HARDWARE EFFICIENT SCALING FREE VECTORING AND ROTATIONAL CORDIC FOR DSP APPLICATIONS

Anita Jain[1] and Kavita Khare[2]

Department of Electronics and Communication Engineering, MANIT, Bhopal, India
[1]anitajainone@gmail.com, [2]kavita_khare1@yahoo.co.in

ABSTRACT

The COordinate Rotation DIgital Computer CORDIC algorithm has proved its versatility in computing various transcendental functions by only using the shift and adds operations. This paper presents a new hardware efficient scaling free CORDIC algorithm to operate in vectoring and in rotation mode. The micro rotation of the vector is always in one direction with no scale factor correction. The Range of Convergence RoC is from 0 to 2π. No pre and post processing circuitry is required. 16 bit Scaling free CORDIC Pipelined architecture based on the proposed algorithm is synthesized on FPGA Xilinx VirtexII P device coded in Verilog. Synthesized results show totally scaling free performance with very small dynamic power consumption of .06 mW and maximum delay of 4.123 ns and 9.925 ns in the rotational and vectoring modes respectively. The minimum BEP of the proposed algorithm implementation is 12. Proposed algorithm is faster and efficient in terms of area and accuracy as compared to conventional CORDIC.

KEYWORDS

DSP, Scale Free CORDIC, Systolic array, rotation mode, vectoring mode, RoC, pipeline architecture.

1. INTRODUCTION

Many Digital Signal Processing applications are based on either trigonometric functions computation or matrix computations. The trigonometric functions are used for calculation of various transforms such as fast fourier transform (FFT), discrete sin/cosine transform (DST/DCT), discrete hartley transform (DHT) and Hough transform (HT)[14] etc. Other signal processing applications, such as the adaptive beamforming, MIMO decoders, the 3G wireless communication, the software defined radio and cognitive radio uses matrix computations such as the QR decomposition [12] etc. To factorize the matrix, systolic arrays processors are used which comprise of CORDIC as processing element (PE). The number of processing element (PE) increases with the order of matrix. Also the CORDIC used as PE needs to be operated in either of rotation mode or vectoring mode depending on their position in systolic array. CORDIC processors are also used to generate signals during modulation and to estimate phase and frequency parameters during demodulation [13].

Efficient hardware implementation of CORDIC on Field Programmable Gate Arrays (FPGA), become necessary for above applications. This paper proposes a hardware efficient scaling free CORDIC algorithm to operate in both modes rotation and vectoring. Being Scaling free CORDIC with no pre and post processing circuitry, the complexity and latency of the system employing large number of CORDIC as PE is reduced significantly.

DOI : 10.5121/vlsic.2013.4302

This paper is structured as follows: Section 2 explains the overview of CORDIC algorithm; Section 3 details the steps of proposed scaling free CORDIC algorithm. FPGA Implementation and results together with error analysis is furnished in Section 4; Section 5 summarizes the conclusion drawn; references are listed in Section 6.

2. CORDIC ALGORITHM OVERVIEW

CORDIC Algorithm is based on rotation of vectors in two dimensional coordinate space using simple shift and add operation. CORDIC Algorithm can operate in two modes namely: rotation and vectoring. In rotation mode, the objective is to convert polar coordinate of a vector into Cartesian where as vice versa in vectoring mode through a series of iterations. The rotation trajectory can be linear, circular or hyperbolic depending upon the requirement.

2.1. Conventional CORDIC Algorithm

The conventional CORDIC algorithm [4] is derived from general equation of vector rotation. If a vector V with components (Xi, Yi) is iteratively rotated through an angle αi, a new vector V' with components (Xi+1, Yi+1) is formed. In matrix form, the value of vector after this micro rotation can be represented as:

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

$$\text{where } K_i = \cos\alpha_i \text{ and } \alpha_i = \tan^{-1}(2^{-i})$$

The sign sequence di Є{1,-1} is so selected that:

$$\theta = \sum_{i=0}^{w-1} d_i \cdot \alpha_i$$

Where, 'w' is the word-length in bits.

Note that the range of convergence of this algorithm is limited to [-99.99°, 99.99°], which can be extended to entire coordinate space using the properties of sine and cosine functions, using an extra iteration for full-range rotation.

