第27卷 第1期 1998年2月
电子科技大学学报 Journal of UEST of China
Vol.27 No.1 Feb. 1998

快速并行 σ_i 预测 CORDIC 算法

范 勇* 王兆明 黄香馥
(电子科技大学电子工程学院 成都 610054)

【摘要】 CORDIC 作为一种计算三角/双曲函数和向量旋转的迭代算法,其硬件结构简单,易于并行化处理和VLSI实现,因而在实时信号处理方面有广泛的应用前景。在CORDIC算法中,旋转迭代方向 σ_i 的快速确定是提高算法运算速度的一个有效方法。文中从CORDIC算法的基本思想出发,提出了一种并行 σ_i 预测算法,直接由输入数据确定迭代方向,同时提高了算法的并行化程度,在保证精度的情况下能大大缩短CORDIC迭代算法的运算时间。

关 键 词 CORDIC 算法; 迭代方向因子; 并行 σ_i 预测; 编码变换
中图分类号 TN941.1

CORDIC(Coordinate Rotation Digital Computer)算法是一种可用于许多基本算术运算、数学函数计算和向量旋转的迭代算法。Volder 于1959年首先提出了 CORDIC 算法解决三角函数、乘法、除法的计算以及数据类型的转换问题[1]。其后,由 Walther 于1971年将其推广到双曲函数的计算[2]。CORDIC 算法的主要优点在于其完全摒弃了费时的乘法器进行的乘法运算,而只需进行简单的移位和加减运算。这种结构特别适合于用大规模集成电路(VLSI)实现,可极大地提高数字运算的速度,因此特别引人注目。近年来,许多科学工作者对算法作了进一步的研究和改进,并应用于各种信号处理技术,特别是用于实时信号处理领域中[3]。例如,在一些阵列信号处理应用中,与常规的方案相比,CORDIC 具有更高的性能,因其特别适于广义的向量旋转变换[4]。本文基于CORDIC算法的基本思想提出的并行 σ_i 预测算法在一定条件下能有效地提高CORDIC的运算时间,同时易于并行处理。

1 CORDIC 算法描述

CORDIC 算法的迭代过程可用递归或者流水线结构实现,迭代方程为[1]

$$x_{i+1} = x_i - m\sigma_i 2^{-S(m,i)} y_i \tag{1}$$

$$y_{i+1} = y_i + \sigma_i 2^{-S(m,i)} x_i \tag{2}$$

$$z_{i+1} = z_i - \sigma_i \alpha_{m,i} \tag{3}$$

式中 整数 m 为+1,或-1,或0,分别表示圆坐标、双曲坐标、线性坐标系;$\{S(m,i)\}$ 为一递增的、由 m 唯一确定的整数序列,并满足以下条件

$$S(m,i) \leqslant S(m,i+1) \leqslant S(m,i)+1$$

一般情况下取 $S(m,i)=i$;$\{\sigma_i\}$ 表示(坐标)旋转的方向,为一仅包含+1和-1的符号序列。$\alpha_{m,i}$ 表示(坐标)旋转的角度,其值取决于 $S(m,i)$ 和 m 并满足

* 1997年5月5日收稿,1997年5月24日修改定稿
* 男 29岁 博士生

$$\alpha_{m,j}=\frac{1}{\sqrt{m}}\tan^{-1}(\sqrt{m}\,2^{-S(m,i)})=\begin{cases}\tanh^{-1}(2^{-S(m,j)}) & m=-1\\ 2^{-S(m,i)} & m=0\\ \tan^{-1}(2^{-S(m,i)}) & m=+1\end{cases}\tag{4}$$

，易

在CORDIC算法中有两种可能的运作模式:旋转模式和向量模式。旋转方向因子 σ_i 由下列方程确定:

$$\sigma_i=\begin{cases}\operatorname{sign}[z_i] & z_i\to 0(\text{旋转模式})\\ -\operatorname{sign}[x_i]\operatorname{sign}[y_i] & y_i\to 0(\text{向量模式})\end{cases}$$

其中 $\operatorname{sign}(x)=\begin{cases}1 & x\geqslant 0\\ -1 & x<0\end{cases}$，迭代结果如表1所示。 gi

表1 CORDIC迭代运算结果

模式	双曲坐标($m=-1$)或圆坐标($m=+1$)	线性坐标($m=0$)
$z_n\to 0$	$x_n=K_m[x_0\cos(\sqrt{m}z_0)-\sqrt{m}y_0\sin(\sqrt{m}z_0)]$	$x_n=x_0$
(旋转)	$y_n=K_m[y_0\cos(\sqrt{m}z_0)+\sqrt{m}x_0\sin(\sqrt{m}z_0)]$	$y_n=y_0+x_0z_0$
$y_n\to 0$	$X_n=K_m\sqrt{x_0^2+my_0^2}$	$x_n=x_0$
(向量)	$z_n=z_0+\frac{1}{\sqrt{m}}\tan^{-1}(\sqrt{m}y_0/x_0)$	$z_n=z_0+y_0/x_0$

