

DOI:10.11918/j.issn.0367-6234.201704019

低消耗免查找表CORDIC算法

姚亚峰, 邹凌志, 王巍, 钟梁

(中国地质大学(武汉)机械与电子信息学院, 武汉 430074)

摘要:为减少传统流水线型CORDIC(Coordinate Rotation Digital Computer)算法的硬件资源消耗和输出时延,在包含查找表的三阶段CORDIC算法实现基础上,提出一种免去查找表环节的CORDIC算法实现方法.提出的改进算法直接使用四次移位相加的迭代运算替换查找表结构从而显著降低寄存器消耗,同时通过合并迭代降低迭代次数进而有效减少最大输出时延,并综合运用角度二极化重编码(Binary To Bipolar Recoding, BBR)方法和角度区间折叠技术保证了输出精度.使用Verilog HDL语言在ISE14.2软件平台上对三种算法进行具体实现,利用XST工具对其进行综合,并通过MATLAB建模计算得到算法的正余弦值输出误差.仿真实验结果表明:在输出位宽均设置为16位的情况下,免查找表CORDIC算法能够有效地输出正余弦值;与传统流水线型算法相比,免查找表算法的寄存器资源消耗减少大约74.42%,计算所需的时钟周期降低68.75%,其输出精度也有明显改善;与三阶段算法相比,免查找表算法的寄存器消耗减少大约43.3%.本文提出的免查找表CORDIC算法具有实时性强、输出精度高、硬件资源消耗少等优势,更适用于高速实时的现代数字通信系统应用.

关键词:坐标旋转数字计算机;免查找表;二极化重编码;可编程逻辑门阵列;数字信号处理

中图分类号: TN492 **文献标志码:** A **文章编号:** 0367-6234(2017)11-0109-07

Low-consumption and LUT-omitted CORDIC algorithm

YAO Yafeng, ZOU Lingzhi, WANG Wei, ZHONG Liang

(Faculty of Mechanical and Electronic Information, China University of Geosciences, Wuhan 430074, China)

Abstract: In order to reduce the hardware resource consumption and output delay in the conventional pipeline CORDIC (Coordinate Rotation Digital Computer) algorithm, a LUT-omitted CORDIC algorithm is proposed on the basis of the three-stage CORDIC with look-up table (LUT). The proposed new algorithm adopts 4 iterations of shifter-adder instead of LUT to reduce the register consumption apparently, and merges some iterations to decrease the output delay effectively, and guarantees the output resolution by using a combination of Binary to Bipolar Recoding (BBR) and angle range folding. The improved algorithm is specifically implemented using Verilog HDL on ISE 14.2 software platform and synthesized using XST tools, and is modeled to analyze the output errors through MATLAB. Simulation experiment results show that the register resource consumption of the new algorithm, producing sine/cosine effectively, is reduced by about 74.42% compared with the conventional one, while the output width is set to 16 equally. The computing clock periods are decreased by 68.75%, and the resolution is also improved obviously. Compared with the three-stage algorithm, the consumption of register is reduced by about 43.3%. The proposed algorithm has some advantages such as brilliant real-time performance, high resolution, and low consumption. And it is more applicable for modern digital communication systems that demand for high speed and promising real-time performance.

Keywords: coordinate rotation digital computer; to omit look-up table; binary to bipolar recoding; field-programmable gate array; digital signal processing

坐标旋转数字计算机(Coordinate Rotation Digital Computer, CORDIC)是J.Volder于1959年提出的一种算法^[1],这种算法通过一系列旋转角不断逼近目标角度,由移位相加迭代得到目标角度的正余弦值^[2].CORDIC算法具有硬件实现简单、速度快、精度可调等优点,在全数字锁相环、直接数字频

率合成、快速傅里叶变换等领域有着广泛应用^[3-6].传统的CORDIC算法采用流水线型结构^[7],通过大量迭代来保证输出精度,因此存在硬件资源消耗过多、输出时延较长的问题.

