



LH-CORDIC: Low Power FPGA Based Implementation of CORDIC Architecture

Sharath Chandra Inguva^{1*} Joseph Beatrice Seventline²

¹*Department of Electronics and Communication Engineering,
Guru Nanak Institute of Technology, Hyderabad, India*

²*GITAM Institute of Technology, Visakhapatnam, India*

*Corresponding Author Email: csharath0510@gmail.com

Abstract: The coordinate rotation digital computer (CORDIC) is a class of shift-add algorithms for rotating vectors in plane. Several techniques use the trigonometry function to compute the digital waves, but that requires expensive memory usage. Due to the flexible characteristics, CORDIC is best alternative and allows high quantization accuracy by maximum word length. The linear-rate convergence creates the major problem in CORDIC algorithm with the source of word-length and iteration speed. The power consumption also a major issue here to affects the performance by array of shift-add operations. For further enhancement, in this paper, we propose a low power and high speed CORDIC (LH-CORDIC) design with an improved power control and hardware reduction techniques. We employ the canonical signed-digit (CSD) technique and Hcub algorithm for reducing the number of shifters and adder/subtractor in the design. Then, we propose an adder based on the advanced Boolean logic technique. These three techniques are used to redesign the entire CORDIC logic stages thereby contributing in power consumption reduction. The functionality of proposed LH-CORDIC algorithm is assessed through FPGA implementations. The simulation result shows that the proposed method has higher frequency of 78.91%, 83.42%, 79.89% and 77.01% when compared with conventional CORDIC method.

Keywords: Coordinate rotation digital computer, Digital waves, LH-CORDIC, Canonical signed-digit, Hcub algorithm, Conventional CORDIC.

1. Introduction

CORDIC has established its popularity in several important areas of application, like generation of sine and cosine functions, calculation of discrete sinusoidal transforms like fast Fourier transform (FFT) [1], discrete sine/cosine transforms (DST/DCT) [2], householder transform (HT) [3], etc. CORDIC algorithm for the calculation of trigonometric functions was derived in the year 1959 by Jack E. Volder [4], from the general equations for vector rotation. Later Walther generalized the equations to solve a wider range of equations, including the hyperbolic equations, multiplication, division and conversion between binary and mixed radix number systems of DSP applications. CORDIC algorithm is commonly used

in those applications where area is a primary constraint. The processing elements performing vector rotations can efficiently implement all the elementary functions without using any multiplier. The operations like addition, subtraction, bit-shift and lookup table are involved in this algorithm. It is well suited for VLSI implementations due to its simplicity.

Many variations have been suggested for efficient implementation of CORDIC with less number of iterations over the conventional CORDIC algorithm. The number of CORDIC iterations is optimized by greedy search at the cost of additional area and time for the implementation of variable scale-factor [5]. The scale-factor compensation technique adversely affects the latency/throughput of computation [6]. Two area and time efficient CORDIC architectures have been suggested for

involve constant scale-factor multiplication for adequate range of convergence (RoC) [7]. In real-time digital control systems such as intelligent robot control system, it is essential to perform a large amount of computation at high speed because the robot must respond quickly to the environmental movement. The robotic system's applications require a real-time operation to interface speed constraints is one of the major trends in current robotic research. It is essential to perform a large amount of computation at high speed because the robot must respond quickly to the environmental movement [8]. Most of the robotic system's applications require a real-time operation to interface speed constraints. There are also scenarios where area and power efficient solutions are valuable.

CORDIC algorithm is an iterative algorithm, which can be used for computation of trigonometric functions, multiplication and division [9]. The CORDIC is very simple and iterative convergence algorithm that reduces complex multiplication, greatly simplifying overall hardware complexity. This serves as an attractive option to system designers as they continue to face the challenges of balancing aggressive cost and power targets with increased performance required in next generation signal processing solutions. The basic principles underlying the CORDIC based computation and present its iterative algorithm for different operating modes and planar coordinate system [10]. This algorithm attracts more and more attention in elementary function evaluation and signal processing applications. A backward angle recoding (BAR) method [11] used to eliminate redundant CORDIC elementary rotations and hence expedite the CORDIC rotation computation. The linear, circular, and hyperbolic CORDIC rotations, the use of BAR guarantees more than 50% reduction of elementary CORDIC rotations provided the scaling factor needs not be kept constant. An on-line CORDIC algorithm with a constant scale factor and latency independent design has been derived through the extension of derivative simplified metrics [12]. Angle quantization (AQ) [13] is used as a design index for vector rotational operation, where the rotational angle with a unified design framework for cost-effective low-latency rotational algorithms and architectures. A special rotational CORDIC processor [14] operated in the circular coordinate system with an unlimited angular convergence range. The algorithm adaptively selects the appropriate iteration steps and thus converges to the target angle executing a minimum number of iterations. A mixed-scaling-rotation CORDIC

(MSR-CORDIC) system [15] eliminates the overhead of the scaling operations that are inevitable of CORDIC algorithms and significantly reduce the total iteration number used to improve the speed performance.

