ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

Volume 3, Special Issue 3, March 2014

2014 International Conference on Innovations in Engineering and Technology (ICIET'14)
On 21st & 22nd March Organized by

K.L.N. College of Engineering, Madurai, Tamil Nadu, India

Modified Scaling-Free Micro-Rotation Based Circular CORDIC Algorithm Using Taylor Series Expansion of Sine and Cosine

Arunnehru S, Paramasivam C
Department of Electronics and Communication Engineering, K. S. Rangasamy College of technology, Tiruchengode, Namakkal, Tamilnadu, India.

***Abstract*—**This paper Proposes the Scaling-free Micro-rotation based CORDIC algorithm that completely eliminates the scale factormultiplication. By employing the Taylor series expansion of sine and cosine in circular trajectory, the proposed method removes the multiplication of the scale factor. The accuracy of the result is based on the selection of the order of approximation of Taylor series, proposed method takes third order approximation which meets the accuracy requirement and attains the desired range of convergence. Also we have suggested an algorithm that redefine the elementary angles with high speed most significant-1 detection for reducing the number of CORDIC iterations. Compared to conventional CORDIC algorithm the proposed algorithm has less time delay and utilizes less number of slices on the Xilinx Vertex XC4VFX12 device.

***Index Terms*—**Co-ordinate Rotation Digital Computer (CORDIC), Cosine/Sine, Field Programmable Gate Array (FPGA), Most Significant-1, Recursive Architecture.

I. INTRODUCTION

The Co-ordinate rotation digital computer (CORDIC) was introduced by Volder [1] in 1959 that performs rotations (to compute sine, cosine, and arctangent functions), multiply or division of numbers by using only shift and add elementary steps. Walther [2] generalized CORDIC algorithm in 1971 to compute logarithms, exponential, and square roots. After that CORDIC has been applied in several important domains of application like generation of sine andcosine functions, calculation of fast Fourier transform (FFT), discrete sine/cosine transforms (DST/DCT), householder transform (HT), etc.. [3], [4].

Many modifications have been done for efficient implementation of CORDIC with less number of iterations over the conventional CORDIC algorithm [5]-[13]. In [5] and [6] efficient scale-factor compensation techniques areproposed which affects the latency/throughput of computation. Greedy search is used in [7]-[9] to optimize the number of iterations at the price of additional area, andtime for the implementation of variable scale-factor.

In [10] area and time efficient CORDIC architectures have been proposed which involve constant scale-factor multiplication for adequate range of convergence (RoC). The virtually scaling-free CORDIC in [11] also requires multiplication by constant scale-factor and relatively more area to achieve respectable RoC. The enhanced scaling-free CORDIC in [12] combines few different types of CORDIC iterations, so it degrades performance. The parallel scaling free approach is proposed in [13] at the expense of loss in accuracy.The proposed recursive architecture has less area complexity with other existing scaling-free CORDIC algorithms, and no scale-factor multiplications needed for extending the RoC to entire coordinate space, as required in [10]-[12].

The rest of this paper is organized as follows. In Section II, a brief recap of the rotation mode of the conventional CORDIC algorithm in a circular coordinate system and existing scaling-free CORDIC algorithm is reviewed. Section III presents the proposed modified scaling-free CORDIC algorithm. Section IV discusses the proposed CORDIC architecture. Section V gives the detailof the FPGA implementation and comparison with conventional method. Section VI concludes this paper.

II. BRIEF REVIEW OF THE CONVENTIONAL AND THE EXISTING SCALING- FREE CORDIC

The CORDIC algorithm operates in two computing modes, either in VECTORING mode or ROTATION mode. These two modes follows linear, circular or

hyperbolic coordinate trajectory. In the ROTATION mode, the co-ordinate components of a vector and an angle of rotation are given and the co-ordinate components of the original vector, after rotation through the given angle, are computed. In vectoring mode, co-ordinate components of a vector are given and the magnitude and angular argument of the original vector are computed. In this paper, we focus on rotation mode CORDIC using circular trajectory.

