Journal of Southeast University (English Edition) Vol. 25, No. 3, pp. 313 - 315 Sept. 2009 ISSN 1003—7985

New scale factor correction scheme for CORDIC algorithm

Dai Zhisheng Zhang Meng Gao Xing Tang Jiajian

(National ASIC System Engineering Research Center, Southeast University, Nanjing 210096, China)

Abstract: To overcome the drawbacks such as irregular circuit construction and low system throughput that exist in conventional methods, a new factor correction scheme for coordinate rotation digital computer (CORDIC) algorithm is proposed. Based on the relationship between the iteration formulae, a new iteration formula is introduced, which leads the correction operation to be several simple shifting and adding operations. As one key part, the effects caused by rounding error are analyzed mathematically and it is concluded that the effects can be degraded by an appropriate selection of coefficients in the iteration formula. The model is then set up in Matlab and coded in Verilog HDL language. The proposed algorithm is also synthesized and verified in field-programmable gate array (FPGA). The results show that this new scheme requires only one additional clock cycle and there is no change in the elementary iteration for the same precision compared with the conventional algorithm. In addition, the circuit realization is regular and the change in system throughput is very minimal.

Key words: coordinate rotation digital computer (CORDIC) algorithm; scale factor correction; field-programmable gate array (FPGA)

The CORDIC (coordinate rotation digital computer) algorithm first introduced by Volder in 1959 and later generalized by Walther[1] can be used to calculate elementary functions such as trigonometric functions, multiplication, division, data type conversion and hyperbolic functions by using simple hardware such as shifters, adders/subtracters and comparers. It has been widely used in several discrete transformations such as the Hartley transform[2], the discrete cosine transform[3], the fast Fourier transform (FFT) and the Kalman filters[4], etc. It can also detect multiusers in code division multiple access (CDMA) wireless systems[5-6].

The CORDIC algorithm consists of two operating modes, rotation and vectoring, and each mode can work at circular, linear and hyperbolic coordinate systems via iteration. Besides rotation, the vector is also scaled at each iteration during the calculation[7]. In some implementations, the elimination is inevitable since the scale factor may affect the entire system performance.

The conventional method to eliminate the scale factor is to multiply the iteration results by the scale factor. The straightforward way can be easily understood and sometimes it can really work well. However, it introduces at least two negative effects: 1) The introduction of multiplication will destroy the regularity of the iteration; 2) The introduction of multiplication will decrease the frequency.

In this paper we propose an improved method to correct the scale factor. This method requires some additional hardware, but it does not need any changes on the input elements such as rotation angles and rotation directions, which means that this method is fully compatible with the conventional circuit.

1 Review of the Conventional CORDIC Algorithm

As shown in Fig. 1, the rotation of vector $\{x_0, y_0\}^{\mathrm{T}}$ in circular coordination can be described as[1]

$$\begin{Bmatrix} x_1 \\ y_1 \end{Bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} x_0 \\ y_0 \end{Bmatrix} \tag{1}$$

where $\{x_1, y_1\}^{\mathrm{T}}$ is the final result and θ is the rotated angle.

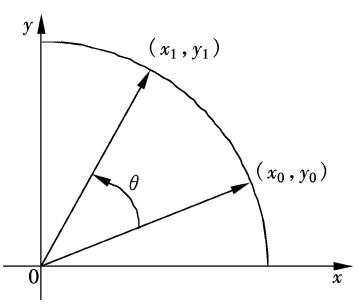

Fig. 1 Vector rotation

To simplify the hardware implementation, the target angle θ can be decomposed into a combination of serial predefined elementary angles, such as

$$\theta = \sum_{i=0}^{n} u_i \alpha_i \tag{2}$$

where n is the number of elementary angles; $\mu_i = \pm 1$ is the rotation sequence which determines the direction of the i-th elementary angle $\alpha_i = \arctan(2^{-i})$. The usage of 2^{-i} makes the operation in the CORDIC be simple shift and add. The steps of rotation are shown in Fig. 2. Based on the above two equations without regard to the scale factor $\cos\theta$, the iteration equations of the CORDIC algorithm can be written as

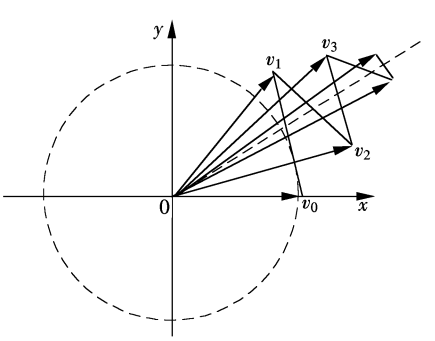

Fig. 2 The steps of rotation

Received 2009-01-19.

