

一种改进双旋转 CORDIC 算法的反正弦函数 IP 设计与实现

黄宝莹¹, 常 胜¹, 黄启俊¹, 王 豪²

(1 武汉大学 物理科学与技术学院, 湖北 武汉 430072; 2 武汉大学 微电子与信息技术研究院, 湖北 武汉 430072)

摘 要: 针对传统 CORDIC 算法在计算反正余弦函数时存在的缺陷, 提出了一种带有常量校正因子和部分方向预测的改进型双旋转 CORDIC 算法, 采用 MATLAB 平台实现算法建模及定点仿真, 并在 FPGA 上完成了流水线结构的 IP 设计. 此 IP 核的速度达到 192 MHz, 平均误差小于 10^{-4} . 结果表明本算法 IP 比传统 CORDIC 算法平均误差小, 比线性补偿方法速度快, 采用流水线结构时比双旋转算法资源占用少.

关键词: CORDIC 算法; 双旋转; 常量校正因子; 反正余弦; 定点仿真

中图分类号: TP332.1

文献标识码: A

文章编号: 1000-7180(2013)12-0146-05

Design and Implementation of Arcsine/Arccosine Function Based on an Improved CORDIC Algorithm with Double Rotations

HUANG Bao-ying¹, CHANG Sheng¹, HUANG Qi-jun¹, WANG Hao²

(1 School of Physics and Technology, Wuhan University, Wuhan 430072, China;

2 Institute of Microelectronics and Information Technology, Wuhan University, Wuhan 430072, China)

Abstract: According to the defect of traditional CORDIC algorithm on calculating arcsine and arccosine function, an improved algorithm is proposed, which uses double rotations with a constant factor and partial direction prediction. The modeling and fixed-point simulation is implemented on MATLAB platform, while the hardware realization is done on FPGA with a pipeline structure. The speed of this IP core reaches 192 MHz and the average error is less than 10^{-4} . This algorithm is of better precision than traditional CORDIC on average, higher speed than linear compensation. And it costs less resource than double rotations algorithm with pipeline structure.

Key words: CORDIC algorithm; double rotations; constant scale factor; arcsine/arccosine; fixed-point simulation

1 引言

在通信、多媒体等对实时处理要求高的领域中, 大量地使用了三角、反三角、对数等超越函数, 其中反正余弦函数在数据分析、信号处理、GIS 坐标计算等领域有着重要的应用^[1-3]. 目前实现反正余弦的方法有查表法^[4]、泰勒展开法^[5]和 CORDIC 算法^[6]. 查表法需要消耗大量的存储单元; 泰勒展开由于存在大量的乘法和多次幂运算, 在硬件上不容易实现; CORDIC 算法则是把计算过程转化成移位和加法操作, 非常适合硬件实现.

然而, 传统 CORDIC 算法在计算反正余弦函数

时存在不足, 在部分区域内的计算误差较大^[7]. 一是由于这些区域内的迭代收敛过快, 二是由于存在着旋转方向的误判. 文献[7]中提出的通过分段作线性补偿方法, 只是对传统 CORDIC 算法的迭代结果进行修正, 而非对算法本身进行修改, 而且其补偿公式中乘法和除法也并不利于在 FPGA 上的高速实现. 文献[8]中采用线性补偿方法并固定补偿斜率, 精度较高但没有覆盖反正弦计算的所有范围. 文献[9]中提出的双旋转迭代 CORDIC 算法, 能有效地消除原算法在部分区域内的误差, 但由于每一步都采用双迭代, 且需要比传统算法增加一条迭代计算路径进行校正因子补偿, 资源消耗较大.