A. Conventional CORDIC Algorithm

The CORDIC algorithm is able to rotate a given vector by an angle "θ". In conventional rotation mode CORDIC, the (i+1)th intermediate rotated vector estimated from the ith vector using circular trajectory as follows,

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \cos\theta \begin{bmatrix} 1 & -\tan\theta \\ \tan\theta & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2)$$

The angle of rotation "θ" is decomposed into a sequence of fixed predefined elementary rotations with variable direction instead of single rotation.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = K_i \begin{bmatrix} 1 & -\tan\alpha_i \\ \tan\alpha_i & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3)$$

Where $K_i = \cos\alpha_i$

$$\alpha_i = \tan^{-1} 2^{-i} \quad (4)$$

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = K_i \begin{bmatrix} 1 & -d_i 2^{-i} \\ d_i 2^{-i} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (5)$$

The sign sequence d_i=1 for anticlockwise rotations and d_i= -1 for clockwise rotations.so,

$$\theta = \sum_{i=1}^{b} d_i \,.\, \alpha_i \quad (6)$$

Where b is the word-length of the machine in bits.

The overall scaling-factor of "b" CORDIC iterations is given by (7). After sufficiently large number of iterations, the scale factor K converges to a constant value K=0.60725.

$$K = \prod_{i=1}^{b} K_i = \prod_{i=1}^{b} 1/\sqrt{1 + 2^{-2i}} \quad (7)$$

In this conventional CORDIC the range of convergence (RoC) is limited to $[-99.99^{\circ}, 99.99^{\circ}]$. RoC can be extended to entire co-ordinate space using the properties of sine and cosine functions, using an extra iteration for full range rotation.

B. Existing Scaling-Free CORDIC algorithm

Conventional CORDIC algorithm suffers from major disadvantages like scale-factor compensation, latency, and optimal identification of micro-rotations. Various scaling free techniques are reviewed in [10]-[13].

The existing techniques of scaling free CORDIC are attempting to completely dispose of the scale-factor, so the final target angle is achieved by rotating the vector in one direction only means that the final target angle is approximated as a pure summation of the elementary angles. Here, the sine and cosine functions

$$\sin\alpha_i = 2^{-i} \quad (8)$$
$$\cos\alpha_i = 1 - 2^{-(2i+1)} \quad (9)$$

This approximation imposes a following restriction on basic shift (minimum possible permissible shifts in the CORDIC iteration is called as basic shift, which is equal to the number of right shifts in the first CORDIC iteration) in the allowed values of iteration index i:

$$(b - 2.585)/3 \leq i \leq b-1 \quad (10)$$

Using (8) and (9) in the functional equation of CORDIC,

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 - 2^{-(2i+1)} & -2^{-i} \\ 2^{-i} & 1 - 2^{-(2i+1)} \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (11)$$

Compare to the conventional CORDIC working equation, this scaling-free CORDIC working equation (11) does not require a scaling operation. The datapath component required for implementing the operation stated in the equation (11) is shown in Fig.1.

Even though the scaling- free CORDIC algorithm provides a better means to get rid of the final scale factor multiplication, its biggest drawback is its extremely low range of convergence.Modified virtually adaptive scaling-free algorithm [11], extends the convergence range to entire co-ordinate space, but it utilizes an adaptive scale-factor.

III. PROPOSED MODIFIED SCALING-FREE CORDIC ALGORITHM

The principal idea behind the development of modified scaling-free CORDIC rotator is to develop a CORDIC rotator algorithm that stretches out the convergence range to entire co-ordinate space. So the proposed design is based on the following two ideas, 1) we used Taylor series expansion of sine and cosine functions to eliminate scaling operation and 2) we proposed a generalized micro-rotation selection technique to extend the adequate range of convergence (RoC) based on the chosen order of approximation of

the Taylor series.

The Taylor expansions of sine and cosine of an angle "α" are given by

$$\sin\alpha = \alpha - (\alpha^3/3!) + (\alpha^5/5!) \quad \text{-……} \quad (12)$$

$$\cos\alpha = 1 - (\alpha^2/2!) + (\alpha^4/4!) \quad \text{-……} \quad (13)$$

A. Expressions for Micro-Rotations Using Taylor Series Approximation and Factorial Approximation

The third order approximation of the Taylor series expansion of sine and cosine functions is used in the CORDIC co-ordinate calculation.

