

文章编号:1007-130X(2011)10-0064-06

全定制 CORDIC 运算器设计^{*}

Full Custom CORDIC Arithmetic Unit Design

毕 卓,戴益君

BI Zhuo,DAI Yi-jun

(上海大学机电工程与自动化学院,上海 200072)

(School of Mechatronic Engineering and Automation,Shanghai University,Shanghai 200072,China)

摘 要:浮点三角函数计算是导航系统、三维图像处理、雷达信号预处理等领域的基本运算。本文采用 CORDIC 算法及全定制集成电路设计方法实现了一种浮点三角函数计算电路,其输出数据兼容 IEEE-754 单精度浮点数标准。本文首先介绍了 CORDIC 算法的原理,并根据性能优先的原则采用了流水线结构;然后给出了基于 SMIC 0.13 μm 1P8M CMOS 工艺下的静态电路结构及版图设计。全定制 CORDIC 运算器的面积为 605 284 μm^2 ,最长路径延时(SS 条件下)为 3.013ns。

Abstract:Floating-point trigonometric computing is a fundamental operation in navigation systems, 3D image processing, radar signal preprocessing and so on. This paper presents a floating point trigonometric computing circuit using a CORDIC algorithm and a full custom layout method, and the output data are compatible with the IEEE-754 signal precision floating-point standard. It describes the CORDIC algorithm principle and chooses a pipeline structure based on the principle of priority on performance. A static circuit structure and a full custom layout are given in the SIMC 0.13 μm 1P8M CMOS process. The silicon area of data path is 605 284 μm^2 and the critical path delay is 3.013ns in the SS (Slow-Slow) corners.

关键词:CORDIC;流水线结构;全定制

Key words:CORDIC;pipeline;full custom

doi:10.3969/j.issn.1007-130X.2011.10.012

中图分类号:TP332

文献标识码:A

1 引言

浮点三角函数计算是许多科学与工程应用中的基本运算,特别是在导航系统、三维图像、雷达信号预处理子系统中包含着大量的浮点三角函数运算。现有的查表、泰勒级数近似、二项式近似等^[1]硬件浮点三角函数运算器存在着面积大、复杂程度高、实时性差等诸多问题。因此,如何以合理的硬件代价来实现快速、高精度、小面积的浮点三角函

数运算电路,提高整个系统的浮点运算能力变得非常重要。坐标旋转数字计算 CORDIC(Coordinate Rotational Digital Computer,简称 CORDIC)算法仅采用简单的移位、加减等操作就可实现复杂运算,硬件实现方便。

2 CORDIC 算法基本原理

CORDIC 是由 Volder J D^[2]于 1959 年首次提出,首先用于导航系统,使得矢量的旋转和定向运

^{*} 收稿日期:2010-05-20;修订日期:2010-10-26
通讯地址:200072 上海市延长路 149 号上海大学电机楼 407A 室
Address:Room 407A,Dianji Building,Shanghai University,149 Yanchang Rd,Shanghai 200072,P. R. China

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。该算法是一种递归算法,通过引入确定的初值,只需结合简单的移位和加减法,就能实现这些复杂的函数运算。为了扩展可解决的基本函数的个数,Walter J^[3]于1971年提出了统一的CORDIC算法。在2004年,Juang Tso-Bing等又提出了一种改进的并行的CORDIC算法,该改进的算法主要运用BBR(Binary-To-Bipolar Recoding,简称BBR)和MAR(Microrotation Angle Recoding,简称MAR)^[4],大大提高了CORDIC算法的迭代速度,并且达到了很高的精度。CORDIC算法主要包含圆周系统、线性系统、双曲系统这三种旋转系统,而每种系统又具有两种模式,即向量模式和旋转模式^[5]。本文主要研究圆周系统的旋转模式。

