FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 872-876 | e-ISSN: 2347-470X

CORDIC Processors: A Comparative Perspective of Radix-8, Radix-4, and Radix-2

K. Victor Rajesh[1], K. Parvateesam[2], Vella Satyanarayana[3] and Sanjeev Kumar[4*]

[1,2]Department of ECE, Aditya College of Engineering & Technology, Surampalem, India, [1]kotavictorrajesh777@gmail.com, [2]parvateesam.kunda@acet.ac.in
[3,4]Department of ECE, Aditya Engineering College, Surampalem, India, [3]vasece_vella@aec.edu.in, [4]sanjeev.kumar@accendere.co.in

***Correspondence:** Sanjeev Kumar; sanjeev.kumar@accendere.co.in

ABSTRACT- Some data streaming applications make use of digital signal processing (DSP). The DSP algorithms include transcendental functions like trigonometry, inverse trigonometry, logarithms, exponentials, and other functions in addition to the basic arithmetic operations of multiplication and division. By modifying a few simple parameters, it is relatively simple to generate a large range of functions, including logarithmic, exponential, and trigonometric ones. Since the CPU needs around n rounds to process n bits of incoming data, the CORDIC radix-2 causes a substantial amount of latency. Therefore, radix-4 (which is radix-4 plus one) and radix-8 (which is radix-8 plus one) may be used to reduce the amount of time needed for the calculation. As a result, the total iterations drop from n/2 to n/4. As simulated and synthesized, it was shown that the radix-8 design had a significantly higher power consumption, with greater throughput than the radix-2 and radix-4 Coordinate Rotational Digital Computer (CORDIC) designs with an additional area overhead. The new algorithm also reduces the amount of twiddle elements to a minimum while also simplifying the address generating process. A novel approach makes integrating FFT processors on single chips much easier. This approach is demonstrated to be notably efficient in procedures such as SVD or matrix triangularization, in which the calculation of the rotation angle is necessary.

Keywords: CORDIC, Radix-8, Radix-4, Radix-2, FFT, DSP.

ARTICLE INFORMATION

Author(s): K. Victor Rajesh, K. Parvateesam, Vella Satyanarayana and Sanjeev Kumar;
Received: 14/06/2022; **Accepted**: 29/09/2022; **Published**: 18/10/2022;
e-ISSN: 2347-470X;
Paper Id: IJEER-RDEC10;
Citation: 10.37391/IJEER.100420
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100420.html

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

CORDIC is based on 2-D statistical principle and it is introduced by 'Volder' in 1959 [1]. It is used to calculate the hyperbolic, trigonometric, exponential functions and to improvement in the scale factor correction of the vector rotation [2]. It uses to shifts and adds to perform a wide range of functions and wide applications. Operations corresponding to the scales are derived when shift operations are applied. The creation of architectures for cost-effective, high-performance hardware solutions has occurred in the algorithmic design field. CORDIC constructed by repeating shift addition operations using simple hardware. In the single-stage iterative technique, less logic gates are required compared to multiplier circuits. CORDIC calculations employ just adders, which is a benefit because of the simple hardware it is implanted in. With CORDIC, numerous functions can be computed with the same hardware [9]. Applications with a focus on cost reduction above increasing speed are best. A good illustration of this method is in the pocket calculator.

Radix is the number of unique digits, including the digit zero. A lot of different locations in microprocessor design use radix multipliers [3]. For digital signal processing applications, multipliers are significant as well. A radix sort the data by separating the individual digits into their own buckets, followed by sorting the data with integer keys. For a Radix operation sorting algorithm also makes a stable algorithm [4]. Radix-2 CORDIC is processed the latency is large, it uses the ten digits from 0 to 9, For radix-2 is the integral power of two, the algorithm performs a subdivision of the input sequence into its. For radix-4 any number in the radix-4 system is half of the value in the radix-2 system [5]. For radix-8 it uses a group of 4-bits. If we want to decrease the latency of radix, Radix-4 and Radix-8 can be used. Hardware architecture iteration of CORDIC procedure is exposed in *figure 1*.

