International Journal of Electronics Communication and Computer Engineering
Volume 11, Issue 4, ISSN (Online): 2249–071X

CORDIC Algorithm: Its Advancement and Applications for Communication

Raju Kumar [1*] and Sandip Nemade [2]
[1] Department of Electronics and Communication TIT, Bhopal, India.
[2] Department of Electronics and Communication TIT, Bhopal, India.
*Corresponding author email id: rajukr40@gmail.com

Date of publication (dd/mm/yyyy): 15/07/2020

Abstract – **CORDIC (Coordinate rotation digital computer) algorithm is used to perform in hyperbolic and trigonometric functions. The CORDIC algorithm has made a tremendous development in the field of architectures and algorithm design for high-performance and low-cost hardware solutions of various applications. It is an eye-catching in the field of academic as well as in industries for countless purposes for example, MIMO System,8087 math coprocessor, HP-35 calculators, radar signal processors, robotics and SDRs. CORDIC has been utilized for applications in miscellaneous areas such as signal and image processing, communication systems, robotics and 3-D graphics. In this review article, we briefly discuss about the key developments in the CORDIC algorithms and architectures along with their potential and future applications.**

Keywords – **CORDIC, Digital Arithmetic, Digital Signal Processing (DSP) Chip, VLSI.**

I. INTRODUCTION

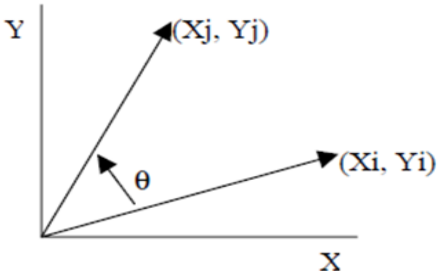
CORDIC algorithm is also named as Volder's algorithm which is an effective algorithm to be used for hyperbolic and trigonometric functions evaluation. Due to its iterative shift-add/substract operations, such algorithm has played a remarkable role in hardware solutions. The strategic concept of CORDIC arithmetic is based on the simple and ancient principles of two-dimensional geometry. However, the iterative design of a computational algorithm for its application was first termed by Jack E. Volder in 1959 [1]. Since for a decades, the CORDIC algorithm has made a tremendous development in the field of architectures and algorithm design for high-performance and low-cost hardware solutions of various applications. Moreover, it eradicates the indigence for explicit multipliers and is condign for calculating a variant of functions, such as sine, cosine, arcsine, arccosine, arctangent, vector magnitude, real and complex multiplications, division, square root, hyperbolic, eigenvalue estimation, singular value decomposition, QR factorization, logarithmic functions and others. CORDIC algorithm is the outcome of the general equation required for vector rotation and is highly efficient, having low complexity and robust technique to calculate exponential function. Now a days it is an eye-catching in the field of academic as well as in industries for countless purposes for example, MIMO System,8087 math coprocessor, HP-35 calculators, radar signal processors, robotics and SDRs. Also CORDIC demonstrated implicant performance in telecommunication notably in processors design for wireless modem. Some of the distinctive methodologies for reducing complexity implementation are engaged on minimization of the complexity of scaling operation and barrel-shifter in the CORDIC engine. Latency of implementation is an inherent disadvantage of the conventional CORDIC algorithm. Angle recoding schemes, mixed-grain rotation and higher radix CORDIC have been established for decreased latency realization. Parallel and pipelined CORDIC have been proposed for high-throughput computation. However, in recent years, a renewed interest in decimal computing has arisen, since it is essential for many applications and scopes [2]. This review article covers a detailed survey of the developments of algorithms, architectures and applications of CORDIC, which would be beneficial for future research development.

II. CORDIC Theory: An Algorithm for Vector Rotation

The Rotation Transform

Every trigonometric functions can be enumerated from function applying vector rotations. The CORDIC algorithm impliment an digit by digit method of performing vector rotations by irrational angles applying only shift and add operations. The algorithm is derived using the general rotation transform.

The CORDIC algorithm performs a planar rotation. Graphicall y, planar rotation means transforming a vector (Xi, Yi) into a new vector (Xj, Yj).



