

## 高速高精度固定角度旋转 CORDIC 算法的设计与实现

张朝柱, 韩吉南, 燕慧智

(哈尔滨工程大学, 黑龙江哈尔滨 150001)

**摘 要:** 固定角度旋转的 CORDIC (Coordinate Rotation Digital Computer) 算法已经广泛的应用于高速数字信号处理、图像处理、机器人学等领域. 针对固定角度旋转 CORDIC 算法在相位旋转过程中, 存在数据吞吐率较高、占用硬件资源较多且资源消耗量大等缺点, 提出了利用混合 CORDIC 算法, 将角度旋转分为单向角度旋转和一次角度估计旋转两部分. 本文根据欠阻尼理论, 将固定角度旋转采用单向旋转 CORDIC 算法实现, 减少了流水线的级数和迭代符号位的判决, 然后通过对角度估计旋转的二进制表示, 修正常数因子, 再根据角度映射关系进行相关处理, 完成高速高精度坐标旋转. 最后在硬件平台上进行了仿真实验. 实验结果表明, 在误差范围一定的前提下, 混合算法进一步的减少了迭代次数, 并且资源消耗较低, 提高了数据吞吐率.

**关键词:** 坐标旋转数字算法; 固定角度旋转; 流水线结构; 现场可编程门阵列

**中图分类号:** TN713.7; TN431.2      **文献标识码:** A      **文章编号:** 0372-2112 (2016)02-0485-06

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.3969/j.issn.0372-2112.2016.02.033

## Design and Implementation of CORDIC Algorithm for High Speed and Precision Fixed Angle of Rotation

ZHANG Chao-zhu, HAN Ji-nan, YAN Hui-zhi

(School of Information and Communication Engineering, Harbin Engineering University, Harbin, Heilongjiang 150001, China)

**Abstract:** Coordinate Rotation Digital Computer (CORDIC) algorithm for fixed angle of rotation has been widely applied in many fields, including digital signal processing, image processing and robotics. CORDIC algorithm, based on fixed angle of rotation, has an advantage of high data throughput, but it takes up much more hardware resources. To remedy these problems, a hybrid CORDIC algorithm is proposed in this paper, which separates the angle rotation into two steps, unidirectional angle rotations and angle estimation ones. According to the under-damping theory, the unidirectional phase rotations are employed to realize the fixed angle of rotation, which reduces the stage of pipeline and judgment of sign bit. Through the binary representation, the constant factor will be fixed. According to the phase mapping relationship, the high speed and precision fixed angle of rotation is completed. Finally the hybrid algorithm is implemented on a hardware platform and the experimental results show that it can further reduce the number of iterations and resource consumption and obtain high throughput rate.

**Key words:** CORDIC (Coordinate Rotation Digital Computer); fixed angle rotation; pipeline architecture; FPGA (Field-Programmable Gate Array)

### 1 引言

Volder 提出的坐标旋转数字计算方法 (CORDIC) 以其运算及硬件结构简单和多能性而倍受关注, 并已广泛应用于信号处理领域. 针对其实现的硬件结构, 国内外学者也进行了大量研究<sup>[1-4]</sup>. 在近期的研究工作中, 乘法器的设计与实现成为了其在数字信号处理领域的一个新热点<sup>[5-8]</sup>. 但是对于固定相位旋转的设计与实现

的研究较少.

固定旋转角度的 CORDIC 算法已经广泛的应用在高速数字信号处理、机器人学、动画片等领域. 在高速数字信号处理领域, 数字控制振荡器 NCO (Numerical Controlled Oscillator), 是数字下变频的重要组成部分. 机器人的平行移动通过小角度的旋转, 从而完成坐标值的连续叠加<sup>[6]</sup>. 计算机图形中关键帧的插入可以采用固定旋转角度的 CORDIC 旋转<sup>[8]</sup>.

