ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Rotation of Coordinates With Given Angle And To Calculate Sine/Cosine Using Cordic Algorithm

A. Ramya Bharathi,
M.Tech Student,
GITAM University Hyderabad

Mr. Md Masood Ahmad, M. Tech (AU), (Ph.D),
Assistant Professor,
GITAM University Hyderabad

ABSTRACT

This year, 2015 make CORDIC (COordinate Rotation DIgital Computer) algorithm to complete its 56 years of invention. Digital Signal Processing domain has long been dominated by software systems; however, the state of art signal processing is now again switching back to hardware based solutions. This requires development of algorithms that can be efficiently implemented on different hardware platforms. CORDIC is one such hardware-efficient algorithm that is used in DSP systems for calculating trigonometric, hyperbolic, logarithmic and other transcendental functions. This paper implements how to use CORDIC algorithm to rotate the given coordinates with the required angle.

INTRODUCTION

Acronym for CORDIC is COordinate Rotation Digital COmputer. CORDIC is invented by Jack E. Volder in 1959. CORDIC is initially evolved in 1956 at the aeronautics department of Convair to resolve the navigation problem in B-58 bomber which is just replacement of analog resolver by some new technique like CORDIC algorithm. CORDIC is well suited for calculators where the cost matter much more than its speed. It is well suited and faster than the other approaches for the application where no multiplier is required in the structure as used in the microprocessors. At that time CORDIC is used for the calculation of the trigonometric functions. But as time passed John Stephen Walther in 1971 derive a number of application in calculating square root, hyperbolic functions, logarithmic functions, exponential functions, multiplication and division etc.

One think that we should keep in mind is that CORDIC is not the fastest technique but its simple structural implementation making it different from the other computational techniques. CORDIC uses the same shift-add operation for all application. The first approach mainly achieved by reducing the complexity of barrel shifter and also by reducing the scaling factor. And reduced latency realization can be achieved by schemes like angle recording, mixed grain and higher radix CORDIC, parallel CORDIC, pipelined CORDIC.

OVERVIEW OF CORDIC:

The CORDIC is very simple and iterative convergence algorithm that reduces complex multiplication, greatly simplifying overall hardware complexity. This serves as an attractive option to system designers as they continue to face the challenges of balancing aggressive cost and power targets with the increased performance required in next generation signal processing solutions. The basic principle underlying the CORDIC-based computation, and present its iterative algorithm for different operating modes and planar coordinate system.

CORDIC algorithm has two types of computing modes Vector rotation and vector translation. The CORDIC algorithm was initially designed to perform a vector rotation, where the vector V with components (X,Y) is rotated through the angle θ yielding an ew vector $V^{'}$ with component (X’,Y’) shown in Fig.1.

$$V^{'} = [R]\,[V]$$

Where **R** is the rotation matrix (1)

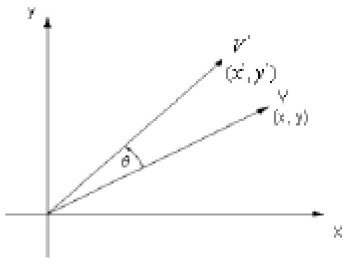


Figure 1: Vector Rotation

R= $\begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$(2)

R= $\begin{bmatrix} \frac{1}{\sqrt{1+tan^2\theta}} & -\frac{tan\theta}{\sqrt{1+tan^2\theta}} \\ \frac{tan\theta}{\sqrt{1+tan^2\theta}} & \frac{1}{\sqrt{1+tan^2\theta}} \end{bmatrix}$(3)

By factoring out the cosine term in (3), the rotation matrix **R** can be rewritten as

R= $\left[(1+tan^2\theta)^{-\frac{1}{2}}\right]\begin{bmatrix} 1 & -tan\theta \\ tan\theta & 1 \end{bmatrix}$(4)and can be interpreted as a product of a scale-factor

K= $\left[(1+tan^2\theta)^{-\frac{1}{2}}\right]$with a pseudo rotation matrix, given by **R$_c$**

R$_c$ = $\begin{bmatrix} 1 & -tan\theta \\ tan\theta & 1 \end{bmatrix}$ (5)

In vector translation, rotates the vector V with component (X, Y) around the circle until the Y component equals zero as illustrated in Figure 2. The outputs from vector translation are the magnitude X' and phase θ', of the input vector V.