The overall scaling-factor of above CORDIC iterations is given by (2)

$$K = \prod_{i=0}^{w-1} K_i = \prod_{i=0}^{w-1} 1/\sqrt{1+2^{-2i}} \qquad (2)$$

2.2. Unified CORDIC Algorithm

Walther [5] has extended the scope of conventional CORDIC algorithm suggested by Volder to include linear and hyperbolic trajectory along with circular trajectory. Due to this extension exponential, logarithmic and various other functions also can be computed using CORDIC. Thus the application and usefulness of it is broadened. A variable (m) for defining the trajectory was introduced to modify the basic CORDIC rotation matrix and elementary angle 'α' as:

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

where, $K_i = \frac{1}{\sqrt{1+m\cdot 2^{-2i}}}$ and $\alpha_i = \frac{1}{\sqrt{m}}\tan^{-1}\left(\sqrt{m}\cdot 2^{-i}\right)$

$$\text{where } m = \begin{cases} 1 & \text{circular} \\ 0 & \text{linear} \\ -1 & \text{hyperbolic} \end{cases}$$

2.3. Scaling-Free CORDIC

Scaling-free CORDIC [8] was the first attempt to completely dispose of the scale-factor. Here, the sine and cosine functions were approximated to:

$$\sin\alpha_i = 2^{-i}$$

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

However, the approximation imposes a restriction on the basic-shift i=⌊((w-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 [9] extends the range of convergence over the entire coordinate space and introduces an adaptive scale-factor.

2.4. Review of Existing CORDIC Architectures

Various architectures have been suggested to improve the efficiency and reduce the complexity of the CORDIC algorithm. The path of development is given chronologically in the paper '50 years of CORDIC' [3]. Vectoring CORDIC processor using conventional CORDIC was first developed by [11] for an arbitrary target value. It has limitation of bulky scale factor compensation circuit and higher hardware cost. Virtually scaling free CORDIC suggested by K. Maharatna eliminates the scaling factor from the CORDIC equations, but the major drawback of this technique is very small range of convergence due to which it can neither be applied for the rotation operation with large angles nor for the vectoring mode of operation. A virtually scaling free vectoring CORDIC using combination of conventional and modified CORDIC was suggested by Stapenhurst [6]. But it has still fixed value of scaling factor and complex pre and post processing unit. Use of radix- 4 booth recoding for reduction of number of iteration is suggested in Enhanced scaling free CORDIC [2] for rotation mode of CORDIC. In revised new CORDIC algorithm [1] an area and time efficient scaling free CORDIC is designed using leading one bit detector. This has efficiently implemented completely scale free rotational CORDIC.

3. PROPOSED ALGORITHM FOR SCALING FREE CORDIC

The proposed algorithm for scaling free CORDIC is based on third order approximation of Taylor series.

The Taylor series expansion of sine and cosine of an angle is:

$$\sin\alpha_i = 2^{-i} - (3!)^{-1}\,2^{-3i} + (5!)^{-1}\,2^{-5i} + \cdots$$
$$\cos\alpha_i = 1 - (2!)^{-1}\,2^{2i} + (4!)^{-1}\,2^{-4i} + \cdots$$

But, this approximation imposes a restriction on the allowed values of iterations i as:

$$i = \lfloor (w - 6.906)/5 \rfloor$$

For 16 bit word length, the initial value of 'i' required for maintaining the accuracy of the calculations comes out to be 2. It divides the coordinate space into eight equal sectors, each of 45 degrees.

CORDIC equation for the above approximation reduces to:

$$x_{i+1} = x_i - (x_i \gg 2i + 1) - (y_i \gg 1) + (y_i \gg 3i + 3)$$
$$y_{i+1} = y_i - (y_i \gg 2i + 1) + (x_i \gg 1) - (x_i \gg 3i + 3)$$

With this division the range of convergence is (0, π/4). To extend the RoC to entire coordinate space quadrant mapping is done as per table 1. for rotation mode and table 2. for vectoring mode.

3.1. Proposed Algorithm for Scaling Free Rotation CORDIC

In rotation mode, angle θ is given as input for which sin θ and cos θ is to be determined as per the CORDIC equations. The Pseudo code for the proposed algorithm is:

Input: z
Output: X, Y

Begin
Step 1: Identify and map the sector and quadrant of the input angle
of the vector.
Step 2: Initialize i = 2; v = word length w;
Step 3: If z [v] = 0; skip the rotation;
Else go to step 4;
Step 4: Rotate the vector and compute its next iterative value.
Step 5: Increment i; Decrement v;
Go to step 3
Step 6: Repeat until (i <= w)
Step 7: Out X, Y
Step 8: Restart with new data in pipeline.
End