In this paper, a low power and high speed CORDIC (LH-CORDIC) design is proposed with an improved power control and hardware reduction techniques. The main objective of proposed LH-CORDIC is used to achieve a high speed or low latency VLSI architecture for CORDIC algorithm. In the design, for reducing the number of shifters and adder/subtractor we employ the CSD technique and the Hcub algorithm. Also, based on the advanced Boolean logic technique an adder is proposed. The remainder of this paper is organized as follows. Section 2 describes the survey of recent works related to our contributions. Section 3 introduces the problem methodology and system model of proposed LH-CORDIC algorithm. Section 4 illustrates the detailed working function of proposed LH-CORDIC algorithm with proper mathematical model. The results and performance analysis are discussed in Section 5. Finally, the paper concludes in Section 6.

2. Related works: A brief review

Banerjee et al. [16] have presented a pipelined architecture using CORDIC for realization of transform domain equalizer. The running DFT was employed as the transform and CORDIC was used for realization of running DFT. Pipelining was applied throughout the architecture, thus limiting the critical path delay to the propagation delay of a single 16 bit adder for 16 bit arithmetic. Madheswaran et al. [17] have presented an improved direct digital synthesizer (DDS) using the hybrid wave pipelining (HWP) technique and CORDIC algorithm for software defined radio (SDR). The HWP can be used to speed up the circuits without insertion of storage elements. The CORDIC algorithm used for phase-to-amplitude conversion using dynamic transformation rather than read only memory (ROM) static addressing.

Huang et al. [18] have introduced a CORDIC based fast radix-2 algorithm for computation of DCT. The introduced algorithm has some distinguish advantages, such as Cooley-Tukey FFT like regular data flow, uniform post-scaling factor, in-place computation and arithmetic-sequence rotation angles. Lakshmi et al. [19] have addressed an area and computation delay in rotational CORDIC. Reduction in area and computation delay was achieved by halving the number of iterations

and pre-computing all the direction of rotations. The latency and area of the presented architecture was computed in terms of full adder delay and full adder area, so that, these architectures can be implemented using any technology by the selection of appropriate logic style for full adder.

Huang et al. [20] have presented CORDIC based fast algorithm for power-of-two point DCT, and develop its corresponding efficient VLSI implementation. Zhang et al. [21] have introduced a hybrid CORDIC algorithm based on phase rotation estimation applied in numerical controlled oscillator (NCO). Through estimating the direction of part phase rotation, the algorithm reduces part phase rotation and add-subtract unit, so that it decreases delay. The results indicated that the improvement over traditional CORDIC algorithm achieved in terms of ease of computation, resource utilization, and computing speed/delay while maintaining the precision.

Moroz et al. [22] have proposed the theoretical bases and practical pipelined FPGA implementation of a hybrid scaling-free CORDIC algorithm. Logical combination of three construction elements of modern FPGAs which were LUT, simple scaling-free CORDIC stages, and multipliers allowed a considerable improvement of calculation efficiency of sine and cosine functions without the loss of accuracy. The implementation is performed in Altera Stratix3 FPGA (EP3SL340F1517C2) using Quartus II version 9.0 shows that the hybrid FPGA architecture significantly reduces latency (42% reduction) with a small area overhead, compared to the conventional version. Bhairannawa et al. [23] proposed a fingerprint-based biometric system using optimized 5/3 DWT architecture and modified CORDIC-based FFT.

Lin et al. [24] have introduced an accelerometer-based sensing system for accurate head position monitoring was developed and realized. CORDIC based tilting sensing algorithm was realized in the system to quickly and accurately convert accelerometer raw data into the desired head position tilting angles. An efficient error detection schemes for Cascaded Single-rotation CORDIC was introduced by Ramadoss et al. [29] that negligibly hamper the architectures performance. To detect both permanent and transient faults, the authors introduced signature-based schemes for this CORDIC variant. Additionally, to other variants of CORDIC the authors present such applied schemes. Through error simulations and FPGA implementations the presented technique effectiveness was assigned.

Yang et al. [25] have proposed a phase frequency detector (PFD) based CORDIC algorithm for a biaxial resonant micro accelerometer. A conventional digital closed-loop self oscillation system based on the CORDIC algorithm was implemented and simulated using Simulink software to verify the system performance. Therefore, digital, closed-loop self-oscillation using the PFD-based CORDIC algorithm was designed to further optimize the system performance. The system experimental results show the PFD-based CORDIC improves the bias stability of the resonant micro accelerometer by more than 5.320 times compared to the conventional system.

3. Problem methodology and system model

3.1 Problem methodology

Garrido et al. [26] presented the CORDIC II algorithm that substitutes the CORDIC micro-rotation by a new angle set. Their new approach involves three new types of rotators: friend angles, USR CORDIC and nano-rotations. By using their proposed micro-rotations, they confirmed that the CORDIC II requires the minimum number of adders among CORDIC algorithms so far. Even though, more merits are there in their new design we have observed that it can be enhanced in terms of architecture by using advanced techniques. The major areas we have to concentrate in order to make the whole design power efficient are architecture of the friend angles (5 Adder, 7 MUX and 9 Shifter) and Nano-rotator (4 Adder, 8 Mux and 5 Shifter).

