

基于 CORDIC 算法的 DDS 的 FPGA 设计与优化

陈义和^{1,2}, 余 磊^{1,2}, 李交美^{1,2}

(1 中国科学院 武汉物理与数学研究所, 湖北 武汉 430071;

2 中国科学院 原子频标重点实验室(武汉物理与数学研究所), 湖北 武汉 430071)

摘 要: 采用 CORDIC 算法计算三角函数值来实现 DDS, 可以减少存储资源, 便于在 FPGA 中实现. 通过对传统 CORDIC 算法流水结构的分析, 提出了一种在迭代过程采用不同位宽的寄存器存储角度值和幅度值的优化方法, 可以节省资源而不影响计算精度, 并且在 FPGA 中实现了该方法.

关键词: DDS; CORDIC 算法; FPGA; 优化

中图分类号: TN741

文献标识码: A

文章编号: 1000-7180(2012)08-0111-04

Design and Optimization of DDS Based on CORDIC in FPGA

CHEN Yi-he^{1,2}, SHE Lei^{1,2}, LI Jiao-mei^{1,2}

(1 Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan 430071, China; 2 Key Laboratory of Atomic Frequency Standards(KLAIFS), Wuhan Institute of Physics and Mathematics, Chinese Academy of Sciences, Wuhan 430071, China)

Abstract: This paper realizes DDS by using the CORDIC algorithm to calculate the value of trigonometric function, which is convenient to the FPGA implement. According to analysis of the traditional pipeline structure CORDIC algorithm, we present a new method of optimization that using different bits to save the angle value and amplitude value during iterations, and realize it in FPGA, which can reduce the FPGA resource.

Key words: DDS; CORDIC; FPGA; Optimization

1 引言

DDS(直接数字频率合成)在通信、雷达以及频标^[1]等领域得到了广泛的应用, 传统的 DDS 一般是基于存储表的方法实现的, 该方法要求存储表容量非常大, 不利于在 FPGA 中实现. 由 Volder^[2] 于 1959 年提出的 CORDIC(coordinate rotation digital computer, 坐标旋转数字计算)算法, 通过迭代的方法, 只需要少量的存储空间, 而且每一级的迭代只需要通过简单的移位、加/减法运算和少量的查表操作, 便能够计算出三角函数值, 因此基于 CORDIC 算法的频率综合器在很多方面得到了应用. 本文通过对 CORDIC 算法的流水结构的分析, 提出了一种可以减少 FPGA 资源而不影响计算精度的优化设

计方法. 通过该方法实现的 DDS 已经成功应用在汞离子微波频标实验中.

2 DDS 基本原理与 CORDIC 算法

传统的 DDS 工作原理框图如图 1 所示. 它的数字部分主要由相位累加器和波形存储表组成, 相位

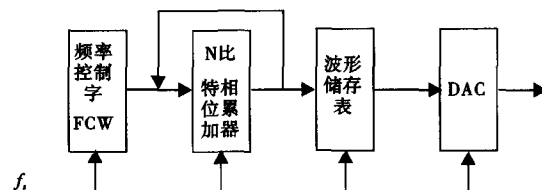


图 1 DDS 工作原理框图

累加器的输出作为波形存储表的地址, 查表得到对

应的三角函数值,经过 DAC 输出,平滑滤波后便得到对应的正弦信号^[3].

如果没有相位截断, N 比特的相位所对应的波形存储器必须有 2^N 个条目,因此不便于在 FPGA 中实现. 而利用 CORDIC 算法则只用存储少量的数据,通过计算便可以得出三角函数值.

CORDIC 算法有两种运算模式:旋转模式和向量模式. 利用 CORDIC 算法可以计算三角函数值、开方及反三角函数值等. 这里只介绍基于 CORDIC 算法的三角函数值的计算. 在旋转模式中,如图 2 所示,在直角坐标系中,向量 $\vec{OA}$ 旋转角度 θ 后得到新的向量 $\vec{OB}$,则它们的坐标满足下面关系^[4]:

$$\begin{cases} x_1 = x_0 * \cos\theta - y_0 * \sin\theta = \cos\theta(x_0 - y_0 * \tan\theta) \\ y_1 = y_0 * \cos\theta + x_0 * \sin\theta = \cos\theta(y_0 + x_0 * \tan\theta) \end{cases}$$

