

高效单精度浮点三角函数计算电路结构与实现

李天立^{1,2}, 尹 韬^{1,2}, 魏 星^{1,2}, 杨海钢^{1,2}

(1 中国科学院 电子学研究所, 北京 100190;

2 中国科学院大学, 北京 100049)

摘 要: 针对基于浮点加法器的 CORDIC(Coordinate Rotation Digital Calculation, 坐标旋转数字计算)实现单精度浮点型三角函数的角度收敛范围受限、处理速度低、电路开销大、响应延时长等问题, 通过将浮点运算转化为定点运算以及对无缩放因子 CORDIC 算法的优化, 提出一种基于查找表技术和双步迭代技术的高计算效率电路设计结构, 解决了无缩放因子 CORDIC 算法计算三角函数需要引入乘法器和迭代次数过高的问题. 在 Stratix IV (EP4SGX70DF29C2X 型 FPGA) 上实现了满足 IEEE-754 标准的单精度浮点正弦、余弦的三角函数运算. 实验结果表明该电路工作频率可达 282 MHz, 对比已有电路结构, 响应延时和电路总面积有效降低, 计算精度达到 $10E-7$.

关键词: 无缩放因子 CORDIC 算法; 单精度浮点型; 三角函数; FPGA

中图分类号: TP391.41

文献标识码: A

文章编号: 1000-7180(2018)12-0033-05

An Efficient Single-Precision Floating-Point Trigonometric Function Calculation Circuit Structure and Implementation

LI Tian-li^{1,2}, YIN Tao^{1,2}, WEI Xing^{1,2}, YANG Hai-gang^{1,2}

(1 Institute of Electrics, Chinese Academy of Sciences, Beijing 100190, China;

2 University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: There are four questions for single-precision floating point trigonometric functions by CORDIC(Coordinate Rotation Digital Calculation)algorithm based on floating point adder, including the range of angles is not enough, large area, low operating frequency and high output delay. In this paper, a high computation circuit structure based on look up table and two-step iterative technology is proposed through converting floating-point operations into fixed-point operations and the optimization of scaling free CORDIC. It solves the problem that the CORDIC algorithm needs to use the multiplier and high numbers of iteration. This circuit is implemented on Altera's Stratix IV(EP4SGX70DF29C2X FPGA)and achieves Sin and Cosine in IEEE-754 standard. The synthesis results show the clock frequency can reach 282 MHz. Comparing to existed circuit structure, output delay and area reduced effectively, the calculation accuracy can reach $10E-7$.

Key words: scaling free CORDIC; single-precision floating point; trigonometric function; FPGA

1 引言

在雷达实时成像、数字信号处理、噪声滤波、移动机器人定位与导航系统中,经常要进行大量的浮点型三角函数求值运算. 在数字电路设计中,三角函

数的实现通常使用 Cordic 算法^[1].

文献[2]采用基于浮点加法器的 Cordic 算法实现了浮点三角函数的计算,由于迭代过程需要大量使用浮点加器,资源消耗过高. 文献[3]采用基于浮点加法器的 Cordic 算法提出了实现不同精度的浮

收稿日期: 2018-01-28; 修回日期: 2018-02-22

基金项目: 国家自然科学基金(61474120, 61404140, 61704173); 北京市科技重大专项课题(Z171100000117019); 北京工商大学北京市重点实验室开放课题基金(BKBD-2017KF05)

点三角函数的电路结构,虽然对浮点加法器进行了改进,但是资源消耗仍然较高.文献[4-6]采用基于定点加法器的 Cordic 算法,将浮点型的输入角度转化为定点的输入角度,用定点加法器代替浮点加法器,有效减少了资源消耗.但是需要迭代的次数较多,电路延时大,而且当输入浮点型角度很小时,由于定点型角度数据宽是一个定值,浮点型角度转成定点型角度会造成数据始终为机器零,从而导致 Cordic 算法失效.

