A Review on Hardware Accelerator Design and Implementation of CORDIC Algorithm for a Gaming Application

Trupthi B, Jalendra H E, Srilaxmi C P, Varun M S, Geethashree A

Abstract - Co-ordinate Digital rotation computer is the full form of CORDIC.CORDIC is a process for finding functions using less hardware like shifts, subs/adds and then compares. It's the algorithm used for some elementary functions which are calculated in real-time and many conversions like from rectangular to polar co-ordinate and vice versa. Rectangular to polar and polar to rectangular is an important operation in CORDIC which are generally used in ALUs, wireless communications, DSP processors etc. This paper proposes the implementation of physical design for CORDIC algorithm for polar to rectangular and rectangular to polar conversions, by the use of RTL code in written in Verilog and fed to pre-processor, cordic core and post processor. This type of implementation results with greater efficiency, throughput by reducing power consumption and implanting it with high frequency. Hence due to great efficiency with low power it can be used for gaming applications.

Keywords: CORDIC, rectangular, polar, Co-ordinates, Digital Signal Processing, FPGA, ASIC, efficiency, Power optimization .

I. INTRODUCTION

Co-ordinate rotation Digital Computer or (CORDIC) is an algorithm given by Volder [8]. It is an algorithm that provides an easy way of rotating vector in same single plane by simple addition and shift operation. This works when we rotate the system co-ordinates through fixed angles until which the resultant angle should reduce until zero. The offsets of angles are zero such that the operations performed on X and Y are added and shifted [19]. It is an easy and efficient way to implement this algorithm used to calculate functions like trigonometric, hyperbolic and other functions others like multiplication, division and also some other operations like logarithmic functions, square roots and exponential functions [24].These are implemented in areas of the design algorithm and developed the architectures to provide high performance.

Revised Manuscript Received on June 30, 2020.
***** Correspondence Author
Trupthi B, Department of Electronics and Communication, Vidyavardhaka College of Engineering, Mysore, Karnataka, India, E-mail: trupthibhyregowda@gmail.com
Jalendra H E, Department of Electronics and Communication, Vidyavardhaka College of Engineering, Mysore, Karnataka, India, E-mail: jalendrahe@gmail.com
Srilaxmi C P, Department of Electronics and Communication, Vidyavardhaka College of Engineering, Mysore, Karnataka, India, E-mail: srilaxmi.kowshik@gmail.com
Varun M S, Department of Electronics and Communication, Vidyavardhaka College of Engineering, Mysore, Karnataka, India, E-mail: varunms594@gmail.com
Dr. Geethashree A, Department of Electronics and Communication, Vidyavardhaka College of Engineering, Mysore, Karnataka, India, E-mail: geethashree.a@vvce.ac.in

The conversion of rectangular to polar and polar to rectangular function are the two major operations in CORDIC. Basically, they are the two different co-ordinates to represent the 2D plane. The Rectangular form is represented by a real part (horizontal axis) and an imaginary (Vertical axis) part of the vector. The Rectangular form is shown by a real part (horizontal axis) and an imaginary (Vertical axis) part of the vector [8].The rectangular co-ordinates are in the form of (x, y) where 'x' stands for a horizontal plane and 'y' stands for the vertical plane from the origin likewise, the Polar Form is represented by vector magnitude and angle which is calculated respect to the real axis. The vector value will be the complex number's modulus, polar co-ordinates are in the form of (r, θ) where 'r' stands for distance from the starting point to the estimated point and 'θ' is an angle measured from positive 'x' axis [15]. These conversions are used in different processors like ALU, DSP and various communication purposes like wireless and satellite communication.

II. THE CORDIC ALGORITHM

The abbreviation for CORDIC is Co-ordinate Rotation Digital Computer. The CORDIC algorithm is used to determine the real-time part of calculation for the functions using the iterative rotation of the vector which is fed as input [2]. The CORDIC algorithm performs a planar rotation. A vector (Xi, Yi) gets transformed means the planar rotation into completely a new vector (X_j, Y_j) that is obtained as shown in Fig 1 [1].
The planar rotation of any vector of (Xi, Yi) in matrix form can be as shown in Equation (1).

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

The angle rotation of θ takes various steps. All these steps will comprise one planar rotation. A single step is given by the equation (2)

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

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

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

The multipliers which added are hence eliminated by choosing the angle steps. Where tangent of step is given by power of 2.

