AREA-TIME EFFICIENT SCALING-FREE CORDIC USING GENERALIZED MICRO-ROTATION SELECTION

Rajalakshmi.S[1], ParthasarathyC.[2], Satheesh Kumar.T[3]
Department of Electronics and Communication Engineering, PRIST University, Tiruchirapalli. India.
[1]srajalakshmiece@gmail.com
[3]kumarsatheesh@in.com
[2]*Department of Mechanical Engineering, PRIST University, Tiruchirapalli. India.*
[2]pristparthasarathy@gmail.com

Abstract— **This paper presents an area-time efficient CORDIC algorithm that completely eliminates the scale-factor. By suitable selection of the order of approximation of Taylor series the proposed CORDIC circuit meets the accuracy requirement, and attains the desired range of convergence. Besides we have proposed an algorithm to redefine the elementary angles for reducing the number of CORDIC iterations. A generalized micro-rotation selection technique based on high speed most-significant-1-detection obviates the complex search algorithms for identifying the micro-rotations. The proposed CORDIC processor provides the flexibility to manipulate the number of iterations depending on the accuracy, area and latency requirements. Compared to the existing recursive architectures the proposed one has 17% lower slice-delay product on Xilinx Spartan XC2S200E device.**

Keywords— **coordinate rotation digital computer (CORDIC), cosine/ sine, field-programmable gate array (FPGA), most-significant-1, recursive architecture.**

I. INTRODUCTION

The coordinate rotation digital computer (CORDIC) has established its popularity in several important areas of application, like generation of sine and cosine functions, calculation of discrete sinusoidal transforms like fast Fourier transform (FFT), discrete sine/cosine transforms (DST/DCT), householder transform (HT), etc. Many variations have been suggested for efficient implementation of CORDIC with less number of iterations over the conventional CORDIC algorithm. The number of CORDIC iterations are optimized in by greedy search at the cost of additional area and time for the implementation of variable scale-factor. In efficient scale-factor compensation techniques are proposed, which adversely affect the latency/throughput of computation. Two area-time efficient CORDIC architectures have been suggested in, which involve constant scale-factor multiplication for adequate range of convergence (RoC). The virtually scale-free CORDIC in also requires multiplication by constant scale-factor and relatively more area to achieve respectable RoC. The enhanced scale-free CORDIC in combines few conventional CORDIC iterations with scaling-free CORDIC iterations for an efficient pipelined CORDIC implementation with improved RoC. However, if used for recursive CORDIC architecture, combining two different types of CORDIC iterations degrades performance. In this paper, we propose a novel scaling-free CORDIC algorithm for area-time efficient implementation of CORDIC with adequate RoC. The proposed recursive architecture has comparable or less area complexity with other existing scaling-free CORDIC algorithms. Moreover, no scale-factor multiplications are required for extending the RoC to entire coordinate space, as required.

II BRIEF OVERVIEW OF CORDIC ALGORITHM

The CORDIC algorithm operates either in, rotation mode or vectoring mode, following linear, circular or hyperbolic coordinate trajectories. In this paper, we focus on rotation mode CORDIC using circular trajectory.

A. Conventional CORDIC Algorithm

In conventional CORDIC to obtain the rotated vector, the angle of rotation "¯ " is decomposed into a sequence of fixed predefined elementary rotations with variable direction. The conventional rotation mode CORDIC estimates the $(i+1)^{th}$ intermediate rotated vector from the i^{th} vector using circular trajectory as

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = K_i \cdot \begin{bmatrix} 1 & -«.\tan\sim_i \\ «.\tan\sim_i. & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

Where $K_i = \cos \alpha_i$

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

The sign sequence μ_i € {1,-1} is so selected that

$$¯ = \sum_{i=0}^{6} «i \cdot \sim i$$

Where, b is the word-length in bits.

The range of convergence of this algorithm is limited to [-99.99°, 99.99°], which can be extended to entire

© 2013 IJAIR. ALL RIGHTS RESERVED

coordinate space using the properties of sine and cosine functions, using an extra iteration for full-range rotation.

