

基于小容量查找表的 CORDIC 算法设计

姚亚峰,徐洋洋,侯强[†],钟梁

(中国地质大学 机械与电子信息学院,湖北 武汉 430074)

摘要: 为了对流水线结构的坐标旋转数字计算(Coordinate Rotation Digital Computer, CORDIC)的实现时延和硬件资源消耗进行优化,提出一种仅基于查找表的新的实现方法,完全免除了迭代运算.该方法只需要一个较低容量的 ROM 表,以及对 ROM 表输出结果进行简单的移位运算,即可得到高精度的正弦波或余弦波输出.分别在 Matlab、Modelsim 以及 XILINX ISE 进行了理论仿真及实际验证,结果表明:这种 CORDIC 实现方法只需要 2 个时钟周期的处理延时,硬件资源消耗与其他实现方法相比也有所降低,最大工作频率也有一定提高.

关键词: 坐标旋转数字计算;流水线结构;查找表;可编程逻辑门阵列;数字信号处理

中图分类号: TN492; TN911.72

文献标志码: A

Implement of CORDIC Algorithm with a Small Capacity ROM Table

YAO Yafeng, XU Yangyang, HOU Qiang[†], ZHONG Liang

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

Abstract: In order to optimize the implementation delay and hardware resource consumption of the Coordinate Rotation Digital Computer(CORDIC) in pipeline architecture, a new implementation method based on look-up table was proposed, which completely eliminated the iterative computation. This method requires only a lower capacity ROM table, as well as a simple shift operation on the output of the ROM table, so that sine wave or cosine wave output with high accuracy can be obtained. Theoretical simulation and practical verification were carried out in Matlab, Modelsim and XILINX ISE, and the results show that this method of CORDIC only requires two clock cycles of processing delay, and the hardware resource consumption is also reduced when compared with other methods. Additionally, the circuit output accuracy and maximum working frequency are also improved at a certain level.

Key words: coordinate rotation digital calculation (CORDIC); pipeline architecture; look-up table; programmable logic gate array; digital signal processing

* 收稿日期:2018-07-20

基金项目:国家自然科学基金资助项目(61601334), National Natural Science Foundation of China(61601334); 中央高校军民融合专项基金培育项目(201708), Central University Military-Civilian Integration Special Fund Cultivation Project(201708)

作者简介:姚亚峰(1970—),男,湖北武汉人,中国地质大学(武汉)副教授,博士

[†] 通讯联系人, E-mail:houq2008@126.com

CORDIC (Coordinate Rotation Digital Computer) 算法核心思想是通过迭代来不断逼近目标值,其精度可以通过迭代次数来控制,具有输出结果精度可调、硬件实现简单等特点^[1],方便实现坐标变换、三角函数计算、矩阵运算等多种数字信号处理功能,在直接数字频率合成器 DDS、数字控制振荡器 NCO、快速傅里叶变换 FFT 等领域都得到广泛应用.随着现代通信技术不断更新发展,通信系统对 CORDIC 算法实现要求也在不断提高^[2-3].近年来,有关 CORDIC 算法的高效实现一直得到人们持续关注和研究.文献[4]提出的实现方法虽然有很高的相位分辨率,但是其最大工作频率受到限制并且功耗较大.文献[5]提出的表驱动缩小目标旋转角度的方法虽然迭代次数有所减少,但是其硬件资源消耗和最大输出时延仍然较大.文献[6-7]提出的固定角度旋转 CORDIC 算法利用混合旋转 CORDIC 算法原理进行一次角度估计和多次单向角度旋转,虽然减少了迭代次数,但同时也降低了精度.文献[8-9]提出的双步旋转方法虽然免去了校正因子,减少了迭代次数,但是该方法硬件结构比较复杂,硬件资源消耗仍然较大.文献[10-11]提出将整个迭代过程分为三个阶段完成:第一阶段查找 4 位地址的查找表;第二阶段进行 4 次蝶形迭代运算;第三阶段通过合并运算得出输出结果.这种典型的实现方法在运算速度、计算精度和电路资源消耗等方面都有一定改善,但是需要 7 个时钟周期的延时才能得到输出结果,不太适合高速、实时的信号处理与应用.