综上所述,相比与基于浮点加法器的 Cordic 算法,基于定点加法器的 Cordic 算法实现浮点三角函数计算,结构简单,资源消耗少,但是迭代次数较多,电路延时较大.本文将浮点三角函数的计算转化为定点三角函数的计算,并针对无缩放因子 Cordic 算法^[7]进行优化来得到定点三角函数值,函数值规格化后转换为符合 IEEE-754 标准的单精度浮点形式,利用查找表技术来消除固定的缩放因子,利用倍角公式原理实现二次迭代,有效减少迭代次数.为了提高数据吞吐率,电路结构采用全流水设计.

2 无缩放因子 Cordic 算法基本原理

无缩放因子 Cordic 算法利用三角函数 $\sin$ 和 $\cos$ 的泰勒级数展开,由文献[7]可知当 i 的取值在式(1)表示的范围时, $\sin$ 和 $\cos$ 的值可以由式(2)表示,向量单步旋转可以由式(3)表示:

$$\left[\frac{N-2.585}{3} \right] < i \leq N-1 \quad (1)$$

$$\begin{cases} \sin(\alpha_i) \approx \alpha_i = 2^{-i} \\ \cos(\alpha_i) = 1 - 2^{-(2i+1)} \end{cases} \quad (2)$$

$$\begin{cases} x_{i+1} = (1 - 2^{-(2i+1)})x_i - 2^{-i}y_i \\ y_{i+1} = 2^{-i}x_i + (1 - 2^{-(2i+1)})y_i \\ z_{i+1} = z_i - 2^{-i} \end{cases} \quad (3)$$

这种迭代表达式很容易在硬件上实现,只需要进行移位和加减操作,并消除了动态的缩放因子 K .经过 N 次迭代后, (x_N, y_N) 坐标值就是余弦和正弦三角函数值.这种方法虽然简单,但是所有的角度都需要用 $\alpha_i = 2^{-i}$ 这样的小角度一点点旋转逼近,随着角度的增大,迭代次数呈 2 的指数倍增加,面积消耗迅速增加.

3 算法优化

3.1 查找表技术和区域折叠

如图 1 所示,将浮点角度 θ 转化为定点角度 ϕ ,查找表 ROM 中存放着定点角度高 7 位对应的三角函数值,用定点角度的高 7 位作为地址线查找 ROM

中对应的函数值.这样初始旋转向量不再是与坐标轴重合的 (x_0, y_0) ,而是与目标向量 (x_ϕ, y_ϕ) 更接近的 (x_1, y_1) .

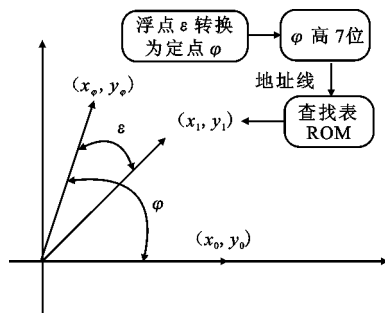


图 1 查找表技术

浮点输入角度处理范围通常为 $[-\pi, \pi]$,在向量旋转模式下,一般的 Cordic 算法收敛范围为 $[-99.9^\circ, +99.9^\circ]$,为了扩大收敛范围文献[7]将所有角度都映射到 $[0, \pi/8]$,但是会引入一个固定的缩放因子 $1/\sqrt{2}$,在最后的结果转换中需要引入乘法器,增加了算法实现的复杂性.文献[8]采用了并行波茨角度编码机制,这种方法能够减少迭代次数,但仍需要消耗较多比较器.文献[9]将所有角度都映射到 $[0, \pi/4]$,消除了固定的缩放因子,但是仍需要以小角度 2^{-i} 去迭代逼近 $\pi/4$,而且在位宽较大时(如 $N \geq 24$) 其所需的迭代次数依然较多.查找表技术的使用,使得图 1 中最大的旋转角度 ϵ 为 2^{-6} rad,不仅有效降低了迭代次数,而且使式(2)的成立更加准确,提高了计算精度.