The overall scaling-factor of "b" CORDIC iterations. The scaling-factor K, after sufficiently large number of iterations converges to a constant value K = 0.60725.

$$K = \prod_{i=0}^{b} Ki = \prod_{i=0}^{b} 1 \Big/ \sqrt{1+2^{-2i}}$$

B. Review of Existing Scaling Free CORDIC Algorithm

Scaling-free CORDIC was the first attempt to completely dispose of the scale-factor. Here, the sine and cosine functions were approximated to

Sin $\alpha_{i\,=}\,(2^{-i})$ cos $\alpha_{i\,=}\,1\text{-}2^{-(2i+1)}$

However, the approximation imposes a restriction on the basic-shift I = [(b – 2.585)/3]. For 16-bit data, the basic-shift = 4 results in extremely low range of convergence. However, modified virtually adaptive scaling-free algorithm, extends the range of convergence over the entire coordinate space and introduces an adaptive scale-factor.

III. PROPOSED ALGORITHM FOR SCALING FREE CORDIC

The proposed design is based on the following key ideas: 1) we use Taylor series expansion of sine and cosine functions to avoid scaling operation and 2) suggest a generalized sequence of micro-rotation to have adequate range of convergence (RoC) based on the chosen order of approximation of the Taylor series.

A. Taylor Series Approximation of Sine and Cosine Functions

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

$$\sin \alpha = \alpha - (3!)^{-1} \cdot \alpha^3 + (5!)^{-1} \cdot \alpha^5 \ldots\ldots\ldots\ldots$$

$$\cos \alpha = 1 - (2!)^{-1} \cdot \alpha^2 + (4!)^{-1} \cdot \alpha^4 \ldots\ldots\ldots\ldots$$

We have estimated the maximum error in the evaluation of sine and cosine functions for different order of approximations. The maximum percentage of error in sine and cosine functions for third order approximation is 0.0033% and 0.0168%, respectively, within the permissible CORDIC elementary angles range of [0, 7π/88]. Therefore, we choose third order of approximation for Taylor's expansion of sine and cosine functions.

1) Representation of Micro-Rotations Using Taylor Series Approximation:

The impact of orders of approximation of Taylor series of sine and cosine functions on the micro-rotations to be used in CORDIC coordinate calculation. Both theoretical and simulation results are discussed to confirm the appropriate selection of the order of approximation. Using different orders of approximation of sine and cosine functions, we can have

$$x_{i+1} = \left(1 - \frac{\sim_i^2}{2!}\right) . x_i - \left(\sim_s - \frac{\sim_i^3}{3!}\right) y_i$$

$$y_{i+1} = \left(1 - \frac{\sim_i^2}{2!}\right) . y_i - \left(\sim_s - \frac{\sim_i^3}{3!}\right) x_i$$

$$x_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!}\right) . x_i - \left(\sim_s - \frac{\sim_i^3}{3!}\right) y_i$$

$$y_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!}\right) . y_i - \left(\sim_s - \frac{\sim_i^3}{3!}\right) x_i$$

$$x_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!}\right) . x_i - \left(\sim_s - \frac{\sim_i^3}{3!} - \frac{\sim_i^5}{5!}\right) y_i$$

$$y_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!}\right) . y_i - \left(\sim_s - \frac{\sim_i^3}{3!} - \frac{\sim_i^5}{5!}\right) x_i$$

$$x_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!} - \frac{\sim_i^6}{6!}\right) . x_i - \left(\sim_s - \frac{\sim_i^3}{3!} - \frac{\sim_i^5}{5!}\right) y_i$$

$$y_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!} - \frac{\sim_i^6}{6!}\right) . y_i - \left(\sim_s - \frac{\sim_i^3}{3!} - \frac{\sim_i^5}{5!}\right) x_i$$

$$x_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!} - \frac{\sim_i^6}{6!}\right) . x_i - \left(\sim_s - \frac{\sim_i^3}{3!} - \frac{\sim_i^5}{5!} - \frac{\sim_i^7}{7!}\right) y_i$$

$$y_{i+1} = \left(1 - \frac{\sim_i^2}{2!} - \frac{\sim_i^4}{4!} - \frac{\sim_i^6}{6!}\right) . y_i - \left(\sim_s - \frac{\sim_i^3}{3!} - \frac{\sim_i^5}{5!} - \frac{\sim_i^7}{7!}\right) x_i$$