Usually, CORDIC algorithms are achieved best in software implementations. A rule of thumb in scalar software implementations is that if a hardware multiplier is available, it should be used. Through series expansions, the trigonometric functions can be conveniently and efficiently computed. However, in many cases the CORDIC algorithms is affected by hardware cost and power consumption problems. Moreover, many practical CORDIC implementations are based on bit serial binary arithmetic's, because bit parallel operations and iterative implementations require very less hardware cost. However, it is affected by a power consumption problem.

From the literature [16-26], much hardware has been presented for CORDIC algorithms. However, the existing techniques exhibits several limitations like hardware cost and power consumption issues. To overcome the above drawbacks, in this paper a low power and high speed CORDIC (LH-CORDIC) design is proposed for an improved power control

and hardware reduction techniques. The main objective of proposed LH-CORDIC as follows:

- The canonical signed-digit (CSD) technique and Hcub algorithm are used to reduce the number of shifts in LH-CORDIC design.
- The adders are designed by the advanced Boolean logic (ABL) technique, merges the two binary adders into one, which allows component sharing, particularly in the preprocessing and the sum-computation stages.
- These three techniques are used to redesign the entire CORDIC logic stages in [26] thereby contributing in power consumption reduction and throughput enhancement.
- The proposed LH-CORDIC design is implemented in Xilinx with four FPGA families and their performance is compared with the existing CORDIC design [28].

3.2 System model of LH-CORDIC design

CORDIC arithmetic could speed up the iterations and have a higher precision than before. It is an iterative algorithm for the calculation of the rotation of a two-dimensional vector in linear, circular and hyperbolic coordinate systems. Each system has two ways to be done which are the rotation mode and the vectoring mode. Consider a vector $A (x_i, y_i)$ is rotating via a sequence of elementary angles whose algebraic sum approximates the desired rotation angle θ to get another vector $B (x_j, y_j)$ and the relationship represents as,

$$\begin{aligned} x_j &= \mathcal{R} \cos(\theta + \beta) = x_i \cos \theta - y_i \sin \theta \\ y_j &= \mathcal{R} \sin(\theta + \beta) = x_i \sin \theta - y_i \cos \theta \end{aligned} \quad (1)$$

Where, \mathcal{R} is radius of circle and the corresponding matrix format as follows;

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (2)$$

The simplified format as,

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \cos \theta \begin{bmatrix} 1 & -\tan \theta \\ \tan \theta & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (3)$$

Then hypothesize to use an iterative method; the desired rotation angle θ can be got after several rotations.

$$\begin{bmatrix} x_{nj+1} \\ y_{nj+1} \end{bmatrix} = \cos \theta_{nj} \begin{bmatrix} 1 & -\tan \theta_{nj} \\ \tan \theta_{nj} & 1 \end{bmatrix} \begin{bmatrix} x_{ni} \\ y_{ni} \end{bmatrix} \quad (4)$$

Where $\theta_{nj} = \arctan 2^{-nj}$ and the hypothesizes that to provides the relationship between θ_{nj} and θ represents,

$$\sum_{nj=0}^{\infty} \chi_{nj} \cdot \theta_{nj} = \theta \quad (5)$$

Where χ_{nj} represents sign function. Then rotated angle denotes as,

$$R_n = \theta - \sum_{a=0}^{nj=1} \chi_a \theta_a \quad (6)$$

If χ_{nj} is 1, and $R_n \leq 0$ means rotator perform in correct sequence, otherwise rotate in inverted sequence. After N time rotation, it gets the general notation as follows;

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} = \prod_{nj=0}^N \cos \theta_{nj} \begin{bmatrix} 1 & -\chi_{nj} 2^{-1} \\ \chi_{nj} 2^{-1} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \quad (7)$$

CORDIC rotator rotates the input vector through whatever angle is necessary to align the result vector with the x-axis. The result of the vectoring operation is a rotation angle and the scaled magnitude of the original vector. The vectoring function works by seeking to minimize the y component of the residual vector at each rotation. The sign of the residual y component is used to determine which direction to rotate next. If the angle accumulator is initialized with zero, it will contain the traversed angle at the end of the iterations.

The CDS derives the quantization process on the rotational angle, θ , directly and decompose the original rotational angle into several sub-angles, θ_s . Then, try to sum up those sub-angles to approximate the original angle as close as possible; and try to minimize the angle quantization error.

$$\varepsilon_e = \theta - \sum_{i=0}^{N_A-1} \theta_i \quad (8)$$

Where N_A denotes the number of sub-angles. Each rotation module is dedicated to performing a particular rotation of sub-angle and the rotation can be accomplished by cascading these rotation modules. The quantization process is described in Fig. 1.