Hence the step angle can be given as

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

$$\sum_{n=0}^{\infty} S_n \theta_n = \theta \quad (5)$$

where

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

This gives

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

Combining Equation (3) and Equation (7) we get

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

Fig 1 Planar rotation of (Xi, Yi).

The coefficient can be eradicated fully by pre-determination of the final result. The coefficient can be given as,

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

Equation (9) can be found for all values of 'n', where K is one such constant.

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

The derivative P (approximate 1.64676) is defined.

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

The coefficient K is pre-determined 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

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

Here the Z represents the part of the angle θ which is not rotated.

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

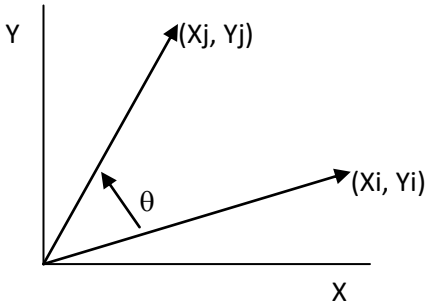
For each step of rotation, value of Sn is determined as a sign of Zn.

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

Combining Equation (5) and Equation (15) gives a system which reduces the un-rotated part of angle θ to zero.

III. CORDIC ALGORITHM ARCHITECTURE

Three fundamental blocks are used to construct any CORDIC Processor cores those are namely pre-processor, the post-processor and the actual CORDIC core. The core part of CORDIC is basically made up of different types of architecture. Then below are the main architectures in the CORDIC algorithm.



A) Iterative Architecture:

Work of shift-add/sub-operations is serial, and architecture of CORDIC core is implemented in sequential configuration, using a single shift-add/sub-stage and feeding back the output. A minimum latency CORDIC core and takes minimum of N cycles to determine the new output. The relationship is that the implementation size of this is related to the precision internally.

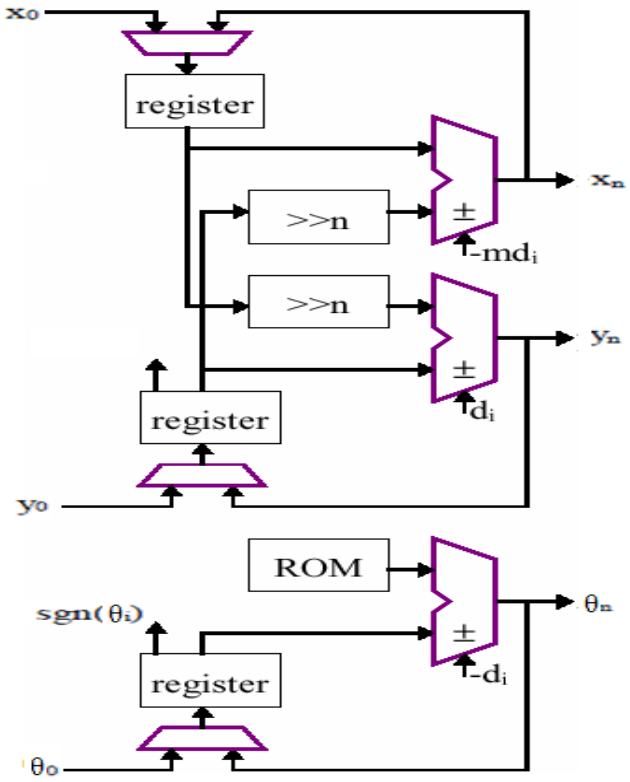


Fig 2: Iterative Architecture [26]

B) Parallel or Cascaded Architecture:

This architecture uses various formulation of Iterative CORDIC. Here it is basically shift-add/sub-stages in CORDIC core with parallel architecture being with it. The throughput or the latency of one such clock cycle is obtained for output of N-bit parallel CORDIC core.