We have used for coordinate calculation for evaluating the best possible combination of approximation, which satisfies the accuracy and RoC requirements, with minimum possible hardware. In Fig. 1, we have plotted the error in magnitude estimated according to (with respect to the corresponding built-in functions of MATLAB). Since the errors resulting from the five combinations are of very small order, we prefer to use for coordinate calculation with minimum complexity.

© 2013 IJAIR. ALL RIGHTS RESERVED

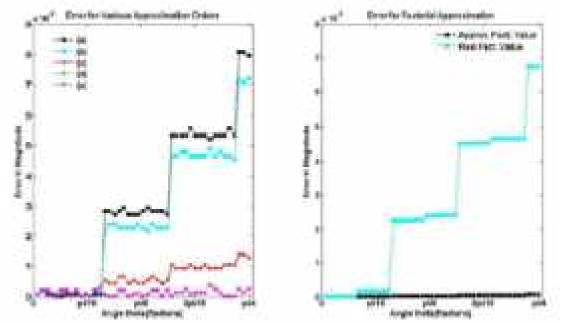

Fig.1. Error in the coordinate values

2) Expressions for Micro-Rotations Using Taylor Series Approximation and Factorial Approximation:

Taylor series expansion with third order of approximation, with desired accuracy and RoC requirement, cannot be used in the CORDIC shift-add iterations. To implement by shift-add operations, we need to approximate the factorial terms by the power of 2 values, replacing 3! by 2^3 in the we find

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} (1-(2!)^{-1}.\sim_i^2) & -(\sim_i - 2^{-3}.\sim_i^3) \\ (\sim_i - 2^{-3}.\sim_i^3) & (1-(2!)^{-1}.\sim_i^2) \end{bmatrix} . \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

…….

In Fig. 1 only, we have plotted the error in magnitude using the approximated factorial values and exact factorial values after a CORDIC rotation for initial vector with coordinates x = 1 and y = 1. The maximum percentage of error in sine and cosine values for both third order of approximation and factorial approximation is 0.0004% and 0.0168%, respectively, within the permissible CORDIC elementary angles range of [0, 7π/88].

B. Determination of the Basic-Shift for a Given Order of Approximation of Taylor Series Expansion

One can find that:

1) the order of approximation of Taylor series expansion of sine and cosine functions determines the basic-shift to be used for CORDIC iterations, and

2) the basic-shift of CORDIC micro operation determines the range of convergence.

The expressions for the basic-shifts, the first elementary angle of rotation (α_i) and RoC for different orders of approximations for different word-length of implementations are as follows:

$$\text{Basic-shift, s} = \left[\frac{b - \log_2 (n+1)!}{(n+1)} \right]$$

Where b is the wordlength, and

$$Roc = n_1 \times \alpha_1$$

n_1 is the number of microtations, and

$$\alpha_1 = 2^{-s}$$

In the order of approximation, the basic-shift decreases, the first elementary angle of rotation increases and RoC is expanded. Very often inclusion of higher order terms does not have any impact on the accuracy for smaller word-lengths.

The basic-shift for third order of approximation, for 16-bit word-length is [2.854]. The RoC (with basic-shift for 16-bit) is large enough to be mapped to the entire coordinate-space.

TABLE I
COMPARISON OF APPROXIMATION ORDERS VERSUS ROC FOR VARIOUS BIT WIDTHS

Order of Approx.	Basic-shift		First Elementary Angle (Radians)		RoC for n_1=4 (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 III
BIT REPRESENTATION OF ELEMENTARY ANGLES AND CORRESPONDING SHIFTS

Shift (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

IV. GENERALIZED MICRO-ROTATION SELECTION

In the proposed generalized micro-rotation sequence, we perform multiple iterations of basic-shift, followed by non-repetitive unidirectional iterations of the micro-rotations corresponding to other shift indices, to minimize the number of iterations and achieve adequate range of convergence.

