国防科技大学学报
JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY
第18卷第4期 1996年12月 Vol. 18 No. 4

一种用浮点除实现整除的方法*

龚雪春 孙衍吉 陈旭华

(国防科技大学计算机系 长沙 410073)

摘 要 本文提出一种用浮点除法来实现整数除法的方法。对这种方法可能带来的误差进行估计并给出了商的修正方法。最后研究一个实例并对该方法进行了评价。

关键词 整除，浮点除，IEEE 浮点格式，舍入误差，微程序

分类号 TP311.11

A Method of Integer Division by Using Floating-point Division

Gong Xuechun Sheng Yeji Cheng Xuhua

(Department of Computer Science, NUDT, Changsha, 410073)

Abstract A method of integer division using floating-point division is introduced in this paper. The possible error produced by this method is estimated and the revising method is given. an implementing example is described and the method is finally evaluated.

Key words integer division, floating-point division, IEEE floating-point format, rounding error, microprogram

VLSI 技术不断发展，如何充分挖掘所选 VLSI 芯片的潜力以减少硬件复杂性、增加可靠性和降低成本，是计算机设计中的新课题。

作者在设计一台计算机的 CPU 时采用了 BIT 公司的 B3110A/B3120A ALU 芯片组。该芯片组具有二百多种指令。能进行 64 位逻辑运算，32 位整数加、减、乘运算，64 位整数加、减运算，单精度(32 位)/双精度(64 位)IEEE/DEC 浮点加、减、乘、除、开方运算，以及数据格式转换、比较、移位等许多其它类型的运算。B3110A 专门进行乘、除、开方的运算，B3120A 则完成加、减、逻辑等所有其余的运算。

尽管 B3110A/B3120A 的指令系统已十分庞大，但它仍无 32 位整除，64 位整乘、除等指令。而这些指令又恰是目标机指令系统所要求的。为了不增加硬件，作者采用了一个

* 1996 年 3 月 21 日收稿

具有 1k×72 位控存的微程序控制器。目标机指令系统中的绝大部分指令都靠 B3110A/B3120A 的相应指令来完成，余下的指令则要靠适当的算法和微程序来实现。

如 64 位整乘可以靠分解成 4 个 32 位数相乘、移位、相加并判溢出来完成。32 位整除则首先变成双精度浮点除，然后将浮点商转换为 32 位整数。这些指令实现起来算法相对简单，故不多叙，本文主要讨论 64 位整除的算法及其实现。

1 用浮点除实现整除的方法

1.1 基本思想

设有计算机系统，其整数有 8 位(字节)、16 位(半字)、32 位(字)、64 位(长字)四种数据格式，浮点有单精度 IEEE 和双精度 IEEE 两种数据格式(对于 DEC 浮点格式本算法也完全适应)。又设有 64 位整数 X 和 Y，今要求 64 位整数商 $Q=X/Y-R/Y$，其中 R 为余数。算法的基本设想如下：

第一步，将 X 转换为双精度 IEEE 浮点数 X_f，Y 转换为双精度 IEEE 浮点数 Y_f；

第二步，求浮点商 $Q_f=X_f/Y_f-R_f/Y_f$，R_f 为浮点除时的余数；

第三步，将 Q_f 转换为 64 位整数 Q'。

1.2 误差分析

若 $Q'=Q$ 则问题便解决了，但由于计算机表示中，64 位整数数据和 64 位浮点数据在精度及数据范围两方面存在差异，因此 Q' 和 Q 不能简单地等同。

64 位整数和 IEEE 浮点数在计算机中的表示分别如图 1(a)和(b)所示。

(a)64 位整数　　(b)64 位 IEEE 浮点数

图 1

图中 s 为符号位，占 1 位；e 为阶码，占 11 位；f 为尾数，占 52 位。整数所能表示的范围为$[-2^{63}, 2^{63}-1]$，但其有效数据位为 63 位，浮点数的数据范围比整数大，但其有效数据位数只有 52 位了。考虑到大部分浮点运算硬件都有一位附加位参与运算(B3110A/B3120A 就是如此)，有效数据位数也只有 53 位，比整数据少了 10 个有效数据位。这样，在 X 转换为 X_f，Y 转换为 Y_f 时存在丢失有效数据位的可能性，并且最多丢掉 10 位。所以 1.1 中的基本方法要产生误差 E_q，即 $Q=Q'+E_q$。

