International Journal of Emerging Engineering Research and Technology
Volume 2, Issue 1, April 2014, PP 26-30

Scaling Free Vectoring CORDIC based Rectangular to Polar Converter

Anita Jain[1], Kavita Khare[2]

[1]Research Scholar, Electronics Department, MANIT Bhopal India

[2] Professor, Electronics Department, MANIT Bhopal India

Abstract: *A Rectangular to Polar coordinate converter based on scaling free vectoring CORDIC is proposed in this research paper. Polar coordinates are most appropriate for the systems displaying radial symmetry. Here scaling free CORDIC algorithm operating in vectoring mode computes Polar coordinate of input vector i.e. absolute magnitude and phase angle from its rectangular coordinate. Using this algorithm, the micro rotation of the vector is unidirectional and totally scaling free. The range of convergence is successfully extended to cover entire coordinate space without increasing any hardware complexity. A 16 bit Rectangular to Polar converter based on Scaling free vectoring CORDIC is synthesized on FPGA Xilinx Virtex-5 device using Verilog hardware description language. Synthesized results show throughput every clock cycle with maximum operating frequency of 227.43 MHz and minimum bit error position of 12.*

Keywords: *CORDIC algorithm, Vectoring CORDIC, Scaling free, Range of convergence ROC, FPGA.*

1. INTRODUCTION

Rectangular coordinates and polar coordinates are two different ways of representing any vector on a 2D plane. Rectangular coordinates are in the form (x, y), where 'x' and 'y' are the horizontal and vertical distances from the origin whereas Polar coordinates are in the form (r, Θ), where 'r' is the distance from the origin to the point, and 'Θ' is the angle measured from the positive 'x' axis to the point.

Polar coordinates are used often in navigation, as the destination or direction of travel can be given as an angle and distance from the object being considered Radially asymmetric systems may also be modelled with polar coordinates like omnidirectional and cardioid microphone. Also the Radar measures the things it is tracking by the angle and distance from the antenna. This maritime navigation radar shows the ship's position in the centre of the display and other ships and land as distance and direction from the ship. Thus Polar coordinates are very useful in practical applications.

It is often required to converts the two coordinates and such conversion can easily be carried out electronically using CORDIC algorithm as compared to other method [9].

CORDIC stands for COordinate Rotation Digital Computer. The CORDIC algorithm is a well-known iterative technique to perform various basic arithmetic operations. The algorithm is very attractive for hardware implementation because it uses only elementary shift and add steps to perform vector rotation in a two dimensional plane. This algorithm can be applied to many DSP applications where rotation based arithmetic functions are heavily utilized, such as linear system solver, digital lattice filter, singular value problems, the Fast Fourier Transforms, Discrete Fourier, Discrete Cosine and Discrete Hartley transforms [12]. It has found a wide range of applications also in wireless communication [6], robotics, computer graphics, navigation and astronomy.

The CORDIC algorithm was first proposed by [7] on the basis of Givens Rotation of vectors in two dimensional space. Since then it has been subjected to continuous development in terms of algorithmic change or architecture variations to achieve higher and higher throughput rate, lesser and lesser hardware-complexity and latency. The path of development of CORDIC is beautifully summarized in [5].

The fundamental idea behind the CORDIC is to carry put a sequence of rotations on two-dimensional vectors using a series of specific incremental rotation angles selected such that each is performed by a shift and add operation. It is relatively simple in design and VLSI implementation, as no multipliers are required.

CORDIC Algorithm can operate in two modes namely: rotation and vectoring. In rotation mode, the objective is to rotate a given vector

from its initial position to the final position which is at target angle Θ, through a series of iterations. The rotation decision at each iteration is made to reduce the magnitude of the residual angle to zero. The rotation trajectory can be linear, circular or hyperbolic depending upon the requirement. In vectoring mode, the aim is to find out magnitude and argument of a vector. The vector is rotated from its initial value to its final value so as to reduce the y component to zero. The magnitude of the vector will get stored in the x component and the angle accrued due to such rotations will be stored in the z register representing its phase.

The conventional Vectoring mode CORDIC has a major drawback of generating a bulk scale factor that needs to be compensated using extra circuitry thus increases the hardware cost significantly. This paper proposes a Rectangular to Polar coordinate converter based on scaling free vectoring CORDIC.

