

ANALYSIS OF AN EFFICIENT CORDIC ARCHITECTURE

Submitted in partial fulfilment of the requirements of the award of
Bachelor of Engineering Degree in Electronics and Communication
Engineering

By

E.SUMA REETHIKA (Register No: 37130122)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600 119.

APRIL 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **E.SUMA REETHIKA (37130122)** who carried out the project entitled “**ANALYSIS OF AN EFFICIENT CORDIC ARCHITECTURE** ” under our supervision from OCTOBER 2020 to MARCH 2021.

Internal Guide

Mr.N.MATHAN, M.Tech.,(Ph.D.,)

Head of the Department

Dr.T.RAVI, M.E.,Ph.D.

Submitted for Viva voce Examination held on

Internal Examiner

External Examiner

DECLARATION

I, **E.SUMA REETHIKA (37130122)** hereby declare that the Project Report entitled **“ANALYSIS OF AN EFFICIENT CORDIC ARCHITECTURE”** done by me under the guidance of **Mr.N.MATHAN, M.Tech.,(Ph.D)** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE : 19/04/2021

PLACE : CHENNAI

1. *E-S-Reethika*

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

I convey our thanks to **Dr. N.M. NANDHITHA, M.E., Ph.D., Dean, School of Electrical And Electronics** and **Dr. T.RAVI, M.E., Ph.D., Head of the Department, Dept. of Electronics And Communication Engineering**, for providing us necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project Guide **Mr.N.MATHAN, M.Tech.,(Ph.D).,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express our thanks to all Teaching and Non-Teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project

ABSTRACT

The CORDIC algorithm also known as the Coordinate rotation digital computer algorithm allows you to calculate the tasks you want in a simple way. The CORDIC algorithm was first introduced by Volder, and Walter later developed it into a cohesive algorithm for performing various transcendental tasks. Due to the simplicity of the functions involved in CORDIC, it is well suited for the implementation of VLSI. One of the most challenging problems is using this algorithm to get low power, high speed and low cost. Power consumption is one of the major concerns of the design of the largest integration circuit and is essential for all applications. In this work we have developed an effective CORDIC algorithm that involves the use of low power. This structure brings performance improvements and opens up various benefits to the use of the VLSI system. We have done several cycles using adders, registers and shifters as well as power analysis of existing and proposed work to determine what is the lowest potential.

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	ABSTRACT	iv
	LIST OF TABLES	viii
	LIST OF FIGURES	ix
1	INTRODUCTION	1
	1 INTRODUCTION	1
	1.1 MOTIVATION	2
	1.2 MULTIPLEXER	3
	1.3 D FLIP FLOP	4
	1.4 SHIFT REGISTERS	5
	1.4.1 UNI DIRECTIONAL SHIFT REGISTERS	5
	1.5 FULL ADDER	6
	1.6 LOOK UP TABLE	7
	1.7 CORDIC ARCHITECTURE	8
2	LITERATURE SURVEY AND OBJECTIVES	12
	2.1 LITERATURE SURVEY	12

2.2	LITERATURE SUMMARY	12
2.3	OBJECTIVE OF THE PROJECT	13
3	EXISTING METHODOLOGY	14
3.1	SOFTWARE IMPLEMENTATION	15
3.2	CORDIC ALGORITHM	17
3.2.1	CORDIC STRUCTURE	20
3.3	REAL IMPLEMENTATION OF HOUGH TRANSFORM	21
3.4	VLSI ARCHITECTURE FOR FEATURE EXTRACTION	23
3.5	NOVEL IMPLEMENTATION OF CORDIC ARCHITECTURE USING BACKWORD ANGLE RECODING	25
3.5.1	BACKWORD CORDIC ANGLE RECODING	26
3.6	A SURVEY OF CORDIC ALGORITHMS FOR FPGA BASED COMPUTERS	27
3.7	MOORE'S LAW	29
3.8	MOSFET	30
3.9	HSPICE	30
3.9.1	HSPICE FEATURES	31
3.10	CORDIC ARCHITECTURE USING 130NM MOSFET	32
3.11	CORDIC ARCHITECTURE USING 32NM MOSFET	33

4	PROPOSED METHOD OF CORDIC ARCHITECTURE	33
4.1	SOFTWARE USED FOR PROPOSED METHOD	34
4.1.2	INSTRUCTIONS FOR HSPICE	34
4.1.2.1	WINDOWS	34
4.1.2.2	CONNECTING FROM HOME	34
4.1.2.3	RUNNING HSPICE	35
5	RESULTS AND DISCUSSION	35
5.1	SIMULATION RESULTS OF PROPOSED CORDIC ARCHITECTURE	35
5.2	RESULT AND DISCUSSION	36
5.3	PERFORMANCE ANALYSIS	39
6	CONCLUSION	41
	REFERENCES	42

LIST OF TABLES

TABLE No.	TITLE	PAGE No.
5.1	PERFORMANCE ANALYSIS OF MOSFET 130NM	42
5.2	PERFORMANCE ANALYSIS OF MOSFET 32NM	43

LIST OF FIGURES

FIGURE No.	TITLE	PAGE No.
1.1	MULTIPLEXER	4
1.2	D FLIP LOP	5
1.3	SHIFT REGISTER	6
1.4	FULL ADDER	7
1.5	LUT	8
1.6	VECTERING MODE CORDIC ARCHITECTURE	10
1.7	CORDIC ARCHITECTURE	11
3.1	CORDIC ARCHITECTURE BASED 2D VECTOR ROTATION	12
3.2	BLOCK DIAGRAM OF CORDIC ARCHITECTURE	13
3.3	EXISTING METHOD OF CORDIC	16
3.4	FPGA BASED CORDIC ARCHITECTURE	18
3.5	SIMULATION RESULTS OF EXISTING METHOD 130NM	32

3.6	SIMULATION RESULTS OF EXISTING METHOD 32NM	33
5.1	SIMULATION RESULTS OF PROPOSED METHOD 130NM	36
5.2	SIMULATION RESULTS OF PROPOSED METHOD 32NM	37
5.3	SIMULATION RESULTS OF SHIFT REGISTER 130NM	38
5.4	SIMULATION RESULTS OF MUX 130NM	39
5.5	SIMULATION RESULTS OF SHIFTER 130NM	40
5.6	SIMULATION RESULTS OF LUT 130NM	41

CHAPTER 1

INTRODUCTION

Advances in advanced integration technology (VLSI) and the advent of design automation tools have guided current research in the areas of DSP, communications and Depending on the speed of construction of real-time algorithms and applications in these areas mentioned above. The development of VLSI technology was predicted by Gordon Moore and since 1965 new technologies developed by the industry include his predicted curve, introduced as Moore's so-called law. These developments have given impetus to developers altering the algorithm into structures. Many DSP algorithms use basic functions such as logarithmic, trigonometric, exponential, division and multiplication. Two ways to do these tasks is to use table view and polynomial expansion. The methods mentioned above require a greater number of duplicates / subdivisions and addition / subtraction.

Complex digital systems use trigonometric functions as a basic component. These functions are widely used in many fields including digital photography and signal processing, cryptography and watermarking. Several methods are designed to calculate trigonometric functions, most of which are based on polynomial or logical measurements that can be easily incorporated into hardware equators.

The integration of Rotation Digital Computer (CORDIC), a special purpose computer program for counting indirect and transient functions, was proposed by Volder in 1959. The functions that can be calculated using a CORDIC computer include trigonometric, logarithmic, exponential, hyperbolic, multiplication, division, square root etc. Although it originally served the purpose of navigation systems, it later became a popular tool for many digital applications. The simplicity of CORDIC is that it can perform any of the above functions using shifts and additions. Many processing and communication systems use CORDIC in a circular communication system or in circulation or voting systems.

At an application level, CORDIC is widely used in the signal processing arena. In CORDIC compensation is implemented instead of using multiplications, helping to significantly reduce hardware complexity. In CORDIC algorithm is included in an efficient filtering design where power of two coefficients are calculated. CORDIC algorithm is applied directly in the implementation of a Givens rotation module used in improving computational time and decreasing complexity. In an adaptive CORDIC rotator with constant scaling factor is proposed aiming to reduce resources usage. This work was supported partially by the Mexican National Council for Science and Technology (CONACYT) through grant number 261243. Researchers have focused on improving the CORDIC algorithmic core and its implementation. Algorithmic improvements commonly consist in reducing the number of rotations modifying the scaling factor or changing the determination of rotations direction. On the other hand, most popular improvements in the hardware arena are related to operational frequency, throughput and occupied area. In this paper, a composed architecture is proposed, it consists in combining a Lookup Table (LUT) with a rotation prediction and a CORDIC pipeline modules. The algorithmic approach takes an input angle and obtains a coarse initial approach of its sine and cosine functions from a LUT. The remaining rotations are algorithmically predicted, and the final result is approximated with the CORDIC pipeline module. Results show a reduction in the number of rotations and in hardware resources usage per iteration.

1.1 MOTIVATIONS

Thus, we can realize that CORDIC is one of the most determining arithmetic techniques that has found far reaching applications in digital systems. Area efficient and power efficient systems are always preferred choice of any designer. Thus, as CORDIC based systems are both area efficient and power efficient, one can design architectures for emerging digital systems based on CORDIC. So, in this work we have designed an efficient architecture with low power consumption.

The CORDIC algorithm also works on square root, Logarithmic, Exponential function and digital computer. In the integration of the trigonometric unit

the functions are very important; Currently most math functions require Sine, Cosine, Tangent etc., and using this algorithm is easy to calculate. With continuous technology and power limits, operating frequency and power consumption, when performing any trigonometry operations using a multiplier, adder, divider, those structures use more hardware and increased computational time, reducing the problem. The CORDIC algorithm converts a Hardware form known as the CORDIC processor. This processor reduces the problem of separation and multiplication.

CORDIC uses simple add-on functions for multiple computer functions such as trigonometric, hyperbolic and logarithmic calculations, real and complex multiplication, division, square root calculation, compatible system resolution, eigen value measurement, single value deviation, override this QR and many others. As a result, CORDIC has been used for applications in a variety of areas.

1.2 MUX

In electronic, a multiplexer (or mux), also known as a data selector, is a device that selects between multiple analog or digital input signals and transmits selected inputs to signal lines. The input multiplexer has selected lines, which are used to select which input line to send to the output.

A multiplexer enables multiple input signals to share a single device or application, for example, a single analog digital converter or one-way communication, rather than one device per input signal. Multiplexer can also be used to perform Boolean operations of many variations.

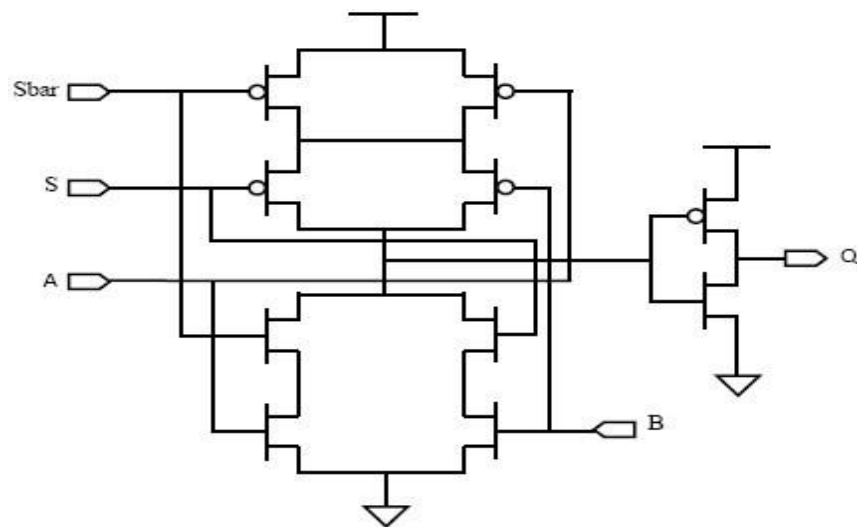


Fig no : 1.1 Multiplexer

1.3 D FLIP FLOP

Electronic, flip-flop or latch is a circuit with two fixed positions that can be used to store data information . The symbols used to control the input may change the state of the circuit. It is the ultimate in sequential logic .