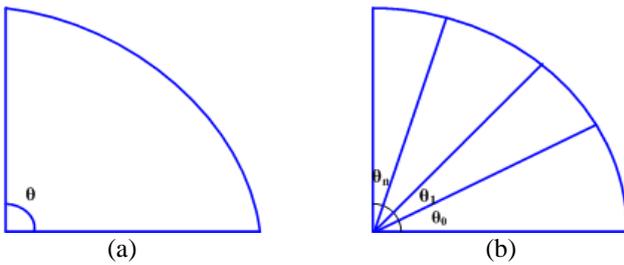


Figure.1 Quantization process: (a) angle quantization and (b) with error notation



Figure.2 System model of CORDIC II algorithm

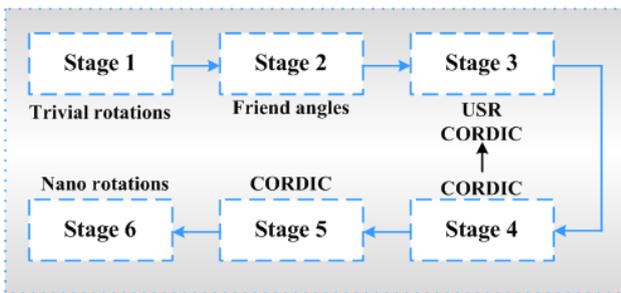


Figure.3 Stages in CORDIC II algorithm

CORDIC II algorithm consists of several rotation stages connected in series. Each rotation stage can be characterized by an input range $[-\alpha_{in}, \alpha_{in}]$ and an output range $[-\alpha_o, \alpha_o]$. In general, a rotation stage may include any number of rotation angles. Each input is rotated by one of these angles and N-rotator as a rotator with N different angles to choose from. The system model of CORDIC II algorithm is shown in Fig. 2 and the stages of CORDIC II algorithm is shown in Fig. 3.

The proposed CSD based shifter, ABL based adder is utilized in CORDIC II algorithm instead of conventional shift/addition, and it consists of six rotation stages in pipeline that use the angle sets describes in [26].

4. Low power and high speed CORDIC (LH-CORDIC)

In this section, first we describe the CSD, Hcub and ABL techniques for logic reduction in complex architecture. Second, the logic reduced design is applied to the CORDIC II algorithm to make design as hardware efficient and low power.

4.1 Hcub algorithm

Hcub algorithm is a multiple constant multiplier (MCM) algorithm. It is used to generate multiplier block from a set of constants. This method is used to reduce the addition, subtraction and shifting operations. The Hcub algorithm has the advantage that it is not limited by the constant bit widths [27].

In our research, the Hcub algorithm reduces the shift operation required for the multiplication operation. Initially, the benefit function $(b(r, S, T))$ is defined to build the Hcub (cumulative benefit), to quantify what extent adding a successor to the ready set enhances the distance to fixed, which is computed as follows.

$$b(r, S, T) = dist(r, T) - dist(r + S, T) \quad (9)$$

Where, the successor is S , the ready set is denoted as r and the target is denoted as T . To enable joint optimization of all targets, about the benefit function the key observation is the different target (T) benefits summed. The Hcub is defined as follows.

$$Hcub(r, s, t) = \arg \max_{s \in S} (\sum_{T \in t} b'(r, S, T)) \quad (10)$$

With respect to all targets in t instead of taking the maximum, the cumulative benefit heuristic adds the weight benefits. Thereby reducing the addition, subtraction and shifting operations.

4.2 Canonical signed-digit (CSD) based shifter modules

In general, the shifters defines the critical path can be calculated as $\lceil \log_2 n \rceil$, where n is the number of non-zeros present within the coefficients. The idea behind the reconfigurable constant multiplier is to consider the condition while the maximum possible numbers of non-zeros occur for a coefficient. Hence, it can be concluded that reduction in the non-zeros ensures reduction in the number of adders as well as the number of addition operations in a chain. Canonical signed digit (CSD) number representation of the coefficients reduces the non-zeros by great extent than that of the binary number representation. More non-zeros in each of the coefficient requires increasing number of shifters for summing up the partial products and more numbers of shifters in a chain for the same. In this work, the CSD representation of the coefficient instead of binary representation used, which reduces the number of non-zeros by more than 50%. The optimization of non-zeros in the coefficients by converting coefficient represented in the signed

binary form to the CSD representation on the hardware itself. The general steps for conversion of binary to CSD are given as follows:

1. Check consecutive number of 1's in the binary sequence.
2. Replace the '0' before the first '1' in the sequence with '+' or '1'.
3. Replace the last '1' in the sequence with '-'-.

If binary representation of a number is used for multiplications then during partial product production each '1' in the multiplier corresponds to a shift and add operation of the multiplicand and '0' represents a shift operation only. But in case of CSD representation the number of non-zero elements in the sequence is less hence the need for shifter can be reduced at the cost of extra subtractions.

4.3 Adder using advanced Boolean logic (ABL) technique

The advanced Boolean logic (ABL) technique structure merges two binary adder structures and maximizes sharing of components. This merger permits shorter cell-interconnects, which reduce unwanted hardware/cell usage. Generally, use two different n-bit binary adders; one to compute $A + B$ and one to compute $A + B - m$. Input, output, and internal sub-module interconnections were routed in two distant groups, where each group dealt with one adder. The ABL based adder can be defined as,

$$S = (M + N)_{i_2} \tag{11}$$

where $i_2 = (2^n + j)$; $3 \leq J \leq 2^{n-1}$; $M, N \in [0, i_2 - 1]$ Eqn. (9) rewritten as follows:

$$S = \begin{cases} M + N; & \text{if } M + N < i_2 \\ M + N - (2^n + j); & \text{if } M + N \geq i_2 \end{cases} \tag{12}$$