随着现代数字通信系统集成化程度和精度的不断提高,许多学者对CORDIC算法提出改进方案.文献[8]提出一种贪婪搜索算法,采用跳过若干旋转角度和重复某些旋转角度的方法进行迭代,迭代次数有一定减少,但硬件复杂度明显增加.文献[9]采用一种混合结构,将旋转角度分为两部分处理,与

收稿日期:2017-04-05

基金项目:国家自然科学基金(61601334)

作者简介:姚亚峰(1970—),男,副教授,硕士生导师

通信作者:王巍,993164156@qq.com

传统流水线型 CORDIC 算法相比提高运算速度,但仍存在硬件消耗大、输出精度低的问题. 文献[10]中的免缩放因子双步旋转算法免去缩放因子,减少迭代次数,但双步旋转的硬件实现结构较为复杂. 文献[11]提出免缩放自适应 CORDIC 算法,虽然硬件资源消耗和误差有所减少,但是计算所需的时钟周期太长. 文献[12-13]提出一种三阶段 CORDIC 算法,第一阶段通过查找表得到若干次迭代后的值,第二阶段采用移位相加的蝶形迭代运算,第三阶段进行合并迭代,相比于传统的流水线型 CORDIC 算法在精度和硬件消耗上都有一定改善,计算所需的周期少,具有一定代表性和优越性.

本文在文献[12-13]的基础上结合角度二极化重编码(Binary To Bipolar Recoding, BBR)、角度区间折叠、合并迭代等手段提出一种免查找表 CORDIC 算法,将三阶段缩减为两阶段实现. 在典型的三阶段 CORDIC 实现方法基础上,将第一阶段的查找表直接替换为有限的几步移位相加迭代运算,有效减少寄存器消耗. 此外,将第二阶段和第三阶段合并为一步迭代,大幅减少输出时延. 用几步迭代运算代替查找表,使输出延时有增加,但第二阶段的合并迭代只需要一个时钟周期,同时省去原有的蝶形运算阶段,整体只需要 5 个时钟周期就能得到输出结果,整体延时还有所降低. 同时在综合采用角度二极化重编码和角度区间折叠技术后,改进算法的精度相比于传统实现方法也有一定改善.

1 CORDIC 算法基本原理

CORDIC 算法通过一系列旋转角不断逼近输入角度,由移位相加迭代得到输入角度的正余弦值. 图 1 为向量旋转图,在平面直角坐标系中将初始向量 $OA = (x_1, y_1)$ 逆时针旋转角度 θ 到向量 $OB = (x_2, y_2)$,则坐标之间的关系为

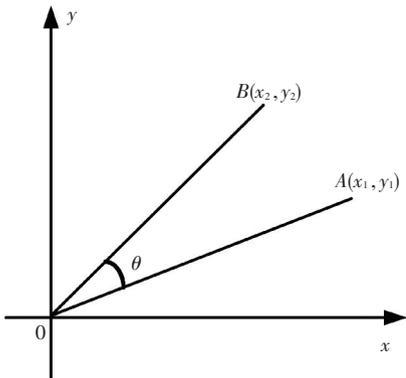


图 1 向量旋转示意

Fig.1 Rotation vector diagram

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} = \cos \theta \begin{bmatrix} 1 & -\tan \theta \\ \tan \theta & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \quad (1)$$

将角度 θ 视为 N 个角度的累加和,即 $\theta =$

$\sum_{i=0}^{N-1} \delta_i \theta_i$, 其中 $\delta_i = \pm 1$, 那么式(1)可写为

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \cos \theta_{N-1} \cos \theta_{N-2} \cdots \cos \theta_0 \cdot \begin{bmatrix} 1 & -\tan \delta_{N-1} \theta_{N-1} \\ \tan \delta_{N-1} \theta_{N-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\tan \delta_{N-2} \theta_{N-2} \\ \tan \delta_{N-2} \theta_{N-2} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan \delta_0 \theta_0 \\ \tan \delta_0 \theta_0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = K \cdot \begin{bmatrix} 1 & -\tan \delta_{N-1} \theta_{N-1} \\ \tan \delta_{N-1} \theta_{N-1} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan \delta_{N-2} \theta_{N-2} \\ \tan \delta_{N-2} \theta_{N-2} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -\tan \delta_0 \theta_0 \\ \tan \delta_0 \theta_0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \quad (2)$$