将 $\vec{A}(x_1, y_1)$ 旋转角度 θ 后得到了一个新的 $\vec{B}(x_2, y_2)$,那么 \vec{A} 、 \vec{B} 之间存在如下关系式:

$$\begin{cases} x_2 = R\cos(\theta + \beta) = x_1\cos\theta - y_1\sin\theta \\ y_2 = R\sin(\theta + \beta) = x_1\sin\theta + y_1\cos\theta \end{cases} \quad (1)$$

其中, R 为圆周的半径,将上式化为矩阵形式,那么平面旋转的定义为:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \text{进一步化简得到} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} =$$

$\cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ 。此处假设使用迭代的方法,每次旋转一定的角度, N 步之后完成给定平面的坐标旋转,此时迭代公式为:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -\tan\theta_n \\ \tan\theta_n & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix}, \text{其中 } \theta_n = \arctan 2^{-n}。 \text{令 } \sum_{n=0}^{\infty} S_n \theta_n = \theta, \text{其中 } S_n = \{-1, +1\},$$

即所有迭代角之和等于给定旋转角度。每一次旋转的角度与目标角度的差值为: $Z_n = \theta - \sum_{i=0}^{n-1} S_i \theta_i$, S_n 为符号函数,当 $S_n = +1$ 时, $Z_n \leq 0$ 即顺时针方向旋转;当 $S_n = -1$ 时, $Z_n > 0$ 即逆时针方向旋转。经过 N 次迭代后:

$$\begin{bmatrix} X_j \\ Y_j \end{bmatrix} = \prod_{n=0}^N \cos\theta_n \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (2)$$

其中:

$$K = \frac{1}{P} = \prod_{n=0}^N \cos\theta_n =$$

$$\prod_{n=0}^N \cos(\arctan 2^{-n}) = \prod_{n=0}^N \frac{1}{\sqrt{1+2^{-2n}}} \quad (3)$$

K 称为聚焦常数, $P = \frac{1}{K}$ 称为旋转增益。当迭代次数趋于无穷大时, K 值收敛于0.607 252 935。以旋转模式为前提,整理式(2), X 、 Y 、 Z 之间的相互关系式为:

$$\begin{cases} X_{n+1} = X_n - S_n Y_n 2^{-n} \\ Y_{n+1} = Y_n + S_n X_n 2^{-n} \\ Z_{n+1} = Z_n - S_n \arctan(2^{-n}) \end{cases} \quad (4)$$

当输入的初值为 $\begin{cases} X_0 = K \\ Y_0 = 0 \\ Z_0 = \theta \end{cases}$,就可推导得到

$$\begin{cases} X_{n+1} = \cos\theta \\ Y_{n+1} = \sin\theta \\ Z_{n+1} = 0 \end{cases}$$

根据以上的分析,CORDIC算法在圆周系统的旋转模式下,通过简单的加、减、移位等,再结合一定的初值,就可以用来计算一个输入角的正弦、余弦值等。此外,CORDIC算法通过相应的初值还能直接求出 $\tan^{-1}z$ 、 $\sinh z$ 、 $\cosh z$ 、 $\tanh^{-1}z$ 等。

3 CORDIC 算法的流水线结构

实现32位的CORDIC算法有几种不同的硬件构架,主要的有迭代法和流水线法^[6]。理想的CORDIC算法要求运算速度快,消耗资源少,但实际中无法同时满足这两点要求。对于迭代法,共用运算器的优点是可节约大量面积,缺点是由于每次迭代所需的数据移位的位数不同,需要配置一个支持0~31位移位的算术移位器,明显制约着速度的提升。而流水线方法的缺点是需要更多的加法器,但由于每次迭代的移位位数是固定的,因此直接采用金属折线取代移位器实现移位,不仅提高了速度,也在一定程度上压缩了面积。对于迭代算法结构,数据的速度是时钟频率的 $1/N$, N 为迭代次数,即需要经过 N 次的迭代才能得到结果,显然效率不高,但这种方法所占用的面积较少。相对于流水线方法,则是采用了空间换取时间的方法,提高了吞吐量。本文选用流水线架构,通过把CORDIC算法的三个不同等式,即式(4)转换为相应的硬件结构,在输入任意角度时,得到三角函数值。

