CALCULATION OF TRIGONOMETRIC FUNCTIONS USING CORDIC ALGORITHM

AMEEN MUSTAFA MOHAMMED ALNAFUTCHY

July, 2018

CALCULATION OF TRIGONOMETRIC FUNCTIONS USING CORDIC ALGORITHM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
CANKAYA UNIVERSITY

BY

AMEEN MUSTAFA MOHAMMED ALNAFUTCHY

IN PARTIAL FULLFILLMENT OF a REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

July, 2018

Title of a Thesis: **Calculation of Trigonometric Function Using Cordic Algorithm**

Submitted by **Ameen Mustafa Mohammed ALNAFUTCHY**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Can ÇOĞUN
Director

I certify that this thesis satisfies all requirements by way of a thesis for a degree of Master of Science.

Assist. Prof. Dr. Özgür ERGÜL
Acting Head of Department

This is to certify that we have read this thesis and in our opinion as a thesis, it is fully adequate, in scope and quality for the degree of Master of Science.

Assoc. Prof. Dr. Orhan GAZİ
Supervisor

Examination Date :17/ 07/ 2018

Examination Committee Members:

Assoc. Prof. Dr. Orhan GAZİ	Çankaya Univ.
Assist. Prof. Dr. Özgür ERGÜL	Çankaya Univ.
Assist. Prof. Dr. Javad RAHEBI	Turkish Aeronautical Association Univ.

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Ameen Mustafa Mohammed ALNAFUTCHY

Signature :

Date :17/07/2018

iii

ÖZET

KDSB (Koordinat Döndüren Sayısal Bilgisayar) KULLANILARAK TRİGONOMETRİK FONKSİYONLARIN HESAPLANMASI

ALNAFUTCHY, Ameen Mustafa Mohammed

M.Sc., Department of Electronics and Communication Engineering

Supervisor: Assoc. Prof. Dr. Orhan GAZİ

Temmuz, 2018

CORDIC kısaltması "rotation digital computer is an algorithm" cümlesi için kullanılan 1959 yılında Jack. E. Volder tarafindan bulunan bir algoritmanın ismidir. CORDIC algoritmasının icadından sonra bu algoritmanın ilerletilmesine yönelik çok sayıda çalışma yapılmıştır. CORDIC algoritması ilk olarak trigonometrik fonksiyonların hesaplanması, çarpma ve bölme işlemlerinin yapılması amacı ile kullanılmıştır. Daha sonra bu algoritma diğer matematik fonksiyonlarının hesaplanması içine de kullanılmıştır. Bu fonksiyonlara örnek olarak logaritma, üstel, karekök fonksiyonları örnek olarak verilebilir. CORDIC algoritması robotik, sinyal işleme, grafik ve animasyon, sayısal iletişim ve görüntü işleme gibi bir çok alanda kullanılmaktadır. CORDIC algoritması matematik fonksiyonlarının donanım cihazlarında gerçekleşmesi amacı ile geliştirilmiştir ve de hesap makineleri tarafindan kullanılmaktadır. Matematiksel bir fonksiyonun donanım gerçekleştirimi için gerekli olan cihazın büyüklüğü ve fiyatı matematiksek fonksiyonun hesaplanması için gerekli işlem miktarına bağlı olarak değişir. Zaman içerisinde daha hızlı yakınsayan ve daha doğru sonuçlar veren CORDIC algoritmaları araştırmacılar tarafindan önerilmiştir. Bu tez çalışmasında radix-2, radix-4, angle recoding, and extended angle recoding

CORDIC algoritmaları çalışılmıştır. Bu algoritmalar bilgisayar programları ile yazılmış ve bilgisayar benzetimleri yapılarak birbirleri ile kıyaslanmıştır. Bu kıyaslamalar sonucunda radix-4, angle recoding, and extended angle algoritmalarının radix-2 algoritmasına göre daha aynı doğruluk kıstasına ulaşmak için daha az sayıda yineleme gerektirdiği görülmüştür.

Anahtar kelimeler: CORDIC, radix-2, radix-4, extended angle recording, trigonometrik fonksiyonların hesabı.

ABSTRACT

CALCULATION OF TRIGONOMETRIC FUNCTIONS USING CORDIC ALGORITHM

ALNAFUTCHY, Ameen Mustafa Mohammed

M.Sc., Department of Electronics and Communication Engineering

Supervisor: Assoc. Prof. Dr. Orhan GAZİ

July 2018

CORDIC which is the abbreviation of coordinate rotation digital computer is an algorithm proposed in 1959 by Jack. E. Volder. Since its introduction, numerous studies are performed for improved versions of the CORDIC algorithm. CORDIC algorithm is initially introduced for the computation of trigonometric functions, multiplication and division operations. Later on, this algorithm is further developed for the calculation of other elementary transcendental functions such as logarithms, exponentials, square roots. CORDIC algorithm is used in many diverse areas such as robotics, signal processing, graphics and animation, digital communication, image processing. CORDIC algorithm is developed for the hardware implementation of mathematical functions, and it is shown by the researchers that CORDIC algorithm is a good choice for scientific calculators. The cost and size of the hardware equipment needed for the implementation of a mathematical function depends on the computation complexity of the algorithm under concern. In time, CORDIC algorithms with higher precision and faster convergence rates are proposed in literature. In this thesis work we study radix-2, radix-4, angle recoding, and extended angle recoding CORDIC techniques and compare the algorithms considering the number of iterations required for a defined precision. Algorithms are simulated via computer programs.

The results show that the radix-2 has requires more number of iterations compared to radix-4, angle recoding and extend angle recoding methods.

Keywords: CORDIC, radix-2, radix-4, angle recoding, extended angle recording, computation of trigonometric functions.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to my supervisor Assoc. Prof. Dr. Orhan Gazi for his supervision and encouragement during my thesis study. I am very grateful for his leading role, advices and vision during this research.

I also wish to thank to all my family for their support and encouragement during my studies.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1. Introduction

In signal processing applications, it may be necessary to compute the precise values of trigonometric functions in real-time. There are different methods existing in the literature for the calculation of trigonometric functions, and two well-know of these methods are Taylor series approximations and CORDIC technique. However, these methods either involve the use of a multiplier or they have iterative structure, and for these reasons their processing latency is high.

The CORDIC algorithms are iterative methods utilized to calculate trigonometric functions without the utilize of the multiplier [1]. After the introduction of CORDIC algorithm in 1956 huge improvement has been done on numerous varieties of this algorithm to calculate more complex functions [2]. However, the method in general uses p iterations to provide an output having p bit precision. This high iteration number implies a high output latency. The method also provides a scaled result, although the scaling factor is constant for basic CORDIC methods. Therefore, a scaled initial vector can be used before the iterations.

These methods have been extensively utilized to numerous applications for example matrix transformations [3-4], decimal-to-binary conversions [5], singular value decompositions [6]. An important step of the original CORDIC algorithm is a determination of sign of residual angle W. The sign of are mining angle is used to determine direction of rotation for next iteration. Some improved version of the CORDIC algorithm [7] can be stated as, binary-to-bipolar recoding (BBR), micro-rotation angle recoding (MAR) [8], modified vector rotational CORDIC (MVR-CORDIC) [9], Hybrid CORDIC [10].

The rotations are performed in the direction determined by the residual angle, and the magnitude of each rotation is predefined. The idea is to perform rotations in some directions such that residual value gets smaller.

While the approach is not offering rotations, which allows a minimum iteration count as provided by the trellis-based searching schemes [11] but, it allows achieving ea. rotation in parallel that results in the reduction in whole delay. It also indicates an optimized method to deem the inverse of trigonometric functions using the extended vectoring mode.

By considering the magnitude of rotations; it is figured that the acceleration of convergence of this mode is difficult to be obtained but; with reducing the residue to a sufficient value, it could be achieved.

The additional representation over the higher level of granularity has been used in both modes to provide an estimation of the variable value with a good accuracy. While minimizing the critical path time, computes the residual angle. The proposed method is designed and verified to work in a circular co-ordinate scheme (although it can be extended to support the other co-ordinate schemes as well).

1.2. Thesis Organization

Remaining sections are outlined below:

In the chapter-2, literature review is considered. In the chapter 3, various CORDIC methods have been explained in details such as radix-2, radix-4, angle recoding and extended angle recoding. Furthermore, we consider different trigonometric functions for evaluation. Also, we support our analysis with numerical examples. In chapter-4, the evaluation of the different CORDIC schemes are analyzed and discussed, finally, in the chapter five the conclusion and future work is given.

CHAPTER 2

FUNDAMENTALS AND EXISTING APPROACHES

2.1. The CORDIC Method

The original CORDIC technique has two operation modes: rotation mode, which it is utilized to calculate the values of a “sine” and “cosine functions”, and a vectoring mode, which is used to compute an inverse of the trigonometric functions. The function of method can also be interpreted as the conversion of a unit vector from one pole to Cartesian co-ordinate scheme and vice versa. It can be designed to accept input (provide output) in any unit, such as degrees, radians and binary fractions of a half revolution for the rotation (vectoring) mode. In general, it takes “*p*” iterations to converge to a result with precision of “*p*” bits for both modes of the method. Also, each iteration involves a “*p*”bit operation.

In a rotation mode of the CORDIC, it started with a known vector value which is generally taken as “0” radian. This vector is then rotated about the predefined angles to converge onto the required input angle. In vectoring mode of the algorithm, a value of the vectors is taken as input and stored as “X “and“Y”. The value of residual angle is initialized to “zero”. Rotations are performed similar to the rotation mode except that the rotations directions are computed via a sign of “Y” instead.

2.2. Existing Approaches

CORDIC method calculates 2D rotation utilizing alternative equations with shift and add operations. A versatility of CORDICs are improved via emerging methods on a similar basis for binary coded decimal (BCD) number illustrational Daggett in 1959 [12]. These iterative approaches are presented utilizing decimal radix for the design of a powerful small machine via Meggitt in 1962 [13]. Then, Walther in 1971 [14] planned the unified algorithm to calculate rotation in circular, linear, and hyperbolic coordinate schemes utilizing similar CORDIC technique, implanting coordinate schemes as parameter.

In the last 50 years of CORDIC algorithm the wide application variety appeared. CORDIC methods have received increased consideration after the unified methods are proposed for its application [15]. CORDIC has been a choice to scientific calculator applications as well as HP-2152A co-processor, HP- 9100 desktop calculator and HP-35 calculator which are some example devices relying on a CORDIC method [16].

CORDIC arithmetic processor chips are planned and applied to achieve numerous functions possible in rotation as well as vectoring mode of circular, linear, as well as hyperbolic coordinate schemes [17]. Furthermore, CORDIC methods have been utilized in numerous applications [18], for example a single chip CORDIC processor for DSP applications in introduced in [19], and linear transformations [20], digital filters [21], [22], as well as matrix-based signal processing algorithms [22] are implemented in CORDIC.

