

一种定点化平方根倒数运算的硬件实现

夏顺兴 罗小华

(浙江大学 电气工程学院, 浙江 杭州 310027)

摘要: 针对平方根倒数运算电路中传统的多次迭代法占用较多运算单元, 以及多项式逼近法占用大量存储单元的问题, 提出一种基于分段二次多项式逼近与牛顿迭代相结合的定点化平方根倒数运算的硬件实现方法。结合两种方法的优点, 即运用少量存储单元存储二次多项式系数用于求解迭代初值; 然后对迭代初值进行一次牛顿迭代使根快速收敛。其中, 对系数及过程变量都进行定点化处理, 避免复杂的浮点运算。实验结果表明, 该实现方法仅需584bit存储单元及少量乘加运算单元, 求解误差小。

关键词: 定点化; 平方根倒数; 多项式逼近; 牛顿迭代

基金项目: 浙江省自然科学基金(LY15F040001)

The Hardware Implement of Fixed-Point Reciprocal Square Root

Xia Shun-xing Luo Xiao-hua

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)

In order to solve the problem that repeatedly iterative methods occupy more computing units and the polynomial approximation method occupies a large number of storage units, a hardware implement of fixed point reciprocal square root(RSR) based on segmented quadratic approximation formula with Newton iteration is proposed. Combining the advantages of the two methods, a quadratic polynomial coefficient is stored in a little memory cells to solve the iteration seed. Then, one time Newton iteration is performed on the iteration seed to make the root converge rapidly. In the mean time, the coefficient and process variables are fixed-point processing, to avoid complex floating-point operations. The results show that this method requires only 584bit memory cells and a small number of multiply-add units, and the solution error is small.

Key words: Fixed-point; square root; polynomial approximation; Newton iteration

0 引言

在图像处理及3D图形编程时, 经常需要求解特征向量的长度或者将向量归一化^{[1][2]}, 其中尤为关键的运算便是平方根倒数运算^[3]。而开平方根运算与倒数运算都是相对比较复杂的过程, 如果在硬件中使用常规的浮点运算单元(FPU)来求解的话需要较多的计算时间。在一些对图像处理运算速度要求较高的场合, 可以适当降低平方根倒数运算解的精度来提高电路的运算速度^[1]。

浮点数的运算需要用到查找表, 会占用大量的ROM存储资源且计算过程复杂, 因此将浮点运算进行定点化处理可以在精度要求允许的范围内有较地降低硬件电路的面积^[4]。平方根倒数运算还经常采用牛顿迭代法^[5], 其优点是电路所需的存储资源小, 解得根的精度较高, 然而多次迭代所占用的运算资源较大, 运算速度也较慢, 并且对迭代初值的要求比较高。

因此, 本文提出了一种基于分段二次多项式与牛顿迭代相结合的定点化平方根倒数运算的实现方

法。该实现方法通过分段二次多项式获取迭代初值, 然后进行一次牛顿迭代运算提高结果的精度。

1 分段二次多项式

分段二次多项式是将某一复杂函数 $f(x)$ 的局部区间分为若干区间段, 并取每段的三个点 x_{i-1} , x_i , x_{i+1} 进行二次插值, 即取

$$f(x) \approx p_2(x) = \sum_{k=i-1}^{i+1} y_k \prod_{j \neq k} \left(\frac{x - x_j}{x_k - x_j} \right) \quad (1.1)$$

即用二次插值多项式 $p_2(x)$ 逼近原函数 $f(x)$ 。其中二次多项式(1.1)可以写成以下标准式:

$$p_2(x) = c_0 + c_1x + c_2x^2 \quad (c_2 \neq 0) \quad (1.2)$$

本文选取16位无符号整数作为输入值, 由于倒数运算不能将0作为除数, 所以输入值的取值范围为:

$x \in [1, 2^{16})$ 。输出设为 x 平方根倒数的 2^{16} 倍, 即 $f(x) = \frac{2^{16}}{\sqrt{x}}$ 的函数值(17位无符号整数), 将输出结果扩大是为了便于以定点数形式运算。为了获得牛顿迭代初值, 首先对输入数 x 左移 $2n$ 位, 进行规格化放大: 将 x 左移得到 x_{regular} , 使得 $x_{\text{regular}} \in [2^{14}, 2^{16})$

