

一种改进型 CORDIC 算法的 FPGA 实现

张建斌¹, 梁 芳², 刘乃安³

(1 江苏技术师范学院 电气信息工程学院, 江苏 常州 213001; 2 陕西理工学院 电信工程系, 陕西 汉中 723003;
3 西安电子科技大学 通信工程学院, 陕西 西安 710071)

摘 要: 为实现 CORDIC 算法在二、三象限内的点的反正切函数的计算, 提出了在传统 CORDIC 算法基础上增加两级初次迭代的改进措施, 给出了改进后算法的硬件流水线实现结构, 并在 FPGA 芯片 EP1S10F484C5 上仿真实现. 仿真结果表明: 修正后的 CORDIC 算法的运算结果与反正切函数的理论计算值基本一致, 误差很小, 可以实现平面上任意一点反正切函数的求解.

关键词: CORDIC 算法; 反正切函数; 流水线结构; FPGA

中图分类号: TN391.9

文献标识码: A

文章编号: 1000-7180(2010)11-0181-04

FPGA Implement of a Modified CORDIC Algorithm

ZHANG Jian-bin¹, LIANG Fang², LIU Nai-an³

(1 Department of E. I., Jiangsu Teachers University of Technology, Changzhou 213001, China;

2 Department of E. I., Shaanxi University of Technology, Hanzhong 723003, China;

3 School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)

Abstract: To achieve arctan calculation in second quadrant or third quadrant, the improvement measure of increasing the two 45° rotation on traditional CORDIC algorithm is proposed. The hardware pipeline structure of the modified CORDIC algorithm is put forward and simulated on FPGA stratix EP1S10F484C5 of Altera company. The simulation results indicated that the modified CORDIC algorithm result is essentially coincident with theoretical value and has less errors, therefore, the modified CORDIC algorithm can be used for arctan calculation of any point in coordinate plane.

Key words: CORDIC algorithm; arctan function; pipeline structure; FPGA

1 引言

在数字信号处理中, 经常会用到三角函数运算、指数函数运算、矢量旋转等基本数学运算, 为了解决这些问题, J. Volder 于 1959 年提出了坐标旋转数字计算机 (Coordinate Rotation Digital Computer, CORDIC) 算法^[1]. J. Walther 于 1971 年提出了统一的改进型 CORDIC 算法^[2], 改进后的算法除了可以完成三角函数、反三角函数和求向量的模, 还可以完成乘法、除法、超越函数等运算. CORDIC 算法的基本思想是用一系列与运算基数相关的基本角度进行不断旋转, 最终逼近需要达到的旋转角度, 从本质上讲该算法是一个数值逼近的方法, 由于这些固定的

基本角度与计算基数有关, 新向量的坐标计算只需通过简单的移位和加减运算即可, 因而该算法便于软硬件实现, 目前已经广泛应用于三角函数计算、FFT、DCT、DWT、DDFS 以及数字调制解调等诸多领域.

文中针对反正切函数的 CORDIC 算法不能正确完成二、三象限内的点的反正切函数的计算的缺陷, 提出了对 CORDIC 算法的修正方案, 即在 CORDIC 算法的基础上, 增加两级 45° 旋转, 从而实现圆周上任意一点的反正切函数的计算. 仿真结果表明这种改进方法是正确可行的. 在此基础上, 文中提出了具有流水线结构的反正切函数的改进算法的 FPGA 实现方法, 通过 Quartus6.0 环境下的仿真表

明,该方法的运算结果与反正切函数的理论运算结果具有一致性.

2 CORDIC 算法原理

J. Walther 提出的统一 CORDIC 算法引入了参数 m ,从而将 CORDIC 算法实现的三角运算、双曲运算和线性运算三种迭代模式统一于一个表达式下,形成 CORDIC 算法最基本的数学基础. 三种运算可用图 1 中的 m 来表征,当 $m = 1$ 、 -1 或 0 时,分别为圆周旋转运算(用于计算正、余弦函数与反正切函数)、双曲旋转运算(用于计算双曲函数、指数函数、平方根函数等)和线性旋转运算(用于乘法与除法运算). 对图 1 中的点 $p(x_0, y_0, z_0)$,其中 z_0 为向量 op 与 x 轴的夹角,定义三参数的序列迭代集为 $\{(x_k, y_k, z_k)\}$,当迭代次数 $k \geq 0$ 时,有:

$$\begin{cases} x_{k+1} = x_k - m\delta_k y_k 2^{-k} \\ y_{k+1} = y_k + m\delta_k x_k 2^{-k} \\ z_{k+1} = z_k - \delta_k \epsilon_k \end{cases} \quad (1)$$

$$\text{其中 } \epsilon_k = \begin{cases} \tan^{-1}(2^{-k}), m = 1 \\ 2^{-k}, m = 0 \\ \tanh^{-1}(2^{-k}), m = -1 \end{cases}$$

