RESEARCH ARTICLE OPEN ACCESS

# CORDIC Algorithm for WLAN

Mr. Gadekar S. R.
(Department of Electronics Engineering, Pune University, Maharashtra)

**ABSTRACT**
This research aims to implement CORDIC Algorithm for WLAN. The design is coded using VHDL language and for the hardware implementation XILINX Spartan-3FPGA is used. VHDL implementation is based on results obtained from Xilinx ISE simulation.
***Keywords:*** CORDIC, FPGA, implementation, WLAN.

## I. INTRODUCTION

CORDIC (for COordinate Rotation DIgital Computer), also called as the digit-by-digit method and Volder's algorithm. It is a simple and efficient algorithm to calculate hyperbolic and trigonometric functions. It is commonly used when no hardware multiplier is available (e.g., simple microcontrollers and FPGAs) as the only operations it requires are addition, subtraction, bit shift and table lookup.

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.

## II. Wireless LAN

A wireless LAN (WLAN) is a wireless computer network that connects two or more devices using a wireless distribution method within a limited area such as a home, school, computer labs, or office building. By using this WLAN users can move around within a local coverage area and still be connected to the network, and can provide a connection to the wider Internet. Most modern WLANs are based on IEEE 802.11 standards, marketed under the Wi-Fi.

## III. CORDIC ALGORITHM

### 3.1 Mode of operation

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

#### 3.1.1 Rotation mode

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 :

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

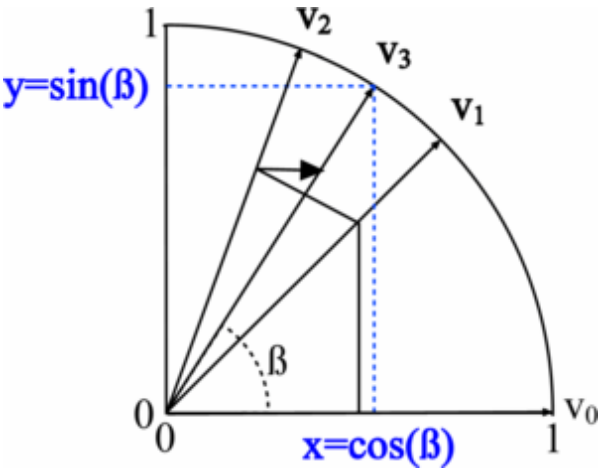



Fig. 1 An example for CORDIC algorithm

In the first iteration, this vector is rotated 45° 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, ….

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:

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

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:

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

The expression for the rotated vector then becomes:

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

Where $x_{i-1}$ and yi-1 are the components of $v_{i-1}$. Restricting the angles yi so that $\tan \gamma_i$ takes on the values +/-$2^{-i}$, 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:

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

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 $v_0$and hence saving a multiplication. Additionally it can be noted that:

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

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 β. 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 counters clockwise at each iteration (choosing the value of σ). 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 β, if $\beta_{n-1}$ is positive, the rotation is clockwise, otherwise it is negative and the rotation is counter clockwise.

$\beta_i = \beta_{i-1} - \sigma_i \gamma_i$. $\gamma_i = \arctan 2^{-i}$

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

Arctan ($\gamma_n$) =$\gamma_n$

In fixed point representation, reducing table size.

As can be seen in the illustration above, the sine of the angle β 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 β. For most ordinary purposes, 40 iterations (n = 40) is sufficient to obtain the correct result to the 10th decimal place.