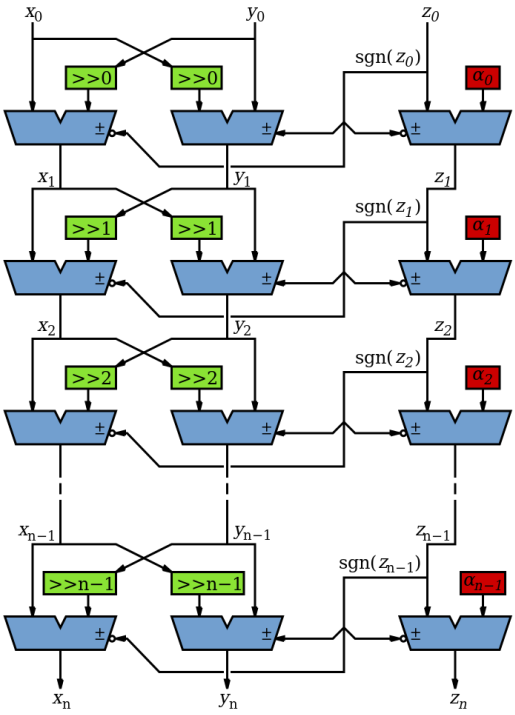


Fig 3: Parallel or Cascaded Architecture

C) Pipelined Architecture:

There is a similarity in the structure used compared to that of a Parallel CORDIC. In between each iteration phase pipeline registers are used. The advantages in pipelined CORDIC is for continuous input values.

IV. SOFTWARE IMPLEMENTATION

For application-specific CORDIC processors, the very common application is in Digital signal processing for vector rotation. In the paper [2], the design of pipeline architecture is defined on the basis of application-specific processor of CORDIC used for the determination of values of sine and cosine. Due to its pipeline architecture, CORDIC's design in rotation mode in a circular way gives overall high system throughput by reducing the latency in every level of pipeline. The synchronizer function is autocorrelation. In the OFDM amplifier, CORDIC is then efficiently used to measure the offset of frequency and to determine the division equation for calculating channels. In the paper [3], by using 130 nm technology, a rapid pipeline CORDIC architecture and auto-correlator are first designed, implemented and then tested. A MATLAB simulation is performed to verify at functional level before the coding in Verilog HDL. Digital signal processing (DSP) has been always driven front in case of DSP applications and also in very-large-scale (VLSI) technology. This raised many issues with DSP applications deployment. These implementations will meet the DSP systems defined real-time sampling rate constraints which covers less space and power consumption. Flexibility, high throughput, low power and low cost are reasons for the development of today's VLSI (FPGA / ASIC) based communication of mobile and wireless medium. The WLAN which is based on OFDM is the next-generation technology, which has two components, namely autocorrelation for the synchronization of the signals which were received and CORDIC for estimation of channel computation and also compensation of OFDM in the wireless receiver. Several components of the electronic circuit, due to the various factors associated throughout the circuitry, do not get the clock at the same time. For generation of frequency and for stability, the design of phase-locked loop is very popular. Authors also discussed in a paper [4] that describes an phase-locked loop which are all-digital is one of all the models of PLL where all the elements used are digital in nature, giving it an advantage over FPGA. PLL's digital design takes the digital component that is flexible for usage and immune to environmental factors such as temperature and parasite capacity. The digital components use, has provided noise immunity, which basically enhances circuit accuracy and functionality. The ADPLL is used in implementations in a wide range as wireless communication, device control, biomedical and many other fields, requiring low power, high speed, and less chip area. The designed ADPLL works on an accumulator for 100 MHz and for 16 bit. The paper [2] discussed the use of the CORDIC algorithm in DSP for vector rotation using the pipeline architecture, which saves area on the silicon substrate and also gives high throughput. In paper three the CORDIC algorithm is used in

an OFDM amplifier in which also saved significant area and power, which are the main constraints in VLSI technology. Paper [4] gave the software implementation of CORDIC in the fields like wireless communication, monitoring and biomedical applications.

With these advantages, there are also drawbacks to the software implementation of CORDIC. With the implementation for DSP, the algorithm must follow an imposed sampling rate and must also save space and power along with it. Thus to make this possible FPGA implementation was introduced to lower the cost and power consumption.