Biographies: Dai Zhisheng (1985—), male, graduate; Zhang Meng (corresponding author), male, professor, zmeng@seu.edu.cn.

Foundation item: The National High Technology Research and Development Program of China (863 Program) (No. 2007AA01Z280).

Citation: Dai Zhisheng, Zhang Meng, Gao Xing, et al. New scale factor correction scheme for CORDIC algorithm[J]. Journal of Southeast University (English Edition), 2009, 25(3): 313 - 315.

$$\left.\begin{aligned} x_{i+1} &= x_i - u_i y_i 2^{-i} \\ y_{i+1} &= y_i + u_i x_i 2^{-i} \end{aligned}\right\} \quad i = 0, 1, \ldots, n \tag{3}$$

Using matrix form, Eq. (3) can be rewritten as

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -u_i 2^{-i} \\ u_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{4}$$

Assume that $\boldsymbol{v}_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$ and $\boldsymbol{P} = \begin{bmatrix} 1 & -u_i 2^{-i} \\ u_i 2^{-i} & 1 \end{bmatrix}$, Eq. (4) can be substituted by $\boldsymbol{v}_{i+1} = \boldsymbol{P}\boldsymbol{v}_i$.

The modes mentioned above are determined by the strategy used in the elementary angle direction. To simplify the discussion, only the rotation mode is discussed in this paper, whereas the algorithm can be applied on both modes.

The iteration number of Eq. (3) depends upon the desired accuracy of the result. The scale factor after n iterations is given by

$$K_n = \prod_{i=0}^{n-1} \cos(\arctan(2^{-s_i})) = \prod_{i=0}^{n-1} \sqrt{1 + 2^{-2s_i}} \tag{5}$$

where s_i is the shift sequence.

2 Proposed Scale Factor Correcting Algorithm

According to the standard CORDIC algorithm mentioned above, for the convenience of analysis, the iteration equation is rewritten as

$$\boldsymbol{v}_{i+1} = \boldsymbol{P}\boldsymbol{v}_i \tag{6}$$

In principle, the scale factor correction is carried out by simplifying the multiplication into an ordinary shift-and-add operation for the convenience of digital realization. Therefore, the scale factor is decomposed into

$$\frac{1}{K_n} = \sum_{i=0}^{n-1} a_i \qquad a_i \in \{0, 2^{-i}\} \tag{7}$$

And the vector after compensation is

$$\boldsymbol{v}'_n = \boldsymbol{v}_n \sum_{i=0}^{n-1} a_i \tag{8}$$

From Eqs. (6) and (8), we can conclude that

$$\boldsymbol{v}'_{r+1} = \boldsymbol{v}_{r+1} \sum_{i=0}^{r} a_i = \boldsymbol{P}\boldsymbol{v}_r \Big(\sum_{i=0}^{r-1} a_i + a_r \Big) = \boldsymbol{P}\boldsymbol{v}'_r + \boldsymbol{v}_{r+1} a_r$$

So far a new iteration is introduced for scale factor correction and it is rewritten as

$$\boldsymbol{v}'_{r+1} = \begin{bmatrix} 1 & -u_i 2^{-i} \\ u_i 2^{-i} & 1 \end{bmatrix} \boldsymbol{v}'_r + \boldsymbol{v}_{r+1} a_r$$

3 Numerical Analysis

As for the quantization effects, the approximation error is the same as that of the conventional algorithm, but the rounding error is not. Upper bounds for these errors are derived in Ref. [8]. The rounding error analysis in Ref. [8] can be extended for the modified algorithm.