然后对函数 $f(x) = \frac{2^{16}}{\sqrt{x}} = \frac{2^{16}}{\sqrt{x_{\text{regular}}}} \cdot 2^{\frac{5}{2}}$ 在区间 $[2^{14}, 2^{16})$ 上进行分段二次插值。本文方法是将区间等分为8段, 并取每段的两个端点和中点做函数 $f(x) = \frac{2^{16}}{\sqrt{x}}$ 的二次插值多项式。根据式(1.1)或(1.2)计算得二次多项式系数 c_0, c_1, c_2 , 见表1。

表1 分段区间端点及二次插值多项式系数

区间左端点	区间右端点	区间中点	c_0	c_1	c_2
16384	22528	19456	887.9605905	-0.030714366	4.74094E-07
22528	28672	25600	771.4965648	-0.020200343	2.36866E-07
28672	34816	31744	691.7181577	-0.014578272	1.37827E-07
34816	40960	37888	632.5959526	-0.011158514	8.83782E-08
40960	47104	44032	586.4887584	-0.008895983	6.06227E-08
47104	53248	50176	549.2159633	-0.00730747	4.36979E-08
53248	59392	56320	518.2680625	-0.006141657	3.27189E-08
59392	65536	62464	492.0339641	-0.005256156	2.52466E-08

由二次多项式的标准式(1.2)与表1中的系数可表示8个二次多项式, 分别在对应区间内近似替代函数

$$f(x) = \frac{2^{16}}{\sqrt{x}}。$$

2 系数定点化与迭代初值

2.1 系数定点化

由表1得到的系数 c_0, c_1, c_2 均为小数, 不利于硬件的运算, 所以可以将系数扩大一定倍数后取其整数部分以定点数形式存储^[5-7]。系数定点化可以有效避免硬件的浮点运算。考虑到系数定点化后的误差影响, 我们将系数 c_2 扩大 2^{46} 倍后以25位无符号整数形式存储; 将系数 c_1 扩大 2^{31} 倍后以32位有符号整数形式存储; 将系数 c_0 扩大 2^6 倍后以16位无符号整数形式存储。定点化后的系数如表2所示。由此计算可得存储定点系数所占ROM资源为584bit。

表2 二次插值多项式系数定点化

系数浮点数			系数定点化		
c_0	c_1	c_2	c_0	c_1	c_2
887.9605905	-0.030714366	4.74094E-07	56829	-65958600	33361387
771.4965648	-0.020200343	2.36866E-07	49376	-43379905	16667931
691.7181577	-0.014578272	1.37827E-07	44270	-31306600	9698743
632.5959526	-0.011158514	8.83782E-08	40486	-23962727	6219063
586.4887584	-0.008895983	6.06227E-08	37535	-19103979	4265941
549.2159633	-0.00730747	4.36979E-08	35150	-15692673	3074964
518.2680625	-0.006141657	3.27189E-08	33169	-13189107	2302384
492.0339641	-0.005256156	2.52466E-08	31490	-11287509	1776574

2.2 迭代初值

根据规格化的输入值 x_{regular} 所在的区间段, 由表2查得二次多项式的系数。迭代初值 y_0 由二次插值多项式(1.2)求得。为了减少乘法运算次数, 将多项式(1.2)改写成:

$$p_2(x) = c_0 + (c_1 + c_2 x) \cdot x \quad (c_2 \neq 0) \quad (2.1)$$

所以用于求解迭代初值的硬件运算资源为: 2次乘、2次加法, 以及一些用于查找系数地址的比较器与选择器。在运用(2.1)式求解初值的时候需要注意的是各系数的放大倍数, 在做加法运算时应对数值位数进行适当取舍。硬件实现结构如图1所示。

