

# Hardware for Calculation of SIN and COSINE Angle using CORDIC Algorithm

Vipin Tiwari

Department of computer Science and Engineering  
 MANIT, Bhopal India

Nilay Khare

Department of computer Science and Engineering  
 MANIT, Bhopal India.

## ABSTRACT

Trigonometric functions have wide variety of applications in real life. Specially SIN and COSINE waves have been very useful in medical science, signal processing, geology, electronic communication, thermal analysis and many more. Real life application requires fast calculation capabilities as much as possible. Hardware, due to its hardwired design, provides high speed calculations for such application. This paper presents a hardware design that calculates SIN and COSINE value of a given angle using COordinate Rotation DIgital Computer (CORDIC) algorithm.

## Keywords

CORDIC; Hardware; sine, cosine;

## 1. INTRODUCTION

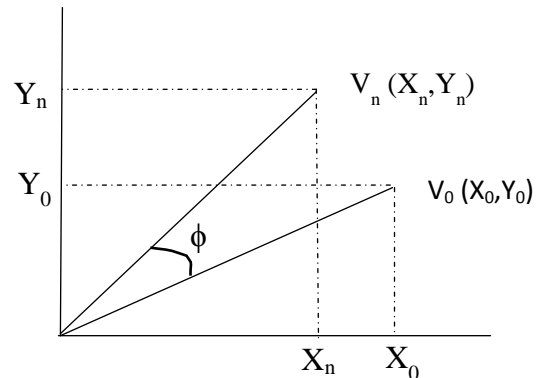
Calculation of sine and cosine of given angle is an essential requirement in many areas of real life. In medical science, medical equipment that measures regular cyclical body functions like heartbeat, breathing etc. use sine and cosine waves. In signal processing, digital audio and high definition videos are based on sums of sine and cosine. In geology, earthquakes are modelled with wave equations which are solved using sums of sine and cosine. Similarly, radio communication used in electronics is based on use of combinations of sine and cosine waves. Hardware provides highest speed due to its hardwired design. This paper presents hardware design for calculating sine and cosine value of given angle using CORDIC algorithm with limited hardware usage.

As proposed by Volder[1] CORDIC is an iterative algorithm which is used to calculate trigonometric, exponential and hyperbolic function. Due to limited hardware architecture CORDIC requires low power as compared to their counterparts because for all calculation, CORDIC architecture uses adder and shifter circuits. Rest of the paper has been organized as follows: section 2 introduces CORDIC algorithm, section 3 summarizes some of the previous work done in the same field. Section 4 explains experimental setup used for the implementation. Section 5 summarizes results of experiments and finally section 6 concludes the work presented in this paper.

## 2. CORDIC ALGORITHM

CORDIC was first proposed by Jack E. Volder[1] in 1959 as an iterative algorithm to convert between polar and Cartesian coordinates using shift, add and subtract operations only. It is also used to compute, trigonometric, hyperbolic, linear and logarithmic functions [2]. Due to its limited hardware requirement it is widely used in math coprocessor, HP-35 calculator, radar signal processor, robotics etc. CORDIC works in two modes: Rotation mode and vectoring mode.

In the rotation mode CORDIC function could compute the Cartesian coordinates of the target vector  $V_n$  by rotating the input vector  $V_0$  by an arbitrary angle  $\phi$ .