为了便于在硬件中实现这种旋转运算,在 CORDIC 算法中,将 θ 分解成一系列小角度 $\theta_i = \arctan(2^{-i})$ 的和,通过一系列的旋转来逼近. 则

$$\theta = \sum_{i=0}^{n-1} \mu_i \theta_i = \sum_{i=0}^{n-1} \mu_i \arctan(2^{-i}), \text{ and } \mu_i = +1 \text{ or } -1$$

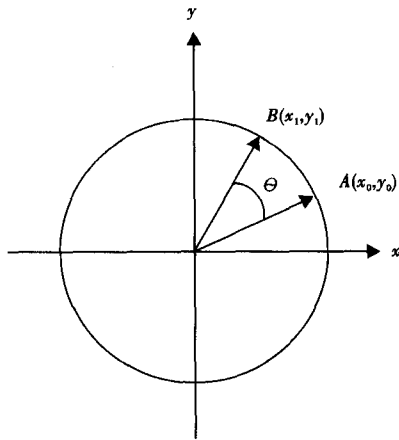


图 2 向量旋转坐标

上式的计算结果是收敛的^[5],因为它满足 $\theta_i - \sum_{j=i+1}^{n-1} \theta_j < \theta_{n-1}$, $\forall i, i = 0, 1, \dots, n-2$. 由于 $\sum_{i=0}^{\infty} \theta_i = 1.743\,286\dots$, 所以通过 CORDIC 旋转算法能够计算的角度必须满足: $-1.743\,29 \leq \theta \leq 1.743\,29$.

当 $i > N/3$ (N 为角度量化时小数部分的位宽) 时,使用 $\arctan(2^{-i}) \approx 2^{-i}$ 近似将不会影响计算精度.

经过 i 次旋转后,向量 $\vec{OA}$ 与向量 $\vec{OB}$ 之间的角度差为 z_i , 满足

$$z_0 = \theta, z_i = z_{i-1} - \mu_{i-1} \theta_{i-1},$$

如果 $z_i \geq 0$, 则 $\mu_i = 1$, 否则 $\mu_i = -1$. 第 i 次旋转后得到的坐标与前一次的坐标满足如下的关系

$$\begin{cases} x_{i+1} = \cos\theta_i(x_i - \mu_i y_i * 2^{-i}) \\ y_{i+1} = \cos\theta_i(y_i + \mu_i x_i * 2^{-i}) \end{cases}, \text{ and } \mu_i = +1 \text{ or } -1.$$

令 $k_i = \cos\theta_i = 1/\sqrt{(1+2^{-2i})}$ 是第 i 次旋转的增益因子,与 μ_i 的值无关,因此在实际的迭代过程中,可以将其放到最后一步进行修正,此时迭代关系式为

$$x_{i+1} = x_i - \mu_i y_i * 2^{-i},$$

$$y_{i+1} = y_i + \mu_i x_i * 2^{-i},$$

$$z_{i+1} = z_i - \mu_i \theta_i.$$

如果经过 n 次旋转,则最终的增益因子:

$$k = \prod_{i=0}^{n-1} k_i = \prod_{i=0}^{n-1} 1/\sqrt{(1+2^{-2i})},$$

该式收敛于 $0.607\,3^{[6]}$.

因此,如果取初始值 $x_0 = k, y_0 = 0, z_0 = \theta$, 则经过 n 次迭代后, $x_n \approx \cos\theta, y_n \approx \sin\theta$, 迭代次数越多,精度越高.

3 基于 CORDIC 算法的 DDS 的 FPGA 设计与优化

由上一部分知道,CORDIC 算法计算的角度范围为: $-1.743\,29 \leq \theta \leq 1.743\,29$, 而在基于 CORDIC 算法的 DDS 中,角度取值范围一般为 $[0, 2\pi)$.