其中: $K = \cos \theta_{N-1} \cos \theta_{N-2} \cdots \cos \theta_0$ 称为校模因子,在 N 确定时可视为常数. 令 $\theta_i = \arctan(2^{-i})$, 在单次迭代时不考虑校模因子, 那么有

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} =$$

$$\begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \text{ 从上式可以看出,CORDIC 算}$$

法只需要移位和加减操作就能完成正余弦的计算. 为判断每次旋转的方向,引入变量 Z , 令 $Z_{N+1} = \theta - \sum_{i=0}^N \theta_i$, 其取值非负时 $\delta_i = +1$, 取值为负数时 $\delta_i = -1$.

2 免查找表 CORDIC 算法

2.1 角度二极化重编码

对于取值为 $[0, \pi/8]$ 的任意输入角度 θ 可用 N 位二进制小数示为 $\theta = \sum_{k=2}^N b_k 2^{-k}$, 其中 $b_k \in (0, 1)$.

假设 $r_k = 2b_k - 1$, 那么有 $\theta = \sum_{k=3}^{N+1} 2^{-k} + \sum_{k=3}^{N+1} r_k 2^{-k} = \varphi_0 + \sum_{k=3}^{N+1} r_k 2^{-k}$, 其中 $\varphi_0 = \sum_{k=3}^{N+1} 2^{-k} = \frac{1}{4} - (\frac{1}{2})^{N+1}$, 这种方法称为角度二极化重编码^[4]. 初始向量为 $(x_0, y_0) = (K, 0)$, 在得到输入角度后,先将初始向量逆

时针旋转角度 φ_0 , 随后根据二进制角度的比特位值进行旋转. 当 $b_k = 1$ 时 $r_k = 1$, 将向量逆时针方向旋转 2^{-k} 弧度; 当 $b_k = 0$ 时 $r_k = -1$, 将向量顺时针方向旋转 2^{-k} 弧度.

2.2 免查找表与合并迭代原理

根据 CORDIC 算法的基本原理, 可通过式(3)迭代得到输入角度的正余弦值

$$\begin{cases} x_{k+1} = x_k - r_k \tan(2^{-k}) y_k, \\ y_{k+1} = y_k + r_k \tan(2^{-k}) x_k. \end{cases} \quad (3)$$

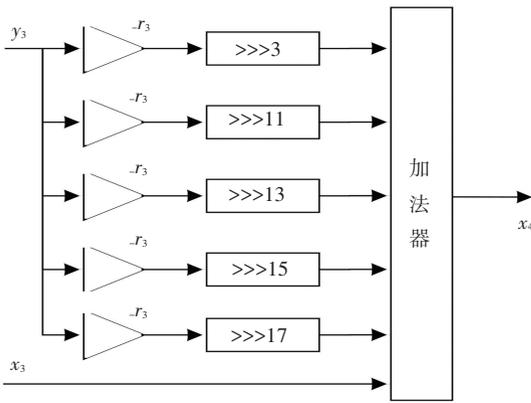
式中 k 代表迭代次数.

在三阶段 CORDIC 算法中, 对于输出 N 位的 CORDIC 算法, 当 $k < N/3$ 时可利用上式(3)建立第一阶段的小容量查找表. 当 $N/3 \leq k < N/2$ 时, 利用 $\tan(2^{-k}) \approx 2^{-k}$ 进行第二阶段的蝶形运算, 当 $k \geq$

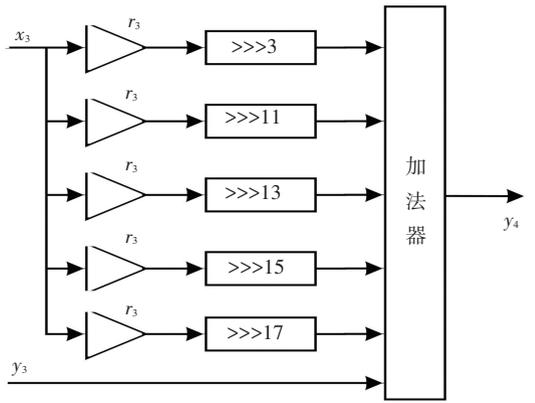
$N/2$ 时进行合并迭代. 本文提出的免查找表 CORDIC 改进算法将第一阶段中 $\tan(2^{-k})$ 用二进制小数近似, 对于输出为 16 位小数的 CORDIC 算法, 第一阶段用到的近似如式(4)所示.