为了估计 E_q 大小，设运算过程中所有舍入方式为截断舍入，即向 0 方向舍入(这种假定不会带来问题。一方面，计算机整除按截尾方式进行，另一方面，像 B3110A/B3120A 等 ALU 芯片有改变舍入方式指令，4 种舍入方式即向最接近数舍入、向$+\infty$方向舍入、向$-\infty$方向舍入和向 0 方向舍入可以通过指令随意设置并更改)，X，Y 均按绝对值情况讨论且假定除数 $Y\neq 0$。则有 $X=X_f+E_x$，$Y=Y_f+E_y$ 其中 E_x，E_y 分别为整数 X 转换为浮点 X_f 和整数 Y 转换为浮点数 Y_f 时的截断误差。下面分四种情况予以讨论。

1.2.1　$X<2^{53}$，$Y<2^{53}$

由于 X 和 Y 的有效位数均不超过 53 位，故 $E_x=E_y=0$。由于此时必有 $Q<2^{53}$，所以 Q 的有效位数也不超过 53 位，在浮点精度的可达范围内。因此，在浮点除以及浮点商转换为整数时仅截去商的小数部分，即 $E_q<1$。从而 $Q'=Q$，可以确保商 Q' 的整数精度。

1.2.2 $X<2^{53}$，$Y\geqslant 2^{53}$

由于 X 的有效位数不超过 53 位，Y 的有效位数则超过 53 位，故 $E_x=0$，$E_y\geqslant 0$。于是，$X=X_f$，$Y=Y_f+E_y\geqslant Y_f$，所以 $X=X_f<Y_f\leqslant Y$，$Q=X/Y=0$，$Q_f=X_f/Y_f<1$，Q_f 转换为 Q' 时必有 $Q'=0$，所以 $Q'=Q$。没有整数误差。

1.2.3 $X\geqslant 2^{53}$，$Y\geqslant 2^{53}$

这时 $X=X_f+E_x$，$Y=Y_f+E_y$ 且 $E_x\geqslant 0$，$E_y\geqslant 0$。但由于整数最大不超过 2^{63}，故必有 $Q\leqslant 2^{10}$。Q_f 转换为 Q' 时不会产生整数误差。整数误差只能由 X 和 Y 的截断误差引起。

由于 $X_f=Q_f\cdot Y_f+R_f$，$X=Q\cdot Y+R$ 将该式变形有 $X/Y=Q+R/Y$，R/Y 为 X/Y 所剩的小数部分。

又 $$\begin{aligned}X/Y&=(X_f+E_x)/Y\\&=(Q_f\cdot Y_f+R_f+E_x)/Y\\&=(Q_f(Y-E_y)+R_f+E_x)/Y\end{aligned}$$

于是，$Q-Q_f=(R_f+E_x-Q_f\cdot E_y-R)/Y$

由于 $R_f<Y_f\leqslant Y$，所以 $R_f/Y<1$。$R/Y<1$。E_x/Y 为被除数 X 的截断误差 E_x 折合到商数上的等效值(记作 E_{xq})，它使商的绝对值减小。由于 $E_x<2^{10}$(至多截去最低 10 位)，$Y\geqslant 2^{53}$，所以有

$$E_{xq}<2^{10}/2^{53}=2^{-43}<<1$$

$(E_y\cdot Q_f)/Y$ 为除数 Y 的截断误差折合到商数上的等效值(记作 E_{yq})，它使商的绝对值增大。由于 $E_y<2^{10}$，$Q_f<2^{10}$，故有 $E_y\cdot Q_f<2^{20}$，又 $Y\geqslant 2^{53}$，因而有