Planar rotation for a vector of (Xi, Yi) is characterize using a matrix form i.e.

$$\begin{bmatrix} X_j \\ Y_j \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \end{bmatrix} \quad (1)$$

The angle rotation can be executed in several steps, using an iterative process. Each step completes a small part of the rotation. Many steps will compose one planar rotation. A single step is defined by the following equation:

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} \cos\theta_n & -\sin\theta_n \\ \sin\theta_n & \cos\theta_n \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (2)$$

Equation 2 can be modified by eliminating the $\cos\theta_n$ factor.

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -\tan\theta_n \\ \tan\theta_n & 1 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (3)$$

Equation 3 requires three multiplies, compared to the four needed in equation 2.

Additional multipliers can be eliminated by selecting the angle steps such that the tangent of a step is a power of 2. Multiplying or dividing by a power of 2 can be implemented using a simple shift operation.

The angle for each step is given by,

$$\theta_n = \arctan\left(\frac{1}{2^n}\right) \quad (4)$$

All iteration-angles summed must equal the rotation angle.

$$\sum_{n=o}^{\infty} s_n \theta_{n=\theta} \quad (5)$$

Where $S_n = \{-1; +1\}$ (6)

This results in the following equation for tan θ_n.

$$\tan\theta_n = S_n 2^{-n} \quad (7)$$

Combining equation 3 and 7 results in

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (8)$$

Besides for the $\cos\theta_n$ coefficient, the algorithm has been reduced to a few simple shifts and additions. The coefficient can be eliminated by pre-computing the final result. The first step is to rewrite the coefficient.

$$\cos\theta_n = \cos\left(\arctan(\tfrac{1}{2^n})\right) \quad (9)$$

The second step is to compute equation 9 for all values of 'n' and multiplying the results, which we will refer to as K.

$$K = \frac{1}{P} = \prod_{n=0}^{\infty} cos\left(\arctan\left(\frac{1}{2^n}\right)\right) \approx 0.607253 \quad (10)$$

K is constant for all initial vectors and for all values of the rotation angle, it is normally referred to as the congregate constant. The derivative P (approx. 1.64676) is defined here because it is also commonly used.

We can now formulate the exact calculation the CORDIC performs.

$$\begin{cases} X_j = K(X_i cos\theta - Y_i sin\theta) \\ Y_j = K(Y_i cos\theta + X_i sin\theta) \end{cases} \quad (11)$$

Because the coefficient K is pre-computed and taken into account at a later stage, equation 8 may be written as,

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \begin{bmatrix} X_n \\ Y_n \end{bmatrix} \quad (12)$$

Or as

$$\begin{cases} X_{n+1} = X_n - S_n 2^{-2n} Y_n \\ Y_{n+1} = Y_n + S_n 2^{-2n} X_n \end{cases} \quad (13)$$

At this point a new variable called 'Z' is introduced. Z represents the part of the angle θwhich has not been rotated yet.

$$Z_{n+1} = \theta - \sum_{i=0}^{n} \theta_i \quad (14)$$

For every step of the rotation Sn is computed as a sign of Zn.

$$S_n = \begin{cases} -1 & if\ Z_n < 0 \\ +1 & if\ Z_n \geq 0 \end{cases} \quad (15)$$

Combining equations 5 and 15 results in a system which reduces the not rotated part of angle θ to zero.

Computing Sine and Cosine functions

Sine and Cosine can be calculated using the first CORDIC scheme which calculates: $[X_j, Y_j, Z_j] = [P(X_i\cos(Z_i) - Y_i\sin(Z_i)),\ P(Y_i\cos(Z_i) + X_i\sin(Z_i)),\ 0]$

By using the following values as inputs $X_i = \frac{1}{P} = \frac{1}{1.6467} \approx 0.60725,\ Y_i = 0,\ Z_i = \theta$

The core calculates: $[X_j, Y_j, Z_j] = [\cos\theta,\ \sin\theta,\ 0]$

The input Z takes values from –180degrees to +180 degrees where:

0x8000 = –180degrees

0xEFFF = +80degrees

But the core only converges in the range –90degrees to +90degrees.

The other inputs and the outputs are all in the range of –1 to +1. The congregate constant P represented in this format results in: $Xi = 2^{15} \bullet P = 19898(dec) = 4DBA(hex)$

Example:

Calculate sine and cosine of 30degrees.