Flip flop delay or D flip flop is a reset flip flop reset with inverter installation to prevent S and R inputs from being at the same level of concept. It always ensures that the inputs S and R are never equal to one at the same time. One data input, d is also used in the set area and the inverter is used to perform reset such as $S = D$ and $R = \text{not } D$.

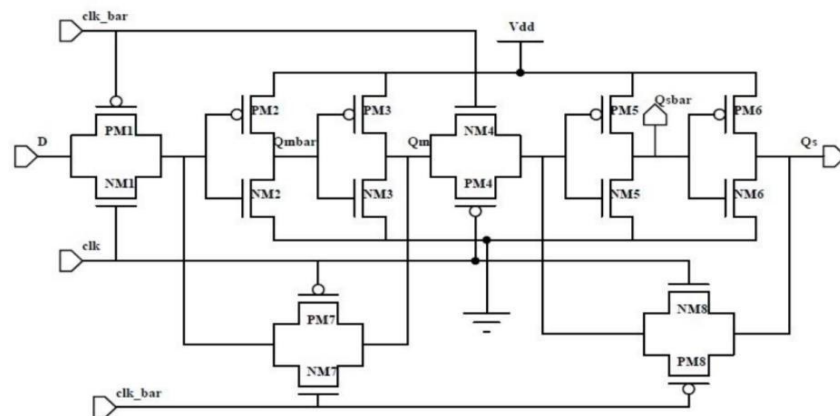


Fig: 1.2 D Flip Flop

1.4 SHIFT REGISTERS

Digital circuit switch switch register using flip flops where single flip-flop output is connected to the next input. They share a single clock signal, which enables data stored in the system to move from one location to another. By connecting the last flip-flop back to the original, the data can be rotated within shifts for a longer period of time, and in this way be used as a form of computer memory. In this role they are very similar to the memory systems of previous delay lines and were widely used in the late 1960s and early 1970s to replace that memory.

In most cases, multiple shift registers will be used to build a large memory pool known as a "bit array". Data was stored on the same members and read back and forth, usually as a computer name, while each component was stored sequentially in shift registers. There is a natural trade in the construction of bit arrays; placing multiple flip-flops in a row allows one shift to save multiple pieces, but it requires additional clock cycles to push data across all shifts before data can be read back.

1.4.1 Uni directional shift register:

These are the simplest types of shift registers. The data cable is introduced in "Data In", and is deleted to the right one section each time "Data Advance" is raised. In each case, the small left-hand switch is replaced by the first flip-flop output. The right portion is removed and lost.

The data is stored on the back of each flip-flop at the "Q" exit, so there are four storage "spaces" available in this setting, which is why it's a 4-bit Register. To give you an idea of the changing pattern, imagine that the register holds 0000 (so all the last spaces are blank). As "Data In" introduces 1,0,1,1,0,0,0,0,0 in the register, this is the result.

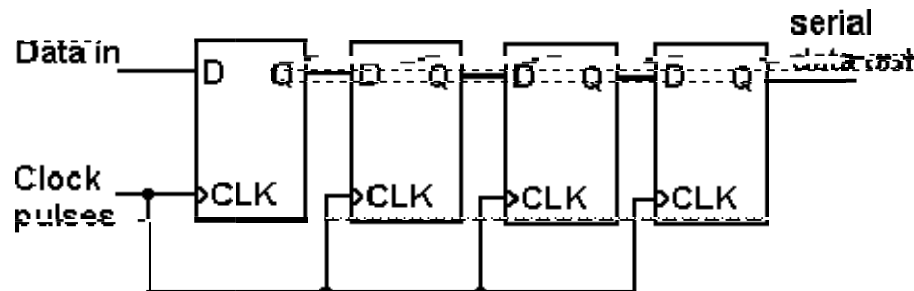


Fig: 1.3 : Uni directional Shift register

1.5 FULL ADDER

In electronics, an adder is a digital circuit that performs addition of numbers. In many computers and other kinds of processors adders are used in the arithmetic logic units or ALU

The full adder adds numbers and binary accounts to the prices carried internally and externally. The full one-bit add-on adds three-bit numbers, which are often spelled like A, B and Cin; A and B are operators, and Cin is loaded slightly from the less important phase. The full size is usually part of the snake cast, adding numbers 8, 16, 32, etc. The region produces two outputs. Output and sum are usually represented by Cout and S symbols, where the sum is equal to $2Cout + S$.

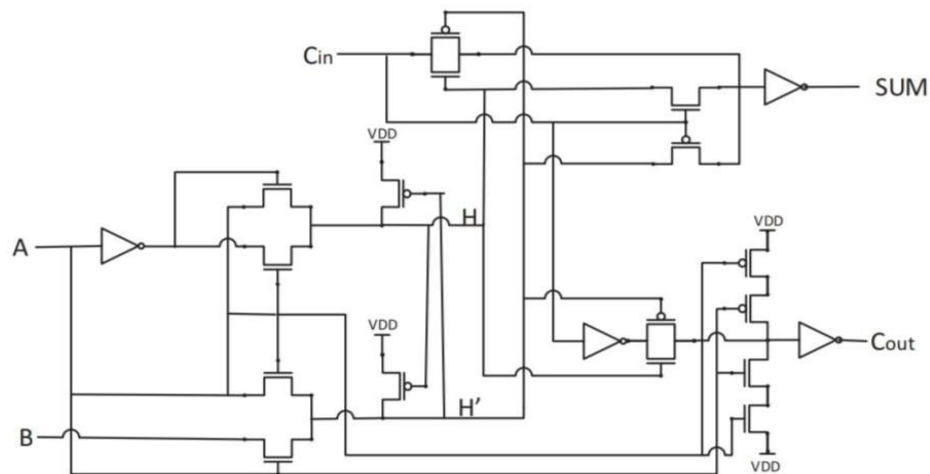


Fig: 1.4 : Full adder

1.6 LUT

A LUT, which stands for Look Up Table, in general terms is basically a table that determines what the output is for any given input or inputs. In the context of

combinational logic, it is the truth table. This truth table effectively defines how your combinatorial logic behaves.

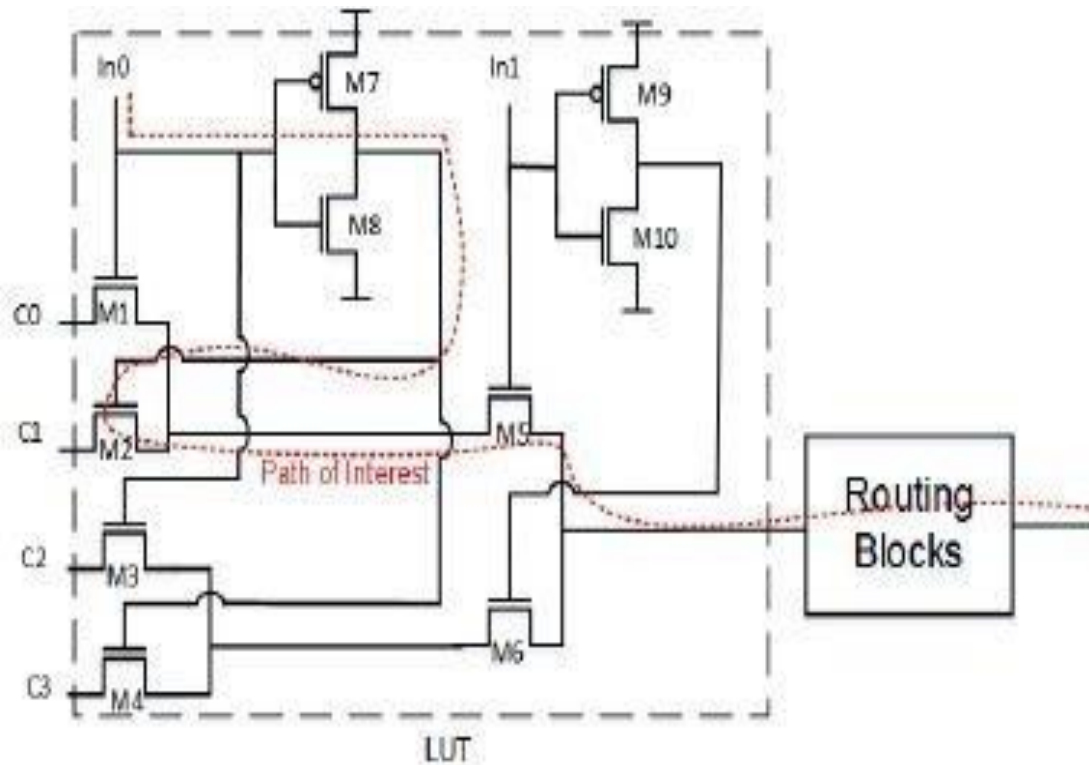


Fig: 1.5 LUT

1.7 CORDIC ARCHITECTURE

The CORDIC algorithm known for linking digital computer algorithm offers the opportunity to calculate the tasks you want in an easy way. The CORDIC algorithm was first introduced by Volder, and Walter later developed it into a cohesive algorithm for performing various extraordinary tasks. The CORDIC region works to calculate several common mathematical functions, such as trigonometric, hyperbolic, logarithmic and object functions.

CORDIC uses adders and shifters, registers to calculate results, with the advantage that it can be implemented using basic Hardware.

Methods such as power series or look up table viewing often require repetition to be performed. If a hardware reporter is not available, CORDIC is usually faster, but if a repeater cannot be used, other methods can be faster. CORDICs can also be implemented in many ways, including a single phase approach, requiring very few gates compared to repeating circuits. Also, CORDICs can calculate multiple functions with the same hardware, so they are suitable for applications with an emphasis on cost reduction (e.g. by reducing gate counts to FPGAs faster. An example of this is the importance of pocket calculators, where CORDIC is widely used.