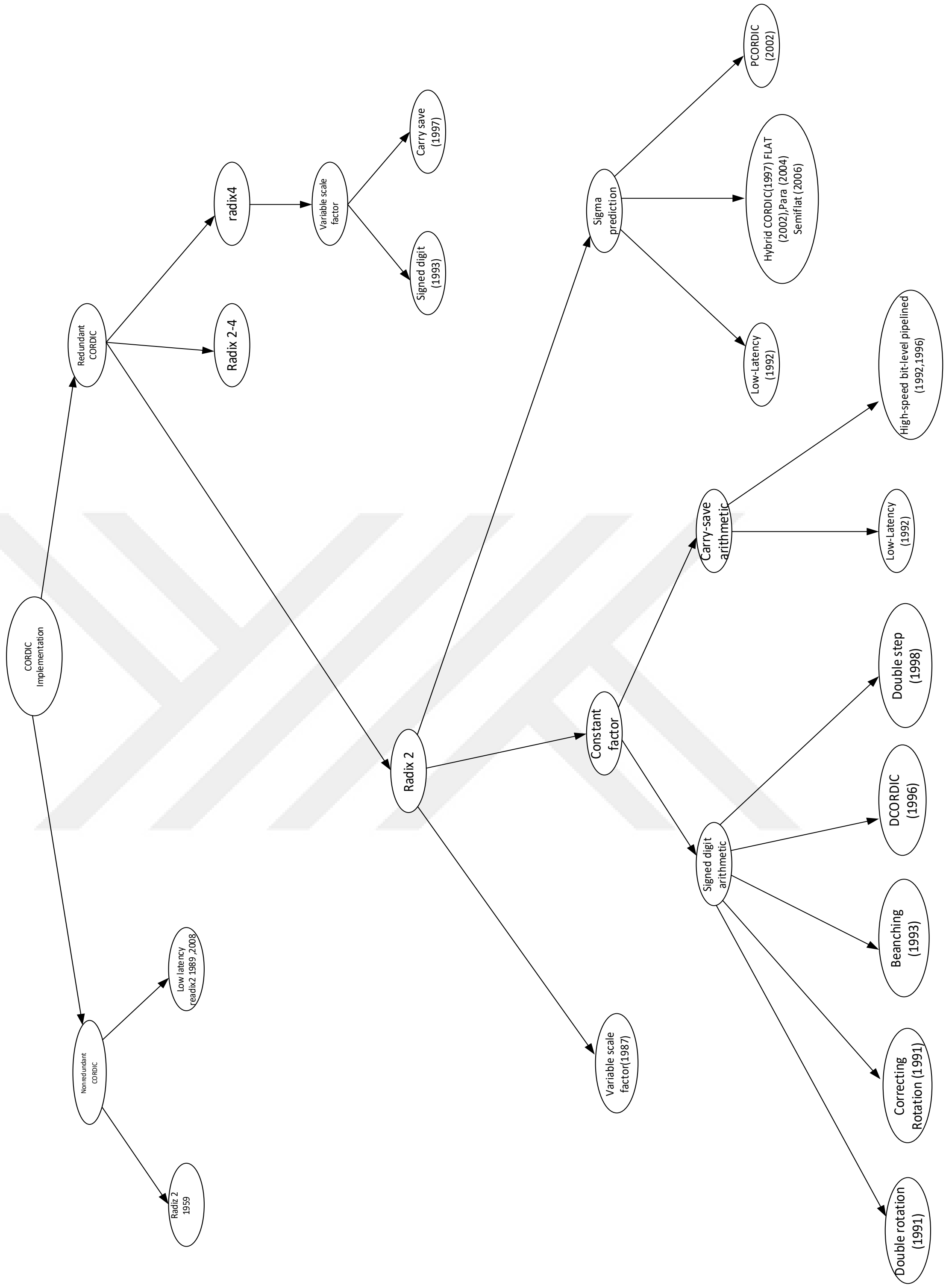

Figure 2.1: Hierarchical of CORDIC methods [17].

While CORDIC is not the fastest method to achieve these operations, it is attractive because of its potential for effective and cost saving operation of a large class of applications. Numerous changes are planned in the researches of the CORDIC method in the last two decades to offer high performance as well as cost saving hardware solutions to real time calculation of double dimensional vector rotation and transcendental functions.

A critical study of various architectures is proposed and implemented in this research for a 2D rotational CORDIC in circular coordinate scheme to initiate the work to the additional delay reduction or throughput enhancement. Also, a different CORDIC methods were studied in this research.

A technique for correction of scale factor of CORDIC methods was proposed [21]. The scheme needs some few extra hardware for its operation, nonetheless do not need altering elementary rotation angles or sequence of iterations of standard CORDIC methods. Upper bounds for quantization error after utilizing proposed technique is formulated. The word serial implementation of method is also set. In the fixed-point arithmetic, area as well as delay of a suggested implementation are with regarding to standard CORDIC.

The CORDIC methods which are utilized in calculation of the extensive variation of an elementary function were studied [20]; studied. It is the easiest and most elegant technique, in spite of its defect of the long delay. Angle Recoding techniques are capable of decreasing the number of iterations via greater than 50 percent, while its operation in hardware needs a large increment in the cycle time to accommodate its complex angle selection function. This situation limits its utilizing for such a case where the angle of rotations is fixed as well as known in advance, hence angle choosing is implemented offline. The simpler operation of angle taking system that do not need an increase in cycle time, thence permitting an Angle Recoding technique must be utilized dynamically for random angles.

The technique has also a benefit in-which the whole angles are fixed and are obtained in parallel in one step by testing only an initial rotation angle and, without having to achieve a successive CORDIC iterations. Such a dynamic Angle Recoding technique is modelled to utilize "sections" to bind a specific number of range comparators required to reach an appropriate value.

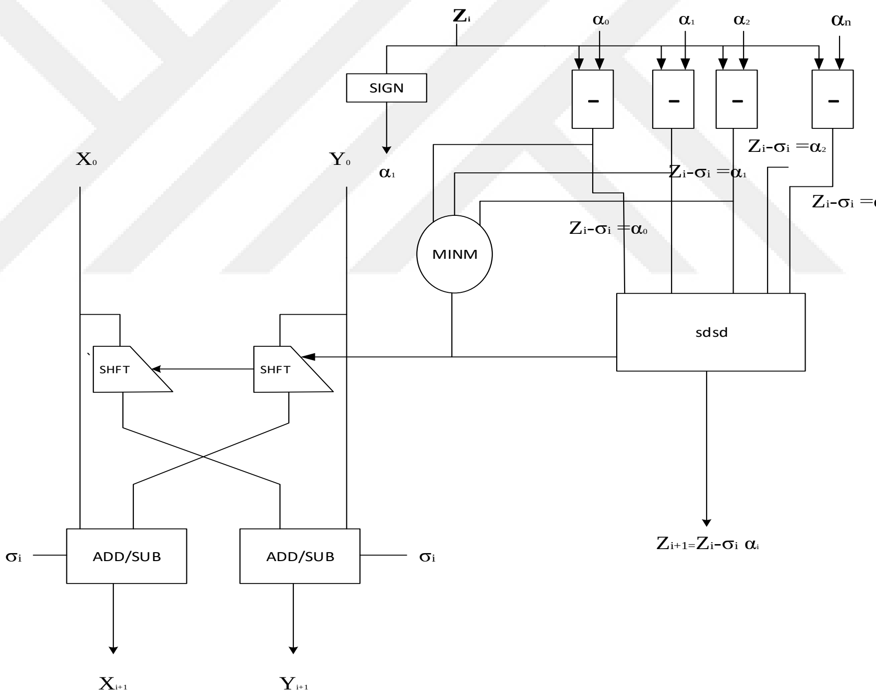

Figure 2.2 Angle Recoding method [21].

The work suggests a modern approximate system for CORDIC design. The scheme is relying on adapting a current Para-CORDIC architecture including approximation which introduced in numerous parts as well as made possible via relaxing a CORDIC method itself.

The fully parallel estimated CORDIC schemes are suggested and; this system avoids a memory register of Para-CORDIC as well as creates a generation of rotation direction fully parallel. The complete examination as well as a calculation of error presented by an approximation along with various circuit associated metrics had been pursued utilizing HSPICE by way of simulation tool.

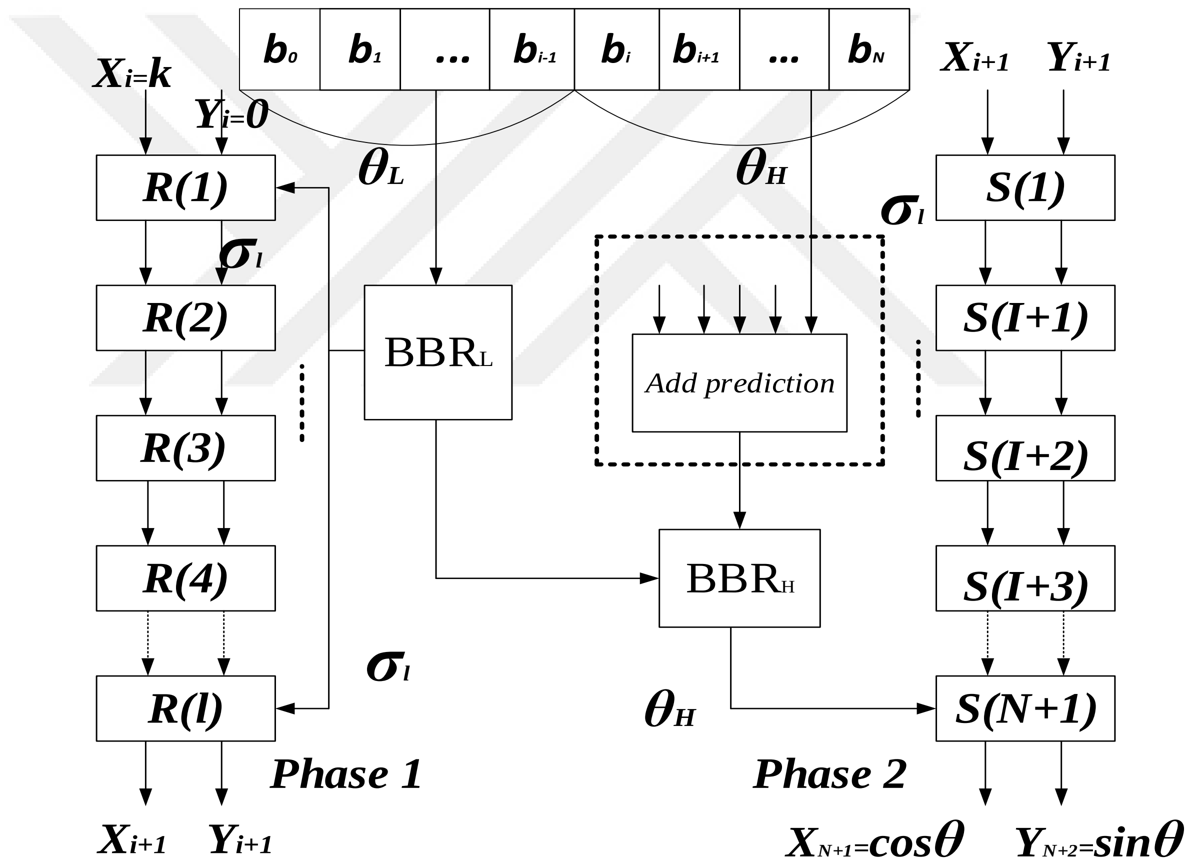

Figure 2.3: The Para CORDIC Architecture. [22]

Additionally, this error examination combines present figures of value for estimated calculating as well as MED Power Product (MPP)) including CORDIC exact parameters. It revealed that, a good agreement of error values is obtained between the predictable and the simulated. A Discrete Cosine Transformation (DCT) and an Inverse DCT (IDCT) transformations by way of the case study of approximate computing to image processing are examined by using a proposed approximate FPAX CORDIC architecture with various accurateness needing. The results confirm the viability of suggested scheme.

