

DOI:10.11918/202002068

双向预判免缩放因子CORDIC算法

邹家轩^{1,2}, 揭 灿², 王 栋², 晏承荣², 程雪峰²

(1. 西安电子科技大学 微电子学院, 西安 710000; 2. 中国电子科技集团第五十八研究所, 江苏 无锡 214000)

摘要: 针对流水线型坐标旋转数字计算机 (coordinate rotation digital computer, CORDIC) 算法的输出精度低、输出时延长、硬件资源消耗大的问题, 提出一种双向预判免缩放因子CORDIC算法。该算法首先将 $[0, \pi/4)$ 内的输入角度通过角度二进制编码后按位值 i 分解为 2^{-i} 的较小角度, 然后使用设立的查找表在初始角度的基础上进行双向免缩放因子旋转, 无需根据中间迭代结果判断下次的旋转方向, 避免了迭代方向的不确定性, 减少了迭代单元级数和迭代次数, 同时提高了运算精度; 最后将 $[\pi/4, 2\pi)$ 内的输入角度通过角度区间折叠技术变换到 $[0, \pi/4)$, 使计算区间扩展到整个圆周 $[0, 2\pi)$, 保证了运算范围, 且只需要使用移位和加减运算即可实现, 避免了进行乘法运算。在MATLAB和Vivado软件平台上进行算法仿真与验证, 结果表明: 在输出位宽均为14位时, 与流水线型和单向免缩放因子型CORDIC算法相比, 输出精度分别提高了47.5%、18.8%, 最大输出时延分别降低了53.8%、40.0%, 硬件资源消耗也有一定的改善。本文提出的CORDIC算法具有输出精度高、输出时延短等特点, 其综合性能具有较大的提升。

关键词: 坐标旋转数字计算机; 角度区间折叠; 双向预判免缩放因子; 现场可编程门阵列; 数字信号处理

中图分类号: TN492

文献标志码: A

文章编号: 0367-6234(2021)02-0047-06

Bi-directional prediction scaling-free CORDIC algorithm

ZOU Jiaxuan^{1,2}, JIE Can², WANG Dong², YAN Chengrong², CHENG Xuefeng²

(1. School of Microelectronics, Xidian University, Xi'an 710000, China;

2. The 58th Research Institute of China Electronics Technology Group Corporation, Wuxi 214000, Jiangsu, China)

Abstract: To solve the problems that pipelined coordinate rotation digital computer (CORDIC) algorithm has low output precision, long output time, and large hardware resource consumption, this paper proposes a bi-directional prediction scaling-free CORDIC algorithm. First, the algorithm decomposes the input angle in $[0, \pi/4)$ into the smaller angle of 2^{-i} according to the bit value i after binary encoding of the angle. Then it uses the set-up lookup table to perform two-way scaling-free factor rotation based on the initial angle. It is not necessary to judge the next rotation direction according to the intermediate iteration result, which avoids the uncertainty of the iteration direction, reduces the number of iteration units and iterations, and improves the calculation accuracy. Finally, the algorithm transforms $[\pi/4, 2\pi)$ to $[0, \pi/4)$ through the angle interval folding technique, so that the calculation interval is extended to the entire circumference $[0, 2\pi)$, which guarantees the calculation range. In addition, the algorithm can be achieved by only using shift and addition and subtraction operations, avoiding multiplication operations. Simulation and verification were carried out on MATLAB and Vivado. Results show that compared with the pipelined and unidirectional scaling-free CORDIC algorithms, when the output bit width was 14 bits, the output accuracy was increased by 47.5% and 18.8% respectively, the maximum output delay was reduced by 53.8% and 40.0% respectively, and the hardware resource consumption was improved to some extent. The algorithm has the characteristics of high output accuracy and short output delay, and its comprehensive performance has been greatly improved.

