

CORDIC 算法及其展开结构的 FPGA 实现

高兵益 徐磊

(上海理工大学光电信息与计算机工程学院 上海 200093)

摘要: 针对 FPGA 系统中涉及三角函数等数学运算,而传统的查找表方法占用资源较多,计算精度不足等问题,提出了基于 CORDIC 算法的三角函数计算的实现方法。首先给出了 CORDIC 算法的硬件结构,然后使用 Verilog HDL 硬件描述语言,在 Quartus II 13.1 中对算法进行编译和仿真,并对仿真结果和实际结果进行误差分析,最后选择 Altera 公司的 Cyclone IV E 系列中 EP4CE30F23C6 器件,完成了 CORDIC 算法的 FPGA 实现。实验结果表明,该算法能够利用较少的硬件资源,实现较高的计算精度,并且运算速度较快,可以满足实际应用中的计算需求。

关键词: 坐标旋转数字计算机算法;现场可编程门阵列器件;展开结构;三角函数

中图分类号: TN791 **文献标识码:** A **国家标准学科分类代码:** 510.1050

CORDIC algorithm and its FPGA implementation of the deployable structure

Gao Bingyi Xu Lei

(School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: The FPGA system involves the trigonometric function as the mathematical operation, but the traditional look-up table method takes up more resources, the calculation problem of the lack of precision, is proposed in this paper calculation method of trigonometric function based on CORDIC algorithm. Firstly, the hardware structure of the CORDIC algorithm, and then use the Verilog HDL hardware description language in Quartus II 13.1 in the algorithm is compiled and simulated, and the simulation results and actual results of error analysis, the final choice of EP4CE30F23C6 Altera devices Cyclone E series IV, implements FPGA CORDIC algorithm. The experimental results show that the algorithm can use less hardware resources to achieve high computational accuracy, and the computing speed is faster and can meet the practical needs of computing.

Keywords: CORDIC algorithm; FPGA device; deployable structure; trigonometric function

0 引言

坐标旋转数字计算机算法 (coordinate rotation digital computer, CORDIC), 通过移位和加减算法, 能递归计算常用函数值, 如 \sin 、 \cos 、 \sinh 和 \cosh 等函数^[1-3]。由 Volder 于 1959 年提出, 首先用于导航系统, 使得向量的旋转和定向计算不需要做查三角函数表、乘法、开方及反三角函数等复杂运算。Walther 在 1974 年用它研究了一种能计算出多种超越函数的统一算法^[4]。

由于 CORDIC 算法可以将三角函数的计算转化为加减和移位运算, 这非常适合于在硬件电路上实现, 该算法不仅可以节约 FPGA 的硬件资源, 还能大大提高计算精度和速度^[5-6]。本文在介绍了 CORDIC 算法原理的基础上, 提出了 CORDIC 的 3 种硬件结构, 并对三种结构进行了比较分析。遵循“面积与速度”平衡的原则选择展开结构在

Quartus II 13.1 中使用 Verilog HDL 硬件描述语言对算法进行编译和仿真, 并对算法进行了改进和优化, 使其充分利用 FPGA 的硬件资源, 最终完成了 CORDIC 算法的 FPGA 实现。对实验结果进行分析论证可知, 该算法具有速度快、精度高、耗费资源少等优点。

1 CORDIC 算法原理

如图 1 所示, 初始向量 (X_0, Y_0) 旋转 θ 角度之后得到向量 (X_1, Y_1) , 此向量有如下关系, 即:

$$\begin{cases} X_1 = X_0 \cos\theta - Y_0 \sin\theta \\ Y_1 = Y_0 \cos\theta + X_0 \sin\theta \end{cases} \quad (1)$$

假设初始向量经过 n 次旋转之后得到新向量, 且每次旋转的角度 δ 都是正切值 2 的倍数^[7], 则第 i 次旋转角度为 $\delta = \arctan(2^{-i})$, 即 $\cos\delta = \left(\frac{1}{1+2^{-2i}}\right)^{1/2}$ 。

