

一种CORDIC算法的FPGA实现

牛晨晓¹, 赵 忠¹, 聂 聪²

(1. 西北工业大学 自动化学院, 陕西 西安 710129;

2. 西北工业大学 航天学院, 陕西 西安 710072)

摘 要:CORDIC算法是实现快速精确的正、余弦函数计算的主要方法,在工程实际中有着广泛应用。在研究正、余弦函数运算的CORDIC算法简单状态机实现和高速全流水处理机实现的基础上,提出了一种单精度浮点数正、余弦函数运算的优化实现方案,并在ALTERA公司的FPGA上实现。结果表明,相比较单精度浮点数正、余弦函数运算的CORDIC算法简单状态机实现,该实现方案不仅计算速度快,而且硬件资源消耗增加少,达到了单精度浮点数正、余弦函数运算硬件实现上速度与资源占用的平衡。

关键词:坐标旋转数字计算机;现场可编程门阵列;正余弦函数;单精度浮点数

中图分类号:TN4

文献标识码:A

文章编号:1673-629X(2011)06-0016-04

FPGA Implementation of a CORDIC Algorithm

NIU Chen-xiao¹, ZHAO Zhong¹, NIE Cong²

(1. College of Automation, Northwestern Polytechnical University, Xi'an 710129, China;

2. College of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract:CORDIC(Coordinate Rotation Digital Computer) algorithm is the main method to realize the fast and accurate trigonometric function, and is widely used in modern engineering. Based on the study of simple state machine realization and fast pipe-line realization of CORDIC algorithm, propose a new floating-point algorithm's optimized realization, and implement it on ALTERA's FPGA. The result indicates that the optimized realization calculates faster than the simple state machine realization of CORDIC algorithm, while use less resources. This realization realizes the balance between the resource consumption and speed.

Key words:CORDIC;FPGA;trigonometric function;floating-point

0 引言

随着计算机技术和大规模集成电路技术的不断发展,FPGA的集成度越来越高,同时成本也在不断下降^[1]。这些趋势使得FPGA在工程实际中的应用越来越广泛。正、余弦函数运算是实际工程应用中经常用到的运算,特别是在捷联导航系统中的应用更加频繁。

利用查表或多项式展开计算正、余弦函数不适合硬件实现,而且计算速度和精度很难达到要求^[2]。CORDIC算法的提出为快速、精确的三角函数运算提供了一种高效的硬件实现方案,一经提出便引起了广泛关注,并在实际工程应用中得到了广泛的应用。而

且其算法实现也在不断地改进。但是,CORDIC算法实现的改进主要集中在算法结构上,关于其实现结构的改进很少,目前,CORDIC算法实现结构主要有两种,即简单状态机实现和高速全流水处理机实现两种实现方案。这两种实现方案都有自己的优缺点。文中在研究上述两种实现结构的基础上提出了一种改进的CORDIC算法实现方案,该实现方案兼顾计算速度和硬件消耗,可以高效地实现单精度浮点数的正、余弦函数运算。

1 CORDIC算法原理

CORDIC(Coordinate Rotation Digital Computer)是坐标旋转数字计算机的简称。其基本思想是用一系列与运算基数相关的角度不断旋转,逼近所需旋转角度,从广义上讲它是一种数值逼近的方法,由于这些固定的角度与计算基数相关,所以计算过程只涉及移位和加减运算。算法最初是由J. E. Volder在1959年提出的。目的是解决用数字计算机取代B58轰炸机上使

收稿日期:2010-11-02;修回日期:2011-02-19

基金项目:陕西省自然科学基金(2010JQ8015)

作者简介:牛晨晓(1985-),男,河南辉县人,硕士研究生,研究方向为测控技术与自动化装置;赵 忠,教授,硕士研究生导师,研究方向为检测技术与自动化装置,导航、制导与控制,精密仪器及机械。

用的模拟计算机时,所面临的导航位置实时解算问题^[3]。该算法只需使用简单的移位和加减运算就可以完成乘法、开方、查三角函数表以及反三角函数等复杂运算^[4]。因此非常适合硬件实现。而后,John Walther 于 1971 年又对 CORDIC 算法做了改进^[5],提出了统一的改进型 CORDIC 算法,改进后的 CORDIC 算法可以在六种不同的模式下工作,除了可以完成三角函数、反三角函数和求向量的模运算,还可以完成乘法、除法、超越函数等运算^[6]。文中利用 CORDIC 算法实现正、余弦函数运算。其算法原理^[7,8]如下。