Keywords: coordinate rotation digital computer; angle interval folding; bi-directional prediction scaling-free factor; field programmable gate array; digital signal processing

坐标旋转数字计算机 (CORDIC) 算法的基本思想是运用旋转迭代等方法, 采用运算基数角度不断逼近目标角度^[1], 只使用简单的移位和加减操作,

即可计算三角函数、反三角函数和双曲函数等超越函数^[2-3]。该算法具有易于实现、可移植性强、精度可调等特点^[4], 常应用于数字频率合成、数字图形处理、机器人学、矩阵计算等领域, 因而受到众多科研院校的广泛研究^[5-6]。

为了适应于现代数字系统的需要, 开发高性能的CORDIC算法, 众多学者对其提出了改进方案。文

收稿日期: 2020-02-23

基金项目: 国家自然科学基金(61704161); 中央军委装备发展部“十三五”微电子预研(31513010412)

作者简介: 邹家轩(1982—), 男, 博士, 高级工程师

通信作者: 揭 灿, jieciddawm123@163.com

献[7]中传统的流水线型 CORDIC 算法虽然具有一定的精度,但所需的迭代单元级数和迭代次数较多,且需要通过上级迭代剩余角度的符号判断下级旋转迭代的方向,当输出位宽增大时,硬件资源消耗将大幅增加.文献[8]中提出的混合型 CORDIC 算法虽然可以快速调节旋转角度基数,但是其以较多的迭代次数来保证输出精度,且调整方程系数的电路结构复杂,存在硬件资源消耗过多的问题.文献[9]中设计的免迭代 CORDIC 算法采用合并迭代结构将多次迭代合并为一次迭代旋转,具有输出时延短的特点,但是运算精度却有待提高.文献[10]中基于离散傅里叶变换的并行补偿型 CORDIC 算法对缩放因子进行补偿计算,具有一定的输出精度,但是电路结构复杂,需要消耗额外的硬件资源用于补偿和存放缩放因子.文献[11]中提出的基于直接旋转的 CORDIC 算法虽然避免了缩放因子的计算,但是其只能逆时针旋转逼近输入角度,仍然具有较大的迭代运算量.

本文在文献[11]的基础上对 CORDIC 算法进行优化与改进,提出了双向预判免缩放因子 CORDIC 算法,其通过角度二进制编码后的码值即可预判每次的旋转迭代方向.此算法将输入角度分解为两个阶段处理,在第 1 阶段中利用查找表一步旋转到初始角度,然后在第 2 阶段中根据二进制编码后码值中 1 的个数,选择恒定顺时针或逆时针方向进行免缩放因子旋转迭代,减少了迭代单元级数和迭代次数,提高了输出精度,最后通过角度区间折叠方法将运算角度覆盖到整个圆周,确保了运算范围.

1 双向预判免缩放因子旋转

1.1 角度二进制编码

对于输入角度为 $[0, \pi/4)$ 内的角度 θ_0 ,采用 n 位二进制小数表示为

$$\theta_0 = \sum_{k=1}^n b_k 2^{-k}, \quad (1)$$

式中 b_k 为 0 或 1.将 θ_0 前 4 位按位展开后为

$$\theta_0 = b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + b_4 2^{-4} + \sum_{k=5}^n b_k 2^{-k}. \quad (2)$$

当式(2)中 b_k 的值为 1 时,则需要进行逆时针旋转 2^{-k} 弧度;当 b_k 为 0 时,则不需要进行任何角度的旋转^[12].经过角度编码后的旋转向量无需根据上次的迭代结果判断下次的旋转方向,避免了迭代方向的不确定性.若式(1)中 k 增大时,则 $\tan(2^{-k}) - 2^{-k}$ 的绝对值越来越小,且当 $k \geq n/3$ 时,有 $|\tan(2^{-k}) - 2^{-k}| < 2^{-n}$,因此 $\tan(2^{-k})$ 与 2^{-k} 之间的

误差小于 n 位二进制小数能够表示的精度,计算误差可以忽略不计.则可用 2^{-k} 近似表示 $\tan(2^{-k})$,只需要通过移位和加减运算即可实现,避免进行乘法运算.

1.2 建立小容量正余弦值查找表

本文采用 13 位二进制数表示 $[0, \pi/4)$ 内的输入角度,而 $[\pi/4, 2\pi)$ 内输入角度的正弦和余弦函数值可以通过三角函数公式变换后获得,则只需要输入 $[0, \pi/4)$ 内的角度值即可.此时 n 值为 13,带入公式(2)可知,当 $k < 13/3$ 时,不可用 2^{-k} 近似表示 $\tan(2^{-k})$,否则将会引入较大的计算误差,因此建立小容量正余弦值查找表(见表 1),用以保证输出精度,且所消耗的寄存器也较少.