本文在文献[10-11]的基础上,提出一种只基于查找表的新的实现方法,完全免除了迭代运算.这种实现方法只需要一个 6 位地址输入的小容量 ROM 表,然后对查表输出的结果采用简单的移位和相加运算进行结果修正,即可得到满足特定精度要求的正弦波或余弦波输出.这种实现方法在电路资源消耗没有增加,输出精度保持一致的前提下,输出延时从 7 个时钟周期缩短为 2 个时钟周期,尤其适合实时性要求高、处理速度快的应用场合.

1 CORDIC 算法实现原理

CORDIC 算法^[12]通过一系列的迭代运算即可得到特定角度的正弦值、余弦值,如图 1 所示,

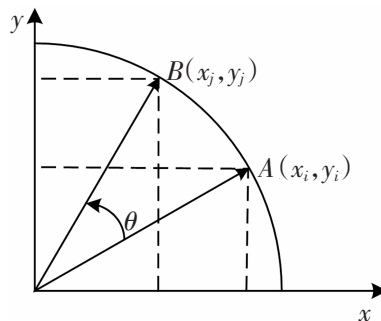


图 1 坐标旋转角度 θ

Fig.1 Coordinate rotation angle θ

根据坐标 A 旋转角度 θ 到达坐标 B 后,两者坐标关系表达式如下:

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

令 $\theta = \sum_{i=0}^{N-1} d_i \theta_i$, 且 $\theta_i = \arctan 2^{-i}$, $d_i = \{-1, 1\}$. 则式(1)可表示为 N 次迭代运算:

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = k \begin{bmatrix} 1 & -d_i 2^{-i} \\ -d_i 2^{-i} & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & -d_1 2^{-1} \\ -d_1 2^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & -d_0 \\ -d_0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \quad (2)$$

式中: $k = \prod_{i=0}^{N-1} \cos \theta_i$, k 为模校正因子; d_i 表示旋转方向.

当剩余角度即 $\theta - \sum_{i=0}^n \theta_i$ 大于 0 时,旋转方向为逆时针, $d_i = 1$, 否则旋转方向为顺时针, $d_i = -1$. 文献[10]将输入角度 θ 限定在 $[0, \pi/4]$, 并且用 N 位二进制数表示, 即 $\theta = \sum_{k=1}^N b_k 2^{-k}$ 其中 $b_k \in \{0, 1\}$. 因为输入角度在 $[0, \pi/4]$ 以外时, 可以利用三角函数对称性进行角度折叠, 能够把它映射到 $[0, \pi/4]$, 所以在此只考虑输入角度范围为 $[0, \pi/4]$ 时的情况. 在文献[10]中, 进行第 i 次迭代时, 对应的旋转角度不再是 $\theta_i = \arctan 2^{-i}$, 而是 $\theta_i = 2^{-i}$. 这就导致了 CORDIC 基本算法中一个关键表达式 $\tan \theta_i = 2^{-i}$ 不再成立, 但通过对 $\tan \theta_i$ 进行泰勒展开即可发现, 例如要得到 16 bit 精度三角函数计算结果, $i \geq 6$ 时, $\tan \theta_i \approx 2^{-i}$, 近似所导致的误差忽略不计. 即在迭代进行一定次数之后, 每次迭代对应的旋转角度是开始能够满足 $\theta_i = \arctan 2^{-i}$ 这一要求的. 所以三段式实现方法的核心思想是把原来需要的 N 次迭代运算划分为三个阶段来实现: 在 $i < N/3$ 时, 即最初的几步迭代运算通过查找表来得到运算结果. 在 $N/3 \leq i < 2N/3$ 时, 迭代运算根据式(2)计算. 在 $i \geq 2N/3$ 时, 最后的几步迭代运算就简化和合

并为一步来完成。

这种三段式实现方法,在保证特定输出精度的前提下,尽可能对计算表达式进行简化或合并,在电路资源消耗、电路工作速度等方面取得了有效平衡,是利用CORDIC算法计算三角函数值时常用的典型的实现方法.ADI公司许多直接频率合成DDS电路或数字控制振荡器NCO电路,均采用此方法。

2 基于小容量查找表的CORDIC实现

本文提出的基于小容量查找表的CORDIC算法,是在三阶段实现CORDIC算法的基础上进行的改进.仍然将输入角度 θ 限定在 $[0, \pi/4]$,并且用 N 位二进制数表示这个输入角度.接着把表示这个输入角度的 N 位bit $b_0b_1 \cdots b_{N-2}b_{N-1}$ 分为两组,一组包含前面6位bit $b_0b_1 \cdots b_5$,另一组包含剩余的全部bit $b_6b_7 \cdots b_{N-2}b_{N-1}$,其中 b_0 表示MSB(Most Significant Bit)。