A) Polar to Rectangular Conversions:

The first CORDIC evaluation includes Sine and Cosine as follows

$[X_j Y_j Z_j] = [P(X_i \cos(Z_i) - Y sin(Z_i)), P(Y_l \cos(Z_i) + Xi \sin Zi, 0]$

The following lines are used as inputs:

$$Xi = \frac{1}{P} = \frac{1}{1.6467} \approx 0.60725 \; Y_i = 0 \; Z_i = \theta$$

The core is then calculated as:

$$\left[X_j, Y_j, Z_j\right] = \left[\cos\theta, \sin\theta, 0\right]$$

The input Z takes values from –180º to +180º
Where :
0x8000 = –180º
0xEFFF = +180º
The constant P represented results in:

$$Xi = 2^{15} \times P = 19898(dec) = 4DBA(hex)$$

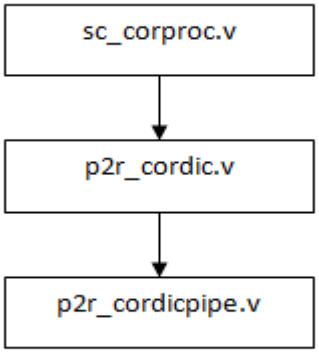


Fig 4: Core Structure of polar to rectangular conversion.

B) Rectangular to Polar Conversion:

The processor of rectangular to polar is constructed around the second CORDIC scheme, this calculates,

$$\left[X_j, Y_j, Z_j\right] = \left[P\sqrt{1+a^2}, 0, \arctan(a)\right]$$

The rectangular co-ordinates of a 2D are given as inputs (Xin, Yin), this takes two 16bit signed Code Structure words. The core then returns the equivalent Polar co-ordinates where here radius is given by Rout and the angle is given by Aout or θ.

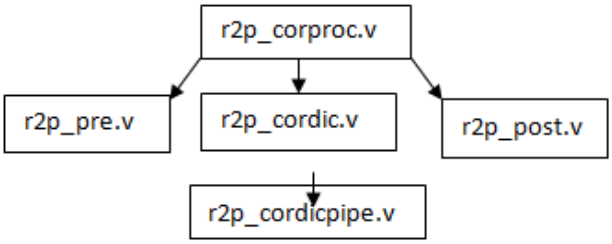


Fig 5: Core Structure of rectangular to polar conversion

V. FPGA IMPLEMENTATION

The Hough Transform (HT) is seen to be employed commonly in image processing or in pattern recognition, such as circles with arcs, ellipses as well as other forms. The authors [5] proposed architecture for straight line detection, implemented on FPGA computers, uses both the gradient algorithm and CORDIC. To minimize the amount of equations, the gradient was implemented to allow less time, and the CORDIC algorithm has been utilized to make the operators less complicated. Increasing the number of samples will increase performance, and will also down the rate of processing. The paper [6] discussed rotator for CORDIC, which is built with the help of pipelined architecture. It has a unit controller which basically controls the data sequence and then gives the output. A CORDIC operates in 2 main modes: rotation and vectoring. In case of rotation mode, the input vector is first rotated by an angle while in vectoring; the inputs are rotated along x-axis recording the amount of rotation. This model is simulated to transform data of 1K from the Cartesian to polar domain and vice versa. The output data is compared with the initial data to find the error data. Advantages of pipelined architecture are that it has throughput which is high, and uses less memory for storage of angle values, which is free from looping iteratively and delay resulting to zero. The pipelined CORDIC was implemented in FPGA ALTERA cycling, and the design worked at 81.31 MHz with a minimum error value of 0.05. A time and area-efficient CORDIC algorithm are designed by properly choosing the approximation of the Taylor series, which meets the required range of convergence (RoC). The area, latency requirements and accuracy is achieved by manipulating the iterations in the series. The algorithm designed by the authors [7] has lower complexity when compared to other available scaling free algorithms which require expansion of RoC over the whole co-ordinate space. Here the rotational CORDIC mode follows the following some steps. To avoid scaling function, we use sine and cosine function of Taylor series and selecting a proper RoC depending on the values of approximation value which are found in Taylor series.

The designed CORDIC algorithm has a low slice delay of 17% with more slice consumption penalty of about 13%. This is implemented in Xilinx Spartan 2E using the code synthesized in Xilinx ISE9.2i.

The architecture designed in the paper [8] is mainly for Attitude determination in satellites using some and cosine functions. The attitude determination system involves higher accuracy, which is important for finding point accuracy. The proposed CORDIC approach is faster when compared to the on-chip software installed in satellites with FPGA implementation. The sine and cosine values are considered as integer values and use two's complement for representation for MSB of angle for rotation. Before this, the multiplication factor is multiplied to avoid the use of multipliers.