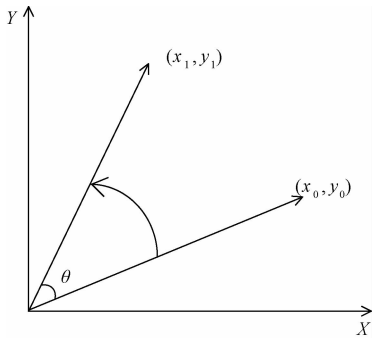


图 1 CORDIC 坐标旋转原理

容易得到角度 $\theta \approx \sum S(i)\delta(i)$, 其中 $S(i) = 1$ 或 -1 , 表示旋转角度的方向, 令 $\Phi_x = (X(i) - S(i)Y(i)2^{-i})$ 、 $\Phi_y = (Y(i) - S(i)X(i)2^{-i})$, 则 i 步旋转可以表示为:

$$\begin{cases} X(i+1) = \Phi_x \left(\frac{1}{1+2^{-2i}} \right)^{1/2} \\ Y(i+1) = \Phi_y \left(\frac{1}{1+2^{-2i}} \right)^{1/2} \end{cases} \quad (2)$$

式中, $\left(\frac{1}{1+2^{-2i}} \right)^{1/2}$ 称为校模因子。当旋转次数一定时, 趋于一个常数, 即:

$$\prod ((1/(1+2^{-2i}))^{1/2}) \approx 0.6073 \quad (3)$$

这样算法每一步就可以简化为:

$$\begin{cases} X(i+1) = (X(i) - S(i)Y(i)2^{-i}) \\ Y(i+1) = (Y(i) + S(i)X(i)2^{-i}) \end{cases} \quad (4)$$

从而可以看出, 对于移动的角度 θ , 现在只需要硬件加、减法和移位器就可以算出结果^[8]。引入 Z , 表示 i 次旋转后相位累加的部分和, 则

$$Z(i+1) = Z(i) - S(i)\arctan(2^{-i}) \quad (5)$$

经过 n 次旋转之后, $Z \rightarrow 0$, 即与目标角重合, 即:

$$\begin{cases} X(n) = X_0 \cos\theta - Y_0 \sin\theta \\ Y(n) = Y_0 \cos\theta - X_0 \sin\theta \end{cases} \quad (6)$$

2 CORDIC 算法的展开结构

在采用 FPGA 实现 CORDIC 算法时, 需要考虑两个方面的因素: 速度与面积。因此 CORDIC 算法有 3 种硬件结构: 紧缩结构、展开结构和展开流水结构。紧缩结构占用资源较少, 可以循环利用 CORDIC 处理单元, 但其时序控制较为繁琐, 系统处理速度较低; 展开结构对紧缩结构进行了扩展, 对每次迭代赋予了独立的 CORDIC 处理单元, 从而不需要控制电路, 只需要移位、加减操作^[9-10]。

相比于紧缩结构, 展开结构需要消耗更多的资源; 展开流水结构在展开结构的基础上添加了流水寄存器, 有效地缩短了关键路径, 展开流水结构具有较高的处理速度, 也需要消耗更多的资源。本文在综合考虑资源与速度的基础上, 选择展开结构来实现, 达到“面积与速度平衡”, 使得

CORDIC 算法与系统结构完美地匹配^[11-12]。CORDIC 算法的展开结构如图 2 所示, 由若干个 CORDIC 处理单元组成。

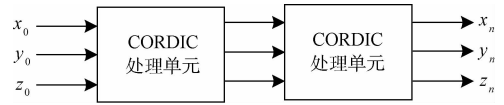


图 2 CORDIC 展开结构

展开结构中的每个 CORDIC 处理单元有相同的架构——移位加减蝶形单元, 它由加法器和移位模块共同构成, 如图 3 所示。展开结构不存在反馈电路, 结构相对简单, 有利于 FPGA 实现, 加法器根据 sgn 确定工作模式^[13]。展开结构每次迭代都有独立的处理单元, 显然, 此结构所用资源与迭代次数有直接关系^[14]。

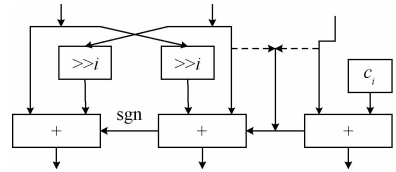


图 3 移位加减蝶形单元

3 CORDIC 算法的 FPGA 实现

3.1 算法的 FPGA 设计

为了在 FPGA 器件上实现 CORDIC 算法, 须对算法作适当的简化和调整。根据 CORDIC 算法原理, 式 (6) 可进一步变形为:

$$\begin{cases} X(n) = X_0 - Y_0 \gg n \\ Y(n) = Y_0 + X_0 \gg n \end{cases} \quad (6)$$