First the Angle has to be Calculated:

$360\text{deg} \equiv 2^{16}$

$1\text{deg} \equiv \frac{2^{16}}{360}$

$30\text{deg} \equiv \frac{2^{16}}{360} \bullet 30 \approx 5461(dec) = 1555(hex)$

The core calculates the following sine and cosine values for Zi = 5461:

Sin : 16380(*dec*) = 3FFC(*hex*)

Cos : 28381(*dec*) = 6EDD(*hex*)

The outputs represent values in the –1 to +1 range. The results can be derived as follows:

$2^{15} \equiv 1.0$ $\quad\quad$ $2^{15} \equiv 1.0$

$16380 \equiv \frac{1.0}{2^{15}} \bullet 16380 = 0.4999$ $\quad\quad$ $28381 \equiv \frac{1.0}{2^{15}} \bullet 28381 = 0.8661$

Where as the result should have been 0.5 and 0.8660.

Table 1. Sin/Cos outputs for some common angles.

	0 deg	**30 deg**	**45 deg**	**60 deg**	**90 deg**
Sin	0x01CC	0x3FFC	0x5A82	0x6EDC	0x8000
Cos	0x8000	0x6EDD	0x5A83	0x4000	0x01CC
Sin	0.01403	0.49998	0.70709	0.86609	1.00000
Cos	1.00000	0.86612	0.70712	0.50000	0.01403

Although the core is very accurate small errors can be introduced by the algorithm (see example and results table). This should be only a problem when using the core over the entire output range, because the difference between +1 (0x7FFF) and –1 (0x8000) is only 1bit.

III. Architecture for CORDIC Algorithm

The basic architecture for the implementation of the CORDIC algorithm for the computation of sine and cosine functions is shown in below figure. This architecture is simplified according to the original CORDIC algorithm for hardware implementation. It uses only 3 adder/subtractor blocks, 2 shifters, 1 counter, and a ROM table for storing the values of the tangents of the angles. The precision is determined by the maximum count of *i* variable.

The Volder's algorithm implicate rotation of a vector on two dimentional plane in circular, linear and hyperbolic coordinate systems subject to function to be determined. Trajectories for the vector for successive CORDIC iterations are shown in below figure. Algorithm implements rotation iteratively using a series of explicit incremental rotation angles selected so that, each iteration is performed by shift and add operation. The norm of a vector in these coordinate systems is defined as $A_k = \sqrt{(I_k^2 + Q_k^2)}$ where m € {1, 0, -1} represents a circular, linear or hyperbolic coordinate system respectively. The criterion of rotation trajectory of circle i.e represented by $x^2+y^2 = 1$ in the circular coordinate system. Similarly, the rotation trajectory in the hyperbolic and linear coordinate systems is interpreted by the function $x^2- y^2 = 1$ respectively. The CORDIC algorithm operates in two distinct modes i.e. rotation and vector mode. The defined angle is achieved by normal rotation in rotation mode and when the angle is undefined, unknown angle of a vector by performing a finite number of micro rotations performed in vectoring mode.

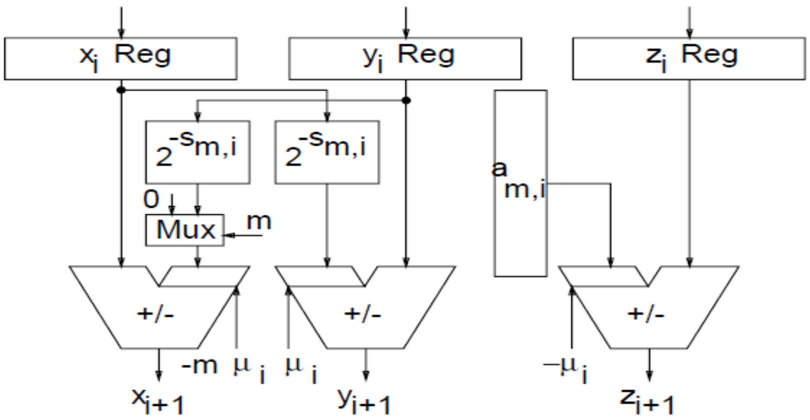


Fig. 2. Basic structure of a processing element for one CORDIC iteration.