把第一组6个bit对应的6次迭代运算结果存储到ROM中,即最初6步的迭代运算是采用一个查找表来代替和实现的.该表里面存储的数据是最初6次迭代运算以及同时考虑了校正因子的结果.由于 $\pi/4 \approx 2^{-1} + 2^{-2} + 2^{-5} + 2^{-8} + 2^{-12}$,所以,6-bit查找表所需最大地址为51而不是63.该查找表的存储内容如表1所示。

表1 ROM查找表

Tab.1 ROM lookup table ADDR

ADDR	$x_7(\cos \alpha)$	$x_7(\sin \alpha)$
000000	01111111111111010	00000000111111111
000001	011111111111011010	00000010111111111
000010	011111111110011010	00000100111111110
000011	011111111100111010	000001101111111100
...
110000	010111001111100010	010101111111101001
110001	010110111001010110	010110010110101101
110010	010110100010110100	010110101101011011

限定在 $[0, \pi/4]$ 范围内的任意输入角度 θ 用 N 位二进制数表示,我们把 N 位bit分为两组,假设这两组bit实际表示的角度分别为 α, β ,则输入角度 $\theta = \alpha + \beta$.若直接把查找表的结果作为最终输出结果,则会存在较大误差,因为未考虑剩余角度 β 的影响。

$$\begin{cases} \cos\theta = \cos(\alpha + \beta) = \cos\alpha\cos\beta - \sin\alpha\sin\beta \\ \sin\theta = \sin(\alpha + \beta) = \sin\alpha\cos\beta + \cos\alpha\sin\beta \end{cases} \quad (3)$$

根据泰勒公式, $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$, $\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$ ($-\infty < x < \infty$), 即当 x 很小时, $\sin x \approx x$, $\cos x \approx 1$. 而剩余比特 $b_6b_7 \cdots b_{N-2}b_{N-1}$ 表示的角度 β 完全满足这个条件,式(3)可简化为:

$$\begin{cases} \cos\theta = \cos(\alpha + \beta) = \cos\alpha - \beta \cdot \sin\alpha \\ \sin\theta = \sin(\alpha + \beta) = \sin\alpha + \beta \cdot \cos\alpha \end{cases} \quad (4)$$

$\cos\alpha$ 和 $\sin\alpha$ 取值通过查找表即可得到.按照式(4)对查找表输出结果进行修正,经过相加和移位运算,即可得到符合精度要求的正弦、余弦值输出。

例如需要得到16位精度要求的正弦、余弦值输出,其输入角度 θ 位宽取18位,其中高3位用来进行角度折叠,即把整个坐标系的任意输入角度折叠到 $[0, \pi/4]$ 范围,接下来的6位bit用于查找表地址,最后的所有比特用来对查表结果进行校正.其具体实现原理框图如图2所示。

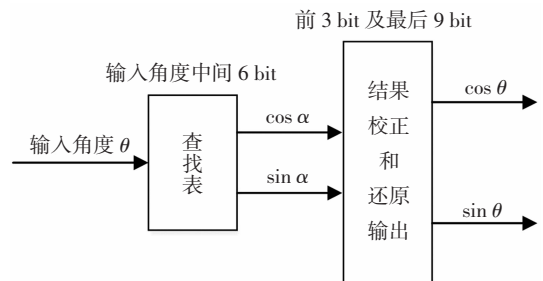


图2 设计实例总体框图

Fig.2 Overall block diagram of design example

3 仿真结果及其分析

将本文的实现方法和文献[11]中提出的三阶段实现方法进行了比较.在ISE14.2环境下,使用XILINX公司的xc7k325t-2ffg900型号的FPGA,利用软件自带的综合工具XST综合,对比分析寄存器消耗、最大时延和电路最大工作频率等性能,综合对比结果如表2所示.改进CORDIC算法的寄存器消耗量与三阶段实现方法相比减少了12.5%。

用Modelsim仿真,分别计算改进CORDIC算法和三阶段实现方法的最大输入输出时延.在实际实现时,由于三阶段CORDIC算法的第二阶段需要多级迭代运算,每级迭代均需要一个时钟周期,所以时延较长.而在改进算法中,完全将三阶段CORDIC

算法的后两步合并,即第一个时钟周期进行查表,第二个时钟周期对查表结果进行校正并输出.由表 2 可知,改进 CORDIC 算法只需要 2 个时钟周期就可以得到输出结果,而三阶段实现方法却需要 7 个时钟周期.改进的 CORDIC 算法与三阶段实现方法相比,尤其适合对实时性要求较高的场合.