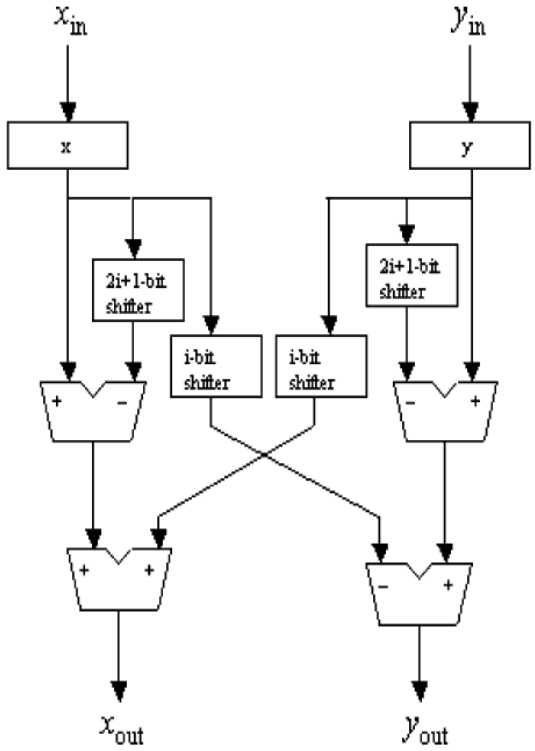

Fig. 1. Elementary rotational section for i> b/2

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 - (\alpha^2/2!) & -(\alpha - (\alpha^3/3!)) \\ \alpha - (\alpha^3/3!) & 1 - (\alpha^2/2!) \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (14)$$

The equation (14) cannot be applied in the CORDIC shift-add iterations. To implement the (14) by shift-add operations, we have to approximate the factorial terms by the power of 2 values, so that replacing 3! by 2^3, and the equation (14) becomes,

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 - (2^{-1}).\alpha_i^2 & -(\alpha_i - (2^{-3}).\alpha_i^3) \\ (\alpha_i - (2^{-3}).\alpha_i^3) & 1 - (2^{-1}).\alpha_i^2 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (15)$$

B. Generalized Micro-Rotation Sequence Selection

The expressions for the basic-shift (s), region of convergence(RoC), first elementary angle of rotation (α_1) for different world-length of different orders of approximations is as follows,

$$\text{Basic-shift, } s = (b - \log_2(n+1)!)/(n+1) \quad (16)$$

Where b is the word length of machine in bits

$$\text{RoC} = n_1 \times \alpha_1 \quad (17)$$

Where $\alpha_1 = 2^{-s}$ and n_1 is the number of micro-rotations

Table I values are computed based on above expressions (16) and (17), and from Table I we know that to increase in the approximation order, the basic-shift decreases, α_1 increases and RoC is extended. For smaller word-lengths the higher order terms of Taylor series do not give any impact on the accuracy.

In the proposed generalized micro-rotation sequence selection, we have done multiple iterations of basic-shift, accompanied by non-repetitive unidirectional iterations corresponding to another shift indices to minimize the number of iterations and achieve an adequate scope of convergence.

In the proposed method the rotation angle is represented as "θ".

$$\theta = I_1.\alpha_s + \sum_{i=1}^{I_2} \alpha_{s_i}, \quad n = I_1 + I_2 \quad (18)$$

Where

$\alpha_s = 2^{-s}$ is the elementary angle corresponding to the basic-shift,

$\alpha_{s_i} = 2^{-s_i}$ are elementary angles for other shifts,

s_i is the shift for *i*th iterations and $s_i > s$,

I_1 and I_2 are non-negative integers and

n represents the total number of iterations.

If we do not use any micro-rotation of angle α_s then I_1 is zero, and I_2 = n. On the other hand, if the desired angle of rotation "θ" is a multiple of α_s then I_2 is zero and I_1 = n. In Table II, we list the decimal and hexadecimal representation of the elementary angles corresponding to different shifts.

TABLE I
ROC FOR DIFFERENT ORDER OF APPROXIMATION OF TAYLOR SERIES BASED ON (16) AND (17)

Approx. order	Basic-shift		First Elementary Angle (Radians)		RoC for n_1 (Radians)	
	16-bit	32-bit	16-bit	32-bit	16-bit	32-bit
3	2	6	0.25	0.01562	1	0.0625
4	1	5	0.5	0.03125	2	0.125
5	1	3	0.5	0.125	2	0.5

TABLE II
SHIFTS AND CORRESPONDING ELEMENTARY ANGLES IN DECIMAL AND HEXADECIMAL REPRESENTATION

Shifts (s_i)	Elementary angle (α_i)	
	Decimal	16-bit Hexadecimal
2	0.25	4000 H
3	0.125	2000 H
4	0.0625	1000 H
5	0.03125	0800 H