After vector translation, output equations are:

X' = K$_i$=$\sqrt{(X^2+Y^2)}$ (6)

Y' = 0 (7)

θ'=$atan\left(\frac{Y}{X}\right)$(8)

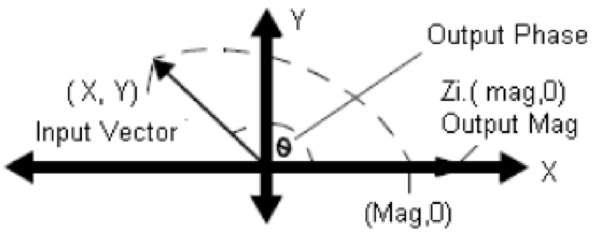


Figure 2: Vector Translation

To achieve simplicity of hardware realization of the rotation, the key ideas used in CORDIC arithmetic are to decompose the rotations into a sequence of elementary rotations through predefined angles that could be implemented with minimum hardware cost and to avoid scaling, that might involve arithmetic operation, such as square-root and division. The second idea is based on the fact the scale-factor contains only the magnitude information but no information about the angle of rotation.

GENERALISED CORDIC ALGORITHM

After few years, Walther found how CORDIC iterations could be modified to compute hyperbolic functions and reformulated the CORDIC algorithm in to a generalized and unified form which is suitable to perform rotations in circular, hyperbolic and linear coordinate systems. The unified formulation includes a new variable m, which is assigned different values for different coordinate systems. The generalized CORDIC is formulated as follows:

$$X_{i+1} = x_i - m\sigma_i.2^{-1}.y_i$$

$$Y_{i+1} = y_i + \sigma_i.2^{-1}.x_i$$

$w_{1+1} = w_i - \sigma_i.\alpha_i$(9)

Here $\sigma_i = \begin{cases} sign(w_i) & for\ rotation\ mode \\ -sign(w_i) & for\ vectoring\ mode \end{cases}$

For m = 1, 0 or −1, and αi=tan−1(2−i), 2−i or tanh−1(2−i), the algorithm given by (11) works in circular, linear or hyperbolic coordinate systems, respectively. Table I summarizes the operations that can be performed in rotation and vectoring modes2 in each of these coordinate systems. The convergence range of linear and hyperbolic CORDIC are obtained, as in the case of circular coordinate, by the sum of all αi given by C = Σ∞i=0αi. The hyperbolic CORDIC requires to execute iterations for i = 4, 13, 40, ... twice to ensure convergence. Consequently, these repetitions must be considered while computing the scale-factor Kh = Π(1 + 2−2i)−1/2, which converges to 0.8281.

m	rotation mode	vectoring mode
0	$x_n = K(x_o \cos\omega_0 - y_o \sin\omega_0)$	$x_n = K\sqrt{x_o^2 + y_0^2}$
	$y_n = K(y_o \cos\omega_0 + y_o \sin\omega_0)$	$y_n = 0$
	$\omega_n = 0$	$\omega_n = \omega_0 + \tan^{-1}(y_0/x_0)$
1	$x_n = x_o$	$x_n = x_o$
	$y_n = y_o + x_o\omega_0$	$y_n = 0$
	$\omega_n = 0$	$\omega_n = \omega_0 + (y_0/x_0)$
-1	$x_n = K_h(x_o \cosh\omega_0 - y_o \sinh\omega_0)$	$x_n = K_h\sqrt{x_o^2 - y_0^2}$
	$y_n = K_h(y_o \cosh\omega_0 + y_o \sinh\omega_0)$	$y_n = 0$
	$\omega_n = 0$	$\omega_n = \omega_0 + \tanh^{-1}(y_0/x_0)$