将整个圆周 $[-\pi, \pi]$ 划分为 A, B, C, D 四个区域(如图 2 所示),即坐标轴的四个象限,采用区域折叠的策略,将所有的角度都映射到 A 区域 $[0, \pi/2]$.假设初始浮点角度 ϕ 对应的定点角度 φ 所在的区间为 $J (J \in (A, B, C, D))$,那么各个区间对应的目标向量为:

$$\begin{aligned} \begin{pmatrix} x_A \\ y_A \end{pmatrix} &= \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\varphi_1) & -\sin(\varphi_1) \\ \sin(\varphi_1) & \cos(\varphi_1) \end{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = R \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \begin{pmatrix} x_B \\ y_B \end{pmatrix} \\ &= R \begin{pmatrix} -y_1 \\ x_1 \end{pmatrix}^T, \begin{pmatrix} x_C \\ y_C \end{pmatrix} = R \begin{pmatrix} -x_1 \\ -y_1 \end{pmatrix} \begin{pmatrix} x_D \\ y_D \end{pmatrix} \\ &= R \begin{pmatrix} y_1 \\ -x_1 \end{pmatrix}^T \end{aligned} \quad (4)$$

式(4)中 (x_0, y_0) 和 φ 分别为初始旋转向量和初始定点角度, (x_1, y_1) 和 φ_1 分别为通过使用查找表技术后的旋转向量和定点角度.由上面的公式可

以看出将所有定点角度 φ 都映射到 $[0, \frac{\pi}{2}]$, 不会产生固定的缩放因子, 避免了乘法器的引入。

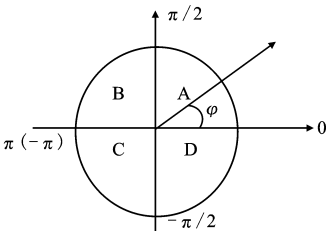


图2 区域折叠策略

输入角度的符号位 S 以及与 $\pi/2$ 比较的标志位 F 作为输出三角函数值符号的判定依据, S 和 F 位与定点角度 φ 所属区域的关系如表 1 所示。当 S 位为 0 代表输入的角度为正角, 当 S 位为 1 时代表输入角度为负角。 F 位为 0 时表示输入角度的绝对值小于 $\pi/2$, F 位为 1 时表示输入角度的绝对值大于 $\pi/2$ 。以 $R(\varphi_1)$ 和 (x_1, y_1) 为基本的旋转向量, 根据 F, S 的值来调整最终的向量坐标, 就能得到不同区域的输入角度对应的正余弦三角函数值。

表 1 定点 φ 角所在区间与 S, F 之间的关系

区间	S	F
A	0	0
B	0	1
C	1	1
D	1	0

3.2 定点角度位数的确立

根据 IEEE-754 标准, 单精度浮点型数据包含了符号位、指数位、尾数位(如表 2 所示)。相比于定点数, 浮点数具有更高的精度和更大的动态表示范围, 但是这也使得计算更复杂, 资源消耗更大。

表 2 IEEE-754 标准(单精度)

符号位	指数位	尾数位
1 bit	8 bit	23 bit

传统的 Cordic 算法需使用 $n + \log_2 n + 2$ 位数据通路^[10]才能达到 n 位精度, 本文选取的定点角度的数据位宽为 37 位, 旋转向量的初始坐标值数据位宽为 33 位。对于指数小于负的 16 次幂的小角度值, 正弦值直接以浮点角度值输出, 而余弦值直接输出为 1。通过 MATLAB 统计发现, 这种对小角度值的处理误差低于 10 的负 11 次幂, 可满足单精度浮点三角函数计算精度的要求。这种处理方式不仅在保证精度的前提下进一步降低面积消耗, 而且能有效解

