

# 基于 CORDIC 算法的载波同步锁相环设计和实现

陈建中, 聂景楠, 张华伟

(解放军理工大学 通信工程学院, 江苏 南京 210007)

**摘要:** 研究了一种利用 CORDIC 算法的矢量及旋转模式对载波同步中相位偏移进行估计并校正的方法。设计并实现了基于 CORDIC 算法的数字锁相环。通过仿真, 验证了设计的有效性和高效性。

**关键词:** CORDIC 算法; 载波同步; 锁相环

中图分类号: TN914.3

文献标识码: B

## Design and implementation of CORDIC-based PLL for carrier synchronization

CHEN Jian Zhong, NIE Jing Nan, ZHANG Hua Wei

(Institute of Communications Engineering, PLA University of Science & Technology, Nanjing 210007, China)

**Abstract:** Estimation and correction of phase offset in carrier synchronization was studied by using the CORDIC algorithm of the vector and rotation mode. Designed and implemented a phase locked loop (PLL) based on CORDIC algorithm. By simulation, the efficiency and high performance of the design was proved.

**Key words:** CORDIC algorithm; carrier synchronization; PLL

在数字接收机中对信号进行相干检测时, 接收端需要提供与发射端调制载波同频同相的相干载波。而本地产生的载波在频率与发送端一致的情况下, 相对于发送端仍会有残余频偏和初始相位引起的相位偏移。

为了能够正确地对接收数据进行解调判决, 必须估计出相位偏移并对接收数据进行校正。针对这种相位偏移, 一般采用锁相环进行跟踪补偿<sup>[1]</sup>, 数字锁相环由鉴相器、环路滤波器、数控振荡器 3 个基本元件构成。CORDIC (Coordinate Rotation Digital Computer) 算法是一种用于计算基本运算函数和算术操作的旋转迭代算法, 其基本思想是通过一系列固定角度的微旋转而逼近所需要的角度, 实现乘、除、平方、三角函数、复数乘法等运算<sup>[2]</sup>。适当选取旋转角度, 可以使 CORDIC 算法只需要进行移位和加减操作, 非常适合硬件实现。将 CORDIC 算法用于锁相环中鉴相器和数控振荡器的设计, 可以避免一般查表法消耗 ROM 资源较多、精度受限以及复数乘法等缺陷。

### 1 CORDIC 算法原理<sup>[2]</sup>

#### 1.1 CORDIC 算法基本旋转迭代方程组

设向量  $(x_i, y_i) = (r \cos \alpha, r \sin \alpha)$ , 将其旋转  $\theta$  角后得到新向量  $(x_j, y_j) = (r \cos(\alpha + \theta), r \sin(\alpha + \theta))$ , 则:

$$\begin{cases} x_j = r \cos(\alpha + \theta) = r(\cos \alpha \cos \theta - \sin \alpha \sin \theta) = x_i \cos \theta - y_i \sin \theta \\ y_j = r \sin(\alpha + \theta) = r(\cos \alpha \sin \theta + \sin \alpha \cos \theta) = x_i \sin \theta + y_i \cos \theta \end{cases} \quad (1)$$

其矩阵形式如下:

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2)$$

对于旋转的角度  $\theta$ , 可以将它分解成若干个微小的旋转, 设第  $n$  次旋转的角度为  $\theta_n$ , 则有:

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

为了将  $\tan \theta_n$  项的乘法操作简化为移位操作, 可以限定  $\tan \theta_n = \pm 2^{-n}$ ,

$$\text{则: } \theta_n = \delta_n \times \arctan 2^{-n}, \quad \delta_n = \{-1, +1\}$$

$$\cos \theta_n = \cos(\delta_n \times \arctan 2^{-n}) = \cos(\arctan 2^{-n})$$

$$\theta = \sum \theta_n$$

经过数次迭代后,  $\cos \theta_n$  的乘积趋于常数, 记为  $K$ 。因此, 在迭代过程中, 可以暂时将  $\cos \theta_n$  忽略。

引入变量  $Z$  表示尚未旋转的角度, 则经过  $n$  次旋转后尚未旋转的角度:

$Z_{n+1} = \theta - \sum \theta_i = Z_n - \theta_n = Z_n - \delta_n \times \arctan 2^{-n}$ , 结合式 (3) 得到 CORDIC 算法的基本旋转迭代方程组:

$$\begin{cases} x_n = x_{n-1} - \delta_{n-1} 2^{-(n-1)} y_{n-1} \\ y_n = y_{n-1} + \delta_{n-1} 2^{-(n-1)} x_{n-1} \\ z_n = z_{n-1} - \delta_{n-1} \arctan(2^{-(n-1)}) \end{cases} \quad (4)$$

经过  $N$  次旋转迭代后得到的  $(x_N, y_N)$  又可以看作是

由初始向量  $(x_0, y_0)$  旋转  $\theta$  得到, 即:

$$\begin{cases} x_N = (x_0 \cos\theta - y_0 \sin\theta)/k \\ y_N = (y_0 \cos\theta + x_0 \sin\theta)/k \\ z_N = z_0 - \theta; \theta = \sum \delta_n \arctan(2^{-n}) \end{cases} \quad (5)$$

$$|\theta| \leq \sum_{n=0}^{\infty} \arctan(2^{-n}) \approx 1.7433 \text{ (rad)}, \text{ 且 } \sum_{n=0}^3 \arctan(2^{-n}) \approx$$

1.6184 (rad)  $> \pi/2$ , 所以只要迭代次数  $N$  大于 4, 就能够保证旋转的角度覆盖第一、第四象限。一般地,  $N$  取  $\min\{[-\log_2 \tan(1.7433 \times 2/2^M)] + 1, M\}$ , 其中,  $M$  为输入输出的量化位数。对于第二、第三象限的角度, 只需要进行相应的预处理和后处理, 将之变换到第一、第四象限进行处理。

### 1.2 CORDIC 算法旋转模式

当  $\delta_n = \text{sign}(Z_n)$  时, 经过  $N$  次式(4)的旋转迭代, 输出  $Z_N \rightarrow 0$  (表示已经旋转了初始输入角度  $Z_0$ ),  $x_N = x_0 \cos z_0 - y_0 \sin z_0$ ,  $y_N = y_0 \cos z_0 + x_0 \sin z_0$ , 称为 CORDIC 算法的旋转模式, 即完成了  $(x_0 + jy_0)$  与  $e^{jz_0}$  的复数乘法运算。

### 1.3 CORDIC 算法矢量模式

当  $\delta_n = -\text{sign}(x_n) \text{sign}(y_n)$  时, 令输入  $z_0 = 0$ , 经过  $N$  次式(4)的旋转迭代, 则输出  $y_N \rightarrow 0$  (表示旋转到了  $X$  轴上),  $x_N = \text{mod}(x_0, y_0)$ ,  $z_N = \arctan(y_0/x_0)$ , 称为 CORDIC 算法的矢量模式, 完成了输入向量  $(x_0, y_0)$  相角的计算。在矢量模式中, 可以取  $\delta_n = -\text{sign}(y_n)$ , 对于初始输入  $x_0 < 0$  的情况, 进行预处理, 将向量变换到一、四象限进行计算, 减少了在每一级中对  $x_n$  符号的判断, 降低了复杂度。