为了解决 CORDIC 算法中由于迭代次数较多,从而处理时间增加的问题,目前国内外一些学者致力于减少算法的延时.文献[9]提出了一种基于控制理论的任意角度旋转方法,但是以迭代次数的增加为代价.在文献[10,11]中,由于每一次迭代会产生更多的位数,高进制数的 CORDIC 算法减少了迭代次数,然而  $S_n$  的判定增加了循环迭代的时间.文献[12]提出了预测旋转方向,并针对误差进行分析和消除,具有速度快等优点,但是并没有给出使用各种硬件结构实现此种算法的效果,更没有将实验结果进行比对.文献[13]提出了基于相位旋转估计的改进混合式 CORDIC 算法,提高了输出数据精度,但是迭代次数依然较多,并且硬件资源的一半以上完成的是移位寄存器的操作.

## 2 传统 CORDIC 算法

1959 年, Voder 提出 CORDIC 算法, 1971 年 Walther 统一了该算法的形式, Meyer-base 第一次利用 FPGA 实现了该算法<sup>[14]</sup>. 其基本思想是从起始点  $(x_i, y_i)$  按照一定的相位不断进行矢量旋转, 逐步逼近最终点  $(x_j, y_j)$ . 其向量旋转示意图如图 1 所示.

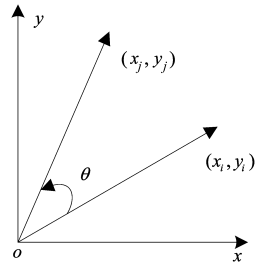


图1 CORDIC向量旋转示意图

根据极坐标知识和图 1 易得:

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

从起始位置到终点位置的旋转过程可以通过多步来实现, 每步只旋转一定相位, 则:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n \\ \sin\theta_n & \cos\theta_n \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (2)$$

把  $\cos\theta_n$  提取出来, 式(2)可重写为:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -\tan\theta_n \\ \tan\theta_n & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (3)$$

为了便于硬件实现、简化运算, 每次旋转的相位设为  $\theta_n = \arctan(2^{-n})$ , 并需要保证总的旋转相位  $\theta = \sum S_n \theta_n$ , 其中  $S_n = \pm 1$ . 这样就有:  $\tan\theta_n = S_n 2^{-n}$ , 代入式(3)可得:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (4)$$

$\cos\theta_n$  可以通过在最终结果上乘一个已知的常数  $K$  来加以消除. 例如, 当迭代次数  $P=16$  时, 且  $|\theta| \leq \pi/4$ , 有:

$$\begin{aligned} K &= \prod_{n=1}^{16} \cos\theta_n = \prod_{n=1}^{16} \cos(\arctan(2^{-n})) \\ &= \prod_{n=1}^{16} (1 - 2^{-2n})^{-1/2} \end{aligned} \quad (5)$$

所以, 在相位旋转过程中,  $\cos\theta_n$  项可以不予考虑, 最后得到旋转迭代公式为:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \quad (6)$$

引入参数  $z$ , 用来判断迭代何时结束,  $z_{n+1} = z_n - \theta_n$ ,  $z_0 = \theta$ . 这样当  $z_n < \theta$  时,  $S_n = -1$ ; 当  $z_n \geq \theta$  时,  $S_n = +1$ . 如果  $(x_i, y_i) = (x_0, y_0) = (K, 0)$ ,  $(x_n, y_n)$  第  $P$  次迭代将收敛于  $(\cos\theta, \sin\theta)$ . CORDIC 算法的收敛性满足收敛定理<sup>[15]</sup>. 对传统 CORDIC 算法的计算精度进行全面的分析后, 得到 CORDIC 算法的迭代次数  $P$  和相位精度  $\Delta\theta_{\min}$  之间的关系式:

$$P \geq -\log_2[\tan(\Delta\theta_{\min})] \quad (7)$$

式中,  $\Delta\theta_{\min} = 2\pi/2^N$ ,  $N$  为输入相位数据 bit 宽度.

