插值开平方的优化及实现

陈锡华,陈 磊
(桂林航天工业学院,广西 桂林 541004)

【摘 要】将被开方数转换为[0.25, 1)范围内进行线性插值计算开平方近似结果;在精度不被劣化情况下,将[0.25, 1)区间分为[0.25, 0.5)和[0.5, 1)两段,分别采用单倍和双倍的间隔构造插值节点,使插值节点数据表减少三分之一;将插值误差纳入节点值,不增加额外运算量使插值绝对误差下降一半。给出了以一次乘法运算、两次加/减法和少量移位操作获得开平方结果的实现方法。

【关键词】开平方;误差;插值;节点
【中图分类号】TP301. 6 【文献标识码】A 【文章编号】1673-1891(2012)04-0029-02

引言

在实际应用系统中,通常会遇到开方运算,高复杂度的开方运算往往要占用较多的系统资源。迭代逼近算法,如Newton - Raphson算法[1]、SRT -Redunant算法[2],是通过循环迭代获得开平方结果;消耗系统运行时间多,不适用于高速系统;查表法,速度快,但高精度运算指标要求将导致数据表变得庞大,系统存储资源开销大,常用于精度不高的场合,实际工程中经常使用[4]。采用插值法可减少数据表容量,避免了除法运算和迭代,速度快,精度高,适用于运算能力不高的系统。

1 插值开平方

采用高阶插值算法可以提高精度,但运算复杂度跟着提高,且累积误差增大。不适用于运算能力低,应用广泛的单片机系统。

1.1 线性插值区间

对于二进制表示的正数C,经过每次移位两位的反复操作,总可以表达为[6]:$C=4^n x$,其中n为整数,x为小数,且0.25≤x<1;C的平方根A:

$$A=\sqrt{C}=2^n\sqrt{x} \qquad (x\in[0.25,1);n\in Z) \tag{1}$$

式(1)中,2^n通过移位容易实现,C的开平方运算复杂度主要是$y=\sqrt{x}$的开平方运算。将插值区间[0.25, 1]扩充到[0, 1]区间实施等间隔分为2^m($m\in N$)段,共2^m+1个节点,仅使用[0.25, 1]范围内的节点。则第i节点有$x_i=i/2^m$(i=0,1,2,……,2^m)和$y_i=\sqrt{x_i}=\sqrt{i/2^m}$。对区间内任意$x\in[x_i, x_{i+1}]$,采用线性插值函数,有近似开平方结果[5]:

$$y\approx N_1(x)=y_i+\frac{y_{i+1}-y_i}{x_{i+1}-x_i}(x-x_i)=(2^m x-i)(y_{i+1}-y_i)+y_i \tag{2}$$

1.2 插值误差

式(2)绝对误差 $\varepsilon_i=|y-N_1(x)|=|R_1(x)|$,$R_1(x)$为插值余项:

$$|R_1(x)|=\left|\frac{y''(\xi)}{2!}(x-x_i)(x-x_{i+1})\right|=\frac{1}{8}\xi^{-\frac{3}{2}}(x-x_i)(x_{i+1}-x)$$

其中 $\xi\in(x_i,x_{i+1})$,易知在$x=\frac{x_i+x_{i+1}}{2}$时,有$(x-x_i)(x_{i+1}-x)\le\frac{h^2}{4}=2^{-(2m+2)}$,得:

$$\varepsilon_{i\max}\le 2^{-5}\xi^{-\frac{3}{2}}h^2\le 2^{-(2m+5)}\cdot x_i^{-\frac{3}{2}} \tag{3}$$

式(3)表明了$[x_i, x_{i+1}]$区间内的最大插值绝对误差;x_i越小,误差ε_i越大。$[x_i, x_{i+1}]$区间内对应的最大相对误差e_{imax}:

$$e_{i\max}=\frac{\varepsilon_{i\max}}{\sqrt{x}}<\frac{2^{-(2m+5)}\cdot x_i^{-\frac{3}{2}}}{\sqrt{x_i}}=2^{-5}i^{-2} \tag{4}$$

式(3)表明,在[0.25, 1]范围内,在x_i=0.25对应的区间,ε_{imax}有最大值$2^{-(2m+2)}$。注意到$y=\sqrt{x}$的单调递增性,$R_1(x)\le 0$,式(2)加上$2^{-(2m+2)}$可使最大绝对误差下降一半,即在[0.25, 1)范围内有最大误差$\varepsilon_{max}=2^{-(2m+3)}$。在构造节点数据表时,节点值$y_i$都加上$2^{-(2m+2)}$,在不改变式(2)的运算表达式情况下,相当于加上$2^{-(2m+2)}$;这样处理可减少一次加法运算。显然,在$x_i$=0.25对应的区间相对误差最大,其值为:

$$e_{\max}=\frac{\varepsilon_{\max}}{\sqrt{x}}\le\frac{2^{-(2m+3)}}{\sqrt{x_i}}=2^{-(2m+2)} \tag{5}$$