**Fig 1: Rotation of vector  $V_0$  to  $V_n$**

As shown in figure 1  $V_0$  and  $V_n$  are initial vector and final vector respectively. Coordinates of final vector  $(X_n, Y_n)$  can be calculated using following equations.

$$X_n = X_0 \cos \phi - Y_0 \sin \phi \quad (1)$$

$$Y_n = Y_0 \cos \phi + X_0 \sin \phi \quad (2)$$

These equations can be rewritten as

$$X_n = \cos \phi [X_0 - Y_0 \tan \phi] \quad (3)$$

$$Y_n = \cos \phi [Y_0 + X_0 \tan \phi] \quad (4)$$

Now if the angle of rotation is restricted to be such that  $\tan \phi = \pm 2^i$ , multiplication by tangent term can be reduced to simple shift operations. If the angle of rotation ( $\phi$ ) is represented as  $Z_0$  then

$$Z_n = 0 \quad (5)$$

Considering eq. (1) and (2), if decision ( $d_i$ ) at each iteration,  $i$  is which direction to rotate, then the term  $\cos \phi$  is constant (because  $\cos \phi_i = \cos(-\phi_i)$ ). The  $\cos \phi_i$  is independent of the direction of rotation. Then iterative rotation can be represented as:

$$X_{i+1} = K_i [X_i - Y_i d_i 2^{-i}] \quad (6)$$

$$Y_{i+1} = K_i [Y_i + X_i d_i 2^{-i}] \quad (7)$$

Where  $K_i = \cos(\tan^{-1} 2^{-i}) = \frac{1}{\sqrt{1+2^{-2i}}}$

$$d_i = \pm 1$$

Removing the scale constant from iterative equations yields a shift-add algorithm for vector rotation. The product of  $K_i$ 's can be treated as part of system processing gain. The rotation algorithm has a gain  $K_n$  of approximately 1.64676. The exact gain depends on number of rotations and can be calculated using relation

$$K_n = \prod_n \sqrt{1 + 2^{-2i}}$$

If total gain for  $n$  iteration be  $K_n$  then equations (1) and (2) can be rewritten as

$$X_n \approx K_n(X_0 \cos Z_0 - Y_0 \sin Z_0) \quad (8)$$

$$Y_n \approx K_n(Y_0 \cos Z_0 + X_0 \sin Z_0) \quad (9)$$

Let if we define a vector where  $Y_0 = 0$  then equations (8) and (9) become

$$X_n \approx K_n X_0 \cos Z_0$$

$$Y_n \approx K_n X_0 \sin Z_0$$

By setting  $X_0 = 1/K_n$  we can calculate the value of sine and cosine of input angle  $Z_0$  as  $Y$  and  $X$  coordinate respectively of the final vector  $V_n$ . Hardware implementation of this is the aim of this work

In vectoring mode, we find angle of vector with respect to axis using micro-rotation. In this mode  $Y$  coordinate of vector is reduced to zero i.e. given vector is rotated using micro-rotations and at each micro-rotation angle is measured and this process continues till given vector coincides with axis. After all micro-rotations angles rotated in each micro-rotation are summed to find desired angle. For vectoring mode, equations for final value of coordinates are as follows:

$$x_n = K \sqrt{x_{in}^2 + y_{in}^2}$$

$$y_n \rightarrow 0$$

$$z_n = \tan^{-1} \left( \frac{x_{in}}{y_{in}} \right)$$

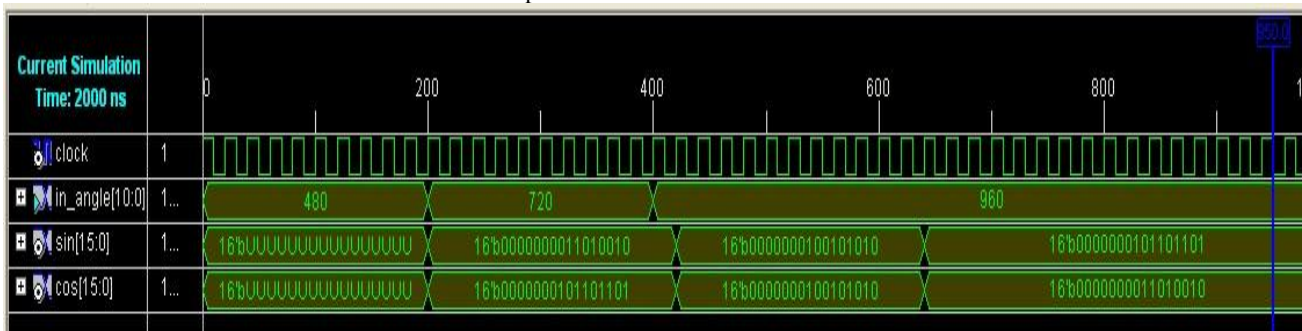
### 3. REVIEWS

Many attempts were made to calculate sine and cosine angle with software and hardware approach. In 1991, Takagi, Asada and Yajima [3] proposed two redundant CORDIC methods with a constant scale factor for sine and cosine computation

called double rotation method. Both methods use redundant binary number representation. In 2001, Kharrat et.al.[4] proposed an optimized CORDIC implementation which offers reduction in silicon area and provides good precision of results but this approach was found suitable for up to 20 bits. Leena Vachhani et.al.[5] introduced an area efficient CORDIC realization for computation of sine and cosine of angle and used this realization in robotics application in 2009. Rodrigues and Swartzlander(2010)[6] developed an adaptive CORDIC to accelerate rotations using parallel angle recoding. This parallel angle recoding reduced iteration by 50 percent but also increases time duration. In 2012 Milica Ristic et.al.[7] presented HDL code generation of CORDIC algorithm with embedded MATLAB code on Altera Cyclone based on FPGA. Chen and Bumko, in 2012 [8] presented a dynamic non uniform segmentation method to evaluate polynomial function and elementary function. This method was implemented using FPGA reduction in hardware requirement, memory usage. In 2013, Shanmuga Kumar et.al. [9] developed high precision and high speed handheld calculator using CORDIC architecture. Using fine and coarse grain method, an effort was made to reduce ROM based Look Up Table (LUT).

### 4. EXPERIMENTAL SETUP

All experiments were performed on Xilinx ISE 9.2i using Very High Speed Integrated Circuit Hardware Description Language (VHDL). CORDIC algorithm, in its rotation mode was implemented with 11 bit input and output of 16 bit. In both the input and output of design, fixed point notation was used to separate integer part and fractional part of the number. Among 11 bits of input least significant 4 bits represent fractional part and remaining 7 bits represent integer part. Similarly in output, among 16 bits least significant 8 bits represent fractional part and next 8 bits represent integer part. As in rotation mode of CORDIC, angle is given as input, which continues to decrease in each micro-rotation, finally reaches to zero. As discussed in section 2 and [2], to calculate sine and cosine of angle given as input, initial coordinate  $x_0$  and  $y_0$  were initialized to 1 and 0 respectively. After each micro-rotations values of  $x$  and  $y$  are updated according to the equations in section 2. After completion of micro-rotations ( $z$  reaches to zero) final values of  $x$  and  $y$  coordinate represent the value of cosine and sine of input angle respectively. For the design, presented in this paper  $x$  and  $y$  coordinate were stored in 16 bits with same fixed point notation of output as discussed above. All the intermediate angles and values of  $x$  and  $y$  were stored in 11 bits and 16 bits registers. Final values of  $x$  and  $y$  coordinates are divided by expansion vector  $K$  as introduced in section 2 to get the final results. Sine and cosine of input angle can be calculated as  $y/K$  and  $x/K$ .



**Fig 2: Values of SINE and COSINE for input angles**

## 5. RESULTS

In this section results of experiments have been discussed. For explanation purposes 3 values of input angle  $z$  have been used as 30o, 45o and 60o. Input angles were presented using fixed point notation as 00111100000, 01011010000 and 01111000000 respectively. In figure 1 results of VHDL implementation has been shown. As can be seen in figure 2, at every 200 ns, this proposed design calculates result for a single value of input angle. Results for input angles 30o, 45o and 60o, as shown in figure above, are as 0000000011010010, 0000000100101010 and 0000000101101101 respectively for sine value. The value of sine of angle can be calculated by dividing the output with 1.64676.