Assume that $\boldsymbol{v}'_e(i)$ and $\boldsymbol{v}_e(i)$ are the results after the i-th iteration if finite precision arithmetic is used. $\hat{\boldsymbol{v}}'(i)$ and $\hat{\boldsymbol{v}}(i)$ are the results before quantization after the i-th iteration. $\boldsymbol{f}'(i)$ and $\boldsymbol{f}(i)$ are the error vectors,

$$\boldsymbol{v}'_e(i) = \boldsymbol{v}'(i) + \boldsymbol{f}'(i), \quad \boldsymbol{v}_e(i) = \boldsymbol{v}(i) + \boldsymbol{f}(i)$$

$\boldsymbol{e}'(i)$ and $\boldsymbol{e}(i)$ are error vectors induced by quantization after the i-th iteration, and they are upper bounds by δ,

$$\boldsymbol{v}'_e(i) = \hat{\boldsymbol{v}}'(i) + \boldsymbol{e}'(i), \quad \boldsymbol{v}_e(i) = \hat{\boldsymbol{v}}(i) + \boldsymbol{e}(i)$$

From Ref. [9], we can induce that

$$|\boldsymbol{f}'(n)| \leqslant \{1 + |a_{n-1}| + \sum_{i=0}^{n-1} \prod_{j=i}^{n-1} \|\boldsymbol{P}\| (1 + A'_i)\}\delta$$

where $A'_i = \sum_{r=i-1}^{n-1} |a_r|$.

From the above result, it is obvious that to minimize the accumulated quantization, a_i should be decreased when i increases. This can be achieved by selecting $a_i \in \{0, 2^{-i}\}$.

4 Implementation

According to section 2, the architecture to implement the proposed algorithm is shown in Fig. 3. At the first clock, the values in Reg1 and Reg2 are $x(0)$ and $y(0)$, respectively. At the second clock, the outputs of Reg1 and Reg3 are $x(1)$ and $x(0)$, and the three inputs of CSA are $x(0)$, $a_0 x(0)$ and $\mu_0 2^{-s_0} y(0)$ where μ_0 is the initial rotation direction. At the third clock, the output of Reg3 is $x'(1)$.

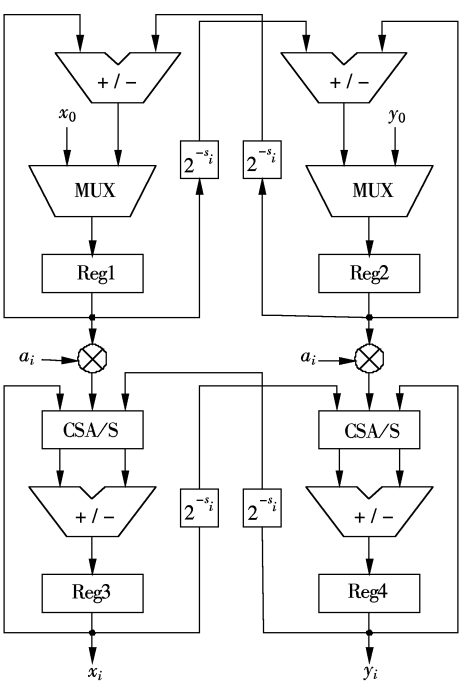

Fig. 3 Iteration process of the proposed algorithm

After n iterations are carried out, the outputs of the CORDIC algorithm are x_i and y_i. Compared with the conventional algorithm, only one additional clock cycle is required. The algorithm is described by Verilog HDL language and implemented in Altera EP2S90F1020C5 FPGA for 2's complement arithmetic, and synthesized by Quartus Ⅱ[10].

The sine value calculated via VCS is shown in Fig. 4 and then exchanged to decimal data by Matlab. The true value vs. the calculated value from VCS is shown in Fig. 5. We

can find that they are very close to each other.

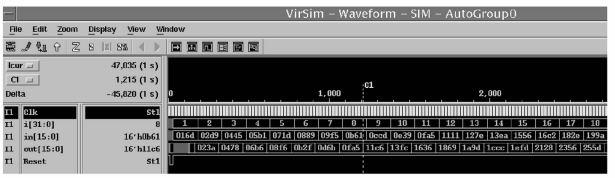

Fig. 4 Sine value calculation

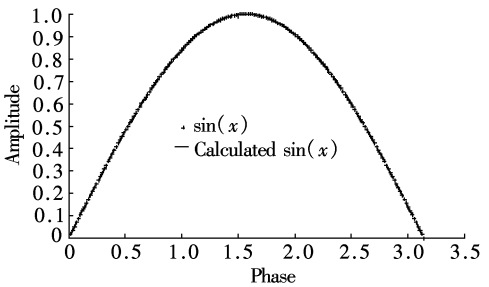

Fig. 5 The calculated value vs. the true value

In Ref. [9], Ahmed proposed that the scale factor can be operated as 2's integral power by a series of iterations. This application requires no additional hardware and is simple in implementation, but it can obtain the same precision. No doubt, more iterations will be required.