根据三角函数的对称性, $[0, \pi/2)$ 内的正余弦值可以表示 $[0, 2\pi)$ 内所有的余弦值,具体对应关系如下:

$$\frac{\pi}{2} \leq \theta < \pi:$$

$$\cos\theta = -\sin\left(\theta - \frac{\pi}{2}\right), \sin\theta = \cos\left(\theta - \frac{\pi}{2}\right) \pi \leq \theta$$

$$< \frac{3\pi}{2}:$$

$$\cos\theta = -\cos(\theta - \pi), \sin\theta = -\sin(\theta - \pi)$$

$$\frac{3\pi}{2} \leq \theta < 2\pi:$$

$$\cos\theta = \sin\left(\theta - \frac{3\pi}{2}\right), \sin\theta = -\cos\left(\theta - \frac{3\pi}{2}\right)$$

因此预处理部分需要将 $[0, 2\pi)$ 内的值映射到 $[0, \pi/2)$, 通过 CORDIC 算法计算出正余弦值,之后根据之前的映射关系,将结果进行调整.

CORDIC 算法硬件实现时的迭代框图如图 3 所示. 在 FPGA 中,基于流水结构的 CORDIC 实现时,

每一级迭代,都需要用寄存器保存 x_i, y_i, z_i 的值,因此需要消耗 FPGA 的寄存器资源。

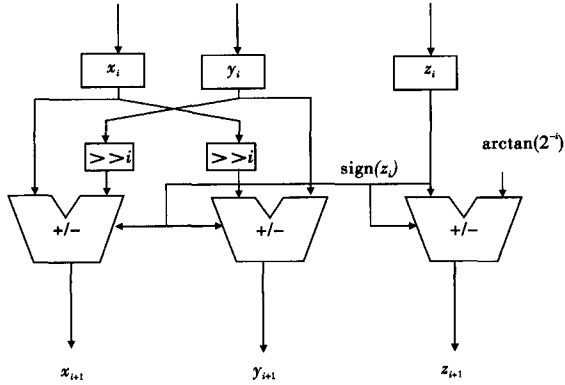


图3 CORDIC算法硬件迭代结构

在传统的CORDIC算法中,对每一步的 x_i, y_i 的量化采用相同的位数,比如 L 比特,在基于式(2)的所有的迭代过程中 $|x_i| < 1.647, |y_i| < 1.647$,传统的基于CORDIC算法的DDS将初始坐标预置 (x_0, y_0) 为 $(1.647, 0)$,在本文的改进算法中,将初始迭代时的坐标为 (x_0, y_0) 预置为 $(1, 0)$,计算结果只是在幅度上有个增益因子,但是并不会影响DDS频率的输出。在迭代过程中有:

第一次迭代: $\begin{cases} x_1: (x_0 \pm y_0 * 2^{-0}) \\ y_1: (y_0 \mp x_0 * 2^{-0}) \end{cases}$, 计算结果

只需要用2比特表示即可;

第二次迭代: $\begin{cases} x_2: (x_1 \pm y_1 * 2^{-1}) \\ y_2: (y_1 \mp x_1 * 2^{-1}) \end{cases}$, 计算结果

只需用 $(2+1)$ 比特表示;

第三次迭代: $\begin{cases} x_3: (x_2 \pm y_2 * 2^{-2}) \\ y_3: (y_2 \mp x_2 * 2^{-2}) \end{cases}$, 计算结果

只需要用 $(2+1+2)$ 比特表示;

第 i 次迭代: $\begin{cases} x_{i+1}: (x_i \pm y_i * 2^{-i}) \\ y_{i+1}: (y_i \mp x_i * 2^{-i}) \end{cases}$, 计算结果

只需要用 $(2+1+2+\dots+(i-1))$ 比特表示。

因此,可以发现,对于前几级在迭代过程中,对幅度的量化,不用采用 L 比特,而是采用 2、3、5、8……,当 $(2+\sum_{j=1}^i j-1) > L$,才统一截断为 L 比特,这样和对所有的幅度都采用 L 比特量化相比较,会节省FPGA资源。

同样,在每一步迭代都需要存储角度值 z_i ,当 $0 \leq \theta < \frac{\pi}{2}$ 时,可以表示为

$$\theta = \sum_{i=0}^{n-1} \mu_i \theta_i, u_i = +1 \text{ or } -1, \theta_i = \arctan(2^{-i})$$