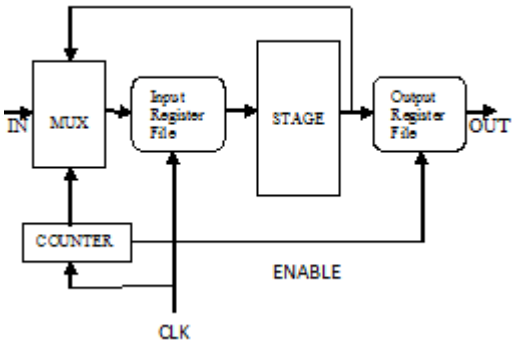

Fig.2. Proposed CORDIC architecture

IV. ARCHITECTURE OF THE PROPOSED CORDIC

The recursive architecture of the proposed CORDIC processor is shown in Fig.2. The proposed CORDIC architecture uses the same stage block for all the iterations of the coordinate calculations, as well as for the generation of shift values.The rollover count of the counter decides the number of iterations required in a CORDIC processor. The rollover count is seven for basic-shift=3 and ten for basic-shift=4. The completion or expiry of the counter signals indicates the completion of a CORDIC operation, depending on this counter signal, the 2:1 multiplexer either loads a new dataset (rotation angle, the initial value of "x" and "y") to start a new CORDIC operation, or recycles the output of the stage to begin a new iteration for the current CORDIC operation. The input and output register files act as latches for synchronization purpose.

The structure of stage block is shown in Fig.3. It consists of three computing blocks,

1. Co-ordinate calculation,
2. Shift-value estimation and
3. Micro-rotation sequence generator

Equation (15) is used for calculating the new coordinate values in each iteration. Recognition of the micro-rotations depending on the bit representation of the rotation angle in radix-2 system using most-significant-1 detection.For this we limit the maximum rotation angle to 45°as the entire coordinate space (0°,360°) can be mapped to the (0°,45°) using octant symmetry of sine and cosine functions. If the most-significant-1 location (M) of the input rotation angle is smaller than the basic-shift "s", the elementary angle of the basic-shift would be applied for the CORDIC iteration. For a fixed word-length of N -bit, the shift (s_i) for the elementary angle is given by

$$s_i = N - M \qquad (19)$$

Based on the above discussion, the pseudo code for micro-rotation sequence generation of 16-bit word-length with basic-shift = 2 is written below.

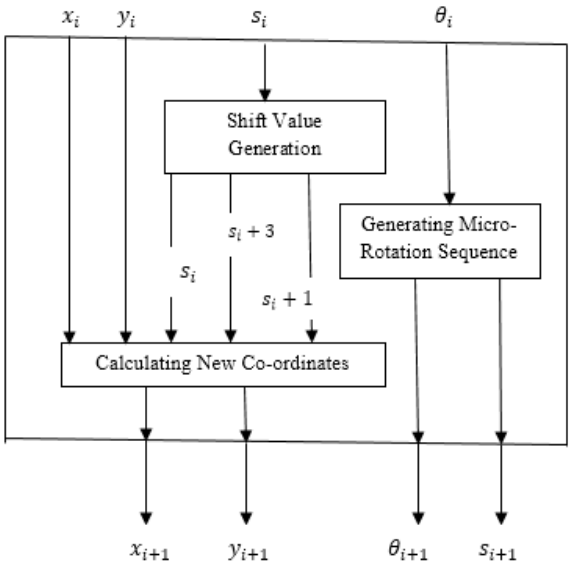

Fig.3. Block diagram of the stage

INPUT: angle to be rotated (θ_i)
Begin
INITIALIZE:i=0
STEP1: Find the most-significant 1 position in the input angle (θ_i)
INITIALIZE: M=most-significant-1 location of (θ_i)
STEP2: If (M==15) then
α=0.25 radians;
Shift, $s_i = 2$ and $\theta_{i+1} = \theta_i - \alpha$;
Increment i;
Go to step 1;
Else
Shift, s_i=16-M
$\theta_{i+1} = \theta_i$ with θ_i[M]=0
Increment i;
Go to step 1;
End

V. FPGA IMPLEMENTATION AND COMPARISION

The proposed modified CORDIC algorithm have been coded in Verilog and simulated and synthesized using Xilinx ISE 13.4i and to be implemented in Xilinx Vertex4 (XC4VFX12-SF363-12) device.TheIsim simulation results of proposed CORDIC are shown in fig. 4. The comparison of synthesis results in Table III shows that Scaling free Micro-Rotation based CORDIC takes less area and

time compare to Conventional Micro-Rotation based CORDIC.