表 1 小容量正余弦值查找表

Tab. 1 Lookup table for small capacity sine and cosine values

前 4 位 位值	初始角度 η/rad	二进制 $\cos \eta$ 值	二进制 $\sin \eta$ 值
0000	0	01000000000000	00000000000000
0001	1/16	00111111110000	00000100000000
0010	2/16	00111111000000	00000111111101
0011	3/16	00111110110000	00001011111011
0100	4/16	00111110000001	00001111110101
0101	5/16	00111100111001	00010011101011
0110	6/16	00111011100011	00010111011100
0111	7/16	00111001111110	00011011000111
1000	8/16	00111000010101	00011110101011
1001	9/16	00110110001001	00100010001001
1010	10/16	00110011111001	00100101011100
1011	11/16	00110001011101	00101000100111
1100	12/16	00101110110101	00101011101000
1101	13/16	00101100000010	00101110011101

根据 $[0, \pi/4)$ 内输入角度的前 4 位分别为 0000 到 1101,且后 9 位全为 0 时的初始角度 η ,建立 14 个 14 位的正余弦值查找表,表中将 $\cos \eta$ 和 $\sin \eta$ 值的输出位宽扩展 1 位,用于减小计算中的截断误差.

1.3 角度区间折叠

对于流水线型 CORDIC 算法,虽然随着迭代次数的增多,其运算范围也随之增加,但是当迭代次数趋近于无穷大时,最大运算范围接近 $[-1.744, 1.744]$ rad,而无法覆盖整个圆周^[13].但正弦函数与余弦函数都具有对称性,因此可以将 $[0, 2\pi)$ 的角度划分为 8 个区间,利用三角公式变换将整个圆周内的角度折叠映射到 $[0, \pi/4)$ 内,角度区间转化如表 2 所示.

表 2 角度区间转化

Tab. 2 Angle interval conversion

角度区间 编码值	区间角度 θ/rad	折叠后角 度 θ_1	$\cos \theta$	$\sin \theta$
000	$[0, \pi/4)$	θ	$\cos \theta_1$	$\sin \theta_1$
001	$[\pi/4, \pi/2)$	$\pi/2 - \theta$	$\sin \theta_1$	$\cos \theta_1$
010	$[\pi/2, 3\pi/4)$	$\theta - \pi/2$	$-\sin \theta_1$	$\cos \theta_1$
011	$[3\pi/4, \pi)$	$\pi - \theta$	$-\cos \theta_1$	$\sin \theta_1$
100	$[\pi, 5\pi/4)$	$\theta - \pi$	$-\cos \theta_1$	$-\sin \theta_1$
101	$[5\pi/4, 3\pi/2)$	$3\pi/2 - \theta$	$-\sin \theta_1$	$-\cos \theta_1$
110	$[3\pi/2, 7\pi/4)$	$\theta - 3\pi/2$	$\sin \theta_1$	$-\cos \theta_1$
111	$[7\pi/4, 2\pi)$	$2\pi - \theta$	$\cos \theta_1$	$-\sin \theta_1$

在本文的双向预判免缩放因子型 CORDIC 算法的实现过程中, 首先计算 $[0, \pi/4)$ 内的正弦值和余弦值, 然后比较 3 位角度区间编码值, 最后通过角度区间折叠转换到不同的角度区间, 输出相应的运算结果. 3 位编码值中的次高位和第 3 位相同与否决定了余弦值与正弦值是否需要进行交换, 如果相同则无需交换, 否则需要进行交换; 余弦值和正弦值是否需要翻转取反取决于编码值中的第 3 位, 若第 3 位的值为 0 时, 则无需翻转取反; 但当第 3 位为 1 时, 则进行翻转取反输出; 若最高位的值为 0, 则正弦值输出为正值, 否则为负值; 当最高位与次高位不同时, 则余弦值输出为负值, 反之为正值. 采用角度区间折叠技术, 保证了运算范围, 同时减少了查找表所需要存储的正余弦值的个数.