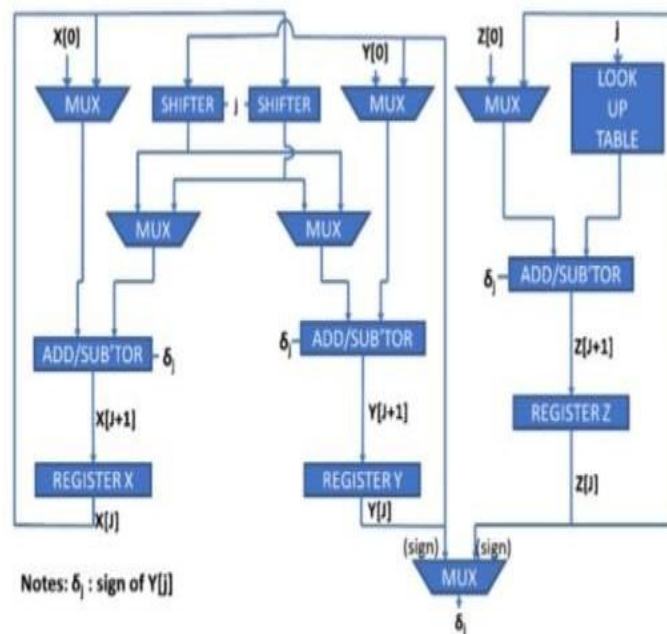


Fig:1.6 Vectoring Mode CORDIC

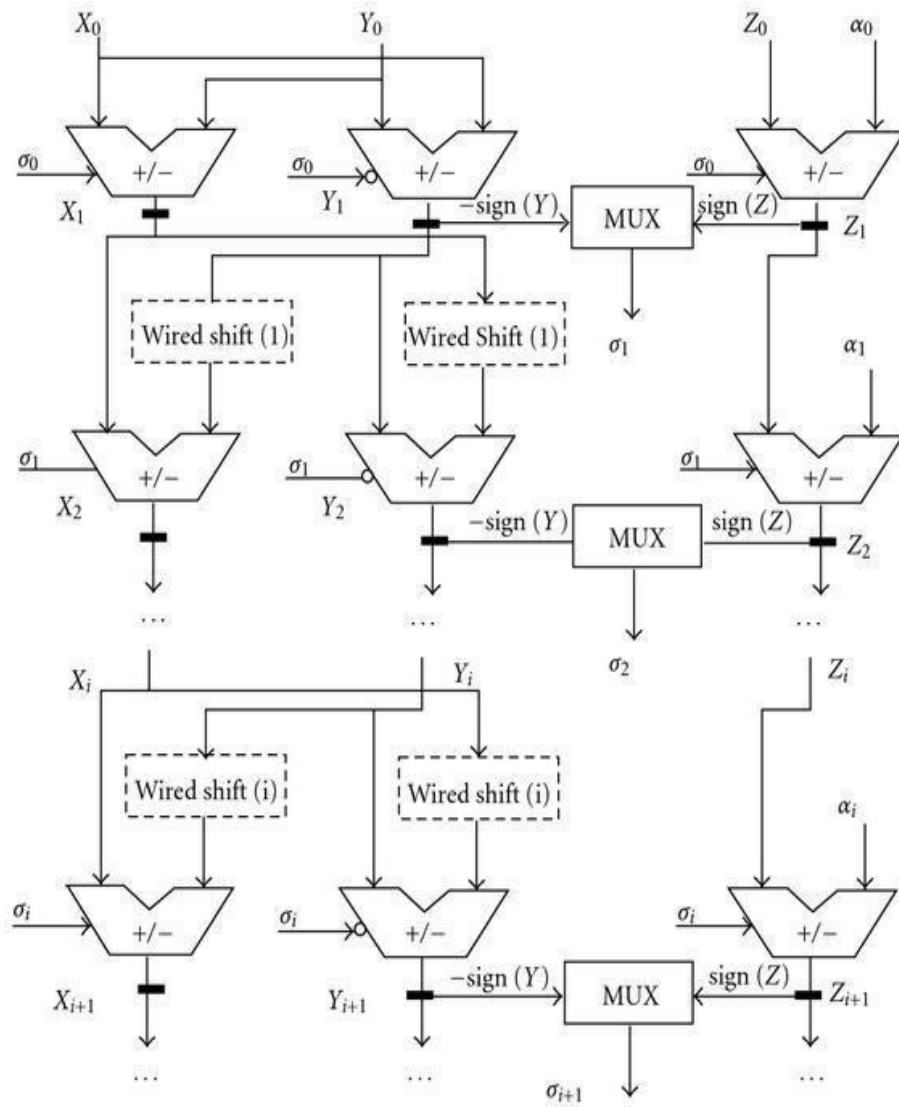


Fig:1.7 CORDIC architecture

CHAPTER 2

2.1 LITERATURE SURVEY

.Arif Sansangko (2019): Recently proposed CORDIC algorithm for feature extraction. In this paper tasks for feature extraction were implemented using vectoring mode CORDIC algorithm. In this paper, hardware level architecture level was implemented for low power and high speed applications without affecting the performance.

Si Mohmoud Karabernou, Lounis Kessal (2005) :

In this paper ,An efficient CORDIC algorithm was implemented for Hough transform and straight lines detection. Methologies are hough transform computations methods, implementation of CORDIC algorithm, CORDIC architecture for FPGA implementation, Real time hough transform using CORDIC.

Ray andraka (2000) :

CORDIC algorithm is used for computing a wide range of functions including mathematical functions. This paper attempts to survey commonly used functions that are accomplished using a CORDIC architecture .

Majority of today's hardware DSP designs are being done by engineer s. The new DSP designers must familiar with these algorithms and techniques for implementing in various applications. This paper shows CORDIC algorithm is available for use in FPGA.

Leena Vachhani, K Sridharan :

This paper presents two area efficient algorithms and their architectures based on CORDIC. While the first algorithm eliminates ROM and requires only low- width barrel shifters, the second eliminates barrel shifters completely. As a consequence, both the

algorithms consume approximately 8% LUTs of a Xilinx sparton device and have a slice delay product of about 3 convergence proofs for the algorithms are presented. Experimental comparisons with prior CORDIC designs confirm the efficiency of the proposed designs.

2.2 LITERATURE SUMMARY

The first CORDIC algorithm was developed in 1959 in which the author proposed a solution to perform trigonometric and rotation functioning, but this method faces many problems such as Hardware required, fixed term and scale factor is also a major problem. in this author discuss other new groups such as log, exponential and square root but this method also faces the same major problems. . Over the past 50 years there has been significant progress in the development of an algorithm to overcome its major challenges. The effectiveness of these methods depends on the chances that the rotation angle is a major problem. Its implementation in relation to Hardware is complex and faces another major problem, the ever-changing scale. This feature reduces the profitability of the delay. To reduce the scale of the scale problem process a complex process is used and the way the extension of the Taylor series also has certain drawbacks. The scope of the Assembly is huge.

The CORDIC system, but this approach is concerned with the low level of integration that provides unsuitable accommodations for active applications, in which the author proposes one new procedure to reduce the scale and number of repetitions.

2.3 OBJECTIVE OF THE PROJECT:

- * To increase the efficiency of the CORDIC architecture for Low Power applications
- * In order to improve Efficiency Low Power Techniques will be applied in all Circuits

- * To compare, the characteristics of MOSFET Technology in terms of Power, Delay and Power Delay Product will be analysed * To identify the Feasibility of CORDIC for feature extraction
- * To improve the efficiency of CORDIC architecture by modifying the digital design for low Power applications. In order to improve Efficiency Low power Techniques will be applied in all circuits
- * To design an efficient CORDIC architecture using shifters ,registers, adders, multiplexers, LUT.

CHAPTER 3

EXISTING METHODOLOGY OF CORDIC ARCHITECTURE

In this age of big data, most of the system is built with the proximity of an intelligent system. The entire program and application needs to be supported by an independent module that can learn and improve its capabilities. The independent module works on four basic concepts: hearing, identification, classification, and learning. In the field of hearing, there are many signals around the world in the form of an image or stream of images (video). They should be extracted from important information through image recognition and image classification. Extensive research has been done to extract this important information using a variety of machines. The device for extracting this information from photo streaming can be used for a wide range of applications such as disaster search, anti-terrorism, intelligent robots, customer analysis [1] [2], etc. The most common problem with that is the use of real-time object detection and recognition.

The biggest problem in detecting an object is the removal of the feature. Many uses are based on principles in the use of digital images that define a unique pattern in an image. Focused gradient histogram (HOG) [3] [4] has become the domain of the most robust feature with low algorithm complexity. Apart from the removal of the robust image feature, another problem is processing the image. Processing digital images usually requires both high-quality resources and long processing time. Many researchers use the Graphical Processing Unit (GPU) to solve this problem [5] [6]. The main goal is to improve the performance of the entire system, especially at the speed of calculation. There are two cases where GPU implementation is inadequate. The first is when a solid (low level) is combined with another required processing material (eg in a chip assembly). Secondly when the distribution of good grain data for a multi-GPU system is required to be uniform. Another problem is the high cost of a GPU where large-scale use of such a system is required. This paper outlines the hardware design and the digital circuit using the HOG design. It can be used in the form of an FPGA or VLSI chip. In addition to system performance and cost, hardware implementation means less form factor and less power consumption [7].

3.1 SOFTWARE IMPLEMENTATION

In some CORDIC system processes the most common system is digital signal processing vector rotation. In Paper [2], the structure of the pipeline structure is described based on a specific CORDIC processor Sine and Cosine calculation process. Due to the construction of its pipelines, the CORDIC design in circular mode provides high system flow by reducing the delay at each level of the pipeline. Saving space on silicon substrate is important in the construction of the CORDIC pipeline, which can be achieved by reducing the amount of rotation. Auto sync function. The auto correlator is used in fourth-generation wireless communication systems to find independent and measure the frequency of a network company. In the OFDM amplifier, CORDIC is best used to measure the offset frequency and determine the ratio of channel counting. In the paper [3], using 130 nm technology, a fast CORDIC structural pipeline and auto correlator were built, reused tested. Simulation is performed on active verification before verification of Verilog HDL.

Digital signal processing (DSP) has been the subject of advances in DSP applications and state-of-the-art technology (VLSI). This has raised many issues with the submission of DSP applications. This implementation will meet the challenges of DSP real-time sample measurements that require less space and energy consumption. Adaptability, high power throughput and low cost are the driving forces behind the development of modern VLSI-based (FPGA / ASIC) communications. The WLAN-based OFDM next-generation technology, which deals with two small and large components, namely autocorrelation of synchronized signal acquisition and CORDIC channel measurement calculation and OFDM wireless receiver compensation. Several components of an electronic circuit, due to various components associated with all circuits, do not receive a clock simultaneously. With the production frequency and stability the section built with loop design is very popular. The authors also discuss in a paper [4] describing a fully-loaded digital loop which is one of the PLL models in which all digital devices are naturally more profitable than FPGA. The overall PLL digital design uses a flexible digital component to operate it and is protected from environmental factors such as heat and parasite power. The use of digital elements has provided sound protection, which improves

circuit accuracy and performance. ADPLL is used for a variety of startups such as wireless communication, device control, biomedical and many other fields, requiring low power, high speed, and less chip area. The designed ADPLL works on a 100 MHz accumulator, 16 bit. This paper [2] discussed the use of the CORDIC algorithm in the DSP vector rotation using the construction of a pipeline that saves space on the silicon stage and also provides high power. In paper [3] the CORDIC algorithm is applied to the OFDM amplifier where it also maintains the location and maximum power which are the main barriers to VLSI technology. Paper [4] provided CORDIC software implementation in areas such as wireless communication, monitoring and environmental use. With these benefits there are also barriers to the use of CORDIC software. For DSP implementation, the algorithm must follow a set sample size and should also save space and power as well. Therefore making this FPGA a reality has been introduced to reduce costs and energy consumption.