图1为采用CORDIC算法的流水线架构,主要由Quadrant模块、CORDIC模块、Post模块和Controller模块组成。其中,Controller模块控制

着整个流水线的时序配合。对于 CORDIC 模块的每次运算,均取 Z 的最高位,即符号位作为同级运算的控制信号。由于采用流水线的结构,可直接将上一级的数据送入下一级,用不同的金属折线来表示相应的移位寄存器。另外,CORDIC 算法的计算过程中每一次迭代都需要通过查找表找到相应的反正切的角度值,对于流水线结构,在确定级数之后,其相应的正切值也就确定下来。在电路实现过程中,本文同样用金属线来代替各级的正切值,提高硬件的运算速度。整个的 CORDIC 模块的实现采用流水线结构,尽管所占面积比迭代架构大,但是吞吐量比迭代法高。

综上所述,CORDIC 算法具有结构简单,易用硬件实现等诸多优势。

CORDIC 算法在圆周系统的旋转模式下输入一定初值就可以直接算出相应的正余弦值,但是对于 CORDIC 模块,并无法直接处理角度值、数据输出值。因此需要在 CORDIC 模块之前,对数据格式进行定义和输入角度进行预处理等操作。浮点数运算比较复杂,因此,需要预先将浮点格式转化为相应的定点格式。定点 CORDIC 模块内部的角度值和数值都采用便于移位和加减运算的整数数据格式。在正余弦函数的具体硬件实现过程中,以度来表示输入角度的单位。为提高计算的精度,本文令 2^{32} 为 360 度,存在如下关系式: $1^{\circ} =$

$0xb60b61$ (十六进制)。

正余弦值的大小总在 $[0,1]$ 内,对于 32 位操作数,本文采用第 31 位表示符号位,第 30 位作为防止数据溢出位,其余的第 0 位至第 29 位共 30 位用二进制数来表示数值 1。如 0.1 用十六进制表示: $0.1 = 0x06666666$ 。

CORDIC 算法在圆周系统下的旋转模式中,按照 Walther J 提出的迭代次序为 $0,1,2,\cdots,n-1$,其覆盖范围的角度约为 $[-99^{\circ},99^{\circ}]$,而在实际系统中需要的范围为 $[-180^{\circ},180^{\circ}]$,才能在输出时得到正确的正弦或余弦值。因此,需要预先对输入的角度进行处理,如图 1 中的 Quadrant 模块,将第 2、3、4 象限的角度通过角度转换关系式,全部转化到第一象限。另外,通过 Post 模块来处理输入角度的正余弦值的分配。综上所述,整个 CORDIC 运算器的硬件结构包含三部分:Quadrant 模块,预处理角度的变化;CORDIC 模块,求取角度的正余弦值;Post 模块,处理正余弦的符号等。

4 CORDIC 算法的流水线结构

4.1 仿真分析

通过 Synopsis 的 Design Compiler 分析(见表 1),考虑了面积和时延等相关因素,本文选用了以 18 级流水线为基准来完成随后的整个设计,其中