$$\begin{cases} \tan \frac{1}{8} \approx 2^{-3} + 2^{-11} + 2^{-13} + 2^{-15} + 2^{-17}, \\ \tan \frac{1}{16} \approx 2^{-4} + 2^{-14} + 2^{-16}, \\ \tan \frac{1}{32} \approx 2^{-5} + 2^{-17}, \\ \tan \frac{1}{64} \approx 2^{-6}. \end{cases} \quad (4)$$

那么式(3)中乘以正切项就可视为移位相加运算, 其中 $k = 3$ 对应的结构见图 2.



(a) 移位相加结构示意图 1



(b) 移位相加结构示意图 2

图 2 移位相加结构

Fig.2 Shifter-adder structure

图 2 中 (a) 和 (b) 分别代表式(3)中的两个子式. 通过这种近似方法, 将查找表替换为移位相加的蝶形运算结构, 从而减少寄存器消耗.

对于输出为 N 位小数的 CORDIC 算法, 当迭代次数 $k \geq N/2$ 时, 可作 $\tan(2^{-k}) \approx 2^{-k}$ 的近似, 那么

上式(3)所示的迭代可写为 $\begin{cases} x_{k+1} = x_k - r_k 2^{-k} y_k, \\ y_{k+1} = y_k + r_k 2^{-k} x_k, \end{cases}$ 所以迭代次数 $k > N/2$ 后的 s 次蝶形运算可简化为

$$\begin{cases} x_{k+s} = x_k - y_k \cdot \sum_{i=k}^{k+s-1} r_i 2^{-i} \\ y_{k+s} = y_k + x_k \cdot \sum_{i=k}^{k+s-1} r_i 2^{-i} \end{cases}, \text{ 在硬件实现时相当于移}$$

位连加结构.

2.3 区间角度折叠及映射关系

对于 $(\pi/8, \pi/4]$ 内的角度, 存在式(5)所示的三角函数关系式

$$\begin{cases} \sin(\frac{\pi}{4} - \theta) = \frac{\sqrt{2}}{2}(\cos \theta - \sin \theta), \\ \cos(\frac{\pi}{4} - \theta) = \frac{\sqrt{2}}{2}(\cos \theta + \sin \theta). \end{cases} \quad (5)$$

通过上式(5)可将 $(\pi/8, \pi/4]$ 内的角度折叠到 $[0, \pi/8]$ 内, 有

$$\begin{cases} \cos \theta = \frac{\sqrt{2}}{2}[\cos(\frac{\pi}{4} - \theta) + \sin(\frac{\pi}{4} - \theta)], \\ \sin \theta = \frac{\sqrt{2}}{2}[\cos(\frac{\pi}{4} - \theta) - \sin(\frac{\pi}{4} - \theta)]. \end{cases} \quad (6)$$

从式(6)看出, 只要得到 $[0, \pi/8]$ 内角度的正余弦值, 进行一次加减法运算和 $\sqrt{2}/2$ 缩放就可得到 $(\pi/8, \pi/4]$ 内的正余弦值. 通过类似的三角函数关系式, 可将角度扩展到 $[0, 2\pi]$ 范围内. 表 1 是具体的角度映射和正余弦转换关系, 其中 θ' 是输入角度经过区间角度折叠预处理后所得到的角度.