3.2 CORDIC ALGORITHM

CORDIC's algorithmic approach performs vector rotations by arbitrary angles using shifts and additions. The algorithm is based on a general rotation transformation with angle restricted to $\tan \theta = \pm 2^{-i}$, reducing multiplications to shift operations. Thus, arbitrary angles are obtained while applying a series of micro-rotations. Basic CORDIC equations are shown in (1) and (2):

$$\theta_{i+1} = \theta_i (\theta_i - \theta_i \theta_i 2^{-i}) \quad (1)$$

$$\theta_{i+1} = (\theta_i + \theta_1 \cdot \theta_i \cdot 2^{-i}) \quad (2)$$

Where $\theta_i = 1 / \sqrt{1 + 2^{-2i}}$ and $\theta_i = \pm 1$ approaches a constant and can be applied

at any stage of the process. The necessary micro-rotations to calculate sine and cosine functions are defined by a sequence of directions represented by a decision

vector. A set of decision vectors can be stored in a LUT or can be integrated into the system through an extra adder-subtractor that accumulates rotation angles at every iteration. Having an angle accumulator requires the extra equation (3).

$$\theta_{n+1} = \theta_n - \theta_{\text{in}} \cdot \text{sgn}(x_n) \quad (3)$$

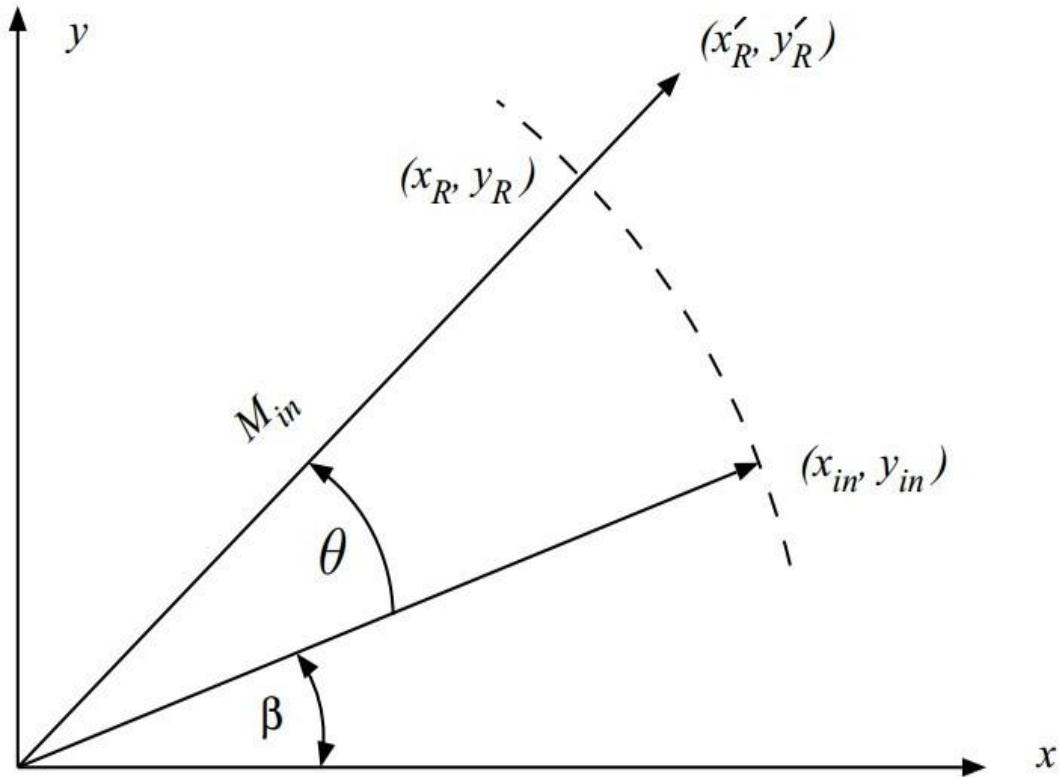


Fig:3.1 CORDIC algorithm based 2D vector rotation

This equation is eliminated in the proposed architecture by pre-calculating rotations directions.

The basic principle of the CORDIC algorithm is based on two-dimensional geometry. This algorithm works internally or alternately or in vectoring mode, following straight, circular or hyperbolic lines. We focus on the rotation mode on the operation of the circular trajectory. A. The standard CORDIC Algorithm Enable vector V_a $[X_a, Y_a]$ to be detected by rotating the vector V_b $[X_b, Y_b]$ at an θ angle, then:

$$X_{n+1} = X_n - Y_n \cdot \text{sgn}(X_n) \quad (1)$$

$$[X \ Y] = R_p [X \ Y], R_p = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

Equation (1) forms the basic principle of calculation of iterative coordination in the CORDIC algorithm [1]. The key to detecting rotation is using CORDIC to produce a rotating angle θ as a combination of pre-defined angles defined as: when b the length of the voice in bits. (2)

$$\theta = \sum_{i=0}^b \theta_i * U_i \quad (2)$$

Where $U_i = -1, 1; a_i = \tan^{-1} 2^{-i}$

The standard CORDIC RoC is $[-99.9 \text{ deg}, 99.9 \text{ deg}]$ and the use of an additional retrieval step can be extended across the entire spacing. The rotation matrix R_p , in its original form is very powerful by calculation; requires sine and cosine computer functions with four repetition tasks and two additional functions. Making the truth with the cosine name simplifies the rotation of the matrix R_p (1) by changing the multiplication to change the performance, as low-angle rotation is defined in two negative forces (2). But the penalty paid for the introduction of the scale-element varies depending on the cosine of the elementary-rotation

The well-designed hardware design for SINE production and As seen in (3), the Ki scale is independent of COSINE waves based on the CORDIC (Coordinate Rotation Digital Computer) algorithm. [16]. This suggests a novel approach, in which the author uses a single lead-in-finding process to identify small rotations. This process removes the complex search algorithm. The free design of the algorithm is based on the expansion of the Taylor series of sine and cosine waves. This proposed algorithm reaches the fourth order of the Taylor series. Micro-Rotation Sequence Generation, in previous operation angle rotation ROM is required but with the use of Micro Rotation Sequence Generation no ROM is required to maintain basic rotation angles. the proposed algorithm also faces many problems such as increased error, iteration rate increases, reducing this problem is another proposed method [17]. In this way the author uses the same process as [16] but converts the structure into the same form and suggests that the use of the CORDIC parallel rotator be significantly improved for high performance at low cost in an area that uses less rotating direction.

By subsequent performance of a large number of recurring trends, referred to as the acquisition of the CORDIC algorithm. The scale is therefore compensated for the post or pre-operation of the unit.

$$R_p = K_i \cdot [1 - \sum_{i=0}^{N-1} 2^{-i} \cdot \cos \theta_i] U_i \cdot 2^N$$

The non-standard CORDIC [12] was the first attempt to create a non-standard CORDIC scale using the extension of the Taylor series. Here, small rotation is limited to anticlockwise direction only, i.e., any rotation angle is represented as the algebraic total of the lower angles. Sine and cosine activities are limited to:

$$\sin \theta_i = 2^{-i}, \cos \theta_i = 1 - 2^{-2i+1}$$

3.2.1 CORDIC STRUCTURE

A graphical representation of the algorithmic flow is shown in figure (3). For N bits output resolution, CORDIC structure consists of a cascade of N butterflies. Each one implements a positive or negative sub-rotation by a fixed angle. The algorithm restricts angle rotation to $k \cdot \theta = 2 \text{ rad}$.

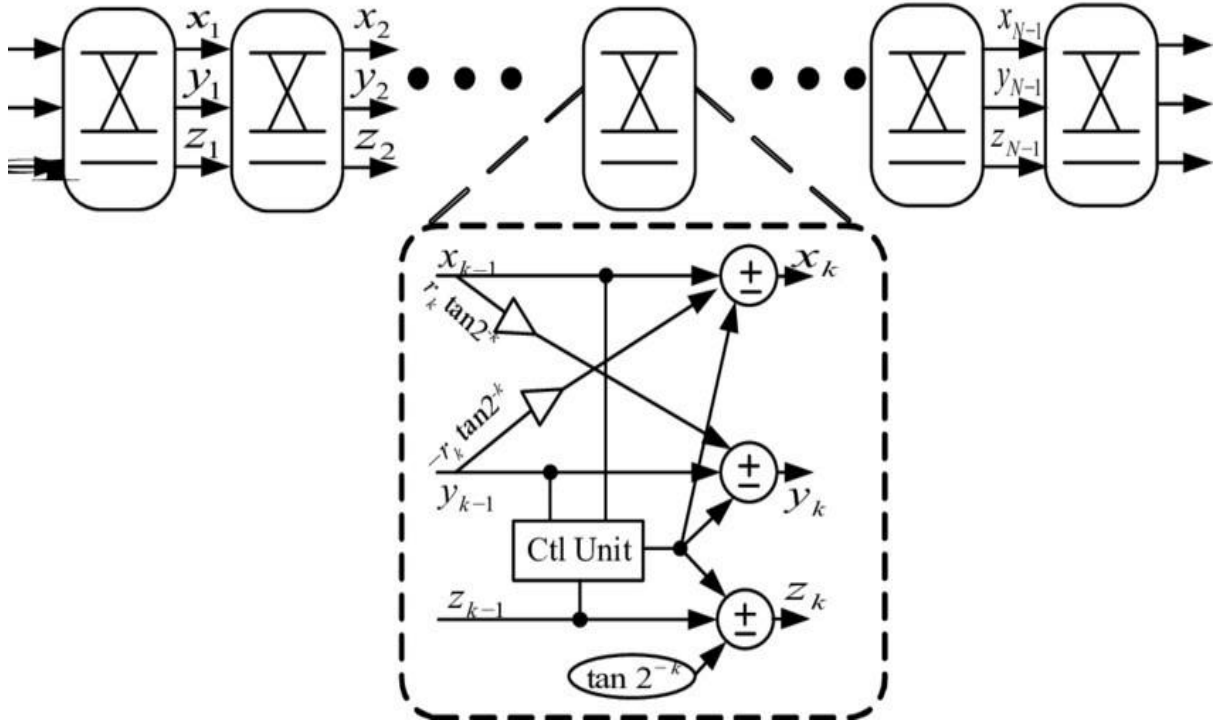


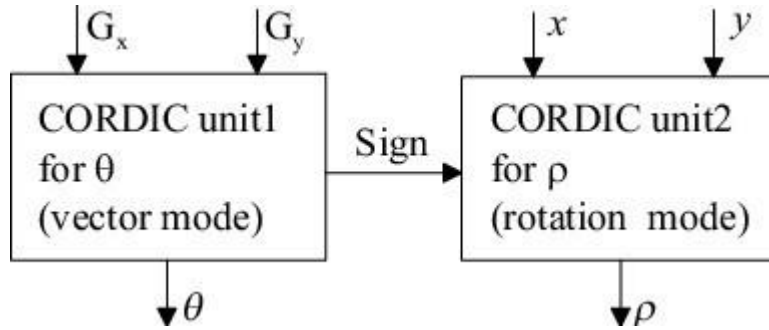
Fig:3.2 Block diagram of CORDIC architecture

3.3 Real Implementation of Hough Transform using CORDIC Algorithm

To describes a real-time architecture for Hough Transform based on CORDIC algorithm and gradient for straight lines detection. The CORDIC algorithm allows the use of simple operators such as adders and shift registers. The proposed real-time architecture can be easily fitted into FPGA devices since the present performance increase of this technology allow the implementation of complex applications while real-time constraints for Hough Transform based lines detection are respected for most of the video transmission Standards.

In order to meet real-time constraints, it is important to define the target of the image/video specification [18]. When conventional TV and some MPEG standards are

considered, the proposed architecture must be able to reach a processing rate of 25 frames per second. The processing should include parameter space calculations as well as straight-lines image building. 720x576 pixels frame must be processed within 40 milliseconds. Therefore, the processing of a single pixel should not exceed 88,25 Nanoseconds (a minimum frequency of 11,3 MHz). The proposed global architecture is shown in Figure 7. Two CORDIC units are used to simultaneously compute ρ and θ . The two units use different modes. This first block is shown in Figure 8. Details of the design



As θ_k is a power of two ($k - \theta = 2$), the implementation of $\theta_k \tan$ can be simplified by employing the approximation $\theta_k \approx \theta_k \tan$ for sufficiently small θ_k (i.e., for sufficiently large k), [13]. Hence, all multiplication operations are reduced to simple shift operations.

The idea behind using two memories is to make, the space parameter computations and straight lines image building process, to be performed in parallel. While RAM1 is used for current image vote storage (Hough parameter space fill), RAM2 which contains the space parameter of the previous image, is read in order to extract straight lines parameters (ρ and θ) to build the output image. The two memories are swapped at the end of each image. The address bus of the parameter space memory is obviously the concatenation of ρ and θ .

3.4 VLSI Architecture for Feature Extraction

Histogram of Oriented Gradient (HOG) Principally, this method is based on the idea that each object in captured image has the unique feature. The unique feature is the light reflected by the object. In the other word, each object has certain shape and

pattern such that could be classified based on the collection of some gradients of curvature. Then, the curvature is represented statistically so that they can be extracted into certain distribution pattern. This distribution pattern is collected in form of a histogram. This histogram is optimized to be learned by the computing system [10]. This method consists of some processing steps including

1. Gamma Normalization
2. Gradient Computation
3. CORDIC implementation
4. Contrast Normalization per spatial block
5. Collect HOG Feature
6. Classifier

The item no 6 (classifier) is not part of the feature extraction part. Some alternatives can be used for classifying the object based on the extracted feature such as SVM or CNN.

