



CORDIC Algorithm based Sine and Cosine Generator

A. M. Borkar¹, A. S. Baradkar², A. D. Bhonde³, P. B. Kude⁴, S. D. Kale⁵, R. J. Giri⁶

Assistant Professor, Dept. of ECE, DBACER Nagpur, India¹

UG Student, Dept. of ECE, DBACER, Nagpur, Maharashtra, India²

UG Student, Dept. of ECE, DBACER, Nagpur, Maharashtra, India³

UG Student, Dept. of ECE, DBACER, Nagpur, Maharashtra, India⁴

UG Student, Dept. of ECE, DBACER, Nagpur, Maharashtra, India⁵

UG Student, Dept. of ECE, DBACER, Nagpur, Maharashtra, India⁶

ABSTRACT: This paper presents an innovative scaling-free CORDIC algorithm in rotation mode for hyperbolic trajectory. We employ most-significant-1 bit revealing method for micro-rotation sequence generation to reduce the number of iterations. The CORDIC algorithmic is an iterative analyzing algorithm capable of evaluate various elementary functions using a combined shift and add approach Used to calculate a wide variety of functions. It consists of structure containing pipelining with internal structure of block. This paper presents mode of operation and control CORDIC algorithm. The product has been revealed in this paper that resolution of CORDIC algorithm is finest for implementing many trigonometrically functions.

KEYWORDS: CORDIC, Rotation Mode, Pipelining, Adder, Subtractor, D-flip-flop.

I. INTRODUCTION

The CORDIC algorithm was first introduced by Jack E. Volder in the year 1959 for the computation of Trigonometric functions, Multiplication, Division, Data type conversion, Square Root and Logarithms. It is a highly efficient, low-complexity and robust technique to compute the elementary functions. The CORDIC algorithm has found its way in various applications such as pocket calculators, numerical co-processors, to high performance radar signal processing, supersonic bomber aircraft with a digital counterpart. The algorithm performs the operation in sequential mode. The CORDIC algorithm requires only shifts, adders, and table lookups, simple integer math. The mode of operation is simple and easy to implement on the hardware.

Nowadays the most of the applications are based on real time results and exchange of information revolves around transmission. The algorithm eliminates the use of unambiguous multipliers and suitable for calculating a variety of functions, such as sine, co-sine, arc-tangent, hyperbolic and logarithmic functions. In this paper the CORDIC results are obtained through the VHDL. The calculation of the sine and cosine is done by rotation method of the algorithm. so it is possible to implement a CORDIC machine slow and fast enough to dedicate for one purpose. In this paper the 16 bit value is given as an input which gives the sine and cosine of the particular input. The output is studied in the form of waveform. The rest of the paper is organized into various sections containing the method, detail of the algorithm and the studied conclusion of the sine and cosine generation.

CORDIC stands for **CO**ordinate **RO**tation **DI**gital Computer. It calculates the value of trigonometric functions like sine, cosine, magnitude and phase (arctangent) to any desired precision. It can also calculate hyperbolic functions (such as sinh, cosh, and tanh). The CORDIC algorithm does not use Calculus based methods such as polynomial or rational function approximation. It is used as approximation function values on all popular graphic calculators, including HP-48G as the hardware restriction of calculators require that the elementary functions should be computed using only

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

additions, subtractions, digit shifts, comparisons and stored constants. Today cordic algorithm is used in Neural Network VLSI design, high performance vector rotation DSP applications, advanced circuit design, optimized low power design. CORDIC algorithm revolves around the idea of "rotating" the phase of a complex number, by multiplying it by a succession of constant values. However, the "multiplies" can all be powers of 2, so in binary arithmetic they can be done using just shifts and adds; no actual "multiplier" is needed thus it simpler and do not require complex hardware structure as in the case of multiplier. Earlier methods used are Table look up method, Polynomial approximation method etc. for evaluation of trigonometric functions. It is hardware efficient algorithm.