## 3 单向固定角度旋转 CORDIC 算法

### 3.1 单向旋转 CORDIC 算法

根据传统的 CORDIC 算法, 迭代  $P$  次后的剩余角度值为:

$$Ang_p = Ang - \sum_{n=0}^p S_n \arctan(2^{-n}) \quad (8)$$

由于  $\arctan(2^{-n})$  为  $n$  的单调递减函数, 本文用比较典型的二阶控制系统作为理论模型来讨论 CORDIC 算法的收敛方式. 一个典型的二阶控制系统的传递函数设为  $G(s) = \frac{K}{S(S+T)}$ , 其闭环的阶跃信号响应为:

$$S(t) = \begin{cases} 1 - \frac{1}{\beta} e^{-\xi w_n t} \sin(w_n \beta t + \phi), & \xi < 1 \\ 1 - (1 + w_n t) e^{-w_n t} \sin(w_n \beta t + \phi), & \xi = 1 \\ 1 - \frac{1}{\beta} e^{-\xi w_n t} \sinh(w_n \beta t + \psi), & \xi > 1 \end{cases} \quad (9)$$

这里,  $\xi = \frac{p}{2\sqrt{K}}$ ,  $w_n = \sqrt{M}$ ,  $\beta = \sqrt{|1 - \xi^2|}$ ,  $\phi = \arctan(\beta/\xi)$ ,  $\psi = \operatorname{arctanh}(\beta/\xi)$ . 在  $\xi \geq 1$  时, 二阶系统的瞬态过程具有单调上升的特性. 如果 CORDIC 算法的收敛特性为临界阻尼或者过阻尼, 那么其收敛特性较欠阻尼更快. 单向旋转 CORDIC 算法将旋转方向限制在  $S_n = +1, 0$  或  $S_n = -1, 0$ . 本文将旋转方向限制在  $S_n = +1, 0$ . 例如, 当旋转角度为  $23.345^\circ$  时, 单向旋转 CORDIC 算法的旋转过程为:

$$23.345 = 14.03624 + 7.12502^\circ + 0^\circ + 1.78991^\circ + 0^\circ$$

$$+0.44716^\circ + 0.22381^\circ + 0.11191^\circ \\ \approx 23.34284$$

图 2 给出了传统的 CORDIC 算法与单向旋转 CORDIC 算法的角度收敛曲线图,初始相位为  $23.345^\circ$ 。图 3 和图 4 给出了单向旋转 CORDIC 算法中不同的旋转角度所需迭代次数的仿真图,相位的位数 16 位,角度从  $0.00699^\circ \sim 90^\circ$ ,样本数为 899721。

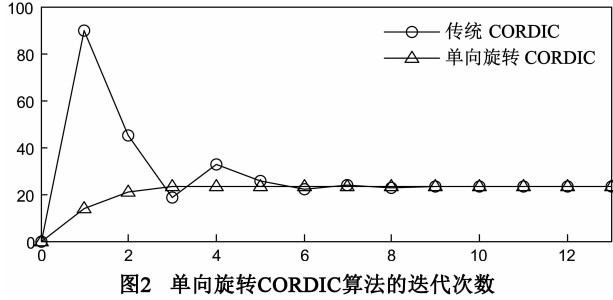


图2 单向旋转CORDIC算法的迭代次数

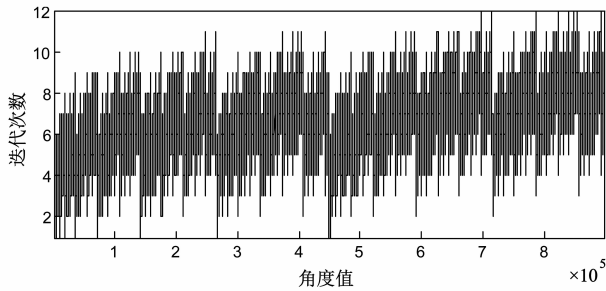


图3 单向旋转CORDIC算法的迭代次数

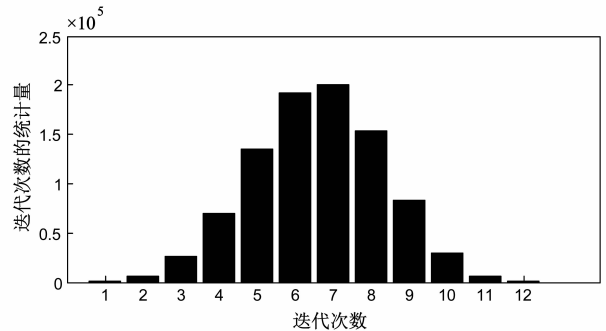


图4 单向旋转CORDIC算法的迭代次数分布

由仿真结果可知,与传统的基于流水线结构的CORDIC 算法相比,平均迭代次数减少了 29.41%。

### 3.2 一次角度估计的单向旋转 CORDIC 算法

假设 CORDIC 算法的输入相位长度为  $N$ , 迭代次数为  $P$ , 则旋转相位  $\theta$  可表示为:

$$\theta = \sum_{n=0}^{P-1} S_n \theta_n = \sum_{n=0}^{P-1} S_n \arctan(2^{-n}) \quad (10)$$