Where $S < i_2 < 2^{n+1}$

Also observe that $-(2^n + J)2^{n+1} = (2^{n+1} - (2^n + J))2^{n+1} = (2^n - J)2^{n+1}$ to get results as,

$$S = \begin{cases} (M + N)_{2^{n+1}}; & \text{if } C_0 = 0 \\ (M + N - \hat{J})_{2^{n+1}}; & \text{if } C_0 = 1 \end{cases} \tag{13}$$

Where $\hat{J} = 2^n - J$ and C_0 is the output carry. Since M and N are integers that belong to the range $[0, i_2 = 2^n - J]$, the binary representations of M , N and \hat{J} are expressed in $(q + 1)$ bits each.

$$\begin{aligned} M' &\rightarrow m'_q m'_{q-1} \dots m'_1 m'_0 + N' \\ &\rightarrow n'_q n'_{q-1} \dots n'_1 n'_0 \end{aligned} \tag{14}$$

Table 1 Trivial rotation stage

Angle	Coefficients	S1	S2	S3	S4	Output
0^0	1	1	1	1	1	x, y
90^0	j	1	0	0	1	$-y, x$
180^0	-1	0	1	1	1	$-x, -y$
270^0	$-j$	0	0	1	1	$y, -x$

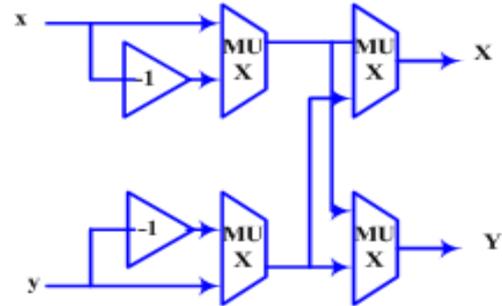


Figure. 4 Trivial rotation module

$$m'_i = \begin{cases} m_i \oplus n_i; & \text{if } \hat{j} = 0 \\ m_i \otimes n_i; & \text{if } \hat{j} = 1 \end{cases} \tag{15}$$

$$n'_i = \begin{cases} m_i \wedge n_i; & \text{if } \hat{j} = 0 \\ m_i \vee n_i; & \text{if } \hat{j} = 1 \end{cases} \tag{16}$$

$$h_i = m_i + n_i \text{ and } h'_i = m'_i + n'_i \tag{17}$$

4.4 CORDIC II algorithm

4.4.1. Stage 1: trivial rotation

The first stage of CORDIC II algorithm uses trivial rotations. It computes the $\pm 180^0$ and $\pm 90^0$ of trivial rotations by arrange the input and output notation by the range of $\pm 45^0$. Table 1 shows the details about the trivial rotation stage. S1, S2, S3 and S4 are the selector input for the multiplexers. The coefficient for the trivial rotation stage is given in Table 1.

Fig. 4 shows the trivial rotator hardware structure and it consists of two negators (adders) and 2:1 multiplexers.

4.4.2. Stage 2: friend angles

We use Canonical Signed Digit (CSD) based shifter and Hcub algorithm for the friend angle stage. The number of adders required for multiplication is reduced by the CSD concept. The Hcub algorithm reduces the shift operation required for the multiplication operation. Here we use an ABL based adder for merging two binary adders. The friend angle hardware structure for CORDIC II architecture is shown in Fig. 5.

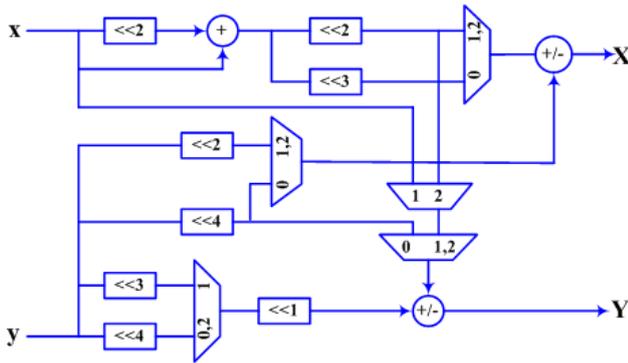


Figure. 5 Friend angle modules

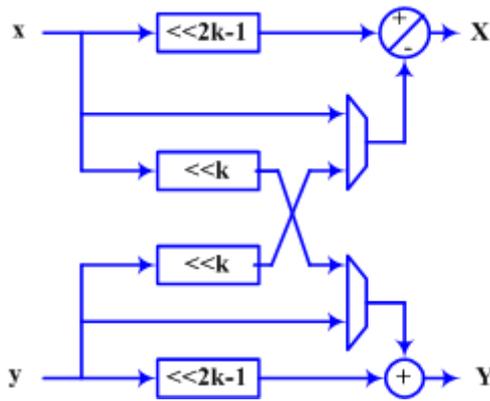


Figure. 6 USR CORDIC module

The second stage provides the friend angles with the normalized scale factor of constant term as follows

$$R_{norm} = \frac{R}{2^{\lfloor \log_2 R \rfloor}} \quad (18)$$

The friend angle is derived by kernel of $[25, 24 + j7, 20 + j15]$ with the scale factor coefficient is written as $25^2 = 24^2 + 7^2 = 20^2 + 15^2$. The friend angles that correspond to the coefficients are $0^\circ, 16.260^\circ, 36.870^\circ$, with normalized scaling $R_{norm} = 1.563$. By using the CSD, Hcub algorithm and ABL based adder our proposed friend angle module have 1 shifter, 2 multiplexer and 2 adder/subtractor less than the conventional CORDIC II architecture.

