ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

Sine-Cosine Computation Using CORDIC Algorithm

Ranjita Naik[1], Riyazahammad Nadaf[2]

UG Student, Dept of Computer Science, BVB College of Engineering & Technology, Hubli, India [1]

PG Student, Dept of Electronics and Communication, SDM College of Engineering & Technology, Dharwad, India [2]

Abstract: CORDIC stands for **Co**ordinate **R**otation **Di**gital **C**omputer. The CORDIC algorithm was introduced for the computation of Trigonometric functions, Multiplication, Division, Data type conversion, Square Root and Logarithms. It is a highly efficient, low complexity, hardware efficient algorithm giving a robust technique to compute the elementary functions. In the paper the CORDIC algorithm, its usage in calculating quadrature functions, its applications and advantages and disadvantages are explained.

Keywords: CORDIC (**Co**ordinate **R**otation **Di**gital **C**omputer), Sine-Cosine, CORDIC Algorithm.

I. INTRODUCTION

CORDIC stands for **Co**ordinate **R**otation **Di**gital **C**omputer. It calculates 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). 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 additions, subtractions, digit shifts, comparisons 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.

However, the "multipliers" can all be powers of 2, so in binary arithmetic they can be done using just shifts and adds. There is no actual "multiplier" needed, thus it is simpler and does not require a complex hardware structure as in the case of multiplier. Earlier methods used for evaluation of trigonometric functions are table look up method, polynomial approximation method etc. CORDIC is useful in designing computing devices. As it was originally designed for hardware applications, there are features that make CORDIC an excellent choice for small computing devices. Since it is an iterative method it has the advantage over the other methods of being able to get better accuracy by doing more iteration, whereas the Taylor approximation and the Polynomial interpolation methods need to be averaged to get better results.

These properties, in addition to getting a very accurate approximation is perhaps the reason why CORDIC is used in many scientific calculators today. Due to thesimplicity of the involved operations the CORDIC algorithm is very well suited for VLSI implementation. However, the CORDIC iteration is not a perfect rotation which would involve multiplications with sine and cosine. The rotated vector is also scaled making a scale factor correction necessary.

II. CORDIC ALGORITHM

Volder's algorithm is derived from the general equations for a vector rotation. If a vector V with coordinates (x, y) is rotated through an angle Ø then a new vector V' with new coordinates (x', y') is formed where x' and y' can be obtained using x, y and Ø from the following method.

For the ease of calculation here only rotation in anticlockwise direction is observed first. So the individual equations for x' and y' can be rewritten as

$$x' = x\cos Ø - y\sin Ø$$

$$y' = x\sin Ø + y\cos Ø$$

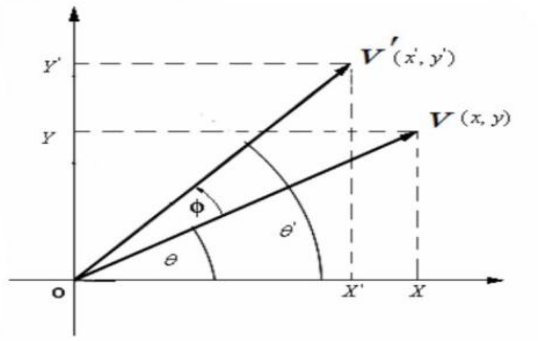

Fig 2.Rotation of vector V by an angle Ø

Volder observed that by factoring out a {cos Ø} from both sides, resulting equation will be in terms of the tangent of the angle Ø. Next if it is assumed that the angle φ is being an aggregate of small angles, and composite angles is chosen such that their tangents are all inverse powers of two, then this equation can be rewritten as an iterative formula

$$x' = \cos Ø(x - y\tan Ø)$$

$$y' = \cos Ø(y + x\tan Ø)$$

$$z' = z + Ø$$

here Ø is the angle of rotation and z is the argument.

DOI 10.17148/IJARCCE.2015.4910

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

The multiplication by the tangent term can be avoided if the rotation angles and tan(Ø) are restricted so that tan(Ø) = 2-i .In digital hardware thisdenotes a simple shift operation.