Table I : Generalized CORDIC Algorithm

SIMPLE CORDIC ALGORITHM

The below figure gives a simple idea about the CORDIC algorithm. Only shifters, registers and adder / subtractor are used for the calculations. Adder/ subtractor are used for the binary addition and subtraction. Shift registers perform the single bit shifting according to the algorithm. And LUTs (look up tables) are used to set the value of the constants according to the demand of angle setting for the algorithm.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

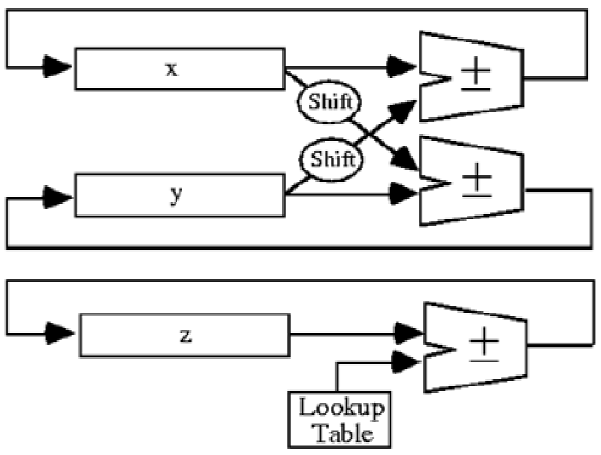


Figure 3: Simple CORDIC Algorithm

CORDIC IMPLEMENTATION

The below figure shows the simulation flow chart of CORDIC algorithm for calculating trigonometric function. The Flow chart shows behavior of the CORDIC algorithm. In the first step we initialize the input parameters to the algorithm. Like the adder x is first set to the scaling parameter having value equal to 0.6072 and y is feeded with zero. Theta is the desired value of the angle for which sin and cosine values to be calculated. If the value of input angle is negative then its sign carries an input equal to 1 otherwise is a 0. The RESET is initially carries an input equal to 1 for putting scaling factor into action, and force to low state for calculating output. The second step declares whether the vector rotation is in the clockwise direction or in the anticlockwise direction and on the basis of that we will find out the value of respective angle rotation for that particular step. Next step defines the value angle rotation value whether it is greater than or lesser than the value of angle of the last step. If present step is greater than we move anticlockwise hence di= +1 is stored otherwise di= +1. Last step carries the updated value of sin and cosine.

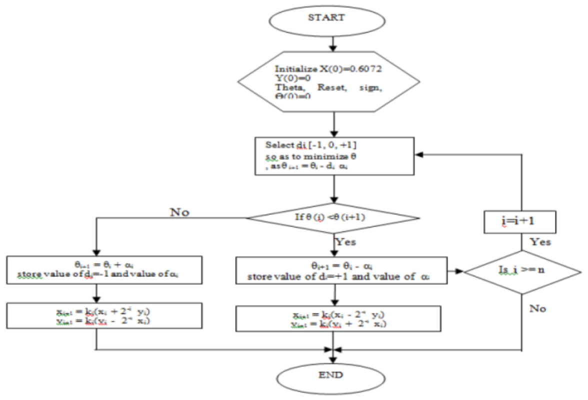


Figure 4: Flowchart of CORDIC Algorithm

Implementation of CORDIC Algorithm to find sine and cosine

Formulae:

Z>0

Z<0

$X[j+1] = X[j] - 2^{-j} Y[j]$

$X[j+1] = X[j] + 2^{-j} Y[j]$

$Y[j+1] = Y[j] + 2^{-j} X[j]$

$Y[j+1] = Y[j] - 2^{-j} X[j]$

$Z[j+1] = Z[j] - \tan^{-1} (2^{-j})$

$Z[j+1] = Z[j] + \tan^{-1} (2^{-j})$

Theoretical Value:

Theta = 30 Z= 30*3.14/180

J	Z[j]	X[j]	Y[j]
0	0.5233	0.60725	0
1	-0.2617	0.60725	0.60725
2	0.2017	0.910	0.303
3	-0.043	0.8342	0.5305
4	0.081	0.9	0.43
5	0.02	0.873	0.48

IMPLEMENTATION OF CORDIC TO ROTATE THE COORDINATES WITH ANGLE θ

ROATATION REDUCTION

In rotation mode, the angle accumulator is initialized with the desired rotation angle. The rotation decision at each iterationis made to diminish the magnitude of the residual angle in the angle accumulator. The decision at each iteration is therefore based on the sign of the residual angle after each step. Naturally, if the input angle is already expressed in the binary arctangent base, the angle accumulator may be eliminated.