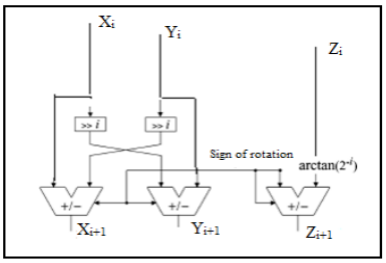

Figure 1: Hardware architecture iteration of CORDIC algorithm

For faster shifting and adding Booth multiplication will be performed. By constructing and synthesizing them, the Radix-2 Booth encoding multiplier, Radix-4 Booth encoding multipliers, Radix-8 Booth encoding multipliers, Radix-16 Booth encoding multipliers, and Radix-32 Booth encoding

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 872-876 | e-ISSN: 2347-470X

multipliers advance the study one step additional. For a Radix operation, FFT can be easy to compute the DFT of sequence order [6, 21]. In a radix operation with an FFT for using the algorithm that can decompose a long FFT into several short FFT's. RAM and twiddle factor ROM on chip are accessible for the memory data. Every FFT processor can increase its performance in the pipelines [20, 22]. For this, the radix algorithm usually calculates with FFT. In this paper what we know about, how the radix operation must be faster than other operations [7, 19]. This is the easy way to perform such a radix operation using CORDIC implementation and other applications [8].

2. RELATED WORK

In 1996, the vectoring mode of the CORDIC approach is based on the use of radix-4 method with radix-2 convolution. The technique for execution and storage instructions is for carrying out certain radix operations for a pipeline process [9, 18]. The approach is therefore particularly effective in procedures such as SVD or matrix triangularization where an assessment of the rotation angle is needed [1]. VLSI technology allowed high-radix CORDIC algorithm for high-speed sinus and cosine computing in CORDIC-based dedicated function generators within digital signal transmitters. When low latency calculation is performed by changing the radix during operation of the radix 2-4-8 CORDIC processors [2]. A tiny ROM with steering and pipeline latches consists of the Bit Level Domain cell. The cell can execute all the computing functions needed in the processor; the unique structure and simplicity of the arrays are demonstrated by typical VLSI manufacturing [10]. The production of the VLSI is possible via an FFT element [3]. DSP was achieved in FFT. Unfortunately, it is difficult to pipeline conventional FFT algorithms, while the FFT VLSI implementation is appropriate for UWB communication because data speed and low power are ensured [11]. This means that the digital signal processing is in the complete world. For a CORDIC twiddles processor, FFT is used to consume the speed and power of the most important parameters that influence high speed and low cost of transmission requirements. With CORDIC processors we prompt the high-speed FFT VLSI hardware architecture [4]. In 2005, the CORDIC algorithm was a processing iterative arithmetic for DSP applications. DSP applications may be subject to the proposed MSR CORDIC. By regulating the internal dynamic range, the MSR CORDIC algorithm enhances signal to quantization of noise (SQNR) [12]. SQNR performs statistical properties both first and second order. The CORDIC method just performs the shift sequence and adds operation; there are much iteration at the disadvantage [13]. The overhead of the scaling procedure, unavoidable in present CORDIC algorithms, can be eliminated. Compared to the newly proposed extension of the basic CORDIC algorithm, it can further minimize the hardware complexity [14]. The MSR CORDIC system was used to create variable FFT processors and to cut hardware in the operation of the twin factor [5].

In 2012 the performance is accomplished for the design and synthesis of all radiix-based multipliers. In comparison with radix-2 booth coding multiplier, the door-level logic performance is again the lowest compared to the synthesizing outputs of the radix-4 multiplier [16]. The multipliers of Radix-4 and Radix-8 are the low-field performance vs the booth encoded multiplier of Radix-2 [6]. In the 2015 FFT pipeline the Radix-2k algorithm plays a part. The development of radix-2k FFT is part of the one-way retardation feedback (SDF) [17]. In digital signal processing, the DFT is an essential instrument [15]. In order to transmit large amounts of digital data across a radio wave, for digital communication multiplexing by the Orthogonal Frequency Division Multiplexing (OFDM) operates by separating the radio signal into several sub signal sub-signals [7].

3. METHODOLOGY FOR CORDIC PROCESSORS

The process of embedding data in medicinal imagery is dissimilar.

