

引用格式:王瑜,蒋平,王建庄. 基于最大逼近角的固定旋转序列 CORDIC 算法[J]. 微电子学与计算机,2020,37(6):57-61.
[WANG Y, JIANG P, WANG J Z. Fixed rotation sequence CORDIC algorithm based on maximum approximation angle[J].
Microelectronics & Computer,2020,37(6):57-61.]

基于最大逼近角的固定旋转序列 CORDIC 算法

王瑜,蒋平,王建庄

(华中科技大学 机械科学与工程学院,湖北 武汉 430074)

摘 要: 为了解决现有 CORDIC 算法旋转序列不固定、算法复杂度高、需要的存储空间大等问题,基于最大逼近角理论对算法加以优化,设计了一种固定旋转序列的 CORDIC 算法结构.通过对角度重编码后二进制的高位旋转策略的分析,确定了对高位每一位都进行两次旋转的固定序列的方法,从而减少了高位旋转方向判断次数,简化了比例常数的计算.最后在 FPGA 中实现对优化前后算法的实验对比分析,验证了优化后的算法相比传统算法在计算复杂度、存储空间使用以及误差控制等方面的优越性和有效性.

关键词: CORDIC 算法;最大逼近角;固定旋转序列;FPGA

中图分类号: TN911.7;TN47

文献标识码: A

文章编号: 1000-7180(2020)06-0057-05

Fixed rotation sequence CORDIC algorithm based on maximum approximation angle

WANG Yu, JIANG Ping, WANG Jian-zhuang

(School of Mechanical Science & Engineering, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract: In order to solve the problems of irregular rotation sequence, high complexity and large storage space in traditional CORDIC algorithm, a CORDIC algorithm structure with fixed rotation sequence is designed based on the theory of maximum approximation angle. By analyzing the strategy of binary high bit rotation after angle re-coding, the method of fixed sequence which rotates each bit twice is determined, which reduces the number of judgments of high bit rotation direction and simplifies the calculation of proportional constant. Finally, the experimental comparison and analysis of the optimization algorithm before and after implementation in the FPGA verifies the superiority and effectiveness of the optimized algorithm compared with the traditional algorithm in the aspects of computational complexity, storage space usage and error control.

Key words: CORDIC algorithm; maximum approximation angle; fixed rotation sequence; FPGA

1 引言

自 Jack Volder^[1]于 1959 年提出 CORDIC 算法用于在硬件平台通过移位和加减的方式实现三角函数、反三角函数、指数函数等复杂函数和功能以来,CORDIC 算法被越来越多的学者优化和实现更多

的函数计算功能. Shaoyun Wang 等人^[2]将旋转角度进行重编码,将二进制的角度值转换成反正切函数序列,使计算过程更加易于硬件实现. Shen-Fu Hsiao^[3]等人在角度重编码基础上使用微角度旋转 (MAR) 算法,使角度划分更加细致,减少了函数计算过程产生的误差,计算结果更加精确. 王建庄等^[4]

基于角度重编码 CORDIC 算法 (BBR) 和 MAR 算法提出了基于最大逼近角的 CORDIC 算法理论, 减少计算过程中的迭代次数, 并针对基于 FPGA 的硬件结构加以优化, 易于流水线设计和硬件实现. 任会全等基于最大逼近角对 CORDIC 算法做了改进, 但忽略了对比例常数的优化. 徐成等^[6]提出免缩放因子的 CORDIC 算法, 但增加了算法复杂度和硬件逻辑资源使用, 不利于算法的高速实现. Linbin Chen 等^[7]设计了一种全并行的 CORIDC 算法结构, 大大降低了计算延时, 但计算精度还有待提高. 李杏梅等^[8]提出单向最有迭代的 CORDIC 算法, 节省了硬件资源, 但是其只能计算反正切函数, 具有一定的应用局限性. 目前 CORDIC 算法仍存在算法计算复杂、消耗硬件资源大、精度低、计算速度慢等问题.

本文提出固定序列的 CORDIC 算法将二进制高位的每一位都进行两次旋转, 旋转方向由符号位决定, 相比现有最大逼近角 CORDIC 算法在保证迭代次数不增加的前提下减少了旋转方向判断次数, 进而使比例常数的计算得到简化, 节省硬件资源并降低了算法复杂度.