1.4 双向免缩放因子旋转

下面首先介绍免缩放因子旋转的基本原理. 在流水线型 CORDIC 算法中, 将正弦和余弦函数用泰勒级数展开式表示可得

$$\sin \theta_0 = \theta_0 - \frac{\theta_0^3}{6} + \frac{\theta_0^5}{120} - \dots, \quad (3)$$

$$\cos \theta_0 = 1 - \frac{\theta_0^2}{2} + \frac{\theta_0^4}{24} - \dots. \quad (4)$$

令式(3)和(4)中的 $\theta_0 = 2^{-i}$, 则 $\sin \theta_0$ 中的三次项可以表示为

$$\frac{\theta_0^3}{6} = \frac{2^{-3i}}{6} = 2^{-(3i+\log_2 6)} = 2^{-(3i+2.59)}. \quad (5)$$

当正弦函数和余弦函数的输出位宽设定为 N 位, 且最高位为符号位时, 则式(5)中 i 的最大值为 $N-1$. 若 $3i+2.59 \geq N-1$, 可得 i 的取值范围为

$$\frac{N-3.59}{3} \leq i \leq N-1, i \in N^*. \quad (6)$$

当 i 的取值在式(6)范围之内时, 则 θ_0^3 及其更高阶的函数均已无法用 $L-1$ 位二进制表示, 此时可以直接取 $\cos \theta_0$ 值为 1 进行计算^[14].

单向免缩放因子旋转可以省略缩放因子的计算, 且迭代方向与角度二进制编码后的位值相对应, 避免了流水线型需要不断顺时针和逆时针旋转修正角度的问题, 同时也不会因为去掉了缩放因子而影响输出精度^[15]. 然而当角度二进制编码后码值中相应位为 0 时, 虽然不需要进行旋转迭代, 但是并不能保证下一个码值在此位的值还是为 0, 且只能进行恒定逆时针免缩放因子旋转, 所以需要消耗一个时钟周期进行此位的旋转判断, 无法跳跃此位为 0 时的旋转判断, 需要较多的迭代单元级数和次数以保证输出精度. 但本文设计的 CORDIC 算法既可以恒定逆时针或顺时针免缩放因子旋转, 也可以跳跃不需要旋转的角度, 减少电路的迭代次数, 提高了运算精度, 其双向旋转示意图 1.

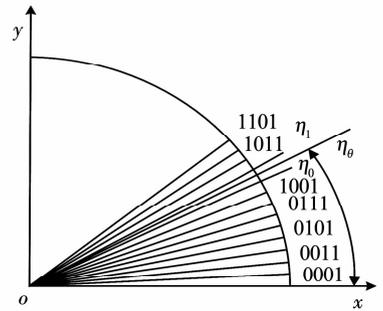


图 1 双向旋转示意

Fig. 1 Schematic diagram of bidirectional rotation

在上图中, 根据表 1 中设立的 14 个初始角度值将 $[0, 13/16)$ 弧度范围内的角度均分为 13 等份, 每一份弧度都为 $1/16$. 当由初始角度 η_0 逆时针旋转到目标角度 η_θ 时, 则 η_θ 可以表示为

$$\eta_\theta = \eta_0 + b_5 \left(\frac{1}{2}\right)^5 + b_6 \left(\frac{1}{2}\right)^6 + \dots + b_{13} \left(\frac{1}{2}\right)^{13}.$$

令 $\eta_\theta = \eta_1 + \eta_k$, 同时由于 $\eta_1 = \eta_0 + \left(\frac{1}{2}\right)^4$, 可得

$$\eta_k = (b_5 - 1) \left(\frac{1}{2}\right)^5 + \dots + (b_{13} - 1) \left(\frac{1}{2}\right)^{13} - \left(\frac{1}{2}\right)^{13}.$$

则由 η_0 和 η_1 旋转到目标角度的迭代次数分别为

$$\lambda_0 = b_5 + b_6 + \dots + b_{13}, \quad (7)$$

$$\lambda_1 = 10 - (b_5 + b_6 + \dots + b_{13}) = 10 - \lambda_0. \quad (8)$$

通过式(7)和(8)可知, 当 $\lambda_0 \geq \lambda_1$ 时, 则 $\lambda_0 \geq 10 - \lambda_0$, 得 $\lambda_0 \geq 5$, 因此当输入角度二进制编码后码值中 1 的个数 ≥ 5 时, 此时用 η_1 旋转迭代的次数比 η_0 少; 当 $\lambda_0 < \lambda_1$ 时, 则 $\lambda_0 < 10 - \lambda_0$, 得

