

# VHDL Implementation of CORDIC Algorithm for Wireless LAN

Mr. Gadekar S. R.

ME Electronics Engineering

Amrutvahini College of Engineering, Sangamner (MS) 422 605

**Abstract:** This paper presents the CORDIC Algorithm for Wireless LAN. The main aim of this topic is to create a VHDL description for CORDIC Algorithm. The basic research is carried out in MATLAB. The VHDL implementation of CORDIC algorithm is based on results obtained from Xilinx ISE simulation. XILINX Spartan-3FPGA is used for hardware implementation.

**Keywords:** CORDIC, FPGA, Wireless LAN, implementation.

## 1. INTRODUCTION:

CORDIC means Coordinate Rotation Digital Computer. CORDIC performs the calculations using shift registers and adders instead of multiplications. By using this it saves much hardware resources. CORDIC is used for calculations of trigonometric functions. It is used for polar to rectangular & rectangular to polar conversion, also in some transformations like DFT, DCT. CORDIC Algorithm mainly developed to find solution for real time navigation problems. CORDIC Algorithm found application in 8087 math coprocessor [1], radar signal processor [6], Hp-35 calculator and robotics. In particular cases, CORDIC Algorithm is used in Wireless LAN at receiver side.

The CORDIC Algorithm calculates the angle of received vector in a signal constellation by means of arctangent function and it is used in receivers to find frequency offset and phase shift.

## 2. CORDIC ALGORITHM:

### 2.1 Mode of operation:

CORDIC rotator works in two modes: Rotation & Vectoring [3].

#### 2.1.1 Rotation mode:

CORDIC can be used to calculate a number of different functions. This explanation shows how to use CORDIC in rotation mode to calculate the sine and cosine of an angle and assumes the desired angle is given in radians and represented in a fixed point format. To determine the sine or cosine for an angle, the y or x coordinate of a point on the unit circle corresponding to the desired angle must be found. Using CORDIC, we would start with the

vector :

$$V_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

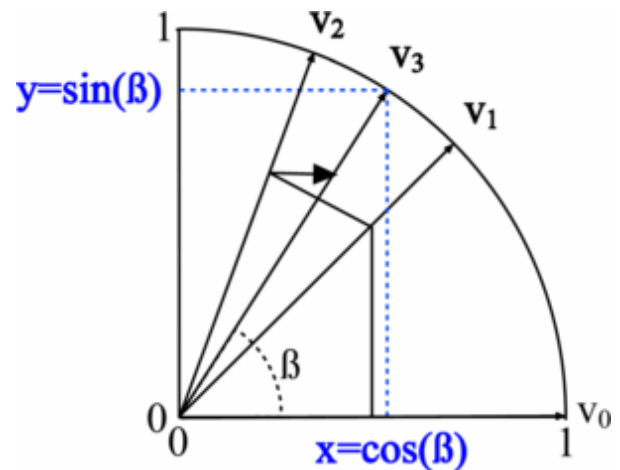


Fig. 1. An example for CORDIC algorithm

In the first iteration, this vector is rotated  $45^\circ$  counter clockwise to get the vector  $V_1$ . Successive iterations rotate the vector in one or the other direction by size-decreasing steps, until the desired angle has been achieved. Step  $i$  size is  $\arctan(1/(2^{i-1}))$  for  $i = 1, 2, 3, \dots$

More formally, every iteration calculates a rotation, which is performed by multiplying the vector  $v_{i-1}$  with the rotation matrix  $R_i$ .

$$v_i = R_i v_{i-1}$$

The rotation matrix is given by:

$$R_i = \begin{bmatrix} \cos Y_i & -\sin Y_i \\ \sin Y_i & \cos Y_i \end{bmatrix}$$