2 传统 CORDIC 算法实现原理

2.1 原始 CORDIC 算法原理

传统 CORDIC 算法的基本思想是从起点向量 (x, y) 进行矢量旋转后得到向量 (x_n, y_n) , 第一次以角度 θ 旋转后得到的向量为^[1]:

$$\begin{aligned} x_1 &= \cos(\theta)[x - y \tan(\theta)] \\ y_1 &= \cos(\theta)[y + x \tan(\theta)] \end{aligned} \quad (1)$$

令 $\tan\theta = \pm 2^{-i}$, 则式(1)中括号内的乘积项就可以通过移位操作完成, 此时旋转角度值为 $\pm \arctan(2^{-i})$. 则式(1)可转换为:

$$\begin{aligned} x_{i+1} &= K_i[x_i - y_i d_i 2^{-i}] \\ y_{i+1} &= K_i[y_i + x_i d_i 2^{-i}] \end{aligned} \quad (2)$$

其中:

$$K_i = \cos[\arctan(2^{-i})] = \frac{1}{\sqrt{1+2^{-2i}}}$$

$$d_i = \pm 1$$

将每一步的 K_i 相乘可得到最终的比例因子 K , 用公式可表示为:

$$K = \prod_{i=0}^{n-1} K_i = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1+2^{-2i}}} \quad (3)$$

2.2 基于最大逼近角的 CORDIC 算法原理

初始的输入角度可表示为: $\theta = (-b_0) +$

$\sum_{i=1}^B b_i 2^{-i}$, $b_i \in \{0, 1\}$, 并且将角度范围定为:

$\theta \in [-\pi/4, \pi/4]$, 超出该范围的角度可以通过归整的方式将角度旋转到该范围内. 将角度 θ 分解为两部分:

$$\theta = \theta^H + \theta^L = \underbrace{(-b_0)}_{\theta^H} + \underbrace{\sum_{i=1}^{m-1} b_i 2^{-i}}_{\theta^L} + \sum_{i=m}^B b_i 2^{-i} \quad (4)$$

式中,

$$m = \left\lceil \frac{B - \log_2 3}{3} \right\rceil \quad (5)$$

经过 BBR 和 MAR 算法, 最终将 θ^H 转化为 $\arctan(2^{-i})$ 的多项式如下:

$$\begin{aligned} \theta^H &= \sum_{i=1}^m r_i 2^{-i} - 2^{-m} = \\ &\sum_{i=1}^m r_i \left\{ \sum_{j=1}^{n(i)} \arctan(2^{-j}) + e_i \right\} - 2^{-m} \\ r_1 &= 1 - 2b_0; r_k = 2b_{k-1} - 1, k = 2, \dots, m \end{aligned} \quad (6)$$

将 θ^H 转化为 $\arctan(2^{-i})$ 的多项式过程中产生的误差项 $e_1, e_2, \dots, e_m$ 及 θ^H 最后一项 (-2^{-m}) 加到 θ^L 部分, 形成修正后的 θ_c^L 如下:

$$\begin{aligned} \theta_c^L &= \theta^L + \sum_{i=1}^{m-1} r_i e_i - 2^{-m} = \\ &(-b_{m-1})2^{-m-1} + \sum_{k=m}^B b_k 2^{-k} \\ b_k &\in \{0, 1\}, k = m, \dots, B \end{aligned} \quad (7)$$

其中对 θ^H 逼近产生误差的限制条件为:

$$\sum_{i=1}^{m-1} r_i e_i < 2^{-m} \quad (8)$$

在区间 $[-\pi/4, \pi/4]$ 内的 θ 的二进制的每个非零位, 都有: $\arctan(2^{-i}) < 2^{-i} < \arctan(2^{-i+1})$, $i \in [1, B]$, 所以 θ 的二进制的每个非零位 2^{-i} 的最大的逼近角为 $\arctan(2^{-i})$. 当 θ 在区间 $[-\pi/4, \pi/4]$ 内, 并且有 θ 的最高非零位的值为 2^{-i} , 有 $\arctan(2^{-i}) < |\theta| < 3\arctan(2^{-i})$. 因此连续两次的最高非零位的最大逼近角一定可以使原有的最高非零位为零.