Unlike the conventional Vectoring CORDIC, the rotation of vector in the proposed algorithm is always in one direction and has convergence range extended over entire coordinate space. It also has eliminated the need of lookup table and has significantly saved the memory area.

The rest of the paper is structured as follows: Section 2 briefly presents the CORDIC algorithm overview. The proposed converter is discussed in Section 3. Section 4 details the FPGA implementation and mathematical verification and section 5 concludes the paper.

2. CORDIC ALGORITHM OVERVIEW

The basic principle of CORDIC algorithm is to iteratively rotate a vector in a plane by simple shift and add operation to compute either phase or magnitude of a vector or sine and cosine of an angle in circular trajectory. Various other trigonometric, hyperbolic and linear functions can also be computed efficiently based on this basic principle.

2.1 Conventional CORDIC algorithm

The conventional CORDIC algorithm [7] 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:

$$\begin{bmatrix} X_{i+1} \\ Y_{i+1} \end{bmatrix} = K_i \cdot \begin{bmatrix} 1 & -\mathrm{d}_i \cdot \tan\alpha_i \\ \mathrm{d}_i \cdot \tan\alpha_i & 1 \end{bmatrix} \cdot \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (1)$$

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

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

$$\theta = \sum_{i=0}^{w-1} d_i \cdot \alpha_i \quad (2)$$

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.

2.2 Unified CORDIC algorithm

Walther [8] has extended the scope of conventional CORDIC algorithm to include linear and hyperbolic trajectory along with circular trajectory. Due to this extension, the application and usefulness of CORDIC is broadened since computing of various other functions such as exponential and logarithmic becomes possible. A variable (m) for defining the trajectory was introduced to modify the basic CORDIC rotation matrix and elementary angle 'αi' 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, $K_i = \dfrac{1}{\sqrt{1 + m \cdot 2^{-2i}}}$ and

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

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

2.3 Scaling-free Vectoring CORDIC algorithm

The major disadvantages of the earlier CORDIC algorithm were its slow computational speed, requirement of compensation of a bulk scale factor and limited convergence range. To wipe off the effect of variable scaling factor and associated complex circuitry, the Scaling free CORDIC algorithms were developed. Use of Taylor series approximation of sine and cosine functions form the basis of making scaling free CORDIC. In [4] a modified virtually scaling free algorithm is proposed whereas in [2] enhanced version of modified virtually scaling free CORDIC is suggested using booth recoding and conventional CORDIC to make it scaling free. Recent research in [1] uses third order approximation of the series together with high speed most significant-1 detection scheme. The proposed converter is based on the scale free CORDIC as suggested in [10].

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. The RoC becomes 0 to π/4 which is extended through quadrant mapping to entire coordinate space.

CORDIC equation for the above approximation reduces to:

$$X_{i+1} = X_i - (X_i \gg 2i + 1) - (Y_i \gg 1) + (Y\_i \gg 3i + 3)$$

$$Y_{i+1} = Y_i - (Y_i \gg 2i + 1) + (X_i \gg 1) - (X_i \gg 3i + 3)$$

3. PROPOSED RECTANGULAR TO POLAR CONVERTER

For Rectangular to polar conversion, CORDIC is to be operated in vectoring mode. Rectangular coordinates X and Y are given as input and Polar coordinates r and θ are computed as per the conversion equations given above.

The block diagram for the above conversion is as given in Figure 1.

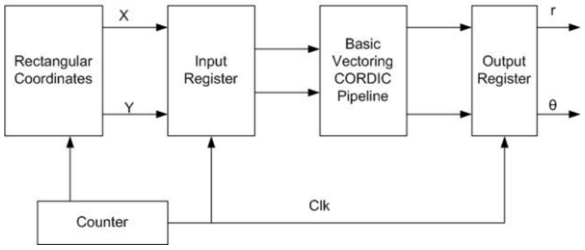

Figure 1. *Block diagram of the Proposed Converter*

The architecture proposed above is fully pipelined architecture. Input vectors are stored in input register and forwarded to next pipeline stage at the positive edge of each clock pulse. There are nine stages of vectoring pipeline which rotates the input vector as per the CORDIC equation and particular value of iterative index. Converted output values are stored in output register which is also synchronized with clock pulse. The complete sequence of conversion is controlled by a counter.

The architecture of one stage of CORDIC pipeline is shown in Figure 2. This pipeline actually implements the CORDIC equation in hardware. The shifters used here are just wired connections and does not add to the complexity of the design. The requirement of the adders and subtractors also reduces to half for iteration index i= 7 onwards.