An adder or substractor computes sum or difference is completely depends on selected input i.eoperand is negative or positive. The basic cell of adder or substractor is disintegrated with 4 bits input of two functions each respectively. One compute the output and ther to transmit the carry. According to this an N-bit A/S can fit in (2N+1)/2 CLB's (configurable logic block). The supplementary half CLB is required to recommend the least significant bit (LSB) encased subtraction. The critical path determines ripple carry propagation and routing delay of the A/S wire. In this illustration net has a fan-out of 2N, i.e. decrease in performance of the circuit and exhibits disadvantage of conventional CORDIC implementation. Such redundant arithmetic could be used to increase the speed of CORDIC and application evades the carry propagation from the LSB to the most significant bit (MSB), due to it's carry-free property. Redundant arithmetic which have a long propagation delay is good to accelerate operations. On the other hand redundant arithmetic also has some disadvantages. For example, it is impossible to detect the sign of a redundant number without checking all the digits which expects propagation from the MSB to the LSB. A different problem, redundant arithmetic which uses digit set {−1, 0, 1}, and required more hardware to execute simple tasks, than the conventional digit set {−1, 1}.

According to results of the research, the redundant arithmetic is more accurate, but it needs much more hardware than the conventional arithmetic and for this reason conventional arithmetic has been used in this work.

From Figure 2.1 it can be observed that the entire architecture can be divided into three major blocks each block involved in the computation of the three variables *X, Y* and *Z*. Moreover it can also be observed that the blocks involved in the computation of *X* and *Y* are nearly similar. This simple and repetitive nature of the CORDIC architecture enables in a simplified design process.

The Angle Computation Unit (Z-Unit)

The unit involved in the computation of the values of Z required for each of the iterative steps involved in the CORDIC algorithm is the most crucial block of the architecture as it involves the design of certain control signals which control the X and Y Units. The third equation of the CORDIC algorithm in (6) is restated here for the purpose of convenience.

$$Zk + 1 = [Zk - .\partial k.\varepsilon k] \qquad (8)$$
$$\partial k = \mathrm{sgn}(Zk)$$

The iterations utilize the values of $\varepsilon_k = tan^{-1}(2^{-k})$ during the progress of the computation. Eight iterations are performed in total in our scheme thereby ensuring that the final value of the function is obtained upto an accuracy of 8-bits. The Z-Unit thus requires a look-up table for the eight values of $\varepsilon_k = tan^{-1}(2^{-k})$ and a ROM has been created for the same. A 3-bit counter is used to access the ROM in each of the iterations. A clocking scheme has been designed that ensures that the iterative additions are performed and a minimal use of latches has been made for the same. The Z-Unit has been implemented as a cascade of two important blocks *Maincell1* and *Maincell2*.

IV. CORDIC TAXONOMY

The CORDIC algorithm has developed over the years to ensemble unpredictable needs of applications from conventional non redundant to redundant nature. The implementation of redundant arithmetic introduced high latency in conventional CORDIC. Subsequently, numerous modifications have been anticipated for redundant CORDIC algorithm to realize decrease in iteration delay, latency, area and power consumption. Figure 3 depicts the development of the unfolded rotational CORDIC algorithms. The taxonomy is objectively amusing, which represents taxonomy in top-down approach. CORDIC is generally categorised as non redundant CORDIC and redundant CORDIC based on the number system. The foremost disadvantage of the conventional CORDIC algorithm [3, 4] was low throughput and high latency because carry propagate adder which is used for the application of iterative equations. This challenged the simplicity and novelty of the CORDIC algorithm fascinating the thoughtfulness of several researchers to device methods to decrease the latency. The observable solution is to decrease the time for every iteration or the number of iterations or both. The CORDIC arithmetic has been utilized to decrease the time for every repetition/ iteration of the conventional CORDIC, executed as the features of different pipelined and non pipelined unfolded implementations of the rotational CORDIC.

International Journal of Electronics Communication and Computer Engineering
Volume 11, Issue 4, ISSN (Online): 2249–071X

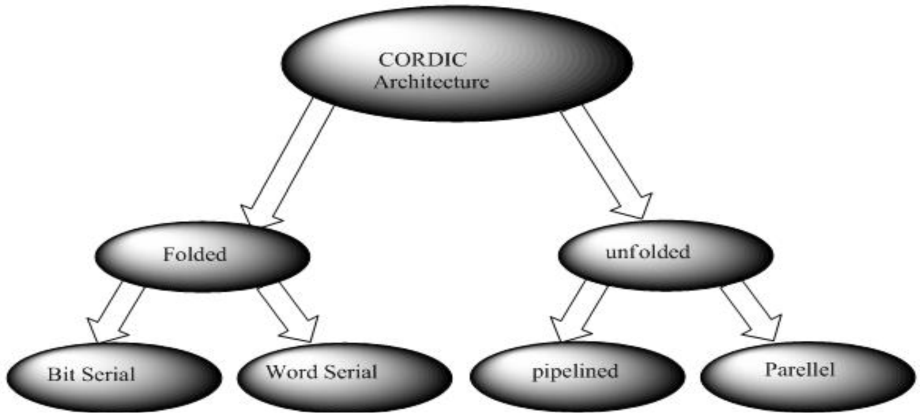


Fig. 3. Taxonomy of CORDIC Architecture.