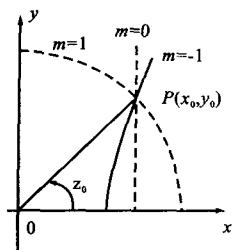


图 1 CORDIC 算法的三种坐标

文中讨论的反正切函数的计算属于圆周旋转运算,因此取 $m = 1$. 于是,由式(1)可得反正切函数的计算迭代公式为:

$$\begin{cases} x_{k+1} = x_k - \delta_k y_k 2^{-k} \\ y_{k+1} = y_k + \delta_k x_k 2^{-k} \\ z_{k+1} = z_k - \delta_k \tan^{-1}(2^{-k}) \end{cases}$$

$$\text{其中 } \delta_k = \begin{cases} +1, y_k < 0 \\ -1, y_k \geq 0 \end{cases}$$

基于 CORDIC 旋转法计算反正切函数值的输入输出关系如图 2 所示^[3]. 要得出平面上一点 $p(x_0, y_0)$ 反正切函数的值,可将坐标 x_0, y_0 分别赋给 CORDIC 旋转计算器的 X、Y 通道,而 Z 通道赋初值 0.

理论上,经 ∞ 次迭代后,有: $y_\infty \rightarrow 0, z_\infty =$

$\arctan(y_0/x_0)$ 此时,图 2 中的 K 值为:

$$K = \prod_{i=0}^{\infty} \sqrt{\frac{1}{1+2^{-2i}}} \approx 0.6073$$

可得 X 通道的输出为:

$$x_\infty = \sqrt{x_0^2 + y_0^2} / \prod_{i=0}^{\infty} \sqrt{\frac{1}{1+2^{-2i}}} \approx 1.6467 \times \sqrt{x_0^2 + y_0^2} \quad (2)$$

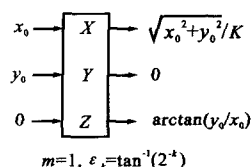


图 2 基于 CORDIC 旋转法计算反正切函数值的输入输出关系

3 CORDIC 算法的改进及 MATLAB 仿真

根据 J. Walther 收敛推导,CORDIC 算法的旋转角度覆盖为 $(-99.9^\circ, +99.9^\circ)^{[4]}$,该覆盖角度不能完成一个圆周角的旋转. 为解决这一问题,文中提出了改进算法,即在上述旋转角度范围的基础上,通过增设两次 45° 旋转,将 CORDIC 算法的旋转角扩大到整个平面: $(-189.9^\circ \sim +189.9^\circ)$. 对改进的 CORDIC 算法进行 MATLAB 仿真,算法如下:

Step1: $x_0, y_0 \leftarrow$ 初值; $z_0 \leftarrow 0; k \leftarrow N$;

Step2: angletable = atan(2.^-(0:N-1));
 $i \leftarrow 0; j \leftarrow 2$

Step3:

if $y(i) < 0$

$$\begin{cases} x(i+1) = x(i) - y(i) * 2^{(-i)}; \\ y(i+1) = y(i) + x(i) * 2^{(-i)}; \\ z(i+1) = z(i) - \text{angletable}(i+1); \end{cases}$$

else

$$\begin{cases} x(i+1) = x(i) + y(i) * 2^{(-i)}; \\ y(i+1) = y(i) - x(i) * 2^{(-i)}; \\ z(i+1) = z(i) + \text{angletable}(i+1); \end{cases}$$

end

Step4: $j > 0?$ ($j = j-1, i = 0$); $i = i+1; i \leq k$
重复 Step3, 否则输出结果.

