单片机浮点数的实用快速除法

中国科学院电工研究所(100080) 张玉明 清华大学电机工程及应用电子技术系(100084) 王 超

摘 要:介绍一种在 8096/98 系列单片机上实现的单精度浮点数快速除法。该算法采用了预估-修正的数值计算方法,并充分利用了 16 位 CPU 中的乘除法指令,计算速度快、精度高,有很强的实用性。

关键词: 浮点数 除法 尾数 预估-修正 误差 精度

在较为复杂的单片机系统中,为扩大取值范围,实现复杂的计算和控制,一般都要涉及浮点数的运算。而一般单片机是没有浮点数运算指令的,必须自行编制相应软件。在进行除法计算时,通常使用的方法是比较除法[1],即利用循环移位和减法操作来得到24~32 位商,效率很低。有些文献给出了一些改进方法[2],但思路不清晰,很难推广使用。这里给出一种浮点数除法运算的实用快速算法。该方法以数值计机的乘除法功能,很轻易地实现了浮点数的除法。

1 浮点数格式

IEEE 的浮点数标准规定了单精度(4字节)、双精度(8字节)和扩展精度(10字节)三种浮点数的格式。最常用的是单精度浮点数,格式如图1所示。但是这种格式的阶码不在同一个字节单元内,不易寻址,从而会影响运算速度。



图 1 IEEE 单精度浮点数格式

通常在单片机上采用的是一种变形格式的浮点数,如图 2 所示。其中的 23 位尾数加上隐含的最高位 1,构成一个定点原码小数,即尾数为小于 1 大于等于 0.5 的小数。有关浮点数格式的详细内容请参考有关文献[1][2]。



图 2 IEEE 的单精度浮点数的变形

2 快速除法的算法原理

在 16 位单片机中只有 16 位的乘除法,而浮点数的精度(即尾数的有效位数)达 24 位,因此无法直接相除,但仍然可以利用 16 位的乘除法指令来实现 24 位除法。不过,如果只进行一次 16 位的除法必定会带来很大误差,因此问题的关键在于如何消除这个误差,从而达到要求的精度。这其实就是通常数值计算中所采用的预估-修正方法。

假设两个浮点数经过预处理后,被除数和除数尾

数扩展为 32 位 (末 8 位为 0)分别放入 X 和 Y 中。令 Y_L 为 Y 的低 16 位,并记 $Y_H = Y - Y_L$ 。显然 $Y_H \approx Y, X/Y$ 与 X/Y_H 相差不多:

$$\begin{split} \frac{X/Y}{X/Y_H} &= \frac{Y_H}{Y} \\ &= \frac{Y_H}{Y_H + Y_L} = \frac{1}{1 + Y_L/Y_H} \\ &\approx 1 - \frac{Y_L}{Y_H} = \frac{Y_H - Y_L}{Y_H} \end{split}$$

可见只需要在 X/Y_H 的基础上再乘以一个修正因子 $(Y_{H}-Y_L)/Y_H$,就可以得到 X/Y 的一次校准值。不难证明这个值已经达到了 24 位的精度要求。事实上,相对误差满足:

$$\Delta = \left| \frac{\frac{X}{Y_H} \frac{Y_H - Y_L}{Y_H} - \frac{X}{Y}}{\frac{X}{Y}} \right|$$

$$= \left| \frac{\frac{X}{Y} \frac{Y_{H^2} - Y_{L^2}}{Y_{H^2}} - \frac{X}{Y}}{\frac{X}{Y}} \right| = \left| \frac{Y_{H^2} - Y_{L^2}}{Y_{H^2}} - 1 \right|$$

$$= \left(\frac{Y_L}{Y_H} \right)^2 \le \left(\frac{0 \text{ ff } 00 \text{ H}}{80000000 \text{ H}} \right)^2$$

$$\approx 0.002 \times 2^{-30}$$

这说明这个一次校准值完全可以作为最终的结果。

3 算法的具体实现

在具体实现本算法时,主要经过下列步骤:

(1)计算预估值
$$Q_0 = \frac{X}{Y_u}$$
;

(2)计算修正因子
$$Q_1 = \frac{Y_H - Y_L}{Y_H}$$
;

(3) 计算校准值 $Q = Q_0 \times Q_1$, 并作为最后结果。

这里的 Y_H 虽仍是 32 位,但其低 16 位已为 0,计算时可以将它视为 16 位数,这不会影响计算精度。通过两次 16 位除法,就可得到精确的 32 位结果。例如,计算 Q_0 时,第一次除法,X 除以 Y_H 的高 16 位,得到的商为 Q_0 的高 16 位,而 16 位余数末尾添 0 成 32位,再除以 Y_H 的高 16 位,得到 Q_0 的低 16 位(余数舍去)。由此得到了 32 位的 Q_0 。