收稿日期: 2013-01-20; 修回日期: 2013-03-27

基金项目: 湖北省自然科学基金(2011CDB272)

本文针对传统 CORDIC 算法在实现反正余弦时,在某些区域中由于迭代收敛过快而导致误差大的问题作出改进,提出一种改进型的双旋转 CORDIC 算法。

2 CORDIC 算法实现反正余弦计算

2.1 算法原理

CORDIC 算法即旋转坐标数字计算器,它的提出为统一硬件实现多功能运算奠定了基础,适用于诸多领域中^[10]。利用 CORDIC 算法进行反正弦计算,采用圆周系统下的向量模式,迭代方程可以写为^[6]

$$\begin{aligned} x_{i+1} &= x_i - \sigma_i y_i 2^{-i}, \\ y_{i+1} &= \sigma_i x_i 2^{-i} + y_i, \\ z_{i+1} &= z_i - \sigma_i \arctan 2^{-i} \end{aligned} \quad (1)$$

旋转方向的判定为

$$\sigma_i = \text{sign}(c - y_i) \quad (2)$$

式中, c 为输入的正弦值。

经过 n 次迭代之后,结果为

$$\begin{aligned} x_n &= \sqrt{(Kx_0)^2 - c^2}, \\ y_n &= c, \\ z_n &= z_0 - \arcsin \frac{c}{Kx_0} \end{aligned} \quad (3)$$

若作如下初始化设置,可得到 $\arcsin(c)$ 的值:

$$x_0 = \frac{1}{K}, y_0 = 0, z_0 = 0$$

$$(K = \prod_{i=0}^n \sqrt{1 + 2^{-2i}}) \quad (4)$$

反余弦的计算有两种方法,一是与反正弦计算类似,但旋转方向的判别用 x 分量与输入参数比较,另一种是用 $\pi/2$ 减去反正弦计算的结果。

2.2 算法改进

本文提出的改进算法是传统的 CORDIC 算法与双旋转 CORDIC 算法的结合,带有常量校正因子,折中考虑了精度和资源。算法的思想是在前 t 级迭代中采用传统的单次迭代,从第 t 级开始采用双旋转迭代。由于加入了双旋转,算法的校正因子改变,会引入某些角度在前 t 级迭代时旋转方向判断错误,因此,为了消除这些错误,在算法的最开始加入了对前 t 级旋转方向进行预测的部分。算法的模型分为预测、迭代计算和后处理三个部分。

2.2.1 预测部分

与传统算法基本一致,作式(4)的初始化,从 0 开始到 $t-1$ 级作式(1)的迭代,旋转方向则改由式(5),以便把计算范围扩展至 $[-1, 1]$,把得出的 t 个

旋转方向保存:

$$\sigma_i = \begin{cases} \text{sign}(c - y_i), & x_i \geq 0 \\ -\text{sign}(c - y_i), & x_i < 0 \end{cases} \quad (5)$$

2.2.2 迭代计算部分

共分为两步,第一步作式(6)的初始化,即采用新的校正因子,从第 0 到 $t-1$ 级作式(1)的迭代;第二步从 t 级开始,采用式(7)的双旋转迭代。其中,旋转方向的前 t 个取值采用预测部分计算时保存下来的,后 $n-t+1$ 个值仍按照式(5)确定:

$$\begin{aligned} x_0 &= \frac{1}{K_1}, y_0 = 0, z_0 = 0 \\ (K_1 &= K \times \prod_{i=t}^n \sqrt{1 + 2^{-2i}}) \end{aligned} \quad (6)$$

$$\begin{aligned} x_{i+1} &= x_i - 2\sigma_i y_i 2^{-i} - \sigma_i^2 x_i 2^{-2i}, \\ y_{i+1} &= y_i + 2\sigma_i x_i 2^{-i} - \sigma_i^2 y_i 2^{-2i}, \\ z_{i+1} &= z_i - 2\sigma_i \arctan 2^{-i} \end{aligned} \quad (7)$$

2.2.3 后处理部分

由于在大角度时迭代会越过第一或第四象限,需要进行象限变换,并对象限边界处的结果作修正。另外,此改进算法同样可以用于反余弦的计算中,只要对 x 与 y 作适当的对换即可。为了在后期硬件实现时使反正弦和反余弦计算能够得到较好统一,故采用对反正弦值取余角得出反余弦值。因此后处理部分包含大角度和边界结果的修正以及反正余弦结果输出。