Furthermore, if those rotations are performed iteratively and in both directions every value of tan(Ø) is representable. With Ø = tan-1(2-i) the cosine term could also be simplified and since cos(Ø) = cos(-Ø) it is constant for a fixed number of iterations. This iterative rotation can now be expressed as:

$$x = r\cos\Theta,\ y = r\sin\Theta$$

$$V' = \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\emptyset & -\sin\emptyset \\ \sin\emptyset & \cos\emptyset \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$x_{i+1} = k_i\left[x_i - d_i y_i 2^{-i}\right]$$

$$y_{i+1} = k_i\left[y_i + d_i x_i 2^{-i}\right]$$

$$z_{i+1} = z_i - d_i\alpha_i$$

Where, i denotes the number of rotations required to reach the required angle of the required vector K_i=cos ($\tan^{-1} 2^{-i}$) and d=±1. $\alpha_i$is nothing but φ_i ,represented for convenience purpose.The product of the K_i represent the so called K factor

$$k = \prod_{i=0}^{n-1} k_i = \cos\emptyset_0 \cos\emptyset_1 \cos\emptyset_2 \ldots\ldots\ldots \cos\emptyset_{n-1}$$

Where $\varphi_i$is the angle of rotation of each iteration.
ki is the gain and its value changes as the number of iteration increases. For 8-bit hardware CORDIC approximation method the value of kias

$$k = \prod_{i=0}^{7} k_i = \cos\emptyset_0 \cos\emptyset_1 \cos\emptyset_2 \cos\emptyset_3 \cos\emptyset_4 \cos\emptyset_5 \cos\emptyset_6 \cos\emptyset_7$$

$$= \cos 45^0 \cos 26.565^0 \ldots\ldots\ldots\ldots \cos 0.4469^0$$

$$= 0.6073$$

From the previous table it can be seen that precision up to 0.44690 is possible for 8-bit CORDIC hardware. These Øi are stored in the ROM of the hard ware of the CORDIC hardware as a look up table.

The CORDIC Algorithm can be used in iterative mode, to simplify each rotation, picking α_i(angle of rotation in ith iteration) such that α_i= (di .2^{-i}). di is such that it has value +1 or -1 depending upon the rotation i. e. di ϵ {+1,-1} .

i	$\tan\alpha_i = 2^{-i}$	$\alpha_i = \tan^{-1}2^{-1}$	α_i *in weighted representation*
0	1	45^0	2000 H
1	0.5	26.565^0	12E4 H
2	0.25	14.036^0	09FB H
3	0.125	7.125^0	0511 H
4	0.0625	3.576^0	028B H
5	0.03125	1.7876^0	0146 H
6	0.015625	0.8938^0	00A3 H
7	0.0078125	0.4469^0	0051 H
8	0.00390625	0.2238105	0029 H
9	0.001953125	0.111905677	0014 H
10	0.000976563	0.05595292	000A H
11	0.000488281	0.027976438	0005 H
12	0.000244141	0.013988248	0003 H

Then

$$x_{i+1} = \left[x_i - d_i y_i 2^{-i}\right]$$

$$y_{i+1} = \left[y_i + d_i x_i 2^{-i}\right]$$

$$z_{i+1} = z_i - d_i \tan^{-1} 2^{-i}$$

The computation of x_{i+1} or y_{i+1} requires an i-bit right shift and an add/subtract. If the function $\tan^{-1}2^{-i}$ is pre computed and stored in table for different values of i, a single add/subtract suffices to compute zi+1. Each CORDIC iteration thus involves two shifts, a table lookup and three additions.
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. For example to rotate by 30 degrees, the following sequence of angles be followed that add up to ϵ 30 degree. 30.0 ϵ 45.0 - 26.6 + 14.0 - 7.1 + 3.6 + 1.8 -0.9 + 0.4 - 0.2 + 0.1 ϵ 30.1 In effect, what actually happens in CORDIC is that z is initialized to 30 degree and then, in each step, the sign of the next rotation angle is selected to try to change the sign of z, that is, di = sign(zi) is chosen, where the sign function is defined to be -1 or 1 depending on whether the argument is negative or nonnegative. This is reminiscent of non-restoring division. The table shows the process of selecting the signs of the rotation angles for a desired rotation of +30 degree. In CORDIC terminology the preceding selection rule for di , which makes z converge to zero, is known as rotation mode. Rewriting the CORDIC iteration, where α_i= $\tan^{-1}2^{-i}$

$$x_{i+1} = \left[x_i - d_i y_i 2^{-i}\right]$$

$$y_{i+1} = \left[y_i + d_i x_i 2^{-i}\right]$$

$$z_{i+1} = z_i - d_i\alpha_i$$

After m iteration in rotation mode, when z (m) is sufficiently close to zero. We have $\Sigma\alpha_i$=z, and the CORDIC equations become:
The constant K in the preceding equation is k = 1.646760258121……