The research suggests a novel CORDIC-based fast radix-2 method for calculation of a discrete sine transforms (DST). A proposed algorithm is making a next higher order transforms from lower order transforms and have a few different benefits, for example regular and purely feed forward data path, in place calculation, unique post-scaling factor and, arithmetic-sequence CORDIC rotation angles.

Related to the current methods such as the suggested algorithms are not merely having lesser arithmetic complexity in addition it is admit effective pipelined VLSI application. Furthermore, the simplicity of getting a fast inverse DST via utilizing orthogonal characteristics.

Rotation-extension CORDIC approaches, i.e. double-rotation and triple rotation, are suggested for objective of enhancement performance and accurateness of a CORDIC calculation method in radix-2. In a two-rotation as well as triple-rotation approaches, a convergence of CORDIC calculations are speeded up via repeating and triplicating micro-rotation angles to be “2θ” and “3θ”. Non-redundant mechanism, wherever rotation direction “δ” is in a set of “1” and “-1”,is relying on intermediate converging parameter and it is applied to fix the scaling factors.

Convergence range and accurateness of elementary functions hardware implemented by utilizing CORDIC approaches in rotation mode and vectoring mode on circular, hyperbolic and, linear coordinate schemes are inspected and related to MATLAB simulation.

A comparison performance demonstration that proposed CORDIC techniques offer greater computational accuracy compared to the traditional one at same number of iterations. The high precision CORDIC method is presented and estimated for VLSI. Finally, speed and area performances of CORDIC hardware rely on pipeline (unfolded) digit-parallel architecture of proposed CORDIC techniques are compared to CORDIC techniques are also specified.

A reduction of CORDIC computational latency with high radix technique creates scaling factor in instability situation. Unstable scaling factor problem is resolved by online computation techniques which are also complex for hardware practice. A parallel method is utilized to accelerate calculations. The techniques are relying on double traditional CORDIC cores processing in parallel. Thus, the scaling factor problems are resolved, then estimation overhead of rotation direction as well as combination of two traditional CORDIC results are added. An extension rotation technique relying on radix-2 accelerates micro-rotation angle "φ" with the integer value "n".Double-rotation technique achieves computational results with micro-rotation angle 2φ. Its scaling factors which are first planned by online calculation technique and with optimizing an error of a computational result in order to fix the scaling factor.

A relying on a decreasing computational delay as well as enhancing computational accurateness of CORDIC including a constant scaling factor relying on radix-2. A radix-2 utilized due to scaling factor available mathematically. A design and architecture of a CORDIC method on the radix-2 is straightforward for hardware operation. Both rotation extension CORDIC techniques are suggested, i.e. non-redundant double-rotation and non-redundant triple-rotation. It is examined explored and simulated to calculate a fix scaling factors and an input domain ability for the elementary functions (an initial parameter and compensation parameter values for each basic function which performed in circular, hyperbolic and, linear).

A CORDIC methods are the iterative arithmetic for the implementing vector rotations in numerous DSP tenders. Though, a large number of iterations are the main weakness of this method for its speed performance. Numerous investigators have planned systems to decrease a number of iterations.

However, in execution a current CORDIC method norms of the vectors are typically an enlarged so that additional scaling processes are needed to the deliver a normalized output. Merging two process stages; micro-rotations as well as scaling phases and the vector rotational system; is named as a mixed-scaling-rotation coordinate rotational digital computer method. It removes overhead of the scaling processes which is inevitable in current CORDIC methods. Henceforth; it can significantly decrease total iteration number to enhance a speed performance.

A suggested MSR-CORDIC is applied to DSP applications that rotational angles are recognized in the advance. Furthermore, most CORDIC methods are usually undergo from the round off noise in the fixed-word-length implementations. The proposed two systems are to control and decrease the weakening. There are simulation results demonstration which an MSR-CORDIC method is improve a signal-to-quantization-noise ratio (SQNR) performance via regulatory internal dynamic range.

The first and second-order statistical characteristics were examined and with a mean and variance of a SQNR. Simulation results demonstration which an MSR-CORDIC is improve SQNR performance of the both first and second-order statistical properties. A VLSI architecture level of the proposed generalized MSR-CORDIC engine for a tradeoff among hardware complexity as well as quantization error performance which decreases the hardware complexity when compared with the newly proposed extend elementary angle set CORDIC method [5]. An MSR-CORDIC schemes have been applied to a variable-length FFT processor design [16], and the results in the significant hardware is decreasing in applying a twiddle factor operation.

The optimized and generalized CORDIC method in a rotation mode of a circular co-ordinate scheme. It calculates values of a trigonometric functions and it organized to give a result with the lower overall delay compared to present schemes. This is accomplished via utilizing redundant representations and estimates of a needed direction as well as angle of each rotation. Some methods have been calculated to offer a result in the stable number of iterations equal to the design parameter by way of chosen via a designer. Every iteration method achieves rotations among zero and a certain value “number”. The method to handle a scaling factor compensation for such methods is proposed. A result of functional verification for various values of a design parameter and the approximation of whole delay are illustrated.

CORDICs are the alternative methods that are utilized to compute mathematical functions; e.g. trigonometric hyperbolic exponential functions. An alternative process of traditional CORDICs is a time-consuming due to its insufficient rotation strategy. Lately, several enhanced rotation plans have been suggested to decrease the disproportionate rotations that are significantly decrease between (30% - 60%) of an alternative process for the CORDIC method. But, there are better rotation strategies which are utilized for several functions. Meanwhile, the only plan to be well-matched with circular coordinate scheme as well as rotation mode of the CORDIC method to overcome such difficulty is presented by a unified rotation plan to allow functions of an enhanced rotation approaches for various coordinate schemes as well as operating modes of a CORDIC method. A suggested rotation strategy contains double mechanisms that have been unified coarse rotation as well as unified precise rotation which is drastically decrease disproportionate rotations temporarily which maintains a higher computing precision than the conventional methods.

CHAPTER 3

CORDIC METHODS

3.1. CORDIC

The CORDIC methods are utilized for the calculation of elementary trigonometric functions. It is also employed for the calculation of other elementary functions, and new variants continue to appear. As implied via its name, the underlying idea is the rotation in the geometric space. In this chapter, we are going to evaluate trigonometric, inverse trigonometric and hyperbolic functions via CORDIC algorithm.

3.2. Trigonometric functions

The calculation of a sine as well as cosine exemplify the best use of the CORDIC algorithm. Therefore, we start with a discussion of these two functions and then proceed to other functions. Calculation of sine and cosine functions are performed in rotation mode, via rotating a unit-length vector over a given angle.

In Figure 3.1, rotation of a unit vector with coordinates (X_i, Y_i) at an angle φ from the x axis by an angle θ_i is demonstrated. The coordinates,$(X_i^* + 1, Y_i + 1)$, of the new vectors are evaluated as

$$X_{i+1}^* = cos(\theta_i + \varphi) \tag{3.1}$$

$$X_{i+1}^* = cos\varphi cos(\theta_i) - sin\varphi sin(\theta_i) \tag{3.2}$$

$$X_{i+1}^* = X_i cos(\theta_i) - Y_i sin(\theta_i) \tag{3.3}$$

$$X_{i+1}^* = (X_i - Y_i tan(\theta_i))cos(\theta_i) \tag{3.4}$$

$$Y_{i+1}^* = sin(\theta_i + \varphi) \tag{3.5}$$

$$Y_{i+1}^* = \sin(\varphi)\cos(\theta_i) - \cos(\varphi)\sin(\theta_i) \tag{3.6}$$

$$Y_{i+1}^{*} = X_i \cos(\theta_i) - Y_i \sin(\theta_i) \tag{3.7}$$

$$Y_{i+1}^{*} = (X_i - Y_i tan(\theta_i)) \, cos(\theta_i) \tag{3.8}$$

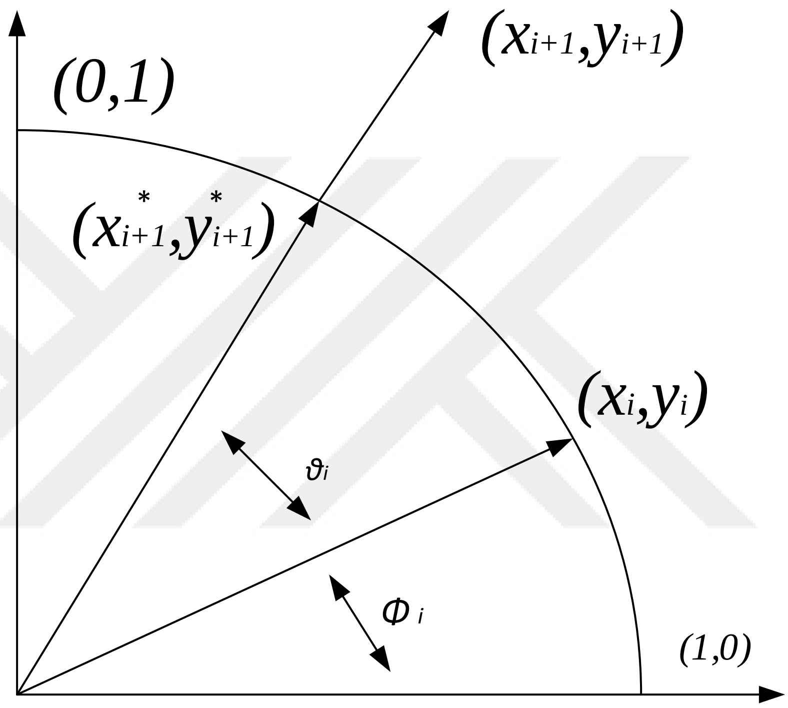

Figure 3.1. CORDIC representation using rotation mode [5].