The CORDIC processor is capable of calculating a wide variety of functions, including sine and cosine of angles. CORDIC's operation can be carried out in two modes, as well as rotation and vectoring. Following rotation by a specific angle, the components 'X' and 'Y' of the original vector's coordinates are determined. These values, which show the sine and cosine values of the with function 'Z' converging to zero, prove that 'X' and 'Y' are the original vector's components.
This is the following equation for making CORDIC to radix-2 CORDIC processor.

$$X_{i+1} = (X_i - \sigma_i Y_i 2^{-i})k_i \quad (1)$$

$$Y_{i+1} = (Y_i + \sigma_i X_i 2^{-i})k_i \quad (2)$$

$$Z_{i+1} = (Z_i - \sigma_i \tan^{-1}(2^{-i})) \quad (3)$$

Where i = 0,1, 2, 3, ……. N-1. σ_i = {-1,0,1} in redundant scheme.

Scale factor:

$$k_i = \Pi_0^{n-1}\ k_i = \Pi_{i=0}^{n-1}\sqrt{1 + 2^{-2i}} \quad (4)$$

σ_i depends on sign of Z_i computation.

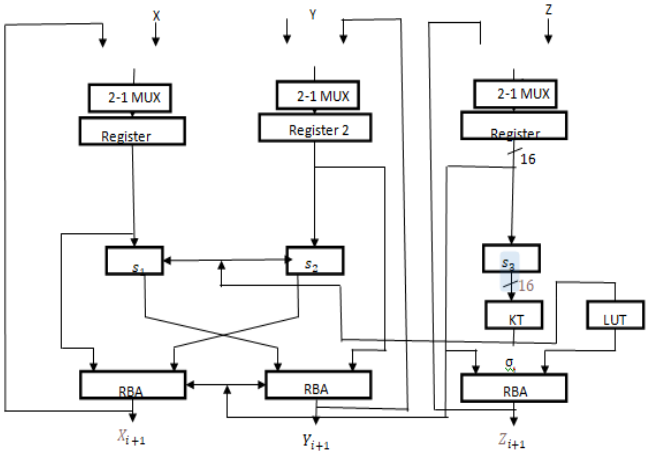

Figure 2: Radix-2 CORDIC Architecture

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 872-876 | e-ISSN: 2347-470X

Equation for making CORDIC to radix-4 CORDIC processor

$X_{i+1} = X_i + \sigma_i Y_i 4^{-I}$ (5)
$Y_{i+1} = Y_i - \sigma_i X_i 4^{-i})$ (6)
$Z_{i+1} = Z_i - |\sigma_i|\alpha_i$ (7)

Where the coefficient $\sigma \varepsilon$ { -2,-1,0,1,2}

$\alpha_i[\sigma_i] = \tan^{-1}(\sigma_i 4^{-i})$ (8)
$w_i = 4^i z$ (9)
$w_{i+1} = 4(w_i - 4^i \tan^{-1}(\sigma_i 4^{-i}))$ (10)

The following selection table for i=0 and i >1 is used by the Radix-4 CORDIC processor.

The equation for Radix – 8 CORDIC processors:
$X_{i+1} = X_i - \sigma_i Y_i 8^{-i}$ (11)
$Y_{i+1} = Y_i + \sigma_i X_i 8^{-i}$ (12)
$Z_{i+1} = Z_i - \sigma_i (\alpha_i)$ (13)
$w_i = 8^i z_i$ (14)
$w_{i+1} = 8(w_i - 8^i \tan^{-1}(\sigma_i 8^{-i}))$ (15)

Where, d_i = {-3,-2,-1, 0, 1, 2, 3}

Table 1: Radix-4 CORDIC Processor selection function

Selection function		For I=0	Selection function		For I>0
	+2	$5/8 \le \hat{w}_0$		+2	$\hat{w}_1 \ge 3/2$
	+1	$3/8 \le \hat{w}_0 < 5/8$		+1	$1/2 \le \hat{w}_1 < 3/2$
σ_o	0	$-1/2 \le \hat{w}_0 < 3/8$	σ_o	0	$1/2 \le \hat{w}_1 < 1/2$
	-1	$-7/8 \le \hat{w}_0 < 3/8$		-1	$3/2 \le \hat{w}_1 < -1/2$
	-2	$\hat{w}_0 < -7/8$		-2	$\hat{w}_1 < -3/2$