DOI 10.17148/IJARCCE.2015.4910

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

$$x_m = k(x\cos z - y\sin z)$$

$$y_m = k(y\cos z + x\sin z)$$

$$z_m = 0$$

Rule: choose di ϵ { -1,1} such that z --> 0

Thus, to compute cos z and sin z, one can start with x = 1/K = 0.607252935….. And y = 0, then, as zm tends to 0 with CORDIC iterations in rotation mode, xm and ym converge to cos z and sin z, respectively. Once sin z and cos z are known, tan z can be found out through necessary division.

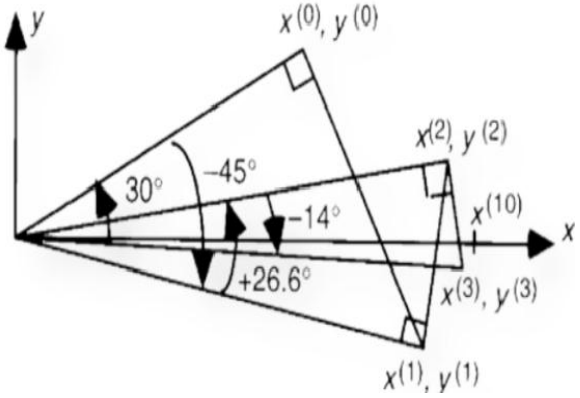

Fig 2.First 3 iteration of rotating the vector

For k bits of precision in the resulting trigonometric functions, k CORDIC iterations are needed. The reason is that for large iit can be approximated that $\tan^{-1} 2^{-i} \approx 2\text{-}i$. Hence, for i> k, the = change in the z will be less than ulp (Unit in the Last Place). In the rotation mode, convergence of z to zero is possible because each angle in table 1 is more than half the previous angle or, equivalently, each angle is less than the sum of the entire angle following it. The domain of convergence is - 99.7 < z < 99.7, where 99.7 is the sum of all the angles in table 3.1. Fortunately, this range includes angle from -900 to +900. For outside the preceding range, trigonometric identities can be converted to the problem, to one that is within the domain of convergence:

III.CORDIC HARDWARE AND ARCHITECTURE

CORDIC is generally faster than other approaches when a hardware multiplier is unavailable (e.g. in a microcontroller) or when the number of gates required toimplement the function is to be minimized (e.g. in an FPGA). On the other hand, when a hardware multiplier is available (e.g. in a DSP microprocessor), table lookup methods and power series are generally faster than CORDIC. In recent years, the CORDIC algorithm has been used extensively for various biomedical applications, especially in FPGA implementations. Various CORDIC architectures like bit parallel iterative CORDIC, a bit parallel unrolled CORDIC, a bit-serial iterative CORDIC and the comparison of various CORDIC architecture has been discussed in the literatures. It can be seen that CORDIC is a feasible way to approximate cosine and sine. There are two ways in CORDIC algorithm for calculation of trigonometric and other related functions. They are rotation modeand Vector mode. Both methods initialize the angle accumulator with the desired angle value. The rotation mode, determines the right sequence as the angle accumulator approaches zero while the vector mode minimizes the y component of the input vector.

A straight forward hardware implementation for CORDIC arithmetic is shownbelow. It requires three registers for x, y and z, a look up table to store the valuesof $\alpha_i=\tan^{-1}2^{-i}$, and two shifter to supply 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.

Of, course a single adder and one shifter can be shared by three computations if a reduction in speed by a factor of 3 is acceptable. In the extreme, CORDIC iterations can be implemented in firmware (micro program)or even software using the ALU and general purpose registers of a standard microprocessor. In this case, the look up table supplying the termα_i can be stored in the control ROM or in main memory.