将角度折叠到 $[0, \pi/8]$ 可以省掉三阶段算法

中第二阶段的蝶形运算过程,只需要进行第一阶段 角度映射和转换关系即可还原输出正余弦值. 的移位相加迭代和第三阶段的合并迭代,随后根据

表 1 角度映射和正余弦转换关系

Tab.1 Angle mapping and sine/cosine transition

| 输入角度 θ | 预处理后的角度 θ' | $\cos \theta$ | $\sin \theta$ |
|---------------------|-------------------|--|--|
| $[0, \pi/8]$ | θ | $\cos \theta'$ | $\sin \theta'$ |
| $(\pi/8, \pi/4]$ | $\pi/4 - \theta$ | $\sqrt{2}/2(\cos \theta' + \sin \theta')$ | $\sqrt{2}/2(\cos \theta' - \sin \theta')$ |
| $(\pi/4, 3\pi/8]$ | $\theta - \pi/4$ | $\sqrt{2}/2(\cos \theta' - \sin \theta')$ | $\sqrt{2}/2(\cos \theta' + \sin \theta')$ |
| $(3\pi/8, \pi/2]$ | $\pi/2 - \theta$ | $\sin \theta'$ | $\cos \theta'$ |
| $(\pi/2, 5\pi/8]$ | $\theta - \pi/2$ | $-\sin \theta'$ | $\cos \theta'$ |
| $(5\pi/8, 3\pi/4]$ | $3\pi/4 - \theta$ | $-\sqrt{2}/2(\cos \theta' - \sin \theta')$ | $\sqrt{2}/2(\cos \theta' + \sin \theta')$ |
| $(3\pi/4, 7\pi/8]$ | $\theta - 3\pi/4$ | $-\sqrt{2}/2(\cos \theta' + \sin \theta')$ | $\sqrt{2}/2(\cos \theta' - \sin \theta')$ |
| $(7\pi/8, \pi]$ | $\pi - \theta$ | $-\cos \theta'$ | $\sin \theta'$ |
| $(\pi, 9\pi/8]$ | $\theta - \pi$ | $-\cos \theta'$ | $-\sin \theta'$ |
| $(9\pi/8, 5\pi/4]$ | $5\pi/4 - \theta$ | $-\sqrt{2}/2(\cos \theta' + \sin \theta')$ | $-\sqrt{2}/2(\cos \theta' - \sin \theta')$ |
| $(5\pi/4, 11\pi/8]$ | $\theta - 5\pi/4$ | $-\sqrt{2}/2(\cos \theta' - \sin \theta')$ | $-\sqrt{2}/2(\cos \theta' + \sin \theta')$ |
| $(11\pi/8, 3\pi/2]$ | $3\pi/2 - \theta$ | $-\sin \theta'$ | $-\cos \theta'$ |
| $(3\pi/2, 13\pi/8]$ | $\theta - 3\pi/2$ | $\sin \theta'$ | $-\cos \theta'$ |
| $(13\pi/8, 7\pi/4]$ | $7\pi/4 - \theta$ | $\sqrt{2}/2(\cos \theta' - \sin \theta')$ | $-\sqrt{2}/2(\cos \theta' + \sin \theta')$ |
| $(7\pi/4, 15\pi/8]$ | $\theta - 7\pi/4$ | $\sqrt{2}/2(\cos \theta' + \sin \theta')$ | $-\sqrt{2}/2(\cos \theta' - \sin \theta')$ |
| $(15\pi/8, 2\pi]$ | $2\pi - \theta$ | $\cos \theta'$ | $-\sin \theta'$ |

3 算法实现

对于三阶段算法,第一阶段是建立小容量查找表,将经过校模因子缩放后的纵横坐标值存放起来^[15]. 向量旋转中初始坐标为 $(x_0, y_0) = (K, 0)$, 其中 K 为校模因子且有: $K = \cos(\varphi_0) \cdot \cos(2^{-3}) \cdot \cos(2^{-4}) \cdot \cos(2^{-5}) \cdots \cos(2^{-N-1})$ 那么经过逆时针旋转 φ_0 后的坐标值为

$$\begin{cases} x_3 = x_0 - \tan \varphi_0 \cdot y_0 \\ y_3 = y_0 + \tan \varphi_0 \cdot x_0 \end{cases}, \text{ 其}$$

