

## Area-Power Efficient Multi Staged Pipelined CORDIC Using Micro-Rotation Selection

Mahendra kumar M D<sup>1</sup>, Sunil M P<sup>2</sup>

<sup>1</sup>MTEch student (SP AND VLSI)

<sup>2</sup>Asst. Professor

ECE Department, School of Engineering and Technology, Jain University, Bangalore, Karnataka, India

---

**Abstract :** CORDIC is an acronym for CORDinate Rotation Digital Computer. The CORDIC method is the most versatile of all the algorithms that can be used to evaluate elementary functions. It 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), discrete sine/cosine transforms (DST/DCT), householder transform (HT), etc. CORDIC algorithm, on the other hand, offers an excellent alternative, and its best characteristic is flexibility. Its quantization accuracy is a function of word length. Hardware implementation of CORDIC results increase in Critical path delay. Pipelined architecture is used in CORDIC to increase the clock speed and to reduce the Critical path delay, it is multiplier less approach and it saves a lot of hardware and hence power dissipation is very low as compared to other methods. Due to the simplicity of the involved operations, the CORDIC algorithm is very well suited for VLSI implementation.

In this paper, a multi stage pipelined CORDIC architecture is compared and used for designing a flexible and scalable digital sine and cosine waves generator. The design of an application specific CORDIC processor in circular rotation mode give a high system throughput due to pipelined architecture by reducing latency in the each individual pipelined stages. Saving area on FPGA is essential to the design of pipelined CORDIC and can be achieved through optimizing the number of micro rotations. An FPGA-based architecture is presented the design has been implemented on a Xilinx Spartan 3 (xc3s200) device. Synthesis and implementation results are shown and also we use Most-significant-1 bit detection technique for micro-rotation sequence generation to reduce the number of iteration.

**Keywords:** Co-ordinate rotation digital computer (CORDIC), cosine/sine, Fast fourier transform (FFT), Discrete sine/cosine transforms (DST/DCT), Householder transform (HT), Most-significant-1, Field programmable gate array (FPGA).

---

### I. Introduction

CORDIC stands for Coordinate Rotation Digital Computer is a shift and add algorithm used to compute trigonometric, hyperbolic, linear and logarithmic functions. The CORDIC algorithm is first introduced by Jack.E Volder in year 1959[1] and further extended by Walther [2]. The CORDIC algorithm has found its various applications such as pocket calculator, numerical coprocessor, and image processing applications, direct digital synthesis and analog digital modulation.

CORDIC operates mainly in two modes for computation of different functions. These modes are known as rotation mode and vector mode. In rotation mode, the co-ordinate components of a vector and an angle of rotation is given and the co-ordinate component of original vector, after rotation through given angle are computed. In vector mode, the coordinate component of a given vector is given and the magnitude and angular argument of original vector are computed. The CORDIC technique uses a one bit at a time approach to make computation to an arbitrary precision [3]-[5]. Typically, these tables only one to two entries per bit of precision. CORDIC algorithms also use only right shifts and additions, minimizing the computation time. It is hardware efficient algorithm because no multipliers are presenting in CORDIC, to save gate required implementing on FPGA. If multiplier is present, then cost and number of gates increases.

The CORDIC algorithm has become a widely used approach to elementary function evaluation where the silicon area is a primary constraint. Pipelined CORDIC architecture is implemented in order to reduce iterative cycle and to increase the clock speed.

This paper is organized as follows. Section 2: Introduces Over view of CORDIC Algorithm. Section 3: Proposed CORDIC Processor. Section 4: Describes the Pipelined architecture. Section 5 and 6: Discuss the simulation and result and Conclusion.

## II. BRIEF OVERVIEW OF CORDIC ALGORITHM

CORDIC or Coordinate Rotation Digital Computer is a simple and hardware-efficient algorithm for the implementation of various elementary, especially trigonometric, functions. Instead of using Calculus based methods such as polynomial or rational functional approximation, it uses simple shift, add, subtract and table look-up operations to achieve this objective. The CORDIC algorithm was first proposed by Jack E Volder in 1959. It is usually implemented in either Rotation mode or Vectoring mode. In either mode, the algorithm is rotation of an angle vector by a definite angle but in variable directions. This fixed rotation in variable direction is implemented through an iterative sequence of addition/subtraction followed by bit-shift operation. The final result is obtained by appropriately scaling the result obtained after successive iterations. Owing to its simplicity the CORDIC algorithm can be easily implemented on a VLSI system.