式中,  $S_n = \pm 1$ 。

随着旋转系数  $n$  的增加,  $\arctan(2^{-n})$  逼近  $2^{-n}$ , 根据文献[12], 当  $n \geq m = \lceil (N - \log_2 3) / 3 \rceil$ , 满足  $\arctan$

$(2^{-n}) \approx 2^{-n}$ 。通过式(7)可知 CORDIC 算法的迭代次数为  $P \geq N - 2$ , 取  $P = N - 2$ 。式(10)可重新表示为:

$$\theta = \sum_{n=0}^{N-3} S_n \theta_n = \sum_{n=0}^{m+1} S_n \arctan(2^{-n}) + \sum_{n=m+2}^{N-3} S_n 2^{-n} \quad (11)$$

迭代  $m+2$  次后, 剩余旋转相位总和为  $\sum \theta_n$ ,

$$\sum \theta_n = \sum_{n=m+2}^{N-3} S_n 2^{-n} = S_{m+2} 2^{-m-2} + S_{m+3} 2^{-m-3} + \dots \\ + S_{N-3} 2^{-N+3} < 2^{-m-1} \quad (12)$$

此时, 实际剩余相位为  $z_{m+2}$ , 按照传统 CORDIC 算法的理论,  $z_{m+2} \approx \sum \theta_n$ , 由此得  $z_{m+2} < 2^{-m-1}$ 。在第  $m+3$  次旋转时, 引入新的旋转相位  $\tilde{\theta}_{m+2} = \phi_{m+2}$ ,  $\phi_{m+2}$  是  $z_{m+2}$  的绝对值,  $\tilde{\theta}_{m+2} < 2^{-m-1}$ 。为了防止旋转角度超过  $(-\pi/2, \pi/2)$ , 此算法做了如下判断:  $\tan \tilde{\theta}_{m+2} = \tilde{S}_{m+2} \tilde{\theta}_{m+2}$ , 其中当  $z_{m+2} < 0$  时,  $\tilde{S}_{m+2} = -1$ ; 当  $z_{m+2} \geq 0$  时,  $\tilde{S}_{m+2} = +1$ 。带入式(3)可得:

$$\begin{bmatrix} x_{m+3} \\ y_{m+3} \end{bmatrix} = \cos \phi_{m+2} \begin{bmatrix} 1 & -\tilde{S}_{m+2} \tilde{\theta}_{m+2} \\ \tilde{S}_{m+2} \tilde{\theta}_{m+2} & 1 \end{bmatrix} \begin{bmatrix} x_{m+2} \\ y_{m+2} \end{bmatrix} \quad (13)$$

本次旋转完之后, 剩余相位为 0, 过程结束。  $N$  位输入相位经过  $m+2$  次迭代得到一个  $N$  位 2 进制数  $W$ , 即此时的剩余相位  $z_{m+2}$ , 而  $\tilde{\theta}_{m+2}$  也由  $N$  位 2 进制数表示为:

$$T = \sum_{i=0}^{N-1} A_i 2^i = A_{N-1} 2^{N-1} + A_{N-2} 2^{N-2} + \dots + A_0 2^0 \quad (14)$$

式中,  $A_i = 1$  或 0; 当  $W[N-1] = 0$  时,  $A_i = W[i]$ , 当  $W[N-1] = 1$  时,  $A_i$  为  $W$  的每一位值取反最后加一。

由相位实际值和它的  $N$  位二进制表示值的比例关系, 可得:

$$\tilde{\theta}_{m+2} = 2\pi T / 2^N = \tilde{T} / 2^N \quad (15)$$

从式(15)可见, 当  $\tilde{T}$  确定后, 易得  $\tilde{\theta}_{m+2}$ 。在硬件实现中, 需要对  $\tilde{T}$  进一步进行如下处理,

$$\tilde{T} = \sum_{i=0}^N B_i 2^i = 2\pi T = \sum_{i=0}^N A_i 2^i (2^2 + 2^0 + 2^{-2} + 2^{-5} + 2^{-9}) \quad (16)$$