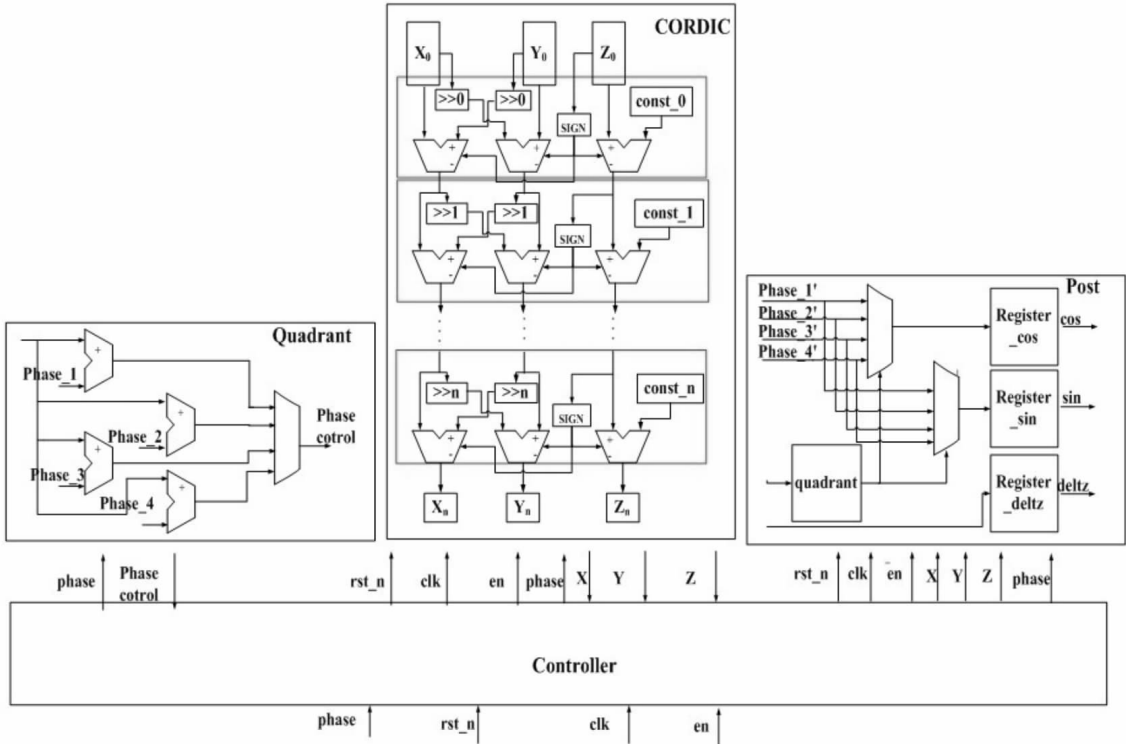


图 1 采用 CORDIC 算法的流水线架构

16 级的 CORDIC 模块采用两个 32 位的加减器和一个 32 位的带有控制端的寄存器组合而成。

表 1 DC 分析 (SMIC 0.13 μ m 1P8M SS 工艺库)

CORDIC 模块 所需流水线的级数	总运算器所需 流水线级数	最长时延 (ns)	面积 (μ m ²)
1	3	101.86	222 249.265 6
8	10	67.45	437 251.968 8
16	18	6.74	610 494.875 0
32	34	4.34	664 274.250

使用 NOVAS 的 Debussy 对本文所设计的模块进行功能仿真,结果如图 2 所示,其中 phase 为输入计算角度,rst_n 为复位信号,clk 为时钟周期信号,en 为使能信号,en_o_post 表示整个模块已完成运算。输出值有 cos、sin 和 deltz,其中 deltz 为 CORDIC 运算器中 Post 模块的 Z 端输出,表示最后的差值,用补码形式表示。仿真结果显示 deltz 为 0,即误差值为 0。将正余弦函数的计算结果用连续的曲线连接起来,同时角度从 0 度开始,以一度为单位,线性递增。通过 18 级的流水线运算,得到数据的精度达到 10^{-8} 。