决由于位宽限制, 小角度引起的 Cordic 算法失效的问题。

3.3 双步迭代算法

双步迭代法的原理是利用了二倍角公式, 将两次旋转迭代合并为 1 次旋转迭代, 以这种方式来减少迭代的次数。例如第 i 次旋转的角度为 ϕ , 接下来两次旋转的角度分别为 $\alpha_m = 2^{-m}, \beta_n = 2^{-n}$, 那么第 $i+1$ 和第 $i+2$ 次旋转后, 对应的坐标值(即正余弦值):

$$\begin{cases} y_{i+2} = \sin(\varphi + \alpha_m + \beta_n) \\ \quad = \sin\varphi\cos(\alpha_m + \beta_n) + \cos\varphi\sin(\alpha_m + \beta_n) \\ \quad = M\sin\varphi + N\cos\varphi \\ x_{i+2} = \cos(\varphi + \alpha_m + \beta_n) \\ \quad = \cos\varphi\cos(\alpha_m + \beta_n) - \sin\varphi\sin(\alpha_m + \beta_n) \\ \quad = M\cos\varphi - N\sin\varphi \end{cases} \quad (5)$$

式中, $M = 1 - 2^{-2m-1} - 2^{-2n-1} + 2^{-2m-2n-2} - 2^{-m-n}; N = 2^{-m} + 2^{-n} - 2^{-2m-n-1} - 2^{-2n-m-1}$

当 m, n 都大于 $N/2$ (N 为输入旋转坐标数据位宽) 时, $2m + 2n + 2, m + n, 2n + m + 1, 2m + n + 1, 2m + 1, 2n + 1$ 均大于 N , 因此任何数据位宽为 N 的数向右位移动如上位宽均为机器零。此时 M, N 可以简化为:

$$M = 1, N = 2^{-m} + 2^{-n}$$

简化后的 M, N 在硬件上很容易实现, 对于 m, n 都大于 $N/2$ 的旋转角度采用式(5) 进行旋转, 这样两步旋转操作可以在一次迭代中完成, 能够有效减少迭代次数和延时。

4 硬件结构

实现浮点三角函数运算的硬件电路结构主要包含浮点转定点, 迭代和定点转浮点三个模块。电路的数据结构如图 3 所示。

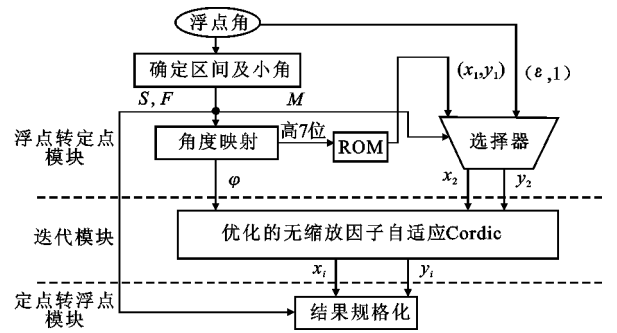


图3 电路的数据结构

浮点转定点模块的功能是将浮点角度 θ 转换为定点角度 ϕ 并映射到 A 区域, 产生控制信号 S, F, M 。其中 S, F 信号标志位是为了调整最终的输出向

量 (x_i, y_i) , 当 (S, F) 为 $(0, 0)$ 时, (x_i, y_i) 坐标不变, 当 (S, F) 为 $(0, 1)$ 时, (x_i, y_i) 坐标变为 $(-y_i, x_i)$, 当 (S, F) 为 $(1, 1)$ 时, (x_i, y_i) 坐标变为 $(-x_i, -y_i)$, 当 (S, F) 为 $(1, 0)$ 时, (x_i, y_i) 坐标变为 $(y_i, -x_i)$. M 为小角度控制信号, 当输入浮点角度的指数位小于负的 16 次幂时, 输入迭代模块的初始向量 (x_2, y_2) 为 $(\theta, 1)$, 输入角 θ 为 0, 大于负的 16 次幂则迭代模块的初始向量由 ROM 表决定. 定点转浮点模块主要完成定点的函数值向 IEEE-754 规定的浮点数据形式的转换.