$$E_{yq}<2^{20}/2^{53}=2^{-33}<<1$$

综上可得

$$-2<(R_f+E_x-Q_f\cdot E_y-R)/Y<2$$

即 $-2<Q-Q'<2$。最大整数误差为 ± 1。亦即可能的 Q 值有三种情况 $Q=Q'-1$，$Q=Q'$，$Q=Q'+1$。

1.2.4 $X\geqslant 2^{53}$，$Y<2^{53}$

由于 $E_y=0$，故 $Y=Y_f$。由 1.2.3 推断可得

$$Q=Q_f+(R_f+E_x-R)/Y$$

下面分两种情况予以讨论

(1)$2^{10}\leqslant Y<2^{53}$，这时 $1<Q<2^{53}$，故商 Q 在浮点精度的可达范围内。由于 $R_f<Y_f=Y$，$E_x<2^{10}$，$Y\geqslant 2^{10}$，$R<Y$，所以，$-1<(R_f+E_x-R)/Y<2$。即 $-1<Q-Q'<2$

所以 $Q=Q'$ 或者 $Q=Q'+1$。

(2)$Y<2^{10}$，这时 $Q\geqslant 2^{53}$，商数的整数部分有效位数超过 53 位，超出了浮点精度可以表示的范围。因此可能有 $R_f>Y_f=Y$，$E_q>1$。用浮点除所求得的商以及在转换过程中要截去商数的低位整数部分，误差较大。

2 商 Q′的修正方法

根据前面的分析可以看出，仅当 $X\geqslant 2^{53}$时用浮点除求得的商数才需加以修正。修正

的方法采用恢复余数法，但对不同的情况可以采用不同的措施。

2.1 $X \geqslant 2^{53}$，$Y \geqslant 2^{10}$

根据 1.2.4 的误差分析可知，这时 Q' 的最大误差为 ± 1。但修正过程却颇为复杂。基本步骤如下

(1)作 $P = Q' \cdot Y$

$$= (Q'_{MSH} \cdot 2^{32} + Q'_{LSH}) \cdot (Y_{MSH} \cdot 2^{32} + Y_{LSH})$$

$$= Q'_{MSH} \cdot Y_{MSH} \cdot 2^{64} + (Q'_{MSH} \cdot Y_{LSH} + Q'_{LSH} \cdot Y_{MSH}) \cdot 2^{32} + Q'_{LSH} \cdot Y_{LSH}$$

由于 B3110A 没有 64 位整数乘法指令，要分解成 4 个 32 位数相乘，乘的结果移位相加才能求得 64 位积。其中 Q'_{MSH}、Q'_{LSH} 分别为 Q' 的高、低 32 位。注意这里的乘法有的有符号，有的无符号。Y_{MSH}，Y_{LSH} 也是同样的含义。

(2)判 $Q' \cdot Y$ 即 P 有否溢出。若有溢出，则表明商 Q' 比真正的 Q 大 1。所以 $Q = Q' - 1$，当 $Q' \geqslant 0$ 时；或者 $Q = Q' + 1$，当 $Q' < 0$ 时。若无溢出，则作(3)。

(3)作 $X - P = R'$。若 $R' = 0$，则表示正好除尽，$Q = Q'$。若 R'，X 异号，则表明商数 Q' 多了 1。所以当 $Q' \geqslant 0$ 时，$Q = Q' - 1$；当 $Q' < 0$ 时，$Q = Q' + 1$。若 R' 和 X 同号，则需进一步作(4)。

(4)作 $|R'| - |Y| = M$。若 $M \geqslant 0$，则 $Q' \geqslant 0$ 时，$Q = Q' + 1$；$Q' < 0$ 时，$Q = Q' - 1$。否则 $Q = Q'$。