中 (x_3, y_3) 即为改进算法第一阶段的初始迭代值. 对于输出位宽 $N = 16$ 位(1 位符号位和 15 位小数位)的改进 CORDIC 算法,输入角度位宽采用 18 位(4 位角度范围和 14 位小数),中间数据计算位宽采

用 18 位以减少截断误差,当 $i \geq (N - 1)/2$ (即 $i \geq 7$) 时进行合并迭代. 第一阶段将 (x_3, y_3) 作为初始迭代值,根据输入角度小数部分的高 4 位的位值以及对应的 r_k 进行式(3)所示的免查找表迭代,共迭代 4 次;第二阶段直接进行合并迭代,根据输入角度的高 4 位所指示的角度范围,对照表 1 所示的角度映射和转换关系,对正余弦值做加减和缩放运算即可还原输出结果,算法整体结构见图 3. 其中 $\sqrt{2}/2$ 近似为

$$\frac{\sqrt{2}}{2} \approx 2^{-1} + 2^{-3} + 2^{-4} + 2^{-6} + 2^{-8} + 2^{-14} + 2^{-16}. \tag{7}$$

因此 $\sqrt{2}/2$ 缩放可视为对数据进行移位连加操作.

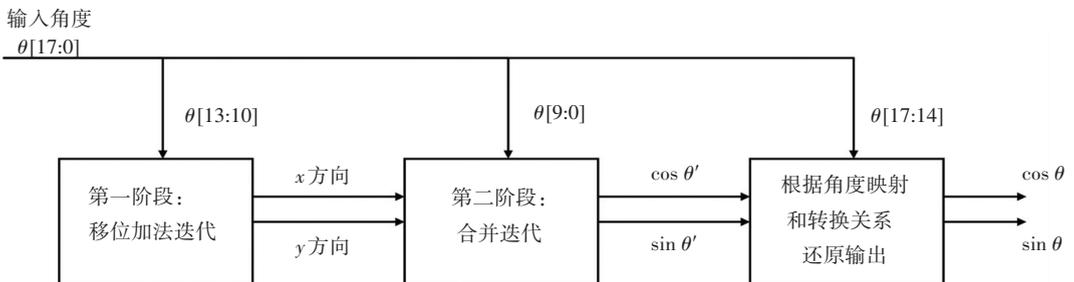


图 3 程序整体结构

Fig.3 Program overall structure diagram

4 仿真结果及分析

用 Verilog HDL 语言对传统流水线型 CORDIC 算法, 三阶段 CORDIC 算法和免查找表 CORDIC 算法进行具体实现, 在相同输出位宽的前提下, 选用 Xilinx 公司的 xc7k325t-2ffg900 型号 FPGA, 利用 ISE14.2 自带的 XST 工具进行电路综合, 得到寄存器和 LUTs 消耗情况以及最大输出时延见表 2。

表 2 三种算法的寄存器和 LUTs 消耗、最大输出时延

Tab.2 Register and LUTs consumption, maximum output delay for three algorithms

| 算法类型 | 输出数据位宽 $N = 16$ | | |
|----------------|-----------------|-------------|--------|
| | 消耗的寄存器个数 | 消耗的 LUTs 个数 | 最大输出时延 |
| 流水线型 CORDIC 算法 | 778 | 836 | 16 |
| 三阶段 CORDIC 算法 | 351 | 679 | 6 |
| 免查找表 CORDIC 算法 | 199 | 696 | 5 |

寄存器的消耗主要取决于查找表的大小和迭代次数, 最大输出时延与迭代次数有关。改进算法通过四次迭代替换查找表, 又通过合并迭代减少大量的迭代, 整体只需要 5 个时钟周期就能得到输出结果。比较三种算法使用的寄存器数量和最大输出时延可看出, 改进算法使用的寄存器数量仅为流水线算法的 25.58%, 相比于三阶段算法节省了 43.3% 的寄存器消耗, 最大输出时延相比于流水线型算法减少了 68.75%。