In the SoC acquisition process, two processors (hardware module) are required, the first one to extract the element, and the second one to be separated as shown in Figure 1. This paper focuses on the output processor that performs the HOG calculation. Contains gamma typical, gradient calculation, gradient phase division and calculation, and standard comparisons. The most popular field to support the growth of a discovery-based program. The basis for the acquisition of an object is the removal of a feature. This paper suggests the creation of a novel hardware release feature based on the HOG.

Hardware-based HOG processor used to extract the feature using the line scanning strategy. The newness of this processor (1) is the use of a good grain piping scheme and (2) they propose to use the Barni method of Histogram Normalization,. The result of doing good is good with less than 12% error. This error does not affect

the apparent difference in item acquisition. Besides, this technique works well with 16 clock latency of the tested image solution due to a good grain pipeline scheme. The main frame tested is 640x480. Compared to a faster HOG using GPU implementation [12], this implementation is better. Our proposed design can reach 63 fps at 50MHz or about 126 fps with a 100 MHz clock. About 1.5 times more than the GPU. This structure is also compared with previous work in the FPGA implementation of the HOG Processor [14]. Limited on time memory usage and delays. In the form of a pipeline, this current project has a refined pipeline. On the other hand, the previous work has only two phases. It means that this technology will show a significant reduction in processing latency than ever before. The task of the future is to use this program to build a adoption system. This module should be integrated with a processor classifier such as SVM direct. With this structure, the human detection process can reach up to 63 fps.

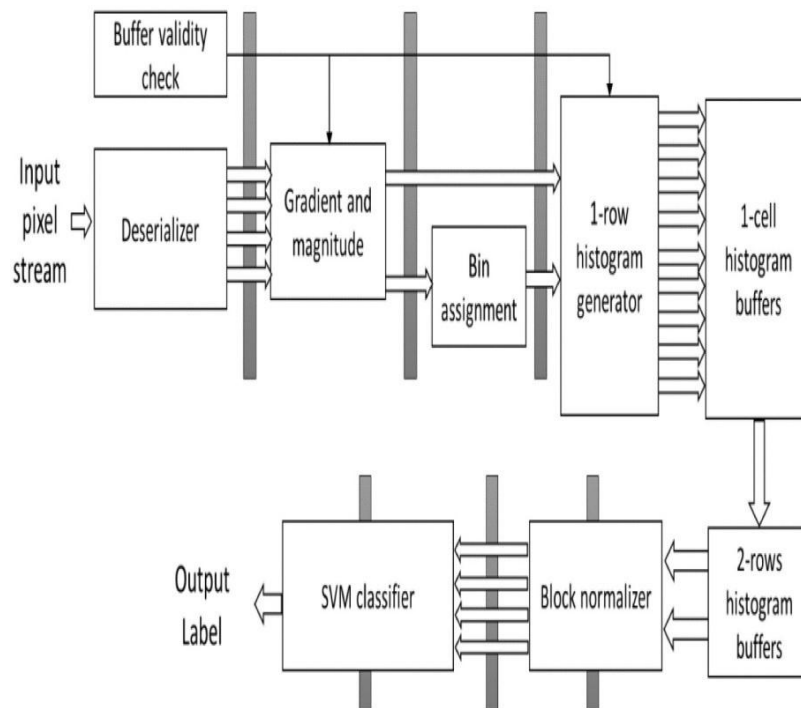


Fig:3.3 Existing Method

3.5 NOVEL IMPLEMENTATION OF CORDIC ALGORITHM USING BACKWORD ANGLE RECODING

To propose a backward angle recoding (BAR) method to eliminate redundant CORDIC elementary rotations and hence expedite the CORDIC rotation computation. It proves that for each of 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 .

3.5.1 BACKWARD CORDIC ANGLE RECODING

1) Knowing that $\alpha_k = \alpha_{k-1} \alpha_k \pm 1$ can be simplified in actual implementation. However, α_k still needs to be computed explicitly because in the implementation of a systolic array of CORDIC processors backward rotation will be performed only in the head processor, and the remaining CORDIC processors will perform forward rotations based on $p(k)$ and α_k computed in the head processor. can be realized with the usual mantissa normalization unit found in a normalized floating point arithmetic unit. It requires the detection of the number of leading zeros (or leading ones if the mantissa is in 2s complement representation, and is negative), and the decoding of it to a binary number. Because the use of mantissa and exponent, it makes the BAR algorithm very suitable for normalized floating point CORDIC operations. We note that with fixed point to floating point format conversion, it will be equally useful for fixed point arithmetic as well. Because $1/2 \leq \alpha_k \leq 1$, the convergence condition $|\alpha_k| < 1$ can

be verified by an equivalent condition $b' + E, > 0$ where $E = 2^{-b'}$.

A Numerical Example

To illustrate the proposed BAR algorithm, let us consider the following numerical example: Suppose $x(0) = 0.10011010$, $x \cdot 2^{-2'}$, and $y(0) =$

0.11001101×2^{-1} , and $m = 1$ (circular rotation mode). Here we use 8 bits for

mantissa and 4 bits for exponent (both excluding sign bits).

Then, we have $\square_0 = (0) - (0) = 0 - (-2) = 2$. Since $My(0) > 3/4$, we compute $q = 3Mx - 2My = 0.00110100$, > 0 . Hence $\square_0 = \square_0 = 2$. Next, $\square_1$, we compute

$$\tilde{\square}(1) = \square \square \cdot (0) + 2^{2-2} \cdot \square \square(0) = 0.11001101$$

$$\tilde{\square}(1) = \square \square \cdot (0) + 2^{2-2} \cdot \square \square(0) = 0.00100110$$

, For normalization, one has $Ax = 0$, and $4 = -2$. This leads to new exponents $EJ1) = 0$, and $Ey(1) = 4$. Also, $Mx(1) = fix(1)$, $My(1) = 0.11001100$,. One may verify that in the remaining of this iteration, $z1 > 0$, and $Mx(2) = 0.10100111z$, $M,(2) = 0.10011000$,. Also, $Ex(2) = 0$, and $E,(2) = -6$. In this iteration, since $M,(2) < 3/4$, we calculate and find

$T,, = -.011010112$. Hence, we still have $S2 = i2$. Finally, since $83 = 6$,

3.6 A SURVEY OF CORDIC ALGORITHMS FOR FPGA BASED COMPUTERS

The state of the digital processing landscape was dominated by microprocessors with enhancements such as single-cycle expanding commands and specialized handling methods. While these processors are inexpensive and offer extremely flexible performance, they are often not fast enough to get the DSP functions they really need. The advent of unparalleled logic computers allows for high-speed hardware solutions that offer cost-effective competition to the traditional software approach. Unfortunately, algorithms designed for these microprocessor-supported systems tend to mark well on hardware.

While hardware-efficient solutions often exist, the dominance of the software systems has kept those solutions out of the spotlight. Among these hardware-efficient algorithms is a class of iterative solutions for trigonometric and other transcendental functions that use only shifts and adds to perform. The trigonometric functions are based on vector rotations, while other functions such as square root are implemented using an incremental expression of the desired function. The trigonometric algorithm is called CORDIC, an acronym for COordinate Rotation Digital Computer. The incremental functions are performed with a very simple extension to the hardware

architecture, and while not CORDIC in the strict sense, are often included because of the close similarity. The CORDIC algorithms generally produce one additional bit of accuracy for each iteration. The trigonometric CORDIC algorithms were originally developed as a digital solution for real-time navigation problems. The original work is credited to Jack Volder [4,9]. Extensions to the CORDIC theory based on work by John Walther[1] and others provide solutions to a broader class of functions. The CORDIC algorithm has found its way into diverse applications including the 8087 math coprocessor[7], the HP-35 calculator, radar signal processors[3] and robotics. CORDIC rotation has also been proposed for computing Discrete Fourier[4], Discrete Cosine[4], Discrete Hartley[10] and Chirp-Z [9] transforms, filtering[4], Singular Value Decomposition[14], and solving linear systems[1]. CORDIC and CORDIC-like algorithms with an eye toward implementation in Field Programmable Gate Arrays (FPGAs). First a brief description of the theory behind the algorithm and the derivation of several functions is presented. Then the theory is extended to the so-called unified CORDIC algorithms, after which implementation of FPGA CORDIC processors is discussed.

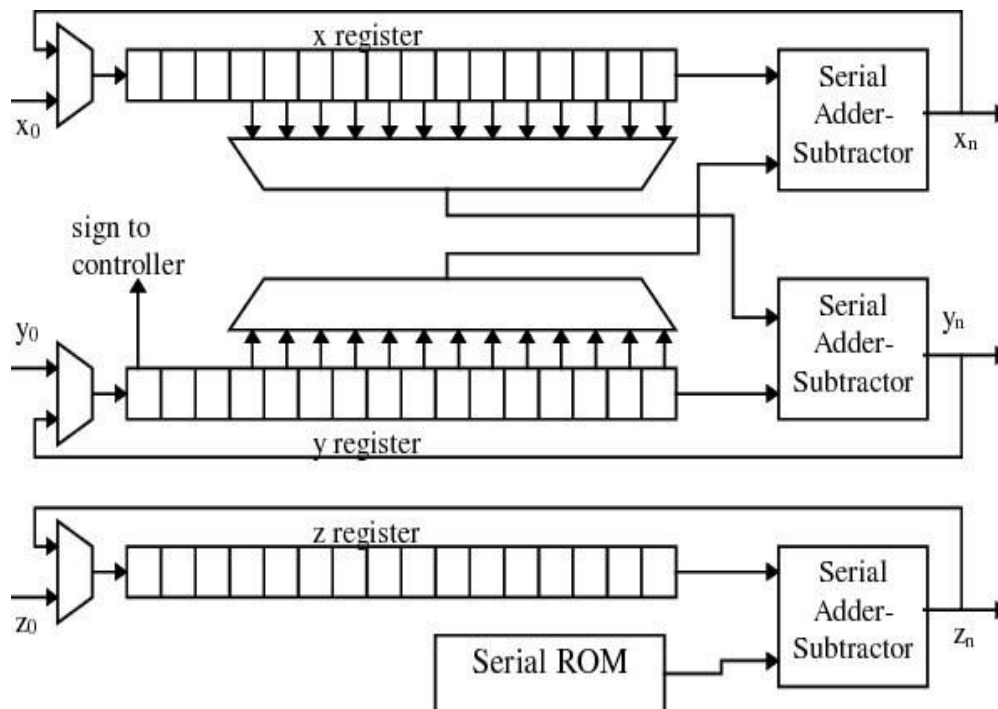


Fig:3.4 cordic algorithms for fpga based computers

3.7 MOORE'S LAW

Moore's Law refers to Moore's perception that the number of transistors on a microchip doubles every two years, though the cost of computers is halved. Moore's Law states that we can expect the speed and capability of our computers to increase every couple of years, and we will pay less for them. So far, **Moore's Law** has been proven correct, time and again, and as a result it has long been said to be responsible for most of the advances in the digital age – from PCs to supercomputers – due to it being used in the semiconductor industry to guide long-term planning and set targets for research and development.

3.8 MOSFET

The MOSFET (Metal Oxide Semiconductor Field Effect Transistor) transistor is a semiconductor device that is widely used for switching purposes and for the amplification of electronic signals in electronic devices. A MOSFET is either a core or integrated circuit where it is designed and fabricated in a single chip because the device is available in very small sizes. The introduction of the MOSFET device has brought a change in the domain of switching in electronics. Let us go with a detailed explanation of this concept.

A MOSFET is a four-terminal device having source(S), gate (G), drain (D) and body (B) terminals. In general, The body of the MOSFET is in connection with the source terminal thus forming a three-terminal device such as a field-effect transistor. MOSFET is generally considered as a transistor and employed in both the analog and digital circuits. This is the basic introduction to MOSFET.