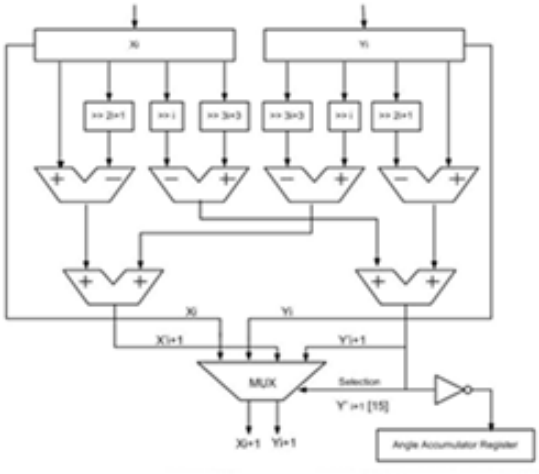

Figure 2. *CORDIC pipeline Architecture*

4. FPGA IMPLEMENTATION AND RESULTS

Functional simulation and Hardware implementation of the proposed Converter is carried out in Xilinx ISE9.2i on Virtex-5 pro device using Verilog. Summary of hardware used is given in Table 1.

Table 1. *Hardware Implementation Summary*

Sl No.	Device Parameter	Number Usage	Number Available	% Utilization
1	Number of Slice Registers	298	19200	1%
2	Number of Slice LUTs	1151	19200	5%
3	Number of Bit Slices used	262	1187	22%
4	Number of Bonded IOB	65	220	29%
5	Number of BUFG	1	32	3%
6	Total Gate Count	15731		

After compiling the simulation libraries and creating the test bench and design code functional simulation is performed on the

design. Simulation waveform of the proposed converter is shown in Figure 3. A uniform clock period of 20 ns is used for the simulation.