Using the following two trigonometric identities:

$$\cos \alpha = \frac{1}{\sqrt{1 + \tan^2 \alpha}}$$

$$\sin \alpha = \frac{\tan \alpha}{\sqrt{1 + \tan^2 \alpha}}$$

the rotation matrix becomes:

$$R_i = \frac{1}{\sqrt{1 + \tan^2 Y_i}} \begin{bmatrix} 1 & -\tan Y_i \\ \tan Y_i & 1 \end{bmatrix}$$

The expression for the rotated vector then becomes:

$$v_i = \frac{1}{\sqrt{1 + \tan^2 Y_i}} \begin{bmatrix} 1 & -\tan Y_i \\ \tan Y_i & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix}$$

Where  $x_{i-1}$  and  $y_{i-1}$  are the components of  $v_{i-1}$ . Restricting the angles  $y_i$  so that  $\tan Y_i$  takes on the values  $\pm 2^{-i}$ , the multiplication with the tangent can be replaced by a division by a power of two, which is efficiently done in digital computer hardware using a bit shift. The expression then becomes:

$$v_i = K_i \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_{i-1} \\ y_{i-1} \end{bmatrix}$$

Where

$$K_i = \frac{1}{\sqrt{1 + 2^{-2i}}}$$

and  $\sigma_i$  can have the values of  $-1$  or  $1$ , and is used to determine the direction of the rotation; if the angle  $\beta_{i+1}$  is positive then  $\sigma_i$  is  $+1$ , otherwise it is  $-1$ .

$K_i$  can be ignored in the iterative process and then applied afterward with a scaling factor:

$$K(n) = \prod_{i=0}^{n-1} K_i = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1 + 2^{-1}}}$$

which is calculated in advance and stored in a table, or as a single constant if the number of iterations is fixed. This correction could also be made in advance, by scaling  $U_0$  and hence saving a multiplication. Additionally it can be noted that:

$$K = \lim_{n \rightarrow \infty} K(n) = 0.6072529350$$

to allow further reduction of the algorithm's complexity.

After a sufficient number of iterations, the vector's angle will be close to the wanted angle  $\beta$ . For most ordinary purposes, 40 iterations ( $n = 40$ ) is sufficient to obtain the correct result to the 10th decimal place.

The only task left is to determine if the rotation should be clockwise or counterclockwise at each iteration (choosing the value of  $\sigma$ ). This is done by keeping track of how much the angle was rotated at each iteration and subtracting that from the wanted angle; then in order to get closer to the wanted angle  $\beta$ , if  $\beta_{n-1}$  is positive, the rotation is clockwise, otherwise it is negative and the rotation is counter clockwise.

$$\beta_i = \beta_{i-1} - \sigma_i Y_i, \quad Y_i = \arctan 2^{-i}$$

The values of  $Y_n$  must also be recomputed and stored. But for small angles,

$$\arctan(Y_n) = Y_n$$

In fixed point representation, reducing table size.

As can be seen in the illustration above, the sine of the angle  $\beta$  is the y coordinates of the final vector  $v_n$ , while the x coordinate is the cosine value.

After a sufficient number of iterations, the vector's angle will be close to the wanted angle  $\beta$ . For most ordinary purposes, 40 iterations ( $n = 40$ ) is sufficient to obtain the correct result to the 10th decimal place.

