# FPGA Implementation of Sine and Cosine Value Generators using CORDIC Design for Fixed Angle Rotation

Manjunath.C.R
PG Student, Dept. of ECE
Sri Krishna College of Engineering and Technology, Coimbatore, India

Sathiyapriya
Assistant Professor, Dept of ECE,
Sri Krishna College of Engineering and Technology, Coimbatore, India

## ABSTRACT
CORDIC algorithm has been utilized for various applications such as DSP, Biometrics, Image processing, Robotics apart from general scientific and technical computation. Rotation of vectors through fixed angles has a wide range of application. But in reference to previous CORDIC models, we don't find any optimized scheme for rotation of vectors through fixed angles. This results in increase in On-chip area and power consumption. Taking all these important factors into consideration, here in this paper, we have proposed a simplified pipelined CORDIC architecture model by writing appropriate programs in VHDL for calculating sine and cosine values of an angle with fixed number of iterations. As a result of this, this proposed CORDIC algorithm model helps in achieving reduced On-chip area and power consumption compared to previous reference conventional CORDIC architecture models. System simulation is carried out using ModelSim 6.3f and Xilinx ISE design Suite 9.2i

## Index Term
CORDIC, sine, cosine, barrel shifter, VHDL

## 1. INTRODUCTION
In this paper, we present new, fast and highly hardware efficient technique for computing sine and cosine functions using CORDIC fixed angle rotation. The Coordinate Rotational Digital Computer (CORDIC) is a very well known and widely used hardware efficient iterative algorithm for the computation of elementary arithmetic functions such as hyperbolic, arithmetic trigonometric, exponential and logarithmic operations [1]. The CORDIC algorithm can also be applied for rotation based arithmetic functions such as Eigen value decomposition (EVD) [9], Discrete Cosine Transform (DCT), Wireless Local Area Network (WLAN), Fast Fourier Transform (FFT) [7] and singular value decomposition [8]. CORDIC algorithm generally works by rotating the coordinate system through a constant set of angles until the angle is reduced to zero. Not only a wide variety of applications have been suggested over time, but also a lot of progress has taken place in the area reduction of algorithm design and development of architectures for high performance and low-cost hardware solutions. The CORDIC algorithm doesn't need multipliers, it requires only shifters, adders and lookup tables (LUT). So, it's easier to implement specialized CORDIC machine, small and fast enough to compute for real time calculations, dedicated to one purpose. The trigonometric and exponential functions are computed fairly often in scientific, business and engineering applications. The computation of these transcendental functions in computers is highly performed using software, which means long computational delays and low performance. Computing the transcendental functions using hardware leads to higher speeds with less computational delay rather than the reference conventional CORDIC algorithm designs that are already available. The CORDIC arithmetic unit can be used to solve either set of following equations to calculate the trigonometric functions:

$$Y' = K\,(Y\cos\lambda + X\sin\lambda) \qquad (1)$$

$$X' = K\,(X\cos\lambda + Y\sin\lambda)$$

Where, K is a referred to be a constant. The trigonometric modified fixed angle rotation CORDIC algorithm was originally developed as a digital solution for real time navigation problems. Due to the simplicity of involved operations the CORDIC algorithm is well suited for VLSI implementations at a reduced cost with low power consumption.

## 2. CORDIC ALGORITHM
The general CORDIC algorithm design is an iterative technique and it consists of two modes of operation referred to as the rotation mode and Vectoring mode. In rotation mode, the coordinate components of a vector and an angle of rotation are given to the coordinate components of original vector, after the rotation are allowed to pass through the given angle set of angle for computation. In vectoring mode, the coordinate components of a vector are given and the magnitude and angular movement of the original vectors are computed.

### 2.1 Mathematical Basis of the Algorithm
The first step to obtain the trigonometric functions is by going for vector rotations mode of operation. It can also be used for rectangular to polar conversions and polar to rectangular conversions for vector magnitude and as a building block in the certain transforms such as the DCT and DFT. The main aim of fixed angle rotation CORDIC algorithm is to compute the sine and cosine of a given angle using fixed iterations, which we call θ (theta). Suppose that we have a specific point on the unit circle, which may be illustrated as follows [7]:

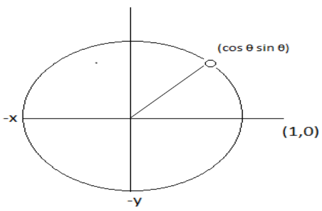


**Figure.1.A point on the unit circle rotated by an angle Θ**

The specific point on the unit circle that has a rotation of θ has coordinates of (Cosθ, Sinθ). This completely implies that if a point on the x-axis of the circle is rotated by a particular angle of θ, then the sine and cosine of the angle of rotation may directly read off from the x-axis and y-axis of unit circle. The rotation of angle can be achieved by rotating the given point on the unit circle in a series of steps may be either in the clockwise direction (decrease in θ) or either in the anti-clockwise direction (increase in θ).