向量 (x_i, y_i) 旋转 θ 角,得到一个新向量 (x_j, y_j) ,如图 1 所示。

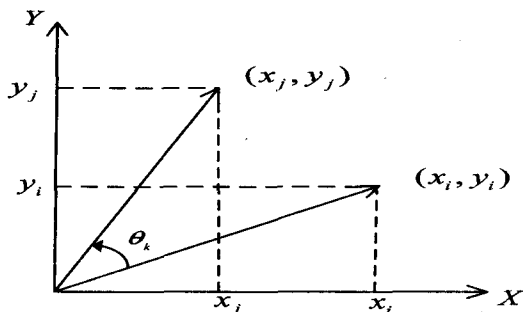


图 1 向量旋转坐标图

那么有

$$x_j = x_i \cos \theta - y_i \sin \theta \quad (1)$$

$$y_j = x_i \sin \theta + y_i \cos \theta$$

写成矩阵形式就是

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \cos \theta \begin{bmatrix} 1 & -\tan \theta \\ \tan \theta & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2)$$

设角度 θ 是由 $n+1$ 个角度 θ_k 线性叠加而成的,即

$$\theta = \sum_{k=0}^n d_k \theta_k + \varepsilon, \text{ 其中 } d_k \text{ 表示旋转的方向,当顺时针旋转时, } d_k = +1, \text{ 当逆时针旋转时, } d_k = -1, \text{ 为了便于算法的硬件实现,将每次的旋转角度 } \theta_k \text{ 加以限制,使得}$$

$\theta_k = \arctan 2^{-k}$, 这样式(2)变为:

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = K \cdot \begin{bmatrix} 1 & -d_k 2^{-n} \\ d_k 2^{-n} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -d_0 2^{-0} \\ d_0 2^{-0} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3)$$

$$\text{其中 } K = \cos \theta_n \cdots \cos \theta_1 \cdot \cos \theta_0 = \prod_n (1/\sqrt{1+2^{-2k}})$$

可以通过预先对向量 (x_i, y_i) 乘以 K 的倒数来消除增益 K 的影响。这样对于 FPGA 来说式(3)中的所有运算都可以由简单的移位和加、减运算单元实现。

从式(3)可以看出,不考虑预调整增益 K , 则每一次旋转可以表示为:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & -d_k 2^{-k} \\ d_k 2^{-k} & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} \quad (4)$$

定义 z_{k+1} 为矢量 (x_i, y_i) 经过 k 次旋转后的角度残差,则有

$$z_{k+1} = z_k - d_k \arctan 2^{-k} \quad (5)$$

综合式(4)、(5),CORDIC 算法的迭代过程可以表示为式(6)。

$$\begin{cases} x_{k+1} = x_k - d_k y_k 2^{-k} \\ y_{k+1} = y_k + d_k x_k 2^{-k} \\ z_{k+1} = z_k - d_k \arctan 2^{-k} \\ d_k = \text{sign}(z_k) \end{cases} \quad (6)$$

在圆周旋转模式下,CORDIC 旋转器对输入向量按照一定的角度进行旋转,使其与目标角度的残差趋向于 0, 式(6)的最终迭代结果为式(7)。

$$\lim_{n \rightarrow \infty} \begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = K \times \begin{pmatrix} x_0 \cos z_0 - y_0 \sin z_0 \\ y_0 \cos z_0 + x_0 \sin z_0 \\ 0 \end{pmatrix} \quad (7)$$

若令初始值 $x_0 = 1/K$, $y_0 = 0$, $z_0 = \theta$, 则迭代终值 x_n 和 y_n 分别是所需角度 θ 的余弦值和正弦值。

