

文章编号: 1672-2892(2010)02-0211-04

## 基于CORDIC算法的双曲正余弦函数FPGA实现

周晓青, 李合生, 陶荣辉, 蔡英武

(中国工程物理研究院 电子工程研究所, 四川 绵阳 621900)

**摘要:** 作为一种快速精确进行超越函数运算的方法, 坐标旋转数字计算(CORDIC)算法在现代工程实践中获得了广泛应用。本文简要介绍了该算法的基本原理, 给出了具体的计算方法, 并以双曲正余弦函数的求取为例, 给出了CORDIC算法在FPGA中的实现方法, 在集成综合环境(ISE)平台上进行了仿真。结果表明, 由于采用了流水线结构, 算法精度较高, 实时性较好。另外, 通过Matlab相应的算法进行了仿真, 得出迭代次数和计算误差之间的关系曲线, 有助于实际应用中选择迭代次数。

**关键词:** CORDIC算法; 双曲正弦函数; 双曲余弦函数; 现场可编程门阵列; 流水线结构  
**中图分类号:** TN74 **文献标识码:** A

## FPGA implementation of hyperbolic function based on CORDIC algorithm

ZHOU Xiao-qing, LI He-sheng, TAO Rong-hui, CAI Ying-wu

(Institute of Electronic Engineering, China Academy of Engineering Physics, Mianyang Sichuan 621900, China)

**Abstract:** The application of CORDIC(COordinate Rotation DIgital Computing) arithmetic in solving transcendental function has become popular in modern engineering. A brief introduction of CORDIC arithmetic was given in this study. Taking the hyperbolic sine and cosine functions as examples, the method of realizing the arithmetic in FPGA was presented and simulated on ISE(Integrated Software Environment) platform. The result showed this arithmetic was of good precision and instantaneity because of using pipeline. This arithmetic has great practical value in some applications. In addition, a curve of the relationship between the iterative number and the error obtained by the Matlab simulation of the arithmetic can help designers planning the number of iterations for practical applications.

**Key words:** COordinate Rotation DIgital Computing arithmetic; hyperbolic sine function; hyperbolic cosine function; FPGA; pipeline

现代数字信号处理的算法结构日趋复杂, 计算量大, 实时性要求高, 且包含大量三角函数、开方等复杂函数计算。处理这些函数的传统方法是使用查表法或多项式展开近似, 但速度和精度难以满足实用要求, 硬件实现也非常困难。近年来, CORDIC方法, 由于其良好的计算精度和实时性而得到迅速发展。双曲正余弦函数在实时语音信号处理、图像信号处理、FFT变换、检波等工程中应用非常广泛。本文基于CORDIC算法, 研究双曲正余弦函数的FPGA实现方法。

### 1 CORDIC 算法原理

CORDIC算法<sup>[1]</sup>是由J Volder等人于1959年在设计美国航空导航控制系统过程中提出的一种算法, 是计算在两个平面直角坐标系 $(x, y)$ 和极坐标系 $(R, \theta)$ 之间进行坐标变换的算法。J Walter于20世纪70年代初推广了CORDIC算法<sup>[2]</sup>, 将圆周、线性和双曲线变换都包括了进来。本文将主要介绍利用CORDIC算法求解反正切函数。如图1, 向量 $V_1$ 旋转角度 $\theta$ 后得到新的向量 $V_2$ , 根据简单的几何关系有:

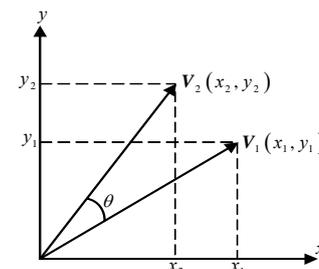


Fig.1 Schematic of CORDIC algorithm  
图1 CORDIC算法原理示意图

收稿日期: 2009-10-18; 修回日期: 2009-12-15

基金项目: 中国工程物理研究院重大预研资助项目(重大预研-05)

$$x_2 = x_1 \cos \theta - y_1 \sin \theta \quad (1)$$

$$y_2 = y_1 \cos \theta + x_1 \sin \theta \quad (2)$$

式(1)和式(2)整理后, 可得:

$$x_2 = (x_1 - y_1 \tan \theta) \cos \theta \quad (3)$$

$$y_2 = (y_1 + x_1 \tan \theta) \cos \theta \quad (4)$$

由上面的分析可知, 假设向量经过  $i$  次微小的旋转, 每次旋转的角度为  $\theta_i$ , 旋转的方向为  $d_i$  (取值为 1 或 -1), 经过  $n$  次旋转后, 总的旋转角度为:  $\varphi = \sum_{i=1}^n d_i \theta_i$ 。式(3)和式(4)可以概括为下式<sup>[3]</sup>:

$$\begin{bmatrix} x(n) \\ y(n) \end{bmatrix} = \prod_{i=1}^n \cos(m^{1/2} \theta_i) \begin{bmatrix} 1 & -m^{1/2} d_i \tan(m^{1/2} \theta_i) \\ m^{1/2} d_i \tan(m^{1/2} \theta_i) & 1 \end{bmatrix} \begin{bmatrix} x(0) \\ y(0) \end{bmatrix} \quad (5)$$

$$z(n) = z(0) + \sum_{i=1}^n d_i \theta_i \quad (6)$$

根据  $m=1, -1$  或  $0$ , 分别称为圆周旋转运算、双曲旋转运算或线性旋转运算。 $d_i$  代表旋转的方向:  $d_i=1$ , 表示逆时针旋转;  $d_i=-1$ , 表示顺时针旋转, 具体取值见式(7)。 $\theta_i$  代表旋转的角度, 由于在数字电路中多用二进制表示数值, 所以  $\theta_i$  的取值多为 2 的幂指数, 如式(8)所示<sup>[4]</sup>。

$$d_i = \begin{cases} \text{sign}(Z_i) & \text{rotation mode} \\ -\text{sign}(Y_i) & \text{vector mode} \end{cases} \quad (7)$$

$$\theta_i = \begin{cases} \arctan(2^{-i}) & m=1 \\ 2^{-i} & m=0 \\ \text{artanh}(-2^{-i}) & m=-1 \end{cases} \quad (8)$$

此时, 式(5)和式(6)可简化为如下迭代关系:

$$\begin{cases} X_{i+1} = X_i - m d_i 2^{-i} Y_i \\ Y_{i+1} = Y_i + d_i 2^{-i} X_i \\ Z_{i+1} = Z_i - d_i \theta_i \end{cases} \quad (9)$$

使结果  $Z_n \rightarrow 0$  的旋转称为旋转模式(rotation mode), 使结果  $Y_n \rightarrow 0$  的旋转称为向量模式(vector mode)。 $K$  为校正系数, 用来校正由于旋转所带来的系数变化。

$$K = \begin{cases} \prod_{i=0}^{n-1} (1 + 2^{-2i})^{-1/2} & m=1 \\ 1 & m=0 \\ \prod_{i=1}^{n-1} (1 - 2^{-2i})^{-1/2} & m=-1 \end{cases} \quad (10)$$

当  $m=-1$ ,  $d_i = \text{sign}(Z_i)$ ,  $\theta_i = \text{artanh}(-2^{-i})$ ,  $X_0 = K = \prod_{i=1}^{n-1} (1 - 2^{-2i})^{-1/2} \approx 1.2048$ ,  $Y_0 = 0$ ,  $Z_0 = \theta$  时, 经过  $n$  次迭代, 如式(9)

所示, 即可求得  $X_n = \cosh \theta$ ,  $Y_n = \sinh \theta$ 。需要注意的是, 为了保证迭代收敛, 当  $i \in \{4, 13, \dots, k, 3k+1, \dots\}$  时, 都必须重复迭代一次,  $Z_i$  的取值范围为:  $[-1.113, 1.113]$ <sup>[5]</sup>。在计算时, 迭代次数越多, 最后得到的目标函数值就会越精确。但是迭代次数的增加会增大系统开销, 延长运算时间, 降低实时性, 所以确定合适的迭代次数是获得精确的结果及良好的实时性的重要条件, 设计者需要在精确度和实时性之间取得一定的平衡。