为了进一步减少迭代次数, 将 θ^H 划分两部分, 即存在常数 t , 当 i 不小于 t 时, $n(i)$ 等于 1, 此部分 θ^H 记为 θ^{H2} ; 反之, $n(i)$ 大于 1, 此部分 θ^H 记为 θ^{H1} , 其中 t 由下式得到:

$$t = \left\lceil \frac{m + \log_2(m-1) - \log_2 3}{3} \right\rceil \quad (9)$$

该算法优化后, θ^L 部分的迭代次数不变, 而整个 θ^H 部分迭代的次数最多为:

$$\sum n(i) = \underbrace{2 \times (t-1)}_{\theta^{H1}} + \underbrace{(m-t+1)}_{\theta^{H2}} = t + m - 1 \quad (10)$$

3 固定旋转序列的 CORDIC 算法

根据基于最大逼近角的 CODIC 算法理论可知:连续两次的最高非零位的最大逼近角一定可以使原有的最高非零位为零.就此,本文提出了一种固定旋转序列的 CORDIC 算法结构.

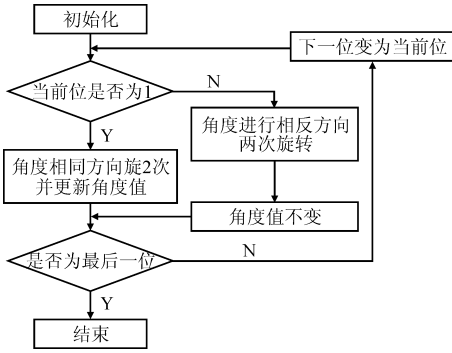


图 1 θ^{H1} 部分旋转策略

本算法对 θ^{H1} 部分每一位都进行两次旋转,使比例系数的计算更加简单,相比原始算法节省了大量存储空间. 因为当 $i = 1, 2, 3, 4 \dots$ 时, 恒有 $2^{-i} > \arctan(2^{-i})$, 所以 2^{-i} 可以用 $\arctan(2^{-i}) + e_i$ 表示, 其中 e_i 为误差项并且 e_i 恒为正值. 为了方便起见, 我们以位宽 $B = 32$ 为例进行说明. 此时, $m = 10, t = 4$, 对于 $i = 1, 2, 3$, 2^{-i} 和 $\arctan(2^{-i})$ 的关系表示如下:

我们将角度 θ 二进制的第 i 位定义为 θ_i , 第 i 位旋转后剩余的角度定义为 θ_c^i , θ_c^i 的符号位定义为 θ_{c0}^i , $i = 0, 1, 2, 3$.

我们以角度 θ 二进制的第 1 位为例进行分析, 此时 $\theta_c^0 = \theta$.

(1) 当 $\theta_1 = 0$ 时, 进行大小相同, 方向相反的两两次旋转. 第一次旋转方向为 $d_1 = 1 - 2\theta_{c0}^0$, 大小为 $\arctan(2^{-1})$, 旋转后剩余角度 $\theta_c^1 = \theta_c^0 - d_1 2^{-1} + d_1 e_1$; 第二次旋转方向与第一次相反, 大小不变, 旋转后剩余角度为 $\theta_c^1 = \theta_c^1 + d_1 2^{-1} - d_1 e_1 = \theta_c^0$. 即该位为 0 时, 进行两次旋转之后, 剩余角度值 θ_c 不变.

(2) 当 $\theta_1 = 1$ 时, 进行大小相同, 方向相同的两次旋转. 旋转方向为 $d_1 = 1 - 2\theta_{c0}^0$, 大小为 $\arctan(2^{-1})$, 两次旋转之后一定可以使第一位变为 0, 剩余角度 $\theta_c^1 = \theta_c^0 - 2d_1 2^{-1} + 2d_1 e_1$. 对于角度 θ 二进制的第 i 位, $i = 2, 3$ 位, 旋转方向 $d_i = 1 - 2\theta_{c0}^{i-1}$, 大小为 $\arctan(2^{-i})$, 旋转规则依 θ_i 而定.