The architecture consists of a clock which makes is a sequential system with required shifts in the clock cycles using barrel shifters. It consists of MUX, adders/subtractions, registers which are co-ordinated using a control unit which produces a lookup table after processing. The RESC and RESET are used for resetting the counter (which count the number of iterations) and the system respectively and SEL for multiplexers. The algorithm was verified using Xilinx ISE 12.1 and implemented on Altera's FPGA. The proposed designed was compared with a C++ program which proved the approach to be 27500 times faster than it. This approach is also used for a calculator with good efficiency in the area.

The proposed architecture in the paper [9] is a combination of parallelism and pipelining method. This processor is speed-area optimized and is applicable for real-time operations. The architecture is formed by combining 'K1 computation architecture' and 'unscaled CORDIC architecture'. In this design, we use the radix-4 method, which in turn reduces the number of iterations when compared to radix-2. This results in a 32-bit output in every eight clock cycle with the completion of CORDIC rotation in 5 cycles for 16-bit precision. Here, the reciprocal of scaling factor (K) is obtained, which is then multiplied with the unscaled factor in an array multiplier. The output of this multiplier will X and Y co-ordinate scaled versions, each of which will be 32 bits.

MATLAB tool is used for processing the algorithm which computes scale factor in parallel with rotation of CORDIC (n/2) clock cycles are needed for determining the 32-bit output accuracy which defines the latency, and the power consumption at 56.96 MHz(operating frequency) is measured to be 380mW using a Power software tool. Thus with the help of the proposed design, speed area are optimized and hence architecture is found. XILINX-FPGA is used for implementing this design which produces a total gate count of 11528.

In another paper, the authors [10] compare the various CORDIC architecture in terms of throughput and area efficiency, which are verified using Spartan -ll FPGA. The CORDIC, being low cost and efficient as various applications like the logarithmic, complex number, solution of linear systems, SVD for signal processing and image processing, scientific computation, etc. The CORDIC algorithm is based on a number of iterations of addition/subtraction or shifting process and doesn't involve division or multiplication. The CORDIC operates in 2 modes. In case of vectoring mode, the input vector fed is studied in terms of its magnitude and phase. In rotation mode, the angle of the input vector is calculated with respect to X and Y co-ordinates. The CORDIC rotator usually analyses arctangent, rectangular and polar co-ordinates and various trigonometric functions.

VI. HARDWARE IMPLEMENTATION

The design is FPGA implementation as it is in a unique way. Further keypad of 4x4 matrix and LCD of 16x2 character is used to build a real time math hardware module [24].Verilog HDL code is used and verified firstly by using simulation results of the ModelSim and later using Altera DE1 board. Results are displayed using keypad and LCD [24].

Xilinx ISE of 8.2 version is used to synthesize the hardware architecture. A Xilinx XC2s200E-PQ 208-5 device is used to map the hardware architecture [9].

This algorithm provides vectors rotation in a single plane by simple add and shift operation to evaluate the basic functions. Those are trigonometric operations, division multiplication and some other like square roots, logarithmic functions and exponential functions [9]. Applications include either in wireless communication or in digital signal processing are microprocessors based which make use of a single instruction and a bunch of addressing modes. These processors are cost-efficient and flexible [9].During a clock cycle, only one shift operation will be executed for every iteration in a sequential design.

This shift operation presents multiplication by two power (-i). Therefore, CORDIC iteration number and the clock period are the factors for total time of vector rotation [25].

The pipelined CORDIC has basically three inputs such as: Clk for clock, Rst for reset (active low), and Angle for the target angle. It also has two outputs, Cos and Sin, which are for cosine and sine result of the target angle, respectively [7].

Two main blocks are here: Quadrant Detector and CORDIC Core.

Cosine and sine value are calculated in the Quadrant Detector and thus, the input angle is obtained [7].

CORDIC Core is the realization of CORDIC's difference equations in pipelined architecture [7].

A register is added in order to reduce the critical delay path at each pipeline stage. As the pipeline stage increases latency also increases leading to penalty [7].

VII. CONCLUSION

This paper gives a review of the CORDIC algorithm. We have surveyed the various implementations of the CORDIC algorithm, mainly the software and FPGA implementation. From the survey, we can conclude that both software implementation and FPGA implementation have advantages and disadvantages of their own.

The paper reviews the work on polar to rectangular and rectangular to polar conversion using CORDIC implementation and their optimization in the power consumption .There is increase in efficiency due to increase in the frequency. In this paper, we study some of the existing CORDIC implementation techniques additionally recommend the use of CORDIC algorithm for conversions. Then further implementations can be carried out by developing an ASIC for this conversion implementation, which reduces the computational cycles hence optimizes the time and speed. By choosing the best hybrid of algorithm and technique which is compatible with the technology used for ASIC development time required for computation and performance can be enhanced.

