International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Realization of Optimized CORDIC Core for Implementing Sine and Cosine Operations

Priyanka Shakya, R. C. S. Chauhan

Abstract: ***Cordic, which is an iterative vector rotation calculation for different coordination systems, has been proposed in this paper that has low latency and low area utilization. The main limitation is that in comparison to standard CORDIC, the number of micro-rotations necessary increases with the input angle bit-width which leads to additional stages of micro-rotation. In order to overcome this, the most area utilizing stages are recoded using the two bits of the input angle simultaneously such that our suggested technique can achieve a smaller micro-rotation for bigger bit width applications. In this article, using parallel and pipelined CORDIC architecture a Digital sine and cosine generator is intended and applied which utilises optimized Micro-rotation Angle Recoding algorithms to achieve low latency and reduces area of the design. The proposed work reduces the delay by 34%.***

Keywords: Cordic; Iterative Approach; Pipelined Analysis; Bbr; Mar;

I. INTRODUCTION

CORDIC algorithm is a sort of computerized cycle strategy used for evaluating the numerous mathematical functions both in rotational mode and circular coordinate system for sine/cosine computation. A wide variety of real time and high speed applications have been implemented using sine/cosine function evaluated using CORDIC algorithm [1] such as adaptive filters [2] in digital signal processing (DSP), generating sinusoidal waveforms [3] in communication, robot control [4], and geometric computations [5] in graphics. In this paper, CORDIC algorithm is proposed in a manner that it provides an ease in the complexity of computing of determining the direction of rotation. All micro-rotations can be pipelined and applied for the realization of high-performance CORDIC rotations.

When such an application is adopted, the actual limitation in performance lies in sequentially defining of rotation direction value given as the function of σ_i. If the determination of the rotation direction in all stages can be parallelized, then the respective CORDIC rotations can be performed simultaneously in the micro-rotation phase.

A major characteristic is that for a specified N-bit entry angle CORDIC rotation $2^{-m} \neq \tan^{-1} 2^{-m}, where\ m < \left[\frac{N-\log_2 3}{3}\right]$. In methods already recorded, CORDIC rotations are speeded up using this property:

Revised Manuscript Received on December 30, 2019.

* Correspondence Author
Priyanka Shakya, M. Tech Scholar, Institute of Engineering and Technology, Lucknow (U.P.) India. 226021
R.C.S. Chauhan, Associate Professor, Institute of Engineering and Technology, Lucknow (U.P.) India. 226021

Wang et al. [6] obtained the rotation path from the point z following the first iteration corresponding to a rotation of partly parallel CORDIC (PARA-CORDIC). In one phase, two rotations leading to a more difficult z-data path are conducted in which Phatak et al. [7] have been examining several most significant digits. A standard CORDIC core has 16 phases with 16-bit fixed-point input and 20-bit fixed-point output outcomes in [8] and has very low processing speed. The non-recursive CORDIC [9] utilises the conventional method of CORDIC algorithm and operates at the frequency of 100MHz, which gives its low processing. The computation of the atan2 function based on a CORDIC algorithm [9], this decreases z-way costs by adjusting its structure to FPGA device assets and continuing to benefit from rapid spread logic. It doesn't control the number of iterations and increases the computational complexity. Adaptive Recoding algorithms [10] utilizing the greedy algorithm and require rigorous computation to determine the rotation angle and direction, thus leading to increase in hardware complexity. In the PP-CORDIC design [11], parallel architecture, pipeline processing together with an angle recoding technique is used to accomplish small latency and floating point accuracy but operates at very small frequencies.

Thus for the optimization of the CORDIC, the huge amount of iterations are needed to be put into effect, which becomes the main constriction in the optimization. The main data depends on the determination of the rotating direction by repetition of z route outcomes. The secondary data dependency depicts that the $(i+1)_{th}$ iteration can only be evaluated after the commencement of the i_{th} iteration. This is due to the fact that the micro-rotations are conducted on the intermediate vectors which are determined using previous iterations. Thus the dependence of these two data becomes hindrance in optimization of the critical path. Accordingly, the high precision requirement can only be fulfilled when the width of the input data and carry propagation delay decreases.

