮点乘的误差不同，Δ_2,Δ_3有四种组合：

$$\begin{cases}-2^{-48}<\Delta_2\leqslant 0\\-2^{-48}<\Delta_3\leqslant 0\end{cases},\quad\begin{cases}-2^{-49}<\Delta_2\leqslant 2^{-49}+2^{-52}\\-2^{-49}<\Delta_3\leqslant 2^{-49}+2^{-52}\end{cases}$$

$$\begin{cases}-2^{-48}<\Delta_2\leqslant 0,\\-2^{-49}<\Delta_3\leqslant 2^{-49}+2^{-52},\end{cases}\quad\begin{cases}-2^{-49}\leqslant\Delta_2<2^{-49}+2^{-52}\\-2^{-48}<\Delta_3\leqslant 0\end{cases}$$

又有　　$0.5\leqslant x_{尾}<1.0$

及　　$\varepsilon_1\approx-\varepsilon_2$

因此，可控制$\Delta_2.\Delta_3$，使ε_3的绝对值尽量小且有利于互相抵消。此时有：

$$-(2^{-48}+2^{-49})<\varepsilon_3<2^{-49}+2^{-52}$$

$$\varepsilon(x)<2^{-48}$$

当$xR_H=1.0$时，是一种特殊情况，有：

$$\varepsilon_1=0;\quad 0\leqslant\varepsilon_2\leqslant 2^{-48}$$

$$-(2^{-48}+2^{-49})<\varepsilon_3<2^{-49}+2^{-52};\quad\varepsilon(x)<2^{-47}$$

综上分析可知，除法指令序列中两次乘法运算应将舍入乘指令和浮点乘指令搭配使用。该方案计算$(1-x/x)$的最大误差$\varepsilon(x)$小于2^{-48}，对于一般的除法x_1/x_2，其运算精度也有2^{-48}，因为x/x和x_1/x_2在处理上并没有区别。

3 除法指令序列的选优

除法运算的第一种指令序列比第二种指令序列的计算结果高一位精度。因此，在一般情况下，使用第一种除法指令序列；在必须采用第二种指令序列或采用第二种指令序列能获得更快计算速度的地方，才使用第二种指令序列。

在使用第一种除法指令序列时，如下两种形式的指令序列是最好的：

a)
R_H	/Hx$_2$	
Y_H	R_H * Fx$_1$	
ΔC	R_H * Ix$_2$	

b)
R_H	/Hx$_2$	
Y_H	x$_1$ * FR_H	
ΔC	x$_2$ * IR_H	

y $\Delta C * RY_H$ y $\Delta C * RY_H$

其中第2、3两条指令操作数位置的安排及第2条指令采用浮点乘，均为了使 $x_1 = x_2$ 时，有 $\Delta_1 = -\Delta_2$；求单精度商乘法指令的操作数位置的安排及采用舍入乘指令，均为了提高除法运算的精度，使 x/x 的计算结果恒为1.0.

两种较好的第二种除法指令序列是：

a) R_H /Hx_2 b) R_H /Hx_2

ΔC $x_2 * IR_H$ ΔC $x_2 * IR_H$

R_F $\Delta C * FR_H$ R_F $\Delta C * RR_H$

y $R_F * Rx_1$ y $R_F * Fx_1$

增加修正措施的第二种指令序列如下：

S3	/HS2	S7	S5	
S4	S2 * IS3	S7		S4!S7&S3
S5	S4 * F[R]S3	JSZ	L2	
S6	S5 * R[F]S2	JSP	L1	
S7	1.0	S5	S5-FS7	
S0	S7-FS6	J	L2	
S3	<48	L1	S5	S5+FS7
S4	1	L2	S6	S5 * R[F]S1

实际计算中，有75%的数据不需修正，因此，计算速度的损失不太大。对于V/S型除法，采用改进的第二种指令序列，比采用第一种指令序列的计算速度仍然要高。

用大量随机数和特殊数进行 x/x，nx/x，x^2/x 的计算，对64种除法指令序列选优。第一种除法指令序列两个优选方案均能满足除法的精度合理性要求，第二种除法指令序列增加修正措施后，也能达到目的。

x/x除法试算所用数据的尾数有如下七种形式：

(1) .1×××……× $\underbrace{1\ 1\ \cdots\cdots 1}_{20位}$

27位

(2) .1×××……× $\underbrace{0\ 0\ \cdots\cdots 0}_{20位}$

27位

(3) .1 0 0……0 ××……×

13位 34位

(4) .1 1 1……1 ××……×

13位 34位

(5) .1××××……× $\underbrace{1\ 1\ \cdots\cdots 1}_{14位}$ $\underbrace{0\ 0\ \cdots\cdots 0}_{7位}$