迭代模块是整个模块的核心, 为了提高数据吞吐率, 如图 4 所示, 该模块采用全流水线设计, 对输入的定点角度采用单比特流水线的方式, 每一个比特位决定下一次的旋转方向, 使得每次旋转都朝向目标角度逼近, 比特位为 1 则进行该次旋转, 比特位为 0 则该次旋转被旁路掉, 这种自适应的旋转模式能够有效减少延时.

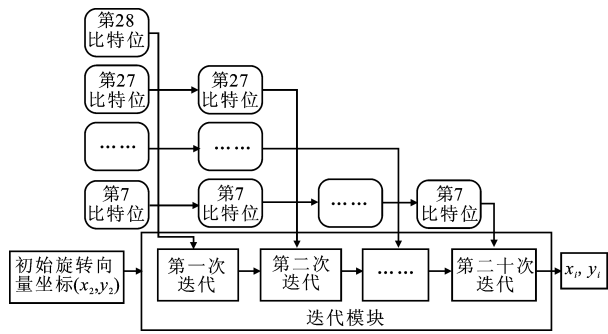


图 4 旋转模块流水线结构

5 性能分析

5.1 资源消耗

如表 3、表 4 所示, 本文在 EP4SGX70DF29C2X 型 FPGA 实现了提出的结构, 模块采用 Verilog 硬件语言进行描述, 并利用 Quartus 13.0 进行综合. 综合的结果与 Altera 公司^[11] 设计的同样功能的软核综合的结果进行对比.

表 3 资源对比情况		
设计方法	输出延时 / 周期个数	最大工作频率 / MHz
Altera 公司设计	36	258
本文设计	24	282

表 4 资源对比情况	
设计方法	逻辑资源使用情况

	M9K	18-Bit DSP	LUT	Register
Altera 公司设计	1	16	2 808	2 190
本文设计	1	0	5 516	5 114
本文设计	1	0	2 810	1 993

与 Altera 公司设计的 IP 核相比, 本文提出的结构只需要 24 个周期就能得到计算结果, 而 Altera 公司设计的 IP 核则需要 36 个周期, 输出延时减少了 33%, 增强了系统的实时性. 如果两种 IP 核在都不占用片上 DSP 资源的情况下, 本文设计的 IP 核需要 LUT 表的个数为 2 810, 寄存器的个数为 1 993, 而 Altera 公司则需要 LUT 表的个数为 5 516, 寄存器的个数为 5 114, 比较之下本文提出的结构面积消耗能节省近 50%. 在占用片上 DSP 资源的情况下, Altera 公司的 IP 核使用了较多的乘法器, 需要 16 个 18 Bit DSP 模块, 本文提出的算法只需要进行简单的移位和加减运算就能实现, 不使用 DSP 模块也能获得更高的速度, 有效降低了结构的复杂性, 面积减小近 16%.

相比于文献[2]只采取了 10 级流水线, 需要消耗的 LUT 个数达到 8 428, 且计算精度只能达到小数点后 2 位, 本文采取 20 级流水线, 只需要 2 810 个 LUT, 计算精度可达小数点后 7 位. 相比于文献[3], 需要消耗 LUT 表的个数为 3 976, 是本文提出电路结构的 1.4 倍, 最高工作频率为 45 MHz, 是本文提出结构的 1/6.

5.2 精度分析

为了验证本文设计结构的精度, 我们以 0.000 15 弧度为步长, 在 $[0, \pi]$ 的区间取 20 000 个角度, 利用 Modelsim SE10.0a 进行仿真, 算出来的结果利用 MATLAB 软件与标准值进行对比, 统计相对误差如图 5 所示, 可以看出, 使用本文提出的结构, 计算结果误差低于 10E-7, 相比于文献[3]精度提高 10 倍.