- Amplifiers made of MOSFET are extremely employed in extensive frequency applications
- The regulation for DC motors are provided by these devices
- As because these have enhanced switching speeds, it acts as perfect for the construction of chopper amplifiers
- Functions as a passive component for various electronic elements.

In the end, it can be concluded that the transistor requires current whereas MOSFET requires a voltage. The driving requirement for the MOSFET is much better, much simpler as compared to a BJT.

3.9 HSPICE

- Simulation Program with Integrated Circuit Emphasis
- General purpose analog circuit simulator
- Used in IC and board-level design for check of integrity of circuit designs and prediction of circuit behavior

- Developed at Electronics Research Laboratory of the University of California, Berkeley
- SPICE simulation is industry-standard for verification of circuit operation at transistor level before manufacturing
- Description of circuit elements (transistors, resistors, capacitors ,etc.) and connections

3.9.1 HSPICE FEATURES

- Superior convergence
- Accurate modeling, including many foundry models
- Hierarchical node naming and reference
- Circuit optimization Circuit optimization for models and cells with for models and cells, with incremental or simultaneous Multiparameter
- optimizations in AC, DC, and transient simulations
- Monte Carlo and worst-case design support
- Input, output, and behavioral algebraics for cells with parameters
- Cell characterization tools to characterize standard cell libraries
- Geometric lossy-coupled transmission lines for PCB.

3.10 CORDIC ARCHITECTURE USING 130NM MOSFET

Technology Node defines the features of any device. If we say 130nm node then it means that device has a 130 nm gate length. As per the nomenclature every Node is decided by the ITRS road map. To meet Moore's law size of transistors should be reduced. And to achieve that every new Node technology has been refunded by the half pitch.

Half pitch is nothing but the area of the transistors should be half of the previous Node. And to achieve that scaling of the transistors is necessary. There are

two types of scaling proposed. One was constant voltage and other one was constant filed scaling.

If we reduce the width and height of the transistors by 0.7 then area of the transistor will be half of the previous one.

And if the area is half of one transistor then then we can place double transistor on same place and like wise our density of transistors per area will be doubled, as Moore's law say it will be doubled in every 18 months.

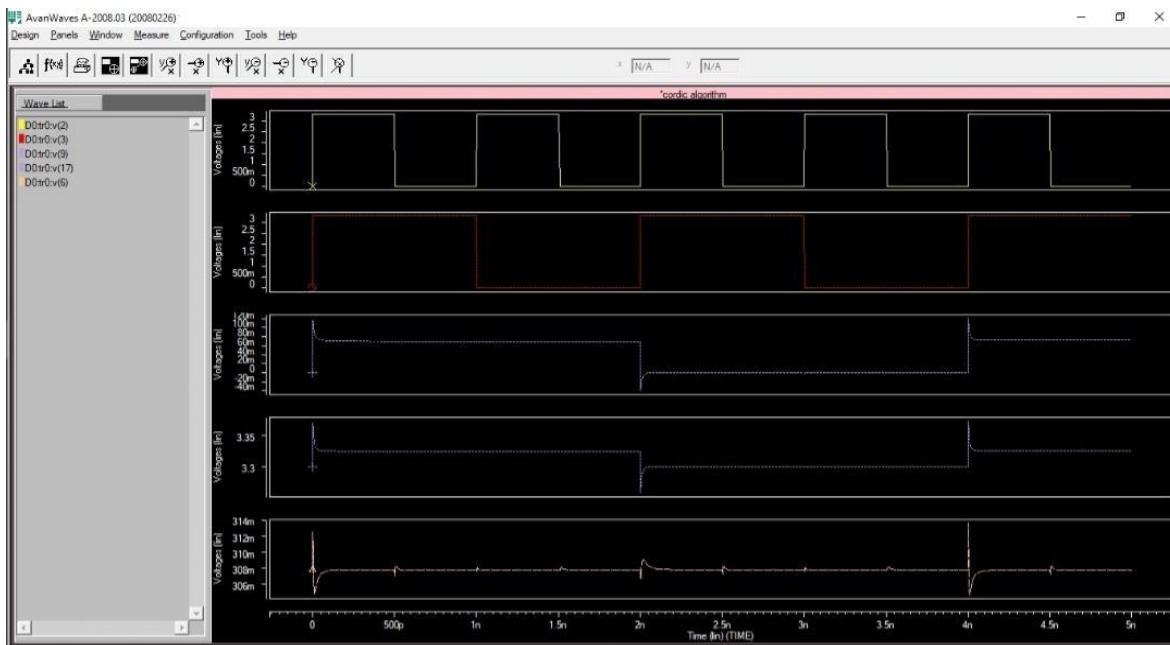


FIG 3.5 : SIMULATION WAVE FORM OF EXISTING CORDIC ARCHITECTURE(130NM)

3.11 CORDIC ARCHITECTURE USING 32NM MOSFET

The 32nm node is the step following the 45nm process in CMOS (MOSFET) semiconductor device fabrication. "32nm" refers to the average half-pitch (i.e., half the distance between identical features) of a memory cell at this technology level. Toshiba produced commercial 32 GiB nand flash memory chips with the 32 nm process in 2009. INTEL and AMD produced commercial microchips using the 32nanometre process in the early 2010s. IBM and the common platform also

developed a 32 nm high k- metal gate process. Intel began selling its first 32 nm processors using the Westmere architecture on 7 January 2010.

The 28-nanometre node was an intermediate half-node die shrink based on the 32-nanometre process.

The 32 nm process was superseded by commercial 22nm technology in 2012.

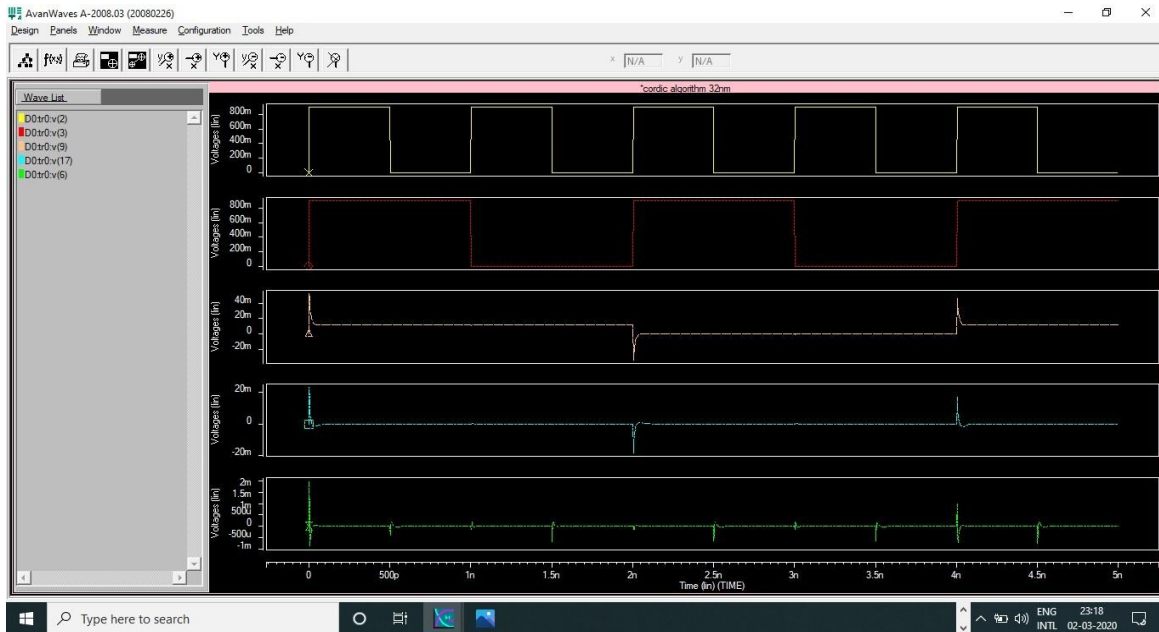


Fig 3.6 : SIMULATION WAVE FORM OF EXISTING CORDIC ARCHITECTURE(32NM)

CHAPTER 4

PROPOSED METHOD OF CORDIC ARCHITECTURE

In the proposed method of the CORDIC architecture we have used a software called hspice .In that hspice we has worked on 130nm and 32nm using MOSFET we did a work on low power efficiency of CORDIC architecture and taken parameters like average power, minimum delay, power product delay, rise time, fall time by taking all these parameters and compared with the existing method .

4.1 SOFTWARE USED FOR PROPOSED METHOD HSPICE:

- Superior convergence
- Accurate modeling, including many foundry models
- Hierarchical node naming and reference
- Circuit optimization Circuit optimization for models and cells with for models and cells, with incremental or simultaneous Multi parameter
- optimizations in AC, DC, and transient simulations
- Monte Carlo and worst-case design support .
- Input, output, and behavioral algebraics for cells with parameters
- Cell characterization tools to characterize standard cell libraries
- Geometric lossy-coupled transmission lines for PCB.

4.1.2 INSTRUCTIONS FOR HSPICE:

4.1.2.1 WINDOWS

If you're working in the lab, you can skip the "Connecting From Home" portion of this section. Just login to a computer and jump directly into the section on "Running HSPICE".

4.1.2.2 CONNECTING FROM HOME

In order to use HSPICE from home in Windows, you will need to have Windows XP Professional or a newer version of Windows installed.

This is because running HSPICE requires the use of a feature called Remote Desktop Connection, which is only available on Windows XP Professional and later versions of Windows.

Follow these instructions to run Remote Desktop Connection:

1. Click Start → All Programs → Accessories → Communications → Remote Desktop Connection. Remote Desktop Connection will start.
2. Click Options >> → Local Resources. Check the box at the bottom that says “Disk Drives”. Click Options <<.
3. In the “Computer” box in Remote Desktop Connection, type: iserver1.eecs.berkeley.edu. Click Connect.
4. Login to your EE105 instructional account (your named account will also work here). The remote Windows desktop will start and you will appear to be working on a lab machine remotely.

4.1.2.3 RUNNING HSPICE

We’re going to go through a sample HSPICE simulation and analysis in Awaves in order to teach you how to run these programs.

1. Start Notepad. You can do this either by clicking Start → Run, typing “Notepad”, and hitting Enter. Or, you can click Start → Programs → Accessories → Notepad.

In the cordic architecture there are used mainly is Dflipflop for the modification of the desired parameter. To acquire the low power and efficient architecture in the proposed method, we have nullified cmos nodes in D flip flop. This improved entire architecture efficiency. We simulated code for the whole architecture and took the readings and simulation results and after wrote the code for modified D-flipflop and took the readings and simulation graphs for the 130nm MOSFET and 32nm MOSFET. .