The CORDIC algorithm operates either in, rotation vectoring mode, following linear, circular or hyperbolic coordinate trajectories. In this paper, we focus on rotation mode CORDIC using circular trajectory.

### 2.1. CORDIC Algorithm

The basic idea of CORDIC is to rotate the vector over given angle. Each basic rotation is realized by using shift and add operations. A vector is rotated through fixed number of steps called as iterations. If a vector  $V$  having co-ordinates (x and y) is rotated through an angle  $\phi$  then obtaining a new vector with co-ordinates where  $x'$  and  $y'$  can be obtained using following method.

$$X = r \cos\theta, Y = r \sin\theta \quad (1)$$

$$V' = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \cos\phi - y \sin\phi \\ y \cos\phi + x \sin\phi \end{bmatrix} \quad (2)$$

Micro rotation  $\phi_i$  is performed by vector at each iteration 'i', so new vector is given by

$$x_{i+1} = x_i \cdot \cos\phi_i - y_i \cdot \sin\phi_i \quad (3)$$

$$y_{i+1} = y_i \cdot \cos\phi_i + x_i \cdot \sin\phi_i \quad (4)$$

Factorizing cos terms vector components given as

$$x_{i+1} = x_i \cdot \cos\phi_i (x_i - y_i \cdot \tan\phi_i) \quad (5)$$

$$y_{i+1} = y_i \cdot \cos\phi_i (y_i + x_i \cdot \tan\phi_i) \quad (6)$$

As cosine is an even function, so  $\cos(\alpha) = \cos(-\alpha)$ . then "equation 5 & 6" becomes

$$x_{i+1} = k_i (x_i - y_i d_i 2^{-i}) \quad (7)$$

$$y_{i+1} = k_i (y_i + x_i d_i 2^{-i}) \quad (8)$$

Where 'i' is the number of iteration required by vector to reach the required angle, k factor is given as

$$K = \prod_{i=0}^{n-1} k_i \quad (9)$$

Where  $k_i$  is CORDIC gain.

Reducing original given rotation to add shift algorithm given as

$$x_{i+1} = x_i - d_i y_i 2^{-i} \quad (10)$$

$$y_{i+1} = y_i + d_i x_i 2^{-i} \quad (11)$$

A new variable known as accumulator is given as

$$z_{i+1} = z_i - d_i \phi_i \quad (12)$$

$$d_i = \pm 1 \quad (d_i \text{ is the direction of angle of rotation})$$

Where  $\varphi_i = \tan^{-1} 2^{-i}$  is pre-computed and stored in table for different value of ‘i’.

### III. Proposed Cordic Processor

In this paper, we propose a novel scaling-free CORDIC algorithm for area-time efficient implementation of CORDIC with adequate RoC. The proposed recursive architecture has comparable or less area complexity with other existing scaling-free CORDIC algorithms. Moreover, no scale-factor multiplications are required for extending the RoC to entire coordinate space, as required in [8]–[10].

The proposed design is based on the following key ideas:

- 1) We use Taylor series expansion of sine and cosine functions to avoid scaling operation and
- 2) Suggest a generalized sequence of micro-rotation to have adequate range of convergence (RoC) based on the chosen order of approximation of the Taylor series.

The block diagram for the proposed CORDIC processor is shown in Figure1. It makes use of the different stage for all the iterations for the coordinate calculations, as well as for the generation of shift values.

The structure of each stage shown in Figure 2 [3] consists of three computing blocks namely: the

- 1) Shift-value estimation;
- 2) Co-ordinate calculation; and
- 3) Micro-rotation sequence generator.