3 参数确定及定点仿真

3.1 参数 t 的确定

针对改进的算法,需要确定 t 参数。由于硬件实现中迭代次数是有限的,因此把 t 取值从 0 到 9 进行了仿真,并从最大绝对误差、最大相对误差、平均绝对误差和平均相对误差四个方面考虑选择 t 的最优值,在分析最大误差时考虑的是绝对值。从图 1 可见,在 $t=4$ 时,最大误差和平均误差都是最小,故取 t 的最优值为 4。

一般而言,迭代次数越多,位宽越大,计算的精度越高,但同时也要消耗更多的硬件资源,折中考虑精度和资源,选择迭代次数为 16。

图 2 所示为 16 次迭代时,本算法与传统 CORDIC 算法计算反正弦的效果比较。可见,同样迭代次数下本算法误差更小,有效地改善了传统算法的误差。

3.2 位宽选择与定点仿真

定点仿真可用于软件模拟硬件实现定点算法的

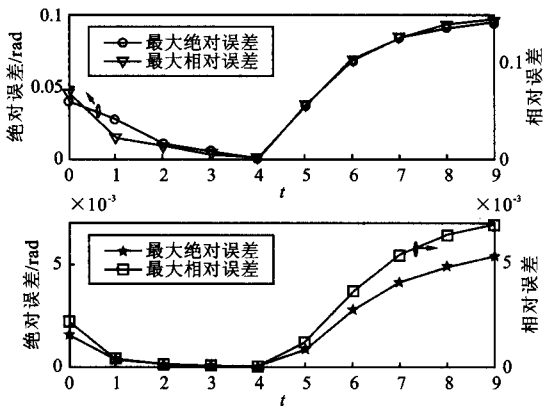


图 1 不同的 t 值与计算误差

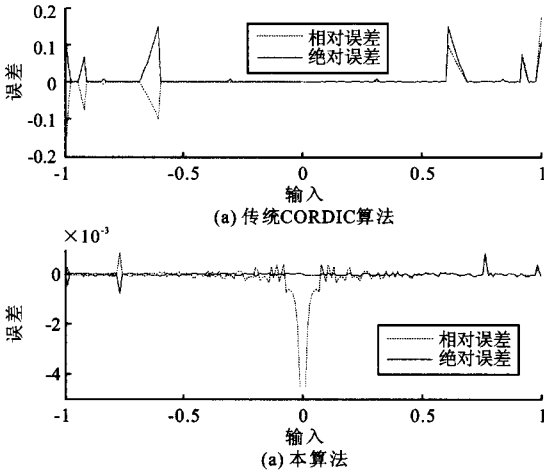


图 2 本算法与传统 CORDIC 算法结果比较

过程,它能指定数据的定点位宽,并且在软件计算时,考虑到位宽限制带来的计算误差. MATLAB 中的 Fixed-Point Toolbox 提供了定点数据类型和运算,还能结合 Simulink 共同提供一个完整的定点设计和验证环境,在 FPGA/DSP 设计中得到应用^[11-12].

本文对位宽选择的仿真就是基于 MATLAB 定点仿真来进行. 由于 x, y 取值范围为 $[-1, 1]$, 符号位和整数位各占 1 位, z 取值范围为 $[-\pi/2, \pi/2]$, 故符号位占 1 位, 整数位占 2 位, 只需要考虑小数位宽的选择即可. 在确定迭代次数为 16 时, 使用 MATLAB 的 fi 函数为迭代计算时的数据构造定点对象, 小数位宽从 14 bit 到 26 bit 进行仿真.

图 3 中所见, 小数位宽约在 19 位以后, 误差开始达到比较小的值并平稳减小, 结合考虑精度和硬件资源, 最后选择小数位宽为 20. 需要提出的是, 同样的迭代次数下, 本算法的小数位宽比传统 CORDIC 算法要大, 这是由算法性质决定的, 因为在每次迭代时, 双旋转操作由于位宽限制造成的误差比传统算法更大, 故需要更多的位宽来保证精度.