3. 对 θ^{H1} 部分旋转完成之后的剩余角度值 θ_c^3 , 采用 BBR 方法和 MAR 算法对其进行转化, 转化过程如下式:

$$\theta_c^3 = \theta^{H2} + \theta^L$$

$$\theta^{H2} = \sum_{i=4}^{10} d_i 2^{-i} - 2^{-10}$$

$$= \sum_{i=4}^{10} d_i \arctan 2^{-i} + \sum_{i=4}^9 d_i e_i - 2^{-10}$$

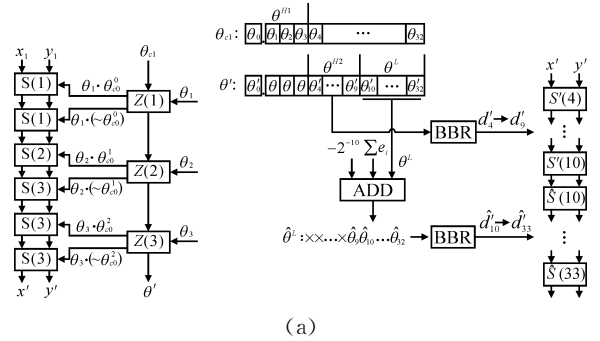
$$d_4 = 1 - 2\theta_3, d_i = (2\theta_i - 1), i = 5, \dots, 10$$

$$\hat{\theta}^L = \theta^L + \sum_{i=4}^9 d_i e_i - 2^{-10} = (-\hat{\theta}_9) 2^{-9} + \sum_{i=10}^{32} \hat{\theta}_i 2^{-i}$$

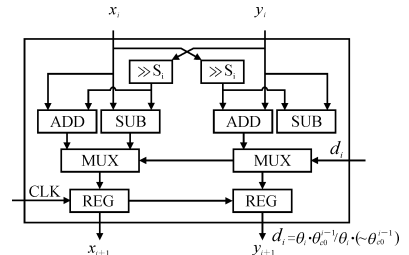
$$= (-\hat{\theta}_9) 2^{-9} + \sum_{i=11}^{33} (2\hat{\theta}_i - 1) 2^{-i} + 2^{-10} - 2^{-33}$$

$$= \sum_{i=10}^{33} r_i 2^{-i} - 2^{-33}$$

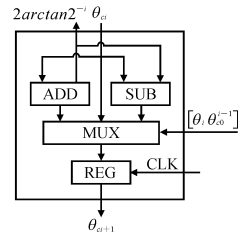
$$r_{10} = 1 - 2\hat{\theta}_9, r_i = (2\hat{\theta}_i - 1), i = 11, \dots, 33$$



(a)



(b)



(c)

图 2 固定旋转序列的 CORDIC 算法结构

优化的 CORDIC 算法计算过程主要有三部分组成, 分别为 θ^{H1} 部分、 θ^{H2} 部分和 θ^L 部分, 三个部分的计算均通过迭代的方式进行, 流水线的方式加以实现. 具体如下:

(1) 根据公式(5)和公式(9)计算出 m 和 t 的值,

确定三部分的计算范围；

(2)在 θ^{H1} 部分,对每一位二进制均进行两次最大逼近角度的旋转,并将产生的误差项传递；

(3)根据 BBR 和 MAR 算法,先判断当前角度正负值,从而确定旋转方向,完成其余部分的角度迭代计算。

对应的流水线结构如图 2 所示,其中(a)为流水线整体结构示意图,(b)为坐标计算模块,(c)为角度值计算模块(b)和(c)两个模块中使用了移位计算、加减法计算、多路选择器模块以及相应的中间寄存器,BBR 算法的功能是计算旋转方向并且和坐标的计算进行同步.整体的计算过程可以看做坐标值得迭代计算和剩余角度值的迭代计算,其中角度值的计算目的是确定当前位的旋转方向。

4 优化后算法性能分析

4.1 旋转方向判断次数分析

迭代计算过程中的旋转方向判断次数体现了算法整体的复杂度。

优化前的最大逼近角 CORDIC 算法除了需要判断当前角度值符号之外,还需要判断旋转一次后当前位是否为 0,因此在 θ^{H1} 部分的旋转判断次数为 $n = 2(t - 1)$ 次。

优化后固定旋转序列的 CORDIC 算法由于只需要判断当前剩余角度值的正负值,即可确定当前位的旋转方向,因此只需要进行的判断次数为 $t - 1$ 次,记为: $n = t - 1$ 。