用该修正算法计算反正切函数的 MATLAB 仿真结果如表 1 所示(迭代 10 次). 由表 1 可以看出,该修正算法的 MATLAB 的仿真结果与理论计算反正切函数的结果基本一致. 当然,由于增加了两级 45° 旋转,对于 K 的影响:

$$K = \sqrt{\frac{1}{1+2^0}} \cdot \sqrt{\frac{1}{1+2^0}} \cdot \prod_{i=0}^{\infty} \sqrt{\frac{1}{1+2^{-2i}}} \approx 0.3036$$

表 1 修正 CORDIC 算法的 MATLAB 仿真结果($n = 10$)

x	y	运算结果($^{\circ}$)	理论值($^{\circ}$)
1	1	44.9257	45
1	-1	-45.0743	-45
-1	1	134.9257	135
-1	-1	-135.0743	-135
0	1	90.0672	90
0	-1	-89.9328	-90
1	0	0.0672	0
-1	0	179.9328	180

对 X 通道的输出结果也产生影响:

$$x_{\infty} = \sqrt{x_0^2 + y_0^2} / K = \sqrt{x_0^2 + y_0^2} / 0.3036 \quad (3)$$

在一个实际的数字通信系统中,无论载波同步单元还是解调电路单元,着重关心的是反正切函数的值的运算结果正确与否,而对 x_{∞} 的大小一般是不关心的.所以 MATLAB 仿真中可以忽略 x_{∞} .但是当该算法采用 VHDL 语言实现时,就必须引起重视(要防止其溢出),否则将不能实现正确的运算结果.

4 修正 CORDIC 算法的 FPGA 实现

CORDIC 算法的实现方式有两种:简单状态机法和高速全流水线处理器.简单状态机结构计算速度慢,迭代时所设计的筒形移位器复杂.流水线 CORDIC 结构虽然占用的硬件资源比简单状态机结构稍多,但是流水线结构可以提高数据的吞吐率.

从当前 FPGA 的发展趋势来看,芯片内的门资源相对富裕,对流水线 CORDIC 的实现规模约束很小,此外,流水线 CORDIC 不存在迭代式 CORDIC 的反馈回路,使得单元结构更加规则,有利于 FPGA 实现.

文中采用流水线 CORDIC 结构基于 FPGA 实现反正切的计算,如图 3 所示.图 3 中的前三级为相同的 45° 旋转,其中第一级旋转是 CORDIC 算法固有的 45° 旋转,另外两级旋转是对它的修正,以增加 CORDIC 算法的覆盖角度.图 3 中的 x_i, y_i, θ_i 分别是点 $P(x_0, y_0)$ 的 x_0, y_0 坐标与角度的偏移量; θ_o 是反正切结果与偏移量 θ_i 之和.

设 CORDIC 算法进行 n 次迭代,则其内部字长 $W = 3 + (n - 1) + \log_2(n)^{[5]}$.文中取 $P(x_0, y_0)$ 的坐标 x_0, y_0 字长为 16bits, $\overrightarrow{OP}$ 向量与 x 轴正向的夹

角 z_0 的字长为 18bits,其中 MSB 为符号位.这样, x_0, y_0 的数据位长 15bit, z_0 的数据位长 17bit.

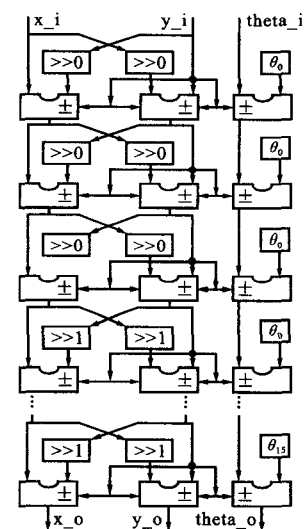


图 3 流水线 CORDIC 结构

为便于 VHDL 编程进行综合,须对 X, Y, Z 三通道上的数据做整数化处理,即 $x_0 \times (2^{15} - 1), y_0 \times (2^{15} - 1), z_0 \times (2^{17} - 1)$ 后取整,其中 z_0 为角弧度.同时,基于 CORDIC 算法计算反正切函数的 VHDL 语言实现对 ϵ_k 做相同处理,如表 2 所示($\epsilon_k = \arctan(2^{-k})$).