从上述推导过程可得,传统 CORDIC 算法角度覆

盖范围为 $[-\sum_{k=0}^{\infty} \arctan 2^{-k}, +\sum_{k=0}^{\infty} \arctan 2^{-k}]$, 即 $\theta \in [-99.88^\circ, 99.88^\circ]$ 角度范围内的正、余弦值。然而一般计算要求 CORDIC 算法的角度计算范围覆盖 $[-\pi, \pi]$, 因此需要对迭代进行修正,扩充角度覆盖范围。这可以通过增加两次 $k=0$ 的迭代来实现,即迭代序列为 $k=0, 0, 0, 1, 2, 3, \dots, n-1$ 。此时,角度计算范围为

$$|\theta| \leq \sum_{k=0}^n \arctan 2^{-k} + \frac{\pi}{2} \quad (8)$$

从式(8)可以看出,只要迭代次数不低于 6 次,CORDIC 算法的角度计算范围就可以覆盖 $[-\pi, \pi]$, 符合一般计算要求。

2 CORDIC 算法的 FPGA 实现

在数值格式上,CORDIC 算法的迭代运算可以使用两种数值格式,即定点数格式和浮点数格式。由于使用 CORDIC 算法计算正、余弦函数时, $|x_n| < 2$, $|y_n| < 2$, $|z_n| < 4$, 而且,相比较浮点数计算,定点数计算可以用较少的资源获得较快的运算速度,因此,文中使用定点数格式进行迭代运算。同时,考虑到 CORDIC 算法迭代计算过程中要使用动态的加、减运算,因此,定点数格式选用二进制补码格式。为了满足单精度浮点数运算的精度要求, x_n 和 y_n 取 31 位定点数,其中符号 1 位,整数 1 位,尾数 29 位。 z_n 取 31 位定点数,其中符号 1 位,整数 2 位,尾数 28 位。迭代次数为 28 次。

在实现方式上,CORDIC 算法的实现方式主要有

两种,即简单状态机实现和高速全流水处理机实现。简单状态机实现方式灵活,只需通过修改数字的位宽和迭代次数,就可以控制正、余弦函数运算的精度和速度,而且消耗的硬件资源少。但是,CORDIC 算法在一次运算中必须多次迭代,而每次迭代必须在前一次迭代完成后进行^[9],因此计算精度要求高时计算速度较慢。高速全流水处理机实现方式可以获得很高的数据吞吐率,但是由于该实现方式在资源复杂性方面随着数字位宽的增加而呈平方的增加^[5],因此主要适用于计算精度要求不高,或计算速度要求很高的场合。

综合上述两种实现方式的优缺点,文中将单精度浮点数正、余弦函数运算的 28 次迭代运算拆分为 7 步实现。第 1 步完成单精度浮点数到二进制补码数转换及 4 次基本 CORDIC 迭代运算。最后一步完成 4 次基本 CORDIC 迭代运算及二进制补码数到单精度浮点数转换。其余各步分别完成 4 次基本 CORDIC 迭代运算。系统结构如图 2 所示。