$\lambda_0 < 5$, 若二进制编码后码值中 1 的个数 < 5 时, 此时用 η_0 旋转迭代的次数比 η_1 少. 因此与单向免缩放因子型和流水线型 CORDIC 算法相比, 迭代单元的级数分别减少了 4 级和 8 级, 降低了电路的硬件资源消耗. 采用 13 位二进制小数表示 $\pi/4$ rad 可为 1100100011110, 对于图 1 中 0000 到 1100 这 12 个分区和 1100 到 $\pi/4$ 区间而言, 根据相应的迭代次数对应的种类, 单向免缩放因子型 CORDIC 算法在初始角度 η_0 的基础上进行恒定逆时针旋转时, 由排列组合相关知识可得每个分区的迭代次数 L_d 和总迭代次数 N_{unid} 分别为:

$$L_d = \sum_{j=1}^m jC_m^j = m \times 2^{m-1}, d=1, \dots, 12;$$

$$L_{13} = mC_m^3 - 4m^2 + 31m - 49 + \sum_{r=5}^{m-3} r[C_{m-1}^r + C_{m-4}^{r-1} - C_{m-8}^{r-5}];$$

$$N_{\text{unid}} = \sum_{c=1}^{13} L_c, c=1, \dots, 13.$$

上式中, m 为建立 4 位查找表后剩余的二进制编码的位数, 其值为 $n - 4$. 当根据码值中 1 的个数合理选择 η_0 和 η_1 进行双向预判免缩放因子旋转迭代时, 对 m 进行分奇偶数讨论后, 可得在单向免缩放因子型基础上减少的总迭代次数为

$$N_{\text{dec_sum}} = \sum_{l=t_0}^{m-1} tC_{\frac{m+1}{2}+\frac{l}{2}}^{\frac{m+1}{2}+\frac{l}{2}} \times 12 + \sum_{h=l_0}^{m-7} h[C_{\frac{m+1}{2}+\frac{h}{2}}^{\frac{m+1}{2}+\frac{h}{2}} + C_{\frac{m-1}{2}+\frac{h}{2}}^{\frac{m-1}{2}+\frac{h}{2}} - C_{\frac{m-8}{2}+\frac{h}{2}-5}^{\frac{m-1}{2}+\frac{h}{2}-5}] + m^2 - 5m + 2.$$

$$\begin{cases} t = 2k, t_0 = 2, h = 2l, 1 \leq l \leq \frac{m-7}{2}, \\ 1 \leq k \leq \frac{m-1}{2}, m \text{ 为奇数}; \\ t = 2k-1, h = 2l-1, t_0 = 1, 1 \leq k \leq \frac{m}{2}, \\ 1 \leq l \leq \frac{m-6}{2}, m \text{ 为偶数}. \end{cases}$$

对于最高位为符号位的 N 位流水线型 CORDIC 算法, 其总迭代次数为

$$N_{\text{pipel}} = (N - 1) \times 2^{(N-1)}.$$

相比于单向免缩放因子型和流水线型 CORDIC 算法的总迭代次数, 其降低率分别为

$$I_{\text{dec_pipel}} = \frac{N_{\text{dec_sum}}}{N_{\text{unid}}} \times 100\%,$$

$$I_{\text{dec_unid}} = \frac{N_{\text{pipel}} - (N_{\text{unid}} - N_{\text{dec_sum}})}{N_{\text{pipel}}} \times 100\%.$$

将 $m=9$ 和 $N=14$ 代入上式计算后可得, 总迭代次数降低率分别为 15.9% 和 76.8%. 总迭代次数的降低减少了旋转迭代运算过程中误差的累积, 提高了算法的输出精度.