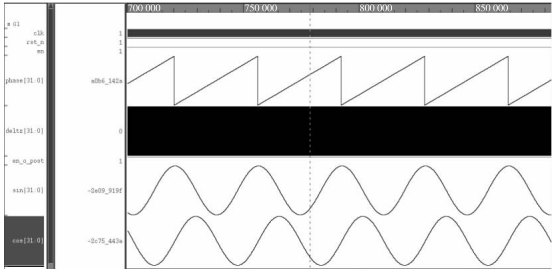


图 2 角度递增的仿真波形图

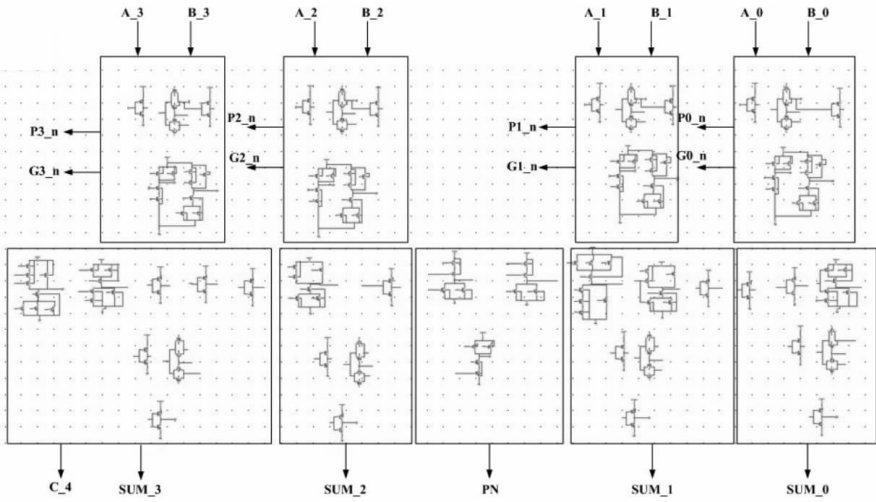


图 3 4 位加法器电路图



图 4 32 位加法器版图

从 CORDIC 算法特性上来说,正余弦函数可以同时计算出来,并从图 2 中可看出,正余弦函数值严格满足 90 度的相位关系。通过仿真波形图中的 deltz 近似为一条为 0 的直线也可以看出,通过 CORDIC 运算后误差小。

4.2 全定制实现

CORDIC 模块中主要为 32 位加减法器和 32 位寄存器的设计。加减法器是 CORDIC 模块的核心部件,其延迟决定了整个模块的关键路径。本文加减法器选用的是以超前进位加法器为基础,并选择组内超前、组间串行的进位方式,每个组的进位传播过程都可以看作是一个单独的实例单元。若本组产生进位信号,可以通过旁路直接传递至下一级。图 3 为含减法的超前进位加法器中的 4 位电路图,其中包含了 PG 函数生成器和控制信号 PN。

版图绘制工作基于 Cadence 公司的 Virtuoso 工具完成,选用 SMIC 的 0.13 μ m 1P8M 的工艺库。图 4 为 32 位加法器的版图概貌,32 位的加法器所占面积为 5 383.265 75 μ m²。

对于全定制设计本文采用层次化的方式,首先是针对 CORDIC 模块为 16 级流水线,设计结构图,进行金属规划;随后绘制各个单元模块,得到一个 4 位单元的数据通道;最后进行数据通道的拼接,构成版图。