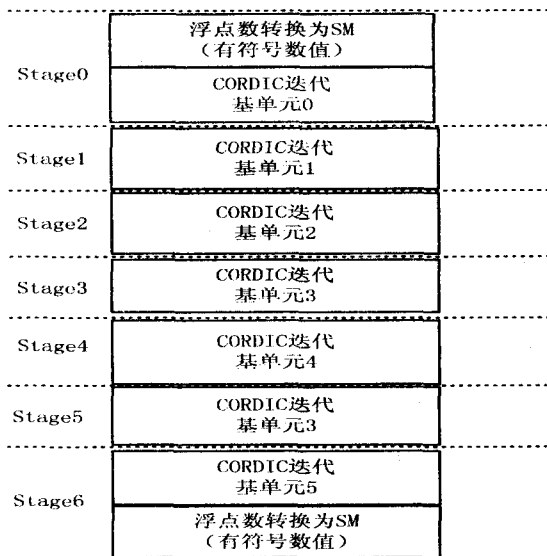


图 2 CORDIC 算法实现结构框图

其中 CORDIC 迭代基单元结构框图如图 3 所示。

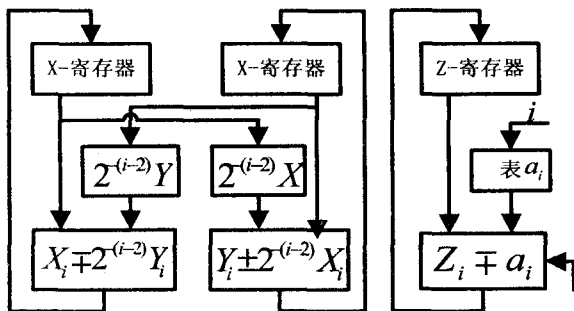


图 3 CORDIC 迭代基单元结构框图

采用上述方式实现 CORDIC 迭代算法时,只需要一步计算完成后就可以进行下一次正、余弦计算。不需要等待所有的迭代过程完成后再进行下一次计算。因此,相比较 CORDIC 算法的纯状态机实现方式,该实

现方式计算速度可以提高约 6 倍。在用 VHDL 语言完成设计输入后,使用 Altera 公司的集成开发环境 Quartus II 8.0 进行编译。在目标芯片为 EP3C16F256I7,工作频率设置为 125MHz 时,占用 2403 个逻辑单元,编译后实际达到的最高时钟频率为 133MHz。

3 CORDIC 算法优化实现的误差分析仿真

3.1 CORDIC 算法优化实现的误差分析

由于文中使用定点数实现正、余弦函数 CORDIC 迭代运算,因此,主要分析定点数实现正、余弦函数 CORDIC 迭代运算的误差来源及其大小。误差来源主要包括由于角度逼近引起的角度近似误差和由于使用有限位定点数迭代计算引起的量化误差^[10~12]。为了方便后面的分析计算,定义

$$V(i) = [x(i) \quad y(i)]^T \quad (9)$$

$$P_m(i) = \begin{bmatrix} 1 & -d_i 2^{-\sigma(i)} \\ d_i 2^{-\sigma(i)} & 1 \end{bmatrix} \quad (10)$$

$$M(i) = \prod_{j=0}^{i-1} P_m(j) \quad (11)$$

3.1.1 角度近似误差

对于文中正、余弦函数的 CORDIC 迭代运算,任何满足 $\theta \in [-\pi, +\pi]$ 的角度 θ ,均由 28 个特定角度 θ_k 的线性组合来逼近,即 θ 表示为:

$$\theta = 2\theta_0 + \sum_{k=0}^{25} d_k \theta_k + \varepsilon \quad (12)$$

其中 ε 为角度近似误差,它是由角度近似引起的。假设 $V^*(n)$ 是理想值,即近似误差 ε 为 0 时的计算值。 $V(n)$ 为实际迭代值,角度近似误差为 ε 。则有:

$$V^*(n) = E \cdot V(n) = \begin{bmatrix} \cos \varepsilon & \sin \varepsilon \\ -\sin \varepsilon & \cos \varepsilon \end{bmatrix} V(n) \quad (13)$$

因此

$$V^*(n) - V(n) = (E - I) V(n) \quad (14)$$

其中 I 为 2×2 的单位矩阵。

由于 $C^{n \times n}$ 的矩阵 F 范数与 C^n 上向量的 2 范数相容,故

$$\|V^*(n) - V(n)\|_2 / \|V(n)\|_2 \leq \|E - I\|_2 \quad (15)$$

其中

$$\|E - I\|_2 = [(\cos \varepsilon - 1)^2 + \sin^2 \varepsilon]^{1/2} = |2 \sin(\varepsilon/2)| \leq \theta_{n-1} \quad (16)$$

故由角度近似误差 ε 引起的相对误差

$$\|V^*(n) - V(n)\|_2 / \|V(n)\|_2 \leq \theta_{n-1} = 2^{-25} \quad (17)$$

3.1.2 量化误差

由于文中采用有限位定点数进行迭代计算,因此迭代计算过程中会由于量化效应而引入误差,称之为

量化误差。

定义 $Q[\cdot]$ 为量化运算符,即

$$Q[V(i)] = V(i) + e(i) \quad (18)$$

其中 $e(i) = [e_x(i) \quad e_y(i)]^T$

经过 n 次迭代运算后,由计算过程中的量化效应引起的误差

$$Q[V(n)] - V(n) = e(n) + \sum_{j=1}^{n-1} \left(\prod_{i=j}^{n-1} P_m(i) \right) e(j) \quad (19)$$