V. Applications to Communication

CORDIC algorithm can be used for efficient implementation of various functional modules in a digital communication system [5]. The circular coordinate system in communications is used CORDIC in one or both CORDIC operating modes. The RM-CORDIC is generally used to generate mixed signals, while the VM-CORDIC is mainly used to estimate phase and frequency parameters. We concisely framework here some of the vital communication applications.

1. *Direct Digital Synthesis*

Direct digital synthesis is the process of creating sinusoidal waveforms directly in the digital domain. A direct digital synthesizer (DDS) contains a phase accumulator and a phase-to-waveform converter [6, 7]. An increments of the phase-generation circuitas per the normalized carrier frequency in every cycle feeds the phase information to the phase-to-waveform converter. The phase-to-waveform converter could be realized by an RM-CORDIC and the cosine - sine waveforms are attained respectively *via* the CORDIC outputs.

2. *Analog and Digital Modulation*

A generic scheme to use CORDIC in RM for digital modulation, where the phase-generation unit is changed to produce the phase as per normalized carrier and the modulating frequencies, respectively, which are the phase of modulating component. Moreover, through suitable selection of the parameters and the CORDIC inputs, the generic scheme could be used for digital realization of analog amplitude modulation (AM), phase modulation (PM), frequency modulation (FM) as well as the digital modulations, e.g., amplitude shift keying (ASK), phase-shift keying (PSK), and frequency-shift keying (FSK) modulators. It could also be used for the up/down converters for quadrature-amplitude modulators (QAM) and full mixers for complex signals or phase and frequency corrector circuits for synchronization [10].

3. *Other Communication Applications*

Through functioning of the CORDIC in vectoring mode, one can compute the magnitude and the angle of an input vector. The magnitude computation can be used for envelope-detection in an AM receiver or to detect FSK signal if it is placed after mark or space filters [11]. On the other hand, the angle computation in VM CORDIC can be used to detect FM and FSK signals and also estimate phase and frequency parameters [12]. A single VM-CORDIC can be used to perform these computations for the implementation of a slicer for a highorder constellation like the 32-APSK used in DVB-S2. CORDIC circuits functioning in both modes are also essential in digital receivers for the synchronization stage to achieve a phase or frequency estimation followed via a correction stage. This can be done by using two different CORDIC units, to meet the high speed requirement in Costas loop for phase recovery in a QAM modulation [13, 14]. Beside this, the burst based communication system that needs a preamble for synchronization purposes, e.g., in case of IEEE 802.11a WLANOFDM receivers, can use a single CORDIC unit configurable for both operating modes since the estimation and correction are not performed simultaneously [15, 16]. Furthermore, the CORDIC-based QR disintegration in multi-input-multi-output (MIMO) systems used to implement V-BLAST sensors and recursive-least-square (RLS) adaptive antenna beam former [17-21].

VI. Conclusion

An increasing trend towards development of CORDIC has arisen an interest amongst the researchers, industrialist and academicians. CORDIC has the potential for a large set of computational tasks involving the evaluation of trigonometric and transcendental functions, calculation of multiplication, division, square-root and logarithm, solution of linear systems, QR-decomposition, SVD and others. Additionally, CORDIC is implemented by a simple hardware through repeated shift-add operations that has made it an attractive choice for a wide variety of applications. Its applications in numerous different areas comprising signal processing, image processing, communication, robotics and graphics have been explored. Latency of computation is the major limitation for CORDIC as we do not have well-organized algorithms for its parallel implementation. Furthermore, different advanced algorithms may be explored in detail and compared with in future work.