As it is indicated in $(3.1) - (3.9)$ after rotation we obtain the new coordinates $(X_{i+1}^{*}, Y_{i+1}^{*})$ which can be written omitting the scaling factor as

$$X_{i+1} = X_i - Y_i \, tan \, (\theta_i) \tag{3.9}$$

$$Y_{i+1} = X_i + Y_i \, tan \, (\theta_i) \tag{3.10}$$

As we shall see in the following paragraphs CORDIC hardware implementation complexity for the calculation of trigonometric functions.

A comparison of $(1) - (2)$and (3) demonstrates that vector(X_{i+1}, Y_{i+1}), is has a scaling factor of$1/\cos\theta_i$. Consequently, at the end of final rotations of method, it is important to multiply, i.e., scale, each X_i and Y_i by this factor. Then, if number of rotations is constant number, then the scaling by $1/\cos\theta_i$ at each given step, gives rise to the total scaling factor

$$K = \prod_{i=0}^{n} \frac{1}{\cos\theta_i} \tag{3.11}$$

Which can be written as

$$K = \prod_{i=0}^{n} \sqrt{1 + \tan(\theta_i)} \tag{3.12}$$

In fact, we can initializeX_0 to$1/K$ to take the scaling factor into account at the beginning of the iterations. This initial scaling in X subsequently introduces a similar scaling in$Y$. We can choose the rotations angles θ_i such that$\tan\theta_i = 2^{-i}$, this yields to

$$X_{i+1} = X_i - Y_i 2^{-i} \tag{3.13}$$

$$Y_{i+1} = X_i - Y_i 2^{-i} \tag{3.14}$$

Where multiplication by2^{-i}equals to a shift to right by i bits, and the initial value $K = X_0$ can be calculated using

$$K = \prod_{i=0}^{n} \sqrt{1 + 2^{-2i}} \tag{3.15}$$

Last issue in the convergence concept. In order to choose whether to add or subtract in the current iteration, total rotation angle is subtracted fromθ. If the difference is negative, then an addition takes place; and if it is positive, then a subtraction takes place.

CORDIC algorithm is based on the rotations on a geometric space. The amount of rotations affects the accuracy of the calculations. In this algorithm, the total rotation angle is calculated incrementally by way of$\theta_0 \pm \theta_1, \theta_0 \pm \theta_1 \pm \theta_2, \theta_0 \pm \theta_1 \pm \theta_2 \pm \theta_3$, etc. rotation actions therefore correspond to making adjustments according to “how much farther we have to go”, angle residual is calculated in a third variable Z_i.

The initialization and iteration steps for the calculation of *sine* and *cosine* trigonometric function can be outlined as

$$X_0 = \frac{1}{K}, Y_0 = 0, Z_0 = \theta \; and \; \theta_i = \tan^{-1} 2^{-i}$$

$$X_{i+1} = X_i - s_i 2^{-i} Y_i \quad (3.26)$$

$$Y_{i+1} = Y_i + s_i 2^{-i} X_i \quad (3.17)$$

$$Z_{i+1} = Z_i - s_i \theta_i \quad (3.18)$$

$$s_i = \begin{cases} 1 & if \; Z_i \geq 0 \\ -1 & if \; Z_i \leq 0 \end{cases} \quad (3.19)$$

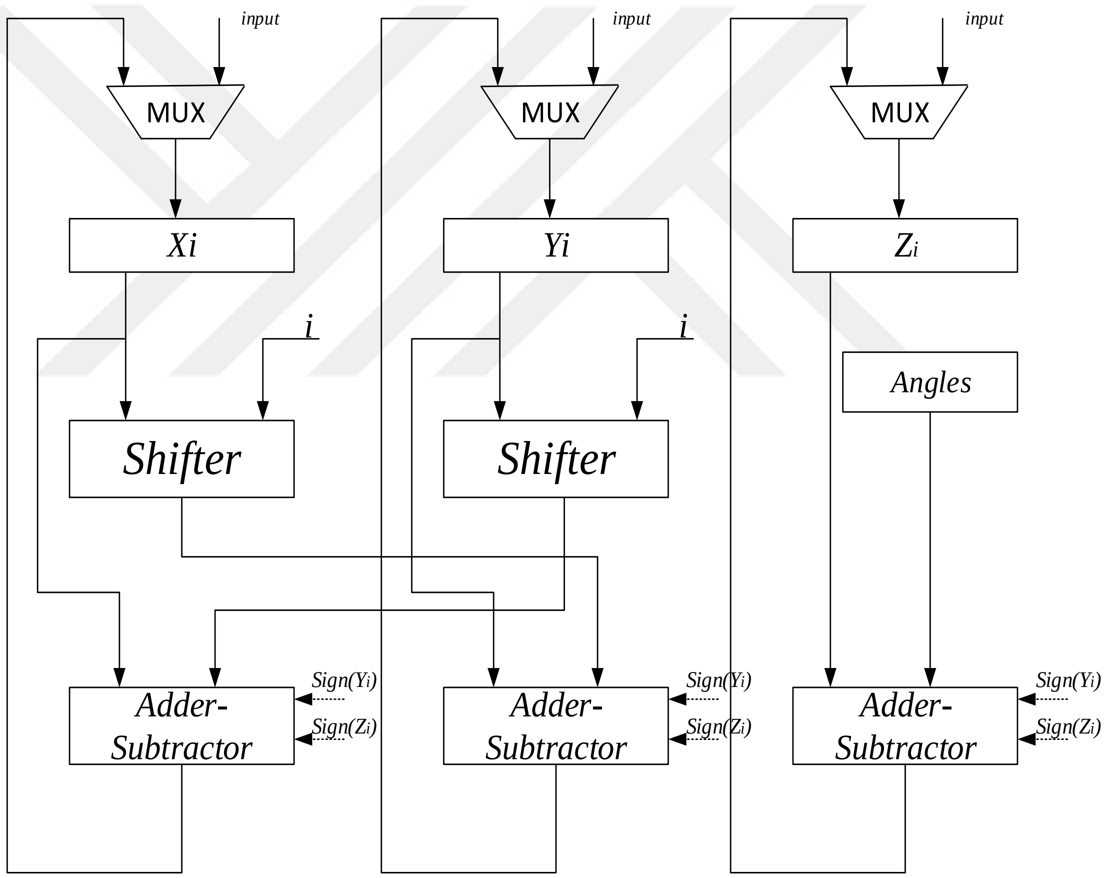

Figure 3.2: Architecture of traditional CORDIC [16].

3.2.1. Numerical Example for Sine and Cosine Function Calculation

In this section we will provide a numerical example for the calculation of sine and cosine function using the rotation mode of CORDIC algorithm. Sine and cosine function are evaluated for a degree of 0.735 radians.

First, we set initial values as:

$$Y_1 = 0\,, \qquad Z_1 = 30^{\mathrm{o}} \to 0.523 \text{ rd}, \qquad \mathrm{s}_1 = +1$$

$$K = \prod_{i=1}^{10} \sqrt{1+2^{-2i}} = 1.1644, X_1 = \frac{1}{K} = 0.8588$$

Next, we perform the iterations as follows:

First iteration:

$$i = 0$$

$$\theta_0 = 0.7853\, rad$$

$$X_2 = X_1 - s_0 2^{-1} Y_1 \to X_2 = 0.8588$$

$$Y_2 = Y_1 - s_0 2^{-1} X_1 \to Y_1 = 0.4294$$

$$Z_1 = Z_0 - s_0 \theta_1 \to Z_1 = -0.2623$$

Second iteration:

$$i = 1$$

$$\theta_1 = 0.4636 \mathrm{rad} Z_1 < 0, s_2 = -1$$

$$X_2 = X_1 - s_1 2^{-1} Y_1 \to X_2 = 0.8588$$

$$Y_2 = Y_1 - s_1 2^{-1} X_1 \to Y_2 = -\,0.4294$$

$$Z_2 = Z_1 - s_1 \theta_1 \qquad \to Z_2 = -0.2016$$

Third iteration:

$$i = 2$$

$$\theta_2 = 0.2450 \mathrm{rad} Z_2 < 0, s_2 = -1$$

$$X_3 = X_2 - s_2 2^{-2} Y_2 \to X_3 = 0.75145$$

$$Y_3 = Y_2 + s_2 2^{-2} X_2 \rightarrow Y_3 = 0.2147$$

$$Z_3 = Z_2 - s_2 \theta_2 \quad \rightarrow Z_3 = -0.0434$$

and we go on like this till the last iteration,

Last iteration:

$$i = 10,$$

$$\theta_{10} = 0.0020\, rd, \qquad Z_2 < 0, s_{10} = -1$$

$$X_{11} = X_{10} - s_{10} 2^{-10} Y_{10} \rightarrow X_{11} = 0.8569$$

$$Y_{11} = Y_{10} + s_{10} 2^{-10} X_{10} \rightarrow Y_{11} = 0.5002$$

$$Z_{11} = Z_{10} - s_{10} \theta_{10} \quad \rightarrow Z_{11} = -0.002$$

After last iteration, we obtain the sine and cosine values as $X_{11} = 0.8569$, $Y_{11} = 0.5002$.

3.3. Calculation of Inverse Trigonometric Function via CORDIC Algorithm

Assume that you know the sine or cosine value of an angle and you want to determine the value of the angle referencing its sine or cosine value. This can be achieved using the CORDIC algorithm either in rotation mode or in vectoring mode. For the calculation arcsine function, the CORDIC algorithm in rotation mode is as follows [2]:

$$X_0 = 1\, , Y_0 = 0\, , Z_0 = 0, y_0 = x \text{ and } \theta_i = \tan^{-1} 2^{-i}$$

$$y_{i+1} = y_i + 2^{-2i} y_i \tag{3.20}$$

$$X_{i+1}^* = X_i - s_i 2^{-i} Y_i \tag{3.21}$$

$$Y_{i+1}^* = Y_i + s_i 2^{-i} X_i \tag{3.22}$$

$$X_{i+1} = X_{i+1}^* - s_i 2^{-i} Y_{i+1}^* \tag{3.23}$$

$$Y_{i+1} = Y_{i+1}^* + s_i 2^{-i} X_{i+1}^* \tag{3.24}$$

$$Z_{i+1} = Z_i + 2 s_i \theta_i \tag{3.25}$$

$$s_i = \begin{cases} 1 & if X_i \leq y_i \\ -1 & otherwise \end{cases} \quad (3.26)$$

The CORDIC algorithm in rotation mode for the calculation of arccosine function is as follow [3]:

$$R_0 = 1\ , Y_0 = 0\ , Z_0 = 0, r_0 = r\ and\ \theta_i = \tan^{-1} 2^{-i}$$

$$y_{i+1} = y_i + 2^{-2i} y_i \quad (3.27)$$

$$R_{i+1}^* = R_i - s_i 2^{-i} Y_i \quad (3.28)$$
$$Y_{i+1}^* = Y_i + s_i 2^{-i} R_i \quad (3.29)$$

$$R_{i+1} = R_{i+1}^* - s_i 2^{-i} Y_{i+1}^* \quad (3.30)$$

$$Y_{i+1} = Y_{i+1}^* + s_i 2^{-i} R_{i+1}^* \quad (3.31)$$

$$Z_{i+1} = Z_i + 2 s_i \theta_i \quad (3.32)$$

$$s_i = \begin{cases} +1 & if\ Y_i \geq y_i \\ -1 & otherwise. \end{cases} \quad (3.33)$$

3.3.1. Numerical Example on Arcsine and Arccosine Function Calculation

In this section we provide numerical examples for the evaluation of arcsine and arccosine function using CORDIC algorithm in rotation mode.