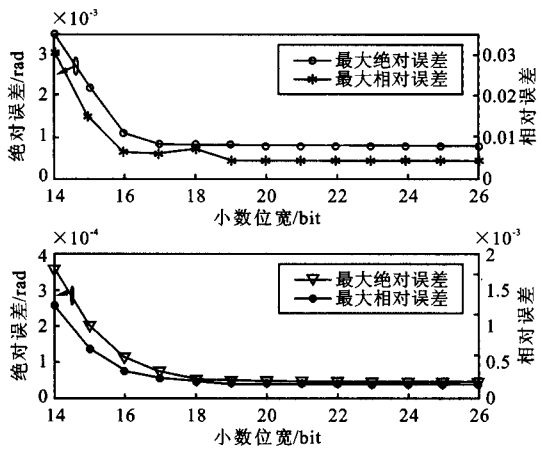


图 3 小数位宽与计算误差

为了能更加真实地预测硬件实现算法的结果, 基于确定的参数对算法进行定点仿真. 对算法作 16 次迭代, 其中 x 和 y 取 22 位定点格式, z 取 23 位定点格式. 仿真结果分析如图 4 所示, 其中的输入值由 MATLAB 软件产生并转化为已构造好的定点格式, 在误差分析时将计算结果由定点转化为浮点格式, 再与 MATLAB 的软件计算结果进行对比分析.

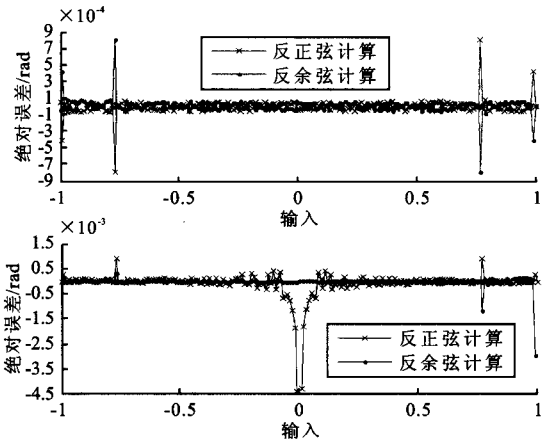


图 4 本算法的反正弦计算误差

具体的分析数据如表 1 所示, 本算法计算反正弦、反余弦的最大绝对误差和最大相对误差分别在 10^{-4} 和 10^{-3} 数量级上, 除了反正弦的平均相对误差为 1.86×10^{-4} 之外, 其他的平均误差基本在 10^{-5} 数量级, 因此精度较高.

表 1 定点仿真误差分析

函数	最大绝对 误差(rad)	最大相对 误差	平均绝对 误差(rad)	平均相对 误差
反正弦	$8.02 \text{ e-}4$	$4.40 \text{ e-}3$	$4.47 \text{ e-}5$	$1.86 \text{ e-}4$
反余弦	$8.02 \text{ e-}4$	$-2.99 \text{ e-}3$	$4.46 \text{ e-}5$	$5.06 \text{ e-}5$

此定点仿真的结果说明本算法计算反正余弦的可行性, 同时也为算法在定点格式下的硬件实现提供了一个指引, 起到了预测效果和相互印证的作用.

4 模块设计与硬件实现

4.1 模块设计

模块的端口定义如图 5 所示. 此 IP 的端口与现有商用 IP 端口基本一致, 此外, 还增加了一个可以检测输入是否在允许范围内的检错信号.

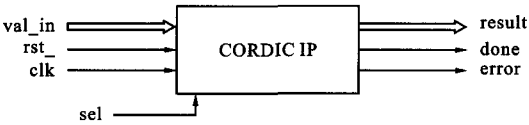


图 5 模块端口定义

4.2 流水线架构实现

流水线结构常用于对算法硬件实现进行加速. 对于 CORDIC 算法而言, 只要将所有迭代都分别展开来并行处理, 每一级的输出用寄存器保存并作为下一级输入, 就能大大地提升模块的吞吐量和运行速度. 图 6 所示为本算法的流水线架构, 采用 Verilog HDL 基于 Quartus II 9.0 进行代码设计.