The coordinates of a point in a 2-Dimensional space can be represented as vectors. If the coordinates of the point on the unit circle are referred to be (x, y) then the point can be equally represented as (x, y)' where the inverted comma indicates the transpose function of the matrix. The rotation of a point in 2D space may be effected by multiplying the coordinates through a specific point on the unit circle by a rotation matrix. Thus:

$$\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}\begin{pmatrix} x \\ y \end{pmatrix}=\begin{pmatrix} x' \\ y' \end{pmatrix} \quad (2)$$

Where (x', y') is the coordinates of (x, y) of the unit circle rotated through an particular constant angle $\theta$. This matrix operation can be expressed as follows using the following expression below as follows:

$$X' = X\cos\theta - Y\sin\theta \quad (3)$$
$$Y' = Y\cos\theta + X\sin\theta$$

In CORDIC fixed angle rotation algorithm,, the final angle $\theta_n$ is the angle for which sine and cosine are to be calculated and the initial vale $\theta_1$ are rotated until the set to conventional value reaches 0 by constant rotation. Instead of rotating from $\theta_1$ to $\theta_n$ in one complete full sweep, we can move in steps with careful choice of step values to calculate the sine and cosine values of angle $\theta$ with reduced or fixed iteration. Rearranging the following above equation (3) gives us the set of equation as follows:

$$x' = \cos\theta\ [\ x\text{-}y\tan\theta] \quad (4)$$
$$y' = \cos\theta\ [\ y\text{+}x\tan\theta]$$

Thus restricting the rotation angles such that it is transformed to a relation which is of the form $\tan(\theta)=\pm 2^{-i}$, the multiplication by the tangent term is reduced by an simple shift operation. Therefore, it is observed that the hardware complexity of barrel shifter is nearly half of the CORDIC circuit. Therefore, we aim at suggesting some simple techniques to reduce complexity of barrel shifters. Keeping these key points in view, in this paper we present some best angle optimization schemes for reducing the micro rotations and reducing the complexity of barrel shifters for fixed angle vector rotation using CORDIC algorithm for fixed angle rotation.

## 2.2 Minimization of Hardware Complexity of Barrel Shifter using Hardwired Pre-Shifting

A barrel shifter for maximum of S shifts for a particular word length L it can be implemented through $\log_2(s+1)$ stages of de-multiplexors, where each stage requires only less number of 1:2 lines MUXes. Using the hardwired pre-shifting, it would be therefore easily possible to reduce the total number of shifts to be implemented by the barrel shifters. Thus, it allows us to sustainably reduce the delay and hardware complexity of barrel shifters. Whereas, the conventional barrel shifter has maximum of S shifts for a particular word length L, it can be implemented through $\log_2(s+1)$ stages of 2:1 MUXes. The various modes of operation of CORDIC algorithm are briefly discussed as follows:

## 2.3 Rotation Mode

The first mode of CORDIC algorithm operation is called as rotation mode by J.E. Volder. In rotation mode, the coordinate components of a vector and an angle of rotation are given and the coordinate components of original vector, after the rotation through the given angle are computed. Here the accumulator angle is initialized by a desired set of angle rotation. The decision of rotation is based on the sign of the residual angle of the angle accumulator. If the input angle is already expressed in the binary arc tangent base, the accumulator angle is not necessary. The equation is expressed as follows:

$$X_{i+1} = x_i - y_i.d^i.\ 2^{-I}$$
$$Y_{i+1} = y_i + x_i.d^i.\ 2^{-i} \quad (5)$$
$$Z_{i+1} = z_i - d_i.\tan^{-1}(2^{-i})$$

Where, $d_i = \begin{cases} -1, & if z_i < 0 \\ +1, & otherwise \end{cases}$

Then,

$$x_n = A_n\,[x_0\cos z_0 - y_0\sin z_0]$$
$$y_n = A_n\,[y_0\cos z_0 - x_0\sin z_0] \quad (6)$$
$$z_n = 0$$
$$A_n = \pi\sqrt{1+2^{-i}}$$

## 2.4 Vectoring Mode

In vectoring mode, the coordinate components of a vector are given and the magnitude and angular movement of the original vectors are computed. The vectoring function works by seeking to minimize the y component of the residual vector at each rotation. The actual sign of the residual y component is used for determining which direction is to be rotated next. When the rotation is initialized to zero, the accumulator contains the traversal angle at the end of iteration. The equation for vectoring mode of rotation is expressed as follows:

$$X_{i+1} = x_i - y_i.d^i.\ 2^{-I}$$
$$Y_{i+1} = y_i + x_i.d^i.\ 2^{-i} \quad (7)$$
$$Z_{i+1} = z_i - d_i.\tan^{-1}(2^{-i})$$

