International Journal of Technology and Science, ISSN (Online) 2350-1111, (Print) 2350-1103 Volume 3, Issue 4, 2016 pp. 1-4

Design And Implementation Of Sine And Cosine Generator Using CORDIC Hardware In VHDL

Kaushik R. Naik Gaonkar[1], and Sonia Kuwelkar[2]
[1] Goa College of Engineering/ Electronics & Telecommunication Department, Farmagudi, Ponda, Goa, India
Email: gaonkarkaushik@gmail.com
[2] Goa College of Engineering/ Electronics & Telecommunication Department, Farmagudi, Ponda, Goa, India
Email: sonia@gec.ac.in

Abstract- **Many Digital signal processing application s are based on real time constraints. Therefore, conventional processors are not suitable for modern day DSP systems, leading major issues pertaining to latency and throughput.CORDIC (Coordinate Rotation digital Computer) is one such hardware efficient algorithm developed, used and is a current trend in hardware intensive signal processing. In recent research works being carried out, there are many applications where sine and cosine values are used. Also, there are many ways to generate digital sine and cosine values, but it requires high memory usage. On the other hand, CORDIC Algorithm offers an excellent alternative which has greater flexibility and optimization of memory. It is a Multiplier-less approach which uses only shift and add operations to estimate the basic elementary function thus, is more economical in terms of area and power consumption. The CORDIC algorithm is a widely researched topic in various technological fields such as digital signal processing (DSP), biomedical signal processing, communication systems, robotics, image processing etc. This paper describes the design and simulation of 25-bit sequential hardware for calculating Sine and Cosine of a given Angle using CORDIC algorithm. The design is implemented using VHDL and simulation is carried out using Model Sim 5.7g .System can be implemented using Virtex 5 FPGA. Accuracy up to 4 bits is achieved.**

Index Terms— **CORDIC, VHDL, FPGA, Sine, Cosine.**

I. INTRODUCTION

CORDIC stands for Coordinate Rotation Digital Computer. The first description for iterative approach of this algorithm was firstly provided by Jack E. Volder in 1959[1]. CORDIC algorithm provides an estimate of the basic elementary functions like trigonometric operations, multiplication, division and some other operations like logarithmic functions, square roots and exponential functions by efficiently rotating the vectors in a plane by simple shift add operation to.

It can calculate the value of trigonometric functions like sine, cosine, magnitude and phase to any desired precision. It can also calculate hyperbolic functions (such as sinh, cosh and tanh). It is used as approximation function values on all popular graphic calculators, including HP-48G as the hardware limitation of calculators require that the elementary functions should be computed using only subtractions, additions, comparisons ,digit shifts and stored constants.

CORDIC algorithm revolves around the idea of "rotating" the phase of a complex number, by multiplying it by a succession of constant values. But, the "multipliers" can all be powers of 2, so in binary arithmetic this can be achieved using just shifts and additions. There is no need of actual "multiplier", thus it is easier and does not require a complex hardware structure which is needed in the case of multiplier. However, the CORDIC iteration is not a perfect rotation .The rotated vector has to be scaled making a scale factor correction necessary.

II. CORDIC ALGORITHM

A. Review of CORDIC Algorithm.

The CORDIC algorithm is coordinate rotation in linear, circular and hyperbolic coordinate systems depending on which function has to be calculated. This is performed in the CORDIC algorithm by rotating a vector through a sequence of arbitrary angles whose algebraic sum approximates the required rotation angle [1], [2]. These arbitrary angles have the property that vector rotation through each of them can be computed easily with a single shift and add operation. If a vector V with initial coordinates (x, y) is rotated through an angle Ø then a new vector V' with new coordinates (x', y') is formed. The following method shows how x' and y' can be obtained.

Considering the rotation is in anticlockwise direction for first rotation. So the individual equations for x' and y' can be rewritten as

$$x' = x\cos\text{ø} - y\sin\text{ø} \quad (1)$$

$$y' = x\sin\text{ø} + y\cos\text{ø} \quad (2)$$

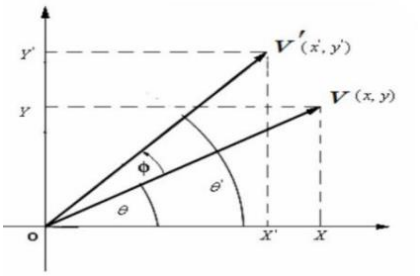

Figure 1 Rotation of vector V by an angle Ø

CORDIC can be operated in two modes of operation: the rotation mode and the vector mode.