The aim of this paper is therefore to develop, analysis and evaluation of CORDIC development optimisation. In addition, CORDIC latency is reduced by using sign prediction scheme. Moreover, low latency CORDIC is proposed by combining sign prediction, Binary to Bipolar Representation method, Microrotation Angle Recoding and pipelined processing techniques.

The following document is arranged as follows: Analysis of traditional and sophisticated CORDIC algorithms is presented in the Section 2; last algorithm partly eliminates the fundamental CORDIC algorithm's iterative essence; Section 3 offers the primary input of this paper, a fully pipelined and parallelized optimized CORDIC algorithm. Section 4 offers a detailed evaluation of the hardware system suggested. Section 5 contains the conclusion.

II. REVIEW

A. Traditional CORDIC Algorithm

In rotation mode and circular motion, the sine/cosine function is calculated.

Matrix form of equation set (2-3) may be represented in as:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} \cos\alpha_i & -\sin\alpha_i \\ \sin\alpha_i & \cos\alpha_i \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (1)$$

After factoring out cosine term, we get:

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = K \begin{bmatrix} 1 & -\tan\alpha_i \\ \tan\alpha_i & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2)$$

The scaling factor after n iterations is given by $K = \prod_{i=0}^{n} K_i = \prod_{i=0}^{n} 1/\sqrt{(1+2^{-2i})}$

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \frac{1}{\sqrt{1+2^{-2i}}} \begin{bmatrix} 1 & -\sigma_i 2^{-i} \\ \sigma_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (4)$$

The fundamental iterative equations for this computation are as follows:

$$x_{i+1} = K_i(x_i - \sigma_i 2^{-i} y_i)$$
$$y_{i+1} = K_i(y_i + \sigma_i 2^{-i} x_i)$$
$$z_{i+1} = z_i - \sigma_i \alpha_i \quad (5)$$

where $\sigma_i = \begin{cases} \text{sign of } z_i \text{ , for rotation mode} \\ -\text{sign of } y_i \text{, for vectoring mode} \end{cases}$ and micro-rotation angle is $\alpha_i = \tan^{-1} 2^{-i}$.

$K_i = 1/\sqrt{(1+2^{-2i}}$ scaling factor

The z_0 is the complete angle of rotation, and the iterations bring z_i to null. After n iterations, the vector related to z_0 can be acquired. The findings of iteration are sine / cosine with $x_0 = K_n, y_0 = 0$, for z_0 in the range of $z_0 \in [0, \frac{\pi}{2}]$.

B. Partially Parallel CORDIC Algorithm

The primary limitation in the efficiency is the sequential calculation of σ_i shown by Eq. (5). Only after the respective step has chosen the right path of rotation can the activities be done in each step. When there is requirement at each stage of finding the direction of rotation which can be pre-determined and parallelized, the respective CORDIC rotations can also be carried out concurrently in the micro-rotation stage.

Different options for simultaneous application of σ_i by θ-path have been suggested in technical literature [12-16]. Para-CORDIC [17] is one the technique that is used to parallelizes the rotation direction generation *i.e.*, the rotation angle θ orientation of entry is discovered with regard to σ_i by binary to bipolar depiction (BBR), micro-rotation angle recoding (MAR) methods are also used in this field.

The angle θ of input is supposed to be within the range $|\theta| \leq \pi/4$ in N+1 bits in binary formats with two's complement and is given by $(-b_0) + \sum_{j=0}^{N} b_j 2^{-j}$, where $b_j \in \{0,1\}$. The higher part θ^H and the lower part θ^L of the input angle θ is given by:

$$\theta = \theta^H + \theta^L$$

$$= (-b_0) + \sum_{j=1}^{l-1} b_j 2^{-j} + \sum_{j=l}^{N} b_j 2^{-j} \quad (6)$$

In Eq. (6), l is the smallest index value such that $2^{-l} - \tan^{-1} 2^{-l} < 2^{-N}$. In [15] it was demonstrated that $l = [(N - \log_2 3)/3]$. Next, short BBR and MAR therapy is given, alternatively for the prediction of σ_i for θ^L and θ^H.

