

一种改进的 CORDIC 算法及其 FPGA 实现

段文伟, 于龙洋, 李署坚

(北京航空航天大学 电子信息工程学院, 北京 100191)

摘 要: CORDIC 算法在很多工程应用中发挥着很大的作用. 在传统的 CORDIC 算法基础上, 通过增加内部相位累加器的位数, 以及采用两位的方向控制因子的方法, 在不影响实现复杂度的基础上可以产生更高相位分辨率的正余弦信号. 采用 QUARTUS II 进行综合仿真, 验证了设计的可行性, 并用 MATLAB 分析了设计的相位分辨率和计算误差等, 证明设计可以有效地提高 CORDIC 算法的性能.

关键词: CORDIC 算法; 现场可编程门阵列; MATLAB; 正余弦

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2012)02-0095-04

A Kind of Improved CORDIC Algorithm and Its Implementation in FPGA

DUAN Wen-wei, YU Long-yang, LI Shu-jian

(School of Electronic and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract: CORDIC is an efficient algorithm that plays an important part in various engineering applications. Method based on traditional CORDIC algorithm is introduced. By increasing the number of bits of inner phase accumulator and using two bits to represent directional control factor, a sine and cosine generator is designed based on FPGA. The result of QUARTUS II shows that the design is feasible. Finally, the phase resolution and error performance is analysed by using MATLAB, the results demonstrate that this optimization design can improve the performance of CORDIC effectively.

Key words: CORDIC algorithm; FPGA; MATLAB; sine cosine

1 引言

坐标旋转数字计算机(Coordinate rotation digital computer CORDIC)由 J. Voder 于 1959 年提出, 已经在很多基本运算中发挥着很大的作用, 其中包括产生 sine 和 cosine 信号. 它通过多次迭代的方法, 来实现相应的运算. 由于 CORDIC 算法只需要一定数量的加法器和移位寄存器, 因此在高速率运算中优势更加明显. 它在直接数字频率合成(DDS)、正交频分复用(OFDM)、快速傅里叶变换(FFT)等方面都有很好的应用. 文献[1]中采用的 Double rotation CORDIC 算法可以减少迭代次数, 文献[2]中采用的 MVR(Modified Vector Rotational

CORDIC)算法可以在减少迭代次数的同时提高相位分辨率, 但是两种算法都在很大程度上增加了硬件资源的消耗和实现的复杂度. 本文将传统 CORDIC 算法稍加改进, 在不增加实现复杂度的基础上获得更高的相位分辨率, MATLAB 和 QUARTUS II 仿真结果说明改进算法后的相位分辨率有了显著提高.

2 CORDIC 基本原理

CORDIC 算法的基本思想就是从起点 (x_i, y_i) 不断进行矢量旋转, 逐步逼近最终点 (x_j, y_j) , 最后所转过的角度 θ 就是所要计算的 $\arctan(x_j/y_j)$ 值. 算法图解如图 1 所示.

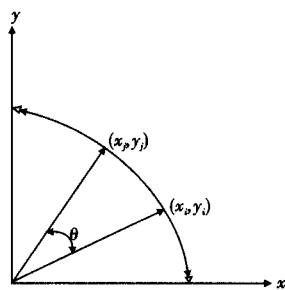


图1 CORDIC 算法图解

根据极坐标知识很容易得到

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (1)$$

从起始点到最终点的旋转过程可以通过多步来实现,每步只旋转一个小角度,则有

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n \\ \sin\theta_n & \cos\theta_n \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (2)$$

式(2)也可写成

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -\tan\theta_n \\ \tan\theta_n & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (3)$$

为了简化运算,把每次旋转的角度设定为

$$\theta_n = \arctan\left(\frac{1}{2^n}\right) \quad (4)$$

并且需要保证总的旋转角度为

$$\sum_{n=0}^N d_i \theta_n = \theta \quad (5)$$

式中, $d_i = \pm 1$, 取决于正时针旋转还是逆时针旋转. 这样就有: $\tan\theta = d_i 2^{-n}$, 代入式(3)可得

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -d_i 2^{-n} \\ d_i 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (6)$$

在上式中除了 $\cos\theta_n$ 系数外, 整个运算只有简单的右移和加法. 而 $\cos\theta_n$ 实际上是通过在最终结果上乘以一个已知的常数 K 来加以消除, 这是因为当 N 很大时有

$$\begin{aligned} K &= \prod_{n=0}^N \cos\theta_n \\ &= \prod_{n=0}^N \cos(\arctan(2^{-n})) \\ &= \prod_{n=0}^N \frac{1}{\sqrt{1+2^{-2n}}} \approx 0.607\,253 \end{aligned} \quad (7)$$

