Subit Abraham, Puran Gour / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp.597-599

Review on Area Efficient CORDIC Algorithm and Reduction in Scale Factor

Subit Abraham[1], Puran Gour[2]
[1, 2](Department of Electronics and Communications, NIIST College, RGPV University, Bhopal (M.P.)

ABSTRACT

Supriya Aggarwal, Pramod K. Meher, and Kavita Khare proposed an area-time efficient CORDIC algorithm that completely eliminates the scale-factor by microrotation selection which is done by the most significant one detector in the micro rotation sequence generation circuit. Proper order of approximation of Taylor series is selected to meet the accuracy requirement, and the desired range of convergence is achieved. [1]. Francisco J. Jaime *et al.* proposed a new CORDIC algorithm, they show an enhanced version of the scaling-free CORDIC algorithm. Booth recoding algorithm is used to eliminate scaling and domain folding is avoided which reduces scaling to minimum but not eliminate it[2]. Leena Vachhani *et al.* proposed two area-efficient algorithms and their architectures based on CORDIC. Both the algorithms are applicable to the range of angles from 0 to 2 pi , and both the rotation and vectoring modes are used. Xilinx Spartan XC2S200E device is used and have the slice-delay products of about three.[3]. Koushik Maharatna *et al.* proposed a CORDIC rotator algorithm that which has the range of convergence from 0 to 2pi , and the target angle converges by executing appropriate iteration steps while keeping the scale factor virtually constant and completely predictable using the domain folding method. The IHP in-house 0.25-m BiCMOS technology device is used on Synopsys' design analyzer and slice delay product of 26.98 is achieved[4].

Keywords - Coordinate rotation digital computer(CORDIC), rotation mode, scale factor, slice delay product, vectoring mode.

I. INTRODUCTION

COORDINATE Rotation Digital Computer is abbreviated as CORDIC. The key concept of CORDIC arithmetic is based on the simple and ancient principles of two-dimensional geometry. But the iterative formulation of a computational algorithm for its implementation was first described in 1959 by Jack E. Volder [5] for the computation of trigonometric functions, multiplication and division. CORDIC may not be the fastest technique to perform these operations, it is attractive due to the simplicity of its hardware implementation, since the same iterative algorithm could be used for all these applications using the basic shift-add operations of the form $a \pm b.2^{-i}$ .Some of the typical approaches for reduced-complexity implementation are focused on minimization of the complexity of scaling operation the basic principle underlying the CORDIC-based computation, and present its iterative algorithm for different operating modes and planar coordinate systems. [6]

There are two computing modes, ROTATION and VECTORING. In the rotation mode, the components of a vector and an angle of the rotation are given and the coordinate components of the original vector, after rotation through the given angle, are computed. In the second mode VECTORING, the coordinate components of the vector are given and the magnitude and angular argument of the original vector are computed.[5]

In essence , the basic computing technique used in both the ROTATION and VECTORING modes in CORDIC is a step by step sequence of pseudorotations which result in an overall rotation through the given angle (ROTATION) or result in a final angular argument of zero(VECTORING).The algorithm is very attractive for hardware implementation because it uses only shift and add operations to perform vector rotation in the two-dimensional(2-D) plane. The rotation of a two-dimensional vector $p_0 = [x_0 \ y_0]$ through an angle , to obtain a rotated vector $p_n = [x_n \ y_n]$could be performed by the matrix product , $p_n = Rp_0$ where R is the rotation matrix:

$$R=\begin{bmatrix}\cos\theta & -\sin\theta \\ \sin\theta & \cos\theta\end{bmatrix}$$

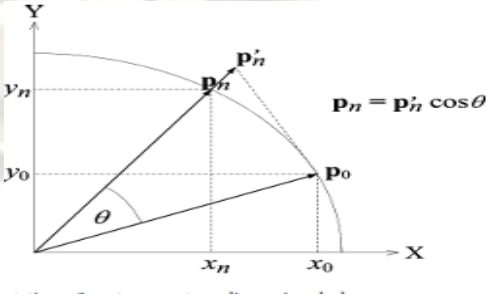

Fig.1 Rotation of vector on a two-dimensional plane

[2]To achieve simplicity of hardware realization of the rotation, the key ideas used in CORDIC arithmetic are to (i) decompose the rotations into a sequence of elementary rotations through predefined angles that could be implemented with minimum hardware cost; and (ii) to avoid scaling, that might

involve arithmetic operation, such as square-root and division. The second idea is based on fact the scale-factor contains only the magnitude information but no information about the angle of rotation.