4.4.3. Stage 3: USR CORDIC

The third stage is USR CORDIC that use the kernel as $[129, 128 + j16]$ to reduce the remaining angle to $\pm 3.563^\circ$.

Fig. 6 shows the USR CORDIC stage hardware structure and it consists of two adders and two 2:1 multiplexers.

4.4.4. Stage 4, 5: CORDIC

The fourth and fifth stages of the CORDIC II use conventional CORDIC rotations by 1.790° and 0.895° .

4.4.5. Stage 6: nano- rotations

The sixth stage is nano-rotations use the kernel as $P_J = 512 + jJ; J = 0, 1, \dots, 8$. The alternative rotation angles represent $\alpha_J = J \cdot 0.112^\circ$ and remaining angle is $\pm 0.056^\circ$.

Fig. 7 shows the nano-rotation stage hardware structure.

5. Experimental results and discussions

In this section, for four families of Xilinx FPGAs we implement the proposed design and discuss the results of the overhead assessment. For the original CORDIC designs the analysis is performed and using Xilinx ISE 14.5 for Virtex-4 (XC4VSX35-10FF668), Artix-7 (XC7A100T-2CSG324), Virtex-5 (XC5VLX20T-2FF323) and Spartan-3A-DSP (XC3SD1800A-4FG676) the structures of proposed error detection is designed. As shown in Figs. 8 to 10 the overheads are benchmarked. In seven rotation modules the designs are divided and in each rotation module there has two ordinary or self checking Subtractor modules /Adder modules.

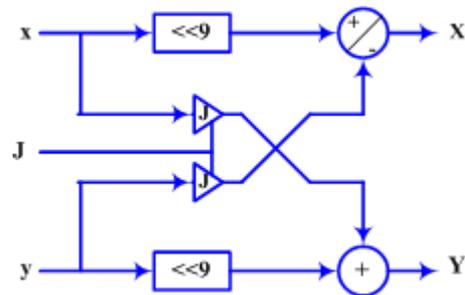


Figure. 7 Nano-rotation module

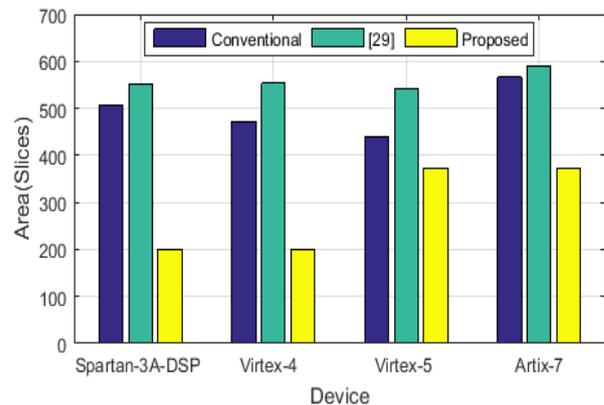


Figure.8 Analysis of area with four FPGA families

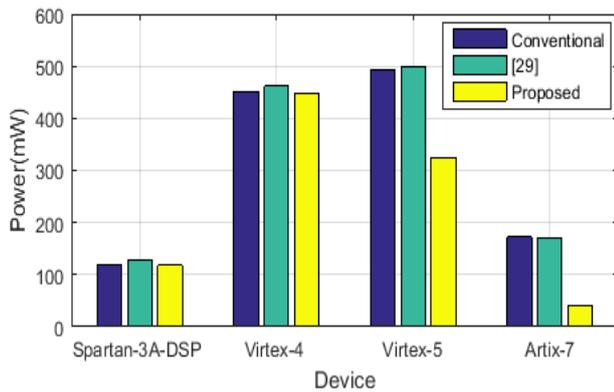


Figure. 9 Analysis of power consumption with four FPGA families

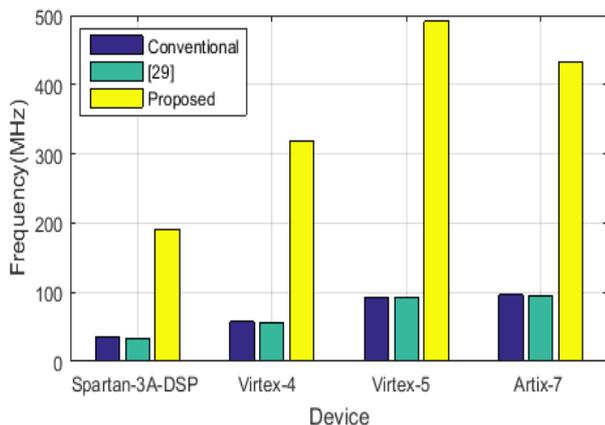


Figure.10 Analysis of frequency with four FPGA families

The analysis of area with four FPGA families is shown in Fig. 8. From the figure it is observed that, in Spartan-3A-DSP the area (slices) of the proposed method is lower by 60% and 63.64%, in Virtex-4 the area (slices) of the proposed method is decreased by 58.53% and 63.63%, Virtex-5 the area (slices) of the proposed method is decreased by 15.97% and 29.53%, and in Artix-7 the area (slices) of the proposed method is decreased by 32.93% and 35.82% when compared with conventional CORDIC algorithm and R. Ramadoss et al. [29] methods. However, the existing methods covers only single bit transient errors. But, the proposed MCM algorithm is used to generate multiplier block from a set of constants.