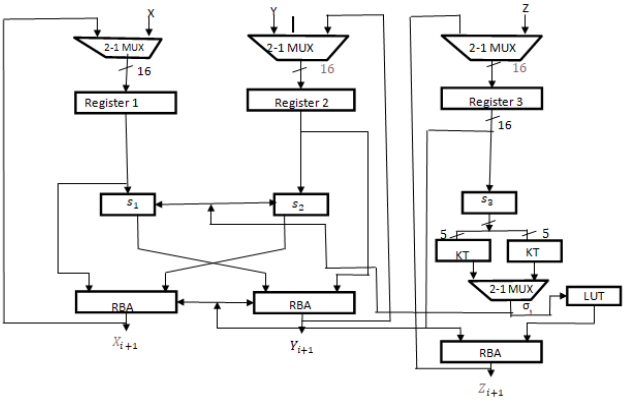

Figure 3: Radix-4 Architecture of CORDIC processor

Table 2: Function for Radix-8 CORDIC Processor

Selection function		For I =0	Selection function		For I =1
	=+4	$9/8 < \hat{w}_0$		=+4	$\hat{w}_1 \ge 7/2$
	=+3	$7/8 \le \hat{w}_0 \le 9/8$		=+3	$5/2 \le \hat{w}_1 \le 7/2$
	=+2	$5/8 \le \hat{w}_0 \le 7/8$		=+2	$3/2 \le \hat{w}_1 \le 5/2$
	=+1	$3/8 \le \hat{w}_0 \le 5/8$		=+1	$1/2 \le \hat{w}_1 \le 3/2$
σ_o	0	$-1/2 \le \hat{w}_0 \le 3/8$	σ_o	0	$-1/2 \le \hat{w}_1 \le 1/2$
	=-1	$-7/8 \le \hat{w}_0 \le -1/2$		=-1	$-3/2 \le \hat{w}_1 \le -1/2$
	=-2	$-5/4 \le \hat{w}_0 \le -7/8$		=-2	$-5/2 \le \hat{w}_1 \le -3/2$
	=-3	$-13/8 \le \hat{w}_0 \le -5/4$		=-3	$-7/2 \le \hat{w}_1 \le -5/2$
	=-4	$\hat{w}_0 \le 13/8$		=-4	$\hat{w}_1 < -7/2$

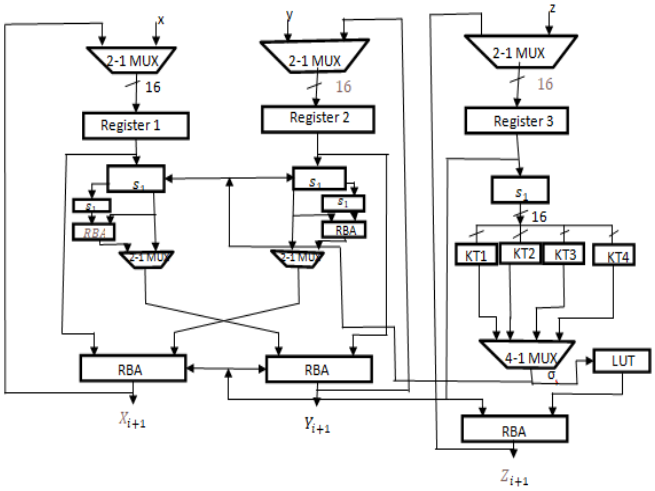

Figure 4: Radix-8 CORDIC Architecture

Radix booth encoding multiplier

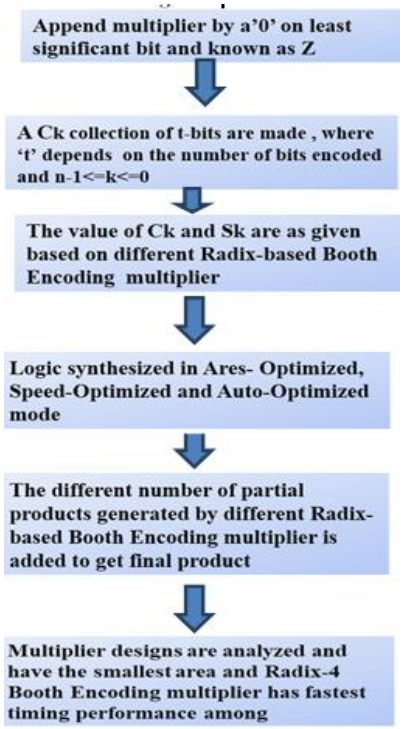

Figure 5: Core methodology of a radix-based algorithm

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 872-876 | e-ISSN: 2347-470X