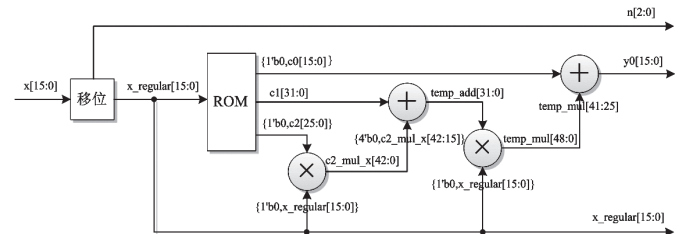


图1 生成迭代初值的硬件结构

图中所有变量均已定点化处理:

$x, x_{\text{regular}}, c_0, c_2$ 均为无符号正数, 在做运算时在其最高位添1位“0”作为有符号数; $c_2_mul_x$ 作为 c_2 乘 x_{regular} 的结果, 其小数位数与 c_2 相同(46位), 又因为 c_1 的低31位表示小数部分, 故取 $\{4'b0, c2_mul_x[42:15]\}$ 与 c_1 小数对齐并相加; 其结果 $temp_add$ 的小数部分也是低31位, 与 x_{regular} 相乘后的结果为 $temp_mul$, 其中高18位为整数部分, 低31位为小数部分; c_0 低6位为小数部分, 故取 $temp_mul[41:25]$ 与 $\{1'b0, c0[15:0]\}$ 对齐相加, 其结果为17位有符号数, 截取低16位作为无符号数输出 $y0[15:0]$, 其低6位为小数部分。

3 牛顿迭代法

牛顿迭代法是求解非线性方程 $f(x)=0$ 的一种重要的迭代法, 其基本思想是将非线性函数 $f(x)$ 线性化, 从而将非线性方程 $f(x)=0$ 近似地转化为线性方程求解:

$$f(x_k) + f'(x_k)(x - x_k) = 0$$

并由此线性方程得到牛顿迭代公式:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad (k = 0, 1, 2, \dots) \quad (3.1)$$

为求解函数 $y = \frac{2^{16}}{\sqrt{x}}$, 可以构造函数:

$$f(y) = \frac{2^{32}}{y^2} - x \quad (3.2)$$

其中 y 为迭代变量值, x 在迭代过程中视为常量。

将函数(3.2)代入迭代公式(3.1), 并将规格化后的 x_{regular} 代替 x , 可得到本文的迭代公式:

$$y_{k+1} = \frac{3}{2}y_k - \frac{1}{2^{32}}x_{\text{regular}} \cdot y_k^3 \quad (3.3)$$

由于牛顿迭代法是以平方收敛的速度收敛, 而且本文已经运用二次多项式插值法得到一个初值 y_0 , 所以本文只需作一次牛顿迭代即可得到合适精度的

解。由迭代公式(3.3)可知,一次牛顿迭代只需进行3次乘法运算、1次加法运算、1次减法运算即可得到结果(乘2的N次幂系数可以直接对结果取对应高位,不需占用乘法器)。硬件结构如图2所示。

最后,需要对函数 $f(x_{\text{regular}}) = \frac{2^{26}}{\sqrt{x_{\text{regular}}}} = \frac{2^{16}}{\sqrt{x}} \cdot 2^{-n} = f(x) \cdot 2^{-n}$ 进行移位,得到最终函数值 $f(x)$ 。

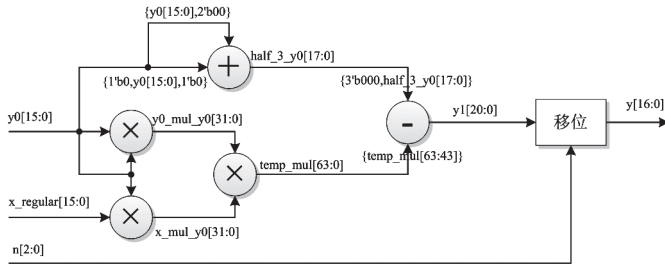


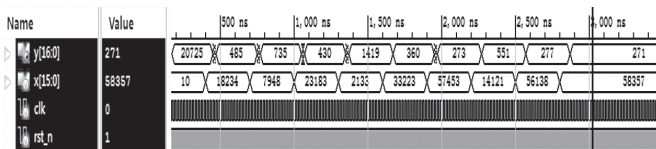
图2 一次牛顿迭代的硬件结构

同样,变量均定点化处理:

y0低6位为小数部分, half_3_y0表示, 低8位为小数部分; 而y0_mul_y0表示低12位为 $\frac{3}{2}y0$ 部分的32位定点数, x_mul_y0表示低6位为小数部分的32位定点数, temp_mul表示低18位为小数部分的64位定点数; 由于系数 2^{-33} , 取temp_mul的低51位作为小数部分; 因此减法运算时取{3'b00, half_3_y0[17:0]}与{temp_mul[63:43]}(其中高13位为整数部分, 低8位为小数部分)相减, 其结果y1经过移位后得到最终输出y[16:0]。

4 实验仿真

将该运算电路用Verilog语言实现, 并生成一些随机输入数据, 在ISim仿真环境下得到如图电路仿真波形。



由仿真运行结果、精确值可计算得误差见表3, 可知误差不会超过。

表3 仿真结果及误差

序号	输入值 x	输出值 y	$\frac{1}{\sqrt{x}}$ 仿真结果	$\frac{1}{\sqrt{x}}$ 精确值	误差
1	10	20725	0.316238403	0.316227766	1.06373E-05
2	18234	485	0.007400513	0.007405579	5.0664E-06
3	7948	735	0.01121521	0.011216854	1.64413E-06
4	23183	430	0.006561279	0.006567728	6.44907E-06
5	2133	1419	0.021652222	0.021652327	1.05069E-07
6	33223	360	0.005493164	0.005486313	6.85111E-06
7	57453	273	0.004165649	0.004171994	6.34429E-06
8	14121	552	0.008422852	0.008415255	7.59670E-06
9	56138	277	0.004226685	0.004220574	6.11045E-06
10	58357	271	0.004135132	0.004139554	4.42187E-06

5 结语

本文提出了一种定点化平方根倒数运算的方法, 并用Verilog描述语言实现硬件电路设计及实验仿真。该电路设计采用分段二次多项式逼近与一次牛顿迭代相结合的方法得到误差值小于的结果, 硬件只需584bit的ROM用于存储二次插值多项式系数以减少多次牛顿迭代所占用的运算逻辑资源。

参考文献:

- [1] Kim K, Hoosung-Lee, Cho S, et al. Implementation of 3D graphics accelerator using full pipeline scheme on FPGA[C]// SoC Design Conference, 2008. ISOC '08. International. IEEE, 2008: II-97-II-100.
- [2] Devi A, Gandhi M, Varghese K, et al. Accelerating method of moments based package-board 3D parasitic extraction using FPGA[J]. Microwave & Optical Technology Letters, 2016, 58(4):776-783.
- [3] Jain V K, Lin L. Square-root, reciprocal, sine/cosine, arctangent cell for signal and image processing[C]// IEEE International Conference on Acoustics, Speech, and Signal Processing. IEEE, 1994:II/521-II/524 vol.2.
- [4] Suresh S, Beldianu S F, Ziavras S G. FPGA and ASIC square root designs for high performance and power efficiency[C]// IEEE, International Conference on Application-Specific Systems, Architectures and Processors. IEEE, 2013:269-272.
- [5] Pizano-Escalante L, Parra-Michel R, Longoria-Gandara O. Fast bit-accurate reciprocal square root[J]. Microprocessors & Microsystems, 2015, 39(2):74-82.
- [6] Sajid I, Ahmed M M, Ziavras S G. Pipelined implementation of fixed point square root in FPGA using modified non-restoring algorithm[C]// The, International Conference on Computer and Automation Engineering. IEEE, 2010:226-230.
- [7] Wang D, Ercegovic M D, Zheng N. Design of High-Throughput Fixed-Point Complex Reciprocal/ Square-Root Unit[J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2010, 57(8):627-631.

作者简介

夏顺兴(1992-), 男, 硕士研究生, 从事超大规模集成电路研究, E-mail: xiasx@vlsi.zju.edu.cn;

联系地址: 浙江省杭州市浙大路38号浙江大学玉泉校区(310027)。