所以, 在中间旋转过程 $\cos\theta_n$ 项可以不予考虑, 最后得到旋转迭代公式:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -d_i 2^{-n} \\ d_i 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (8)$$

$$\text{或者: } \begin{cases} x_{n+1} = x_n - d_n 2^{-n} y_n \\ y_{n+1} = y_n + d_n 2^{-n} x_n \end{cases} \quad (9)$$

再引入第三个方程, 称为角度累加器, 用来在每次迭代过程中追踪累加的旋转角度:

$$z_{n+1} = z_n - d_n \theta_n \quad (10)$$

在 CORDIC 旋转模式下的旋转方向控制序列 $\{d_n = \pm 1; n=0, 1, 2, \dots\}$ 由下述旋转方式决定. 设 θ 为正角, 即 $\theta \in (0, \theta_{\max})$, 每次旋转后终止边与 x 轴的夹角称为当前旋转角. 从 x 轴开始, 先连续逆时针(正向)旋转 $\theta_0, \theta_1, \dots, \theta_{k_1}$, 直到旋转首次大于 θ 角, 与这些角相应的方向控制因子为 $d_n = +1, n=0, 1, 2, \dots, k_1$; 接着连续顺时针(反向)旋转 $\theta_{k_1+1}, \theta_{k_1+2}, \dots, \theta_{k_2}$, 直到旋转角再次小于 θ 角, 与这些角相应的方向控制因子为 $d_n = -1, n=k_1+1, k_1+2, \dots, k_2$; 然后再改为正向旋转; 如此反复, 直到旋转向量与 θ 角充分接近. 这样角度累加器 $z_{n+1} = z_n - d_n \theta_n$. 其中方向控制因子为 $d_n = \text{sign}(z_n)$.

3 改进算法产生正余弦信号

为了便于计算本文选取初始值分别为: $x_0 = 1, y_0 = 0, z_0 = \theta$, 这样经过 N 次迭代之后, 输出结果为 $x_n = (\cos z_0)/K, y_n = (\sin z_0)/K$. 为了简单起见, 在满足要求的情况下, 直接乘以伸缩因子 K 即可得到要求角度的 sine 和 cosine 值.

首先, 为了增加输出信号的频带范围, 需要减少计算周期, 所以我们将系统设计成流水线结构, 避免了因为多次迭代导致产生信号周期长的问题, 在一个时钟周期就可以产生有效信号, 而且使系统有很好的扩展性, 如果要提高精度, 只需要简单的增加流水单元即可.

其次, 因为在每一级相位累加器中都会产生截尾误差, 而且由于后级是在前级的结果上进行运算, 所以各级误差会不断累积, 因此对于一定位数的输入相位, 为了使总的累积误差在输入相位所能表示的最小范围之内, 要求每一级的截尾误差都要满足“小于等于最小范围除以迭代次数”的关系, 我们根据迭代次数将内部相位累加器增加一定的位数, 以提高运算精度.

第三, 不同于传统 CORDIC 算法中用一位方向控制因子决定下级流水操作的加减, 我们采用两位方向控制因子 d_n , 设置停滞状态, 当某级相位累加

器的结果小于最小旋转角度的一半时,将 d_n 设定为“01”状态,则下一级流水只保存上一级的结果,而不进行任何操作。

d_n 与相关运算的映射关系如表 1 所示。

表 1 方向控制因子与相关运算映射表

方向控制因子 $d_n[1..0]$	运算
11	$\begin{cases} x_{n+1}=x_n-2^{-n}y_n \\ y_{n+1}=y_n+2^{-n}x_n \end{cases}$
00	$\begin{cases} x_{n+1}=x_n+2^{-n}y_n \\ y_{n+1}=y_n-2^{-n}x_n \end{cases}$
01	$\begin{cases} x_{n+1}=x_n \\ y_{n+1}=y_n \end{cases}$
10	$\begin{cases} x_{n+1}=x_n \\ y_{n+1}=y_n \end{cases}$