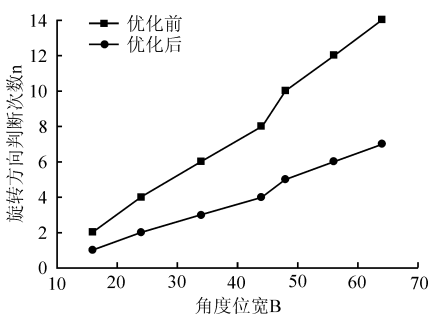


图 3 优化前后 θ^{H1} 部分旋转方向判断次数对比

4.2 存储空间使用分析

基于最大逼近角的 CORDIC 算法存储空间使用数量只由比例常数 K 决定.由于在 θ^{H1} 部分增加了额外的判断过程,即比常数 K_1 在 θ^{H1} 部分会有 $2^{2(t-1)}$ 种可能,因此在每一个角度位宽 B 下都会占用 $2^{2(t-1)}$ 个存储空间对比比例常数 K 进行存储.因此 $N = 2^{2(t-1)}$ 。

固定旋转序列的 CORDIC 算法存储空间由两

部分组成,分别是比例常数 K 和 θ^{H1} 部分误差项 e_i 。

将 θ^{H1} 部分比例系数定义为 K_1 .因为我们将 θ_i 每一位都进行了大小相同的两次旋转所以 K_1 的计算公式如下：

$$K_1 = \prod_{i=1}^{t-1} [(1 + 2^{-2i})^{-\frac{1}{2}}]^2 = \prod_{i=1}^{t-1} (1 + 2^{-2i})^{-1} \quad (11)$$

剩余部分比例系数定义为 K_2 ,根据 BBR 方法和 MAR 算法可知：

$$K_2 = \prod_{i=t}^m (1 + 2^{-2i})^{-\frac{1}{2}} \times \prod_{i=m}^B (1 + 2^{-2i})^{-\frac{1}{2}} \quad (12)$$

流水线的总的比例系数 K 为：

$$K = K_1 \times K_2 \quad (13)$$

因此可以得出结论:固定旋转序列的 CORDIC 算法对于确定角度二进制位宽 B ,只需要存储一个固定的比例常数 K 即可。

误差项 e_i 的存储数量由 t 决定,且数量为 $t - 1$ 。

因此,固定旋转序列的 CORDIC 算法所需的存储空间 $N = t$ 。

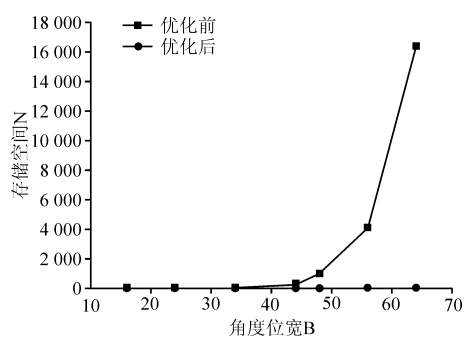


图 4 优化前后存储空间使用对比

4.3 仿真测试与结果分析

本小节基于 Quartus II 12.1 软件使用 Verilog HDL 语言进行编程.在 Altera 公司生产的 Cyclone IV 系列 EP4CE115F29I7 芯片以及未作资源约束条件下以角度位宽 $B=16$ 为例进行测试,并与 QuartusII12.1 软件自带的 CORDIC 算法 IP 核综合结果进行比较。

表 1 综合后资源消耗对比

CORDIC 算法	逻辑单元	寄存器	时钟数
IP 核	2 723	2 551	16
最大逼近角	2354	2128	19
固定旋转序列	1982	1336	15

由表 1 计算可知,固定旋转序列的 CORDIC 算法相比 QuartusII12.1 自带 IP 核 CORDIC 算法逻辑单元使用数量降低 27.2%。寄存器使用量降低 47.7%；相比于最大逼近角 CORDIC 算法逻辑单元数量降低 15.8%，寄存器使用量降低 37.2%。同时在计算时所需时钟数量有所降低.因此,固定旋转序

列的 CORDIC 算法达到了节省硬件资源以及加快计算速度的优化效果。

通过使用 Verilog HDL 编程后,在 modelsim 中对改进后的算法代码进行波形仿真,测试结果如图 5 所示。由图中可以看出正余弦波形在若干个周期中均保持稳定,没有出现波形失真。