式(5)可用于估算精度和确定在一定的误差e_c要求下m的大小,如要求结果保证四位有效数字,即$e_c<10^{-4}$,可算出m=6,$e_{max}\approx 6.1\times 10-5<e_c$。

1.3 节点数据表容量

式(4)表明,i每增大2倍,e_{imax}下降1/4;在[0.25, 1)范围内,x_i=0.25对应$[x_i, x_{i+1}]$区间最大,其2倍处x_i=0.5对应$[x_i, x_{i+1}]$区间e_{imax}将下降1/4;为减少插值节点的数据表容量,使[0.5, 1]内的插值区间间隔h

收稿日期:2012-11-07
作者简介:陈锡华(1969-),男,广西博白人,副教授,主要研究方向为无线通信技术、单片机技术应用。

增大一倍，由式(3)可知，ε_{imax}增大4倍，这样，e_{imax}在x_i=0.5和x_i=0.25对应$[x_i, x_{i+1}]$区间基本相同。即将[0.25, 1)分为[0.25, 0.5)和[0.5, 1)两段，分别采用h和$2h$的插值区间间隔，这两段的最大相对误差基本相同。这样处理可减少节点数据表容量，使在[0.25, 1)内原来的$3/4\times2^{m+1}$个节点变为$2^{m-1}+1$个，节点数据表约减少1/3。表1是m=6的节点数据表，表中数据表用十六进制表示，节点数据已考虑了加上$\varepsilon_{imax}=2^{-14}=(0.0004)_{16}$。$k$是对节点的重新编号，$k$=16~32相当于$m$=5对应[0.5, 1)段的节点，这样，式(2)修改为：

$$y\approx\begin{cases}(2^m x-i)(y_{k+1}-y_k)+y_k & x\in[0.25,0.5),k=i-2^{m-2};\\(2^{m-1}x-i)(y_{k+1}-y_k)+y_k & x\in[0.5,1),k=i;\end{cases}\qquad(6)$$

表1 $y=\sqrt{x}+\varepsilon_{max}$等距节点值($m$=6，$k$=0~15：$h$=1/64，$k$=16~32：$h$=1/32)

k	i	y	k	i	y	k	i	y
0	16	0x8004	11	27	0xA64B	22	22	0xD44A
1	17	0x 83F4	12	28	0xA958	23	23	0xD90D
2	18	0x87C8	13	29	0xAC57	24	24	0xDDB8
3	19	0x8B80	14	30	0xAF49	25	25	0xE24A
4	20	0x8F20	15	31	0xB22F	26	26	0xE6C5
5	21	0x92A8	16	16	0xB509	27	27	0xEB2B
6	22	0x961C	17	17	0xBA9B	28	28	0xEF7B
7	23	0x997B	18	18	0xC004	29	29	0xF3B8
8	24	0x9CC8	19	19	0xC547	30	30	0xF7E1
9	25	0xA004	20	20	0xCA67	31	31	0xFBFC
10	26	0xA32F	21	21	0xCF66	32	32	0x10004

2 算法实现

$\sqrt{x}$可通过式(6)进行计算获得近似结果。设微控制器运算位数为r，每单元数据比特为$(D_{r-1}\ D_{r-2}\cdots D_1D_0)_2$。被开方数$C$的整数和小数分别存储于两个单元。算法主要包括以下几个步骤：

(1)将C每两位逻辑左移一次，直到整数单元$D_{r-1}D_{r-2}$有非0比特，移动次数为p，则$n=r/2-p$。移位结果为小数x。

(2)将x逻辑左移m-1比特，若最高位为“1”，移出比特为$k=(D_{r-1}\ D_{r-2}\cdots D_{r-m+1})_2$；剩余为$2^{m-1}x$-i。若最高位为“0”，$x$再逻辑左移1比特，移出比特弃掉高两位，取为$k=(D_{r-3}\ D_{r-4}\cdots D_{r-m})_2$；剩余为$2^m x$-$i$。由$k$查表得$y_k$、$y_{k+1}$。

(3)计算式(6)，获得y。

(4)逻辑左移(n为负右移)n比特，移出部分为整数，剩下取高单元数据为小数。即为开平方结果。

3 算法实例

使用r=16的十六位微控制器[7]，要求结果四位有四位有效精度，即$e_c<10^{-4}$，可知m=6，插值节点数据表见表1，设被开方数$C=(333.32)_{10}\approx(014D.51EC)_{16}$。