26位

(6) .1 * * …… * （加同余法产生的随机数）

47位

(7) .1 * * …… * （乘同余法产生的随机数）

47位

其中：×、＊取值0或1，×为穷举计数值，＊为随机数值。试验数据量共1100多亿个。

nx/x除法试验所用的数据形式为：

(1) $.1\underbrace{\times\times\cdots\cdots\times}_{23位}\quad\underbrace{0\ 0\cdots\cdots 0}_{24位}$

(2) $.1\quad\underbrace{*\ *\cdots\cdots *}_{35位}\quad\underbrace{0\ 0\cdots\cdots 0}_{12位}$

(3) $n=2,3\cdots\cdots,1025$

x^2/x除法试验所用的数据形式为：

$.1\underbrace{\times\times\cdots\cdots\times}_{23位}\quad\underbrace{0\ 0\cdots\cdots 0}_{24位}$

nx/x，x^2/x除法试验数据的选择主要考虑到两点：(1) 数据应具有代表性；(2) 应保证nx，x^2的计算结果无误差。

试验结果证明，优选除法方案是最好方案，它保证除法的精度合理性。

4 优选方案的机器证明

在优选方案中，最后的将半精度商修正成单精度商的乘法指令两源操作数是不能交换位置。用理论分析来证明其交换两操作数位置，使x/x产生不等于1.0的结果是相当困难的。但考虑到这是一次特殊的乘法，即两个操作数均是接近1.0的数，当$\Delta C=1.0+2^{-24}\Delta$时，$Y_H=1.0-2^{-24}\Delta$. 反之亦然。因此可采用穷举法用机器来证明。

根据机器的实际结构，用如下三种数据来做$\Delta C * RY_H$和$Y_H * R\Delta C$的计算：

(1) 当$Y_H=1.0$时，$\Delta C=1.0-2^{-48}$；

(2) 当$1.0<Y_H\leqslant 1.0+(2^{-25}+2^{-47})$时，$\Delta C=1.0+(1.0-Y_H)$；

(3) 当$1.0-(2^{-25}+2^{-26})\leqslant Y_H<1.0$时，$\Delta C=1.0+(1.0-Y_H)$.

计算结果表明：$\Delta C * RY_H$的结果全部等于1.0，$Y_H * R\Delta C$的结果中有小于1.0的情况出现。可知，最后一条乘法指令两源操作数不能交换位置是机器特殊结构决定的。

5 结 束 语

优选除法方案装机使用后，解决了x/x不等于1的精度合理性问题。试算测试表明：某些题目收敛速度加快，处理效果得到改善；原来由于精度问题引起计算结果有差异，现在也能得到正确的满足精度要求的计算结果。

本文中采用的软硬结合的迭代除法方案可以广泛应用于大型计算机的设计。

本工作得到周兴铭教授、陈立杰教授、杨晓东教授以及西南计算中心王嘉谟、侯富贵二位高级工程师支持和帮助，在此深表谢意。

参 考 文 献

[1] 周兴铭、张民选. 计算机学报. 1981; 4(5)

[2] Krishramuthy E V. IEEE Trans.E.C.,1970; C-19(3)

[3] Ferrari D.IEEE Trans.E.C,1967; C-16(2)
[4] 张民选. 电子学报, 1984; 12(3)
[5] Kai Hwang.Computer Arithmetic principles.Archtecture And Design,John Wiley & Sons,1979

Iterative Division Schemes With Hardware-Softwave and its Accuracy

Zhang Minxuan Li Xiaomei

Abstract

Iterative division schemes using hardware—software are studied in this paper. The accuracy control method is suggested for the $x/x=1$. The division schemes are used for the supercomputer.

Key Words: interation, diviser, optimum seeking method, accuracy; reasonableness, hardware-software