## 3. SCHEME OF IMPLEMENTATION:

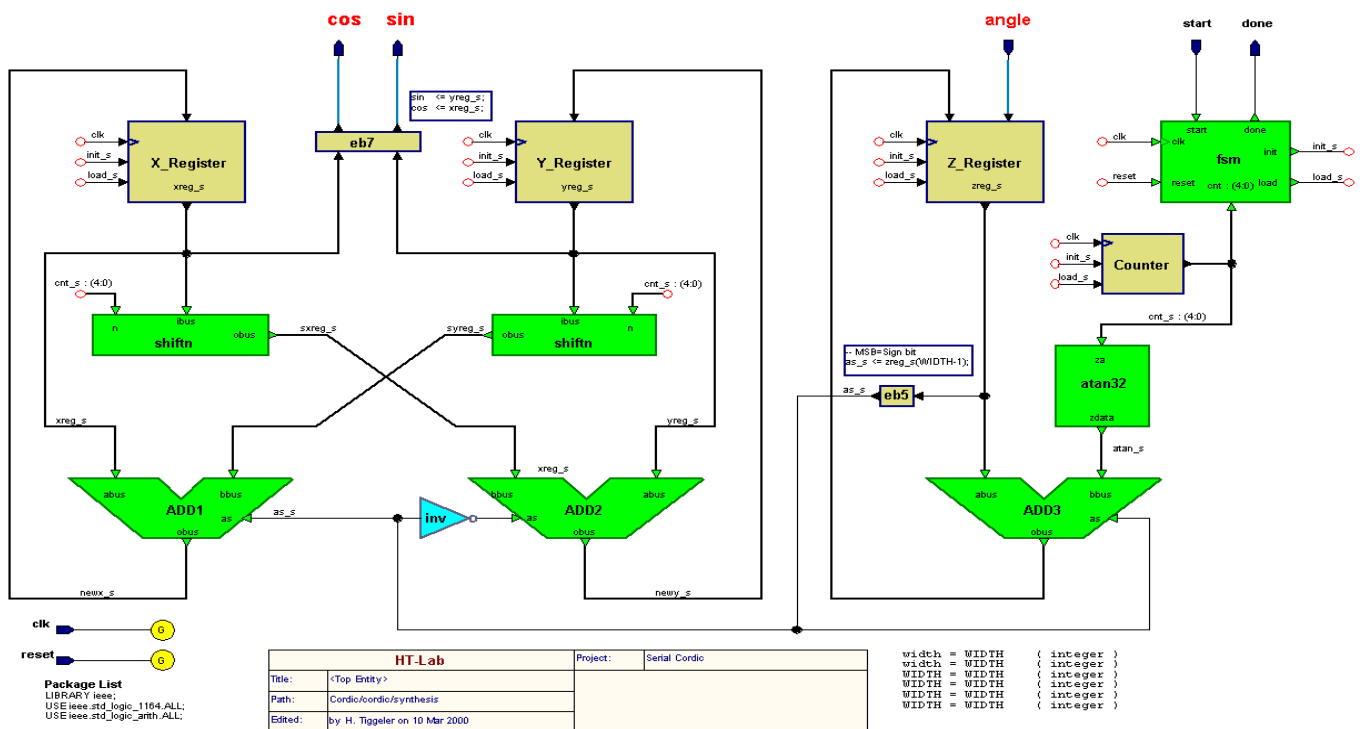
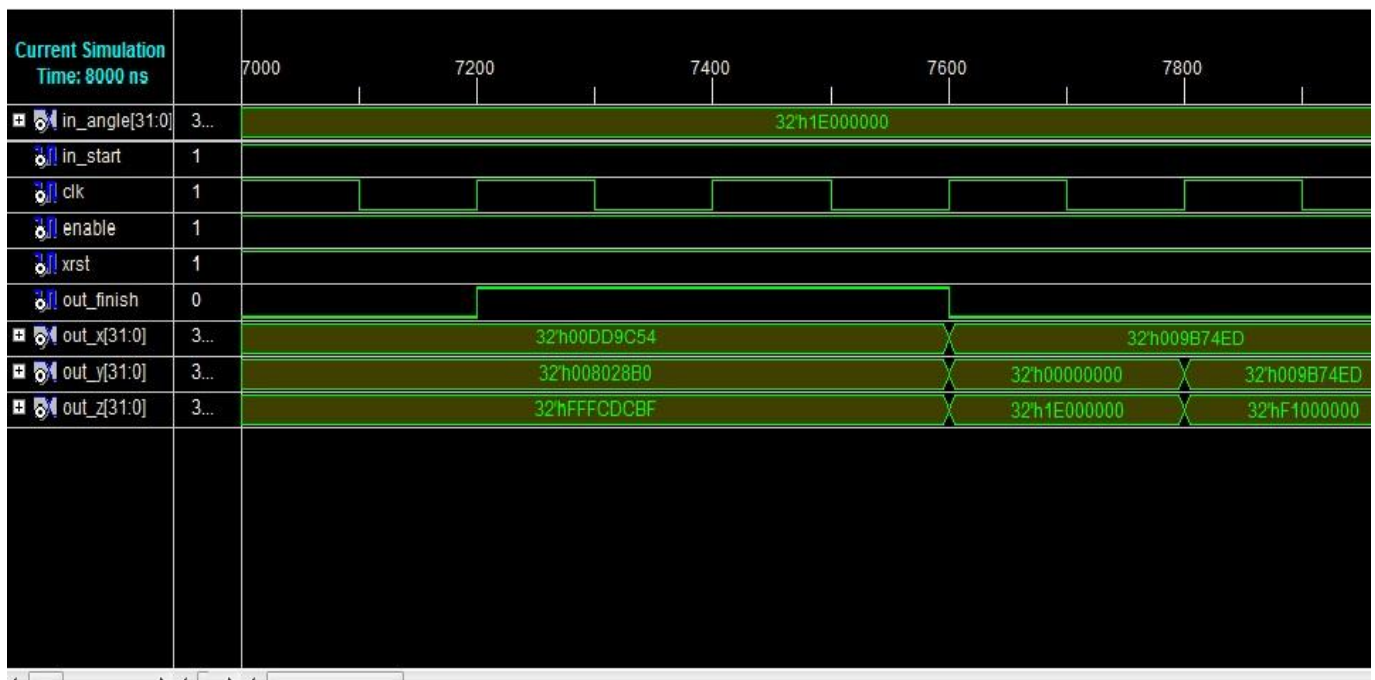