## IV. SCHEME OF IMPLEMENTATION

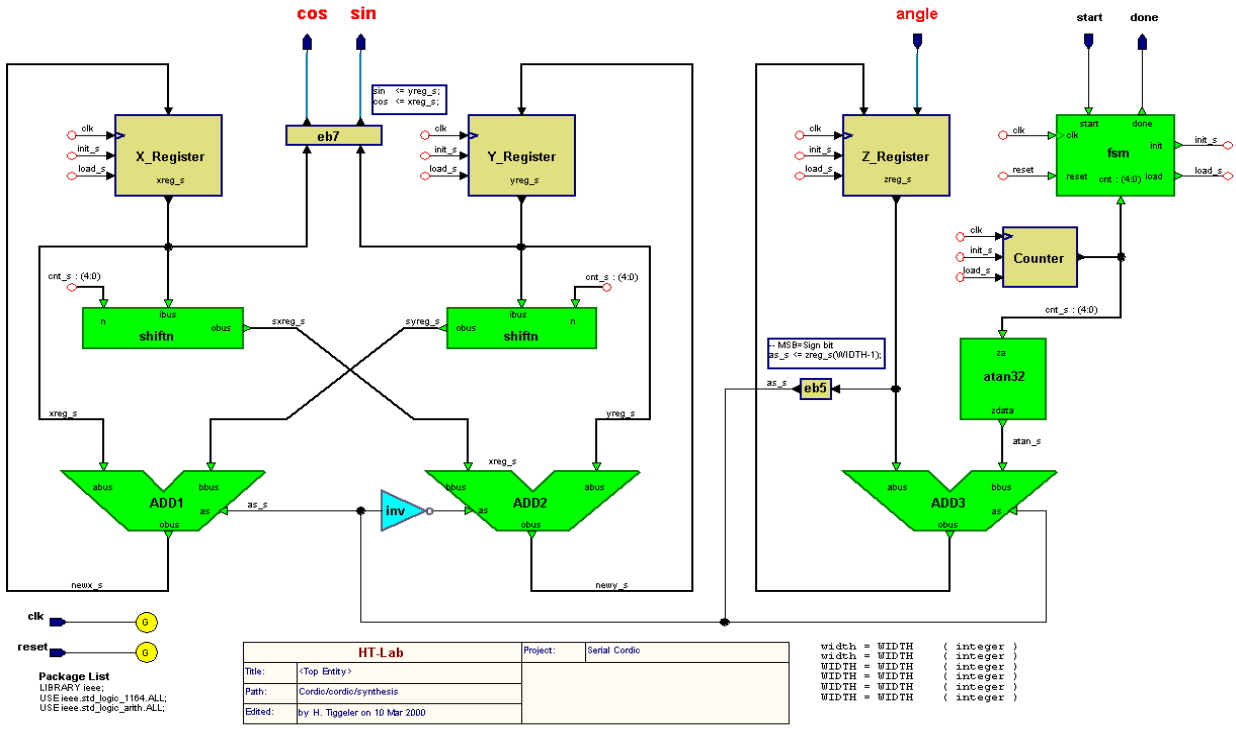
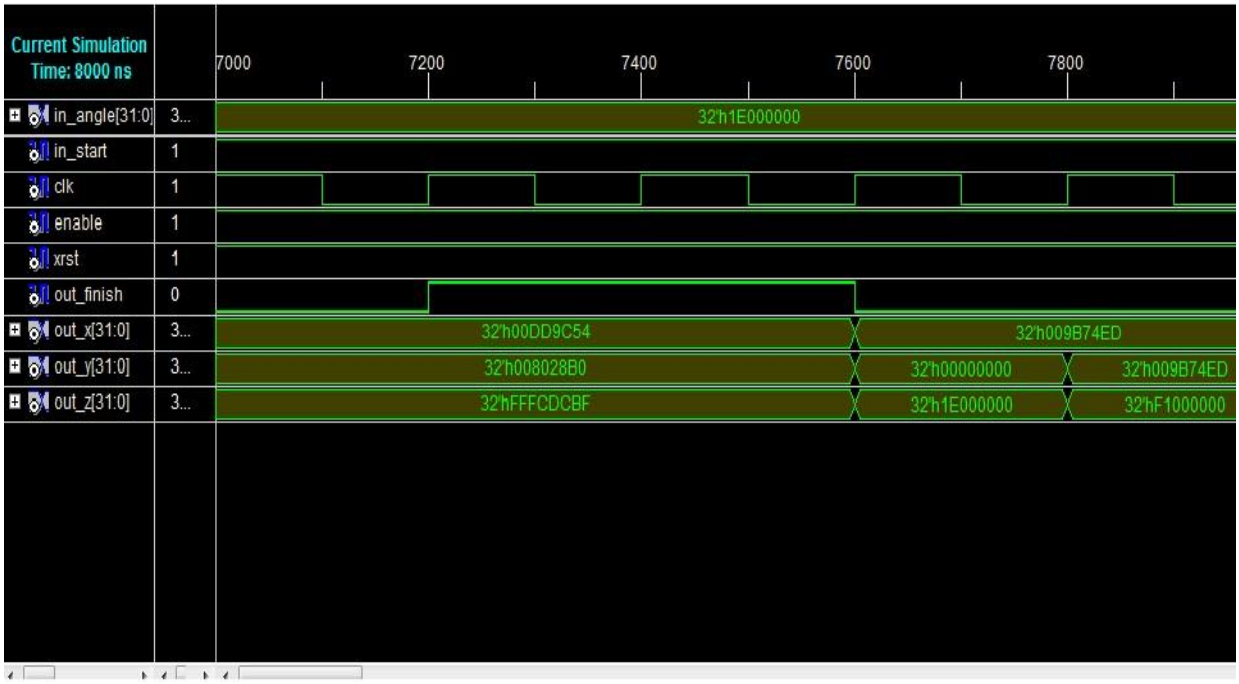



Fig. 2 Block diagram for CORDIC Algorithm.

## V. RESULT




## VI. CONCLUSION

This paper shows that CORDIC Algorithm is useful for use in WLAN technology. By the regularity, the CORDIC based architecture is very suitable for implementation with pipelined VLSI array processors.

This paper represents method to calculate angle in wireless LAN receiver block by using CORDIC Algorithm.

**REFERENCES**

[1] 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.

[2] Duprat J., and Mullar J. M., The CORDIC Algorithm: New results for VLSI implementation, *IEEE Transactions on Computers, Vol. 42* pp. 168-178, 1993.

[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] Despain A. M., Fourier Transform Computations using CORDIC Iteration, *IEEE transactions on computers Vol. 23*, 1974, pp.93-1001.

[8] 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.

[9] Nee R., Prasad R., *OFDM for Wireless Multimedia Communications*, Boston: Artech house, 2000.

[10] Walther J. S., A unified algorithm for elementary functions, *Spring Joint computer Conf 1971*, proc., pp.379-385.

[11] Volder J., *Binary computation algorithm for coordinate rotation and function generation*, Convair Report IAR-1 148 Aeroelectrics Group June 1956.