2 算法实现

本文提出的双向预判免缩放因子 CORDIC 算法的整体结构框图见图 2. 根据结构框图, 在 MATLAB 工具上进行算法建模与仿真, 并与单向免缩放因子型和流水线型 CORDIC 算法进行精度对比与分析, 同时采用 Verilog HDL 硬件描述语言对各模块以及整体电路进行自顶向下的 RTL 级设计, 最后在 Xilinx 公司的 XC7X200t-3fbg484 型号 FPGA 上进行电路实现, 比较分析 3 种算法的硬件资源消耗等性能参数.

3 仿真结果与分析

在相同的 14 位输出位宽的情况下, 使用 MATLAB 软件对 3 种算法建模分析. 输入角度以 2^{-13} 的分辨率遍历第一象限, 并且与标准正余弦函数分别相减后得到其绝对误差, 同时在绝对误差图中标识最大值点和最小值点, 然后将取绝对值后的绝对误差进行算术平均以获得平均误差, 最后采用相应的函数分别计算出标准差与方差, 比较分析 3 种算法的输出精度. 第一象限内 3 种算法的余弦函数绝对误差结果如图 3 所示, 其精度对比见表 3, 并进行了精度对比分析.

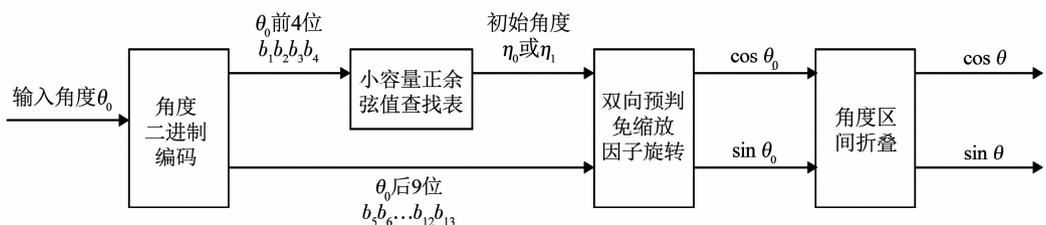


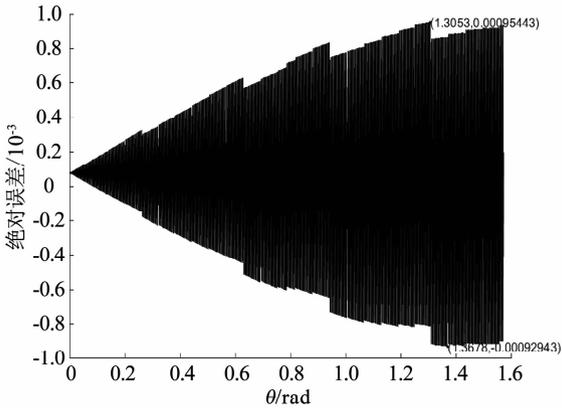
图 2 新型 CORDIC 算法整体结构框图
Fig. 2 Block diagram of CORDIC algorithm

表 3 3 种算法余弦函数的精度对比

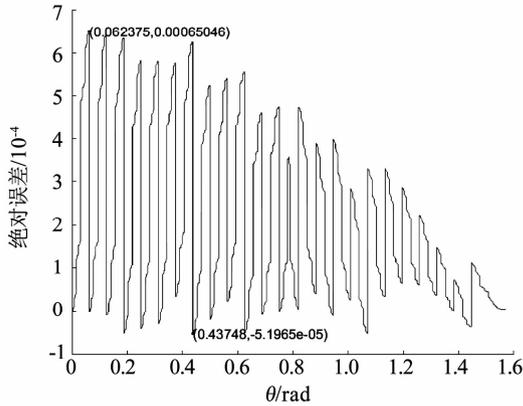
Tab.3 Accuracy comparison of cosine function of three algorithms

10⁻⁴

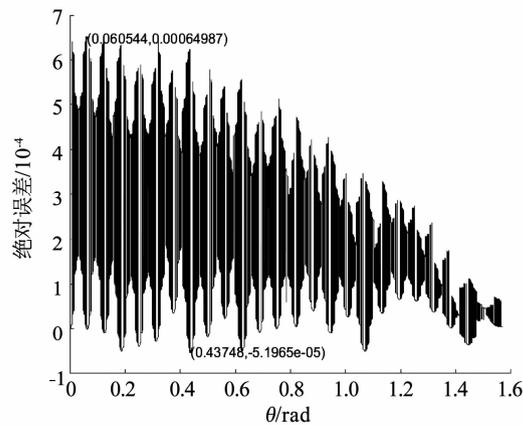
CORDIC 算法类型	绝对误差的最大值	平均误差	标准差	方差
流水线型	9.544 3	3.176 4	2.459 2	6.047 8
单向免缩放因子型	6.504 6	2.055 2	1.943 6	3.777 8
双向预判免缩放因子型	6.498 7	1.668 5	1.774 6	3.149 1