For $i = 0$

$$\theta_0 = \ 0.4636, X_0 = 1\ , Y_0 = 0\ , Z_0 = 0, x_0 = 0.5\ , s_0 = +1$$

$$X_1^* = X_0 - s_0 2^{-1} Y_0 \ \rightarrow\ X_1^* = 1$$

$$Y_1^* = Y_0 - s_0 2^{-1} X_0 \ \rightarrow\ X_1^* = 0.5$$

$$X_2 = X_1^* - s_0 2^{-1} Y_1^* \rightarrow\ X_2 = 0.75$$

$$Y_2 = Y_1^* + s_0 2^{-1} X_1^* \ \rightarrow\ Y_1 = \ 1$$

$$Z_1 = Z_0 + s_0\theta_1 \rightarrow \; Z_1 = \; 0.9273$$

For$i = 1$

$$\theta_1 = 0.4636, X_1^* = 1, Y_1^* = 0.5 \quad X_1 = 1\ , Y_1 = 0\ , Z_1 = 0.9273, x_1 = 0.5\ , s_1 = \ +1$$

$$X_2^* = X_2 - s_1 2^{-2} Y_2 \; \rightarrow \; X_2^* = 0.5$$

$$Y_2^* = Y_2 - s_1 2^{-2} X_2 \; \rightarrow \; Y_1^* = \; 1.1875$$

$$X_2 = X_2^* - s_1 2^{-1} Y_2^* \; \rightarrow \; X_2 = 0.2031$$

$$Y_2 = Y_1^* + s_1 2^{-1} X_1^* \; \rightarrow \; Y_2 \; = \; 1.3125$$

$$Z_2 = Z_1 + s_1\theta_1 \qquad \rightarrow Z_1 \; = \; 1.4173$$

For$i = 2$

$$\theta_2 = 0.1244, X_2^* = 0.5 \;\; Y_2^* = 1.1875 \;\; X_2 = 0.2031\ , Y_2 = 1.3125 \;\; Z_2 = 1.4173 \;\; x_2 = 0.6744\ , s_2 = \ -1$$

$$X_2^* = X_2 - s_1 2^{-2} Y_2 \; \rightarrow \; X_2^* = 0.3672$$

$$Y_2^* = Y_2 - s_1 2^{-2} X_2 \; \rightarrow \; Y_1^* = \; 1.2871$$

$$X_2 = X_2^* - s_1 2^{-1} Y_2^* \; \rightarrow \; X_2 = \; 0.5281$$

$$Y_2 = Y_1^* + s_1 2^{-1} X_1^* \; \rightarrow \; Y_2 \; = \; 1.2412$$

$$Z_2 = Z_1 + s_1\theta_1 \qquad \rightarrow Z_1 \; = \; 1.1685$$

For $i = 3$

$$\theta_2 = 0.0624, X_2^* = 0.3672, Y_2^* = 1.2871, \qquad X_2 = 0.5281\ , Y_2 = 1.2412\ , Z_2 = 1.1685, x_2 = \; 0.6771\ , s_2 = \ -1$$

$$X_2^* = X_2 - s_1 2^{-2} Y_2 \; \rightarrow \; X_2^* = 0.6057$$

$$Y_2^* = Y_2 - s_1 2^{-2} X_2 \; \rightarrow \; Y_1^* = \; 1.2082$$

$$X_2 = X_2^* - s_1 2^{-1} Y_2^* \; \rightarrow \; X_2 = \; 0.6812$$

$$Y_2 = Y_1^* + s_1 2^{-1} X_1^* \; \rightarrow \; Y_2 \; = \; 1.1704$$

$$Z_2 = Z_1 + s_1\theta_1 \rightarrow Z_1 = 1.0437$$

Now we provide a numerical example for the calculation of arcsine function using the CORDIC algorithm in rotation mode as follows:

For $i = 0$

$$\theta_0 = 0.4636, X_0 = 1, Y_0 = 0, Z_0 = 0, y_0 = 0.5, s_0 = +1$$

$$X_1^* = X_0 - s_0 2^{-1} Y_0 \rightarrow X_1^* = 1$$

$$Y_1^* = Y_0 - s_0 2^{-1} X_0 \rightarrow X_1^* = 0.5$$

$$X_2 = X_1^* - s_0 2^{-1} Y_1^* \rightarrow X_2 = 0.75$$

$$Y_2 = Y_1^* + s_0 2^{-1} X_1^* \rightarrow Y_1 = 1$$

$$Z_1 = Z_0 + s_0\theta_1 \quad \rightarrow Z_1 = 0.9273$$

For $i = 1$

$$\theta_1 = 0.4636, X_1^* = 1, Y_1^* = 0.5 \ X_1 = 1 \ Y_1 = 0 \ Z_1 = 0.9273, y_1 = 0.6250, s_1 = +1$$

$$X_2^* = X_2 - s_1 2^{-2} Y_2 \rightarrow X_2^* = 0.5$$

$$Y_2^* = Y_2 - s_1 2^{-2} X_2 \rightarrow Y_1^* = 1.1875$$

$$X_2 = X_2^* - s_1 2^{-1} Y_2^* \rightarrow X_2 = 0.2031$$

$$Y_2 = Y_1^* + s_1 2^{-1} X_1^* \rightarrow Y_2 = 1.3125$$

$$Z_2 = Z_1 + s_1\theta_1 \quad \rightarrow Z_1 = 1.4173$$

For $i = 2$

$$\theta_2 = 0.1244, X_2^* = 0.5, Y_2^* = 1.1875, \quad X_2 = 0.2031, Y_2 = 1.3125, Z_2 = 1.4173, y_2 = 0.6744, s_2 = +1$$

$$X_2^* = X_2 - s_1 2^{-2} Y_2 \rightarrow X_2^* = 0.0391$$

$$Y_2^* = Y_2 - s_1 2^{-2} X_2 \rightarrow Y_1^* = 1.3379$$

$$X_2 = X_2^* - s_1 2^{-1} Y_2^* \rightarrow X_2 = -0.1282$$

$$Y_2 = Y_1^* + s_1 2^{-1} X_1^* \rightarrow Y_2 = 1.3428$$

$$Z_2 = Z_1 + s_1\theta_1 \quad \rightarrow Z_1 = 1.6660$$

For $i = 3$

$$\theta_2 = 0.0624, X_2^* = 0.0391, Y_2^* = 1.3379 \;\; X_2 = -0.1282\;, Y_2 = 1.3428\;, Z_2 = 1.6660, y_3 = 0.6744\;, s_2 = +1$$

For $i = 3$

$$\theta_2 = 0.1244, X_2^* = 0.5, Y_2^* = 1.1875 \;\; X_2 = 0.2031\;, Y_2 = 1.3125\;, Z_2 = 1.4173, y_2 = 0.6744\;, s_2 = +1$$

$$X_3^* = X_2 - s_1 2^{-2} Y_2 \;\rightarrow\; X_2^* = 0.0391$$

$$Y_3^* = Y_2 - s_1 2^{-2} X_2 \;\rightarrow\; Y_1^* = 1.3379$$

$$X_3 = X_2^* - s_1 2^{-1} Y_2^* \;\rightarrow\; X_2 = -0.1282$$

$$Y_3 = Y_1^* + s_1 2^{-1} X_1^* \;\rightarrow\; Y_2 = 1.3428$$

$$Z_3 = Z_1 + s_1 \theta_1 \qquad \rightarrow Z_1 = 1.6660$$

Now we provide a numerical example for the calculation of arctan function using the CORDIC algorithm in rotation mode as follows:

For$i = 1$

$$\theta_1 = 0.5493, X_1 = 0.5\;, Y_1 = 0.25\;, Z_1 = 0, s_1 = -1$$

$$X_2 = X_1 - s_1 2^{-1} Y_1 \;\rightarrow\; X_2 = 0.0400$$

$$Y_2 = Y_1 - s_1 2^{-1} X_1 \;\rightarrow\; Y_1 = 0.0050$$

$$Z_2 = Z_1 - s_1 \theta_1 \qquad \rightarrow Z_1 = 0.6693$$

For $i = 2$

$$\theta_2 = 0.5493, X_2 = 0.0400\;, Y_2 = 0.0050\;, Z_2 = 0.6693, s_2 = -1$$

$$X_3 = X_2 - s_2 2^{-2} Y_2 \;\rightarrow\; X_3 = 0.0413$$

$$Y_3 = Y_2 - s_2 2^{-2} X_2 \rightarrow\; Y_3 = -0.0050$$

$$Z_3 = Z_2 - s_2 \theta_2 \qquad \rightarrow Z_3 = 0.9247$$

For $i = 3$

$$\theta_3 = \ 0.5493, X_3 = 0.0413, Y_3 = -0.0050\ , Z_3 = 0.9247, s_3 = \ +1$$

$$X_4 = X_3 - s_3 2^{-2} Y_{10} \ \rightarrow \ X_3 = 0.0419$$

$$Z_4 = Z_3 - s_3 \theta_3 \qquad \rightarrow Z_3 \ = \ \ 0.7991$$

For $i = 10$

$$\theta_{10} = \ 0.0010, X_{10} = 0.0420, Y_{10} = 0.0000\ , Z_{10} = \ 0.8021, s_{10} \ = \ -1$$

$$X_{11} = X_{10} - s_{10} 2^{-10} Y_{10} \ \rightarrow \ X_{11} = 0.0420$$

$$Y_{11} = Y_{10} - s_{10} 2^{-10} X_{10} \rightarrow \ Y_{11} \ = \ 0.0000$$

$$Z_{11} = Z_{10} - s_{10} \theta_{10} \qquad \rightarrow Z_{11} \ = \ \ 0.8025$$