表 2 综合结果对比

Tab.2 Comparison of comprehensive results

CORDIC 算法类型	消耗寄存器个数	最大工作频率/MHz	最大输出时延
三阶段	359	182.170	7
改进	314	242.251	2

在 ISE14.2 环境下,用软件自带的功耗分析工具 Xpower 分别在 50 MHz、100 MHz 和 150 MHz 的频率上进行功耗测试,数据如表 3 所示.改进 CORDIC 算法比三阶段 CORDIC 算法的功耗有所减少,且随着工作频率的增大,改进算法的功耗增加速率明显低于三阶段实现方法.

表 3 功耗对比

Tab.3 Comparison of power consumption

CORDIC 算法类型	工作频率/MHz		
	50	100	150
三阶段	0.148	0.174	0.200
改进	0.143	0.163	0.184

图 3 给出在不同输出位宽的情况下的最大输出时延.可以看出,改进算法的最大输入输出时延始终为 2 个时钟周期,但是三阶段实现方法的最大输出时延却呈缓慢阶梯型增长.

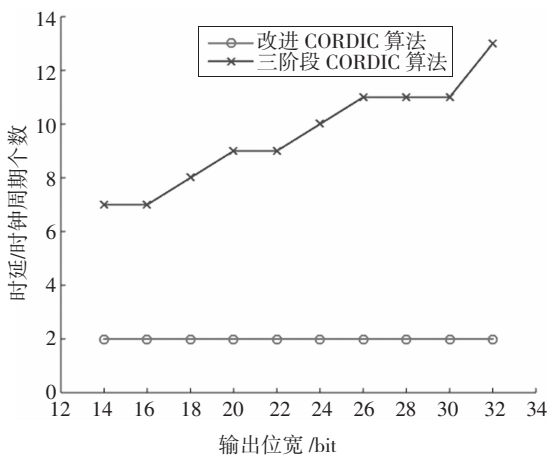


图 3 不同输出位宽的时延对比

Fig.3 Different output bit width of the delay comparison

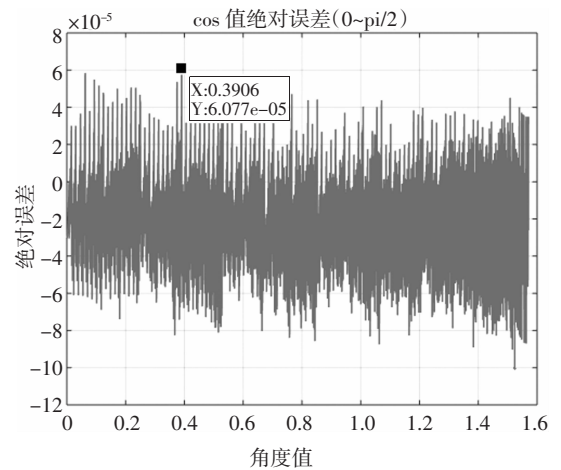
用 Matlab 对改进 CORDIC 算法和三阶段方法进行建模分析.表 4 给出在不同的输出位宽下两种算法的绝对误差.从表中可以看出,改进算法的精度要稍优于三阶段方法.

表 4 误差分析对比

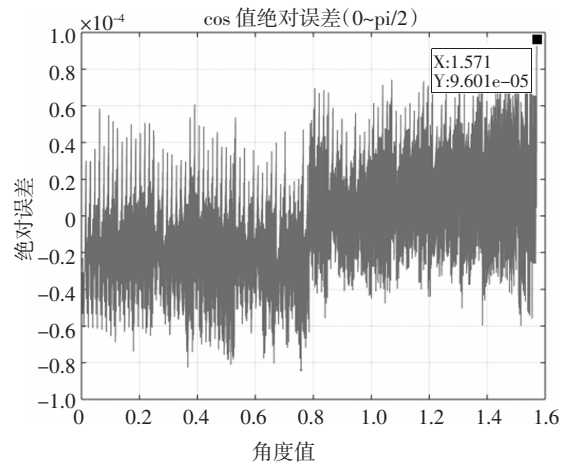
Tab.4 Error analysis and comparison

CORDIC 算法类型	输出位宽/bit			
	14	16	18	20
改进	6.977×10^{-5}	1.396×10^{-5}	8.936×10^{-6}	8.968×10^{-6}
三阶段	1.008×10^{-4}	3.025×10^{-5}	1.431×10^{-5}	1.671×10^{-5}

图 4 给出输出位宽为 16-bit 时改进算法和三阶段实现方法的误差仿真结果.在图 4 中,以角度分辨率为 2^{-15} ,绘制 $[0, \pi/2]$ 上的 cos 的误差.从图中可以看出,两种算法的最大误差分别为 6.077×10^{-5} 和 9.601×10^{-5} .