CHAPTER 5

RESULTS AND DISCUSSION

5.1 SIMULATION RESULTS OF PROPOSED CORDIC ARCHITECTURE



Fig 5.1 : Simulation waveform of proposed cordic architecture (130nm)

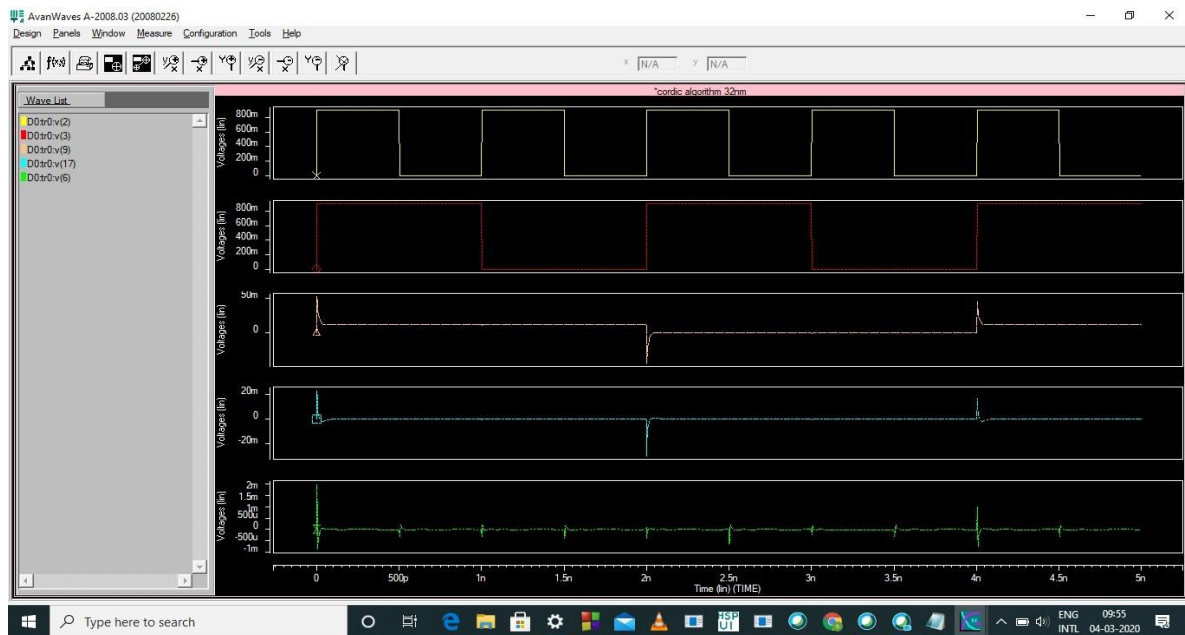


Fig 5.2 : Simulation waveform of proposed cordic architecture (32nm)

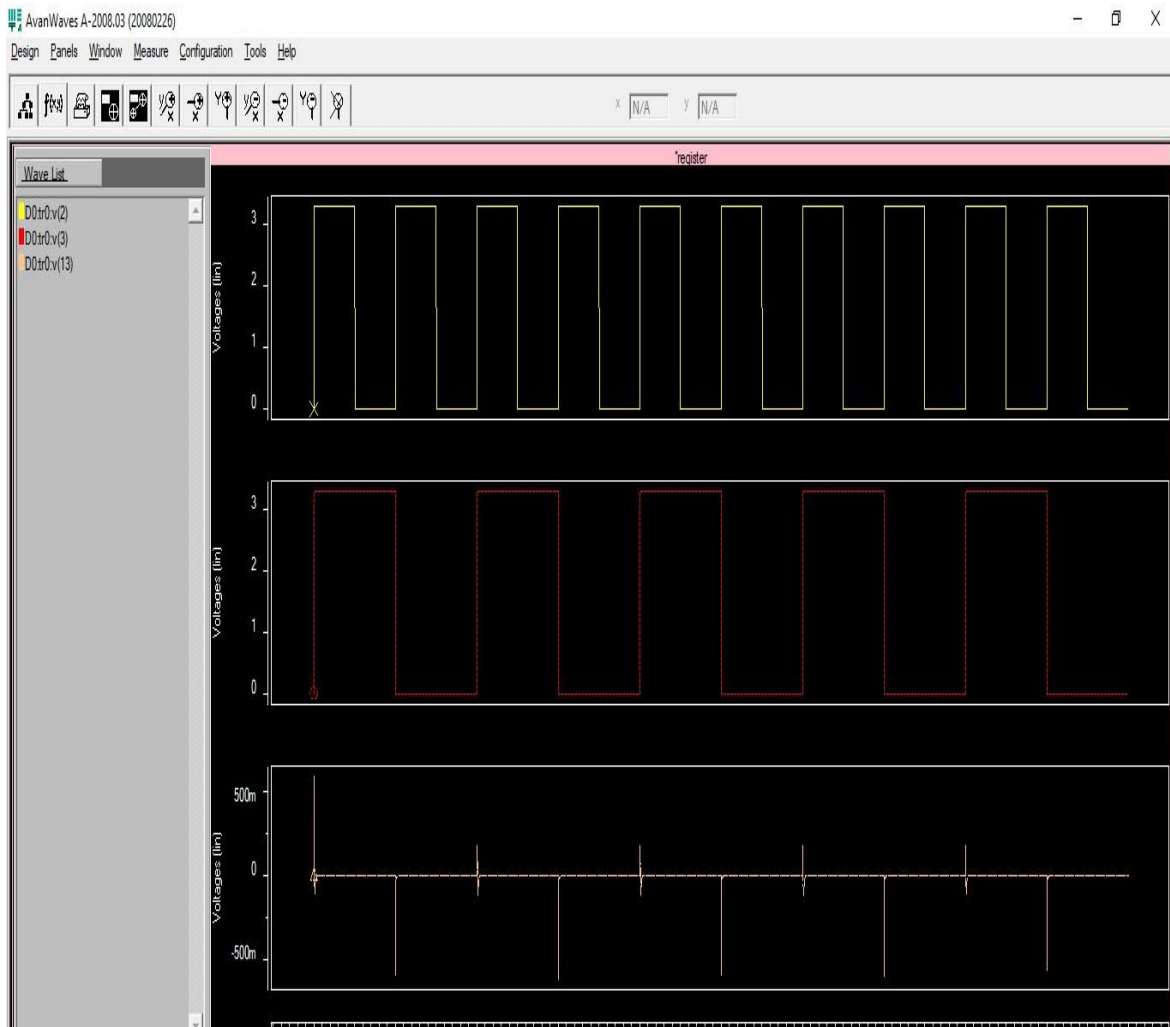


Fig 5.3 :Simulation Output For Register Using 180nm Mosfet

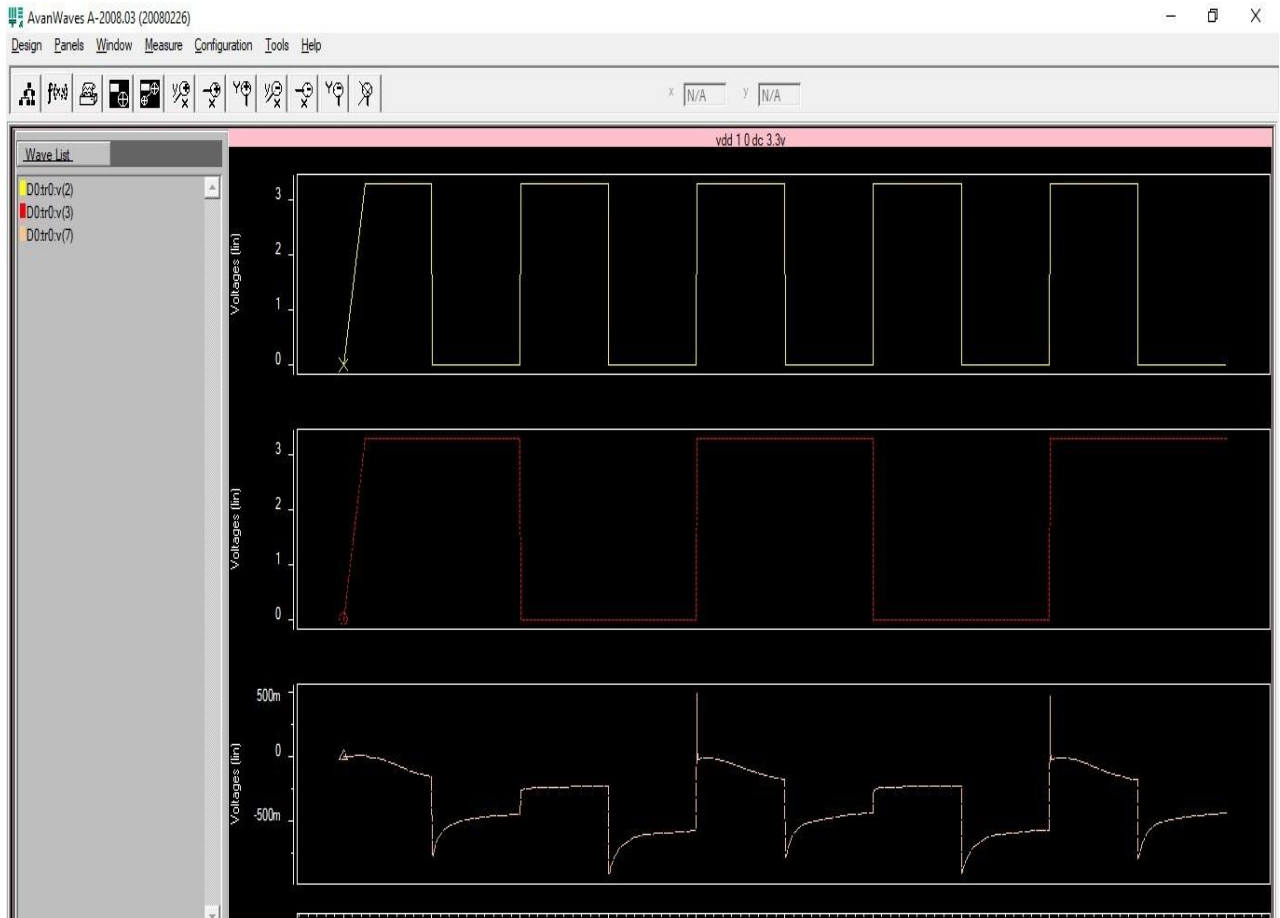


Fig 5.4:Simulation Output For Mux Using 180nm.