A. Organization of Micro-Rotation Sequence

In the proposed scheme, we represent the rotation angle “ ¯ ” as

$$- = n_1 . \alpha_s + \sum_{i=1}^{n2} \sim_{si} .n \qquad n = n_1 + n_2$$

Where α_s is the elementary angle corresponding to the basic-shift, α_{si} are elementary angles for other shifts, n_1 and n_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 n_1 is zero, and n_2 = n. On the other hand, if the

© 2013 IJAIR. ALL RIGHTS RESERVED

desired angle of rotation “¯ ” is a multiple of α_s then n_2 is zero and $n_1 = n$.

B. Defining the Elementary Angles

The elementary angles α_s and α_{si} are given by $\alpha_s = 2^{-s}$ and $\alpha_{si} = 2^{-si}$ where, s is the basic-shift and $s_i > s$ is the shift for ith iteration. For basic-shift = 2, we can find $\alpha_{si} = 7\pi/88$ and for basic-shift =3, we can find $\alpha_s = 7\pi/176$. In Table II, we list the decimal and (0, 16) fixed point binary representation of the elementary angles corresponding to different shifts.

C. Generalized Micro-Rotation Sequence Identification

The micro-rotations depending on the bit representation of the desired rotation angle in radix-2 system using most-significant-1 detector. For this we restrict the maximum rotation angle to π/4 radians as the entire coordinate space [0, 2π] can be mapped to the [0, π/4] using octant symmetry of sine and cosine functions.

TABLE IIIII
PSEUDO CODE FOR GENERATING THE MICRO-ROTATION SEQUENCE

Input: angle to be rotated ($¯_i$)
Begin
M=Most Significant-1 Location of $¯_i$
  if (M = 15) then
    α = 0.25 radians
    shift, s_i = 2 and $¯_{i+1} = ¯_{i-\alpha}$
Else
  Shift, s_i = 16 – M
  $¯_{i+1} = ¯_i$ with $¯_i$[M = ‘0’
End

If the most-significant-1 location(M), of the rotation angle “¯ ” is smaller than the basic-shift “s”, elementary angle of the basic-shift would be used for the CORDIC iteration. For a fixed word-length of N-bit, the shift (s_i) for the elementary angle is given by $S_i = N - M$

D. Number of Iterations to Have Desired RoC

We decide on a suitable value of “n” for realizing rotations by angles in the range [0, π/4]. The basic-shift for 16-bit word-length is “2.854”. But the basic-shift should be an integer, so we design the iterations for both “2” and “3”. With basic-shift = 2 ($\alpha_s = 7\pi/88$) no more than three iterations of α_s are required; therefore, the maximum value of $_{n1}$ is 3. The iterations corresponding to n_2 depend on the accuracy requirements. With various values of n_2 the accuracy varies and is different for “x” and “y” coordinates.

In Fig. 2, we plot the error in the “y” coordinates estimated using for the initial vector with coordinates “x=1” and “y=1”, for n=7 and 8. We observe that no better bit error position (BEP) is obtained for n=8 Therefore, to minimize the number of iterations, we restrict the maximum value of “n” to 7. The error in the “x” coordinate is nearly of same magnitude as that of “y” coordinate error.

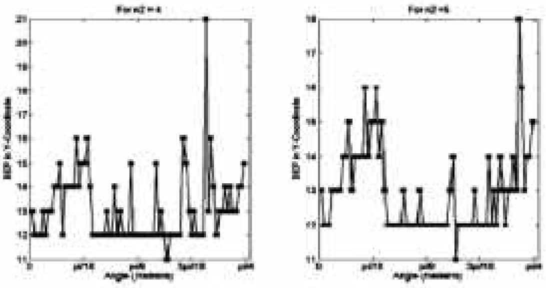

Fig. 2. Error plot for varying n_2 iterations.