式中,  $B_i = 1$  或 0。

得到  $\tilde{T}$  各位值后, 带入式(15)。考虑到  $\tilde{\theta}_{m+2} < 2^{-m-1}$ , 进一步简化得:

$$\tilde{\theta}_{m+2} = \sum_{i=0}^{N-m-2} B_i 2^{i-N} = B_{N-m-2} 2^{-m-2} + \dots + B_1 2^{1-N} + B_0 2^{-N} \quad (17)$$

联立式(13)和式(17)得:

$$\begin{bmatrix} x_{m+3} \\ y_{m+3} \end{bmatrix} = \begin{bmatrix} x_{m+2} - \tilde{S}_{m+2} \cdot \sum_{i=0}^{N-m-2} B_i 2^{i-N} \cdot y_{m+2} \\ \tilde{S}_{m+2} \cdot \sum_{i=0}^{N-m-2} B_i 2^{i-N} \cdot x_{m+2} + y_{m+2} \end{bmatrix} \quad (18)$$

式中,  $\tilde{S}_{m+2} = \pm 1, B_i = 1$  或 0。

由上可看出, 一次相位估计 CORDIC 算法将传统的

旋转相位集  $\{\arctan 2^0, \arctan 2^{-1}, \dots, \arctan 2^{-m-1}, \dots, \arctan 2^{-p}\}$  简化为  $\{\arctan 2^0, \arctan 2^{-1}, \dots, \arctan 2^{-m-1}, \bar{\theta}_{m+2}\}$ . 虽然增加一次移位加法运算, 但是与传统的 CORDIC 算法相比, 减少了  $N - m - 3$  次旋转.

最后, 随着流水线级数的减少, 常数系数  $K$  需要重新规定, 即:

$$\hat{K} = \prod_{n=0}^{m+1} \cos(\arctan(2^{-n})) \quad (19)$$

此时把初始值设为:  $(x_0, y_0) = (\hat{K}, 0)$ , 按照上述过程, 最后  $(x_{m+3}, y_{m+3})$  将收敛为  $(\cos\theta, \sin\theta)$ . 由于是固定角度旋转, 所以  $\hat{K}$  值可以根据所要旋转的角度值预先计算出迭代次数和每一级的旋转角度值. 图 5 和图 6 给出了基于旋转相位估计的单向旋转 CORDIC 算法中, 不同的旋转角度与迭代次数的仿真图, 相位的有效位数 14 位, 角度从  $0.00699^\circ \sim 90^\circ$ , 样本数为 899931.

由仿真结果可知, 迭代次数同样近似服从高斯分布, 与传统的基于流水线结构的 CORDIC 算法相比, 平均迭代次数至少减少了 75.82%.