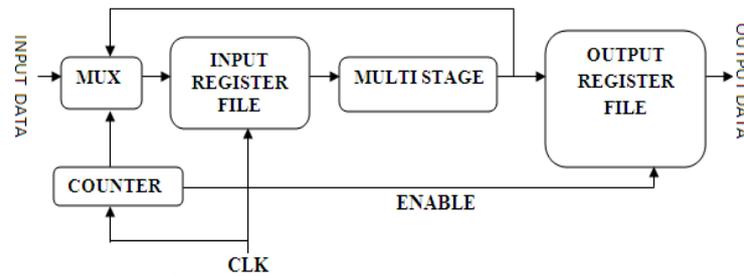


Fig 1: Recursive architecture of the proposed CORDIC processor.

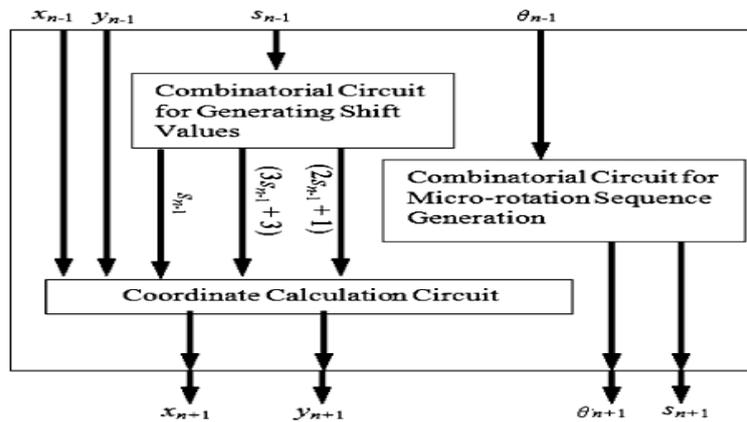


Fig 2: Block diagram for the each stage

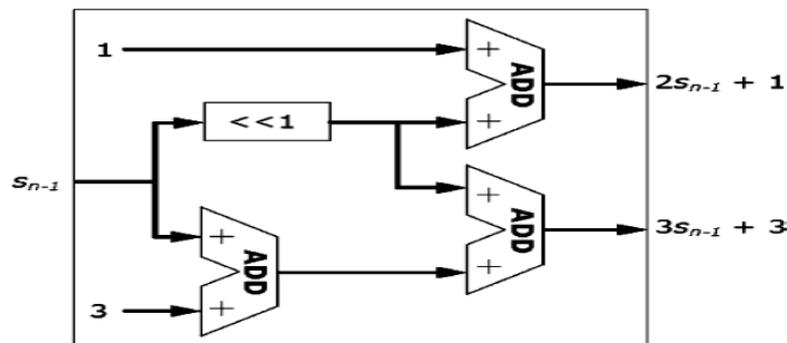


Fig 3: Combinational circuit for generating the shift value

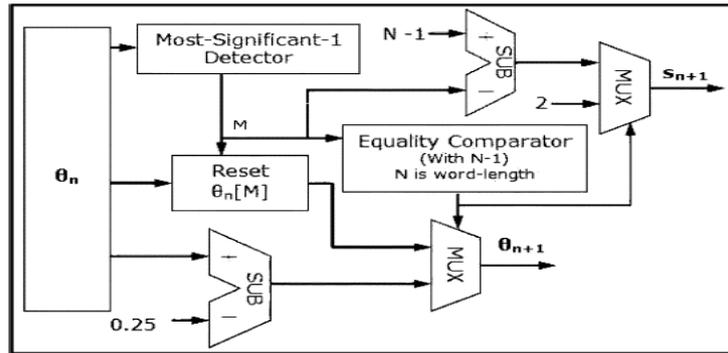


Fig 4: Micro rotation sequence generation

**Advantages:-**

- This architecture has an advantage over other implementation algorithm in terms of speed and accuracy.
- Area consumption is less.
- CORDIC is generally faster than other approaches when a hardware multiplier is unavailable (e.g., in a microcontroller based system), or when the number of gates required to implement the functions it supports should be minimized (e.g., in an FPGA).
- Better throughput.
- Less power consumption.

**Applications:-**

- The algorithm was basically developed to offer digital solutions to the problems of real-time navigation in B-58 bomber [6].
- CORDIC algorithm has also been described for the calculation of DFT,DHT [5], Solving linear systems [8].
- Most calculators especially the ones built by Texas Instruments and Hewlett-Packard use CORDIC algorithm for calculation of transcendental functions.
- John Walther extended the basic CORDIC theory to provide solution to and implement a diverse range of functions [7].