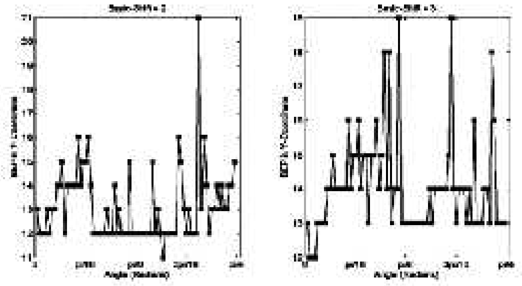

Fig. 3. Error plot for varying basic-shift.

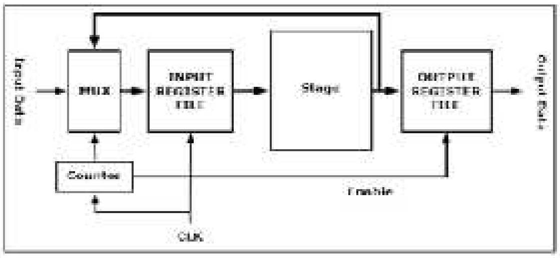

Fig. 4. Recursive architecture of the proposed CORDIC processor.

Similarly, for basic-shift = 3 (α_s=7π/176), no more than six iterations of α_s are equired; therefore, n=10. In Fig. 3, BEP for basic-shift = 2 and 3 are compared.

V. PROPOSED CORDIC ARCHITECTURE

The block diagram for the proposed CORDIC architecture is shown in Fig. 4. It makes use of the same stage for all the iterations for the coordinate calculations, as well as for the generation of shift values. The structure of each stage (shown in Fig. 5) consists of three computing blocks namely: the 1) shift-value estimation; 2) coordinate calculation; and 3) micro-rotation sequence generator.

The combinatorial circuit for the evaluation of desired shift values is shown in Fig. 6; the coordinate calculation is implemented according to (6); the combinatorial circuit for generating the micro-rotation sequence is shown in Fig. 7.

© 2013 IJAIR. ALL RIGHTS RESERVED

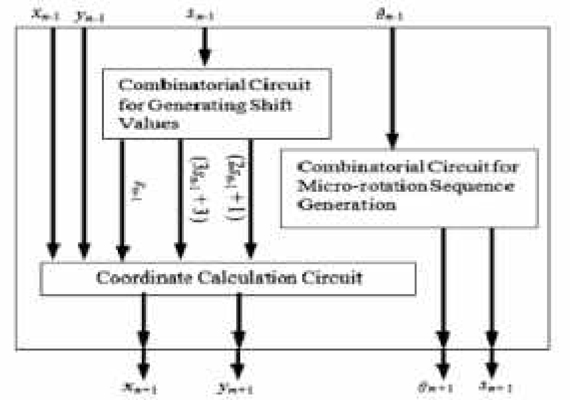

Fig. 5. Block diagram for the stage.

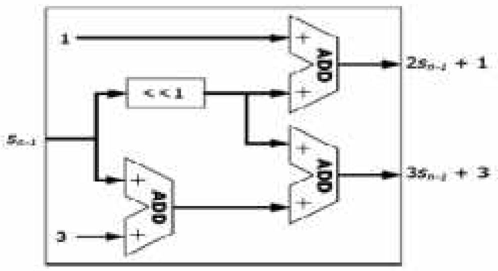

Fig. 6. Combinatorial circuit for generating the shift values.

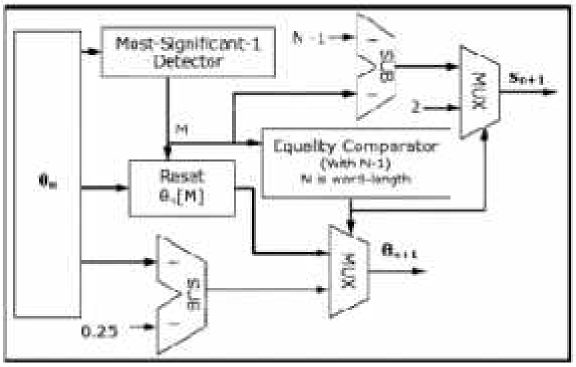

Fig. 7. Micro-rotation sequence generation.