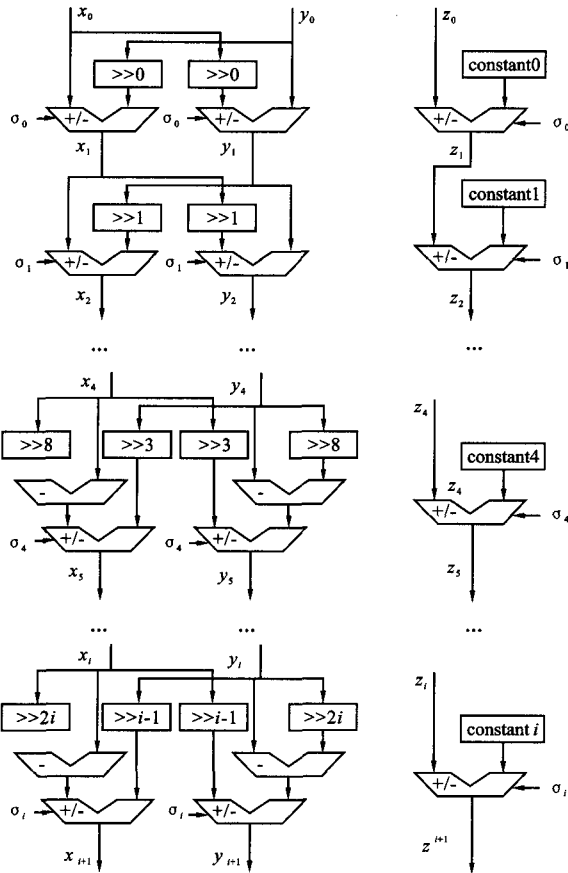


图 6 流水线架构

5 仿真结果

5.1 仿真结果与性能

硬件仿真平台采用 ModelSim SE 6. 2b, 通过

sel 设置不同的电平来选择是计算反正弦还是反余弦. 仿真时输入值是由 MATLAB 定点仿真来产生并保存成文件, 再由 ModelSim 读入并仿真. 仿真结果表明, 此流水线架构 IP 需要 22 个时钟周期输出第一个结果, 随后每隔一个周期输出一个结果.

把 ModelSim 仿真的输出数据存成文件, 与 MATLAB 定点仿真的结果进行对比. 对比结果显示两者计算结果完全相同, 同时也进一步说明了 MATLAB 定点仿真可用于定点算法的设计与硬件实现的结果预测.

硬件实现时, 选择 ALTERA 公司的高速器件 Stratix II, 综合结果显示该流水线架构的最高频率为 192. 49 MHz, 资源占用情况如图 7 所示.

Family	Stratix II
Met timing requirements	Yes
Logic utilization	21 %
Combinational ALUTs	2, 139 / 12, 480 (17%)
Dedicated logic registers	1, 660 / 12, 480 (13%)
Total registers	1660
Total pins	50 / 343 (15%)

图 7 资源占用

5.2 结果对比

将本算法的实现结果与现有的 CORDIC 算法硬件实现结果^[7]进行对比, 从误差和性能两个方面考虑. 文献^[7]的反正弦实验结果中, 误差值在各点上比较平均, 主要集中在 10^{-4} 量级, 而本算法硬件实现除了个别点的误差在 10^{-4} 数量级之外, 其余点的误差绝大部分在 10^{-5} 数量级上, 总体比文献^[7]中的误差小, 整理数据得出表 2 中的误差分析对比.

表 2 误差分析对比

函数	算法	最大绝对误差 (rad)	最大相对误差	平均绝对误差 (rad)	平均相对误差
反正弦	本算法	8.02×10^{-4}	4.40×10^{-3}	4.47×10^{-5}	1.86×10^{-4}
	文献 ^[7]	6×10^{-4}	—	$\sim 10^{-4}$	—
反余弦	本算法	8.02×10^{-4}	-2.99×10^{-3}	4.46×10^{-5}	5.06×10^{-5}

性能上的对比如表 3 所示, 可以看出, 文献^[7]的算法实现采用流水线架构, 时钟频率为 150 MHz, 但是每个结果的输出需要两个时钟周期, 相当于其模块的时钟频率只有 75 MHz, 而本算法的硬件实现现在流水线架构下最高时钟频率为 192 MHz, 每个结果输出只需要 1 个时钟周期间隔, 明显比文献^[7]的算法速度更快. 另外, 迭代次数同为 16 次, 本算法的第一个结果输出所用的时间更短, 因此, 总体上本算法的硬件实现性能比文献^[7]要好.