4. EXPERIMENTAL RESULTS AND DISCUSSION

VHDL has carried out the proposal for the hardwired pre-shifted CORDIC bi-rotation. There could not be optimized precise angles for basic CORDIC designs for the vector rotation, nor for standard HDCCORDIC pre-shift bi-rotation with barrel shifter and adder complications for fixed and known angular rotation. In order to reduce complications of the barrel and adder, the proposed CORDIC cascade bi-rotation. The combined SDF FFT Radix-2,4 and 8 were created with MODELSIM 6.3C. For the synthesis of FFT architecture, the Xilinx ISE 10.1i Design Tool. *Figure 5* shows the results of the simulation. The examination of the R2SDF FFT now in use and the combined radix-2,4 and 8 based FFT proposed are shown in *table 1*. The current and proposed SDF FFT combined with radix 2.4 and 8 is illustrated in the performance of *figure 6*.

Table 3: Comparison with the combined SDF FFT Radix-2, 4 and 8 suggested

Types of parameters	No of Slices occupied	No. of LUTs	Delay (ns)	Power (W)
SDF-FFT Existing Radix-2	609	850	15.062	2.197
Combined radix-2,4 and 8 SDF FFT proposed	251	319	9.360	1.519
Percentage reduction %	58.78	62.47	37.85	30.86

R2MDC (Spin Delay Commutator) Radix-2 and R2SDF (Spin Delay Feedback Radix-2) are provided. R2MDC is greater than the R2SDF FFT structure in the delay element. Compared with R2SDF, hardware usage for R2MDC is negligible.

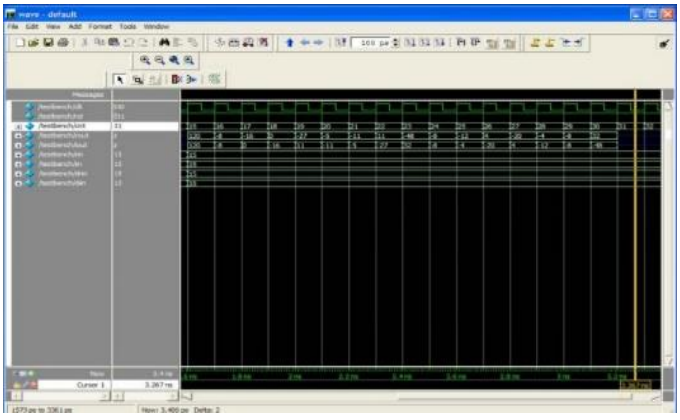

Figure 6: The combined radix-2, 4, and 8 SDF ft simulation result

The combined radius 2, 4, and 8 results are shown in *figure 6*. R2SDF FFT provides sequentially inputs and parallel intermediate operations. This architecture is hence known as parallel pipelines. The current Radix-2 SDF FFT approach provides increased use of hardware as well as computing steps.

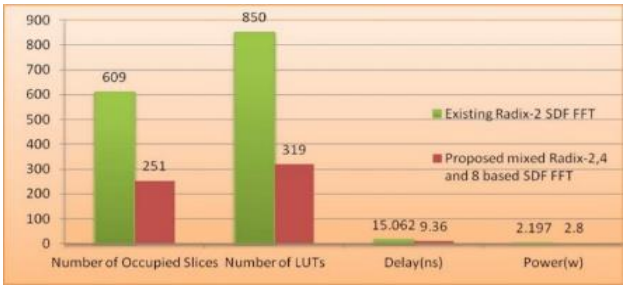

Figure 7: Performance assessment of SDF FFT Radix-2, 4 and 8

Figure 7 illustrates the current R2SDF FFT performance evaluation and suggested combined SDF FFT radix-2, 4 and eight. The number of occupied slices is higher than power when suggested SDFFT- 2, 4 and 8 based radix is smaller than the existing Radix-2 SDF FFT (w).

5. CONCLUSION

This research notes that radix-8 CORDIC is additional energy-efficient than radix-2 and radix-4 CORDIC architectures and provides a high level of energy efficiency. But the use of hardware is higher for radix-8. Therefore, the projected FFT architecture with an input and output sequence in normal order is particularly desirable for the radix-2 FFT single-path processors. In Xilinx ISE 10, the algorithms Radix-8, Radix-4 and Radix-2 are explored and simulated. These are obtained with circular shift operation and lowered the twelfth factor. In literature it was a major problem to develop low power and high speed CORDIC, and there was always space for further improved outcomes. In this thesis, which is called Low Power and High Speed CORDIC, a simple method is given to handle the problem. Compared to the present on-demand algorithm, it has the lowest power and area.