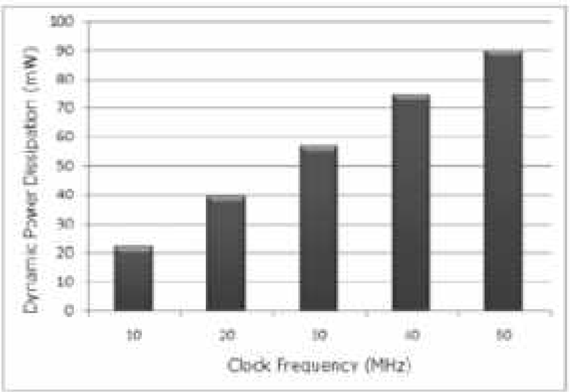

Fig. 8. Dynamic power dissipation for the proposed architecture.

TABLE IVV
SLICE DELAY PRODUCT COMPARISON FOR DIFFERENT APPROCHES

Design	No. of Slices (A)	Max. Freq. MHz (B)	Worst Case Iter. (C)	Slice Delay Product (A*C/B)	Max Error (%)
ALGO-I [9]	186	54.35	10	34.2	0.79
ALGO-II [9]	203	60.80	10	33.4	0.79
Scale-Free [10]	945	52.54	15	269.85	-
Proposed	231	58.37	7	27.7	0.45

The number of iterations required in a CORDIC processor decides the rollover count of the counter. The rollover count is seven for basicshift =2 and ten for basic-shift =3. The expiry of the counter signals the completion of a CORDIC operation; depending on this signal, the multiplexer either loads a new data-set (rotation angle, initial value of “x” and “y”) to start a fresh 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.

VI. FPGA IMPLEMENTATION

The proposed architecture is coded in Verilog and synthesized using Xilinx ISE9.2i to be implemented in Xilinx Spartan 2E (XC2S200EPQ208-6) device. Slice-delay-product of the proposed architecture is compared with the existing CORDIC designs in Table IV; where, all designs are synthesized on Xilinx Spartan 2E XC2S200E device to maintain uniformity. The power dissipation of the proposed architecture for different clock frequencies is estimated by Xilinx XPower tool, and plotted in Fig. 8.

VII. CONCLUSION

The proposed algorithm provides a scale-free solution for realizing vector-rotations using CORDIC. The order of Taylor series approximation is decided appropriately by the proposed algorithm, not only to meet the accuracy requirement but also to attain adequate range of convergence. The generalized micro-rotation selection technique is suggested to reduce the number of iterations for low latency implementation. Moreover, a high speed most-significant-1 detection scheme obviates the complex search algorithms for identifying the micro-rotations. The proposed CORDIC processor has 17% lower slice-delay product with a penalty of about 13% increased slice consumption on Xilinx Spartan 2E device.

REFERENCES

© 2013 IJAIR. ALL RIGHTS RESERVED

[1] J. E. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Comput.*, vol. EC-8, pp. 330–334, Sep. 1959.
[2] K. Maharatna, A. S. Dhar, and S. Banerjee, "A VLSI array architecture for realization of DFT, DHT, DCT and DST," *Signal Process.*, vol. 81, pp. 1813–1822, 2001.
[3] 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.
[4] 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.
[5] C.-S.Wu, A.-Y.Wu, and C.-H. Lin, "A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 9, pp. 589–601, Sep. 2003.
[6] Y. H. Hu and S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," *IEEE Trans. Comput.*, vol. 42, no. 1, pp. 99–102, Jan. 1993.
[7] M. G. B. Sumanasena, "A scale factor correction scheme for the CORDIC algorithm," *IEEE Trans. Comput.*, vol. 57, no. 8, pp. 1148–1152, Aug. 2008.
[8] J. Villalba, T. Lang, and E. L. Zapata, "Parallel compensation of scale factor for the CORDIC algorithm," *J. VLSI Signal Process. Syst.*, vol. 19, no. 3, pp. 227–241, Aug. 1998.
[9] 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.
[10] 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.
[11] 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.

© 2013 IJAIR. ALL RIGHTS RESERVED