在具体运算中, X应先除以4(X右移2位),以保 证 Q_0 不会溢出 $(Y_H$ 取高 16 位):

$$Q_0' = \frac{X/4}{Y_H} \le \frac{0 ffffff 00 H/4}{8000 H} = 7 fff h$$

由于X为32位(末8位为0),这一操作不影响 有效数字。而 $Q_1 = \frac{Y_H - Y_L}{Y_H} \le 1$,不存在溢出的问题。

最后计算校准值 Q 时,有 $Q=4Q_0'Q_1$ 。

在计算 $Q_0' \setminus Q_1$ 时,均进行了两次 16 位除法,使得 $Q_0' \setminus Q_1$ 均为精确的 32 位,保证了计算过程中的精度, 减小了累积误差。对于 $Y_L=0$ 即除数只有 16 位有效数 字的特殊情况,直接有 $O_1=1$,还能省去两次16位除法。

在计算 Q 时,则通过 3 次 16 位乘法实现了 32 位 乘法,取结果的高 32 位,即得 Q。

整个算法至多只须用4次除法、3次乘法和5次 加法,就求得了浮点数商的尾数,可见计算效率是很 高的,保证了运算速度。

浮点数除法流程图如图 3 所示。

4 程序源代码

限于篇幅,只给出源代码中的关键部分,即有效 数字的计算部分。

;被除数为 x,除数为 y

;用 yh,yl 分别表示 y 的高 16 位和低 16 位

;假设x,y的有效数字部分分别在(dx,cx)和(bx,ax)中

; 计算预估值 Q₀'=(x/4)/yh

:计算 x/4 shrl cx, #2 divu cx, bx ; 计算(x/4)÷vh

ld fx, cx :把商暂放入寄存器 fx,即 O_0 '的高 16 位有

;效数字

;把余数末尾添 0 后再除以 vh divu cx, bx ld ex, cx ;把商暂放入寄存器 ex, 即 Q。'

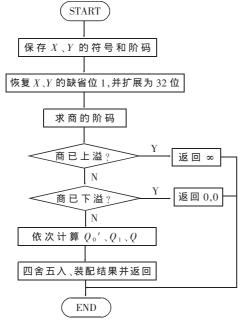


图 3 浮点数除法流程图

;的低 16 位有效数字

 $;(fx,ex) = Q_0'$

; 计算修正因子 Q₁=(yh-yl)/yh

cmp ax, 0 ;判断 yl 是否为 0

;若 yl 非 0,计算修正因数 Q1 ine getQ1

1d;若 yl=0, 修正因数 Q1=1 ax. ex

;(Q₀'×Q₁)=Q₀',可以直接计算 Q bx, fx

sjmp getQ

 $getQ_1$: ld hx, bx

:把 vh 放于寄存器 hx 中

neg ax

dec;计算 yh-yl bx

; 计算 Q1=(yh-yl)÷yh divu ax. hx

;把商暂时放入寄存器 dx,即 Q1 的高 16 位有 dx, ax

:效数字

clr ax

;把余数末尾添0后再除以 yh,得 Q」的 divu ax, hx

;低 16 位有效数字

ld by dy $;(bx,ax) = Q_1$

; 计算 $O_0' \times O_1 = (fx, ex) \times (bx, ax)$, 只取 32 位有效数字

ld hx. bx

mulu cx, bx, ex $;(dx,cx) = bx \times ex$ mulu ax, fx $;(bx,ax) = ax \times fx$

add cx. ax

adde dx, bx

addc ex. 0 ; (ex, dx, cx) = (dx, cx) + (bx, ax)

mulu ax, fx, hx $;(bx,ax) = fx \times hx$

add ax, dx ;(bx,ax) = (bx,ax)+(ex,dx)addc bx. ex $(bx,ax) = Q_0' \times Q_1$

: 计算校准值 $O = (O_0' \times O_1) \times 4$ 并调整阶码 getQ:

代码到这里为止,浮点数商的有效数字已经全部 求出。只要再执行一些调整浮点数阶码的操作,就可 以得到最终结果。

在作者开发的一个 80C196KC 单片机系统中,涉及到 了二进制-十进制数制转换、分段线性插值、数字滤波等大 量浮点数的运算,都是靠加减乘除等底层函数来实现的。

此外,本算法思路清晰,因此很容易加以推广。例 如,为了得到更高的精度,可取修正因子:

$$Q_{1} = \frac{Y_{H}^{2} - Y_{H}Y_{L} + Y_{L}^{2}}{Y_{H}^{2}}$$

则 相 对 误 差 $\Delta = \left(\frac{Y_L}{Y_H}\right)^3 \leqslant 2^{-45}$, 转 化 为 十 进 制,有 效数字高达 14 位。

参考文献

- 1 复旦大学计算机系微机开发研究室.十六位单片机 8096的原理和设计方法.重庆:科学技术文献出版社重 庆分社,1988
- 2 涂时亮,姚志石.单片微机 MCS-96/98 实用子程序. 上海:复旦大学出版社,1991
- 3 李庆扬.数值分析. 武汉:华中工学院出版社, 1986

(收稿日期:2000-11-20)