II. DESCRIPTION OF CORDIC ALGORITHM

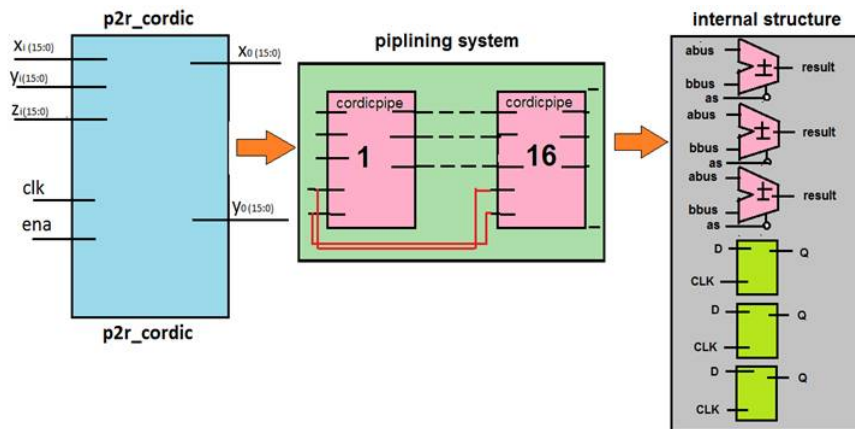


Fig. 1: Block diagram of CORDIC processor

This is the generalized block diagram of the CORDIC processor which shows the three main module of the algorithm. The first block represents the main processor of the algorithm which includes the input and output ports. The second block represents the pipelining of the processor. The third block gives us the information about the internal block diagram of the each block in the pipelining.

III. CORDIC ALGORITHM

All of the trigonometric functions can be evaluated and obtained through the vector rotation method. A vector with the certain angle is rotated and the obtained equation can be written as,

$$x' = x \cos\theta - y \sin\theta \text{ and } y' = y \cos\theta + x \sin\theta$$

This rotates a vector in plane by the angle θ

Now after the rotation the previous equation can be rewrite as,

$$x' = \cos\theta * [x - y \tan\theta] \text{ and } y' = \cos\theta * [y + x \tan\theta]$$

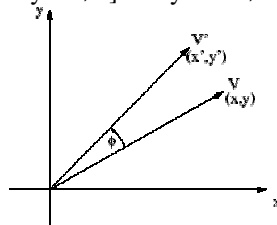


Fig. 2: Rotation of vector by given angle

This is the generalize equation of the vector rotation method. So far nothing is simplified. However if the rotation of angles are restricted so that $\tan(\theta) = i - (\pm 2)$, the multiplication by the tangent term is reduced to simple shift operation. arbitrary angles of revolution are available by performing a series of sequentially smaller elementary operations. If the decision at each iteration i , is which path to rotate rather than whether or not to rotate, then $\cos(\delta_i)$ term becomes a constant, because $\cos(\delta_i) = \cos(-\delta_i)$.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

$$X_{(i+1)} = K_i [x_i - y_i \times d_i \times 2^{-i}] \text{ and } Y_{(i+1)} = K_i [y_i + x_i \times d_i \times 2^{-i}] \text{ where, } K_i = \text{COS}(\tan^{-1} 2^{-i}) = 1/\sqrt{1+2^{-2i}} \text{ and } d_i = 1$$

The K_i can be represented as the part of system processing gain. The product approaches 0.61 as the number of iteration goes to infinity. Therefore the rotation algorithm has a gain of approximately 1.65. The precise gain depends on the no of iterations and obeys the relation.

$$A_n = \prod_{i=1}^n \sqrt{1 + 2^{-2i}}$$

The angle of a complex rotary motion is distinctively definite by the chain of the directions of the elementary rotations. That sequence can be represented by a verdict vector. The set of all possible decision vectors is an angular measurement system based on binary arctangents: $Z_{(i+1)} = z_i - d_i \times \tan^{-1}(2^{-i})$