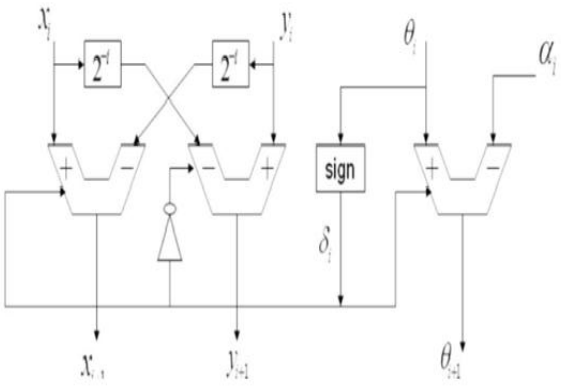

Fig3.Hardware Elements and Architecture for the CORDIC Algorithm (Single Stage)

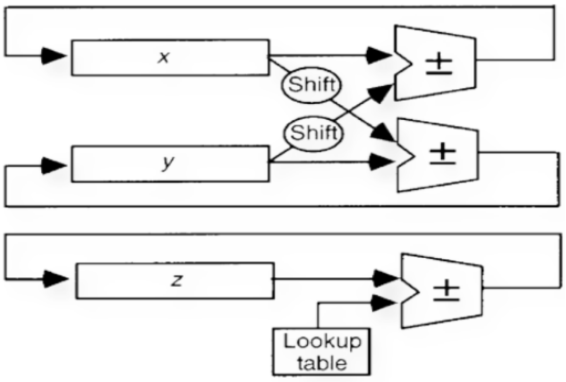

Fig 4.Hardware Elements and Architecture for the CORDIC Algorithm (Iterative)

Each branch consists of an adder-subtractor combination, a shift unit and a register for buffering the output. At the beginning of a calculation initial values are fed into the