由于CORDIC算法的优化实现方式使用的定点数 x_n 和 y_n 有29位小数位,因此

$$|e(i)| \leq [e_x^2(i) + e_y^2(i)]^{1/2} \leq 2^{1/2} \sigma \quad (20)$$

其中 $\sigma = 2^{-29}$

由式(19)和式(20)得,由计算过程中的量化效应引起的误差

$$\|Q[V(n)] - V(n)\|_2 \leq 2^{1/2}(1 + B_m)\sigma \quad (21)$$

其中

$$B_m = \sum_{j=1}^{n-1} \prod_{i=j}^{n-1} \|P_m(i)\|_2 = 123.7219 \quad (22)$$

综合式(21)和式(22)得,由计算过程中的量化效应引起的误差

$$\|Q[V(n)] - V(n)\|_2 \leq 3.2854e-007 \quad (23)$$

3.2 CORDIC 算法优化实现的仿真

上述分析计算只是从数学的角度对CORDIC算法优化实现方式的误差源和误差的大小进行了初步评估。

为了评估CORDIC算法优化实现方式实际运算时的精度指标,还必须进行实际计算来对其实现精度进行评估。为了增加评估的可靠性,在 $[-\pi, \pi]$ 之间等间隔地取129个数据样本进行仿真计算。仿真实验步骤如下:

1) 使用C语言产生 $[-\pi, \pi]$ 之间等间隔的129个点,并将其以16进制格式保存到数据文件 radix_generate.dat 中;

2) 使用 Modsim 读取 radix_generate.dat 中的测试激励,将其施加到待测模块中,并将仿真结果及测试激励一起保存到数据文件 result.dat 中;

3) 使用C语言读取 result.dat 中的测试激励和仿真结果,计算测试激励对应的真实值,并与仿真结果比较,将误差及测试激励一起保存至数据文件 bias.dat 中;

4) 使用 Matlab 对 bias.dat 中保存的误差进行统计分析,并将测试激励及其对应的误差可视化。

仿真得到的129个样本的正弦计算误差曲线如图4所示。

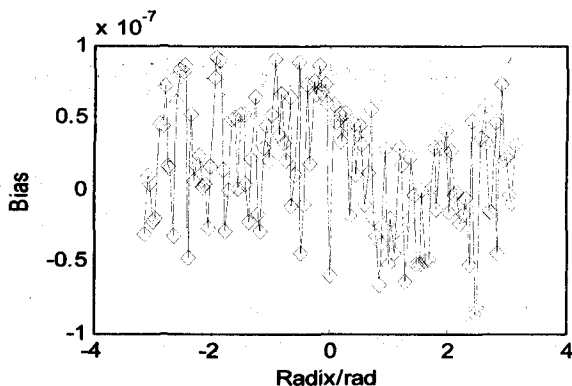


图4 正弦计算误差曲线

余弦计算误差曲线如图5所示。

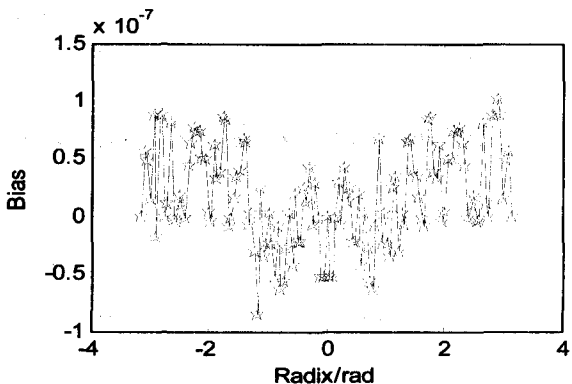


图5 余弦计算误差曲线

从上述大量样本的仿真计算结果可以看出正、余弦函数运算的CORDIC算法优化实现,误差均不超过 2^{-23} ,达到了单精度浮点数正、余弦函数计算要求。

4 结束语

文中提出了一种改进的单精度浮点数正、余弦函数运算的FPGA实现方案,该实现方案结构简单,硬件实现较易。并在Altera公司的Cyclone III系列芯片EP3C16F256I7进行仿真验证,从仿真结果来看,该实现方案可以用较少的资源实现快速、高精度的正、余弦函数运算。它可以应用于导航位置解算、数字音频处理以及其它需要单精度浮点数正、余弦函数计算的工程应用,具有很强的实用价值。