## 2 迭代次数与误差的关系分析

通过上面的分析可以看出, 确定合适的迭代次数是获得精确的结果及良好的实时性的重要条件。而由迭代关系式(9)可知, 影响结果精确度的主要因素就是旋转角度  $\theta_i$ 。由式(8)所示, 当需要求取双曲正余弦函数时,  $\theta_i$  的取值为  $\text{artanh}(-2^{-i})$ , 随着迭代次数的增加,  $\theta_i$  的值越来越小, 将无限趋近于零, 其精度将不断增加, 但对于结果值的影响将逐渐减小。所以迭代次数的选取只需满足系统设计的要求, 并略有冗余即可。为了加以验证, 本文在 MATLAB 中对迭代次数与误差的关系进行了仿真, 如图 2 所示。图 2(a)代表的是用 CORDIC 算法求解双曲余弦

函数时计算的绝对误差与迭代次数的关系曲线,而图 2(b)则代表的是求解双曲正弦函数时的关系曲线,“\*”点表示零误差。该仿真共取了 20 个随机数  $\theta$ , 其取值范围为  $[-1.113,1.113]$ 。

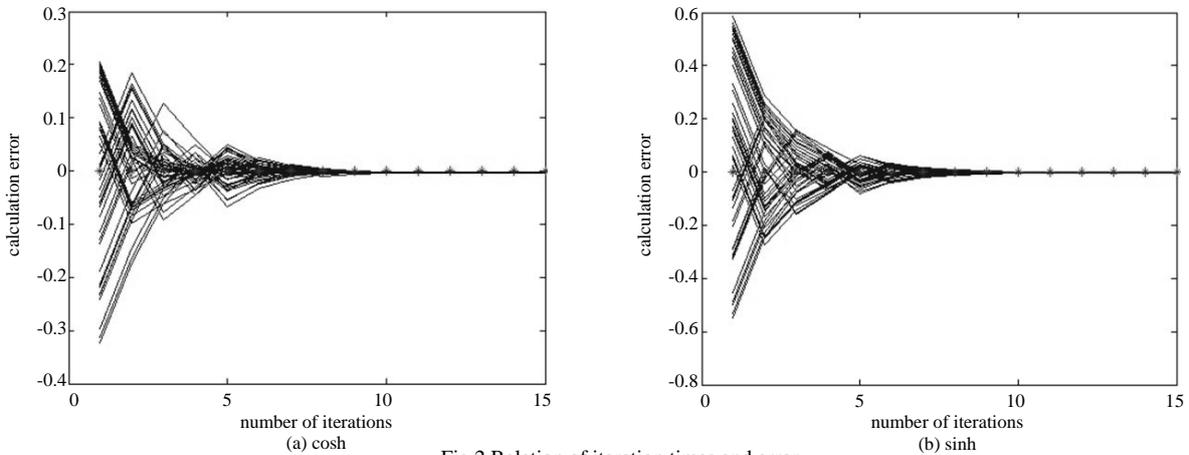


Fig.2 Relation of iteration times and error  
图 2 迭代次数与误差的关系曲线

从图 2 中可以看出,在大多数情况下,当迭代次数为 6 时,误差函数开始收敛;当迭代次数为 9 时,大部分的误差函数已经收敛至较小值。在达到一定的迭代次数后,精度的改善已经不再明显了,所以在设计迭代次数的时候没有必要为了追求精度,将迭代次数设计得过多。

此外,这种情况并不仅仅出现在求取双曲正余弦函数时,用 CORDIC 算法进行计算时,迭代次数与误差关系曲线的趋势大致相同。但对于求解不同的函数,所需要的迭代次数不尽相同,需要在实际应用时加以选择。

### 3 基于 CORDIC 算法的双曲正余弦函数的 FPGA 实现

本文为了兼顾算法的实时性,在 FPGA 实现时将迭代次数设为 8 次。CORDIC 迭代算法的流水线实现方法如图 3 所示。图中,“ $\gg 1$ ”表示向右移 1 位,  $C_n$  为常数,即每次微小旋转的角度  $\theta_i$ , 为:  $C_1 = \text{artanh}(2^{-1}) \approx 0.549306$ ,  $C_2 = \text{artanh}(2^{-2}) \approx 0.255413$ ,  $C_3 = \text{artanh}(2^{-3}) \approx 0.125657$ ,  $C_4 = \text{artanh}(2^{-4}) \approx 0.062582$ ,  $C_5 = \text{artanh}(2^{-4}) \approx 0.062582$ ,  $C_6 = \text{artanh}(2^{-5}) \approx 0.031260$ ,  $C_7 = \text{artanh}(2^{-6}) \approx 0.015626$ ,  $C_8 = \text{artanh}(2^{-7}) \approx 0.007813$ 。