References

[1] M.F. Cowlishaw, Decimal floating point: algorism for computers, in: Proc. 16th IEEE Symp. Computer arithmetic, 2003, pp. 104–111.
[2] A. J. E. Volder, "The CORDIC trigonometric computing technique," IRE ransactions on Electronic Computers, vol. 8, no. 3, pp. 330–334, 1959.
[3] J. S. Walther, "A unified algorithm for elementary functions," in *Proceedings of the AFIPS spring joint computer conference*, pp. 379–385, May 1971.
[4] J. Valls, T. Sansaloni, A. Perez-Pascual, V. Torres, and V. Almenar, "The use of CORDIC in software defined radios: A tutorial," IEEE Commun. Mag., vol. 44, no. 9, 2006.
[5] L. Cordesses, "Direct digital synthesis: A tool for periodic wave generation (part 1)," IEEE signal process. Mag., vol. 21, no. 4, 2004.
[6] J. Vankka, Digital synthesizers and transmitters for software radio. Dordrecht, Netherlands: Springer, 2005.
[7] J. Vankka, "Methods of mapping from phase to sine amplitude in direct digital synthesis," in 50th IEEE International frequency control symposium, Jun. 1996, pp. 942–950.
[8] F. Cardells-Tormo and J. Valls-Coquillat, "Optimisation of direct digital frequency synthesisers based on CORDIC," Electron. Lett., vol. 37, no. 21, 2001.
[9] J. Valls, T. Sansaloni, A. Perez-Pascual, V. Torres, and V. Almenar, "The use of CORDIC in software defined radios: A tutorial," IEEE Commun. Mag., vol. 44, no. 9, 2006.
[10] M. E. Frerking, Digital Signal Processing in Communication Systems. New York: Van Nostrand Reinhold, 1994.
[11] H. Meyr, M. Moeneclaey, and S. A. Fechtel, Digital communication receivers: Synchronization, Channel estimation, and Signal processing. New York: Wiley, 1998.
[12] F. Cardells, J.Valls, V. Almenar, and V. Torres, "Efficient FPGA-based QPSK demodulation loops: Application to the DVB standard," Lectures Notes on Computer Sci., vol. 2438, pp. 102–111, 2002.
[13] C. Dick, F. Harris, and M. Rice, "FPGA implementation of carrier synchronization for QAM receivers," J. VLSI signal process., vol. 36, pp.57–71, 2004.
[14] M. J. Canet, F. Vicedo, V. Almenar, and J. Valls, "FPGA implementation of an IF transceiver for OFDM-based WLAN," in IEEE Workshop on Signal Process. Syst., SIPS'04, 2004, pp. 227–232.
[15] A. Troya, K. Maharatna, M. Krstic, E. Grass, U. Jagdhold, and R. Kraemer, "Low-power VLSI implementation of the inner receiver for OFDM-based WLAN systems," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 55, no. 2, pp. 672–686, Mar. 2008.
[16] Z. Khan, T. Arslan, J. S. Thompson, and A. T. Erdogan, "Analysis and implementation of multiple-input, multiple-output VBLAST receiver from area and power efficiency perspective," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 14, no. 11, pp. 1281–1286, Nov. 2006.
[17] F. Sobhanmanesh and S. Nooshabadi, "Parametric minimum hardware QR-factoriser architecture for V-BLAST detection," Proc. IEE Circuits, Devices and Syst., vol. 153, no. 5, pp. 433–441, Oct. 2006.
[18] G. Lightbody, R. Woods, and R. Walke, "Design of a parameterizable silicon intellectual property core for QR-based RLS filtering," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 4, pp. 659–678, Aug. 2003.
[19] C. M. Rader, "VLSI systolic arrays for adaptive nulling [radar]," IEEE signal process. Mag., vol. 13, no. 4, pp. 29-49, July 1996.
[20] R. Hamill, J.V. McCanny, and R.L. Walke, "Online CORDIC algorithm and VLSI architecture for implementing QR - array processo-

-rs," IEEE Trans. Signal Process., vol. 48, no. 2, pp. 592–598, Feb. 2000.

AUTHOR'S PROFILE

First Author
Raju Kumar, M. Tech. Scholar, Department of Electronics and Communication, Technocrats Institute of Technology (TIT), Bhopal, Madhya Pradesh, India.

Second Author
Prof. Sandip Nemade, Associate Professor, Department of Electronics and Communication, Technocrats Institute of Technology (TIT), Bhopal, Madhya Pradesh, India.