3.4. Higher Radix CORDIC Method

The new radix-4 CORDIC method for circular coordinates in rotation mode is presented in this section. This method is an extension of a radix-2 method, and it utilizes powers of four instead of powers of two. The complexity of a radix-4 micro rotations is less than that of a conventional radix-2 micro rotations considering the total number of iterations required for the convergence of the algorithm. The radix-4 CORDIC algorithm is given as

$$X_1 = \frac{1}{K} \tag{3.38}$$

$$Y_1 = 0 \tag{3.39}$$

$$U_1 = 4\theta \ \ 0 < \theta < \frac{\pi}{4} \tag{3.40}$$

$$\theta_i = \tan^{-1}\left(s_i 4^{-i}\right) \tag{3.41}$$

$$X_{i+1} = X_i - s_i 2^{-i} Y_i \tag{3.42}$$

$$Y_{i+1} = Y_i + s_i 2^{-i} X_i \tag{3.43}$$

$$U_{i+1} = 4\,(U_i - 4^i \theta_i) \tag{3.44}$$

$$s_i = \begin{cases} 1 & if\ Z_i \geq 0 \\ -1 & if\ Z_i < 0 \end{cases} \quad i = \ 1,2,3,\cdots,\frac{n}{2} \tag{3.45}$$

$$s_i = \begin{cases} 2 & if\ \frac{21}{8} > U_i \geq \frac{-21}{8} \\ 1 & if\ \frac{5}{8} < U_i < 13/8 \\ 0 & if\ \frac{-5}{8} > U_i < \frac{5}{8} \\ -1 & if\ \frac{13}{8} < U_i < \frac{-5}{8} \\ -2 & if\ \frac{-21}{8} > U_i < \frac{-13}{8} \end{cases} \quad for\ i = \frac{n}{2} + 1, \ldots, \frac{3n}{4} \tag{3.46}$$

Where $s_i = \{-2, -1, 0, 1\ 2\}$ and an elementary angle $\theta_i = \arctan(s_i 4^{-i})$, are the factors used at the i^{th} iteration. The constant term K is calculated depending on the total number of iterations as

$$K = \left(\prod_{i}^{n} \sqrt{1 + \ s_i 4^{-2i}} \right). \tag{3.47}$$

3.4.1. Numerical Example for Cosine and Sine Function Calculation Using Radix-4 CORDIC Algorithm

The numerical example is given about rotation mode of CORDIC methods for the calculation of cosine and sine functions using radix-4 CORDIC algorithm.

For $i = 0$

$$X_0 = 1\ , Y_0 = 0\ , U_0 = 0, s_0 = 2\ and\ \theta_0 = 0.2450$$

$$X_1 = X_0 - s_0 2^{-0} Y_0 \quad \rightarrow X_1 = \ 0.8588$$

$$Y_1 = Y_0 + s_0 2^{-0} X_0 \quad \rightarrow Y_1 = 0.2147$$

$$U_1 = 4\,(U_0 - 4^0\theta_0) \;\rightarrow\; U_1 = 4.4579$$

For $i = 1$

$$X_1 = 0.8588\;, Y_1 = 0.2147\;, U_1 = 4.4579, s_1 = 1, \text{and}\theta_1 = 0.0624$$

$$X_2 = X_1 - s_1 2^{-1} Y_1 \;\rightarrow\; X_2 = 0.8454$$

$$Y_2 = Y_1 + s_1 2^{-1} X_1 \;\rightarrow Y_2 = 0.4831$$

$$U_2 = 4\,(U_1 - 4^1\theta_1) \;\rightarrow\; U_2 = 14$$

For $i = 2$

$$X_2 = 0.8454, Y_2 = 0.4831, U_2 = 14, s_2 = 1, \text{and}\theta_2 = 0.0156$$

$$X_3 = X_2 - s_2 2^{-2} Y_2 \;\rightarrow\; X_2 = 0.8244$$

$$Y_3 = Y_2 + s_2 2^{-2} X_2 \;\rightarrow Y_2 = 0.4961$$

$$U_3 = 4\,(U_2 - 4^2\theta_2) \;\rightarrow\; U_2 = 51$$

For $i = 3$

$$X_3 = 0.8244, Y_3 = 0.4961, U_3 = 51, s_3 = 1, \text{and}\theta_3 = 0.0039$$

$$X_4 = X_3 - s_3 2^{-3} Y_3 \;\rightarrow\; X_4 = 0.8225$$

$$Y_4 = Y_3 + s_3 2^{-3} X_3 \;\rightarrow Y_4 = 0.4993$$

$$U_4 = 4\,(U_3 - 4^3\theta_3) \;\rightarrow\; U_4 = 202$$

For $i = 4$

$$X_4 = 0.8225, Y_4 = 0.4993, U_4 = 202, s_4 = 1, \text{and}\theta_4 = 0.0010$$

$$X_5 = X_4 - s_4 2^{-4} Y_4 \;\rightarrow\; X_5 = 0.8220$$

$$Y_5 = Y_4 + s_4 2^{-4} X_4 \quad \rightarrow Y_5 = 0.5001$$

$$U_5 = 4\ (U_4 - 4^4 \theta_4) \ \rightarrow \ U_5 = 802$$

Hence, we find that $\cos(30) = X_5 = 0.8220$ and $\sin(30) = Y_5 = 0.5001$

3.5. Angle Recoding (AR) Methods

AR Method aims to decrease the number of CORDIC iterations via encoding the angle of rotation as the linear combination of the set of chosen elementary angles of micro-rotations. AR techniques are well-suited for many signal processing and image processing applications where the rotation angle is known priori.

In the traditional CORDIC methods the rotation direction is determined using $s_i = \pm 1$. Hence n CORDIC iterations will be required even if $\theta = 0$, because each time for $s_1 = 1$, or $s_1 = -1$ an iteration is to be calculated. They suggested to relax this constraint via letting $s_i = 0$ so that total number of the CORDIC iterations can be minimized. It is called as Angle Recoding since it resembles a multiplier recoding technique employed in modern multiplier design. Angle according technique can be outlined as:

Initial: $input\ angle\ (S)$

Angel set: $\theta = \mathrm{atan}^{-1}\, 2^{-i}$

Determining minimum angle:

$$\mathrm{for\ index} = 1{:}\,10$$

$$\mathrm{d} = \mathrm{theta(index)}$$

$$\mathrm{if(abs(S - d) < 1000)}$$

$$\mathrm{mymin} = \mathrm{abs}(S - \mathrm{d})$$

i = index;

end

end

When the best angle for rotation is determined, we simply use CORDIC algorithm for the update of parameter values as

$$X_0 = \frac{1}{K}, Y_0 = 0, Z_0 = \theta \ and \ \theta_i = \tan^{-1} 2^{-i}$$

$$X_{i+1} = X_i - s_i 2^{-i} Y_i \tag{3.48}$$

$$Y_{i+1} = Y_i + s_i 2^{-i} X_i \tag{3.49}$$

$$Z_{i+1} = Z_i - s_i \theta_i \tag{3.50}$$

$$s_i = \begin{cases} 1 & if \ Y_i \geq 0 \\ -1 & if \ Z_i < 0 \end{cases} \tag{3.51}$$

3.5.1. Numerical Example on Sine and Cosine Function Calculation Using Angel Recoding Method

A numerical example is given about angle recoding method. The sine and cosine values are calculated for 30 degree.

$$Y_3 = 0, \quad Z_3 = -0.0607, \quad s_3 = -1$$

$$K = \prod_{i=1}^{10} \sqrt{1 + 2^{-2i}} = 1.1644, \ X_1 = \frac{1}{K} = 0.8588$$

For $i = 3$,

$$\theta_0 = 0.1244 \ rd$$

$$X_4 = X_3 - s_3 2^{-3} Y_3 \rightarrow X_3 = 0.8319$$

$$Y_4 = Y_3 - s_3 2^{-3} X_3 \ \rightarrow \ Y_3 = \ 0.5502$$

$$Z_4 = Z_3 - s_3 \theta_3 \qquad \rightarrow Z_3 \ = \ -0.0607$$

For $i = 4$

$$\theta_4 = \ 0.0624\, rd\,, \quad Z_4 < 0, s_4 = \ -1$$

$$X_5 = X_4 - s_4 2^{-4} Y_4 \ \rightarrow \ X_4 \ = 0.8663$$

$$Y_5 = Y_4 - s_4 2^{-4} X_4 \ \rightarrow \ Y_4 = 0.4982$$

$$Z_5 = Z_4 - s_4 \theta_4 \qquad \rightarrow Z_4 \ = \ 0.0017$$

For $i = 5$

$$\theta_5 = 0.0017\, rd, \qquad Z_5 \ < 0, s_5 = \ +1$$

$$X_6 = X_5 - s_5 2^{-5} Y_5 \ \rightarrow \ X_5 = 0.75145$$

$$Y_6 = Y_5 + s_5 2^{-5} X_5 \ \rightarrow \ Y_5 = \ 0.2147$$

$$Z_6 = Z_5 - s_5 \theta_5 \qquad \rightarrow Z_5 \ = \ -0.0434$$

For $i = 6$

$$\theta_6 = 0.0017\, rd, \qquad Z_6 \ < 0, s_6 = \ -1$$

$$X_7 = X_6 - s_6 2^{-6} Y_6 \ \rightarrow \ X_7 = 0.8590$$

$$Y_7 = Y_6 + s_6 2^{-6} X_6 \ \rightarrow \ Y_7 = \ 0.5119$$

$$Z_7 = Z_6 - s_6 \theta_6 \qquad \rightarrow Z_7 \ = \ -0.0139$$

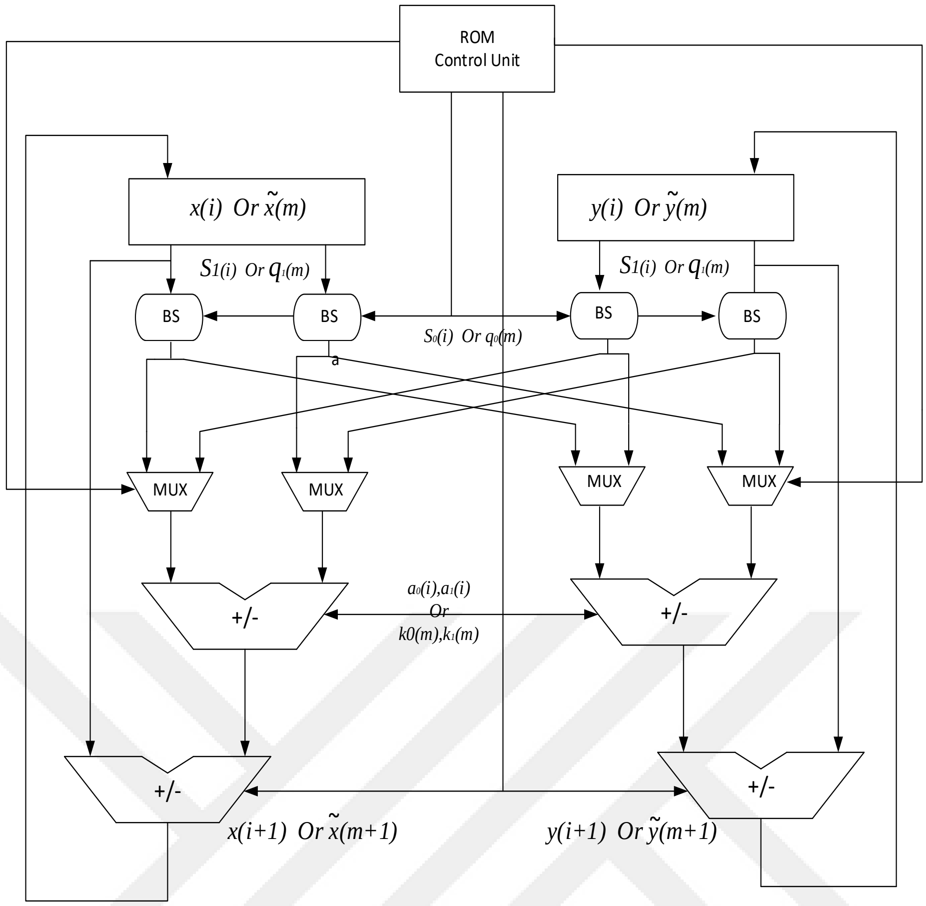

Figure 3.3: Hardware architecture of extended elementary-angle-set recoding [17].