第四,对文献[2]中的 MVR 算法进行分析后,发现它虽然有很高的相位分辨率,但是由于每次转过的角度的未定,而且同样的角度转的次数也不同,所以要在每级使用乘法器或者在最后用很大的查找表来解决伸缩因子的问题,所耗费资源较多.本文在借鉴 MVR 算法的一些优点的同时,使转过的角度依然按照式(10)依次累加,这样转过的角度不会出现很大的跳变,而且角度转的次数都只有一次,所以只需要在流水线的最后一级根据实际迭代的次数,乘以相应的伸缩因子 K 就可以得到正确的结果,且可以满足系统对相位分辨率的要求。

本文采用硬件编程语言 VHDL 在 QUARTUS II 中设计生成的顶层文件如图 2 所示。

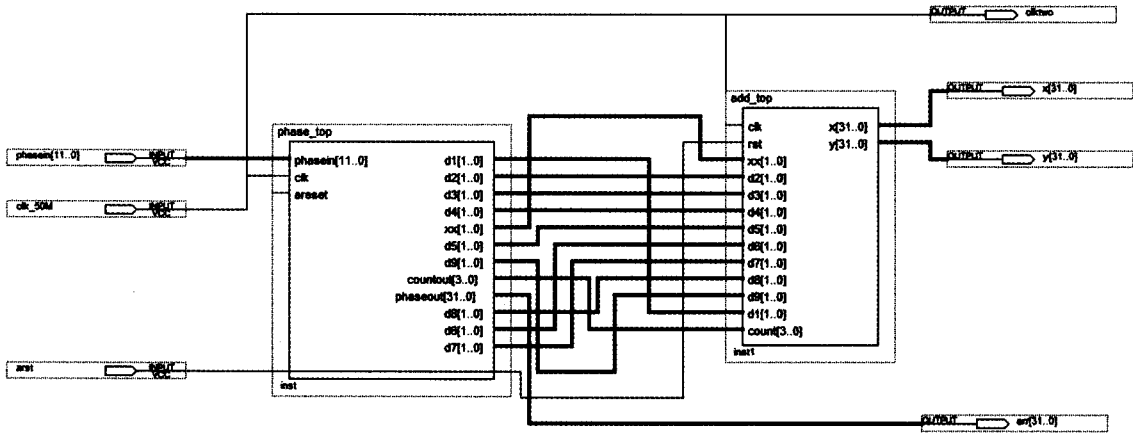


图 2 顶层设计

其中 phase_top 模块实现相位累加,add_top 模块实现坐标 x,y 的迭代运算,都采用流水线结构.在具体的流水线实现时将坐标位数和移位个数全部设为可变,以便实现模块化设计.首先根据输入的相位值判定所处象限,统一转化到第一象限进行后级运算,象限映射关系如表 2 所示。

表 2 象限映射表

输入相位高两位	所处象限	sine	cosine
00	I	y	x
01	II	y	x 补
10	III	y 补	x 补
11	IV	y 补	x

12 位,内部扩展为 17 位,旋转坐标 x,y 为 32 位,采用 10 级流水,进行 10 次迭代,经验证可以很好地满足系统要求,仿真波形如图 3 所示。

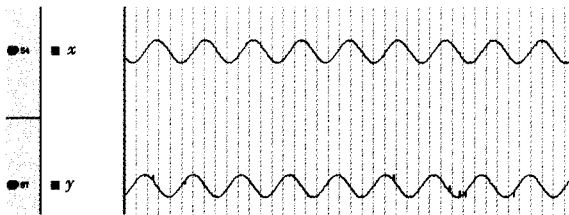


图 3 仿真结果

4 性能分析

用 MATLAB 对设计的相位分辨率等性能进行分析。

首先,通过将内部相位累加器位数增加 4 位的方法可以将相位的微小变化扩大,增加运算精度.有效的提高系统性能.如图 4 所示。

然后将转化到第一象限的相位扩展一定位数,进行相位累加,同时每一级相位累加流水线产生方向控制因子 d_n 传给相应的坐标运算流水线,进行坐标累加, N 次迭代之后将 x,y 乘以相应的伸缩因子 K 输出,即为输入相位对应的正弦和余弦值.本文设计系统时钟为 100 MHz,相位累加器初始输入为