由式 (7) 可以看出, 式中只包含加减和移位运算, 所以非常适合在硬件上实现。

此外, 由于浮点数运算很慢, 而且需要消耗大量的硬件资源, 会严重影响系统的性能。因此设计中采用定点数计算, 其计算结果再转化为实际计算值。

本设计采用展开结构, 输入数据位宽和输出数据位宽均为 32 位。选择 Verilog 硬件描述语言, 以 Altera 的 FPGA 器件为验证平台, 选用 Cyclone IV E 系列中 EP4CE30F23C6 器件, 在 Quartus II 软件中对设计进行综合。Verilog 设计流程如图 4 所示。

CORDIC 算法的核心代码 ($y > 0$ 时) 如下:

```

ty = y >>> i - 1;
tx = x >>> i - 1;
x <= x + ty;
y <= y - tx;
deg <= deg + atan[ i - 1 ];
i <= i + 1'b1;
    
```

其中, x, y 为输入, deg 为角度, i 为迭代变量。

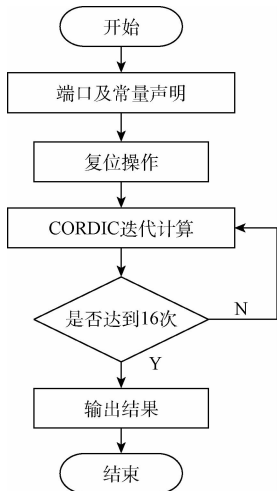


图4 Verilog设计流程

本设计的建模模型如图5所示,输入、输出数据位宽均为32位。

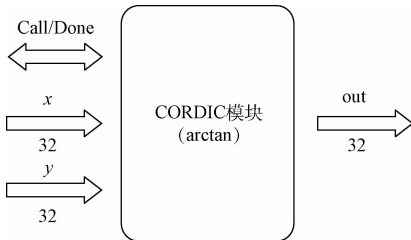


图5 CORDIC算法模型

3.2 实验结果及分析

在Quartus II 13.1开发环境下使用ModelSim仿真工具对上述设计进行仿真验证,得到如图6所示的仿真波形。

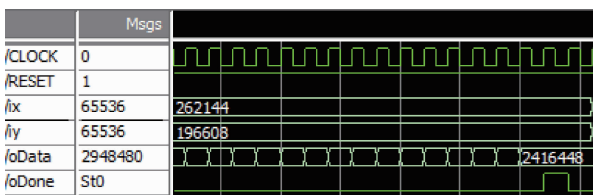


图6 仿真波形

图6对应的具体仿真结果及分析如表1所示,其中 x 、 y 为输入, out 为输出, $arctan$ 为反正切值,即角度值($^{\circ}$),计算结果精确到0.001。从表1可以看出,当 $x=4, y=3$ 时,输出结果 out 为2416448,即 $2416448/65536=36.872^{\circ}$,与理论计算值 $atan(3/4)=36.870^{\circ}$ 相符。此外,由仿真波形图可知,一次计算可在20个时钟之内完成。由此可说明,本设计完成了CORDIC算法在FPGA器件上实现,且计算精度和速度都较为理想。

将上述算法设计下载至Altera公司的Cyclone IV E系列EP4CE30F23C6芯片中,编译成功后算法所消耗资源

表1 仿真结果及分析

x	y	out	arctan	理论值	误差
4	1	919872	14.036	14.036	0
2	1	1740864	26.563	26.565	0.002
4	3	2416448	36.872	36.870	0.002
1	1	2949056	44.999	45.000	0.001
1	2	4157376	63.437	63.435	0.002
1	4	4978368	75.964	75.964	0

的报告如图6所示,从图中可以看出,该算法只使用了2%的系统资源,可以有效地作为其他系统的计算单元。

Flow Summary	
Flow Status	Successful - Mon Apr 24 23:20:48 2017
Quartus II 64-Bit Version	13.1.0 Build 162 10/23/2013 SJ Full Version
Revision Name	cordic
Top-level Entity Name	cordic
Family	Cyclone IV E
Total logic elements	370 / 15,408 (2%)
Total combinational functions	354 / 15,408 (2%)
Dedicated logic registers	136 / 15,408 (< 1%)
Total registers	136