利用 MATLAB 对三种算法进行建模和误差分析, 输入角度以 2^{-15} 分辨率遍历 $[0, \pi/2]$, 计算得到绝对误差。表 3、4 分别列出三种算法正余弦误差的最大值、最小值以及平均绝对误差值, 其中三种算法的余弦值绝对误差如图 4 所示, 其中红色点代表的是绝对误差的最大值与最小值。通过比较三种算法的平均绝对误差值, 可看出改进算法与流水线型算法的平均绝对误差处于同一数量级, 但改进算法的平均绝对误差相对较小。和三阶段算法相比, 改进算法的平均绝对误差相差一个数量级, 其主要原因是在中间数据位宽较少时式 (4) 的近似误差较大, 这一误差可通过采用更大的中间数据位宽来减小。

表 3 三种算法的正弦值误差

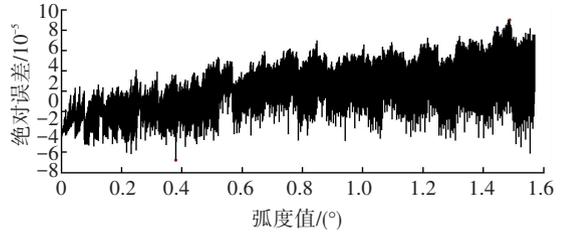
Tab.3 Sine errors of three algorithms

| 算法类型 | 正弦值误差/ 10^{-5} | | |
|----------------|------------------|---------|------------------------|
| | 最大值 | 最小值 | 平均绝对误差 |
| 流水线型 CORDIC 算法 | 12.262 | -18.964 | 6.925 |
| 三阶段 CORDIC 算法 | 1.192 | -1.448 | 3.245×10^{-1} |
| 免查找表 CORDIC 算法 | 9.802 | -6.872 | 2.496 |

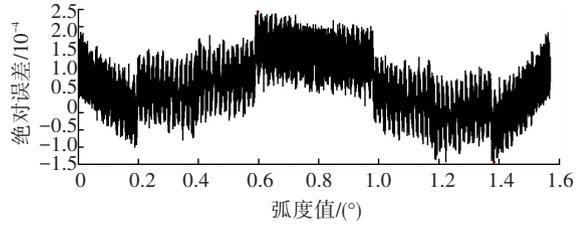
表 4 三种算法的余弦值误差

Tab.4 Cosine errors of three algorithms

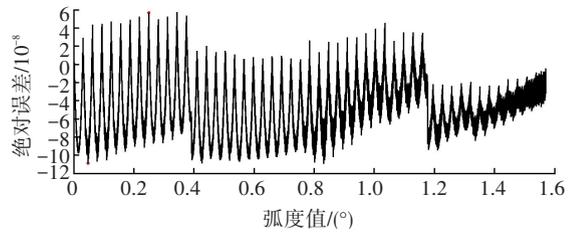
| 算法类型 | 余弦值误差/ 10^{-5} | | |
|----------------|------------------------|---------|------------------------|
| | 最大值 | 最小值 | 平均绝对误差 |
| 流水线型 CORDIC 算法 | 22.526 | -11.973 | 6.834 |
| 三阶段 CORDIC 算法 | 5.703×10^{-1} | -1.178 | 3.549×10^{-1} |
| 免查找表 CORDIC 算法 | 9.699 | -6.887 | 2.528 |



(a) 流水线型 CORDIC 算法的余弦值绝对误差



(b) 三阶段 CORDIC 算法的余弦值绝对误差



(c) 免查找表 CORDIC 算法的余弦值绝对误差

图 4 三种算法的余弦值绝对误差

Fig.4 Cosine absolute errors of three algorithms

5 结论

针对传统流水线型 CORDIC 算法实现中存在的硬件资源消耗多、输出时延大等问题, 结合角度二极化重编码、合并迭代以及角度区间折叠等技术提出了一种免查找表 CORDIC 算法。该算法将三阶段算法中的查找表替换为移位相加的迭代结构并省去了蝶形运算阶段, 随后直接进行合并迭代, 并根据角度映射关系还原输出结果。结果表明: 与传统流水线型算法相比, 免查找表算法的寄存器消耗减少了约 74.42%, 计算所需的时钟周期降低了 68.75%。本文提出的免查找表 CORDIC 算法在实时性、输出精度、硬件资源消耗方面具有优势, 适合高速实时的现代数字通信系统应用。