Fig. 9 shows the analysis of power consumption with four FPGA families. It is clearly noticed from the Fig. 9; the power consumption of the proposed method in Spartan-3A-DSP is lower by 3.1% for conventional CORDIC method and 14.04% for R.Ramadoss et al. [29] method. Because, the proposed method has lower friend angles when compared with other existing research works. Similarly in Virtex-4 the power consumption of the proposed method is decreased by 0.54% and 3.67%,

Virtex-5 the power consumption of the proposed method is decreased by 33.99% and 35.88% and in Artix-7 the power consumption of the proposed method is decreased by 76.46% and 76.69% when compared with conventional CORDIC algorithm and R. Ramadoss et al. [29] methods.

The analysis of frequency with four FPGA families is illustrated in Fig. 10. From the figure, the frequency of the proposed method is superior to 78.91% for conventional CORDIC method and 79.42% for R. Ramadoss et al. [29] method in Spartan-3A-DSP. In Virtex-4 the frequency of the proposed method is increased by 83.42% and 83.54%, Virtex-5 the frequency of the proposed method is superior to 79.89% and 80.09%, and in Artix-7 the frequency of the proposed method is increased by 77.01% and 77.11% when compared with conventional CORDIC algorithm and R. Ramadoss et al. [29] methods. The theoretical reason for better performance of proposed method is using Hub algorithm. Because, it generate multiplier block from a set of constants and reduced addition, subtraction and shifting operation. With 16-bit floating point inputs the FPGA experiments were shown above. Correspond to these inputs the obtained overheads are size of 16-bits. Thus, the overheads changes are negligible and to a large design space of CORDIC the proposed hardware reduction scheme is applied.

6. Conclusion

In this paper a low power and high speed CORDIC design with an improved power control and hardware reduction techniques has been proposed. In the design, the number of shifters and adder/subtractor are reduced by employing the canonical signed-digit (CSD) technique, and the Hcub algorithm. The proposed LH-CORDIC design is implemented in Xilinx ISE 14.5 with Spartan-3A-DSP (XC3SD1800A-4FG676), Virtex-4(XC4VVSX35-10FF668), Virtex-5 (XC5VLX20T-2FF323) and Artix-7 (XC7A100T-2CSG324) FPGA families. The proposed method performs better in terms of frequency, area and power consumption when compared with other architectures. Simulation results shows that the proposed method has higher frequency of 78.91%, 83.42%, 79.89% and 77.01% when compared with conventional CORDIC method. Also, the proposed method has lower power consumption of 28.52% and 32.57%, and lower area of 41.86% and 48.16% when contrasted with other architectures. In future scope, the CORDIC algorithm is further extended in VERILOG language for clock- pipelined architecture and calculates the

higher order and more complex problems and designs the digital filter with high speed, low power and more accuracy in VLSI and DSP domain. Also, to achieve better performance in the proposed system, different optimization techniques may also be incorporated.