Table 1. Quadrant Mapping For Rotation Mode

Range of Target Angle		X	Y
Degrees	Radians		
[0°, 45°]	[0, π/4]	x	y
(45°, 90°]	(π/4,π/2]	y	x
(90°, 135°]	(π/2,3π/4]	-y	x
(135°, 180°]	(3π/4,π]	-x	y
(180°, 225°]	(π,5π/4]	-x	-y
(225°, 270°]	(5π/4,3π/2]	-y	-x
(270°, 315°]	(3π/2,7π/4]	y	-x
(315°,360°]	(7π/4,2π]	x	-y

3.2. Proposed Algorithm for Scaling Free Vectoring CORDIC

In vectoring mode, Cartesian to polar coordinate conversion is carried out as per the CORDIC equation. The Pseudo code for the proposed algorithm is:

Input: X, Y
Output: z, Xout

Begin
Step 1: Identify and map the sector and quadrant of the input vector.
Step 2: Initialize i = 2;

Step 3: Rotate the vector and compute its next iterative value.
Step 4: If MSB of yi = 1; z = 0; i = i+1; go to step 3,
Else z = 1;
i = i+1;
Go to step 3
Step 5: Repeat until (i <= word length)
Step 6: Out Xout = mag and z = phase
Step 7: Restart with new data in pipeline.
End

Table 2. Quadrant Mapping For Vectoring Mode

X	Y	Sector	Final Angle θ
x	y	I	θ
x	y	II	(90°-θ)
-x	y	I	(180°-θ)
-x	y	II	(90°+θ)
-x	-y	I	(180+θ)
-x	-y	II	(270°-θ)
x	-y	I	(360°-θ)
x	-y	II	(270°+θ)

4. FPGA IMPLEMENTATION OF PROPOSED ALGORITHMS

Functional simulation and Hardware implementation of the proposed algorithms for pipeline architecture is carried out in Xilinx ISE9.2i on VirtexII pro device using Verilog. Test bench waveforms are used for simulation purposes. Summary of hardware used is given in table 3.

Table 3. Hardware Summary

Parameter	Rotation mode	Vectoring mode
Flipflops	387	462
LUTs	1344	1482
IOB	86	66
BUFG	1	1
Maximum delay	4.123 ns.	9.925 ns.

Dynamic power dissipation at 20 MHz is found to be .06 mW.

4.1. Error analysis and Mathematical Verification of Simulated Values.

There are two types of error in the results produced by CORDIC algorithm namely quantization error [16] and truncation error. The first arises due to rotation of vector by finite number of iterations and second one is due to truncation of intermediate values because of fixed word length.

The cumulative effect of both these error is analysed by verifying the simulated values mathematically. The error is specified in terms of Bit Error Position BEP. According to Kota [15], using 16 bit word length, the achievable upper bound of the error is approximately 10 b.

Mathematical verification is shown in Table 4. and 5. Results show that the proposed algorithm is better in performance as its minimum BEP is 12 i.e 12 bits are error free out of 16.

This algorithm has used 16 bit fixed point binary representation with 1 represented as 0100_0000_0000_0000. X and Y outputs shown are hexadecimal values.

Table 4. Mathematical Verification And Bit Error Position (Rotation Mode)

Angle		X output		BEP	Y output		BEP
degree	hexadecimal	Theoretical value	Actual Value		Theoretical value	Actual Value	
30°	2183	376D	376C	15	2000	2002	14
60°	4305	2000	2001	15	3763	3764	14
135°	96CC	D2BF	D2BA	13	2D41	2D56	**12**
180°	C910	C000	C000	16	0000	0000	16

Table 5. Mathematical Verification And Bit Error Position (Vectoring Mode)

X	Y	Phase angle $= tan^{-1}(Y/X)$		BEP
Hexadecimal Value	Hexadecimal Value	Theoretical value	Actual Value	
C89D	DFEE	EA92	EA94	14
DFEE	C89D	10C15	10C18	13
2012	C89D	14F1A	14F1C	14
0000	C000	12D98	12D98	16

5. CONCLUSIONS

This paper has presented a novel algorithm for vectoring and rotation mode of CORDIC which is free from any scaling factor correction and complex circuitry for pre and post processing of vectors. It has also proved its efficiency in hardware implementation. Logic blocks used for its implementation are less as compared to conventional CORDIC. It consumes less power and is faster in convergence with RoC covering complete coordinate from 0 to 360 degrees. Its latency is also low as it skips the operation not required as per the algorithm and also by rotating unidirectionally. The accuracy of the proposed algorithm is better as it uses higher order approximation of Taylor series for deriving its CORDIC equation.

Future scope of this algorithm can be extended to cover three dimensional coordinate space as well as other DSP applications.