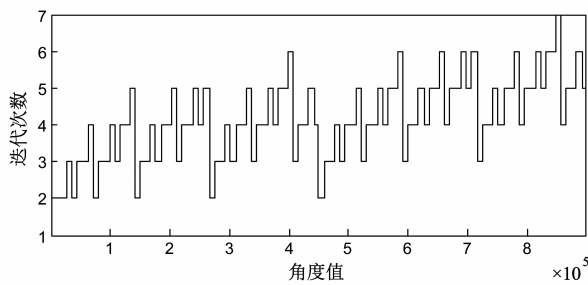


图5 基于旋转相位估计的单向旋转CORDIC算法的迭代次数

#### 4 固定角度旋转的硬件设计与实现

本文实现高速高精度固定角度的CORDIC算法的

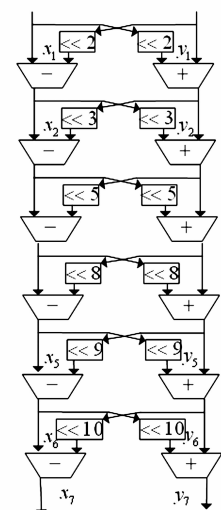


图8 单向旋转CORDIC算法的硬件实现

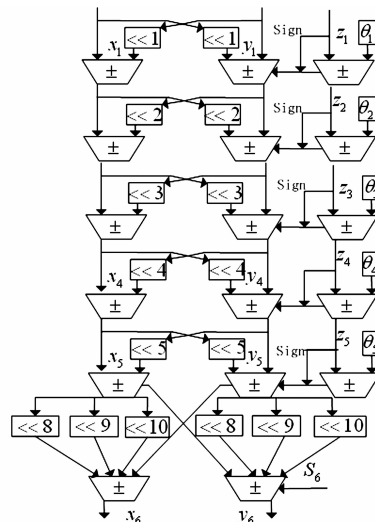


图9 相位旋转估计CORDIC算法的硬件实现

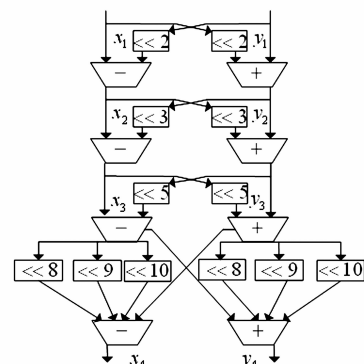


图10 本文提出的固定旋转CORDIC算法的硬件实现

硬件结构图如图 7 所示. 表 1 为固定角度旋转 CORDIC 算法输出端的象限映射表.

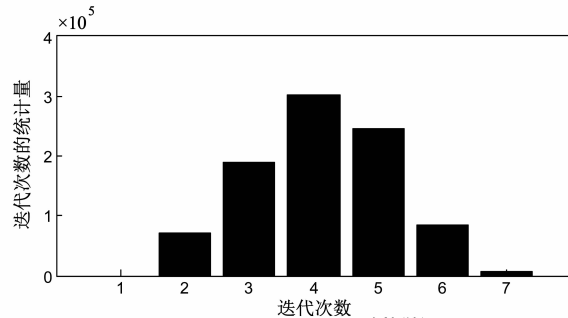


图6 基于旋转相位估计的单向旋转CORDIC算法的迭代次数统计量

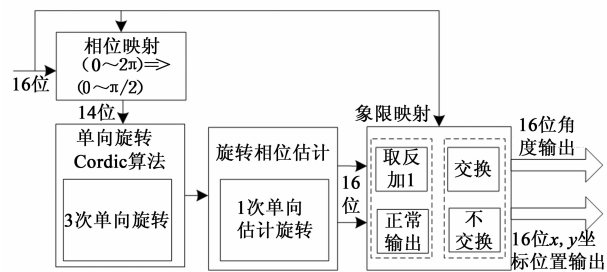


图7 固定角度旋转CORDIC算法硬件结构图

图 8 ~ 图 10 分别给出了三种 CORDIC 算法的硬件结构, 表 2 对三种算法的硬件结构进行了对比. 基于相位估计的单向旋转 CORDIC 算法采用高效流水线结构, 需要 3 次单向相位旋转, 1 次估计相位旋转. 在高速高精度的数控振荡器模块中, 设定每次固定旋转角度为  $23.345^\circ$ , 可知单向旋转次数为三次, 通过 3.2 节的计算可知,  $\bar{\theta}_3 = 2^{-8} + 2^{-9} + 2^{-10}$ , 最后一级流水线单元只需 3 个加减法器, 6 个移位寄存器和一个相位系数存储器, 且无需旋转方向判定.

表 1 固定角度旋转 CORDIC 算法输出端的象限映射表

相位的最高三位	相位操作	相位范围	正弦输出	余弦输出
00	不变	$(0, \pi/4)$	$y$	$x$
01	减去 $14'h1000$	$(\pi/4, \pi/2)$	$x$	$y$ 补
10	减去 $14'h2000$	$(\pi/2, 3\pi/4)$	$y$ 补	$x$ 补
11	减去 $14'h2FFF$	$(3\pi/4, \pi)$	$x$ 补	$y$

表 2 硬件结构比较

算法	数据宽度(位)		
	移位寄存器	是否需要符号位判决	迭代次数
单向旋转 CORDIC 算法	12	否	6
相位估计 CORDIC 算法	16	是	6
相位估计的单向旋转 CORDIC 算法	12	否	4

### 5 系统仿真与验证

本文根据图 8~图 10 所示硬件结构,在不同相位宽度的条件下,将本文算法与现有的 CORDIC 算法进行资源使用量,旋转相位后的相位误差、坐标位置误差进行比较.硬件描述语言采用 Verilog 语言,基于 EP2C8Q208C8 芯片和 QuartusII 软件平台搭建模块进行系统级仿真.