References

- [1] H. Huang, L. Xiao, and J. Liu, "CORDIC-Based Unified Architectures for Computation of DCT/IDCT/DST/IDST", *Circuits, Systems, and Signal Processing*, Vol.33, No.3, pp.799-814, 2013.
- [2] K. Ray and A. Dhar, "CORDIC-Based VLSI Architectures of Running DFT with Refreshing Mechanism", *Journal of Signal Processing Systems*, Vol.90, No.2, pp.1-12, 2018.
- [3] D. Chen and M. Sima, "Fixed-Point CORDIC-Based QR Decomposition by Givens Rotations on FPGA", In: *Proc. of International Conference on Reconfigurable Computing and FPGAs*, pp.327-332, 2011.
- [4] P. Meher, J. Valls, Tso-Bing Juang, K. Sridharan, and K. Maharatna, "50 Years of CORDIC: Algorithms, Architectures, and Applications", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.56, No.9, pp. 1893-1907, 2009.
- [5] Y. Liu, L. Fan, and T. Ma, "A Modified CORDIC FPGA Implementation for Wave Generation", *Circuits, Systems, and Signal Processing*, Vol.33, No.1, pp.321-329, 2013.
- [6] R. Shukla and K. Ray, "Low Latency Hybrid CORDIC Algorithm", *IEEE Transactions on Computers*, Vol.63, No.12, pp.3066-3078, 2014.
- [7] S. Aggarwal, P. Meher, and K. Khare, "Area-Time Efficient Scaling-Free CORDIC Using Generalized Micro-Rotation Selection", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol.20, No.8, pp.1542-1546, 2012.
- [8] P. Vyas, L. Vachhani, K. Sridharan, and V. Pudi, "CORDIC-Based Azimuth Calculation and Obstacle Tracing via Optimal Sensor Placement on a Mobile Robot", *IEEE/ASME Transactions on Mechatronics*, Vol.21, No.5, pp.2317-2329, 2016.
- [9] H. Nguyen, X. Nguyen, C. Pham, T. Hoang, and D. Le, "A parallel pipeline CORDIC based on adaptive angle selection", In: *Proc. of International Conference on Electronics, Information, and Communications*, pp.1-4, 2016.
- [10] P. Velrajkuma, C. Senthilpar, G. Murthy, and E. Wong, "Low Energy, Improved Speed and High Throughput CORDIC Cell to Improve Performance of Robots' Processor", *Asian Journal of Scientific Research*, Vol.8, No.3, pp.381-391, 2015.
- [11] Y. H. Hu and H. Chern, "A novel implementation of CORDIC algorithm using backward angle recoding (BAR)", *IEEE Transactions on Computers*, Vol.45, No.12, pp.1370-1378, 1996.
- [12] T. Hoang, H. Nguyen, X. Nguyen, C. Pham, and D. Le, "High-performance DCT architecture based on angle recoding CORDIC and Scale-Free Factor", In: *Proc. of the Sixth International Conference on Communications and Electronics*, pp.199-204, 2016.
- [13] A. Y. Wu and C. S. Wu, "A unified view for vector rotational CORDIC algorithms and architectures based on angle quantization approach", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, Vol.49, No.10, pp.1442-1456, 2002.
- [14] C. S. Wu, A. Y. Wu, and C. H. Lin, "A high-performance/low-latency vector rotational CORDIC architecture based on extended elementary angle set and trellis-based searching schemes", *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol.50, No.9, pp.589-601, 2003.
- [15] J. Mehta and P. Trivedi, "An enhanced mixed-scaling-rotation CORDIC algorithm with weighted amplifying factor", In: *Proc. of the IEEE International Conference on Digital Signal Processing*, pp.527-531, 2016.
- [16] A. Banerjee and A. Dhar, "Pipelined VLSI Architecture using CORDIC for Transform Domain Equalizer", *Journal of Signal Processing Systems*, Vol.70, No.1, pp.39-48, 2012.
- [17] M. Madheswaran and T. Menakadevi, "An Improved Direct Digital Synthesizer Using Hybrid Wave Pipelining and CORDIC algorithm for Software Defined Radio", *Circuits, Systems, and Signal Processing*, Vol.32, No.3, pp.1219-1238, 2012.
- [18] H. Huang, W. Wang, B. Wu, and T. Gao, "CORDIC-based fast radix-2 DST algorithms", In: *Proc. of the International Conference on Software Intelligence Technologies and Applications & International Conference on Frontiers of Internet of Things*, pp.246-249, 2014.

- [19] B. Lakshmi and A. Dhar, "Low latency pipelined CORDIC-like rotator architecture", *International Journal of Electronics*, Vol.104, No.1, pp.64-78, 2016.
- [20] H. Huang and L. Xiao, "CORDIC based fast algorithm for power-of-two point DCT and its efficient VLSI implementation", *Microelectronics Journal*, Vol.45, No.11, pp.1480-1488, 2014.
- [21] C. Zhang, J. Han, and K. Li, "Design and Implementation of Hybrid CORDIC Algorithm Based on Phase Rotation Estimation for NCO", *The Scientific World Journal*, Vol.2014, No.1, pp.1-8, 2014.
- [22] L. Moroz, S. Nagayama, T. Mykytiv, I. Kirenko, and T. Boretsky, "Simple Hybrid Scaling-Free CORDIC Solution for FPGAs", *International Journal of Reconfigurable Computing*, Vol. 2014, No.6, pp.1-4, 2014.
- [23] S. Bhairannawar, S. Sarkar, K. Raja, and K. Venugopal, "Implementation of Fingerprint Based Biometric System Using Optimized 5/3 DWT Architecture and Modified CORDIC Based FFT", *Circuits, Systems, and Signal Processing*, Vol.37, No.1, pp.342-366, 2017.
- [24] W. Lin, W. Chou, T. Shiao, G. Shiao, C. Luo, and M. Lee, "Realization of a CORDIC-Based Plug-In Accelerometer Module for PSG System in Head Position Monitoring for OSAS Patients", *Journal of Healthcare Engineering*, Vol.2017, No.1, pp. 1-9, 2017.
- [25] B. Yang, L. Wu, B. Wang, and Q. Wang, "Digital Closed-Loop Driving Technique Using the PFD-Based CORDIC Algorithm for a Biaxial Resonant Microaccelerometer", *Journal of Sensors*, Vol.2017, No.1, pp.1-14, 2017.
- [26] M. Garrido, P. Kallstrom, M. Kumm, and O. Gustafsson, "CORDIC II: A New Improved CORDIC Algorithm", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.63, No.2, pp.186-190, 2016.
- [27] Y. Voronenko and M. Püschel, "Multiplierless multiple constant multiplication", *ACM Transactions on Algorithms*, Vol.3, No.2, p.11-es, 2007.
- [28] H. Liu and B. Yuan, "Low-power design and application based on CSD optimization for a fixed coefficient multiplier", *Science China Information Sciences*, Vol.54, No.11, pp.2443-2453, 2011.
- [29] R. Ramadoss, M. Mozaffari Kermani, and R. Azarderakhsh, "Reliable Hardware Architectures of the CORDIC Algorithm With a Fixed Angle of Rotations", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.64, No.8, pp. 972-976, 2017.