本文对 CORDIC 模块中一级流水线进行仿真,电路如图 5 所示,一级流水线主要由 2 个 32 位加减器和 32 位的含控制端的改进型寄存器组成,

采用分段 π 模型来作为线负载模型。

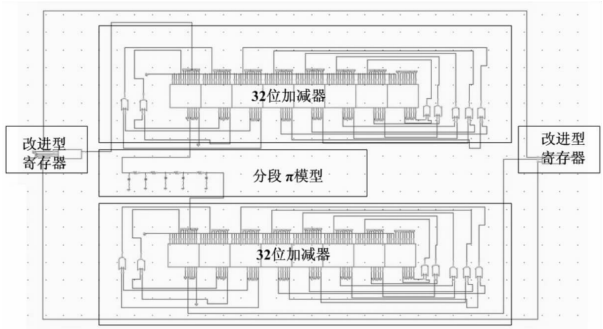


图 5 CORDIC 模块中 1 级流水线结构

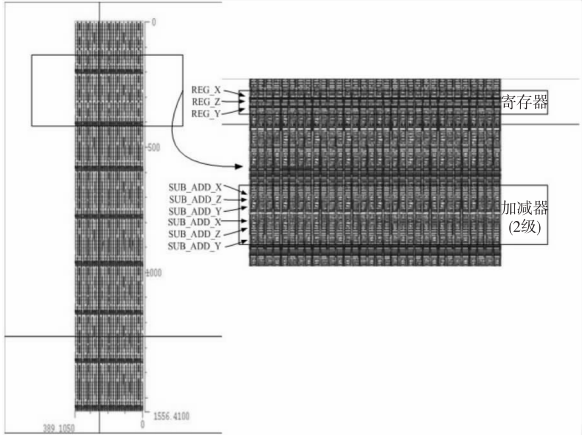
从仿真结果(见表 2)可知,在进位端 C0 和 C0_2 不变的情况下,从输入端(In)到输出的平均时延为 0.608ns;在输入端和进位端 C0_2 不变的情况下,进位端 C0 至最终进位端的平均时延为 2.811ns;在 C0 端和输入端不变的情况下,C0_2 的平均延迟为 2.811 5ns;在输入端不变的情况下,C0 和 C0_2 同时变化,其平均延迟为 2.748ns,最长时延为 3.013ns,这是一个最差情况下的时延,在实际工作条件下,绝大多数情况会好于这个数据。

表 2 CORDIC 模块中 1 级流水线结构时延分析结果

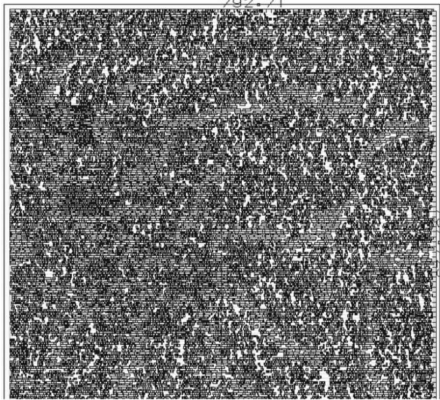
	In	In	C0	C0	C0_2	C0_2	C0	C0_2	C0	C0_2
In	↑	↓	0	0	0	0	0	0	0	0
C0	0	0	↑	↓	0	0	↑	↑	↓	↓
C0_2	0	0	0	0	↑	↓	↑	↑	↓	↓
Delay (ns)	0.543	0.673	2.61	3.012	2.61	3.013	2.551	2.54	3.013	2.887

图 6a 为采用全定制方式设计得到的 32 位 18 级流水线形式的 CORDIC 模块的版图概貌,图 6b 为采用标准单元设计方法实现的 32 位 18 级流水线形式的版图概貌。

本文采用同一方法对 CORDIC 模块分别用全定制方式和标准版图方式实现,并将本文所作的研



a 32次迭代CORDIC模块数据通道版图



b 32次迭代后标准单元版图

图 6 版图概貌

究结果与相关的参考文献在时延和面积上进行比较,结果如表 3 所示。

从表 3 中可知,通过全定制的设计,所需的时延和面积均小于采用 FPGA 和标准单元,采用基 2 方法得到的数据精度高于采用基 4 的方法。文献[9]中的低延迟角度记录法是一种类似基 4 的方法,表 3 中标 * 处为单元门总数,根据文献[10]所提供的公式,本文 * 处面积约为 2.8033mm²,约

表 3 全定制版图与标准单元版图比较

	文献[7]	文献[8]	文献[9]	本文	
位数	32	32	64	32	
工艺(nm)	90	130	130	130	130
方法	基 4	基 2	低延迟角度记录	基 2	基 2
实现方式	FPGA(XC3S1500-4FG676)	FPGA(EP1C6Q240C8)	标准单元库		全定制
面积(μm^2)/逻辑单元数	/	4 287	645.7K(gates) *	623 304	605 284
占用率(%)	/	72	/	95	/
流水线级数	/	34	34	18	18
频率(MHz)	71.233 4	134.10	250	148.37	331.90
时延(ns)	14.039	7.457	4	6.74	3.013
精度	10^{-4}	10^{-8}	10^{-4}	10^{-8}	10^{-8}