(a)改进算法



(b)三阶段方法

图 4 输出位宽为 16-bit 时的误差对比

Fig.4 Error comparison of output bit width of 16-bit

4 结语

本文描述了一种基于小容量查找表的CORDIC算法实现原理. 该算法针对三阶段实现方法时延和硬件资源消耗较大的问题, 在保证该算法和三阶段实现方法的输出位宽相同的情况下, 免除了三阶段实现方法的迭代和合并迭代步骤, 有效降低了时延, 并且输出精度有所提高. 用MATLAB对该改进方法和三阶段实现方法建模并进行了性能的比较和分析, 同时在XILINK公司的xc7k325t-2ffg900型号的FPGA上对改进的方法进行具体的设计和实现. 仿真结果表明: 在输出位宽为16-bit时, 最大输入输出时延降低了71.4%, 寄存器消耗降低了12.5%, 并且输出精度稍优于三阶段方法. 和三阶段CORDIC算法相比, 该算法的性能有了明显改善.

参考文献

- [1] SHUKLA R, RAY K C. Low latency hybrid CORDIC algorithm [J]. IEEE Transactions on Computers, 2014, 63(12): 3066—3078.
- [2] MUNOZ S D, HORMIGO J. High-throughput FPGA implementation of QR decomposition [J]. IEEE Transactions on Circuits and Systems II: Express Briefs, 2015, 62(9): 861—865.
- [3] PRASAD N, TRIPATHY M R, DAS A D, *et al.* Efficient VLSI implementation of CORDIC based direct digital synthesizer [J]. Intelligent Computing, Communication and Devices, 2015, 308(1): 597—603.
- [4] NGUYEN H T, NGUYEN X T, PHAM C K. A low power hybrid adaptive CORDIC [J]. IEEE Transactions on Circuits & Systems II Express Briefs, 2018, 65(4): 496—500.
- [5] 史方显, 曾立, 陈昱, 等. 改进型高速高精度CORDIC算法及其在DDFS中的应用[J]. 电子学报, 2017, 45(2): 446—451.
- [6] SHI F X, ZENG L, CHEN Y, *et al.* Direct digital frequency synthesizer based on an improved high speed & high precision CORDIC algorithm [J]. Acta Electronica Sinica, 2017, 45(2): 446—451. (In Chinese)
- [7] 张朝柱, 韩吉南, 燕慧智. 高速高精度固定角度旋转CORDIC算法的设计与实现[J]. 电子学报, 2016, 44(2): 485—490.
- [8] ZHANG C Z, HAN J N, YAN H Z. Design and implementation of CORDIC algorithm for high speed and precision fixed angle of rotation [J]. Acta Electronica Sinica, 2016, 44(2): 485—490. (In Chinese)
- [9] MEHER P K, PARK S Y. CORDIC design for fixed angle of rotation [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2013, 21(2): 217—228.
- [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.
- [11] XU C, QIN Y C, LI K L, *et al.* Double-step scaling free CORDIC [J]. Acta Electronica Sinica, 2014, 42(7): 1441—1445. (In Chinese)
- [12] 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.
- [13] 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, 37(2): 25—27.
- [14] YAO Y F, FU D B, YANG X F. Implement of high speed DDS circuit design using improved CORDIC algorithm [J]. Journal of Huazhong University of Science and Technology (Nature Science Edition), 2009, 37(2): 25—27. (In Chinese)
- [15] MADISETTI A, KWENTUS A Y, WILLSON A N. A 100MHz, 16b, direct digital frequency synthesizer with a 100dbc spurious free dynamic range [J]. IEEE Journal of Solid-state Circuits, 1999, 34(8): 1034—1043.
- [16] VOLDER J E. The CORDIC trigonometric computing technique [J]. IRE Transactions on Electronic Computers, 1959, 8(3): 330—334.