**IV. PIPELINED ARCHITECTURE**

Depending upon the application, CORDIC Processor is implemented in number of ways. The simple architecture is serial architecture consist of three adder/rom containing lookup table. Serial architecture perform one micro rotation for every clock cycle. Output is obtained after n clock cycle. Since serial architecture uses n clock cycle for every rotation hence it is very slow. Figure5 shows the serial architecture.

It requires...

- Maximum number of Clock Cycles to calculate output.
- Minimum Clock Period per iteration.
- Variable Shifters do not map well on certain FPGA's due to high Fan-in.

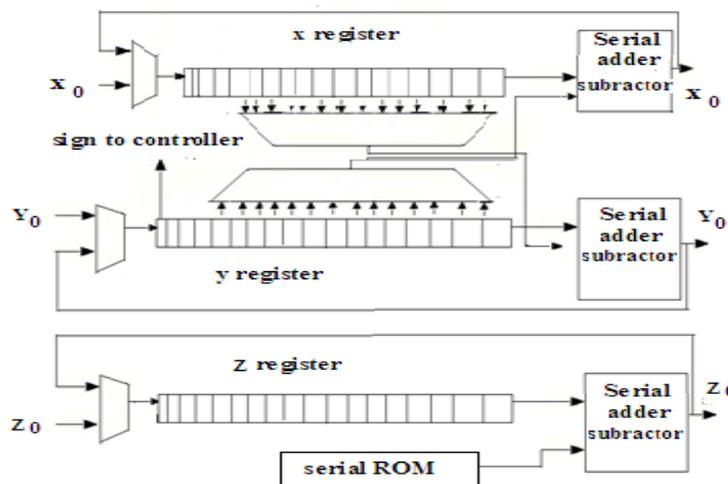


Fig 5: Shows Serial architecture

Pipelined architecture converts iterations in to pipeline phrases. It consists of n cascaded blocks. The first output of n stage CORDIC is after every clock cycle. Pipelined architecture having shift register that perform fixed number of shifts every time. Registers are used to store the angle for a particular micro rotation.

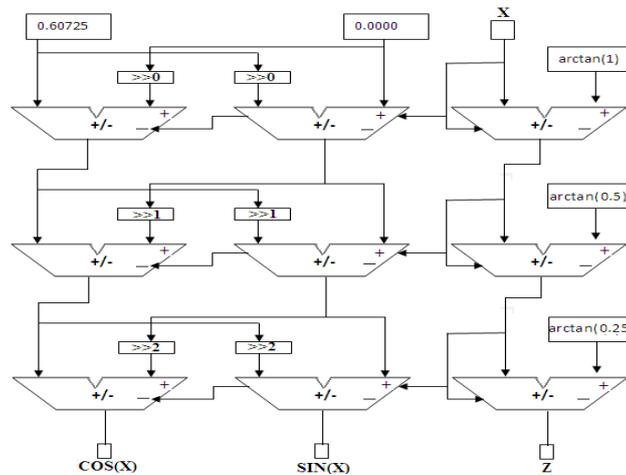


Fig 6: The pipelined architecture

Pipelined architecture is much faster than serial iteration at each stage. Sign ‘z’ gives the direction of iterations at each stage. In this paper a sixteen stage pipeline sine cosine wave generator is developed specific micro-rotations[3].

It has....

- Combinational circuit.
- More Delay, but processing time is reduced as compared to iterative circuit.
- Constants can be hardwired instead of requiring storage space.
- Shifters are of fixed shift, so they can be implemented in the wiring.

This architecture is fast than serial architecture since it doesn't require any lookup table. It operates in circular rotation mode . Sine and cosine terms are given by

$$X_n = \cos\theta \tag{13}$$

$$Y_n = \sin\theta \tag{14}$$

### V. RESULTS AND DISCUSSIONS

CORDIC algorithm is used to compute  $\sin\theta$  and  $\cos\theta$  by vector rotation method. Section (5.1), (5.2) consists of ModelSim simulation result and Xilinx Simulation results for input angle and outputs angle sine and cosine in the form of the waveform for 4 staged pipelined CORDIC and their corresponding magnitude.