为本文 32 位全定制版图的 4.6 倍,但本文所提出的用全定制 32 位实现 CORDIC 算法得到的数据精度比文献[9]的精度高 10^4 倍。

5 结束语

本文采用基 2 的 CORDIC 算法,选用 SMIC 0.13 μm 1P8M CMOS 工艺库,采用正向全定制的电路及版图设计方法,实现了一个支持 IEEE-754 标准的 32 位浮点数的三角函数计算。整个设计的精度可达到 1.334×10^{-8} ,面积为 605 284 μm^2 ,延迟为 3.013ns。

全定制方式得到的时延相较于标准单元的时延要少,但是芯片所占面积并未明显降低。后续改进可以针对面积进行优化,如采用锁存器替代现在使用的寄存器。此外,加法器是整个计算电路的核心,也可以在超前进位链的设计中使用不对称晶体管来加速进位操作。

参考文献:

- [1] 陆志坚. 基于 CORDIC 的可重构处理单元结构研究与设计[D]. 哈尔滨:哈尔滨工程大学,2008.
- [2] Volder J E. The CORDIC Trigonometric Computing Technique[J]. IRE Transactions on Electronic Computers, 1959, 8(3): 330-334.
- [3] Walther J S. A Unified Algorithm for Elementary Functions[C]//Proc of Joint Computer Conference,1971:379-385.
- [4] Juang Tso-Bing, Hsiao Shen-Fu, Tsai Ming-Yu. ParacORDIC: Parallel CORDIC Rotation Algorithm[J]. Circuits and Systems I: Regular Papers, 2004, 51(8): 1515-1524.
- [5] Vachhani L, Sridharan K, Meher P K. Efficient CORDIC Al-

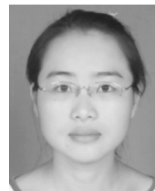
gorithms and Architectures for Low Area and High Throughput Implementation[J]. IEEE Transactions on Circuits and Systems - II: Express Briefs, 2009, 56(1): 61-65.

- [6] 赵锦江. 基于 CORDIC 算法的数字调制器设计与实现[D]. 长沙:国防科技大学,2009.
- [7] Kaushik B, Rakesh B. Architectural Design and FPGA Implementation of Radix-4 CORDIC Processor[J]. Microprocessors and Microsystems, 2010, 34(2-4): 96-101.
- [8] 陈石平, 李全, 付佃华, 等. 32 位浮点正余弦函数的 FPGA 实现[J]. 微计算机信息, 2008, 24(5): 176-178.
- [9] Juang Tso-Bing. Low Latency Angle Recoding Methods for the Higher Bit-Width Parallel CORDIC Rotator Implementations[J]. IEEE Transactions on Circuits and Systems—II: Express Briefs, 2008, 55(11): 1139-1143.
- [10] Adbelhalim K, Smolyakov V, Genov R. A Phase Synchronization and Magnitude Processor VLSI Architecture for Adaptive Neural Stimulation[C]//Proc of BioCAS'10, 2010: 5-8.



毕卓(1979-),男,吉林敦化人,博士,讲师,CCF 会员(E200010117M),研究方向为微处理器设计及高性能数字电路设计。**E-mail:**Bi_zhuo@yahoo.com.cn

BI Zhuo, born in 1979, PhD, lecturer, CCF member(E200010117M), his research interests include microprocessor and high performance digital circuit design.



戴益君(1987-),女,浙江上虞人,硕士生,研究方向为全定制数字运算电路设计。**E-mail:**daiyijun@shu.edu.cn

DAI Yi-jun, born in 1987, MS candidate, her research interest includes full custom digital circuit design.