Where, $d_i = \begin{cases} -1, & if y_i < 0 \\ +1, & otherwise \end{cases}$

Then,

$$x_n = A_n\sqrt{x_0^2 + y_0^2}$$
$$y_n = 0 \quad (8)$$
$$Z_n = Z_0 + tan^{-1}\begin{pmatrix} x \\ y \end{pmatrix}$$
$$A_n = \pi\sqrt{1+2^{-2i}}$$

The CORDIC rotation and vectoring functions as stated are limited for rotation angles between $-\pi/2$and $\pi/2$. For the composite rotation angles that are larger actually than $\pi/2$, an additional rotation is required. J.E.Volder describes the rotation as $\pm-\pi/2$.This gives us the corrected form of iteration equation and it's as follows:

$$x' = -d.\ y$$
$$y' = d.\ x \quad (9)$$

$$z' = z + d.\frac{\pi}{2}$$

Where, $d_i = \begin{cases} -1, & if\ y < 0 \\ +1, & otherwise \end{cases}$

Since, there is no actual growth for the initial condition. Alternatively an initial rotation of either $\pi$ *or* 0 can be made, avoiding the reassignment of x and y components to the initial rotation of coordinates on the plane of specific point on the unit circle.

$$x' = d.\ x$$

$$y' = d.\ y$$

$$Z_i = \begin{cases} z, & if\ d = 1 \\ z - \pi, & d = -1 \end{cases} \quad (10)$$

Where, $d_i = \begin{cases} -1, & if\ x < 0 \\ +1, & otherwise \end{cases}$

### 2.5 Evaluation of Fixed Angle Rotation Algorithm for Sine and Cosine Value Generators using CORDIC

In the rotation mode of operation the sine and cosine of the given input angle can be computed simultaneously at an equal set of intervals through constant micro rotations. By setting the y component of the input vector to be zero, it reduces the rotation mode result to:

$$x_n = A_n [x_0 \cos z_0 - y_0 \sin z_0] \quad (11)$$

$$y_n = A_n [y_0 \cos z_0 - x_0 \sin z_0]$$

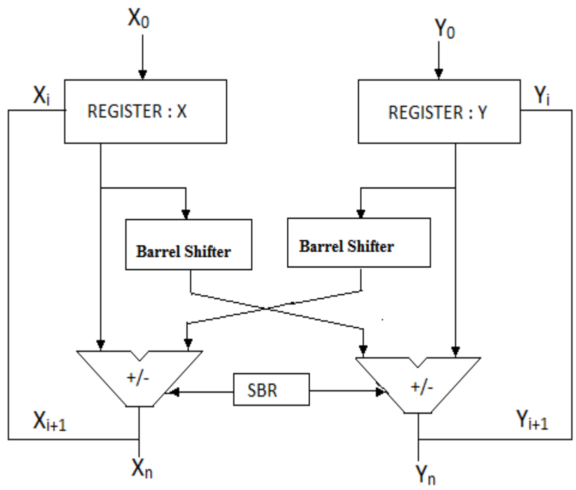


**Fig 2: Cordic Block For Fixed Angle Rotation**

By setting $x_0$ equal to 1/ $A_n$, the rotation of angle through the specific point on the unit circle produces the unscaled sine and cosine values of the angle argument, $z_0$. The magnitude value is very of modulated by the sine and cosine values. Using the some special techniques such as look up table requires a pair of multipliers to obtain the modulation. The CORDIC algorithm performs the multiply as a part of rotation operation. Therefore it eliminates the need of explicit multipliers. The output of the CORDIC rotator is scaled by a rotator again. If the obtained gain is not acceptable, a single multiply is performed by the reciprocal of the gain constant placed before the CORDIC rotator and this will yield the unscaled results[1]. It can be noted that the hardware complexity of fixed angle CORDIC rotator is approximately equally to that of the single multiplier unit with the same word size.

## 3. IMPLEMENTATION

In this paper, the FPGA implementation for calculating the sine and cosine values of a given angle using CORDIC fixed angle rotation is presented. The module was actually implemented by using the Xilinx ISE Design Suite 9.2i and VHDL. The ModelSim simulator is used for verifying the functionalities of the module and this module is described using VHDL and synthesized using Xilinx ISE Design.