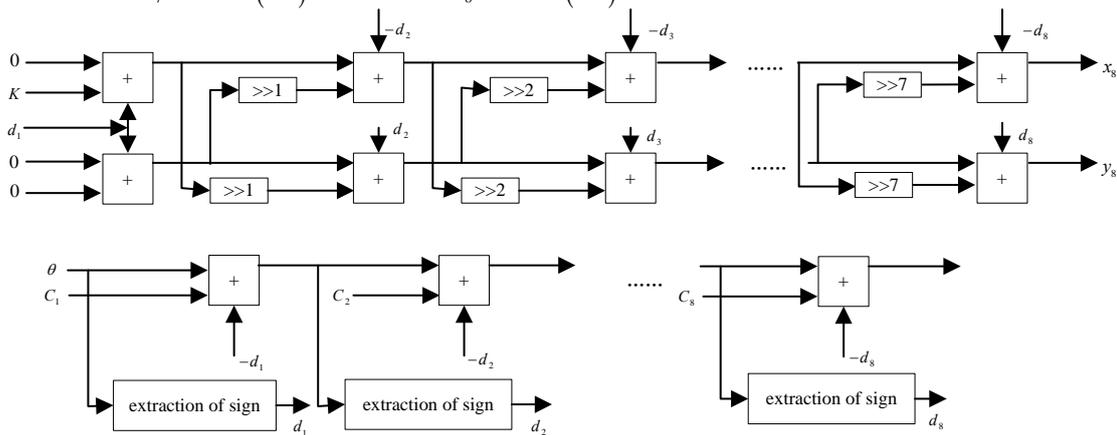


Fig.3 Pipelining structure of CORDIC  
图 3 CORDIC 算法的流水线实现结构

运用流水线结构可以大大提高算法的实时性,只需在等待  $n$  个( $n$  为迭代次数)时钟后就会输出运算结果值,且在其后的每个时钟都会输出一个结果值,提高了运算的效率。

本文利用 Xilinx 公司的 FPGA 芯片 XC2V250 进行了 8 级流水线的算法仿真,仿真结果如图 4 所示。图 4 中  $sita$  即为输入的  $\theta$ ,  $x_1$  到  $x_8$  分别是对应的每一次迭代后的双曲余弦值,  $y_1$  到  $y_8$  则是每一次迭代后的双曲正弦值,以 16 bit 二进制数(其中用 14 bit 表示小数位)表示。图中所示为  $\theta=1$  时的仿真结果,  $x_8$  和  $y_8$  即为最终的计算结果。从图中可以清楚地看到,在复位结束后,经过了 8 个周期开始输出结果值。

表 1、表 2 列出了双曲正弦函数、双曲余弦函数的仿真结果和误差。从几个具有代表性特殊点的仿真结果来看,CORDIC 算法的仿真精度还是比较高的。最大误差为 0.007 800,而最小误差仅为 0.001 404,完全可以满足工程实用需求。

表 3 给出了本文算法和 ISE 自带的 IP 核 CORDIC v3.0 算法所用系统资源及最大时钟频率。可以看出,本文所用实现方法的系统资源消耗少,同时综合后的最高频率高于 IP 核综合后的最高频率,性能良好。