2.2 $X \geqslant 2^{53}$，$Y < 2^{10}$

尽管这时 Q' 的整数误差较大，但修正方法比较简单。步骤如下：

(1)作 $X - Q' \cdot Y = R'$。由于 $R' = R_f + E_s$，而 $E_s < 2^{10}$，$R_f < Y_f = Y < 2^{10}$。$R' < 2^{11}$。所以在作 $Q' \cdot Y$ 时只考虑低半部分的乘积。即 $R' = X_{MSH} \cdot 2^{32} + X_{LSH} - (Q'_{MSH} \cdot 2^{32} + Q'_{LSH}) \cdot Y = X_{LSH} - (Q'_{LSH} \cdot Y)_{LSH}$. 这样只需作一次 32 位乘法 $Q'_{LSH} \cdot Y$ 和一次 32 位减法 $X_{LSH} - (Q'_{LSH} \cdot Y)_{LSH}$ 即可得到 R'。

(2)将 R' 变成单精度浮点数 R'_f，Y 转换为单精度浮点数 Y_f。因 $R' < 2^{11}$，$Y < 2^{10}$，单精度浮点数已足能表达了。

(3)作单精度浮点除 $q_f = R'_f / Y_f$.

(4)将 q_f 转换为整数 q'。

(5)形成最终整商 $Q = Q' + q'$.

3 实例研究

3.1 基本硬件结构

作者设计的计算机运算器如图 2 所示。采用微程序控制方式。图中，A、B 寄存器各为 72 位(8 个奇偶位)，其数据来自存贮器，工作频率 20MHz. RH/RL 为结果寄存器，各为 36 位(4 个奇偶位)，同时也能反馈到 B3110A/B3120A 的输入端作为操作数，工作频率为 40MHZ。B3110A/B3120A 内部各有 I、XA、YA、Z、FLAG、MODE、INT 等寄存器，其中 XA、YA 存放数据，Z 存放结果，FLAG 为标志寄存器，MODE 为方式寄存器，INT 为中断寄存器，I 为指令寄存器。只要提供相应的微命令以及数据，B3110A/B3120A 可以并行工作。运算部件为四级流水线结构：A、B 为第一级，XA、YA 处于第二级，Z 为第