表中，x_0、y_0、z_0 为迭代的初值，K_m 为比例因子：$K_m=\prod_{i=1}^{n}\sqrt{1+m\sigma_i^2 2^{-2S(m,i)}}$。

CORDIC算法属于数字—数字算法，因此它有线性的收敛域和序列的特性，亦即是：对于字长为 N 的向量计算，$(N+1)$次迭代即可保证足够的精度，而第$(i+1)$次迭代必须在第 i 次迭代完成之后进行，这一特性直接影响了CORDIC的运算速度。此外，方程(1)~(3)的迭代运算不能使比例因子 $K_m\equiv 1$，迭代的最后结果需进行校正，这样也降低了CORDIC的速度。由方程(1)~(3)可以看出，在迭代过程中决定运算速度最主要的因素是加/减运算。在非冗余数字系统中，进位的传输使加/减运算的时间下限通常与字长 n 成正比，这样使CORDIC的速度大大降低，因为在每一次迭代中旋转方向的确定都需要全精度的计算，冗余CORDIC的提出对解决这一问题作出了重大贡献[4]。采用冗余CORDIC算法可以减少每一步的基本运算时间：1）采用不需进位传输的冗余加法；2）采用残数估算取代完全计算的方法来确定旋转的方向。

2 并行 σ_i 预测算法

对于一般结构的CORDIC算法，其运算速度受加/减运算的进位传输限制而较慢。这不仅是因为加法器运算时间长，而且因为CORDIC的迭代方向由中间结果的符号位唯一确定，而在最高有效位(MSB)计算出来之前符号位是未加的。对于冗余CORDIC算法，用数字集{-1, 0, 1}表示二进制数的快速冗余加法克服了加/减运算受进位传输限制的缺点，加快了运算的速度。旋转迭代方向 σ_i 也是用整数子集{-1, 0, 1}来表示，即允许 σ_i 为零并将其作为一个有效的选择，但在迭代过程中不进行任何操作，从而减少了整个运算的迭代次数。为保证比例因子 K_m 为常数，补偿是必须的，具体算法见文献[5]。虽然用冗余数字系统结构能有效地提高CORDIC的运算速度，但是迭代方向 σ_i 的判定问题依然没有得到解决：1）冗余数字的符号位是隐式的；2）有序的符号判别不利于并行化处理。因此，我们提出并行 σ_i 预测算法来解决这一问题。

并行 σ_i 预测算法的主要思想是直接从旋转角度 z_0 推导出方向集{σ_i}。对于线性坐标系统($m=0$)，旋转模式($z_i\to 0$)的迭代结果为

$$y_n=y_0+z_0x_0\qquad(y_0,\ x_0,\ z_0\text{ 为迭代初值})$$

其中初值 z_0 用二进制表示为　$z_0=\sum_i Z_i 2^{-1}$　$Z_i\in\{0,1\}$

可以证明，在冗余数字系统中，z_0 可以表示为

$$z_0=\sum_i \sigma_i 2^{-i+1}\qquad \sigma_i\in\{-1,0,1\}$$

例如，设初值 $z_0=0.855\ 468\ 75$，即

$$z_0=2^{-1}+2^{-2}+2^{-4}+2^{-5}+2^{-7}+2^{-8}=2^0+(-1)\times2^{-2}+2^{-3}+(-1)\times2^{-5}+2^{-6}+(-1)\times2^{-8}$$

与此相对应，$\{Z_i\}=\{1,1,0,1,1,0,1,1,0\}$，$\{\sigma_i\}=\{1,0,-1,1,0,-1,1,0,-1\}$。这就意味着在二重旋转迭代过程中，旋转方向集$\{\sigma_i\}$的确定过程可以等效为对二进制码集$\{Z_i\}$的编码变换过程，按照一定的变换规则，可以直接从 z_0 得到全部的 σ_i。这一过程可以并行处理，因此采用这一方法可以有效地提高运算的速度，下面给出变换方法。

设 $z_0=0, Z_1, Z_2, Z_3, \cdots Z_{i-1}, Z_i\cdots Z_n$。则$\{Z_i\}\to\{\sigma_i^m\}$的编码规则如表 2 所示。

表 **2**　位编码规则 **1**

Z_{i-1}	Z_i	σ_i^m
0	0	0
0	1	1
1	0	-1
1	1	0

表 **3**　位编码规则 **2**

σ_{i-1}^m	σ_i^m	σ_{i-1}	σ_i
1	-1	0	1
-1	1	0	-1
其他		σ_{i-1}^m	σ_i^m

利用上面的例子，$\{Z_i\}\to\{\sigma_i^m\}$的变换过程为($\bar{1}=-1$)：

$$\begin{array}{cccccccccc}
Z_i & 0. & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0\\
 & & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow & \Downarrow\\
\sigma_i^m & & 1. & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1
\end{array}$$