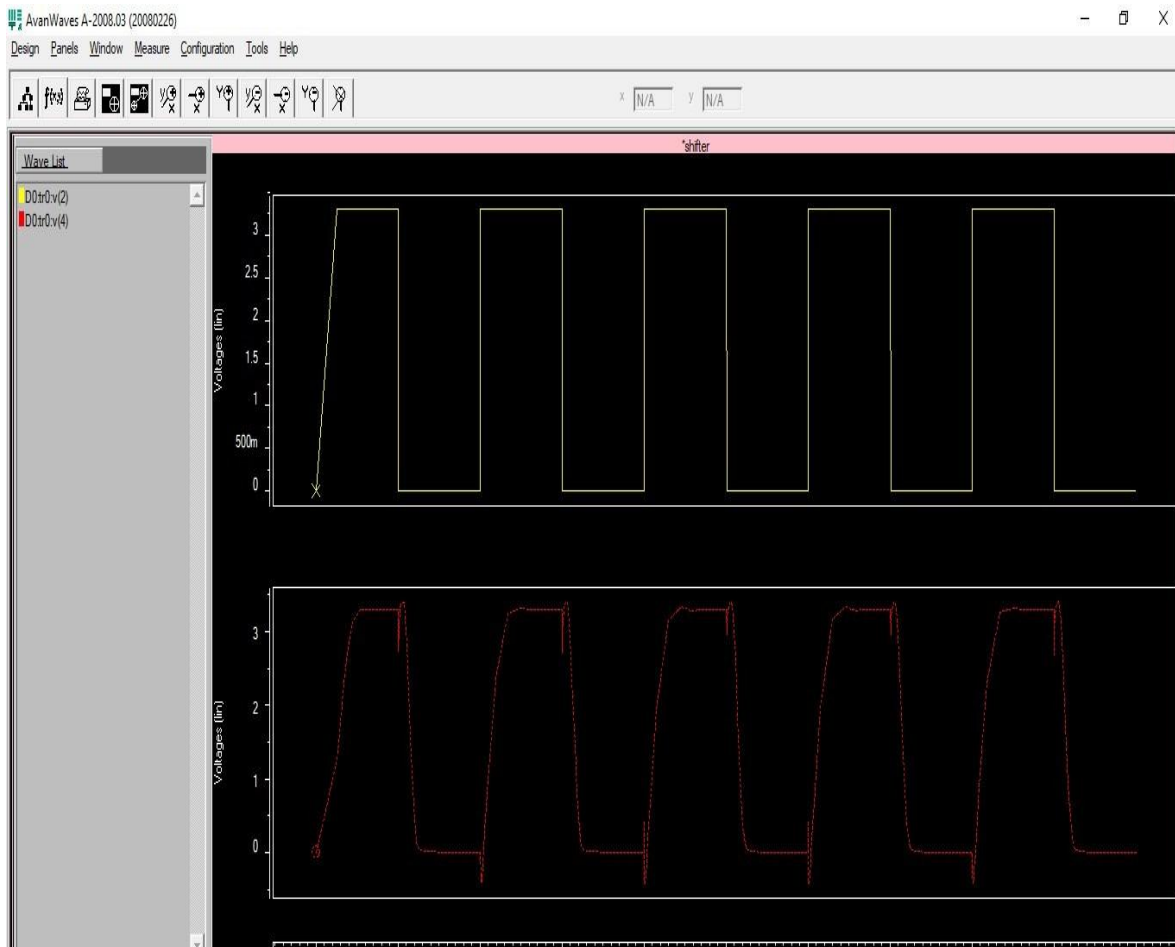


Fig 5.5: Simulation output for SHIFTER

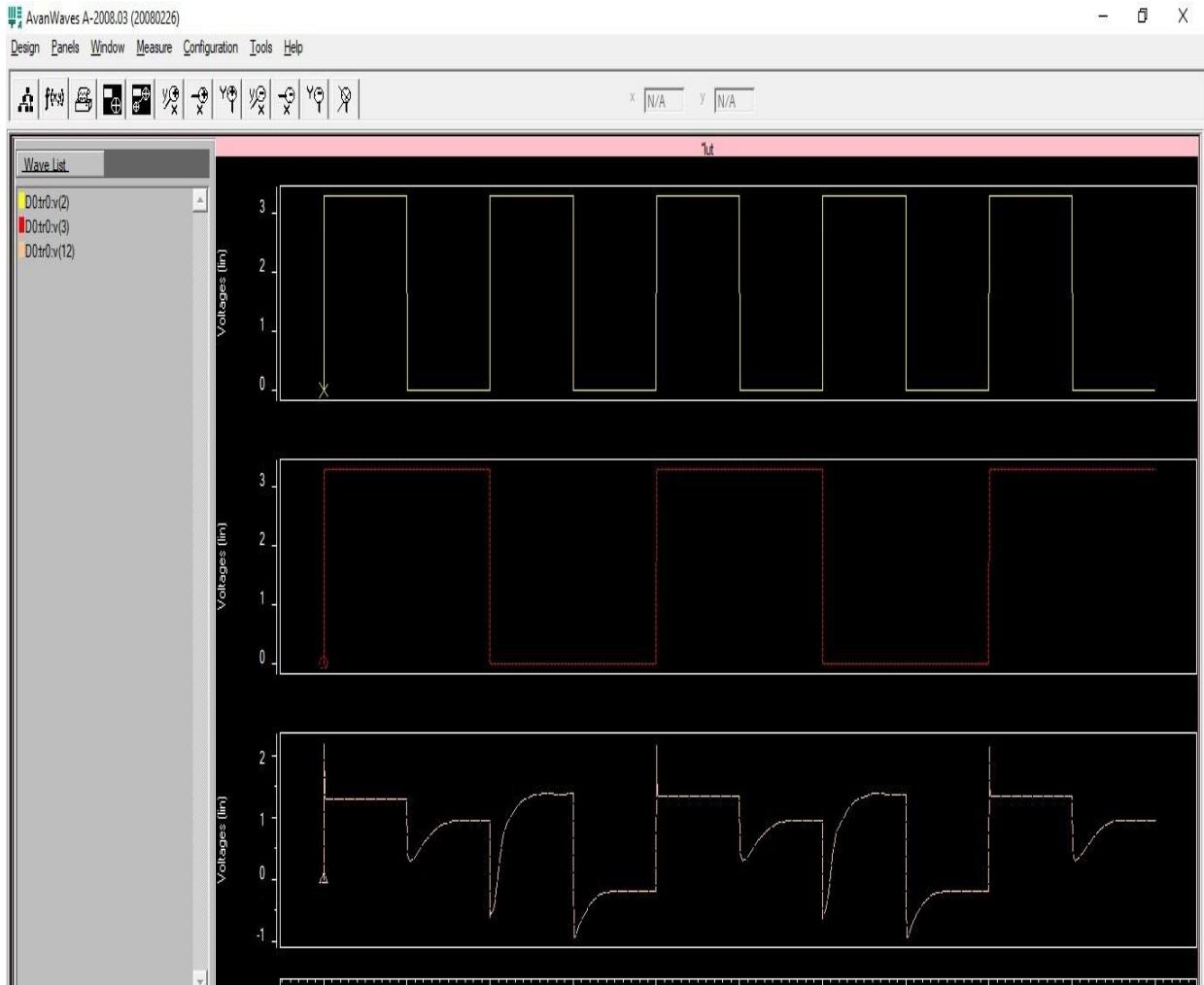


Fig 5.6 : Simulation output for lut

5.2 RESULT AND DISCUSSION:

CORDIC architecture is simulated by Avan waves simulator. Output of the existing and proposed CORDIC and existed CORDIC are analysed from the results and it is concluded that the proposed architecture reduces the power consumption compared with the existing system. The synthesis is done using Hspice UI.

5.3 PERFORMANCE ANALYSIS :

Performance analysis of CORDIC architecture in terms of power, minimum delay, rise time, fall time, power delay product has been measured.

Table 5.1: Performance analysis of existing and proposed method using 130nm MOSFET technology

cordic	Avg power	Minimum delay	Rise time	Fall time	Power delay product
Existing method	493.1uW	4.97ps	53.479ps	4.97ps	0.2450e-08 uWS
Proposed method	491.1uW	4.93ps	53.093ps	4.26ps	0.2092e-08 uWS

Table 5.2 :Performance analysis of existing and proposed CORDIC architecture using 32nm MOSFET technology

Cordic	Avg power	Minimum delay	Power delay product
Existing method	671.9uW	2.6091ps	0.1753e-08 uWS
Proposed method	614.1uW	1.3509ps	0.0829e-08 uWS

CHAPTER6

CONCLUSION

At an application level, CORDIC is widely used in the signal processing arena. In CORDIC compensation is implemented instead of using multiplications, helping to significantly reduce hardware complexity. In CORDIC algorithm is included in an efficient filtering design where power of two coefficients are calculated. CORDIC algorithm is applied directly in the implementation of a given rotation module used in improving computational time and decreasing complexity. In an adaptive CORDIC rotator with constant scaling factor is proposed aiming to reduce resources usage. Various CORDIC architectures exist. Determining the selection of a proper architecture for a CORDIC processor is a complex task that requires the analysis of different design strategies and evaluation of the principal parameter, power minimum delay, power product delay, fall time, rise time. In this work, we have successfully implemented an efficient CORDIC architecture. Performance analysis of both existing and proposed method is implemented through the hspice. Improving its performance can improve the overall efficiency of the system. The low power efficient resources usage achieved by the proposed architecture.

REFERENCES

1. J.E. Volder "The CORDIC Trigonometric computing technique" IEEE transactions on electronic computers, 1959
2. P.V.C. Hough. Method and Means for Recognizing Complex Patterns, United State Patent. No. 3 069654, Dec. 1962
3. R.O. Duda, P.E. Hart. Use of the Hough Transform to Detect Lines and Curves in Pictures, CACM, Vol. 15, No. 1, pp. 11-15, 1972.
4. H. Maître. Un Panorama de la Transformée de Hough, Traitement du Signal, Vol. 2, No. 4, 1985.
5. M. Meribout, M. Nakanishi, T. Ogur. A Real-Time Image Segmentation on Massively Parallel Architecture, Real-Time Imaging 5, pp. 279-291, 1999.
6. Y.H. Hu. The Quantization Effect of the CORDIC Algorithm, IEEE TRANS. on signal Processing, Vol. 40, No. 4, pp. 834-844, 1992.
7. N. Lindlbauer. Application of FPGA's to Musical Gesture and Processing, Master Thesis, Dept. of computer science and Electrical of Landshut University Germany, Nov. 1999.

8. R. Andrak. A Survey Algorithm For FPGA Based Computers, In Proceeding of the 1998 ACM/ SIGDA Sixth International Symposium on Field Programmable Gate Arrays, pp. 191-200, Monterey,CA, Feb.22-24,1998.<http://www.andraka.com/biblio.htm>
9. APS-X84 VHDL/FPGA Synthesis Tutorial, FPGA Basics, <http://users.erols.com/aaps/x84lab/FPGA.htm>
10. M. G. Albanesi, M. Ferretti, D. Rizzo. Benchmarking Hough Transform Architectures for Real-Time, Real-Time Imaging 6, 155-172 (2000).
11. C.-Y. Chen and W.-C. Liu, "Architecture for CORDIC algorithm realization without ROM lookup tables," in Proc. 2003 Int. Symp. on Circuits Syst., ISCAS'03, May 2003, vol. 4, pp. (544-547)
12. P. Dewilde, E. F. Deprettere, and R. Nouta, "Parallel and pipelined VLSI implementation of signal processing algorithms," in VLSI and Modern Signal Processing, S. Y. Kung, H. J. Whitehouse, and T. Kailath, Eds. Englewood Cliffs, NJ: PrenticeHall, 1995.
13. C.-H. Lin and A.-Y. Wu, "Mixed-scaling-rotation CORDIC (MSR-CORDIC) algorithm and architecture for high-performance vector rotational DSP applications," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 11, pp. 2385–2396, Nov. 2005.
14. M. S. Sinith and K. Jismi, "A comparison of pipelined parallel and iterative CORDIC design on FPGA," in 2010 5th International Conference on Industrial and Information Systems, 2010, pp. 239–243. Available: <http://dx.doi.org/10.1109/ICIINFS.2010.5578702>
15. Despain , A.M., "Fourier Transform Computations using CORDIC Iterations," IEEE Transactions on Computers, VOL.23

16. Duprat.J and Miller, J.M., " The CORDIC Algorithm: New results for Fast VLSI implementation ."IEEE Transactions on computers,vol.42,pp.
17. * Deprettere, E.Dewilde,P., and Undo.R, "Pipelined CORDIC Architecture for fast VLSI Filtering and Array Processing",proc.ICASSP'84.
18. * Aggarwal, S.; Meher, P. K.; khare,K., "Area Time Efficient Scaling Free CORDIC using generalized Micro Rotation selection, "VLSI systems, IEEE Transactions on VOL.20,2012.

REPORT

ORIGINALITY REPORT

9%

SIMILARITY
INDEX

4%

INTERNET
SOURCES

2%

PUBLICATIO
NS

3%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to 9561

Student Paper

2%

2

Leena Vachhanl"Efficient CORDIC Algorithms and Architectures for Low Area and High Throughput Implementation", IEEE Transactions on Circuits and Systems II Express Briefs, 01/2009

Publication

2%

3

www.investopedia.com

Internet Source

1%

4

Submitted to Siddaganga Institute of Technology

Student Paper

1%

5

repository.ntu.edu.sg

Internet Source

1%

6

Submitted to Grimsby College, South Humberside

Student Paper

<1%

7

"Chapter 6 6. Cordic Algorithm", Springer Science and Business Media LLC, 2005

Publication

<1%

8

www.citeulike.org

Internet Source

<1 %

9

Shyamkumar Thoziyoor, Jung Ho Ahn, Matteo Monchiero, Jay B. Brockman, Norman P. Jouppi. "A Comprehensive Memory Modeling Tool and Its Application to the Design and Analysis of Future Memory Hierarchies", ACM SIGARCH Computer Architecture News, 2008

Publication

<1 %

10

fdocuments.in

Internet Source

<1 %

11

Submitted to KUMARAGURU COLLEGE OF TECHNOLOGY

Student Paper

<1 %

12

www.inass.org

Internet Source

<1 %

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		