三级，RH/RL 为第四级。在微程序实现时，要充分发挥 B3110A/B3120A 的并行特点，以便减少微程序的拍数。

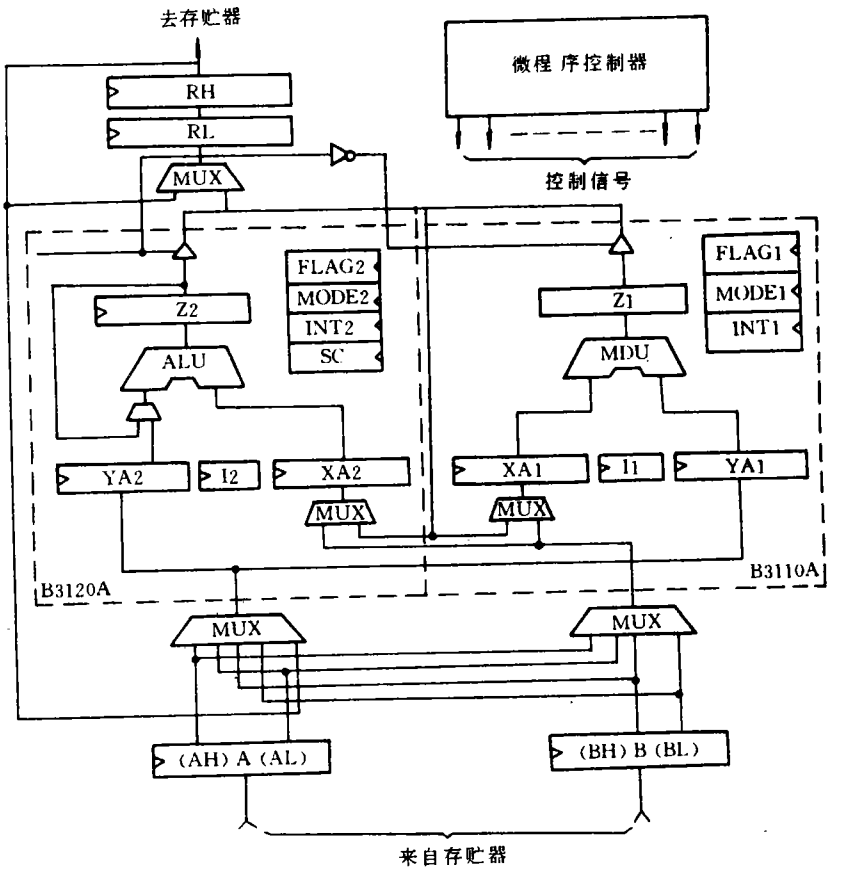

图 2　运算器基本硬件框架

3.2　64 位整除的微程序流程

据前面的讨论可知，若允许整除有±1 的误差，则只有当 $X \geqslant 2^{53}$，$Y < 2^{10}$时才需对商加以修正。下面给出允许有±1 的误差时整除算法的微程序流程，见图 3。读流程时要注意 B3110A/B3120A 尽可能并行操作，这样可以减少很多节拍。

4　结　论

假定被除数 X、除数 Y 落在取值范围内各种数值的概率相同。从 3.2 的流程图可以看出，不修商做一次除法用 16 拍时间，修商时用 29 拍。所以平均拍数可近似为：

$$T = (16 \cdot (2^{63} - 2^{10}) \cdot (2^{63} - 2^{53}) + 16 \cdot 2^{53} \cdot 2^{63} + 29 \cdot (2^{63} - 2^{53}) \cdot 2^{10}) / 2^{126}$$