The elementary angles can be expressed in any convenient angular unit. Those angular standards are supplied by a miniature look up table or are hardwired depending on the application. The angle accumulator sum up 3rd difference equation to the CORDIC algorithm. Obviously, in cases where the angle is useful in the arc tangent base, this extra element is not needed. The CORDIC rotator is normally operated in one of two modes. The first called rotation by Volder rotates the input vector by a specified angle. The second mode called vectoring rotates the input vector to the x axis while recording the angle required making that rotation.

Rotation Mode:

In the rotational mode, the vector magnitude and an angle of rotation are known and the coordinate (X-Y) components are computed after rotation mode. The CORDIC begins by initializing an angle collector with the desired angle. Next his rotation conclusion at each CORDIC iteration is done in way that decreases the magnitude of the lasting angle. The rotation decision is based on the sign of the enduring angle in the angle accumulator after each iteration.

For rotation mode, the CORDIC equations are:

$$X_{(i+1)} = x_i - y_i \times d_i \times 2^{-i}, Y_{(i+1)} = y_i + x_i \times d_i \times 2^{-i} \text{ and } Z_{(i+1)} = z_i - d_i \times \tan^{-1}(2^{-i})$$

where, $d_i = -1$ if $z_i < 0$ and $+1$ otherwise

This provides the following result, $x_n = A_n[x_0 \cos z_0 - y_0 \sin z_0]$, $y_n = A_n[y_0 \cos z_0 + x_0 \sin z_0]$ and $z_n = 0$

$$A_n = \prod_{i=1}^n \sqrt{1 + 2^{-2i}}$$

In rotation mode, the CORDIC algorithm is limited to rotation angles between $-\pi/2$ to $\pi/2$. To support angles outside of the range, quadrant correction is often used.

i	α_i	$\tan(\alpha_i)$	
0	45.0°	1	= 2 ⁻⁰
1	26.6°	0.5	= 2 ⁻¹
2	14.0°	0.25	= 2 ⁻²
3	7.1°	0.125	= 2 ⁻³
4	3.6°	0.0625	= 2 ⁻⁴
5	1.8°	0.03125	= 2 ⁻⁵
6	0.9°	0.015625	= 2 ⁻⁶
7	0.4°	0.0078125	= 2 ⁻⁷
8	0.2°	0.00390625	= 2 ⁻⁸
9	0.1°	0.001953125	= 2 ⁻⁹

Sine and Cosine of CORDIC Operation:

The CORDIC operation simultaneously computes sine and cosine of input angle.

$$x_n = A_n \times x_0 \cos z_0 \text{ and } y_n = A_n \times x_0 \sin z_0$$

By setting x_0 equal to $1/A_n$ rotation produces the unscaled sine and cosine of the angle disagreement z_0 . Very regularly the sine and cosine values modulate a magnitude. Using other techniques requires a no. of multipliers to obtain the modulation. The CORDIC operation performs the multiply as part of the rotation operation and therefore eliminates the need for a pair of obvious multipliers. The output of the CORDIC rotator is scaled by the rotator gain. If the gain is not acceptable, a single multiply by the reciprocal of the gain constant placed before the CORDIC rotator will yield the unscaled results. It is worth noting that the hardware difficulty of the CORDIC rotator is approximately equivalent to that of a single multiplier with the same word size.

Polar to Rectangular Transformation:

The one of the important procedure to compute the sine and cosine is by polar to rectangle: $x = r \cos \theta$ and $y = r \sin \theta$

As pointed out above, the multiplication by the enormity comes for gratis via the CORDIC rotator. The transformation is accomplished by selecting the rotation mode with x_0 =polar phase and $y_0=0$. The vector result represents the polar input transformed to Cartesian space. The transform has a gain equivalent to the rotator gain, which desires to be

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

accounted somewhere in the system. If the gain is unacceptable the polar amount may be multiplied by the reciprocal of the rotator gain before it is presented to the CORDIC rotator.