International Journal of Technology and Science, ISSN (Online) 2350-1111, (Print) 2350-1103 Volume 3, Issue 4, 2016 pp. 1-4

In vector mode, only the coordinate of original components are given, where as in rotation mode, angle of rotation and coordinate components of original vector are given. Given angle, rotation mode is used to perform general rotation and to compute elementary operations such as hyperbolic functions, multiplication, exponential, and trigonometric functions depending on the coordinate system in which rotations are performed. The vectoring mode can be used to compute the angular argument of the original vector and to compute divisions, logarithmic functions. The number of micro rotations to be performed in both the modes depends on the application. In Cartesian plane rotating a vector by an angle θ can be arranged and equations are as follows

$$x' = \cos ø(x - y\tan ø) \quad (2)$$

$$y' = \cos ø(y + x\tan ø) \quad (3)$$

If the rotation angles are restricted to $\tan(\theta) = \pm 2^{-i}$, so that the multiplication by the tangent term is simply reduced to a shift operation. Arbitrary angles of rotation are available by performing a series of consecutively smaller micro rotation. If the decision at each iterations i, is which direction to relate rather than whether or not to rotate, then the cos (θ) term becomes a constant. The iterative rotation can now be expressed as

$$x_{i+1} = k_i[x_i - d_i.y_i.2^{-i}] \quad (4)$$

$$y_{i+1} = k_i[y_i + d_i.x_i.2^{-i}] \quad (5)$$

Where, $Ki = 1/(1+2^{-2i})^{1/2}$ known as scale constant. di=±1; known as decision function. The scaling constant can be removed from the iterative equations thus yielding a shift-add algorithm for vector rotation. The scaling by K can be done by initiating the rotating vector by the reciprocal of the gain of a certain number of iterations.

The elementary angles can be expressed in any appropriate angular unit either degrees or radians. They can be are stored in small lookup table or can be hardwired, depending on the implementation. The angle accumulator adds a third equation to the CORDIC algorithm

$$z_{i+1} = z_i - d_i . \tan^{-1}(2^{-i}) \quad (6)$$

The CORDIC in rotation mode, a vector (x, y) is rotated by an angle θ. The angle accumulator has to be initialized with the desired rotation angle θ. The rotation decision per iteration is made so as to reduce the value of the remaining angle in the angle accumulator. Hence, the decision per iteration is based on the sign of the remaining angle after each step.

The equations for this are:

$$x_{i+1} = x_i - y_i . d_i . 2^{-i} \quad (7)$$

$$y_{i+1} = y_i + x_i . d_i . 2^{-i} \quad (8)$$

$$z_{i+1} = z_i - d_i . \tan^{-1}(2^{-i}) \quad (9)$$

Where

$d_i = -1$ if $z_i < 0$

$= +1$, Otherwise

Then

$$xn = An[x0\cos z0 - y0\sin z0] \quad (10)$$

$$yn = An[y0\cos z0 + x0\sin z0] \quad (11)$$

Where An is reciprocal of the Gain.x0, y0, z0 are the initial coordinates and angle.

i	$\alpha i = \tan^{-1}2^{-i}$
0	45°
1	26.565°
2	14.036°
3	7.125°
4	3.576°
5	1.7876°
6	0.895°
7	0.448°
8	0.224°
9	0.112°
10	0.056°
11	0.028°
12	0.014°
13	0.00699°

Table 1 Angle Values for corresponding iteration

These Øi are stored in the ROM of the hardware of the CORDIC hardware as a look up table. If the rotation is done by the same set of angles (with + or - signs), then the expansion factor K, is a constant, and can be pre computed.

B. CORDIC Hardware and Architecture

In this paper, the method for calculating the sine and cosine values of given angle using CORDIC algorithm is presented. The module is described in VHDL and Modelsim simulator is used to verify the functionalities of the module. Fig.2 shows the Hardware Elements and Architecture for the CORDIC Algorithm (Single Stage).

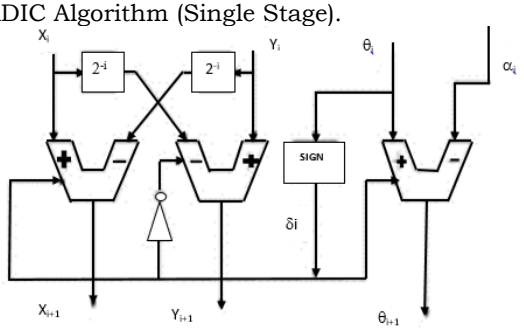

Figure 2 Hardware Elements and Architecture for the CORDIC Algorithm (Single Stage)

International Journal of Technology and Science, ISSN (Online) 2350-1111, (Print) 2350-1103 Volume 3, Issue 4, 2016 pp. 1-4

A straight forward hardware implementation for CORDIC arithmetic is shown above. It requires a lookup table to store values of $\alpha i=\tan^{-1}2^{-i}$, three registers for x, y and z, and two shifter to provide the terms 2^{-i} x and 2^{-i} y to the adder/subtractor units. The di factor (-1 and 1) is accommodated by selecting the (shift) operand or its complement [10].

Each branch consists a shift unit, a register for buffering the output and an adder-subtractor combination. At the beginning initial values are fed into the register by the multiplexer where the magnitude of phase value in the z-branch determines the operation mode for the adder-subtractor, signal in the x and y branch pass the shift units and are then added to or subtracted from the unshifted signal in the opposite path.