为了进一步提高算法的运算速度，我们对$\{\sigma_i^m\}$再进行一次编码变换，规则如表 3 所示。

仍利用上面的例子，$\{\sigma_i^m\}\to\{\sigma_i\}$的变换过程为($\bar{1}=-1$)：

$$\begin{array}{cccccccccc}
\sigma_i^m & 1. & 0 & \bar{1} & 1 & 0 & \bar{1} & 1 & 0 & \bar{1} & 0\\
 & & \Downarrow & & \Downarrow & & \Downarrow & \Downarrow & & \Downarrow & \Downarrow\\
\sigma_i & 1. & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0
\end{array}$$

这样，我们用 σ_i 预测算法使给定迭代运算的迭代次数减少了至少 50%以上。

以上讨论的是线性坐标系统($m=0$)下的情况，而在圆及双曲坐标系统($m=\pm1$)中，情况有所不同。这是因为 z_0 的分解比较复杂，此时

$$z_0=\sum_i\sigma_1\alpha_{m,i}=\sum_i\sigma_i\frac{1}{\sqrt{m}}\tan^{-1}(\sqrt{m}\,2^{-2S(m,i)})$$

由于 $\alpha_{m,i}\neq2^i$，不能直接运用以上规则。但是随着 i 的增加，2^i 与 $\alpha_{m,i}$之差变小。因此，如果对预测的误差及时进行补偿调整，以上的变换规则仍然可以适用。下面我们推导出预测误差的上界。

CORDIC 算法的收敛条件是：在第 i 次迭代结束后，剩余旋转角度 z_i 的绝对值不大于下次将要迭代的实际旋转角度的两倍，即：$|z_i|\leqslant2\alpha_{m,i}$。因此，如果利用我们的编码变换规则作近似处理，则在第 j 次到第 k 次的迭代过程中，近似误差就必须小于或等于 $\alpha_{m,k}$。在这种情况下，如果第

k 次迭代后误差大于 $\alpha_{m,k}$，重复该次迭代即可使 $|z_k| \leqslant 2\alpha_{m,k}$，从而保CORDIC算法收敛。考虑下式

$$\sum_{i=j}^{k}\left|2^{-i}-\frac{1}{\sqrt{m}}\tan^{-1}(\sqrt{m}\,2^{-S(m,\,i+1)})\right| \leqslant \frac{1}{\sqrt{m}}\tan^{-1}(\sqrt{m}\,2^{-S(m,\,k)})$$

将不等式的两边用极数展开，得

$$\sum_{i=j}^{k}\left|2^{-i}-(2^{-S(m,\,i+1)}+\frac{1}{3}m2^{-3S(m,\,i+1)}-\frac{1}{5}m^2 2^{-5S(m,\,i+1)}+\cdots)\right|$$

$$\leqslant 2^{-S(m,\,k)}-\frac{1}{3}m2^{-3S(m,\,k)}+\frac{1}{5}m^2 2^{-5S(m,\,k)}-\cdots$$

设 $S(m,i)=i$，$m=1$，忽略高阶误差，可得

$$\frac{1}{3}\sum_{i=j}^{k}2^{-3i} \leqslant 2^{-k}\text{，或近似为：} k \leqslant 3j+1.5$$

上式表明，确定 σ_i 的全并行位编码变换规则适用于从第 j 次开始到第 k 次（$=3j$）的迭代过程，随后进行一次补偿调整以校正可能有的误差。此调整迭代与上次迭代相似，但旋转方向 σ_i 可能不同。

3 结 论

冗余数字的系统结构能有效地提高CORODIC算法的运算速度。本文讨论的并行 σ_i 预测算法，能快速得到迭代方向 σ_i，并且可以并行处理，从而进一步提高了旋转模式冗余CORDIC的运算速度。

参 考 文 献

1 Volder J E. The CORDIC trigonometric computing technique. IRE Trans Electron Comput, 1959, EC-8(3):330～334

2 Walter J S. A unified algorithm for elementary functions. In AFIPS Spring Joint Computer Conference, 1971, 38:379～385

3 David H, Meyr H. VLSI implementation of the CORDIC algorithm using redundant arithmetic. Proc ISCAS'92, 1992:1 089～1 092

4 Ercegovec M D, lang T. Redundant and on-line CORDIC: application to matrix triangulation and SVD. IEEE Trans on Comput, 1990, C-39(6):725～740

5 Takagi N, Asada T, Yajima S. Redundant CORDIC methods with a constant scale factor for sine and cosine computation. IEEE Trans on Comput, 1991, 40:989～995

Fast Parallelizing σ_i Prediction CORDIC Algorithm

Fan Yong Wang Zhaoming Huang Xiangfu

(College of Electronic Eng., UEST of China Chengdu 610054)

Abstract The CORDIC algorithm is well known as an efficient method for the computation of trigonometric/hyperbolic functions and vector rotations. The CORDIC iteration is directed by the rotation direction factor σ_i. Therefore the fast determination of σ_i is important to accelerate the speed of CORDIC processors. σ_i determination in redundant number system is not straightforword because an exact sign detection is not possible efficiently. Therefore, an predict algorithm of the original CORDIC is proposed which parallelize the σ_i determination and reduce the latency time of CORDIC algorithm considerably.

Key words coordinate rotation digital computer algorithm; rotation direction factor; parallel σ_i prediction; recording conversion

编辑 叶 红