#### 5.1 Xilinx Simulation results 4 stages Cordic:

Block diagram generated by XILINX 10.1i for sine-cosine using CORDIC is shown in Figure6. Here inputs are angle (binary input), clk clock), reset and outputs are sine (binary output), cosine (binary output), done. Figure 5 shows the Top level RTL schematic for sine-cosine generator. And Figure7 shows the inner view of internal block diagram RTL schematic where CORDIC consist of four stages.

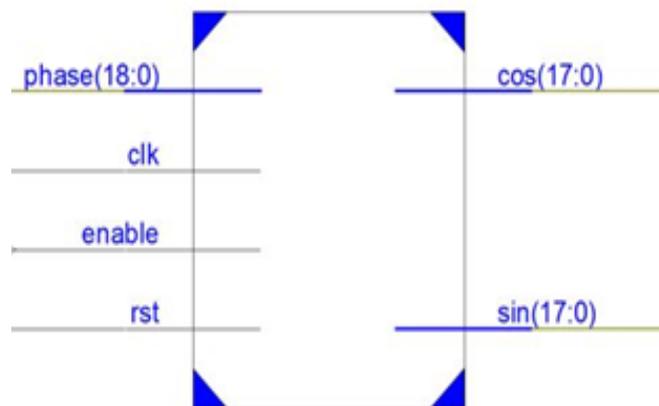


Fig 5: Top level RTL schematic for sine-cosine (CORDIC) for 19 bit

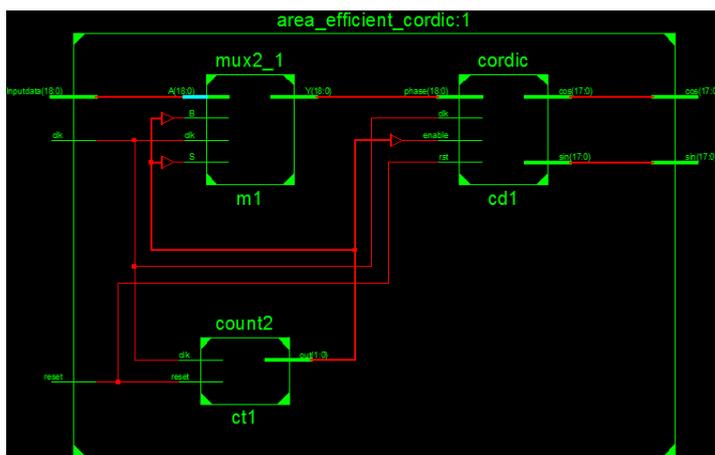


Fig 6: Internal RTL schematic of sine-cosine for 19 bit

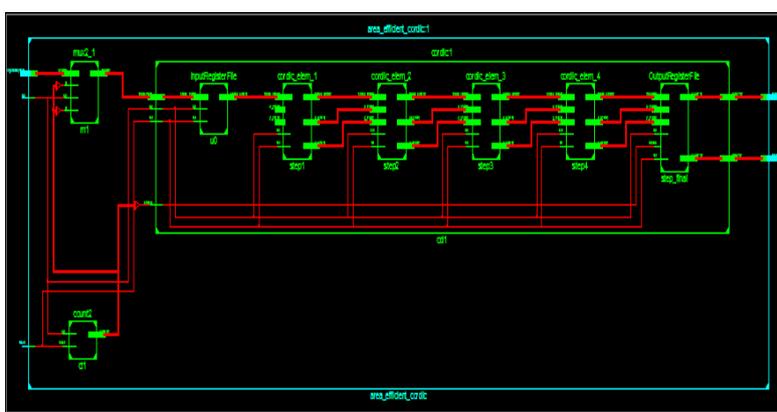


Fig 7: Inner view of Internal RTL schematic of sine-cosine for 19 bit

The code for sine and cosine wave generator is written in Verilog and simulated using ModelSim 10.0a. The digital wave generator to be implemented on XILINX Spartan 3xc3s200 using XILINX 14.1. Area and timing reports are given for particular target device shown in Figure8.