(1) $(014D.51EC)_{16}=(0000,0001,0100,1101.0101,0001,1110,1100)_2$，数据每两位左移3次后：$(0101,0011,0101,0100,0111,1011,0000,0000)_2$；有16比特可保证$e_c<10^{-4}$，$x$取高16比特数据进行运算即可，得$n$=16/2-3=5；$x=(0.0101,0011,0101,0100)_2$。

(2)左移5比特后：$(01010.0110,1010,1000,0000)_2$，最高位为“0”，再逻辑左移1比特，移出比特弃掉高两位，$k=(0100)_2=(4)_{10}$，

对应 $2^6x-20=(0.1101,0101,0000,0000)_2=(0.D500)_{16}$。查表1得：

$y_4=(0.8F20)_{16}$，$y_5=(0.92A8)_{16}$。$y_5-y_4=(0.0388)_{16}$

(3)计算式(6)，$y=(2^6x-20)(y_5-y_4)+y_4$。

$(0.D500)_{16}\times(0.0388)_{16}+(0.8F20)_{16}=(0.9210,2800)_{16}$。将$(0.9210,2800)_{16}$逻辑左移5比特：$(12.4205,0000)_{16}\approx(18.257889)_{10}$。

实际计算$\sqrt{333.32}\approx18.257053$，相比结果，算法优于四位有效值精度。

3 结束语

使用上述插值算法进行开方运算仅需一次乘法、两次加/减法和少量的移位运算，无需除法运算，运行速度快。在考虑结果精度时要求运算精度相配合，实例中可以看出，十六位运算精度是$1/2^{16}\approx1.526\times10^{-5}$，满足了四位有效值精度要求，达万分之一精度的节点数据表消耗66字节单元存储器；这样的存储开销是可以接受的。

注释及参考文献：

[1]张小明,李永新.基于牛顿迭代法的高精度快速开方算法[J].电力自动化设备,2008.28(3):75−77.

[2]张新贺,张月华.Non − Restoring开方算法及FPGA实现[J].信息技术, 2007.17(8):80−82.

[3]石一辉,易攀,张承学.快速开方算法在微控制器上的实现[J].计算机技术与发展, 2007.17(4):80−82.

[4]陈明凯.一种归一化开方算法的单片机程序实现[J].厦门大学学报(自然科学版),2001.40:150−152.

[5]曾绍标,熊宏许,毛云英等.应用数学基础(下册)[M].天津:天津大学出版社,2003.7.

[6]裴亚强,胡仁杰.移位技术在交流采样计算中的应用研究[J].电气电子教学学报,2003.25(5):30−33.

[7]罗亚非等编著.凌阳16位单片机应用基础[M].北京：北京航空航天大学出版社,2003.12.

(下转46页)

[4]曹光跃.模拟电子技术及应用[M].北京:机械工业出版社,2008.
[5]林春方,杨建平.模拟电子技术[M].北京:高等教育出版社,2006.
[6]朱钰铧.电路基础实验实训指导教程[M].合肥:安徽大学出版社,2008.

Scheme & Realization of Lab Improvement Based on Plug-in Board

ZHANG Ren-lin, ZHANG Liu-zhong
(Experiment Training Center, Anhui Vocational College of Electronics & Information Technology, Bengbu, Anhui 233000)

Abstract: The performance of the current electrical and electronic laboratories is analyzed. The plug-in board, component plug-in box, conductor and equipment box in the electrical and electronic laboratories are improved after careful redesigns. The regular experiment projects and comprehensive training projects after technical improvement are tested, which gains good results.

Key words: Scheme; Plug-in board; Experiment project; Comprehensive test

(上接30页)

Optimization and Realization of Interpolation Open Square

CHEN Xi-hua, CHEN Lei
(Guilin Aerospace Industry College, Guilin,Guangxi 541004)

Abstract: Radicand number is converted to [0.25, 1) within the range of the linear interpolation calculation open square approximation result; Under the accuracy not being deteriorated case, [0.25, 1) is divided into [0.25, 0.5) and [0.5, 1) two sections, respectively using single and double interval constructed interpolation nodes, so the interpolation node data table reduced by one third; bringing interpolation error into the value of the node, not to increase the additional amount of computation so that the the interpolated absolute error decreased half . the paper gives one multiplication, two plus / subtraction, and a small amount of shift operations to gain the implement methods of the square root.

Key words: Open square; Error; Interpolation; Node