C. Binary to Bipolar Representation (BBR)

In the BBR method, rotation directions (σ_1 *to* σ_l) are evaluated for the first *l*-1 bits of the input angle (*i.e.*, θ^L). The bipolar representation $r_k \in \{-1,1\}$ is used to represent the binary value $b_j \in \{0,1\}$ and this BBR format can be represented as follows:

$$\theta^L = (-b_0) + \sum_{j=1}^{l-1} b_j 2^{-j}$$
$$= (-b_0) + \sum_{j=1}^{l-1} [2^{-j-1} + (2\,b_j - 1)\, 2^{-j-1}]$$
$$= \sum_{i=1}^{l} r_i 2^{-i} - 2^{-l},$$
where $r_1 = 1 - 2b_0$ (7)
$r_i = 2b_{i-1} - 1, i = 2,3, \ldots, l$ (3)

The starting values for the evaluating the rotation direction ($\sigma_1 to\ \sigma_l$) from equation (7) are directly established and the $r_1\ to\ r_l$ bipolar values are also evaluated. Overall expression for θ^L is:

$$\theta^L = \sum_{i=1}^{l} r_i 2^{-i} - 2^{-l}$$
$$= \sum_{i=1}^{l} \sigma_i \{\sum_{j=1}^{n(i)} \tan^{-1}(2^{-s_i^j}) + e_i\} - 2^{-l}$$
where $\sigma_i = r_i, i = 1,2, \ldots, l$ (8)

In the Eq. (8), 2^{-i} is expanded and represented as the combination of the error component and arctangent value [17].

D. Micro-rotation Angle Recoding (MAR)

When the term 2^{-i} is broken into the sum of the arctangent and error value, it gives the following expression:

$$2^{-i} = \tan^{-1} 2^{-i} + \sum_{j=2}^{n(i)} \tan^{-1}\left(2^{-s_i^j}\right)$$
$$= \sum_{j=1}^{n(i)} \tan^{-1}(2^{-s_i^j}) + e_i$$
$$s_i^1 = i, i = 1,2,..,l-1 \quad (9)$$

This equation is commonly referred to as MAR [17].

In order to recode the value 2^{-i} using MAR recoding, $n(i)$ is the number of micro-rotations required, and s_i^j is the shift sequences for $j = 1,2, \ldots \ldots .., n(i)$ with the first shift $s_i^1 = i, i = 1,2,..,l-1$. [14] Suggest the algorithm of MAR recoding in detail. In combination with Eq. (6) and Eq. (8-9), corrected rotational angle $\hat{\theta}^H$ is indicated by

$$\hat{\theta}^H = \theta^H + \sum_{i=1}^{l-1} \sigma_i e^i - 2^{-l} \quad (10)$$

The initial binary positional weightings are followed by BBR for θ^L and MAR, for binary depiction of the adjusted $\hat{\theta}^H$, another BBR is implemented as follows:

$$\hat{\theta}^H = (-\hat{b}_{l-1})2^{-l+1} + \sum_{k=l}^{N} \hat{b}_k\, 2^{-k}$$

$$= (-\hat{b}_{l-1}) + \sum_{k=l+1}^{N+1} [(2\hat{b}_{k-1} - 1)\, 2^{-k+2^{-l}} + 2^{-N-1}]$$

$$\sum_{i=l}^{N+1} \hat{r}_i 2^{-i} - 2^{-N-1},$$

where $\hat{r}_i = 1 - 2\hat{b}_{i-1}$, (11)

$$\hat{r}_i = 2\hat{b}_{i-1} - 1, i = l+1, \ldots \ldots, N+1$$

The last N-l+2 curve ($\hat{\sigma}_l\ to\ \hat{\sigma}_{N+1}$) direction is indicated by equation (11) and is straight discovered from the bipolar significance of $\hat{r}_l\ to\ \hat{r}_{N+1}$.

E. Para-CORDIC architecture

Fig. 1 shows the Para-CORDIC architecture for rotation. Para-CORDIC architecture consists of two BBR components. BBR_L and BBR_H are responsible for evaluating the computations given in the Eq. (7) and Eq. (11). Thus providing the proper direction of rotation related to σ_1 to σ_l for phase 1 and $\hat{\sigma}_l$ to $\hat{\sigma}_{N+1}$ in phase 2. $Add_{prediction}$ block serves the operations related to Eq. (10) and the memory block is used to store the error e_i which are already evaluated.