图7 系统资源使用报告

4 结论

CORDIC是一种基于向量旋转的迭代算法,将CORDIC应用于FPGA等数字电路中,解决了传统的基于ROM查找表消耗系统资源多且速度不足的缺点。此外,该算法可以被扩展至其它数学函数的计算,可以很好地解决数字电路本身计算性能的瓶颈。本文基于CORDIC算法讨论了该算法的3种硬件结构,并实现了CORDIC算法在FPGA平台上运行。设计中采用展开结构,综合考虑了硬件资源与运行效率的平衡,使得CORDIC算法与系统结构达到最佳匹配。实验结果与实际值误差小且计算速度较快,可以在20个时钟周期内完成。CORDIC算法应用广泛,在离散傅里叶变换、离散余弦变换、离散Hartley变换、Chirp-Z变换、各种滤波及矩阵的奇异值分解中都可应用CORDIC算法。因此,本设计可进一步用于其它方面的研究^[15-16]。

参考文献

[1] YU B, GUANG Y, HUANG H. The implementation of single precision sin&-cos function on FPGA based on the CORDIC algorithm [J]. Applied Mechanics and Materials, 2013, 2617 (380): 1812-1815.

[2] WONG C C, CHENG C C, LIU C C, et al. CORDIC-based FPGA hardware design method for biped walking gait [J]. Journal of the Chinese Institute of Engineers, 2015, 38(5): 610-620.

- [3] 骆钟频,陈晓,武艺鸣,等. 用于地铁的 OFDM 测距雷达的 FPGA 实现[J]. 电子测量技术,2015,38(3):87-91,99.
- [4] 宋晓梅,朱辉,王文静. 基于 CORDIC 的旋转变压器解码算法研究[J]. 电子测量技术,2010,33(6):39-43.
- [5] 杨健,刘渝,狄慧,等. 长基线宽带测向算法及 FPGA 实现[J]. 电子测量与仪器学报,2013,27(9):897-905.
- [6] 魏岳伦,岳庆英,齐庆堃,等. 基于 Nios II 的 DDS 函数信号发生器[J]. 国外电子测量技术,2016,35(7):13-16.
- [7] 王少军,张启荣,彭宇,等. 超越函数 FPGA 计算的最佳等距分段线性逼近方法[J]. 仪器仪表学报,2014,35(6):1209-1216.
- [8] 祁艳杰,刘章发. 基于 Parallel_CORDIC 的高精度高速度直接数字频率合成器的 FPGA 实现[J]. 电子学报,2014(7):1392-1397.
- [9] 常柯阳,曾岳南,陈平,等. CORDIC 算法在正余弦函数中的应用及其 FPGA 实现[J]. 计算机工程与应用,2013(7):140-143.
- [10] 刘小会,许蕾,刘海颖,等. 基于 CORDIC 改进算法的反正切函数在 FPGA 中的实现[J]. 计算机技术与发展,2013(11):103-107.
- [11] 李全,李晓环,陈石平. 基于 CORDIC 算法高精度浮点超越函数的 FPGA 实现[J]. 电子技术应用,2009,35(5):166-170.
- [12] 常柯阳,曾岳南,陈平,等. CORDIC 算法在正余弦函数中的应用及其 FPGA 实现[J]. 计算机工程与应用,2013(7):140-143.
- [13] 彭奇,习友宝. 基于 FPGA 和改进 CORDIC 算法的 NCO 设计与实现[J]. 微型机与应用,2013(5):60-62.
- [14] 刘春雅,冯向莉. 基于 CORDIC 算法的数控振荡器及其 FPGA 实现[J]. 自动化与仪器仪表,2015(8):131-133.
- [15] 唐文明,刘桂雄. 指数函数 CORDIC 算法的 FPGA 定点化技术[J]. 华南理工大学学报:自然科学版,2016,44(7):9-14.
- [16] 王伟珽,张文旭,陶东香. 利用 CORDIC 算法提取脉内特征在 FPGA 上的实现[J]. 现代电子技术,2016,39(9):1-5,11.

作者简介

高兵益,1991 年出生,硕士研究生,研究方向为数字系统设计。

E-mail:bingyi.gao@foxmail.com