Device Utilization Summary			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	195	3,840	5%
Number of 4 input LUTs	276	3,840	7%
Number of occupied Slices	192	1,920	10%
Number of Slices containing only related logic	192	192	100%
Number of Slices containing unrelated logic	0	192	0%
Total Number of 4 input LUTs	296	3,840	7%
Number used as logic	273		
Number used as a route-thru	20		
Number used as Shift registers	3		
Number of bonded IOBs	57	97	58%

Fig 8: View summary report for 19bit (4 Staged CORDIC)

The power dissipation of the proposed architecture for different clock frequencies is estimated by Xilinx XPA tool show in Figure 9..

Quiescent(W)	0.037
Dynamic (W)	0.004
Total (W)	0.041

Fig 9: View power report for 19bit (4 staged CORDIC)

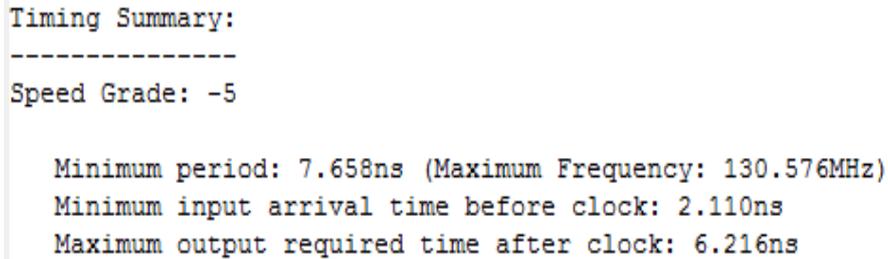


Fig 10: View timing report for 19bit (4 staged CORDIC)

**5.2 ModelSim Simulation Results:**

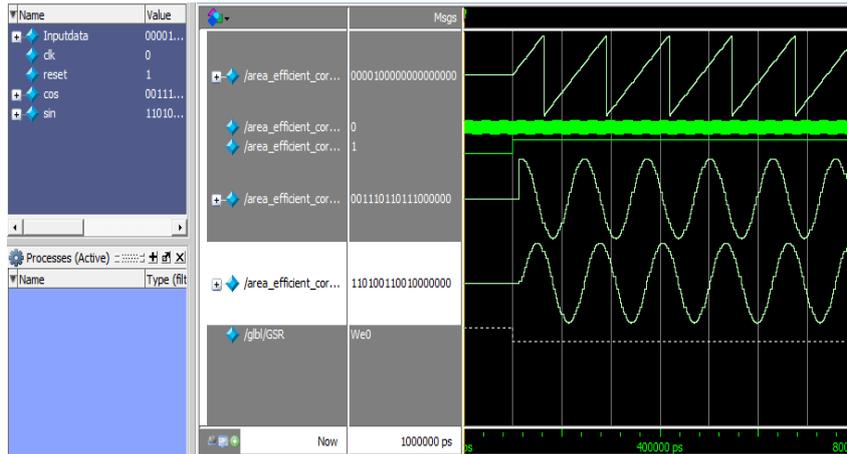


Figure11: Sine and Cosine waveform using Modelsim simulation

The simulated results for pipelined CORDIC architecture using 4 staged are shown above. The area and timing report for 4 staged states that the required number of slices i.e area is consumed less, speed of the architecture is more and the consumption is less compared to that of succeeding stages. Similarly the simulation results are calculated for 8,12 and 16 stages of pipelined CORDIC architecture and tabulated results are shown in Table 1.

TABLE 1 Parameters Comparisons for Different Stages of Pipelined Architecture

No. Of Stage	Area (slices)	Speed (frequency) MHz	Dynamic power (mW)	Minimum period (ns)	Slice product delay
4	192	130.576	4	7.658	11.76
8	517	114.108	4	8.764	36.24
12	767	113.222	5	8.832	54.19
16	978	114.108	5	8.764	68.56

**5.3 Discussions**

Analyzing the Table 1, it concludes that as the number of stages increases there is corresponding increase in the area, power & slice delay product and also decrease in the speed of the architecture. So it can be concluded that the 4 staged pipelined architecture has the better performance than the succeeding stages. Hence the 4 stage architecture is efficient when compared to previous work as shown in Table 2