REFERENCE

1. Katkar, Prajakta J. and Yogesh S. Angal. “Realization of cordic algorithm in DDS: Novel Approch towards Digital Modulators in MATLAB and VHDL.” 2015 International Conference on Information Processing (ICIP) (2015): 355-359.
2. Mandal, Amritakar & Tyagi, K.C. & Kaushik, Brajesh Kumar. VLSI Architecture Design and Implementation for Application Specific CORDIC Processor: 191-193.10.1109/ARTCom.2010.94 (2010).
3. Singhal, Akarshika & Goen, Anjana & Mohapatra, Tanutrushna. FPGA implementation and power efficient CORDIC based ADPLL for signal processing and application: 325-329 (2017).
4. Si Mahmoud Karabernou and Fayçal Terranti. 2005. Real-time FPGA implementation of Hough Transform using gradient and CORDIC algorithm.
5. Image Vision Comput. 23, 11 (October 2005), 1009–1017.
6. T. Adiono and R. S. Purba, "Scalable pipelined CORDIC architecture design and implementation in FPGA," 2009 International Conference on Electrical Engineering and Informatics, Selangor,pp: 646-649 (2009).
7. S. Aggarwal, P. K. Meher and K. Khare, "Area-Time Efficient Scaling-Free CORDIC Using Generalized Micro-Rotation Selection," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 8, pp. 1542-1546(Aug. 2012).Conference on Power, Control and Embedded Systems (2010).
8. Kaushik Bhattacharyya, Rakesh Biswas, Anindya Sundar Dhar, and Swapna Banerjee.Architectural design and FPGA implementation of radix-4 CORDIC processor. Microprocess. Microsyst. 34, 2–4 (March 2010), 96–101.
9. Er. Mnaoj Arora, Er. R S Chauhan, Er. Lalit Bagga, “FPGA Prototyping of Hardware Implementation of CORDIC Algorithm”in International journal of scientific & Engineering Research International Journal of Scientific & Engineering Research, Volume 3, Issue 1,(January-2012).
10. S. Aggarwal, P. K. Meher and K. Khare, "Concept, Design, and Implementation of Reconfigurable CORDIC," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 4, pp. 1588-1592,(April 2016.
11. Angarita, Fabian & Pérez-Pascual, A. & Sansaloni, T. & Vails, J.Efficient FPGA implementation of Cordic algorithm for circular and linear coordinates. 535- 538. 10.1109/FPL.2005.1515779. CORDIC and SVD Implementation in Digital Hardware, Przemysaw M. Szecówka, Piotr Malinowski, 17th International Conference Mixed Design of Integrated Circuits and Systems - MIXDES 2010.
12. Lin, Chih-Hsiu & Wu, An-Yeu. Mixed-scaling-rotation CORDIC (MSR-CORDIC) algorithm and architecture for high-performance vector rotational DSP applications. Circuits and Systems I: Regular Papers, IEEE Transactions on. 52. 2385 - 2396. 10.1109/TCSI.2005.853908 (2005).
13. Xiao-Gang Jiang, Jian-Yang Zhou, Jiang-Hong Shi and Hui-Huang Chen, "FPGA implementation of image rotation using modified compensated CORDIC," 2005 6th International Conference on ASIC, Shanghai, 2005, pp. 752-756.
14. Vadlamani, S. and Wagdy H Mahmoud. “Comparison of CORDIC algorithm implementations on FPGA families.” Proceedings of the Thirty-Fourth Southeastern Symposium on System Theory (Cat. No.02EX540) (2002): 192-196.
15. Shen Lin, M. Marek-sadowska, Emest S. Kuh, “Delay and Area Optimization in Standard-Cell Design” Department of Electrical Engineering and Computer Science University of Clifornia, Berkeley CA 94720, 27th ACM/IEEE Design Automation Conference.
16. S. Hsiao, M. Tsai and C. Wen, "Low Area/Power Synthesis Using Hybrid Pass Transistor/CMOS Logic Cells in Standard Cell-Based Design Environment," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 57, no. 1, pp. 21-25, Jan. 2010.
17. S. Timarchi and M. Alioto, "Ultra-low voltage standard cell libraries: Design strategies and a case study,"IEEE International Conference on Electronics, Circuits and Systems (ICECS), Monte Carlo, 2016, pp. 520-523.
18. M. S. I. bin Hussin, Y. W. Lim, N. A. Kamsani, S. J. Hashim and F. Z. Rokhani, "Development of automated standard cell library characterization (ASCLIC) for nanometer system-on-chip design," IEEE 15th Student Conference on Research and Development (SCOReD), Putrajaya, 2017, pp. 93-97.
19. R. Thapa, S. Ataei and J. E. Stine, "WIP. Open-source standard cell characterization process flow on 45 nm (FreePDK45), 0.18 µm, 0.25 µm, 0.35 µm and 0.5 µm,"IEEE International Conference on Microelectronic Systems Education (MSE), Lake Louise, AB, 2017, pp. 5-6.
20. T. Adiono and R. S. Purba, "Scalable pipelined CORDIC architecture design and implementation in FPGA,"International Conference on Electrical Engineering and Informatics, Selangor, 2009, pp. 646-649.
21. J. Morris, P. Prabhat, J. Myers and A. Yakovlev, "Unconventional Layout Techniques for a High Performance, Low Variability Subthreshold Standard Cell Library,"IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Bochum, 2017, pp. 19-24.
22. G. Ivanova, A. Korshunov, T. Zhukova and I. Tiunov, "Development of mathematical models for standard cells timing analysis at the 28 nm technology and below,"IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), St. Petersburg, 2017, pp. 1401-1405.
23. M. Li, C. Ieong, M. Law, P. Mak, M. Vai and R. P. Martins, "Sub-threshold standard cell library design for ultra-low power biomedical applications," 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, 2013, pp. 1454-1457.
24. Muhammad Nasir Ibrahim ,Chen Kean Tack,Mariani Idroas ,Siti Noormaya Bilmas ,Zuraimi Yahya “Hardware Implementation of Math Module Based on CORDIC Algorithm Using FPGA,”International Conference on Parallel and Distributed Systems (2013).
25. A. Boudabous, F. Ghozzi, M. W. Kharrat and N. Masmoudi, "Implementation of hyperbolic functions using CORDIC algorithm," Proceedings. The 16th International Conference on Microelectronics, ICM 2004., Tunis, Tunisia, 2004, pp. 738-741.
26. Bhattarai, Bibek, FPGA Implementation of CORDIC Processor. 10.13140/RG.2.1.4432.1364 (2013).