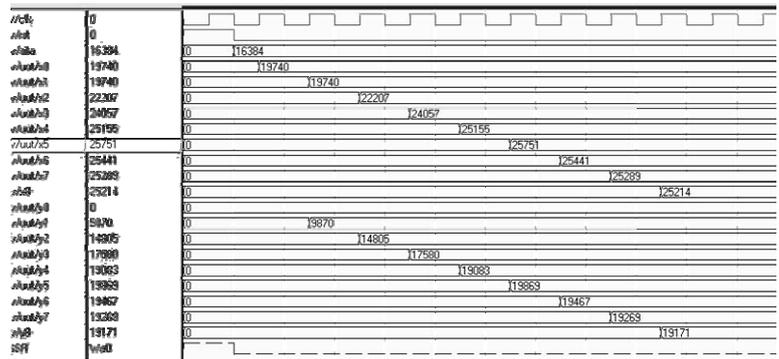


Fig.4 Simulation result of CORDIC  
图 4 CORDIC 算法仿真结果

表 1 双曲正弦函数的仿真结果和误差

Table1 Simulation results and error of hyperbolic sine function

$\theta$	$\sinh(\theta)$	$\sinh(\theta)$ theoretical value	absolute value error
-1	-1.170 166	-1.175 201	0.005 035
-1/2	-0.513 916	-0.521 095	0.007 179
-1/4	-0.244 812	-0.252 612	0.007 800
0	0.005 493	0	0.005 493
1/4	0.244 873	0.252 612	0.007 739
1/2	0.513 977	0.521 095	0.007 118
1	1.170 105	1.1752 01	0.005 096

表 2 双曲余弦函数的仿真结果和误差

Table2 Simulation results and error of hyperbolic cosine function

$\theta$	$\cosh(\theta)$	$\cosh(\theta)$ theoretical value	absolute value error
-1	1.539 001	1.543 081	0.004 080
-1/2	1.124 146	1.1276 26	0.003 480
-1/4	1.029 358	1.031 413	0.002 055
0	1.001 404	1	0.001 404
1/4	1.029 419	1.031 413	0.001 994
1/2	1.124 084	1.127 626	0.003 542
1	1.538 940	1.543 081	0.004 141

表 3 本算法与 IP 核所用系统资源消耗比较

Table3 Comparison of resource consumption

method	slice registers	slice flip flops	slice LUTs	maximum frequency/MHz
method in this paper	180	341	338	265.684
IP Core	676	1 268	1 304	182.357

4 结论

本文在 CORDIC 算法基本原理的基础上,给出了迭代次数和误差的关系曲线,证实了迭代次数越多误差越小的结论。同时根据关系曲线也可知道,在迭代次数达到 10 次以后,误差基本趋近为 0,此时再增加迭代次数对误差的减小作用不大。

另外,本文还给出了 CORDIC 算法在 FPGA 中的实现方法。通过仿真结果与真实值的比较可以看出,CORDIC 算法的误差较小,可以满足大部分工程实用要求。并且与 IP 核相比,该实现方法所用的系统资源少,最大时钟频率高,性能良好,适用于对资源消耗及实时性要求较高的场合。

参考文献:

[ 1 ] Volder J E. The CORDIC trigonometric computing technique[J]. IRE Trans. Electronic Computers, 1959,8(3):330-334.  
 [ 2 ] Walther J S. A unified algorithm for elementary functions[C]// Proceedings of spring joint computer conference, 1971,38: 379-385.  
 [ 3 ] 田耘,徐文波,张延伟. 无线通信 FPGA 设计[M]. 北京:电子工业出版社, 2008.  
 [ 4 ] Angarita F,Perez-Pascual A,Sansaloni T,et al. Efficient FPGA Implimentation of CORDIC algorithm for circular and linear coordinates[C]// Proceedings of International Conference on Field Programmable Logic and Applications. 2005:535-538.  
 [ 5 ] Hahn H,Timmermann D,Hosticka B J,et al. A Unified and Division-Free CORDIC Argument Reduction Method with Unlimited Convergence Domain Including Inverse Hyperbolic Function[J]. IEEE Transactions on computers, 1994,43(11):1339-1344.  
 [ 6 ] UweMeyer-Baese. 数字信号处理的 FPGA 实现[M]. 刘凌,胡永生,译. 北京:清华大学出版社, 2003.

作者简介:



周晓青(1985-),女,四川绵阳人,在读硕士研究生,主要研究方向为通信与信息系统,email:zoe.in.rain@163.com.

李合生(1970-),男,四川巴中人,博士,研究员,主要从事雷达信号处理、雷达对抗、模式识别等研究。

陶荣辉(1969-),男,四川达州人,硕士,副研究员,主要从事雷达对抗、嵌入式系统研发。

蔡英武(1971-),男,四川乐山人,硕士,研究员,主要从事雷达信号处理及雷达对抗研究。