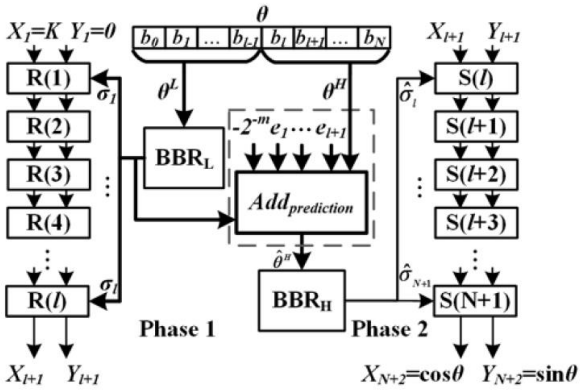

Figure 1 Para CORDIC Architecture [17]

III. PROPOSED PIPELINED CORDIC

Table I demonstrates that the most arctangents terms are required for the first two iterations (i.e.,2^{-1} and 2^{-2}). We aim therefore to suggest a technique that in the first two iterations can decrease the amount of recoded arctangent terms. Table I indicates the decomposition for one part of the input angle of the arctangent conditions. Since the respective polarity of the direction is known by the BBR method, we can decompose arctangent terms more effectively by taking two bits of the input angle into account simultaneously, rather than one at a time. Compared to the initial recoding techniques for para-CORDIC rotator applications, our suggested technique may attain low latency.

Table I: MAR values for B=24

Binary weight	$n(i)$	$s_i^j, j = 1, \dots n(i)$	$\lvert e^i \rvert$
2^{-1}	4	1,5,8	2^{-9}
2^{-2}	2	2,8	2^{-9}
2^{-i}, i=3,…7	1	1	2^{-10}

In our technique, two binary weights can be decomposed into their arctangent configurations and then a minimum respective amount of micro-rotations can be achieved for each coding. We will then decide to recode for all possible recoding results with the least number of micro-rotations. Table II demonstrates in the two initial iterations the outcomes of our suggested 24 bit para-CORDIC rotation techniques.

Table II shows the maximal number of terms of the four possible combinations for the $n(i)$ arctangent conditions decomposed

Table II: Results of our proposed work for B=24

Combined binary weights	$n(i)$	$s_i^j, j = 1, \dots n(i)$
$2^{-1} + 2^{-2}$	2	1,2
$-2^{-1} - 2^{-2}$		1,8*
$2^{-1} - 2^{-2}$		2,8
$-2^{-1} + 2^{-2}$		2*,8*
*: negative arctangent terms		

by every combined binary weight. Once the input angle is prepared there will be only one of the four feasible values. Table II also sums up the error terms to meet the value of $\lceil \hat{\theta}^H \rceil$ [$\hat{\theta}^H$ is shown in Fig. 1(a)] meet inequality $|\hat{\theta}^H| < 2^{-l}$ which is the convergence criteria in the rotation of CORDIC [17]. Table II shows that in the suggested technique, there are 2 phases of micro-rotation, which are less than the MAR technique earlier mentioned.

Figure 2 is the architecture of microrotation required for our method in the first two iterations. The cascaded phases of the first two iterations are combined into one phase (i.e. R'(1)) rather than two different phases. The first and second digits are used and displayed in the total weights of Table II, e.g., $\sigma_u = 1$ and $\sigma_v = 2$.