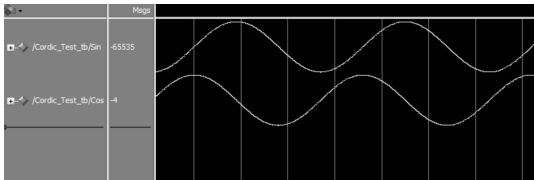
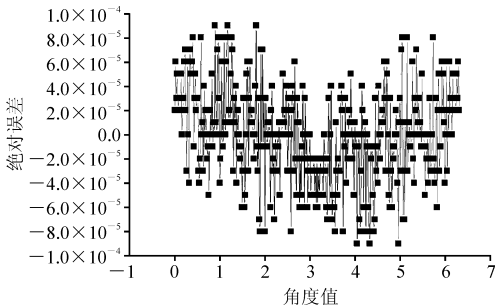


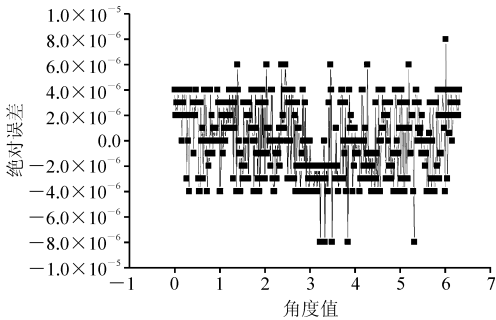
图 5 modelsim 仿真波形

将同等位宽 $B=16$ 情况下,基于最大逼近角的 CORDIC 算法和固定旋转序列的 CORDIC 算法得到的一个周期正弦波形数据(角度分辨率为 1°)分别与 Matlab 三角函数生成的标准数据比较后发现改进后的算法误差控制良好。

基于最大逼近角的 CORDIC 算法产生的三角函数值绝对误差的范围是 $-1.0 \times 10^{-4} \sim 1.0 \times 10^{-4}$,如图 6(a)所示。而固定旋转序列的 CORIDC 算法产生的三角函数值绝对误差的范围是 $-8.0 \times 10^{-6} \sim 8.0 \times 10^{-6}$ 如图 6(b)所示。相比之下提升了一个数量级的精度,表明优化后算法提高了计算精度,完全可以满足使用过程中的高精度要求。



(a)



(b)

图 6 误差分析

5 结束语

本文基于最大逼近角算法理论,确定了对角度值高位的旋转序列加以固定,均进行两次旋转的固定旋转序列的方法。通过设计算法流水线结构,以 16 位为例在 QuartusII 12.1 及 modelsim10.1c 环境下进行仿真实验,从实验结果可以看出固定旋转序列的 CORDIC 算法在进一步降低了计算复杂度、节省了大量存储空间的使用。同时在 FPGA 中实现了逻辑资源和寄存器使用量减少以及误差控制等方面有明显的优势,非常适合高速高精的数字信号处理领域使用。

参考文献:

[1] VOLDER, JACK E . The CORDIC Trigonometric Computing Technique [J]. Electronic Computers, IRE Transactions on, 1959,8(3):330-334.

[2] WANG S , PIURI V , WARTZLANDER E E . Hybrid CORDIC algorithms[J]. IEEE Transactions on Computers, 1997,46(11):1202-1207.

[3] HSIAO S F , HU Y H , JUANG T B . A Memory-Efficient and High-Speed Sine/Cosine Generator Based on Parallel CORDIC Rotations[J]. IEEE Signal Processing Letters, 2004,11(2):152-155.

[4] 王建庄. 基于 FPGA 的高速图像处理算法研究及系统实现[D]. 武汉:华中科技大学,2011.

[5] 任全会,吕春峰. 一种优化的最大逼近角的 CORDIC 算法改进[J]. 微电子学与计算机,2013,30(9):50-52.

[6] 徐成,秦云川,李肯立,等. 免缩放因子双步 CORDIC 旋转算法[J]. 电子学报,2014,7(7):1441-1445.

[7] CHEN L , LOMBARDI F , HAN J ,et al . A fully parallel approximate CORDIC design[C]// 2016 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH). IEEE,2016:197-202.

[8] 李杏梅,付裕,高翟,等. $\pi/4$ 单向最优迭代 CORDIC 算法的反正切计算[J]. 华中科技大学学报(自然科学版),2019,47(8):29-33.

作者简介:

王瑜 男,(1993-),硕士. 研究方向为 FPGA 高速图像处理系统. E-mail:787388622@qq.com

蒋平 男,(1981-),博士,教授,博士生导师. 研究方向为激光及其复合能场焊接机理、工艺与装备。

王建庄 男,(1982-),博士,讲师. 研究方向为 FPGA 图像处理、机器视觉。