Analysis of Error:

There are two types of error encountered during the rotation mode CORDIC iterations. Those are: estimate error and round-off error. Estimate error arises due to estimate of angle of rotation and scaling factor, although the round-off error comes due to the finite word-length of the output components. We derive the expression for these couple errors in the next subsections

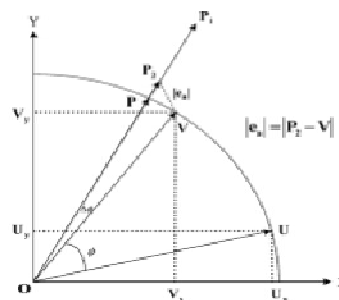


Fig. 3: Rotation mode CORDIC iterations

Approximation of Error:

Proposed CORDIC operation and approximation inaccuracy elaborate the CORDIC iteration which consists of a pseudo-micro-rotations and a scaling. In the figure, is an key in vector to be spin through angle ϕ . It is assumed that scaling and micro-rotations are implement in 2 separate stages. P-1 be the spined vector after m-1 micro-rotations given by

$$P_1 = \prod_{i=0}^{m-1} R(i)U.$$

The rotation matrix R(i) is given by (3). The i th measuring factor is specified by

$$S(i) \triangleq 1 + \delta_i 2^{-s(i)}$$

Such that after m2 iterations of scaling we get

$$P_2 = \prod_{i=0}^{m_2-1} S(i)P_1.$$

After the micro-rotations, there is a incongruity $\Delta\phi$ among the preferred angle and the resultant angle due to the limited number of micro-rotations. Likewise, P-1 cannot attain P on the circle following the measuring since KA is an estimated worth which is not equivalent as the vital K. Alike to the technique used, the estimated error is evaluated as a distance between the chosen output V and the actual CORDIC output P-2 as follows

$$e_a = |V - P_2|.$$

For the recognized and predetermined angle, an anticipation of the approximation error can be estimated once we know the input statistics as

$$E(|e_a|^2) = \left(\left(1 - \frac{K_A}{K} \right)^2 + \Delta\phi^2 \right) E(|U|^2).$$

IV.SIMULATION RESULT

After the programming of CORDIC in VHDL we obtain the results of the modules which are shown below. Following are the RTL (Resister Transfer Logic) of the modules which are schematic of the CORDIC.