设定每次固定旋转角度为  $23.345^\circ$ .将本文所提出的 CORDIC 算法与其他几种算法的使用资源进行对比,如图 11、12 所示.通过图 11、12 可知,各种算法的资源使用量都是随着相位宽度的增加而增加;在相同的相位宽度下与其他几种 CORDIC 算法相比,相位估计的单向旋转 CORDIC 算法资源消耗量最少.

将文所提出的 CORDIC 算法与其他几种算法的固定角度旋转理论值与实际值进行比较,误差统计结果如表 3 所示.通过比较可知,在相同的数据宽度下,相位估计的单向旋转 CORDIC 算法输出的坐标位置误差最大.但是随着数据宽度的增加它的误差范围将被缩小,当数据宽度增加到 18 以上时,坐标位置误差将被控

表 3 坐标位置误差值比较

算法	数据宽度(位)									
	14		16		18		22		24	
	X	Y	X	Y	X	Y	X	Y	X	Y
传统流水线结构 CORDIC 算法	2.97e-4	8.31e-4	-4.16e-5	5.87e-5	2.12e-6	5.12e-6	4.18e-7	2.32e-6	-1.23e-7	2.47e-07
单向旋转 CORDIC 算法	2.47e-4	1.28e-4	8.05e-5	-2.38e-6	6.35e-5	6.13e-6	-9.16e-7	8.86e-5	2.98e-6	8.37e-9
相位估计 CORDIC 算法	5.86e-3	9.53e-3	1.94e-5	-3.69e-4	4.78e-4	3.53e-4	1.67e-4	3.58e-4	-2.15e-6	1.28e-5
本文算法	6.47e-3	1.23e-4	5.38e-4	-2.99e-3	2.85e-4	5.28e-4	2.87e-4	5.32e-4	7.39e-6	-2.85e-5

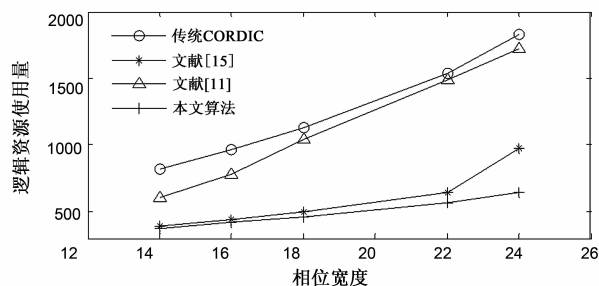


图 11 各种CORDIC算法逻辑资源使用比较

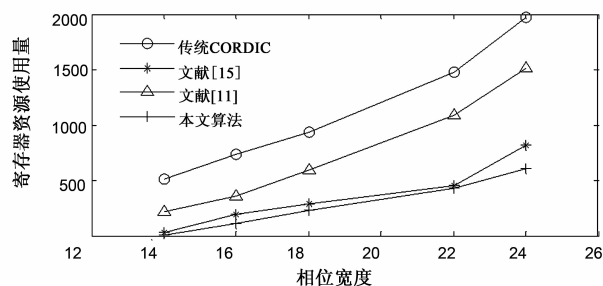


图 12 各种CORDIC算法寄存器资源使用比较

制在  $(-6 \times 10^{-4}, 6 \times 10^{-4})$  内,说明本文的所提出的算法在减少迭代次数的基础上,保证了输出正余弦数据的精度.