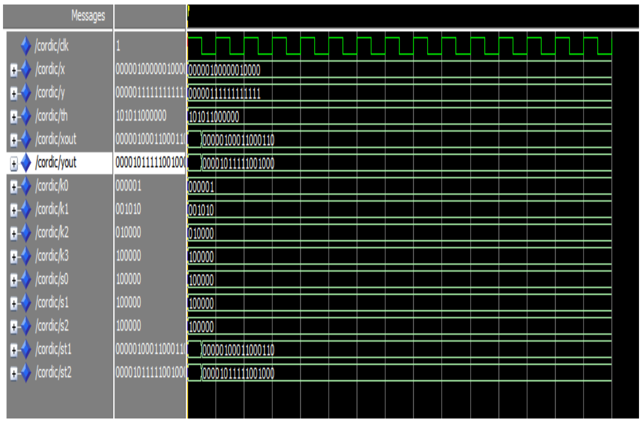

**Figure 3: consist the ModelSim results for binary input angle $A_n$ =45deg and binary outputs $X_n$ (cos ($Z_o$)), $y_n$(sin ($Z_0$)) in the form of waveforms and their corresponding magnitude respectively.**

The following table shows the synthesis report for CORDIC fixed angle rotation for calculating sine and cosine of an angle.

**Table I Synthesis Report**

| | |
|---|---|
| Number of Slices | **130** |
| Maximum Frequency | 228.7Mhz |
| Slice Delay Product | 8.35nSec |

## 4. CONCLUSION

We have successfully implemented the fixed angle rotation CORDIC algorithm for calculating the sine and cosine of angle, on ModelSim simulator using the VHDL language through repeated shift-add operations and this is one of the main features which makes the CORDIC algorithm to be more attractive for a wide variety of applications. Our design can be implemented on Xilinx Spartan 3 XC3S50 using the Xilinx ISE Design Suite 9.2i and VHDL language. The design module is much more efficient and consumes less area than the conventional CORDIC algorithm design models. As part of future work, fixed angle CORDIC algorithm can be implemented on FPAA (Field Programmable Analog Array), Wireless Local Area Network, Hough Transform.

## 5. REFERENCES

[1] Ray Andraka, FPGA '98. Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays, Feb. 22-24, 1998, Monterey, CA. pp191-200

[2] Vikas Sharma,"FPGA Implement999ation of EEAS CORDIC based Sine and Cosine Generator",M.Tech Thesis,Dept.Electron.Comm.Eng.,Thapar Univ., Patiala, 2009.

[3] Satyasen Panda,"Performance Analysis and Design of a Discrete Cosine Transformusing CORDICAlgorithm",M.TechThesis,Dept.Electron.Comm. Eng.,NITRourkela, Rourkela,Orissa,2010.

[4] Kris Raj, Gaurav Doshi, Hiren Shah [online] Available: http://teal.gmu.edu/courses/ECE645/projects_S06/talks/CORDIC.pdf(URL)

[5] S.K.Pattanaik and K.K.Mahapatra,'DHT Based JPEG Image Compression Using a Novel Energy Quantization Method", IEEE International Conference on Industrial Technology,pp.2827-2832,Dec 2006.

[6] Volder, J.,"Binary Computation Algorithms for coordinate rotation and functiongeneration,"Convair Report IAR-1 148 Aeroelectrics Group, June 1956.

[7] Volder, J.,"The CORDIC Trigonometric Computing Technique,"IRE Trans.Electronic Computing, Vol EC-8, pp330-334 Sept 1959.

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

[9] Ahmed,H.M.,Delosme, J.M.,and Morf,M.,"Highly Concurrent Computing Structure for Matrix Arithmetic and Signal Processing," IEEE Comput. Mag,,Vol.15,1982,pp.65-82.

[10] Pramod Kumar Meher., and Sang Yoon Park., “CORDIC Designs for Fixed Angle Rotation,” IEEE Comput. Mag., Vol.22,2013,pp.217-228.

[11] Vladimirova, T. and Tiggler, “FPGA Implementation of Sine and Cosine Generators Using the CORDIC Algorithm”, Proc. of Military and Aerospace Application of Programmable Devices and Technologies Conference (MAPLD 99), Sep. 1999, Laurel, MA, A-2, pp. 28-30.

[12] Naveen Kumar and Amandeep Singh Sappal” Coordinate Rotation Digital Computer Algorithm: Design and Architectures”, International Journal of Advanced Computer Science and Applications, Vol. 2, No. 4, 2011, pp 68-71.

[13] S.F.Hsiao and J.M.Delosme, \Parallel Singular Value Decomposition of Complex Matrices usijng Multi{dimensional CORDIC Algorithms," IEEE Transactions on Signal Processing, vol. 44, no. 3, pp. 685{697, March 1996.

[14] J. Lee and T. Lang, \On-line CORDIC for generalized singular value decomposition," in SPIE Vol. 1058 High Speed Computing II, pp. 235{47, 1989.

[15] S.F.Hsiao and J.M.Delosme, \Householder CORDIC algorithms," IEEE Transactions on Computers, vol. C{44, no. 8, pp. 990{1001, Aug. 1995.