表 2 CORDIC 迭代算法所用角度

K	ϵ_k	K	ϵ_k	K	ϵ_k	K	ϵ_k
0	25735	4	2045	8	128	12	8
1	15192	5	1024	9	64	13	4
2	8027	6	512	10	32	14	2
3	4075	7	256	11	16	15	1

根据确定的字长并考虑到式(3)的成立,如果取 x_{∞} 的最大整数化后的值为 32767,则 $|x_0|$ 的最大整数化后的值为 9947,其对应平面上的单位圆 $r = 1$;该圆内的点的反正切函数为 $(-\pi, +\pi)$,对应的整数是 $(-102940, +102939)$.

根据修正后 CORDIC 算法计算反正切函数的迭代算法,用 VHDL 语言描述如下(其中 X, Y 两个通道的输入端子与输出端子是为便于结果测试):

```
process(clk)
begin
if clk'event and clk = '1' then
if (y(17)<0) then
x_o<= x(17) - shiftright(y(17),15);
y_o<= y(17) + shiftright(x(17),15);
```

```

theta_o <= z(17) - 1;
else
    :
else
    x(1) <= x(0) + y(0);
    y(1) <= y(0) - x(0);
    z(1) <= z(0) + 25736;
end if
end if
end process

```

上述的 VHDL 代码,在 Quarts II 6.0 环境下,选取 Altera 公司的 FPGA stratix EP1S10F484C5 器件后进行编译,该代码占用了该器件的资源如图 4 所示.可见该代码的硬件资源消耗较少.

Logic Element Usage	■ 1,180 / 10,570 (11 %)
Memory Block Usage	■ 0 / 920,448 (0 %)
DSP Block Usage	■ 0 / 48 (0 %)
I/O Usage	■ 83 / 336 (25 %)

图 4 消耗的硬件资源

最大半径为 9947 的圆上点的反正切的仿真结果与理论结果一致.半径小于 9947 的某些圆上点的反正切计算结果也是修正 CORDIC 算法计算反正切后的结果.说明文中所述的反正切函数的计算方法是正确的,可以进行 FPGA 实现.

5 结束语

文中提出了 CORDIC 算法的改进方法,给出了

FPGA 的实现结构,其结构简单,硬件实现方便,基于 Quarts II 6.0 环境,在 Altera 公司的 FPGA stratix EP1S10F484C5 进行仿真验证,其误差较小,占用资源较少,运算速度快.文中提出的改进 CORDIC 算法实现反正切函数的计算已用于实际数字通信系统中,该算法也适合于其他高速运算的场合,具有较好的实用价值.

参考文献:

- [1] Volder J E. The CORDIC trigonometric computing technique[J]. IRE Transactions on Electronics Computers, 1959,8(3):330-334.
- [2] Walther J S. A unified algorithm for elementary functions[J]. Spring Joint Computer Conference Proceedings, 1971 (38):379-385.
- [3] 张欣. 扩频通信数字基带信号处理算法及其 VLSI 实现[M]. 北京:科学出版社,2004.
- [4] 李滔,韩月秋. 基于流水线 CORDIC 算法的三角函数发生器[J]. 电子技术应用,1999,25(6):52-53.
- [5] 刘福奇,刘波. Verilog HDL 应用程序设计实例精讲[M]. 北京:电子工业出版社,2009.

作者简介:

张建斌 男,(1966-),副教授.研究方向为无线通信、扩频通信、EDA 技术等.

梁芳 女,(1980-),实验师.研究方向为 DSP 与 FPGA 应用系统设计.

刘乃安 男,(1966-),教授,硕士生导师.研究方向为无线通信与射频技术、无线 IP 网络与技术、扩频通信.

(上接第 180 页)

- [4] 陈智育. VxWorks 程序开发实践[M]. 北京:人民邮电出版社,2004.
- [5] 孔祥营. VxWorks 驱动及分布式编程[M]. 北京:中国电力出版社,2007.
- [6] 周启平. VxWorks 下设备驱动程序及 BSP 开发指南[M]. 北京:中国电力出版社,2004.
- [7] Wind River. Tornado training workshop[R]. Wind River 培训资料. 1999.
- [8] Wind River. VxWorks:guide to writing and porting device

drivers[R]. Wind River. 1999.

- [9] 严蔚民. 数据结构[M]. 北京:清华大学出版社,1996.
- [10] Stephen Prata. C primer plus[M]. 北京:人民邮电出版社,2005.
- [11] 张海藩. 软件工程导论[M]. 北京:清华大学出版社,1998.

作者简介:

温婷 女,(1985-).研究方向为军事装备学.