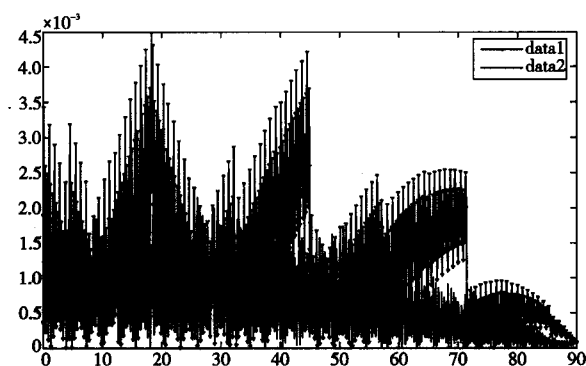


图4 旋转角度统计截尾误差造成的精度下降分析曲线

图中灰色无标记实线 data2 表示用比输入相位的二进制表示多四位的二进制值来表示每一次迭代的旋转角度后计算得到的正弦函数值与标准正弦函数的值绝对误差,而 data1 曲线表示用与表示相位相同位数的二进制值表示旋转角度时得到的函数值与标准正弦函数值绝对误差.由图 4 可知,扩展 4 位后,运算误差显著减少,有利于提高系统性能.

其次,传统 CORDIC 迭代算法能够覆盖的角度为 $(-99.9^\circ \sim 99.9^\circ)$,统一转化到第一象限后也不能够达到实际应用中角度覆盖 $(-180^\circ \sim 180^\circ)$ 的要求.而且迭代次数有限,所以传统算法的相位分辨率比较低.而在本设计中,相位累加时不仅判定方向控制因子 d_n 的正负,而且设置停滞状态,处于该状态时后级相位累加和坐标迭代不再进行运算,而是保存当前值直至输出.具体实现映射如表 1 所示.

采用这种迭代算法之后不仅可以增加可计算的相位值,而且可以很好地提高迭代精度.如图 5、图 6 所示.

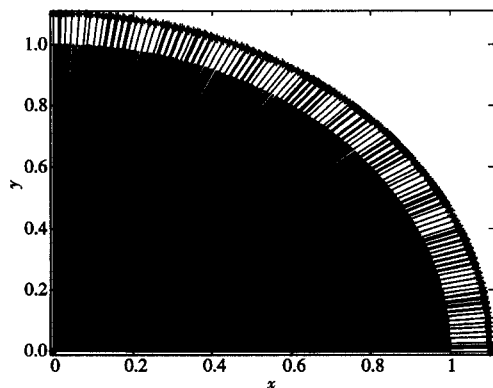


图5 采用优化算法前后的迭代结果比较

图 5 中,半径为 1.1,标记有星号的直线表示采用该算法后比一般迭代算法增加的可计算的相位

值.由图可见,通过增加停滞状态,可以计算得到更多的 sine 和 cosine 值,有效地增加相位分辨率.由图 6 可见,采用优化算法之后,迭代误差得到有效的改善.表 3 所示为不同迭代次数的情况下,改进算法前后可以覆盖的相位数目.

由表 3 看到采用改进算法后相位分辨率约为传统算法的两倍,说明该算法在不增加硬件复杂度的情况下,提高相位分辨率的效果比较明显.

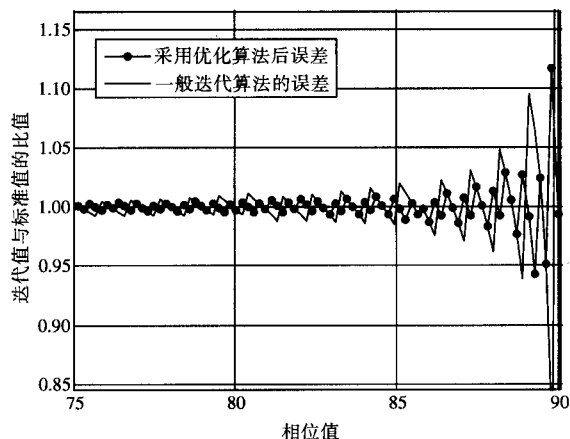


图6 采用优化算法前后误差比较

表3 改进算法前后的可取的相位数目

迭代次数	传统算法可取的 相位数目	改进算法可取的 相位数目
10	468	935
11	936	1871
12	1 871	3 742
13	3 742	7 484
14	7 485	14 969

5 结束语

本文通过增加内部相位的处理位数以及改进的迭代机制,有效地提高了 CORDIC 算法的相位分辨率,降低了计算误差,尤其是在 sine 和 cosine 函数导数较大的地方改善效果尤为明显.该设计在 NCO,FFT 等应用中可以发挥很好的作用.