参考文献

- [1] VOLDER J E. The CORDIC trigonometric computing technique [J]. IRE Transactions on Electronic Computers, 1959, EC-8(3): 330-334.
- [2] AGGARWAL S, MEHER P K, KHARE K. Concept, design, and implementation of reconfigurable CORDIC [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, 24(4): 1588-1592.
- [3] DAS A, DASH S, BABU B C, et al. CORDIC algorithm based novel phase detection system for all digital phase locked loop [J]. Journal of Computational Intelligence & Electronic Systems, 2012, 1(1): 23-30.
- [4] KUMM M, KLINGBEIL H, ZIPF P. An FPGA-based linear all-digital phase-locked loop [J]. IEEE Transactions on Circuits & Systems I: Regular Papers, 2010, 57(9): 2487-2497.
- [5] MEHER P K, VALLS J, JUANG T B, et al. 50 years of CORDIC: algorithms, architectures, and applications [J]. IEEE Transactions on Circuits & Systems I: Regular Papers, 2009, 56(9): 1893-1907.
- [6] BOHER L, RABINEAU R, HELARD M. FPGA implementation of an iterative receiver for MIMO-OFDM systems [J]. IEEE Journal on Selected Areas in Communications, 2008, 26(6): 857-866.
- [7] HUANG Hai, XIAO Liyi. CORDIC based fast radix-2 DCT algorithm [J]. IEEE Signal Processing Letters, 2013, 20(5): 483-486.
- [8] 梁源, 王兴华, 向新, 等. 一种基于贪婪算法的 CORDIC 改进算法[J]. 电讯技术, 2014, 54(3): 312-317.
LIANG Yuan, WANG Xinghua, XIANG Xin, et al. An improved CORDIC algorithm based on greedy algorithm [J]. Telecommunication Engineering, 2014, 54(3): 312-317.
- [9] WANG Shaoyun, PIURI V, WARTZLANDER E E. Hybrid CORDIC algorithms [J]. IEEE Transactions on Computers, 1997, 46(11): 1202-1207.
- [10] 徐成, 秦云川, 李肯立, 等. 免缩放因子双步旋转 CORDIC 算法 [J]. 电子学报, 2014, 42(7): 1441-1445.
XU Cheng, QIN Yunchuan, LI Kenli, et al. Double-step scaling free CORDIC [J]. Acta Electronica Sinica, 2014, 42(7): 1441-1445.
- [11] MAHARATNA K, BANERJEE S, GRASS E, et al. Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture [J]. IEEE Transactions on Circuits & Systems for Video Technology, 2005, 15(11): 1463-1474.
- [12] 姚亚峰, 付东兵, 杨晓非. 基于 CORDIC 改进算法的高速 DDS 电路设计 [J]. 华中科技大学学报自然科学版, 2009(2): 25-27.
YAO Yafeng, FU Dongbing, YANG Xiaofei. Implement of high speed DDS circuit design using improved CORDIC algorithm [J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2009(2): 25-27.
- [13] MADISETTI A, KWENTUS A Y, WILLSON A N. A 100-MHz, 16-b, direct digital frequency synthesizer with a 100-dBc spurious-free dynamic range [J]. IEEE Journal of Solid-State Circuits, 1999, 34(8): 1034-1043.
- [14] JUANG T B, HSIAO S F, TSAI M Y. Para-CORDIC: parallel CORDIC rotation algorithm [J]. IEEE Transactions on Circuits & Systems I: Regular Papers, 2004, 51(8): 1515-1524.
- [15] 牟胜梅, 李兆刚. 基于查找表和 SF CORDIC 的高精度正余弦函数求值方法 [J]. 计算机与数字工程, 2014, 42(3): 359-363.
MOU Shengmei, LI Zhaogang. A computation method of high-accuracy sine & cosine function based on lookup-table and SF CORDIC algorithm [J]. Computer & Digital Engineering, 2014, 42(3): 359-363.

(编辑 苗秀芝)