The resource usage and the maximum clock frequency are compared with the conventional CORDIC algorithm in Tab. 1[11]. The proposed algorithm occupies more hardware than the standard method, but it is easier to update to higher precision since the elementary iteration needs no change.

Tab. 1 Resource consumption

Algorithm	Resource		Maximum clock frequency/MHz
	Combinational ALU	Total registers	
Conventional algorithm	488	500	304.97
Proposed algorithm	972	1017	299.43

5 Conclusion

A new method for correcting the scale factor of the CORDIC algorithm is proposed and the FPGA implementation is carried out. According to the simulation, only one additional iteration is required compared with the conventional algorithm without the scale factor correction unit for the same precision, and the performance is improved.

References

[1] Walther J S. A unified algorithm for elementary functions [C]//*Spring Joint Computer Conference*. Boston, USA, 1971, **38**: 379 – 385.

[2] Chang L W, Lee S W. Systolic arrays for the discrete Hartley transform [J]. *IEEE Trans Signal Processing*, 1991, **39**(11): 2411 – 2418.

[3] Maharatna K, Dhar A S, Banerjee S. A VLSI array architecture for realization of DFT, DHT, DCT and DST [J]. *Signal Proc*, 2001, **81**(9): 1813 – 1822.

[4] Sima M, McGuire M, Iancu D, et al. CORDIC scenario for Kalman-based channel estimation [C]//*Proceedings of the IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*. Victoria, Canada, 2005: 165 – 168.

[5] Wu Cheng Shing, Wu An Yeu, Lin Chih Hsiu. A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes [J]. *IEEE Trans on Circuits and Systems—Ⅱ: Analog and Digital Signal Processing*, 2003, **50**(9): 589 – 601.

[6] Williams L, Takala J. Rotation-based adaptive equalizer architecture for high data rate CDMA systems [C]//*Proc IEEE International Symposium Signal Process Information Tech*. Rome, Italy, 2004: 18 – 21.

[7] Maharatna K, Banerjee S, Grass E, et al. Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture [J]. *IEEE Trans on Circuits and Systems for Video Technology*, 2005, **15**(11): 1463 – 1474.

[8] Hu Yuhen. The quantization effects of the CORDIC algorithm [J]. *IEEE Trans on Signal Processing*, 1992, **40**(4): 834 – 844.

[9] Ahmed H M. Signal processing algorithm and architectures [D]. Palo Alto: Information Systems Laboratory of Stanford University, 1982.

[10] Chu Pong P. *FPGA prototyping using Verilog examples* [M]. New York: Wiley Interscience, 2008: 48 – 51.

[11] Andraka R. A survey of CORDIC algorithms for FPGA based computers [C]//*Proceedings of the 6th International Symposium on FPGAs*. Monterey, USA, 1998: 191 – 200.

用于 CORDIC 算法的一种新的模校正方法

戴志生　张　萌　高　星　汤佳健

（东南大学国家 ASIC 工程技术研究中心，南京 210096）

摘要：提出一种新的纠正 CORDIC 算法中模因子的方法以解决传统方法所带来的电路结构不规则、系统吞吐率降低等弊端. 首先根据迭代方程之间的关系，通过推导引入一个新的迭代方程，将模因子的校正过程转化为只需要移位和加法运算即可实现的简单的迭代过程. 然后分析了该算法量化误差中的舍入误差所带来的影响，并提出该误差可以通过对迭代方程中的系数进行合适取值来降低. 最后对提出的算法通过 Matlab 建模并利用 Verilog HDL 语言进行 RTL 级编程，经过综合后在 FPGA 上进行了验证. 仿真结果表明，与传统方法相比，在相同精度条件下使用所提方法只需要额外的一个时钟周期即可达到模校正的目的，且不需要修改基本的迭代操作. 因此电路实现比较规则，同时系统吞吐率变化较小.

关键词：CORDIC 算法；模校正；现场可编程门阵列

中图分类号：TP391