### 6 结束语

针对固定角度旋转 CORDIC 算法在相位旋转过程中,存在数据吞吐率较高、占用硬件资源多且资源消耗量大等缺点,提出了利用混合 CORDIC 算法.将本文算法与现有的 CORDIC 算法进行资源使用量、旋转相位后的相位误差、坐标位置误差进行比较,实验数据表明,当数据宽度增加到 18 位以上时,该算法将误差控制在  $(-6 \times 10^{-4}, 6 \times 10^{-4})$ ,迭代次数、逻辑资源使用量以及寄存器资源使用量较现有的算法有了较为明显的减少.

## 参考文献

- [1] Pramod Kumar Meher, SangYoon Park. CORDIC designs for fixed angle of rotation[J]. IEEE transactions on very large scale integration (VLSI) systems, 2013, 21(2): 217 - 227.
- [2] Shen-Fu Hsiao, Yu-Hen Hu, Tso-Bing Juang. A memory-efficient and high-speed sine/cosine generator based on parallel CORDIC rotations[J]. IEEE signal processing letters, 2004, 11(2): 152 - 155.
- [3] C-S Wu, A-Y Wu, C-H Lin. A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes[J]. IEEE Trans Circuits Syst II, Analog Digit. Signal Process, 2003, 50(9): 589 - 601.
- [4] T-B Juang, S-F Hsiao, M-Y Tsai. Para-CORDIC: Parallel cordic rotation algorithm[J]. IEEE Trans Circuits Syst I, Reg Papers, 2004, 51(8): 1515 - 1524.
- [5] Zhang G, Chen, F Parallel FFT with CORDIC for ultra wide band[A]. In Proc of 15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications[C]. Barcelona: IEEE, 2004. 1173 - 1177.
- [6] P K Meher, J K Satapathy, G Panda. Efficient systolic solution for a new prime factor discrete Hartley transform algorithm[J]. IEE Proc Circuits, Devices Syst, 1993, 140(2): 135 - 139.
- [7] 张晓彤, 辛茹, 王沁, 李涵. 基于改进混合式 CORDIC 算法的直接数字频率合成器设计[J]. 电子学报, 2008, 36(6): 1144 - 1148.  
Zhang Xiao-tong, et al. Design of direct digital frequency synthesizer based on improved hybrid CORDIC algorithm[J]. Acta Electronica Sinica, 2008, 36(6): 1144 - 1148. (in Chinese)
- [8] T Lang, E Antelo. High-throughput CORDIC-based geometry operations for 3D computer graphics[J]. IEEE Trans. Computer, 2005, 54(3): 347 - 361.
- [9] S Wang, E E Swartzlander, Jr. Critically damped CORDIC algorithm[A]. The 37th Midwest Symp Circuits and Systems[C]. Lafayette, LA: IEEE, 1994. 236 - 239.
- [10] Kaushik B, Rakesh B. Architectural design and FPGA implementation of radix-4 CORDIC processor[J]. Microprocessors and Microsystems, 2010, 34(2-4): 96 - 101.
- [11] E Antelo, J Villalba, J D Bruguera, E L Zapata. High performance rotation architectures based on the radix-4 CORDIC algorithm[J]. IEEE Trans Computers, 1997, 46(8): 855 - 870.
- [12] Wang S, Piuri V, et al. Hybrid CORDIC algorithms[J]. IEEE Transaction on Computers, 1997, 46(11): 1202 - 1207.
- [13] Chaozhu Zhang, Jinan Han, Ke Li. Design and implementation of hybrid CORDIC algorithm based on phase rotation estimation for NCO[J/OL]. The Scientific World Journal, <http://www.hindawi.com/journals/tswj/2014/897381>, 2014-7-7.
- [14] J Volder. The CORDIC trigonometric computing technique[J]. IRE Trans on Electronic Computers, 1959, 8: 330 - 334.
- [15] J Walther. A unified algorithm for elementary functions[A]. Joint Computer Conference[C]. Atlantic: Spring, 1971. 379 - 385.

## 作者简介



张朝柱 男, 1970 年生于辽宁省大连市, 博士生导师, 研究方向为信号处理在雷达、通信中的应用研究; 雷达对抗技术研究。  
E-mail: zhangchaozhu@hrbeu.edu.cn



韩吉南 (通信作者) 男, 1988 年生于黑龙江省牡丹江市, 男, 博士研究生, 研究方向为导航卫星信号处理算法。  
E-mail: hanjinanbuaa@163.com