AUTHORS PROFILE

Trupthi B is a final year student pursuing her Bachelor of Engineering in the Electronics and Communication department of Vidyavardhaka College of engineering, Mysore. She is also pursuing a two year course specialized in physical design domain in VLSI by a company. She has published a paper previously in titled "Online food ordering system". The current paper "Review on design and implementation of cordic algorithm for accelerator in gaming application" is a part of her final year project and the research work.

Jalendra H E is a final year student pursuing his Bachelor of Engineering in the Electronics and Communication department of Vidyavardhaka College of engineering, Mysore. He is also pursuing a two year course specialized in physical design. The current paper "Review on design and implementation of cordic algorithm for accelerator in gaming application" is a part of his final year project and the research work.

Srilaxmi C P is a final year student pursuing her Bachelor of Engineering in the Electronics and Communication department of Vidyavardhaka College of engineering, Mysore. She is also pursuing a two year course specialized in physical design. She has published a paper previously in titled "Online food ordering system". The current paper "Review on design and implementation of cordic algorithm for accelerator in gaming application" is a part of her final year project and the research work.

Varun M S is a final year student pursuing his Bachelor of Engineering in the Electronics and Communication department of Vidyavardhaka College of engineering, Mysore. He is also pursuing a two year course specialized in physical design. The current paper "Review on design and implementation of cordic algorithm for accelerator in gaming application" is a part of his final year project and the research work.

Dr. Geethashree A received the BE degree in Telecommunication Egineering in 2002, M.Tech in VLSI Design and Embedded Systems in 2010 and PhD in speech processing in 2020 from Visvesvaraya Technological University, Karnataka, India. She is currently working as Associate professor in the department of ECE, Vidyavardhaka college of Engineering, Mysuru, India. Her main research interest includes Expressive Speech Synthesis, Speech Signal Processing and Digital System Design in VLSI .