因此第 i 次旋转后的角度 $|z_{i+1}| = |z_i - \mu_i \theta_i| < \theta_i < 2^{-i}$, 当 $i > N/3$ 时, $|z_{i+1}| = |z_i - \mu_i \theta_i| < \theta_i < 2^{-i}$, 意味着 z_{i+1} 最高位到第 i 位都为符号位,可以只保留一位符号位,减少 z_{i+1} 的存储位宽,当 $i > N/3$ 时,每次迭代后角度 z_{i+1} 只需要用 $(N-i)$ 位存储即可。这样可以减少对FPGA资源的消耗。

4 FPGA优化实例

下面以角度量化为22比特,其中小数部分为20比特,整数部分为1比特,最高为符号位,幅度在计算过程中,最多量化为32比特为例。那么对计算过程中角度和幅度的存储位宽如表1所示。

表1 角度和幅度存储位宽表

i 值	x_i, y_i 存储位宽	z_i 的存储位宽
0	2	22
1	2	22
2	3	22
3	5	22
4	8	22
5	12	22
6	17	22
7	23	22
8	30	15
9	32	14
10	32	13
11	32	12
12	32	11
13	32	10
14	32	9
15	32	8
16	32	7
17	32	6
18	32	5
19	32	4
20	32	3

在 altera 公司 CycloneII 系列 EP2K8F256C8 的FPGA中实现基于CORDIC算法的DDS,本文

提出的优化算法和传统 CORDIC 算法的资源消耗情况如表 2 所示。

表 2 不同算法下, FPGA 资源的消耗情况

	Logic elements	Registers	Memory bits
传统算法	2 588	1 476	34
本文算法	2 147	1 218	34

可以看出, 使用本文提出的算法实现的 DDS 比传统算法在不仅在 register 方面比减少了约 17% 的资源, 而且在 logic elements 方面也减少了约 17% 的资源, 这是因为在减少角度和幅度量化的同时, 也就相应的减少了加/减法器的位宽, 也就会减少 logic elements 资源。

5 结束语

本文通过对传统 CORDIC 算法中流水结构的存储单元进行优化, 在不影响精度的情况下节约了 FPGA 的资源. 而且当迭代次数和量化位宽越宽时, 节约的 FPGA 资源将会更多. 应用本文算法在 altera 公司的 FPGA 中实现的 DDS 已经成功应用在频标实验中。

参考文献:

- [1] 王勇, 吕善伟, 冯克明, 等, 基于 CORDIC 算法的 DDS 在铷原子频标中的应用[J], 宇航计测技术, 2005, 25(3): 15-18.
- [2] Volder J E, The CORDIC trigonometric computing technique[J]. IRE Trans Electron Computers, 1959 (EC-8): 330-334.
- [3] 沈志, 王宏远, 王振. 无相位截断误差的 DDS 设计[J]. 微电子学与计算机, 2011, 28(3): 70-72.
- [4] Pramod K Meher, Javier Valls, Sridharan K. 50 Yearsof CORDIC: algorithms, architectur-es and applications [J]. IEEE Trans On Circui-ts and systems-I: Regular papers, 2009(56): 1893-1906.
- [5] Walther J S. A unified algorithm for elementary functions[C]//Proc 38 th Spring Joint Computer Conf Atlantic City, NJ: IEEE, 1971: 379-385.
- [6] 张天瑜. 基于旋转模式的改进型 CORDIC 算法研究 [J]. 微电子学与计算机, 2010, 27(3) 93-97.

作者简介:

陈义和 男, (1983-), 硕士, 工程师. 研究方向为电路和信号处理.

余磊 男, (1979-), 博士, 副研究员. 研究方向为原子频标技术与实验.

李交美 男, (1956-), 研究员, 博士生导师. 研究方向为原子频标.

(上接第 110 页)

- [9] 戴永寿, 牛慧, 彭星, 等, 基于自回归滑动平均模型和粒子群算法的地震子波提取 [J]. 中国石油大学学报, 2011, 35(3): 47-50.
- [10] 孙敦灿, 陈运. 功耗分析平台中混合编程的应用研究 [J]. 成都信息工程学院学报, 2011, 26(2): 127-131
- [11] 陈俊, 陈运. 抗功耗分析攻击的椭圆曲线梳状优化算法

[J]. 成都信息工程学院学报, 2010, 25(4): 341-344

作者简介:

陈艾东 女, (1978-), 博士生, 讲师. 研究方向为边信道攻防与物联网安全.