### 1.4 CORDIC 算法功能示意

CORDIC 算法的两种模式功能示意如图 1 所示。

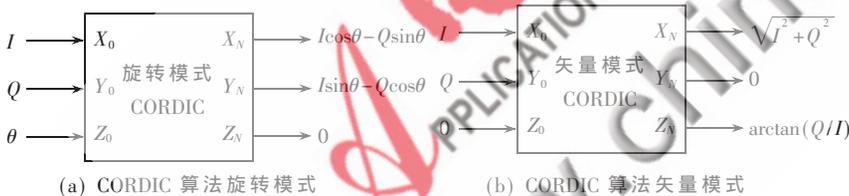


图 1 CORDIC 算法的两种模式示意图

## 2 基于 CORDIC 算法的载波同步锁相环系统原理<sup>[3-5]</sup>

将 CORDIC 算法用于载波同步锁相环, 系统框图如图 2 所示。

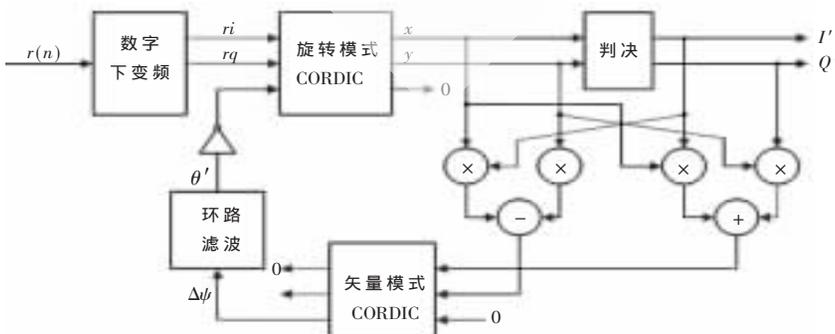


图 2 CORDIC 算法用于载波同步锁相环系统框图

图 2 中, 接收端输入数据  $r(n) = Si \times \cos(\pi \times fc \times n) - Sq \times \sin(\pi \times fc \times n)$ ,  $fc$  为载波频率,  $(Si, Sq)$  为发载波携带向量。经过数字下变频到基带后,  $ri = Si \times \cos\theta - Sq \times \sin\theta$ ,  $rq = Si \times \sin\theta + Sq \times \cos\theta$ ,  $\theta$  为本地载波相对于发送端载波的相位偏移。写成矩阵形式如下:

$$\begin{bmatrix} ri \\ rq \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} Si \\ Sq \end{bmatrix} \quad (6)$$

旋转模式 CORDIC 对  $(ri, rq)$  进行相位偏移的校正。 $(ri, rq)$  经过旋转模式 CORDIC 后输出向量  $(x, y) = (Si \times \cos(\theta - \theta') - Sq \times \sin(\theta - \theta'), Si \times \sin(\theta - \theta') + Sq \times \cos(\theta - \theta'))$ ,  $\theta'$  为本地对相位偏移的估计, 初始为 0。对  $x, y$  的值进行判决, 对于 QPSK,  $I', Q'$  可根据  $(x, y)$  所在象限取  $\pm 1$ ; 对于 QAM, 可将星座图进行划分, 根据  $(x, y)$  所在区域取  $\{\pm 1, \pm 1/2, \dots\}$ , 从而避免了乘法运算。图 2 中所示乘法运算通过简单的逻辑运算即可实现。

矢量模式 CORDIC 对相位偏移和估计的差值进行  $\Delta\psi$  计算。根据  $\tan(\theta - \theta') = (I'y - Q'x)/(I'x + Q'y)$ , 将  $I'x + Q'y, I'y - Q'x$  分别输入矢量模式 CORDIC 的  $x, y$  端, 可以得到  $\Delta\psi = \theta - \theta'$ 。

$\Delta\psi$  经过环路滤波, 得到  $\theta$  的最大似然估计  $\theta^{(1)}$ 。环路滤波一般选取二阶环路, 主要对信号噪声起到抑制作用, 同时调节环路的校正速度。

### 3 功能仿真及硬件实现

(1) 基于 CORDIC 算法的载波同步锁相环在 Matlab 中进行浮点仿真。

环路滤波采用二阶环路, 数据采取 QPSK 调制, CORDIC 算法选取 9 级旋转迭代, 假定存在  $\pi/8$  的公共相位偏移。初始接收信号经过环路相位校正后的信号如图 3 所示。

#### (2) CORDIC 算法的硬件实现

由式(4)的旋转迭代方程组可以看出, CORDIC 算法实现只需要基本的移位和加法操作, 它的每一级迭代旋转的硬件实现基本单元如图 4 所示。CORDIC 算法旋转模式和矢量模式实现的区别仅在于角度旋转方向的判定(分别依赖于  $z(i), y(i)$  的符号)。

整个 CORDIC 算法的硬件实现可以采用图 4 所示的基本单元复用形成迭代结构或级联形成流水线结构。为了提高运算的效率, 一般采用流水线结构, 在经过  $N$  (迭代次数) 个时钟的建立时间之后, 每隔一个时钟便能输出一个运算结果。

#### (3) 硬件仿真

采用 Xilinx 公司的 ISE9.1 进行硬件仿真。选取向量  $(1/2, 1/2), (1/2, 1/4)$  对 CORDIC 算法的矢量模式进行仿真, 相位  $\pi/4, \pi/3$  对 CORDIC 算法的旋转模式进行仿真。输入、输出用 16 位表示, 仿真结果