REFERENCES

[1] Supriya Aggarwal, Pramod K. Meher, and Kavita Khare, (2012) " Area-time efficient scaling-free cordic using generalized micro-rotation selection", IEEE Transactions on VLSI Systems, 20(8):1542–1546.

[2] F. J. Jaime, M. A. Sanchez, J. Hormigo, J. Villalba, and E. L.Zapata, (2010) " Enhanced scaling-free cordic. IEEE Transactions on Circuits and Systems", 57(7):1654-1662.

[3] Pramod Kumar Meher, JavierWalls, Tso-Bing Juang, K. Sridharan, and Koushik Maharatna, (2009)"50 years of cordic : Algorithms and architectures and applications", IEEE Transactions on Circuits and Systems I, 56(9):1893–1907.

[4] J. E. Volder, (1959) "The CORDIC Trigonometric Computing Technique, " IRE Trans. Electronic Computing, Vol. EC-8, pp. 330-334.

[5] J. S. Walther, (1971) "A unified algorithm for elementary functions", Spring Joint Computer Conf., Proc., pp. 379-385.

[6] R. Stapenhurst, K. Maharatna, J. Mathew, J.L. Nunez-Yanez, and D.K. Pradhan, (2007) "A cordic like processor for computation of arctangent and absolute magnitude of a vector", In Proc. IEEE International Symposium on Circuits and Systems ISCAS 2007, pages 3002–3005.

[7] J. Valls, T. Sansaloni, A. Perez-Pascual, V. Torres, and V. Almenar,(2006) " The use of cordic in software defined radios: A tutorial", IEEE Communication Mag., 44(9).

[8] Maharatna, A. Troya, S. Banerjee, and E. Grass,(2004) "Virtually scaling free adaptive CORDIC rotator," IEE Proc.-Comp. Dig. Tech., vol. 151, no. 6, pp. 448–456.

[9] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, (2005) " Modified virtually scaling-free adaptive cordic rotator algorithm and architecture", IEEE Trans. Circuits Syst. Video Technol., 11(11):1463–1474.

[10] K. Maharatna, A. Troya, M. Krstic, E. Grass, and U. Jagdhold,(2004) " A cordic like processor for computation of arctangent and absolute magnitude of a vector", In Proc. ISCAS, volume 2, pages 713–16.

[11] T. Lang and E. Antelo,(1998) "Cordic vectoring with arbitrary target value", IEEE Transactions on Computers, 47(7):736–749.

[12] Dongdong Chen and Mihai SIMA,(2011) "Fixed-Point CORDIC-Based QR Decomposition by Givens Rotations on FPGA", International Conference on Reconfigurable Computing and FPGAs.

[13] José Marcelo L. Duarte, Francisco das Chagas Mota,(2007)" Digital PM Demodulator for Brazilian Data Collecting System", Proceedings of the 20th annual conference on Integrated circuits and systems design , Pages 142-146

[14] K. Maharatna, A. S. Dhar, and S. Banerjee, (2001)"A VLSI array architecture for realization of DFT, DHT, DCT and DST," Signal Process., vol. 81, pp. 1813–1822.

[15] K. Kota, J.R. Cavallaro, (1993)" Numerical Accuracy and Hardware Tradeoffs for CORDIC Arithmetic for Special-Purpose Processors", IEEE Transactions on Computers,vol. 42, no. 7, pp. 769-779.

[16] Y. H. Hu, (1992) "The Quantization Effects of the CORDIC Algorithm," IEEE trans. Signal Processing, Vol. 40, No. 4, pp. 834-844.

AUTHORS

Anita Jain received the B.Tech in 1994 and M. Tech degree in 2009 with specialization in Electronics and Communication Engineering . Currently, she is pursuing her Ph.D. as full time research scholar in Electronics and Communication Engineering in MANIT, Bhopal, India. Her fields of interest are VLSI design. Her research mainly includes front end designing of VLSI circuits. She has ten papers in the proceedings of various Intern ational and National Conferences. Ms. Jain is a life member of ISTE national society.

Kavita Khare received the B.Tech degree in electronics and communication engg in 1989, M.Tech. degree in digital communication systems in 1993, and the Ph.D. degree in the field of VLSI design in 2004.Currently, she is working as Professor in Electronics and Communication Engineering in MANIT, Bhopal, India. Her fields of interest are VLSI design and communication systems. Her research mainly includes design of arithmetic circuits and various commn algorithms related to synchronization, estimation and routing. She has nearly 110 publications in various international conferences and journals. Dr. Khare is a Fellow of IETE (India) and a Life Member of ISTE.