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

Output waveform of sine computation using CORDIC algorithm

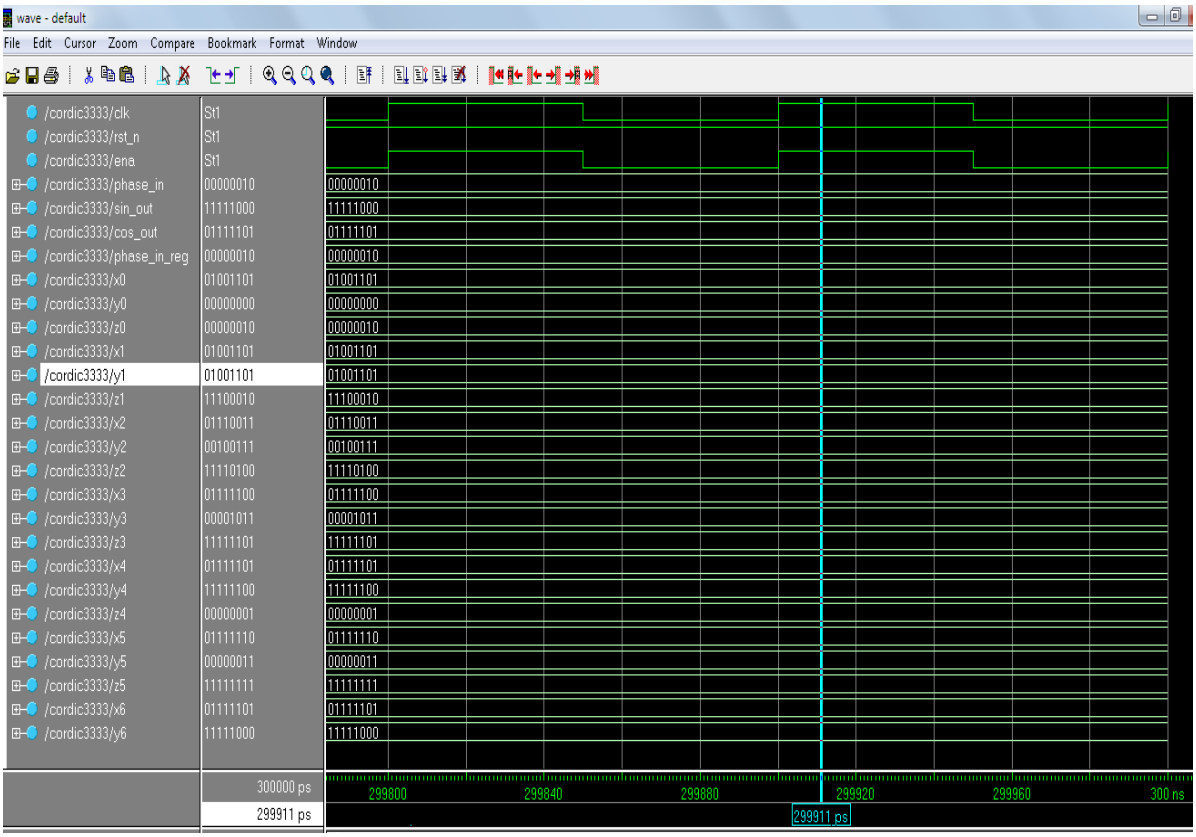

Fig 5 . verilog Implementation Showing Outputs

register by the multiplexer where the MSB of the stored value in the z-branch determines the operation mode for the adder-subtractor, signal in the xand y branch pass the shift units and are then added to or subtracted from the unshifted signal in the opposite path.

The z branch arithmetically combines the registers values with the values taken from a look up table whose address is changed accordingly to the number of iteration. For n iterations the output is mapped back to the registers before initial values are fed in again and the final
sine value can be accessed at the output. A simple finite-state machine is needed to control the multiplexers, the shift distance and the addressing of the constant values.

IV.PIN DIAGRAM OF CORDIC

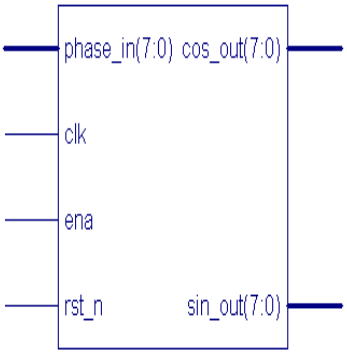

Fig6. Pin Diagram of CORDIC Implementation

Advantages

- Hardware requirement and cost of CORDIC processor is less as onlyshift registers, adders and look-up table (ROM) are required.
- Number of gates required in hardware implementation, such as on an FPGA, is minimum as hardware complexity is greatly reduced compared to other processors such as DSP multipliers.
- It is relatively simple in design.
- No multiplication and only addition, subtraction and bit-shiftingoperation ensures simple VLSI implementation.
- Delay involved during processing is comparable to that during the implementation of a division or square-rooting operation.
- Either if there is an absence of a hardware multiplier (e.g. uC, uP) or there is a necessity to optimize the number of logic gates (e.g. FPGA)CORDIC is the preferred choice.

Disadvantages

- Large number of iterations required for accurate results and thus the speed is low and time delay is high.
- Power consumption is high in some architecture types.
- Whenever a hardware multiplier is available, e.g. in a DSP
- Microprocessor, table look-up methods and good old-fashioned power series methods are generally quicker than this CORDIC algorithm.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 9, September 2015

IV. CONCLUSION

Efficient method to calculate sine and cosine is implemented using Verilog coding. Applications in several diverse areas including signal processing, image processing, communication, robotics and graphics apart from general scientific and technical computations have been explored.

REFERENCES

[1] J. Volder, The CORDIC trigonometric computing technique, IRE Transactions on Electronic Computers, Vol. EC-8, 1959,pp. 330-334.
[2] R. Andraka, A survey of CORDIC algorithms for FPGA-based computers, Proceedings of ACM/SIGDA Sixth International Symposium on Field Programmable Gate Arrays, 1998, pp.191-200[]
[3] LeenaVachhani, K. Sridharan, P.K. Meher, Efficient CORDIC algorithms and architectures for low area and high throughput implementation, IEEE Transactions on Circuits and Systems - Part II: Express Briefs, Vol. 56, No. 1, January 2009, pp. 61-65

BIOGRAPHIES

RANJITA NAIK Pursuing Bachelor of Engineering in Computer science from BVB College of engineering and technology Hubli. Area of interest is cryptography.

RIYAZAHAMMAD NADAF did Bachelor of Engineering in Electronics and communication Engineering from Tontadarya College of Engineering, Gadag and currently pursuing M.Tech in Digital Electronics from Sri Dharmasthala Manjunatheshwara college of engineering and Technology (SDMCET), Dharwad, Karnataka, India. Area of interest includes cryptography, Nanotechnology and Atomic Physics.