Iteration	tan(α(i))	α(i) (in degrees)
0	1	45
1	0.5	26.5
2	0.25	14.03
3	0.125	7.125
4	0.0625	3.576
5	0.03125	1.7899
6	0.015625	0.895
7	0.00781	0.4476
8	0.00390	0.2238
9	0.001953125	0.1
10	0.0009765625	0.055

Table 2: Example of successive angle Rotation Values

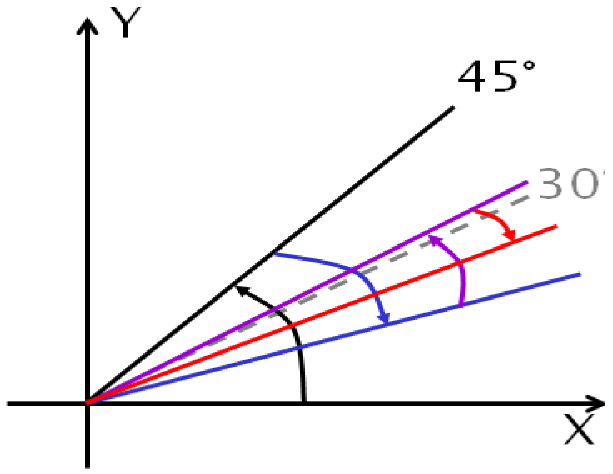


Figure 5: reviewing angles interms of α_i

The equation can be rewritten as

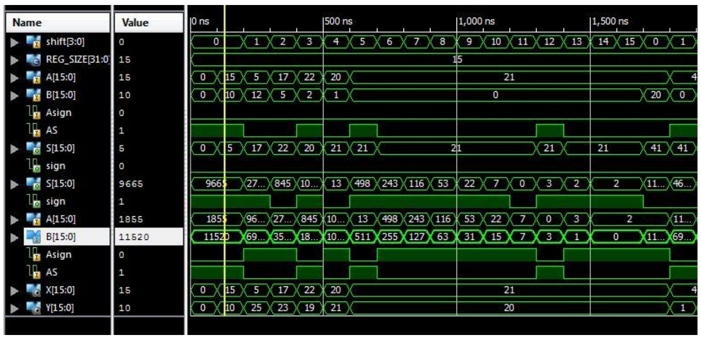
$$x' = \cos(\phi).[x - y.\tan(\phi)] \qquad x_{i+1} = \cos(\alpha_i).[x_i - y_i.d_i.\tan(\alpha_i)]$$
$$y' = \cos(\phi).[y + x\tan(\phi)] \qquad y_{i+1} = \cos(\alpha_i).[y_i + x_i.d_i.\tan(\alpha_i)]$$

Example: ϕ =30.0°

Start with a_0 = 45.0 (> 30.0)
45.0 – 26.6 = 18.4 (< 30.0)
18.4 + 14.0 = 32.4 (> 30.0)
32.4 – 7.1 = 25.3 (< 30.0)
25.3 + 3.6 = 28.9 (< 30.0)
28.9 + 1.8 = 30.7 (> 30.0)

Φ = 30.0
≈ 45.0 – 26.6 + 14.0 – 7.1 + 3.6
+ 1.8 – 0.9 + 0.4 – 0.2 + 0.1
= 30.1

SIMULATION RESULTS



Name	Value
shift[3:0]	0
REG_SIZE[31:0]	15
A[15:0]	15
B[15:0]	10
Asign	0
AS	1
S[15:0]	5
sign	0
S[15:0]	9665
sign	1
A[15:0]	1855
B[15:0]	11520
Asign	0
AS	1
X[15:0]	15
Y[15:0]	10