II. REVIEW TABLE

Cordic parameters	S.Aggarwal P.K. Meher K. Khare	F.Jaime M.Sanchez J.Hormigo J. Villalba E.L. Zapata	L.Vach- hani P.K. Meher K.Sridh aran	K.Mahar-atna S. Banerjee M. Kristic A. Troya E. Grass
Year	2012	2010	2009	2005
Publicati-on	IEEE transactions	IEEE transactions	IEEE transactions	IEEE transactions
Mode of Operation	Rotation	Rotation	Rotation Vectoring	Rotation Vectoring
Scale factor	Elimination	elimination	1 and $1/\sqrt{2}$	elimination
Algorithm used	Taylor series	Scaling free and Booth recoding	ALGO-I ALGO-II	Scaling free adaptive Cordic rotator
Technique	microrotation	Booth algorithm	_	Domain folding
Range of angles	0 to π /4	0 to 2 π	-π to π	0 to2 π
Bit of data	16 bit	16 bit	16 bit	16 bit
Slice delay product	2.77	_	3.42 and 3.34	26.98
Software	Xilinx ISE 9.2i	Synopsys' design compiler	Xilinx ISE 9.2i	Synopsys' design analyzer
Coding language	Verilog	VHDL	Verilog-2001	VHDL
FPGA used	XL2S200E-PQ208-6	_	XL2S200E-PQ208-6	IHP in house 0.25 BiCMOS
Frequency In MHz	58.7	50 to 700	54.35 and 60.80	20

III. REVIEW WORK

Supriya Aggarwal, Pramod K. Meher, and Kavita Khare proposed an area-time efficient CORDIC algorithm that completely eliminates the scale-factor by microrotation selection and mode of operation is rotation. By proper selection Taylor series approximation of sine and cosine functions, the microrotations are determined by using Taylor series, basic shift is determined for the order of approximation it depends on the elementary angles, number of microrotations, and the rotation angle which depends on its entirety on the elementary angles and number of iterations is determined the range of convergence is achieved which is from 0 to π/4. Basic shift is determined for the approximation considerations and all the values are calculated keeping a particular basic shift like 2 and 3 , and the basic shift has to be an integer if the value calculated is not an integer then both the values are considered like if basic shift is 2.85 the basic shift considered will be 2 and 3. The elementary angles are calculated as 2^{-si} where si is shift. The CORDIC processor provides the flexibility to manipulate the number of iterations depending on the accuracy, area and latency requirements. The proposed CORDIC gives a slice-delay product of 2.77 on Xilinx Spartan XC2S200E device.[1]

Francisco J. Jaime, Miguel A. Sánchez, Javier Hormigo, Julio Villalba, and Emilio L. Zapata proposed a new CORDIC by modifying some of the parts of the previously known scaling free CORDIC algorithm, by using Booth recoding algorithm the mode of operation is rotation the scale factor is completely eliminated and this algorithm works in the range of angles from 0 to 2π radians and they obtained new architectures which are able to reach a 35% lower latency and a 36% reduction in area and power consumption compared to the original scaling-free CORDIC architecture. Booth recoding algorithm is used to eliminate scaling and domain folding method is avoided which reduces scaling to minimum but does not eliminate it.[2]

Leena Vachhani, K. Sridharan, and Pramod K. Meher proposed two area-efficient CORDIC algorithms. While ALGO-I (the first algorithm) eliminates ROM and requires only low-complexity barrel shifters and works in both the rotation and vectoring mode, The main idea in first algorithm is that only a small set of angles is chosen for rotation to carry out, multiple and single rotations are carried out, by any angle in order to reduce the size of barrel shifters and contribute further by removing the need for multiple scale factors from multiple rotations. the second algorithm eliminates barrel shifters completely and works in rotation mode only. The algorithms work on the range of angles from -π to π . Finite numbers of rotations are sufficient in the second algorithm to reach within 1degree of the desired angle of rotation to achieve convergence. The Xilinx Spartan XC2S200E device is used for implementation of the algorithms and have a slice-delay product of about 3 is achieved in both the algorithms.[3]