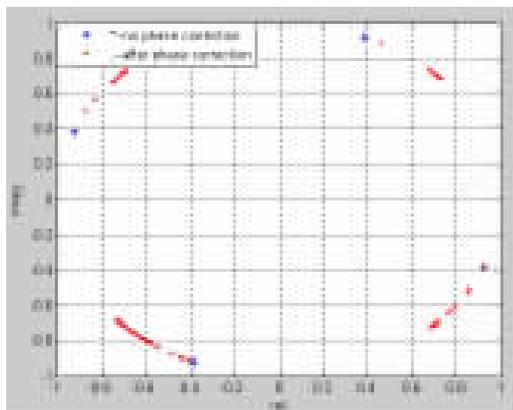


图3 环路相位校正

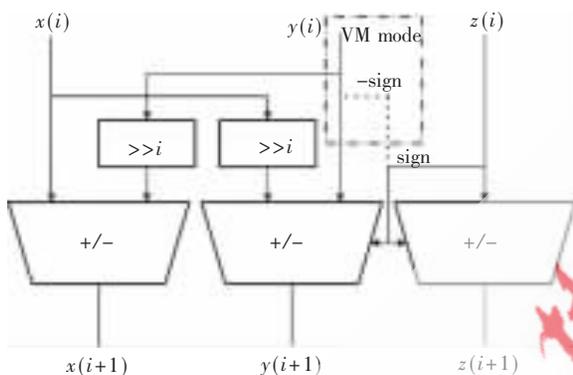


图4 CORDIC 算法基本单元

时序波形如图5所示。

结果分析: 向量  $(1/2, 1/2)$ ,  $(1/2, 1/4)$  理论相角值为  $0.785398, 0.4636476$ , 仿真值为  $6434/2^{13}=0.7854, 3799/2^{13}=0.46374$ ; 向量  $(1/2, 1/2)$  旋转  $\pi/4, \pi/3$  后的理论值为  $(0, 0.7071), (-0.18301, 0.68301)$ , 仿真结果为:  $(0, 0.70709), (-0.18307, 0.68298)$ 。

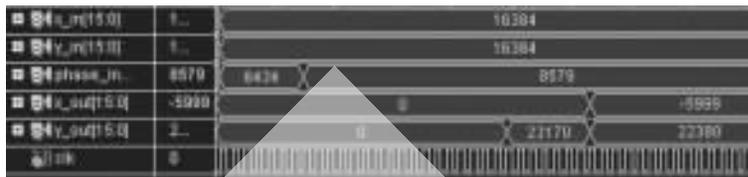
利用CORDIC算法进行载波同步,消耗资源少,运算精度高,可以广泛应用于各种调制模式下的数字接收机设计。

参考文献

[1] JHON G P 著. 数字通信(第4版)[M]. 张力军, 张宗橙,



(a) 矢量模式CORDIC算法



(b) 旋转模式CORDIC算法

图5 CORDIC 算法时序仿真波形

郑宝玉,等译. 北京:电子工业出版社,2003.

[2] YU Hen Hu. CORDIC-based VLSI architectures for digital signal processing[J]. IEEE Signal Processing Magazine, 1992(7):16-35.  
 [3] 郑宇, 沈翥, 李小进, 等. CORDIC算法与闭环控制在OFDM同步校正中的应用[J]. 微电子学与计算机, 2006, 23(12):154-156.  
 [4] VALS J, SANSALONI T, PASCUAL A P. The use of CORDIC in software defined radios: a tutorial[J]. IEEE Communication Magazine, 2006(9):46-50.  
 [5] DICK C, HARRIS F, RICE M. FPGA implementation of carrier synchronization for QAM Receivers[J]. Journal of VLSI Signal Processing, 2004, 36:57-71.

(收稿日期: 2009-11-28)

作者简介:

陈建中,男,1984年生,硕士研究生,主要研究方向:无线通信技术。

聂景楠,男,1962年生,教授,博导,主要研究方向:无线AdHoc网络,传感器网络及移动通信。

张华伟,男,1983年生,硕士研究生,主要研究方向:数据库单向同步。