$$= 16 + 13 \cdot 2^{-53} - 13 \cdot 2^{-63} \approx 16$$

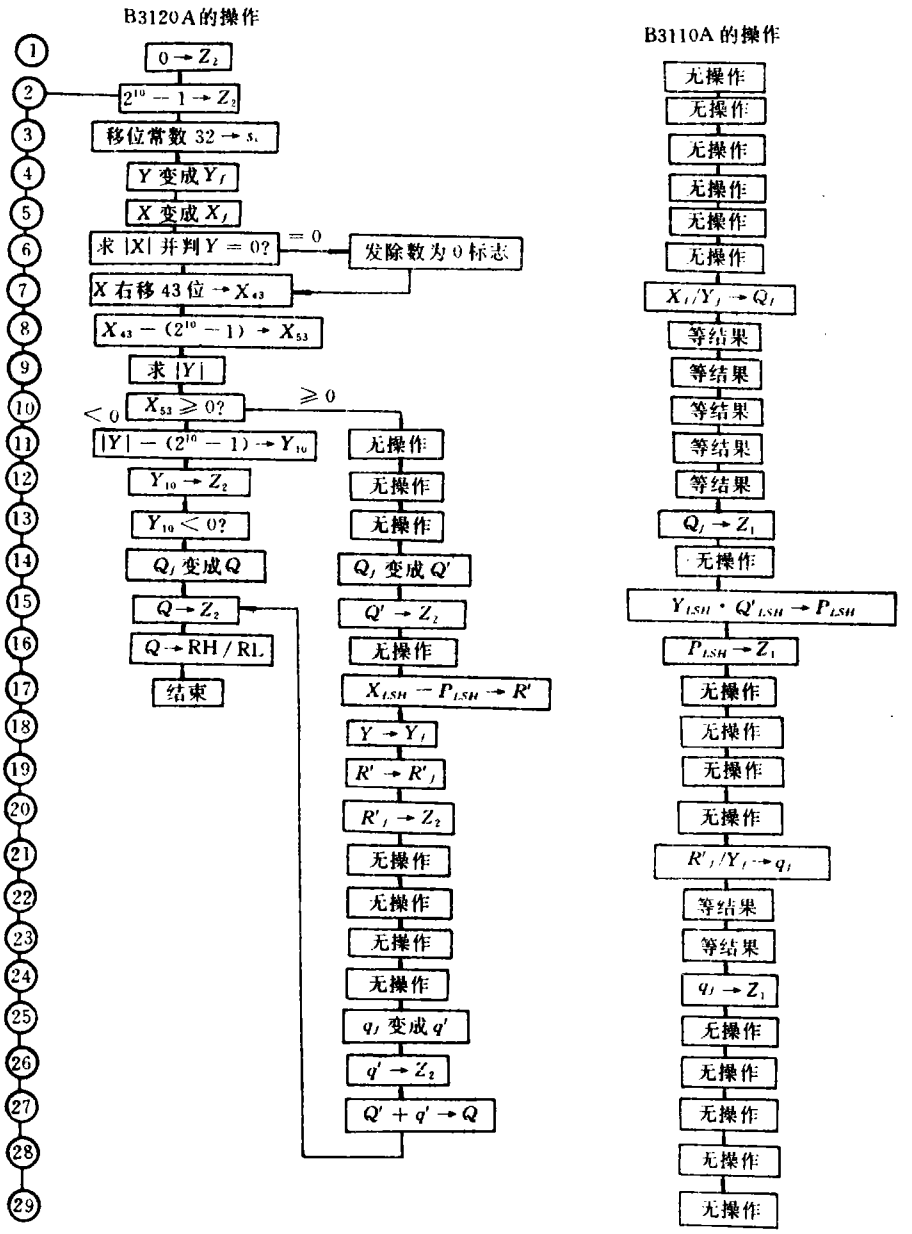

图 3　微程序流程

这是因为只有当 $X \geqslant 2^{53}$，$Y<2^{10}$时才需修商，即要修商的情况远少于不需修商的情况。

若采用补码一位除法，则对 64 位整数除要循环执行 64 次，每次循环包括上商，被除数(余数)和商左移以及一次余数(被除数)加或减除数的操作，还要不断检测循环结束条件。所以总拍数相当多，假设每次循环 3 拍，就要 192 拍。在这种硬件情况下，实现阵列除法、迭代除法都是极为困难的，由此可以看出这一算法的效果。

参考文献

1 白中英，韩兆轩．计算机组成原理教程．北京：科学出版社，1988

2 黄铠．计算机算术运算——原理、结构与设计．北京：科学出版社，1980

3 IEEE. IEEE Standard 754—1985 for binary floating-poing arithmetic, IEEE, 1987

4 David Goldberg. What every computer scientist should know about floating-point arithmetic, ACM computing surveys, 1991(3)

（责任编辑　张　静）

（上接第 65 页）

4 结　语

本文把雷达目标的一维距离像当作一幅弧线图像来进行处理，通过弧线的链码和与之有关的傅里叶系数提取了 5 个的形状特征；然后，利用神经网络技术对其进行识别。实验结果表明，在一定的角度变化范围内，对目标属性的判别效果良好。

对于全姿态角变化范围内的目标识别，可以先将全角度区域化，然后在各个角度范围内利用本文所介绍的方法进行操作，即可获得全姿态角变化范围内的识别结果。对此将另文介绍。该算法为雷达目标一维距离像识别算法的实时处理提供了一条有效途径。

参考文献

1 何松华．国防科技大学博士学位论文，1993

2 边肇祺．模式识别．北京：清华大学出版社，1988

3 张文峰，何明一，林崇平．多层前馈神经网络的一种快速学习算法．西安中国神经网络学术大会，1993，11

4 He S H, Zhuang Z W, Guo G R. A fast millimeter wave imaging algorithm with application to active guidance. SPIE. 1994, 2212: 318～323

5 Guo G R, He S H, Zhuang Z W. milimeter-wave radar target identification by using high resolution range profiles. SPIE. 1944, 2212: 324～332

（责任编辑　潘　生）