Fig. 2. Block diagram for CORDIC Algorithm.

## 4. RESULT:



## 5. CONCLUSION:

CORDIC Algorithm is most powerful tool which is used in DSP toolbox. This paper shows that this tool is available for use in FPGA based computing machines. This paper represents method to calculate angle in wireless LAN receiver block by using CORDIC Algorithm.

## REFERENCES:

- [1] Duprat J., and Mullar J. M. "The CORDIC Algorithm: New results for VLSI implementation", IEEE Transactions on Computers, Vol. 42 pp. 168-178, 1993.
- [2] Ahmed H. M., Delosme J. M., Morf M., "Highly concurrent structure for matrix arithmetic and signal processing" IEEE comput. Mag. Vol. 15, 1982, pp. 65-82.
- [3] Valls, J., Kuhalmann, M., Parhi K.K., "Efficient mapping of CORDIC Algorithms for FPGA Based computers", 1998.
- [4] Terry J., Heiskala J.: "OFDM Wireless LANs: A Theoretical and practical Guide", Indianapolis, Ind.: Sams, 2002.
- [5] Alfke P., "Efficient shift registers, LFSR counters and long pseudo random sequence generator", Xilinx application note, August, 1995.
- [6] Andraka R. J. "Building a high performance Bit serial processor in an FPGA", proceeding of design supercon 96, Jan 1996, pp 5.1-5.21.
- [7] Deprettere E. Dewilde P., and Udo R. "Pipelined CORIC Architecture for fast VLSI filtering and array processing" proc.ICASSP'84, 1984, pp. 41.A.6.1-41.A.6.4.
- [8] Despain A. M., Fourier Transform Computations using CORDIC Iteration" IEEE transactions on computers Vol. 23, 1974, pp.93-1001.
- [9] Volder J., "Binary computation algorithm for coordinate rotation and function generation" Convair Report IAR-1 148 Aeroelectrics Group June 1956.
- [10] Walther J. S. "A unified algorithm for elementary functions" Spring Joint computer Conf 1971, proc., pp.379-385.
- [11] Nee R., Prasad R.: "OFDM for Wireless Multimedia Communications", Boston: Artech house, 2000.