参考文献:

- [1] Sung T Y, Hsin H C. Design and simulation of reusable IP CORDIC core for special-purpose processors [J]. IET Comput Digit Tech, 2007, 1(5), 581-589.

(下转第 102 页)

4 算法时间复杂度分析

假设使用大小为 block 的数据块 S 上构建一个基础分类器的时间复杂度为 $O(f(\text{block}))$. 令计算单分类器权值的时间复杂度为 $O(CW(i))$, 则系统分类器整体权值计算复杂度为 $O(\sum_{i=1}^n CW(i))$, 加权投票的时间复杂度为 $O(1)$ 与以上相比可忽略不计. 传统的分类器系统算法串行的在 n 个数据块上构建 n 个基础分类器并计算分类器权值, 故整体时间复杂度为 $O(n * f(\text{block})) + O(\sum_{i=1}^n CW(i)) = O(f(s) + \sum_{i=1}^n CW(i))$. 而基于 MapReduce 的云计算系统分类算法提供了并行构建基分类器的机制, 故其整体时间复杂度仅为 $O(f(\text{block}) + \frac{1}{n} \sum_{i=1}^n CW(i) + T(n))$, 其中 $T(n)$ 为云计算系统的通讯开销. 当分布式结点的通讯开销 $T(n) \ll f(\text{block}) + \frac{1}{n} \sum_{i=1}^n CW(i)$ 时, 与传统系统分类算法相比, 云计算平台上的系统分类算法时间复杂度大大降低, 提高了算法的执行效率. 反之, 说明当云计算系统中的结点个数为 n 时, 通信开销对算法效率的影响很大, 算法在这个特定云环境下不具有可扩展性. 但仅升级云计算平台的软硬件配置, 并不能解决云计算自身的弊端, 这也是未来需进一步完善的方面.

5 结束语

本文提出了一种在云计算环境下以 MapRe-

duce 为基本编程模型的数据流系统分类算法. 此算法将传统基于动态权重的数据流系统分类算法进行 MapReduce 并行化改造, 并应用到 Hadoop 云计算平台来提升算法处理大规模快速数据流分类问题的能力. 通过对算法复杂度进行理论分析可知, 算法的执行效率大大超过了串行算法. 但是我们同样注意到由于云计算环境受网络带宽、通信开销等诸多因素影响, 如何优化现有 MapReduce 编程模式来降低云平台固有的缺陷仍然有待进一步的研究.

参考文献:

- [1] 程转流, 胡为成. 基于 DSFCI-tree 的分布式数据流频繁闭合模式挖掘 [J]. 微电子学与计算机, 2007, 24(9): 120-125.
- [2] 刘鹏. 云计算 [M]. 北京: 电子工业出版社, 2010: 1-4.
- [3] White T. Hadoop, the definitive guide [M]. O'Reilly Media, Inc, 2009.
- [4] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. [C]//Proc of the 6th Symposium on Operating Systems Design and Implementation. San Francisco: Google Inc, 2004.
- [5] Wang H, Fan W, Yu P S, et al. Mining concept-drifting data streams using ensemble classifiers. [C] Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington D C, USA: ACM, 2003: 226-235.

作者简介:

钱琳 男, (1984—), 硕士. 研究方向为数据挖掘与知识发现.

(上接第 98 页)

- [2] Wu Cheng-Shing, Wu An-Yeu. Modified vector rotational CORDIC (MVR-CORDIC) algorithm and architecture [J]. IEEE Transactions on Circuits and Systems- II: Analog and Digital Signal Processing, 2001, 48(6): 548-561.
- [3] 张尊, 李绍荣. 基于改进的 CORDIC 算法的 NCO 实现 [J]. 信息通信, 2009(4): 11-13.
- [4] 杨小牛, 楼才义, 徐建良. 软件无线电技术与应用 [M]. 北京: 北京理工大学出版社, 2010.
- [5] 何宾. FPGA 数字信号处理实现原理及方法 [M]. 北京: 清华大学出版社, 2010.

- [6] 施国勇. 数字信号处理 FPGA 电路设计 [M]. 北京: 高等教育出版社, 2010.

作者简介:

段文伟 女, (1987—), 硕士研究生. 研究方向为扩频通信、卫星导航.

于龙洋 男, (1989—), 硕士研究生. 研究方向为扩频通信、卫星导航.

李署坚 男, (1952—), 高级工程师, 硕士生导师. 研究方向为扩频通信.