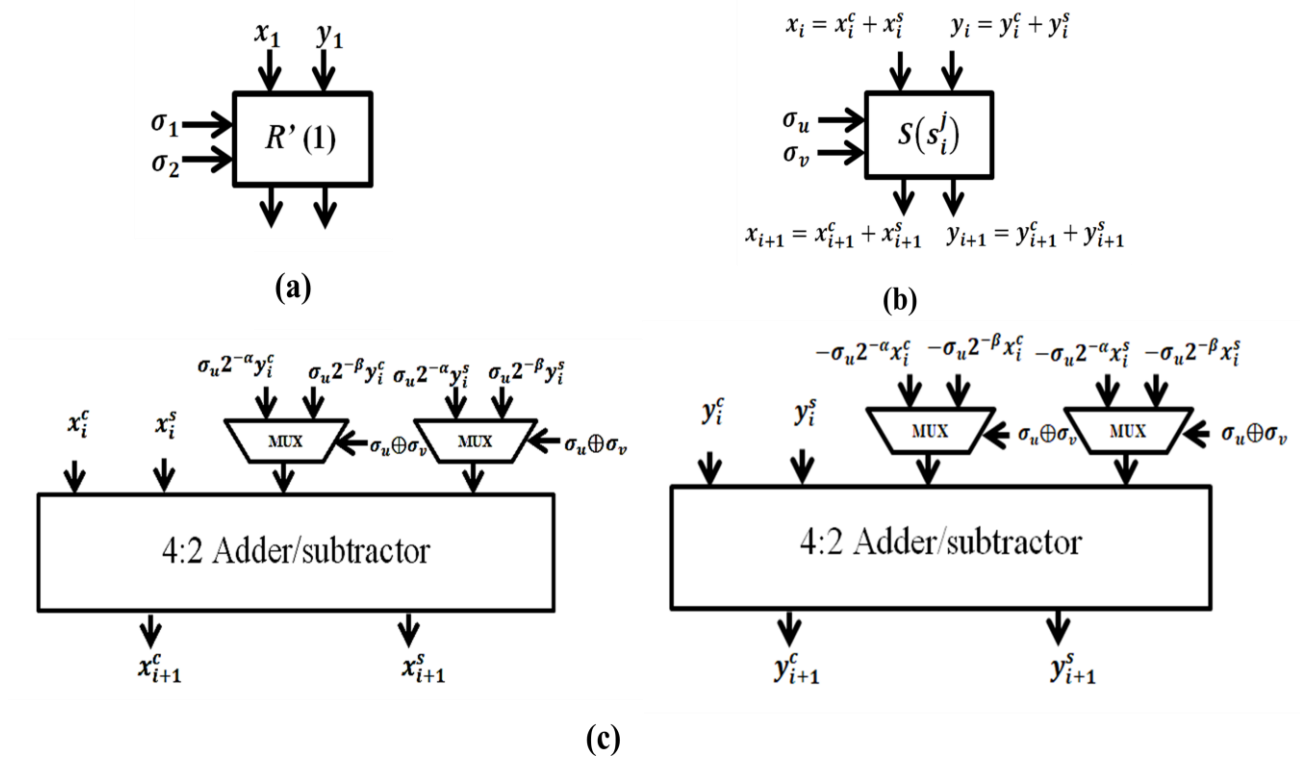

Figure 2 (a) Architecture of first two combined iteration (b) Architecture of micro-rotation stage s_i^j (c) Implementation using 4:2 adder/subtractor

The respective s_i^j values for each sum total of binary weight of the first and second rows is given by the corresponding values of α and β as shown in Figure 2. For each micro-rotation stage, a 4:2 adder/subtractor is used to provide a result for each of the two outputs at any stage. Furthermore, the polarity of σ_u can regulate the procedure of addition / subtraction and $\sigma_u \oplus \sigma_v$ (excluded factor of the two polarities) defines the right conditions of shifting. The architecture remained identical to the previous para-CORDIC rotation for the remainder of the micro-rotations.
Because of the application of the BBR technique, our proposed method could attain the convergence properties as in [17]. In addition, the number of microrotations is fixed, and hence the scaling factor is constant and can be computed in advance (similar to [17]). In order to increase the efficiency of the para-cordic rotation in every bit width, we can use our suggested technique of low latency angles recalling together with pipeline handling.

IV. EXPERIMENTAL RESULTS

VHDL modelling was provided and analysed using Xilinx ISE for the suggested CORDIC framework. In comparison with the earlier suggested CORDIC architectures, equipment usage, speed and latency of a suggested work is contrasted. Table III shows the use of resources (Lookup tables (LUTs), Registers), the frequency and latency which are utilized in Optimized CORDIC core and other designs.

Table III: Comparison table

Algorithm	[8]	[9]	[10]	Proposed
Method	Conventional	Conventional	ARC	MAR
Input bit width	20	24	24	24
Area (LUT+Registers)	2200	2891	103000	990
Max Frequency (MHz)	35	462	442.4	709.572
Delay (ns)	28.57	2.16	2.26	1.409

Comparison of the experimental results of the suggested Optimized CORDIC with some previous work is performed and shown in Table III. The [8-10] model used standard CORDIC algorithms, while the [11] model used the optimised CORDIC algorithm for recoding angles. Our model proposed the use of BBR and MAR to optimize CORDIC algorithms. The working frequencies of previous models are higher when we use the pipeline and fixed point data formats, but after utilising the parallel BBR approach in the CORDIC design, the frequency of the design increases and the chip area of our design also reduces than the other proposed works. Our layout delivers a stronger performance in precision than other designs.