Figure 6: Waveform for rotation of coordinates with a given angle

APPLICATIONS

CORDIC has a number of applications in the various fields. As we know in CORDIC algorithm no multiplication but only binary addition and bit-shifting operation ensures simple VLSI implementation. Hardware requirement and cost of CORDIC processor is less as only shift registers, adders and look-up table (ROM) are required. So 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. This makes it relatively simple in design. Delay involved during processing is comparable to that during the implementation of a division or square-rooting operation. So due to these advantages CORDIC is still used in various applications and there are a number of field in the present where we can take benefit of these advantages. Following are the main area where CORDIC can be used:

1.CORDIC to calculate DCT

Discrete Cosine Transformation (DCT) [10] is the most widely used transformation algorithm. DCT, first proposed by Ahmed et al, 1974, has got more importance in recent years, especially in the fields of Image Compression and Video Compression. This chapter focuses on efficient hardware implementation of DCT by decreasing the number of computations, enhancing the accuracy of reconstruction of the original data, and decreasing chip area. As a result of which the power consumption also decreases. DCT also improves speed, as compared to other standard Image compression algorithms like JPEG.

2.CORDIC in communication

CORDIC also have some useful applications in communication. CORDIC can be used for efficient generation of amplitude modulation, Frequency modulation, phase modulation, ASK, FSK, PSK, orthogonal frequency division multiplexing. So with these applications CORDIC can be used in software defined Radios [14] which involve modulation and demodulation of digitally generated waves. It has been presented a pipelined CORDIC-based architecture for sine and cosine waves generator targeted to support modulation and demodulation in SDR. Compared with other techniques, CORDIC had shown to have benefits when applied to SDR. The main one was CORDIC make it possible of creating high accuracy waves, even for low frequencies. In the Use of CORDIC in Software Defined Radios is discussed with Direct digital synthesis, which is a method to generate waveforms directly in the digital domain. It show generation of various modulator systems and also cover up-/down converters of in-phase and quadrature signals, full mixers for complex signals, and phase detection for synchronizers which are generally used in Software defined radio.

3. Other application

The algorithm was basically developed for replacing the anlog resolvers by the digital resolvers for solving real-time navigation problems of B-58 bomber. After that John Walther extended the basic CORDIC theory to provide solution to and implement a diverse range of functions. This algorithm finds use in 8087 Math coprocessor, the HP-35 calculator, radar signal processors, and robotics. Most calculators especially the ones built by Texas Instruments and Hewlett-Packard use CORDIC algorithm for calculation of transcendental functions.

CONCLUSION

In this paper CORDIC is implemented to rotate the given coordinates (X,Y) with the given angle θ and also sine and cosine of given angle is found and waveforms are generated using Xilinx ISE tool.

REFERENCES

[1] Naveen Kumar, Amandeep Singh Sappal "Coordinate Rotation Digital Computer Algorithm: Design and Architecture", IJACSA, 2011.

[2] Mr. BijenderMehandra, Mr. JitenderChhabra, Mr. Ajay khokhar, Mr. HunnyPahuja, "Implementation of Sine and Cosine using Volder's CORDIC Algorithm", IJERA, 2012.

[3] Er. LalitBagga, Er. ManojArora, Er. R.S. Chauhan, Er. Parshant Gupta, "Technology Roadmap Of CORDIC algorithm", IJEBEA 2012.

[4]Swati Sharma ,MohitBansal,"Designing of CORDIC processor in Verilog using XILINX ISE simulator"IJRET 2014.

[5] R. RangaTeja, P. SudhakarReddy, " Sine/Cosine Generator using pipelined CORDIC Processor", IACSIT 2011.

[6] Pramod K. Meher, Senior Member, IEEE, Javier Valls, Member, IEEE, Tso-Bing Juang, Member, IEEE,K. Sridharan, Senior Member, IEEE and KoushikMaharatna, Member, IEEE "50 Years of CORDIC algorithms, Architectures and Applications", IEEE 2009.