REFERENCES

[1] Villalba, J., Zapata, E. L., Antelo, E., & Bruguera, J. D. (1998). Radix-4 vectoring cordic algorithm and architectures. Journal of VLSI signal processing systems for signal, image and video technology, 19(2), 127-147.

[2] Villalba, J., Arrabal, J. C., Zapata, E. L., Antelo, E., & Bruguera, J. D. (1996, August). Radix-4 vectoring CORDIC algorithm and architectures. In Proceedings of International Conference on Application Specific Systems, Architectures and Processors: ASAP'96 (pp. 55-64). IEEE.

[3] Aoki, T., Nogi, H., & Higuchi, T. (1997, November). High-radix CORDIC algorithms for VLSI signal processing. In 1997 IEEE Workshop on Signal Processing Systems. SiPS 97 Design and Implementation formerly VLSI Signal Processing (pp. 183-192). IEEE.

[4] Antelo, E., Villalba, J., & Zapata, E. L. (2008). A low-latency pipelined 2D and 3D CORDIC processors. IEEE Transactions on Computers, 57(3), 404-417.

[5] Zhang, G., & Chen, F. (2004, September). Parallel FFT with CORDIC for ultra-wide band. In 2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No. 04TH8754) (Vol. 2, pp. 1173-1177). IEEE.

[6] Lin, C. H., & Wu, A. Y. (2005). Mixed-scaling-rotation CORDIC (MSR-CORDIC) algorithm and architecture for high-performance vector rotational DSP applications. IEEE Transactions on Circuits and Systems I: Regular Papers, 52(11), 2385-2396.

[7] Wang, S., Piuri, V., & Swartzlander, E. E. (1996, August). A unified view of CORDIC processor design. In Proceedings of the 39th Midwest Symposium on Circuits and Systems (Vol. 2, pp. 852-855). IEEE.

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 4 | Pages 872-876 | e-ISSN: 2347-470X

[8] Brundana, M. S. S., Rajeswari, P. S. R., Sravani, N., & Kumar, S. (2021, April). Successive Approximation Compressor for Efficient FIR Filters in C-MOS VLSI Design. In 2021 6th International Conference for Convergence in Technology (I2CT) (pp. 1-4). IEEE.

[9] Nandini, A., Kumar, R. A., & Singh, M. K. (2021, October). Circuits Based on the Memristor for Fundamental Operations. In 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC) (pp. 251-255). IEEE.

[10] Swee, K. L. S., & Hiung, L. H. (2012, June). Performance comparison review of Radix-based multiplier designs. In 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012) (Vol. 2, pp. 854-859). IEEE.

[11] Kavitha, M. S., & Rangarajan, P. (2020). An efficient FPGA architecture for reconfigurable FFT processor incorporating an integration of an improved CORDIC and radix-2r algorithm. Circuits, Systems, and Signal Processing, 39(11), 5801-5829.

[12] Raut, G., Bhartiy, V., Rajput, G., Khan, S., Beohar, A., & Vishvakarma, S. K. (2019, July). Efficient low-precision cordic algorithm for hardware implementation of artificial neural network. In International Symposium on VLSI Design and Test (pp. 321-333). Springer, Singapore.

[13] Duprat, J., & Muller, J. M. (1993). The CORDIC algorithm: new results for fast VLSI implementation. IEEE Transactions on Computers, 42(2), 168-178.

[14] Hu, X., Harber, R. G., & Bass, S. C. (1991). Expanding the range of convergence of the CORDIC algorithm. IEEE Computer Architecture Letters, 40(01), 13-21.

[15] Hu, Y. H., & Naganathan, S. (1993). An angle recoding method for CORDIC algorithm implementation. IEEE Transactions on Computers, 42(1), 99-102.

[16] Sharma, N. K., Rathore, S., & Khan, M. R. (2020, January). A comparative analysis on coordinate rotation digital computer (CORDIC) algorithm and its use on computer vision technology. In 2020 First International Conference on Power, Control and Computing Technologies (ICPC2T) (pp. 106-110). IEEE.

[17] M. R. Ezilarasan and J. Britto Pari, A. Dahir Alramadan and M. AL-Shakban (2022), An Optimized Pipeline Based Blind Source Separation Architecture for FPGA Applications. IJEER 10(3), 632-638. DOI: 10.37391/IJEER.100336.