V. CONCLUSION

This paper has proposed a method for Optimization of CORDIC design by implementing slight modifications in the Para-CORDIC architecture. This leads to the fully optimised parallel and pipelined CORDIC scheme. The parallel architecture, pipelined processing collectively with BBR and MAR for recoding the direction of rotation and angle recoding technique are applied on this layout to achieve the low area and occasional latency. This leads to 34% delay reduction compared to previously reported 24 bit CORDIC architectures.

REFERENCES

1. J. Zhou, Y. Dou, Y. Lei, J. Xu, and Y. Dong, "Double precision hybrid-mode floating-point FPGA CORDIC co-processor," in Proc. 10th IEEE Int. Conf. High Performance Computing Communications (HPCC), Aug. 2008, pp. 182–189.
2. M. Chakraborty, A. S. Dhar, and M. H. Lee, "A trigonometric formulation of the LMS algorithm for realization on pipelined CORDIC," IEEE Transactions Circuits and Systems II, Express Briefs, vol. 52, no. 9, pp. 530–534, Sep. 2005.
3. L. Cordesses, "Direct digital synthesis: A tool for periodic wave generation (part 1)," IEEE Signal Processing Magazine, vol. 21, no. 4, pp. 50–54, Jul. 2004.
4. Y. Wang and S. Butner, "A new architecture for robot control," in Proceedings IEEE International Conference Robotics and Automation, vol. 4. Mar. 1987, pp. 664–670.
5. T. Lang and E. Antelo, "High-throughput CORDIC-based geometry operations for 3D computer graphics," IEEE Transactions on Computers, vol. 54, no. 3, pp. 347–361, Mar. 2005.
6. S. Wang, V. Piuri, and E. E. Swartzlander, "Hybrid CORDIC algorithms," IEEE Transactions on Computers, vol. 46, no. 11, pp. 1202–1207, Nov. 1997.
7. D. S. Phatak, "Double step branching CORDIC: A new algorithm for fast sine and cosine generation," IEEE Transactions on Computing, vol. 47, no. 5, pp. 587–602, May 1998.
8. Z. Qi, A. C. Cabe, R. T. Jones, Jr., and M. R. Stan, "CORDIC implementation with parameterizable ASIC/SoC flow," in Proceedings IEEE SoutheastCon, Concord, NC, USA, 2010, pp. 13–16.
9. V. Torres, J. Valls and M.J. Canet, "Optimised CORDIC-based atan2 computation for FPGA implementations", Electronics Letters, 14th September 2017, Volume 53, No. 19, pp. 1296–1298.
10. J. Zhang, H. Liu, W. Hu, D. Liu, and B. Zhang, "Adaptive recoding CORDIC," IEICE Electron. Exp., vol. 9, no. 8 pp. 765–771, 2012.
11. Hong-Thu NGUYEN, Xuan-Thuan NGUYEN, "A Low-Latency Parallel Pipeline CORDIC", Institute of Electronics, Information and Communication Engineers Transactions Electronics, VOL.E100–C, NO.4 April 2017.
12. D. Timmermann, H. Hahn, and B. J. Hosticka, "Low latency time CORDIC algorithms," IEEE Transactions on Computers, vol. 41, pp. 1010-1015, 1992.
13. T. Srikanthan and B. Gisuthan, "A novel technique for eliminating iterative based computation of polarity of micro-rotations in CORDIC based sine–cosine generators," Microprocessors and Microsystems, vol. 26, pp. 243-252, 2002.
14. B. Gisuthan and T. Srikanthan, "Pipelining flat CORDIC based trigonometric function generators," Microelectronics Journal, vol. 33, pp. 77-89, 2002.
15. S. Wang, V. Piuri, and E. E. Swartzlander, "Hybrid CORDIC algorithms," IEEE Transactions on Computers, vol. 46, no. 11, pp. 1202–1207, November 1997.
16. B. Lakshmi and A. S. Dhar, "CORDIC Architectures: A Survey," VLSI Design, 2010.
17. T.-B. Juang, S.-F. Hsiao, and M.-Y. Tsai, "Para-CORDIC: parallel CORDIC rotation algorithm," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 51, pp. 1515-1524, 2004.