(a) 流水线型



(b) 双向预判免缩放因子型



(c) 单向免缩放因子型

图 3 3 种算法余弦函数的绝对误差

Fig.3 Absolute errors of cosine function of three algorithms

通过图 3 与表 3 可以看出,与流水线型和单向免缩放因子型相比,本文提出的 CORDIC 算法的平均误差分别降低了 47.5% 和 18.8%,绝对误差的最

大值以及取绝对值后绝对误差的方差和标准差都具有一定的降低,提高了算法的输出精度.主要原因在于即使流水线型 CORDIC 算法中的旋转向量旋转到了目标角度,但如果未完成 13 次迭代,则还需要顺时针或者逆时针旋转不同角度,直到完成 13 次迭代,因此输出误差较大.单向免缩放因子型 CORDIC 算法虽然避免了不必要的旋转迭代,但是其迭代次数还是较多,运算过程中产生的累积误差较大.而改进算法能够根据角度二进制编码后码值中 1 的个数,在准确的初始角度的正余弦值基础上,进行恒定顺时针或逆时针免缩放因子旋转迭代,减少了总迭代次数,降低了运算过程中误差的累积,提高了算法的输出精度.

在相同的时序、面积和电路环境属性等约束条件下,使用 Vivado 软件对 3 种算法进行综合,对寄存器消耗量、查找表消耗量等硬件资源消耗参数,以及最大输出时延进行对比分析,逻辑综合结果如表 4 所示.同时,在表 5 中列出了本文提出的双向预判免缩放因子型 CORDIC 算法与近年来国内外部分 CORDIC 算法文献的精度对比.

表 4 三种算法的逻辑综合结果

Tab.4 Logical synthesis results of three algorithms

CORDIC 算法类型	寄存器消耗量/个	查找表消耗量/个	最大输出时延/周期
流水线型	713	806	13
单向免缩放因子型	396	657	10
双向预判免缩放因子型	328	591	6

表 5 CORDIC 算法的精度对比

Tab.5 Accuracy comparison of CORDIC algorithms

文献	绝对误差的最大值	平均误差/10 ⁻⁴	最大输出时延/周期
文献[9]		1.809 5	12
文献[10]	7.220 8 × 10 ⁻⁴	1.973 6	9
文献[11]	1.390 0 × 10 ⁻³	3.900 0	10
本文	6.498 7 × 10 ⁻⁴	1.668 5	6

从表 4 中可以看出本文提出的 CORDIC 算法相比于流水线型和单向免缩放因子型 CORDIC 算法,硬件资源消耗具有一定的下降.虽然本文的算法增

设了双向预判选择单元,消耗了少量的寄存器和查找表,但是迭代单元级数由流水线型的 13 级和单向免缩放因子旋转型的 9 级缩减到了 5 级,使硬件资源消耗有所下降.从表 5 中可知,与文献[9-11]相比,CORDIC 算法的平均误差分别降低了 7.8%、15.5%、57.2%,同时在绝对误差的最大值和最大输出延时方面也都具有较大幅度的降低.

4 结 论

针对流水线型 CORDIC 算法的运算精度低、输出时延长等问题,本文结合角度二进制编码和角度区间折叠等手段,建立了小容量正弦值查找表,提出并实现了双向免缩放因子 CORDIC 算法.提出的 CORDIC 算法对于某一输入目标角度,根据角度二进制编码后码值中 1 的个数,选择初始角度并预判迭代旋转方向,避免了每次旋转迭代方向的不确定性,减少了迭代单元级数和迭代次数,最短在两个时钟周期内即可输出运算结果.在相同的 14 位位宽的情况下,对比流水线型和单向免缩放因子型 CORDIC 算法的运算精度,则分别提高了 47.5%、18.8%,最大输出时延分别降低了 53.8%、40.0%,算法实现电路的寄存器和查找表的消耗量也有所下降,接下来还可以采用格雷码对角度区间码值进行编码,降低相邻状态转换时出错的可能性.本文设计的 CORDIC 算法在输出精度与输出时延等方面具有显著的优势,适用于实时性强、高精度、低功耗的现代数字通信系统.