**5.3.1 Comparison of present Pipelined CORDIC Sine and Cosine wave generator with previous work**

Present work is implemented on XILINX Spartan 3 (xc3sv200) using XILINX 14.1. Thus, finally comparison of present work with previous work is done as shown in Table 2

TABLE 2 Slice Delay Product Comparison for Different Approaches

Design *Values taken from[9]*	No. of Slice (A)	Max Freq MHz (B)	Worst Case iter. (C)	Slice Delay product (A*C/B)	Power in mW
ALGO-I [8]*	186	54.35	10	34.2	-
ALGO-II [8]*	203	60.80	10	33.2	-
SCALE-FREE[9]*	945	52.54	15	269.85	-
BASE PAPER [3]*	231	58.37	7	27.7	-
PROPOSED	192	130.57	8	11.76	4

## VI. CONCLUSION

The proposed architecture provides a scale-free solution for realizing vector-rotations using CORDIC algorithm technique. The generalized micro rotation selection technique is suggested to reduce the number of iterations for low latency implementation. Moreover, a high speed most-significant-1 detection scheme obviates the complex search algorithms for identifying the micro-rotations. The proposed multi pipelining technique is implemented on CORDIC digital wave generator which shows the better results in terms of speed ,power and area utilization. The design operates on maximum frequency of 130.576MHz.A Considerable increase in speed made the design suitable for many wireless applications like SDR and GSM. In this thesis CORDIC has been implemented in multi staged pipeline architecture in order to avoid iterative cycle that is output obtained at every clock cycle and also proposed CORDIC processor has 11.76 lower slice-delay product and the dynamic power consumption of 4mW on Xilinx Spartan 3 (xc3s200) device.

## Reference

### Journal Papers:

- [1] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., vol. EC-8, pp. 330–334, Sep. 1959.
- [2] J.S Walther. "A Unified Algorithm for Elementary Functions", AIFS Spring Joint Computer Conference, pp.375-385, 1971.
- [3] Supriya Aggarwal, Pramod K. Meher, and Kavita khare, "IEEE transactions on" very large scale integration (vlsi) systems", vol. 20, no. 8, august 2012
- [4] P. K. Meher, J.Walls, T.-B. Juang, K. Sridharan, and K. Maharatna, "50years of CORDIC: Algorithms, architectures and applications," IEEETrans. Circuits Syst. I, Reg. Papers, vol. 56, no. 9, pp. 1893–1907, Sep.2009.
- [5] 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 Trans. Circuits Syst. II, Analog Digit. Signal Process, vol. 50, no. 9, pp. 589–601, Sep.2003.
- [6] M. G. B. Sumanasena, "A scale factor correction scheme for the CORDIC algorithm," IEEE Trans. Comput., vol. 57, no. 8, pp.1148–1152, Aug. 2008.
- [7] J. Villalba, T. Lang, and E. L. Zapata, "Parallel compensation of scale factor for the CORDIC algorithm," J. VLSI Signal Process. Syst., vol.19, no. 3, pp. 227–241, Aug. 1998.
- [8] L. Vachhani, K. Sridharan, and P. K. Meher, "Efficient CORDIC algorithms and architectures for low area and high throughput implementation,"IEEE Trans. Circuit Syst. II, Exp. Briefs, vol. 56, no. 1, pp. 61–65,Jan. 2009.
- [9] K. Maharatna, S. Banerjee, E. Grass, M. Krstic, and A. Troya, "Modified virtually scaling-free adaptive CORDIC rotator algorithm and architecture,"IEEE Trans. Circuits Syst. Video Technol., vol. 11, no. 11,pp. 1463–1474, Nov. 2005.
- [10] F. J. Jaime, M. A. Sanchez, J. Hormigo, J. Villalba, and E. L. Zapata,"Enhanced scaling-free CORDIC," IEEE Trans. Circuits Syst. I, Reg.Papers, vol. 57, no. 7, pp. 1654–1662, Jul. 2010.
- [11] Rajesh Mehra, Bindiya Kamboj., "FPGA Implementation of Pipelined CORDIC Sine Cosine Digital Wave Generator" Int. J. Comp.Tech. Appl, Vol 1 (1), 54-5
- [12] Y. H. Hu and S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," IEEE Trans. Comput., vol. 42, no. 1, pp. 99–102, Jan. 1993.