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

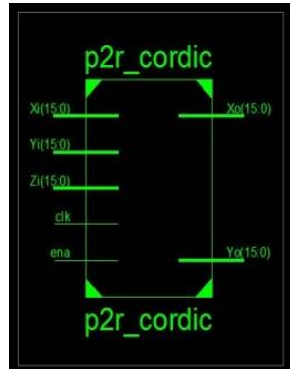


Fig. 4: RTL Schematic

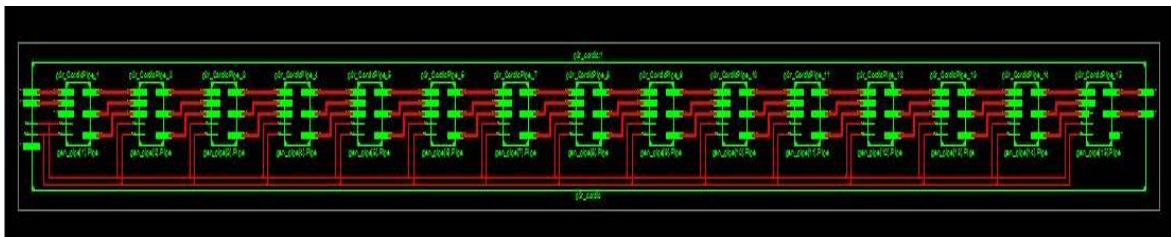


Fig. 5: RTL internal structure of the p2r CORDIC Algorithm

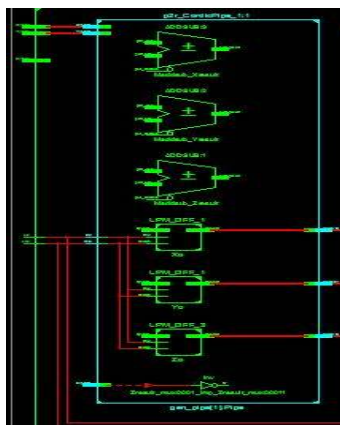


Fig. 6: Internal structure of pipelining

Final output of the CORDIC Algorithm:

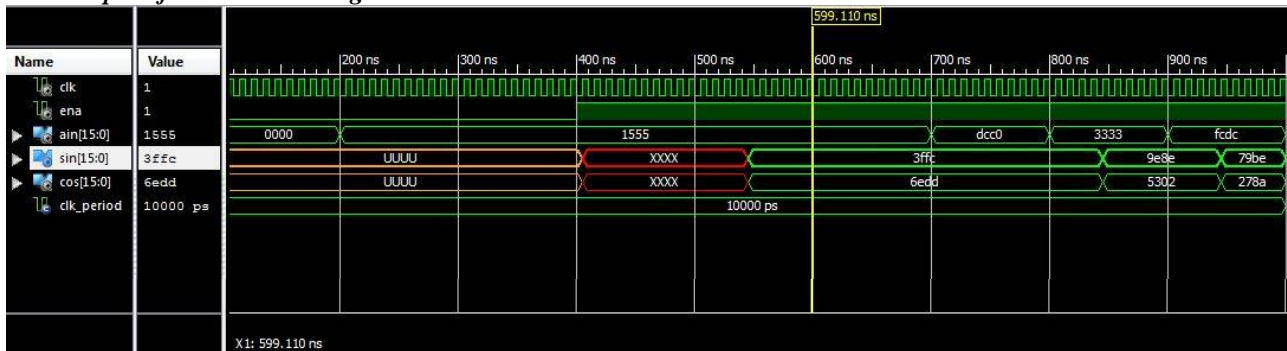


Fig. 7: The waveform of Sine and Cosine of CORDIC Algorithm



ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

V.CONCLUSION

This paper has presented the different architecture of the CORDIC algorithm with the rotation method. The programming of the CORDIC is done in VHDL.CORDIC algorithm mostly suitable for the calculation of the trigonometric function. In this paper the value of the sine and cosine is calculated and the waveform is studied. The CORDIC algorithm is signal and image processing, digital communication, robotics and 3-D graphics.

REFERENCES

- [1] J.E Volder, "the CORDIC trigonometric computing technique", IRE transactions on electronic computer, vol. EC-8, pp. 330-334, 1959
- [2] V.Naresh , B.Venkataramani and R.Raja , "An area efficient multiplexer based CORDIC", National Institute of Technology, Tiruchirappalli, India
- [3] Supriya Aggarwal, Pramod Kumar Meher, Senior Member, IEEE, and Kavita Khar, "Scale-Free Hyperbolic CORDIC Processor and Its Application to Waveform Generation" IEEE transactions on circuits and systems- I : regular papers, vol.60, no.2, February
- [4] Andraka.R.J., "building a high performance bit-serial processor in an FPGA," proceedings of design supercon'96, jan 1996, pp5.1-5.21.
- [5] Pramod K.Meher, Senior Member, IEEE, Javier Valls, Member, IEEE, Tso- Bing Juang , Member, IEEE, K.Sridhan, Senior Member, IEEE, and Koushik Maharatna, Member, IEEE " 50 Years of CORDIC: Algorithms, Architectures, and Applications" , IEEE transactions on circuits and systems, VOL.56.NO.9, September, 2009.