3.6. Extended Elementary-Angle-Set Recoding

In the traditional CORDIC, any given rotation angle is expressed as the linear combination of values of elementary angles that belong to a set. But in AR approaches, this constraint is relaxed via adding zeros to the linear combinations to obtain a desired angle using relatively fewer terms of the form $\sigma \arctan^{-1} 2^{r}$, where $\sigma \in \{-1, 1, 0\}$, and a elementary-angle set is extended further and it is given as:

$$\theta = \arctan(\sigma_1 2^{-r1} + \sigma_2 2^{-r2}) \tag{3.52}$$

EEAS has better recoding efficiency in terms of the number of iterations and it can yield better error performance than the AR system. In EEAS the iterative terms become as in:

$$X_{i+1} = X_i - ((\sigma_1 2^{-r1} + \sigma_2 2^{-r2})Y_i \quad (3.53)$$

$$Y_{i+1} = Y_i + ((\sigma_1 2^{-r1} + \sigma_2 2^{-r2})X_i \quad (3.54)$$

CHAPTER 4

PERFORMANCE EVALUATION

4.1. Introduction

In this chapter we compare different CORDIC algorithms considering the number of iterations required for the convergence of the algorithm.

4.1.1. Traditional Cordic Method (Radix-2)

The performance of radix -2 is calculated. In Figure 4.1, the x-axis represents the computation accuracy or residual tolerated angle and the vertical axis denotes the number of iterations. The rotation angle set is defined as a vector and it is chosen as [45 26.6 14 7.1 3.6 1.8 0.9 0.4 0.2 0.1]. The results apparent in the Figure are summarized as follow:

1. As computation accuracy increases, the number of iteration increases
2. Number of iterations for angle 50 is less compared to another angle
3. As angle diverts or converges from the angle 50, the number of iteration increases.

4.1.2. Cordic Method (Radix-4)

The performance of radix-4 is calculated. In the Figure 4.2, the x-axis represents the computation accuracy or residual tolerated angle and the vertical axis denotes the number of iterations. The rotation angle set is defined as vector and it is chosen as [45 26.6 14 7.1 3.6 1.8 0.9 0.4 0.2 0.1]. The results apparent in the Figure 4.2 are summarized as follow:

1. As computation accuracy increases, the number of iteration increases
2. Number of iterations for angle 50 is less compared to another angle
3. As angle diverts or converges from the angle 50, the number of iteration increases.
4. Number of iterations required for radix-4 is less compared to radix-2

4.1.3. Cordic Method (Angle Recoding)

The performance of angle recording is calculated. From the Figure 4.3, the x-axis represents the computation accuracy or residual tolerated angle and the vertical axis denotes the number of iterations. The rotation angle set is defined as vector and it is chosen as [45 26.6 14 7.1 3.6 1.8 0.9 0.4 0.2 0.1].