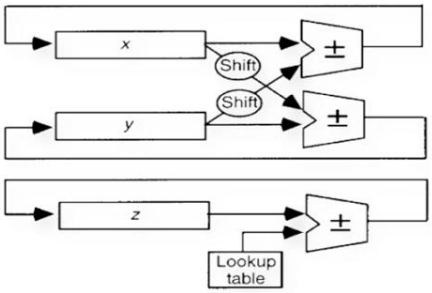

Figure 3 Hardware Elements and Architecture for the CORDIC Algorithm (Iterative)

III. Evaluation of Sine and Cosine

The sine and cosine of the input angle can be calculated in rotational mode. Setting the y component of the input vector to zero reduces the rotation mode result to:

$$xn = An.x0.cosz0 \quad (12)$$

$$yn = An.x0.sinz0 \quad (13)$$

By setting x0 equal to 1/An, the rotation produces the unscaled sine and cosine of the angle given, z0.The algorithm eliminates the need for a pair of explicit multipliers by performing the multiplication as part of the rotation operation. The output of the CORDIC rotator is scaled by the gain of rotator.

IV. Simulation Results

A. Simulation

With the above idea CORDIC based Sine-Cosine calculator is implemented in Hardware Description Language.

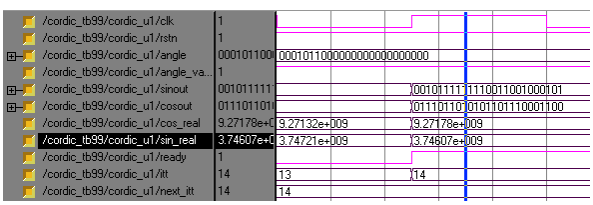

Figure 4 Simulation for angle 22

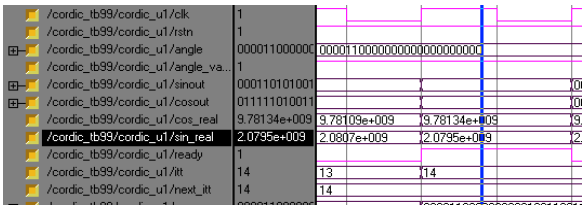

Figure 5 Simulation for angle 12

B. Verification of Results

Input Angle		22°	12°
Cosine	Calculated	0.927178	0.978134
	Theoretical	0.927183	0.978147
	Deviation of Output	0.00001	0.00001
Sine	Calculated	0.374607	0.20795
	Theoretical	0.374606	0.207911
	Deviation of Output	0.00001	0.00001

Table 2 Observation Table

CONCLUSIONS

Hardware for calculating the Sine and Cosine of an angle based on CORDIC algorithm is successfully designed and simulated on Model Sim simulator using the VHDL language.As targeted, the simulation results and theoretical values are consistent. The proposed design can generate accurate results up to 4 decimal places with 25-bit inputs and outputs.

ACKNOWLEDGMENT

I would like to thank the H.O.D of our ETC Department Dr.H.G. Virani for their valuable inputs, encouragement and support. I would like to thank all those who helped me whenever I called on them for any assistance. I am grateful to my parents and all my friends who were supportive throughout the course of this work.

REFERENCES

[1] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Computers, vol. EC-8, pp. 330–334, Sept. 1959.

[2] J. S. Walther, "A unified algorithm for elementary functions," in Proc. 38th Spring Joint Computer Conf., Atlantic City, NJ, 1971, pp. 379–385.

[3] Ashutosh Gupta, Manoj Duhan, Soloman Raju Kota," HDL Implementation of Sine—Cosine Function Using CORDIC Algorithm in 32-Bit Floating Point Format", The ICFI University Press, 2009.

[4] Srinivasa Murthy H N, Roopa M F,"FPGA Implementation of Sine and Cosine Generators using CORDIC Algorithm", International Journal of Innovative Technology and Exploring Engineering", November 2012.

[5] Pramod K. Meher, Javier Valls, Tso-Bing Juang, K. Sridharan, Koushik Maharatna,"50 Years of CORDIC: Algorithms, Architectures and Applications", IEEE transactions on Circuits and Systems – I , VOL.56, NO. 9, September 2009.

International Journal of Technology and Science, ISSN (Online) 2350-1111, (Print) 2350-1103 Volume 3, Issue 4, 2016 pp. 1-4

[6] J. Villalba, J. A. Hidalgo, E. L. Zapata, E. Antelo, and J. D. Bruguera, "CORDIC architectures with parallel compensation of scale factor," Proc. Application Specific Array Processors Conf., pp. 258–269, Jul. 1995.

[7] 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.

[8] Ranjita Naik, Riyazahammad Nadaf, "Sine-Cosine Computation Using CORDIC Algorithm" International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 9, September 2015.