TABLE III
COMPARISON OF SYNTHESIS RESULTS

	CONVENTIONAL CORDIC	PROPOSED SCALING FREE CORDIC
No. of Slices	26%	17%
No. of Slice Flip flop	5%	8%
No. of 4 input LUTs	24%	8%
No. of IOs	52	99
Minimum input arrival time before clock	4.557ns	3.135ns
Maximum output required time after clock	6.280ns	3.793ns
Total Path Delay	7.547ns(49.1% logic, 50.9% route)	3.793ns (93.0% logic, 7.0% route)
Maximum Frequency	233.56MHz	334.353MHz

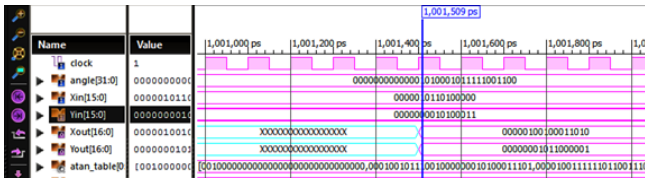

Fig.4. simulation result of proposed CORDIC

VI. CONCLUSION

The proposed algorithm makes a complete scale-free solution for realizing rotation mode circular CORDIC algorithm. The third order approximation of Taylor series is taken by the proposed algorithm and it not only meet the accuracy requirement but also attain an adequate range of convergence. The generalized micro-rotation sequence selection technique is suggested to reduce the number of iterations for low latency implementation. Comparison of synthesis results shows that the proposed modified scaling-free CORDIC algorithm has low time delay and utilized a less number of slices and flip flop on the Xilinx vertex4 device.

REFERENCES

[1] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sep. 1959.

[2] J. S. Walther, "A unified algorithm for elementary functions," in *Proc.Joint Spring Comput. Conf.*, vol. 38, Jul. 1971, pp. 379–385.

[3] K. Maharatna, A. S. Dhar, and S. Banerjee, "A VLSI array architecturefor realization of DFT, DHT, DCT and DST," *Signal Process.* , vol. 81,pp. 1813–1822, 2001.

[4] P. K. Meher, J.Walls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50years ofCORDIC: Algorithms, architectures and applications," *IEEETrans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 9, pp. 1893–1907, Sep.2009.

[5] M. G. B. Sumanasena, "A scale factor correction scheme for theCORDIC algorithm," *IEEE Trans. Comput.*, vol. 57, no. 8, pp.1148–1152, Aug. 2008.

[6] J. Villalba, T. Lang, and E. L. Zapata, "Parallel compensation of scalefactor for the CORDIC algorithm," *J. VLSI Signal Process. Syst.*, vol.19, no. 3, pp. 227–241, Aug. 1998.

[7] C. S. Wu and A. Y. Wu, "Modified vector rotational CORDIC (MVRCORDIC) algorithm and architecture," *IEEE Trans. Circuits Syst. II,Exp. Briefs*, vol. 48, no. 6, pp. 548–561, Jun. 2001.

[8] C.-S.Wu, A.-Y.Wu, and C.-H. Lin, "A high-performance/low-latencyvector rotational CORDIC architecture based on extended elementaryangle set and trellis-based searching schemes," *IEEE Trans. CircuitsSyst. II, Analog Digit. Signal Process.* , vol. 50, no. 9, pp. 589–601, Sep.2003.

[9] Y. H. Hu and S. Naganathan, "An angle recoding method for CORDICalgorithm implementation," *IEEE Trans. Comput.*, vol. 42, no. 1, pp. 99–102, Jan. 1993.

[10] L. Vachhani, K. Sridharan, and P. K. Meher, "Efficient CORDIC algorithmsand architectures for low area and high throughput implementation,"*IEEE Trans. Circuit Syst. II, Exp. Briefs*, vol. 56, no. 1, pp. 61–65,Jan. 2009.

[11] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 11,pp. 1463–1474, Nov. 2005.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[12] F. J. Jaime, M. A. Sanchez, J. Hormigo, J. Villalba, and E. L. Zapata, "Enhanced scaling-free CORDIC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1654–1662, Jul. 2010.

[13] Matteo Causo, Ting An and Lirida Alves de Barros Naviner,"Parallel scaling-free and area-time efficient CORDIC algorithm," *IEEE Trans. Circuits syst.*, December 2012.