Koushik Maharatna, Swapna Banerjee, Eckhard Grass, Milos Krstic, and Alfonso Troya,

proposed a CORDIC rotator algorithm working in rotation and vectoring mode and domain folding method is used which reduces the scale factor to minimum depending on the input angle the scale factor is 1 and $1/\sqrt{2}$ the CORDIC rotator algorithm is used which converges to the final target angle executing appropriate iteration steps adaptively which keeps the scale factor constant and completely predictable. , and it is independent of the number of executed iterations, nature of iterations, and word length. In the algorithm, the range of angles is from 0 to 2π radians compared to the conventional CORDIC whose range lies between -99 to 99º, and a reduction of 50% iteration is achieved on an average without compromising the accuracy since domain folding method is used to minimize scaling and keep it constant . The binary representation of the target angle is done , and no further arithmetic computation in the angle approximation data path is required. The convergence range of the CORDIC rotator is the entire coordinate space from 0 to 2π radians. Bit shifters are used for the elementary rotation selection, the algorithm works in the case where target angle is not known before. The CORDIC rotator requires 22% less adders and 53% CORDIC rotator core is 0.7 mm^2 and its power dissipation is 7 mW and the scale factor achieved is 26.98, IHP in-house 0.25- m BiCMOS technology hardware is used for implementation using VHDL on Synopsys' design compiler.[4]

IV. PROPOSED CORDIC

Observing all the authors works we propose a Scale Free CORDIC in Hyperbolic mode that is a Scale free hyperbolic CORDIC , The hyperbolic CORDIC was first proposed by John Walther in 1971 [7], in fact it is an extension of the CORDIC given by Volder. Convergence is achieved by repeating iterations [4, 13, 40, 121,.....]. Hyperbolic will converge for only the suitable restricted values of the coordinate components. Using hyperbolic functions of sinh and cosh in hyperbolic trajectory , here we propose to use Taylor series approximation to eliminate scaling in the hyperbolic functions. Here we work in the rotation mode of the algorithm, and a scale free hyperbolic CORDIC processor is designed using VHDL in Xilinx for 16 bit word length. The input values are converted to 16 bit binary since we are designing for 16 bit Scale free hyperbolic CORDIC. The hyperbolic CORDIC is efficient in calculation and implementation in the hardware, many devices can be designed using the hyperbolic scale free processor , the hardware implementation is efficient. The sinh and cosh expansions of the Taylor series is used for the approximation. The basic shift used for the calculations would be similar to the one used for the circular CORDIC for a circular trajectory used in [1].

V. CONCLUSION

All the authors have proposed CORDIC algorithms using Taylor series, Booths recoding algorithm , scaling free algorithm , domain folding all these methods are used to reduce the scale factor or eliminate it and the slice delay product is reduced, the minimum achieved is by Supriya Aggarwal et al., and the area is minimized. Our aim is to reduce the slice delay product and eliminate scaling as well by using Taylor series approximation method for scale free hyperbolic CORDIC in Hyperbolic trajectory. And the wordlength which is considered is 16 bit so all inputs and outputs will be in 16 bits.

REFERENCES

[1] S. Aggarwal, P. K. Meher, K. Khare, "Area Time Efficient scaling free CORDIC using generalized microrotation selection," *IEEE Trans. Very Large Scale Integration (VLSI) System, vol. 20, no.8, pp 1542-1548, Aug. 2012.*

[2] 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.*

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

[4] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling- adaptive CORDIC rotator algorithm and architecture," *IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 11, pp. 1463–1474, Nov. 2005.*

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

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

[7] J. S. Walther, "A Unified algorithm for elementary functions", *in Proc. 38th spring Joint Computer Conf., 1971,pp 379-385*