参考文献:

- [1] 王本有. 一种基于FPGA的CPU设计[J]. 化工自动化与仪表, 2008, 18(6): 221-224.
- [2] 陆鹏威, 梅杓春. 基于CORDIC算法实现三角函数的运算[J]. 国外电子测量技术, 2008, 27(1): 10-11.
- [3] Volder J E. The CORDIC trigonometric computing technique[J]. IRE Transactions on Computers, 1959(9): 330-334.
- [4] 杨军. 基于FPGA的FFT处理器研究与设计[J]. 计算机技术与发展, 2009, 19(9): 225-227.
- [5] Meyer-Baese U. 数字信号处理的FPGA实现[M]. 北京: 清

(下转第23页)

征,也可以通过交叉和变异得到具有创新性的结果,为产品的外形创新设计提供了新的方法。但是该方法主

要是面向特定的领域,主要使用于由曲线或曲面构成的简单实体外观造型。



图3 花瓶的初始种群



图4 花瓶的交叉操作



图5 花瓶的变异操作



图6 部分满意的花瓶设计方案

参考文献:

- [1] 刘弘,刘希玉,马丽娜. 支持进化的计算机辅助设计环境[J]. 计算机辅助设计与图形学学报, 2003, 15(2): 167-173.
- [2] 刘弘,刘希玉. 支持外观造型创新设计的进化计算方法[J]. 计算机辅助设计与图形学学报, 2006, 18(1): 101-107.
- [3] 刘弘,李焱. 遗传算法在建筑概念设计中的应用[J]. 软件学报, 2006, 17(11): 161-168.
- [4] 邱莉榕,刘弘. 支持创新概念设计的多Agent系统[J]. 计算机集成制造系统, 2003, 12(9): 38-42.
- [5] 蚁平,曹先彬. 基于交互式遗传算法的个性化建筑物外观设计[J]. 计算机仿真, 2006, 23(5): 156-159.
- [6] 盛立,高明,刘希玉. 基于免疫遗传算法的建筑构件创新设计[J]. 计算机工程与应用, 2006(23): 183-185.
- [7] 徐江,孙守迁. 基于正交-交互式遗传算法的产品造型设计[J]. 计算机集成制造系统, 2007, 13(8): 1470-1475.
- [8] Quiroz J C, Louis S J, Banerjee A, et al. Towards Creative Design Using Collaborative Interactive Genetic Algorithms[C]// 2009 IEEE Congress on Evolutionary Computation. [s. l.]: [s. n.], 2009: 1849-1856.
- [9] 施法中. 计算机辅助几何设计和非均匀有理b样条[M]. 北京:高等教育出版社, 2005: 35-40.
- [10] 易鹏飞,魏小鹏,赵婷婷. 基于遗传算法和非均匀有理B样条表达的产品外型创新设计[J]. 计算机集成制造系统, 2008, 14(2): 221-225.
- [11] 詹海生,李广鑫,马志欣. 基于ACIS的几何造型技术与系统开发[M]. 北京:清华大学出版社, 2002: 305-412.

(上接第19页)

- [6] 华大学出版社, 2006.
- [6] 骆艳卜. 一种CORDIC算法的FPGA实现[J]. 计算机仿真, 2009(9): 305-307.
- [7] 姚亚峰. 高速CORDIC算法的电路设计与实现[J]. 半导体技术, 2008, 33(4): 346-348.
- [8] 刘晓明. 基于IEEE 802.16e的LDPO编译码方案设计及实现[J]. 计算机技术与发展, 2009, 19(5): 205-208.
- [9] 田书林. 一种基于CORDIC算法的信号发生器技术研究[J]. 仪器仪表学报, 2002(3): 150-153.
- [10] 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, 11(10): 1139-1143.
- [11] Ahmed H M, Delosme J M, Morf M. Highly concurrent computing structures for matrix arithmetic and signal processing[J]. IEEE Computer, 1982, 11(15): 65-82.
- [12] Metafas E D, Goutis E C. A Floating Point Advanced CORDIC Processor[J]. Journal of VLSI Signal Processing, 1995, 10: 53-63.