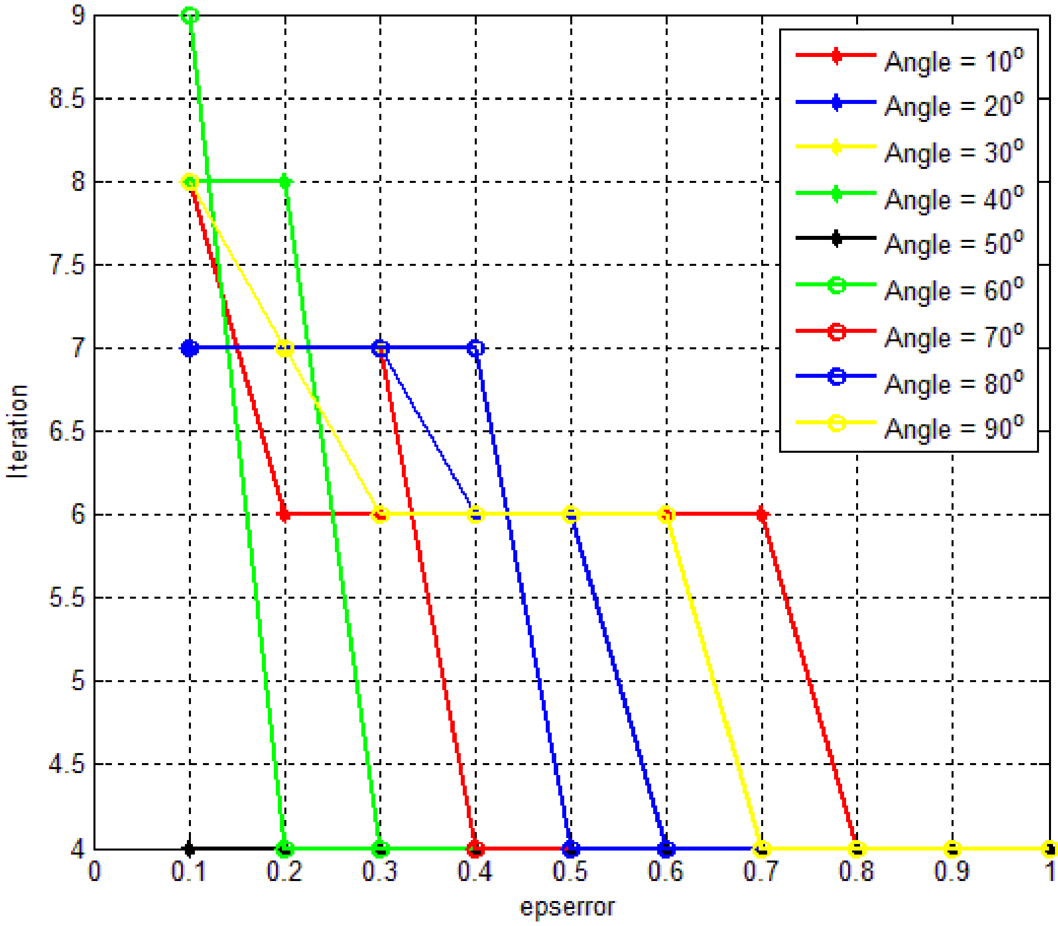

Figure 4.1 Computation accuracy vs number of iterations for radix-2

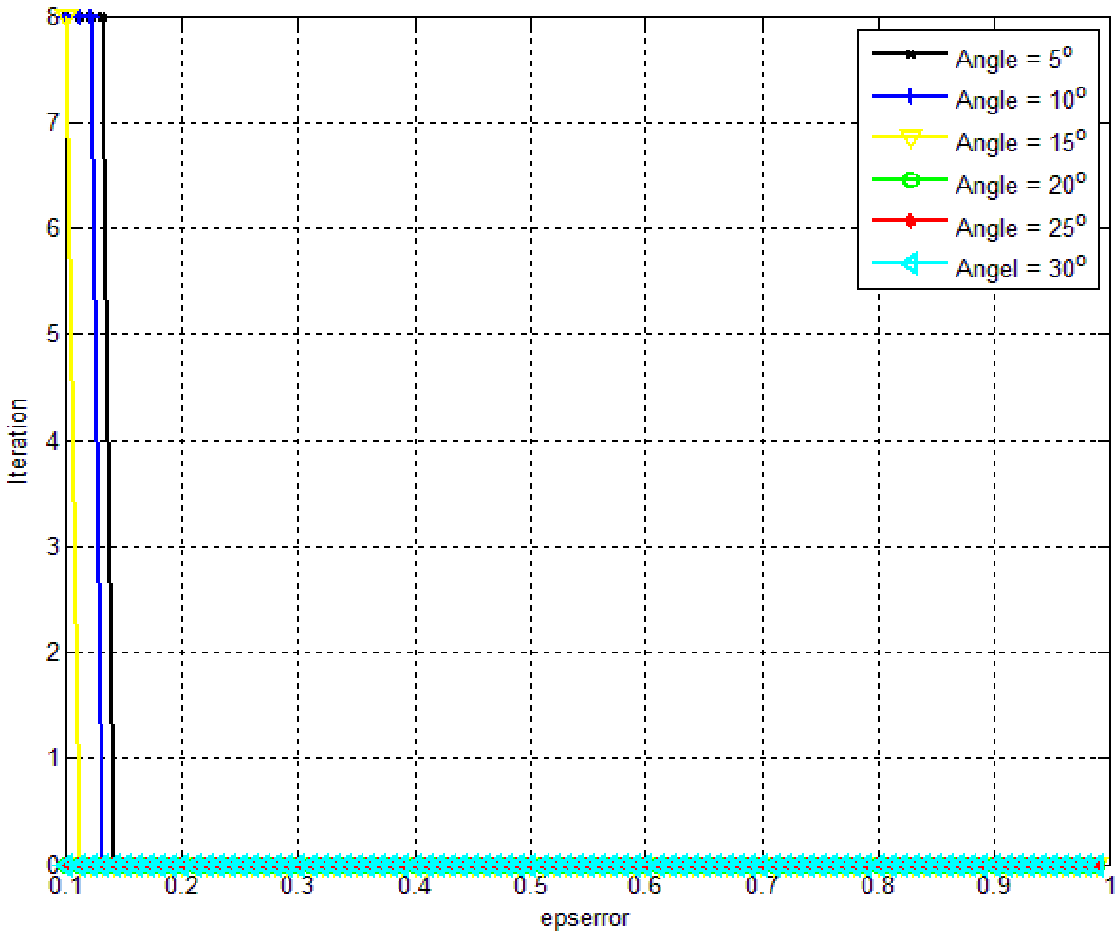

Figure 4.2 Computation accuracy vs number of iterations for radix-4

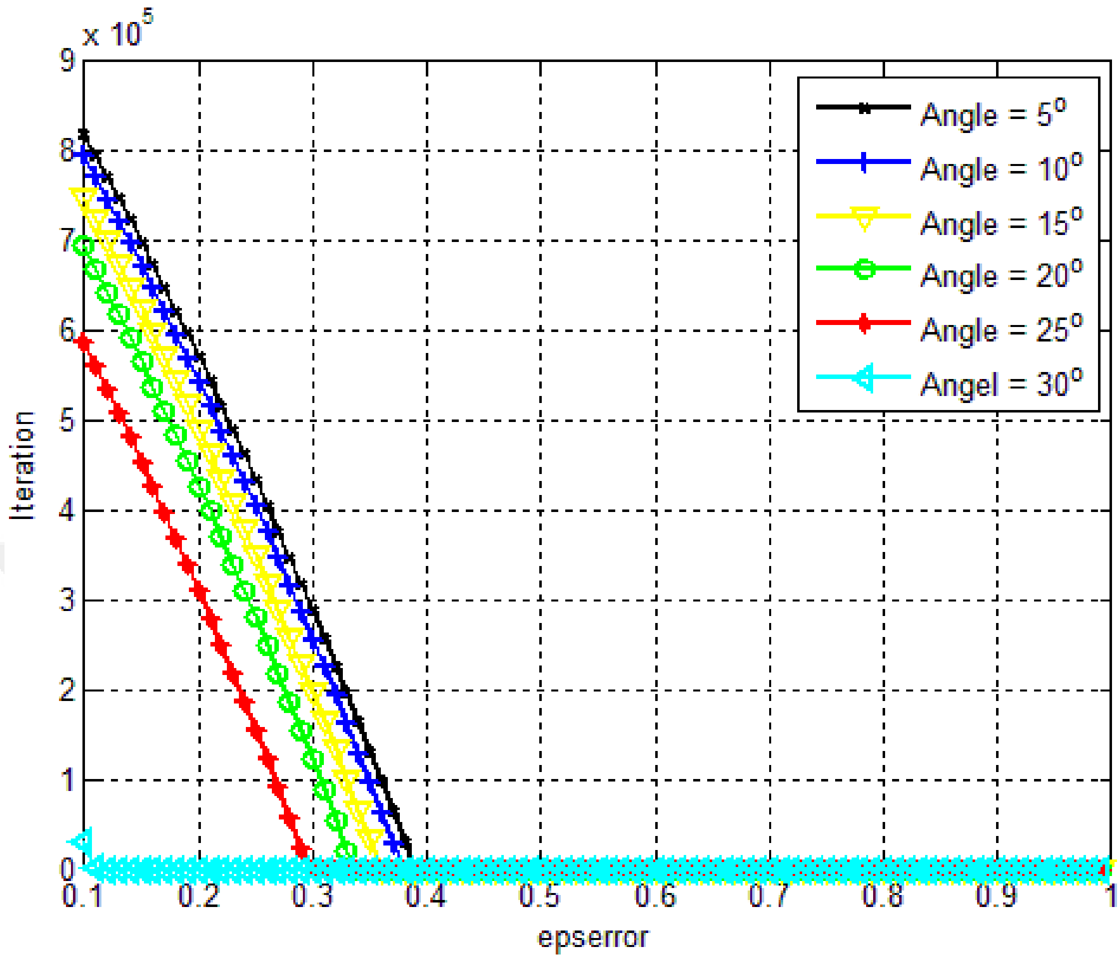

Figure 4.3 Computation accuracy vs number of iterations for angle recoding.

The results apparent in the figure are summarized in the following:

1. As computation accuracy increases, the number of iteration increases
2. Number of iterations for angle 50 is less compared to another angle
3. As angle diverts or converges from the angle 50, the number of iteration increases.
4. Number of iterations required for angle recoding is less compared to radix-4

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

CORDIC algorithm is one of the most widely used function calculation algorithm used in electronic world. Without the invention of CORDIC algorithm the computation of the trigonometric functions would be overhead for the electronic world. CORDIC algorithm is introduced in 1956, since then numerous works have been done for faster and lower complexity CORDIC algorithm. There of these algorithms areRadix-4 CORDIC, Angle Recording, and Extended Angle Recording. In this thesis work, we implemented the rotation mode CORDIC algorithms for the calculation of trigonometric and inverse trigonometric function. The chosen functions are sine, cosine, tangent, arccos, arcsin, acrtan.

We obtained iteration with respect to computation accuracy graphs for all the three techniques we mentioned. We have seen that Radix-4 CORDIC, Angle Recording techniques uses less iterations considering the classical Radix-2 CORDIC algorithm. In current literature it can be seen that researchers still working on lower complexity and faster convergence CORDIC algorithms. In current technology, there seems to be a saturation in the speed of electronic devices. The electronic devices made using semiconductor technology has limited operating speed. For this reason, it is essential to improve the algorithms used for the evaluation of mathematical function in hardware platform.

From the simulation results we see that some of the angles like 50 degree requires less iterations compared to the others.

5.2. Future Work

Optimized hardware implementation of CORDIC algorithm can be stated as a future work. CORDIC algorithm can be implemented in FPGA devices using the VHDL language. However, in this implementation care should be taken for the employment recent and optimized CORDIC algorithms using less iterations and having large accuracy.

REFERENCES

1. **Antelo, E., Villalba, J., Bruguera, J. D., and Zapata, E. L.,** "High performance rotation architectures based on the radix-4 cordicmethod," Computers, IEEE Transactions on, vol. 46, no. 8, pp. 855–870, 1997.

2. **Daggett, D.,** "Decimal-binary conversions in cordic," *Electronic Computers, IRE Transactions on*, no. 3, pp. 335–339, 1959.

3. **Ercegovac, M. D. and Lang, T.,** "Redundant and on-line cordic: Application to matrix triangularization and svd," *Computers, IEEE Transactions on*, vol. 39, no. 6, pp. 725–740, 1990.

4. **Hsiao, S.-F., Hu, Y.-H., and Juang, T.-B.,** "A memory-efficient and high-speed sine/cosine generator based on parallel cordic rotations," *Signal Processing Letters, IEEE,* vol. 11, no. 2, pp. 152–155, 2004.

5. **Hu, Y. H. and Naganathan, S.,** "An angle recoding method for cordicmethod implementation," *Computers, IEEE Transactions on*, vol. 42, no. 1, pp. 99–102, 1993.

6. **Hu, Y. H. and Wu, Z.,** "An efficient cordic array structure for the implementation of discrete cosine transform," *Signal Processing, IEEE Transactions on*,vol. 43, no. 1, pp. 331–336, 1995.

7. **Lang, T. and Antelo, E.,** "Cordic-based computation of arccos and arcsin," in *Application-Specific Schemes, Architectures and Processors, 1997. Proceedings.,IEEE International Conference on*, pp. 132–143, IEEE, 1997.

8. **Mazenc, C., Merrheim, X., and Muller, J.-M.,** "Computing functions cos^*{-1}* and sin^*{-1}* using cordic," *IEEE Transactions on Computers*, vol. 42, no. 1, pp. 118–122, 1993.

9. **Meher, P. K., Valls, J., Juang, T.-B., Sridharan, K., and Maharatna, K.,** "50 years of cordic: Methods, architectures, and applications," *Circuits and Schemes I: Regular Papers, IEEE Transactions on*, vol. 56, no. 9, pp. 1893–1907, 2009.

10. **Rodrigues, T. K. and Swartzlander Jr, E. E.,** "Adaptive cordic: Using parallel angle recoding to accelerate rotations," *Computers, IEEE Transactionson*, vol.59, no. 4, pp. 522–531, 2010.

11. **Takagi, N., Asada, T., and Yajima, S.,** "Redundant cordic methods with a constant scale factor for sine and cosine computation," *Computers, IEEE Transactions on*, vol. 40, no. 9, pp. 989–995, 1991.10.
12. **Volder, J. E., "The cordic trigonometric computing technique,"** *IRE Transactions on Electronic Computers*, vol. 8, pp. 330–334, Sept. 1959.

13. **Walther, J. S.,** "A unified method for elementary functions," in *Proceedings of the May 18-20, 1971, spring joint computer conference*, pp. 379–385, ACM, 1971.

14. **Wang, S., Piuri, V., and Swartzlander Jr, E. E.,** "Hybrid cordicmethods," *IEEE Transactions on Computers,* no. 11, pp. 1202–1207, 1997.

15. **Wu, C.-S. and Wu, A.-Y.,** "Modified vector rotational cordic (mvr-cordic) method and architecture," *Circuits and Schemes II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 48, no. 6, pp. 548–561, 2001.

16. **Wu, C.-S., Wu, A.-Y., and Lin, C.-H.,** "A high-performance/low-latency vector rotational cordic architecture based on extended elementary angle set and trellis-based searching schemes," *Circuits and Schemes II: Analog and Digital Signal Processing, IEEE Transactions on*, vol. 50, no. 9, pp. 589–601, 2003.

17. **Yu, S. and Swartzlander Jr, E. E.,** "A scaled dct architecture with the cordicmethod," *Signal Processing, IEEE Transactions on*, vol. 50, no. 1, pp. 160–167, 2002.
18. **J. Han and M. Orshansky,** "Approximate computing: An emerging paradigm for energy-efficient design," in Proc. 18thIEEE European Test Symposium (ETS), 2013, pp. 1-6.

19. **H. Jiang, J. Han, and F. Lombardi,** "A Comparative Review and Evaluation of Approximate Adders," in Proc. 25th Great Lakes Symposium on VLSI, Pittsburgh, Pennsylvania, USA, 2015, pp. 343-348.

20. **J. Liang, J. Han, and F. Lombardi,** "New Metrics for the Reliability of Approximate and Probabilistic Adders," IEEE Trans. on Computers, vol. 62, pp. 1760-1771, 2013.

21. **M. J. Schulte and E. E. Swartzlander,** "Truncated multiplication with correction constant," VLSI Signal Processing VI, pp. 388-396, 1993

22. **J. E. Volder,** "The CORDIC Trigonometric Computing Technique," IRE Trans. on Electronic Computers, vol. EC-8, pp. 330-334, 1959.

APPENDICESA

CURRICULUM VITAE

PERSONAL INFORMATION

Surname,Name: ALNAFUTCHY, Ameen Mustafa Mohammed

Nationality: Iraqi (IQ)

Date and Place of Birth: 15January 1986, Kirkuk

Marital Status: Single

Phone: +90 5314304551

E-mail: ameen.alnafutchy@gmail.com

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya University Electronic and Communication Engineering	July, 2018
B.Sc.	Technical university– Kirkuk	June, 2007
High School	Al-Waleed – Kirkuk	June, 2003

FOREIGN LANGUAGE

Arabic - fluent

English - very good

Turkish - fluent