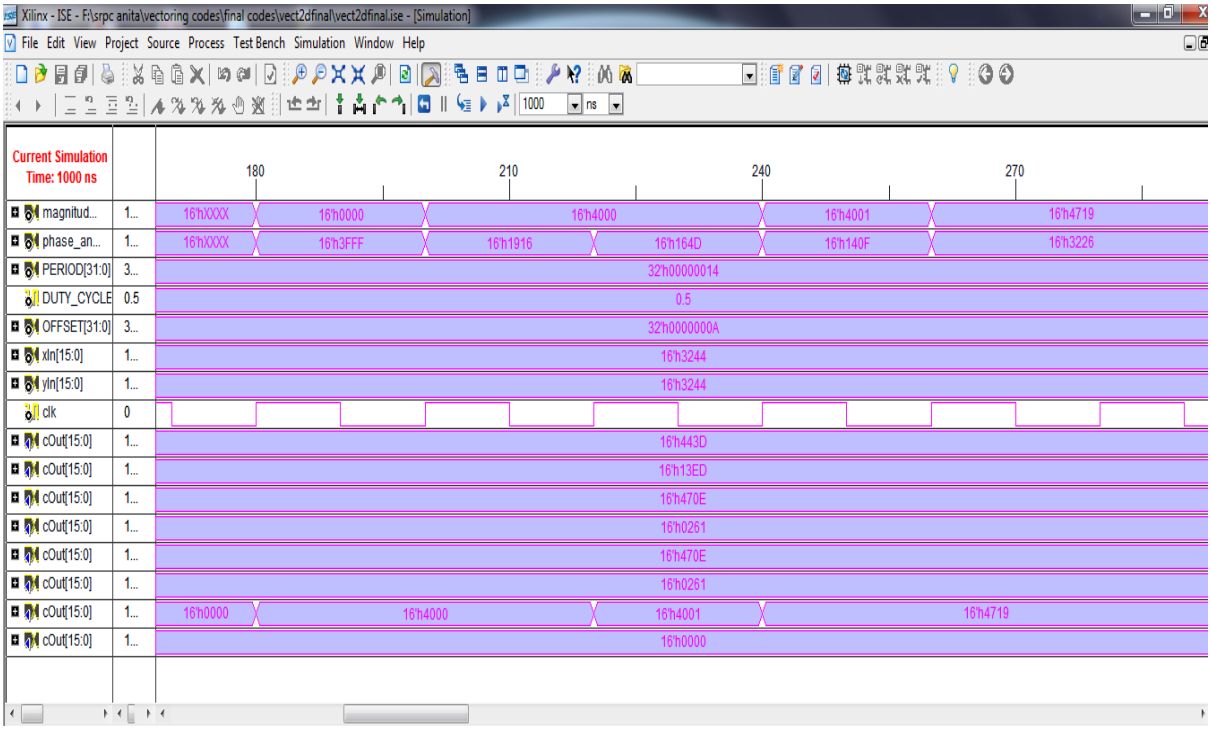

Figure 3. Simulation Waveform of Proposed Converter

The simulated values are verified mathematically and Bit Error Position BEP is calculated. BEP is the indicator of error performance of the converter. It shows that out of 16 bits how many bits are free of error. Table 2 indicates that the minimum BEP for the proposed converter is 12.

Table 2. *Mathematical Verification and Bit Error Position*

Clock Period = 20 ns				
X_In	Y_In	Phase_angle (Output) = $tan^{-1}(Y\_In/X\_In)$		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	12D97	12

The BEP for polar coordinate Θ is plotted in Figure 4 as shown below:

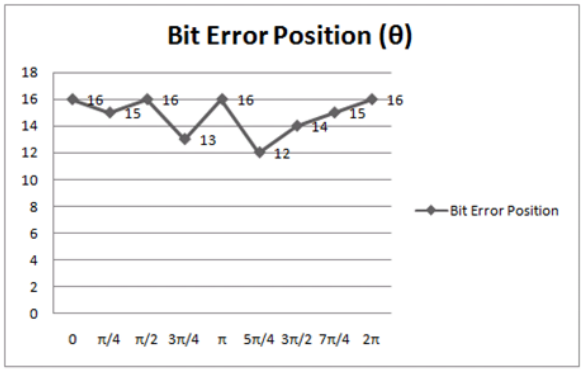

Figure 4. *Bit Error Position Plot for polar coordinate Θ*

5. CONCLUSION

This paper has presented a new Rectangular to Polar coordinate Converter based on scaling-free Vectoring CORDIC. Results have proved its efficacy and accuracy in hardware implementation. The CORDIC method offers a hardware-simple, pipeline-capable, low-transistor count hardware implementation with high degree of accuracy. It can convert any Rectangular coordinate into Polar as its RoC being complete 2D coordinate space.

REFERENCES

[1] Supriya Aggarwal, Pramod K. Meher, and Kavita Khare, "Area-time efficient scaling-free cordic using generalized micro-rotation selection," IEEE Transactions on VLSI

Systems, vol. 20, no. 8, pp. 1542–1546, Aug 2012.

[2] F. J. Jaime, M. A. Sanchez, J. Hormigo, J. Villalba, and E. L.Zapata, "Enhanced scaling-free cordic," IEEE Transactions on Circuits and Systems, vol. 57, no. 7, pp. 1654-1662, 2010.

[3] 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.

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

[5] Pramod Kumar Meher, Javier Walls, Tso-Bing Juang, K. Sridharan, and Koushik Maharatna, "50 years of cordic : Algorithms and architectures and applications," IEEE Transactions on Circuits and Systems I, vol. 56, no. 9, pp. 1893–1907, Sept 2009.

[6] J. Valls, T. Sansaloni, A. Perez-Pascual, V. Torres, and V. Almenar, "The use of cordic in software defined radios: A tutorial," IEEE Communication Mag., vol. 44, no. 9, 2006.

[7] J. E. Volder, "The cordic trigonometric computing technique," IEEE Transactions on Electronic and Computers, vol. 8, pp. 330–334, 1959.

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

[9] Sung-Won Lee, Ki-Seok Kwon and In-Cheol Park, "Pipelined Cartesian-to-Polar Coordinate Conversion Based on SRT Division", IEEE Transactions On Circuits and Systems—II, vol. 54, no. 8, pp. 680-684, August 2007.

[10] Anita Jain and Kavita Khare, " A Novel Scaling Free Vectoring CORDIC and its FPGA Implementation" International Journal of Computer Applications, vol 63, no. 14, pp. 1-5, Feb 2013.

[11] K. Kota, J.R. Cavallaro," Numerical Accuracy and Hardware Tradeoffs for CORDIC Arithmetic for Special -Purpose Processors", IEEE Transactions on Volume 63 Computers, july 1993, vol.42, no. 7, pp. 769-779.

[12] 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.