## 6. CONCLUSION AND FUTURE WORK

This paper proposed an efficient method to calculate sine and cosine of input angle using CORDIC algorithm with minimum hardware requirement. CORDIC algorithm was implemented in rotation mode with initial vector coordinates as (1, 0). Proposed hardware generates results in 200 ns with clock of 20 ns cycle duration. The accuracy of results given by proposed design can further be increased using larger size buffer to store inputs and outputs. The design can be enhanced to work with floating point notation instead of fixed point notation to represent fractional numbers.

## 7. REFERENCES

- [1] JACK E. VOLDER, The CORDIC Trigonometric Computing Technique, IRE TRANSACTIONS ON ELECTRONIC COMPUTERS
- [2] Ray Andraka, A survey of CORDIC algorithms for FPGA based computers, Copyright 1998 ACM 0-89791-978-5/98/01
- [3] Naofumi Takagi, Member, IEEE, Tohru Asada, and Shuzo Yajima, Senior Member, IEEE, Redundant CORDIC Methods with a Constant Scale Factor for Sine and Cosine Computation, IEEE TRANSACTIONS ON COMPUTERS, VOL. 40, NO. 9, SEPTEMBER 1991
- [4] KHARRAT M. W., LOULOU M., MASMOUDI N., KAMOUN L., A New Method To Implement CORDIC Algorithm, 7803-70.57-0/01/2001 IEEE
- [5] Leena Vachhani, Student Member, IEEE, K. Sridharan, Senior Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE, Efficient FPGA Realization of CORDIC With Application to Robotic Exploration, IEEE

TRANSACTIONS ON INDUSTRIAL ELECTRONICS,  
VOL. 56, NO. 12, DECEMBER 2009

- [6] Terence K. Rodrigues and Earl E. Swartzlander, Jr., Fellow, IEEE, Adaptive CORDIC: Using Parallel Angle Recoding to Accelerate Rotations, IEEE TRANSACTIONS ON COMPUTERS, VOL. 59, NO. 4, APRIL 2010
- [7] Milica Ristic, Slobodan Lubura, Dejan Jokic, Implementation of CORDIC Algorithm on FPGA Altera Cyclone, 20th Telecommunications forum TELFOR 2012, Serbia, Belgrade.
- [8] Dongdong Chen, Seok-Bum Ko, A dynamic non-uniform segmentation method for first-order polynomial function evaluation, Microprocessors and Microsystems 36 (2012) 324–332
- [9] Shanmuga Kumar M, Mohamed Asan Basiri M, and Noor Mahammad Sk, High Precision and High Speed handheld Scientific Calculator Design using hardware based CORDIC Algorithm, International Conference ON DESIGN AND MANUFACTURING, IConDM 2013.

## 8. AUTHOR'S PROFILE

**Vipin Tiwari** is a research scholar at Maulana Azad National Institute of Technology. He did B.E. in Computer Science and Engineering in 2003. He completed M.Tech. in Computer Technology and Application from School of Information Technology, Rajiv Gandhi Technological University, Bhopal. His research interests include neural networks, Algorithm Design and Analysis, Computer Hardware.

**Nilay Khare** is an associate professor at Maulana Azad National Institute of Technology. He did M.Tech. in Computer Science and Engineering from IIT Delhi. He received Ph.D degree in Computer Science and Engineering. He has been professor in Department of Technical Education, Government of Madhya Pradesh, India. He has also worked as head of State Project Facilitation Unit, Government of Madhya Pradesh, India and as head of department of Computer Science and Engineering, Rajiv Gandhi Technological University, Bhopal, Madhya Pradesh, India. He has published more than 50 papers in national and international journals. He is member of Indian Society of Technical Education (ISTE) and Computer Society of India (CSI). His research areas include Wireless Networks and Theoretical Computer Science.