表 3 性能对比

算法	迭代次数	数据位宽/bit	时钟频率/MHz	第一个结果输出 时间/cycle	每个结果输出 间隔 (cycle)	器件
本算法	16	17	150	36	2	Xilinx Virtex-5 XCSVXS0t
文献[7]	16	22,23	192	22	1	Altera Stratix II EP2S15F484C3

6 结束语

本文提出了一种带常量因子和预测方向的改进型双旋转 CORDIC 算法计算反正余弦函数,有效改善了传统 CORDIC 算法在计算该函数时因迭代收敛过快引起的误差. 基于 FPGA 实现的算法 IP 核最高时钟频率为 192 MHz,平均绝对误差达到 10^{-5} ,平均相对误差达到 10^{-4} 以下. 结果表明本算法具有速度快、精度高、计算范围全面、具有检错功能等优点. 本算法仍存在不足,没有对传统算法中存在的方向误判问题进行修正,因此个别点以及大角度时的计算精度不高.

参考文献:

[1] 王晓芳, 候恩科, 张志华. 坐标方位角算法在 GIS 中的应用[J]. 测绘科学, 2011, 36(2): 238-240.

[2] 徐志军. 无损音频编码 (MPEG-4ALS) 的原理分析与仿真[D]. 南京: 东南大学, 2006.

[3] Narasmha M, Shenoi K, Peterson A. The arcsine transform and its application in signal processing[C]// IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Hartford: IEEE, 1977(2): 502-505.

[4] 王子炜, 王安印. FPGA 中反正弦函数的实现方法[J]. 大众科技, 2012, 14(150): 43-72.

[5] Roberto Barrio. Performance of the taylor series method for ODEs/DAEs [J]. Applied Mathematics and Computation, 2005, 163(2): 525-545.

[6] Ray Andraka. A survey of CORDIC algorithms[M].

New York: ACM, 1998: 191-200.

[7] 谢珊英, 齐伟民, 蔡晓宁. 基于 FPGA 的反正弦函数的实现[J]. 电子器件, 2010, 33(3): 344-347.

[8] 赵毅强, 于海霞, 高兴波, 等. 一种电磁耦合集成角度传感器的关键模块设计[J]. 北京理工大学学报, 2012, 32 (7): 735-739.

[9] Mazenc C, Merrheim X, Muller J M. Computing functions arccos and arcsin using CORDIC [J]. IEEE Transactions on Computers, 1993, 42(1): 188-122.

[10] Meher P K, Valls J, Tso-Bing Juang, et al. 50 Years of CODIC: algorithms, architectures, and applications [J]. IEEE Transactions on Circuits and Systems, 2009, 56(9): 1893-1907.

[11] Mathus A, Chang T N. A FPGA/DSP design for real-time fracture detection using low transient pulse[C]// IEEE International Symposium on Industrial Electronics (ISIE), Bari: IEEE, 2010: 2753-2758.

[12] 孙重磊, 王大庆. 基于 FPGA 的高阶 FIR 抽取滤波器有效实现结构[J]. 电子科技, 2012, 25(11): 42-44.

作者简介:

黄宝莹 女, (1988—), 学士, 硕士研究生. 研究方向为电路与系统.

常 胜 男, (1980—), 博士, 副教授. 研究方向为嵌入式系统设计与可编程逻辑器件应用.

黄启俊 男, (1965—), 博士, 教授, 博士生导师. 研究方向为半导体光电子、微电子器件和电路的模拟与仿真、微电子及 FPGA 系统设计.

王 豪 男, (1983—), 博士, 讲师. 研究方向为研究方向为集成电路设计和半导体器件模型.