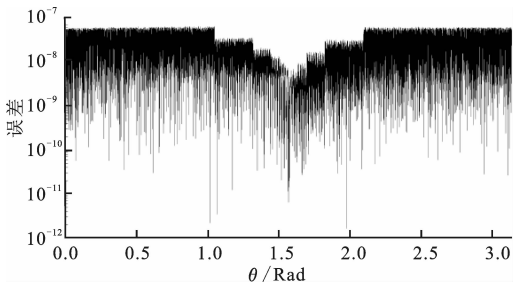


图 5 正余弦值相对误差

6 结束语

本文采用高精度定点加法器代替浮点加法器,通过查找表技术和区域折叠策略扩展了无缩放因子 Cordic 算法角度收敛范围并消除了缩放因子,利用双步迭代技术压缩了迭代次数,流水线的设计提高了数据吞吐率.通过这些方法,我们实现了单精度浮点三角函数的计算,电路面积小,角度收敛范围覆盖整个周期.所提出的设计结构可解决浮点运算面积消耗大、无缩放因子 Cordic 算法收敛域小,存在固定缩放因子等缺点.电路结构在 FPGA 平台上进行了实现和验证,与已知电路对比,在面积、速度、延时上都有较明显的优势.此外,通过 MATLAB 工具对三角函数的相对误差进行统计分析,计算精度可达 $10E-7$.

参考文献:

- [1] Voldet. The cordic trigonometric computing technique. IRETrans [J]. Electronic Computers, 1959, EC-8(3): 334-334 .
- [2] 吴庆达,何书专,潘红兵,等. 32 位定浮点数正余弦函数 FPGA 实现方法[J]. 微电子学与计算机, 2012, 29(1):113-116.
- [3] Arun Kumar Singh, Madhav Kumar Singh, Kailash Chandra Ray. Design and implementation of quadruple floating-point CORDIC[C]//IEEE International Symposium on Nanoelectronic and Information Systems, 2015 IEEE. Indore, India, IEEE ,2015:286-290.
- [4] Chen Dong, Chen He, Sun Xing, et al. Implementation of single-precision floating-point trigonometric functions with small area[C]// International Conference on Control Engineering and Communication Technology. Washington DC :IEEE, 2012:589-592.
- [5] Lei Y, Zhu B, Peng Y, et al. Low overhead iterative trigonometric function device has pretreatment module that judges signal output of trigonometric function and convert output from fixed-point to floating-point for-

mat:China, CN106155627-A[P]. 2016-11-23 .

- [6] Peng Y, Zhu B, Lei Y, et al. CORDIC and Taylor algorithm based combined full pipeline floating-point trigonometric function device, has post-processing module for calculating cordic algorithm processing result and outputting sine and cosine of calculation result: China, CN106202890-A[P]. 2016-12-07.
- [7] Koushik Maharatna, Swapna Banerjee, Senior Member. Modified virtually scaling-free adaptive cordic rotator algorithm and architecture[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(11):1463-1474.
- [8] Rodrigues T K, Swartzlander E. Adaptive cordic; using parallel angle recoding to accelerate rotations [J]. IEEE Transactions on Computers, 2010, 59(4): 522 - 531.
- [9] 徐成,秦云川,李肯立,等. 免缩放因子双步旋转 CORDIC 算法[J]. 电子学报, 2014, 42(7):1441-1445.
- [10] Kishore Kota Joseph. Numerical accuracy and hardware tradeoffs for cordic arithmetic for special-purpose processors[J]. IEEE Transactions on Computers, 1993, 42(7):769-77.
- [11] Altera. Floating-point ip cores user guide[EB/OL]. [2018-01-05]. <http://www.altera.com>.

作者简介:

李天立 男, (1991-), 硕士研究生. 研究方向为超大规模数字集成电路.

尹 韬(通讯作者) 男, (1980-), 博士, 副研究员, 硕士生导师. 研究方向为数模混合芯片设计、MEMS 传感器高性能接口芯片设计、高效信号处理芯片设计.

E-mail: yint@mail. ie. ac. cn.

魏 星 男, (1991-), 博士研究生. 研究方向为算法硬件加速与高性能芯片架构开发.

杨海钢 男, (1960-), 博士, 研究员, 教授, 博士生导师. 研究方向为高性能超大规模集成电路芯片设计技术、高性能模拟和混合信号 SOC(片上系统)设计技术.