参 考 文 献

- [1] HOANG T T, LE D H, PHAM C K. Minimum adder-delay architecture of 8/16/32-point DCT based on fixed-rotation adaptive CORDIC[J]. IEICE Electronics Express, 2018, 15(10): 20180302. DOI: 10.1587/elex.15.20180302.
- [2] 姚亚峰,邹凌志,王巍,等.低消耗免查找表 CORDIC 算法[J].哈尔滨工业大学学报,2017,49(11):109
YAO Yafeng, ZOU Lingzhi, WANG Wei, et al. Low-consumption and LUT-omitted CORDIC algorithm[J]. Journal of Harbin Institute of Technology, 2017, 49(11): 109. DOI: 10.11918/j.issn.0367-6234.201704019
- [3] NGUYEN H T, HUYEN X T, PHAM C K. A low power hybrid adaptive CORDIC[J]. IEEE Transactions on Circuits and Systems, 2018, 65(4): 496
- [4] RAMADOSS R, KERMANI M M, AZARDERAKHSH R. Reliable hardware architectures of the CORDIC algorithm with a fixed angle of rotations[J]. IEEE Transactions on Circuits and System II: Express Briefs, 2017, 64(8): 972
- [5] 王国洪,宛强,姚亚峰,等.精确频率输出的超低时延 DDS 电路设计[J].哈尔滨工业大学学报,2019,51(5): 44
WANG Guohong, WAN Qiang, YAO Yafeng, et al. A DDS circuit design with ultra-low latency and exact output frequency [J]. Journal of Harbin Institute of Technology, 2019, 51(5): 44. DOI: 10.11918/j.issn.0367-6234.201805108
- [6] 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. DOI: 10.1109/TVLSI.2015.2445855
- [7] HUANG Hai, XIAO Liyi. CORDIC based fast radix-2 DCT algorithm [J]. IEEE Signal Processing Letters, 2013, 20(5): 483
- [8] SHUKLA R, RAY K C. Low latency hybrid CORDIC algorithm[J]. IEEE Transactions on Computers, 2014, 63(12): 3066
- [9] YAO Yafeng, FENG Zhongxiu. BBR-based iteration-free CORDIC algorithm[J]. Journal of Circuits Systems and Computers, 2018, 27(5): 101
- [10] KULSHRESHTHA T, DHAR A. CORDIC-based Hann windowed sliding DFT architecture for real-time spectrum analysis with bounded error-accumulation[J]. IET Circuits, Devices & Systems, 2017, 11(5): 487
- [11] 姚亚峰,冯中秀,陈朝.直接旋转 CORDIC 算法及其高效实现[J].华中科技大学学报(自然科学版),2016,44(10): 113
YAO Yafeng, FENG Zhongxiu, CHEN Zhao. Direct rotation CORDIC algorithm and its implementation[J]. Journal of Huazhong University of Science and Technology (Natural Science Edition), 2016, 44(10): 113. DOI: 10.13245/j.hust.161022
- [12] PRASAD N, TRIPATHY M R, DAS A D, et al. Efficient VLSI implementation of CORDIC based direct digital synthesizer[C]// Intelligent Computing, Communication and Devices. New Delhi: Springer, 2015, 308: 597
- [13] KAMBO H M A D, KHAN S A. IS-CORDIC: a fixed-point encoded single iteration CORDIC architecture [J]. International Journal of Electronics, 2014, 101(6): 789
- [14] AGUIRRE-RAMOS F, MORALES-REYES A, CUMPLIDO R, et al. An area efficient composed CORDIC architecture[J]. Advances in Electrical and Computer Engineering, 2014, 14(2): 113
- [15] MEHER P K, PARK S Y. CORDIC designs for fixed angle of rotation[J]. IEEE Transactions on Very Large Scale Integration Systems, 2013, 21(2): 217

(编辑 苗秀芝)