Design and Analysis of Extension-Rotation CORDIC Algorithms based on Non-Redundant Method

Pongyupinpanich Surapong, Faizal Arya Samman and Manfred Glesner
Microelectronic Systems Research Group,
Technische Universität Darmstadt, Darmstadt, Germany
Email: {surapong; faizal.samman; glesner}@mes.tu-darmstadt.de

Abstract

In this paper, rotation-extension CORDIC methods, i.e. double-rotation and triple-rotation, are proposed for the objective of improving the performance and accuracy of the CORDIC computational algorithm in radix-2. In the double-rotation and triple-rotation methods, the convergences of the CORDIC computations are accelerated by duplicating and triplicating the micro-rotation angles to be 2θ and 3θ, respectively. The non-redundant mechanism, where a rotation direction δ is in a set of 1 and -1, depending on an intermediate converging parameter (either y or z), is applied to constant scaling factors. Convergence range and accuracy of elementary functions hardware performed by using the CORDIC methods in rotation mode and vectoring mode on the circular, hyperbolic, and linear coordinate systems are examined, investigated and compared to Matlab/Simulink simulation results. The comparison results show that the proposed CORDIC methods provide higher computational accuracy than the conventional one at the same number of iterations. A high precision CORDIC algorithm is introduced and evaluated for VLSI implementation. Finally, speed and area performance of the CORDIC hardware based on the pipeline (unfolded) digit-parallel architecture of the proposed CORDIC methods are compared to the CORDIC methods previously published in the literature.

1. Introduction

The COordinate Rotation DIgital Computer (CORDIC) is known as an iterative algorithm using only shift-and-add operations to perform several mathematic functions for scientific and engineering fields. CORDIC was firstly described in 1959 by J.E. Volder [1] as an elegant way to evaluate trigonometric functions. It was originally developed to replace the analog navigation computer on the B-58 aircraft bomber due to a need for higher performance and higher accuracy [2]. In 1971, J. Walther [3] extended the CORDIC algorithm to hyperbolic functions and the algorithm is today used in many application areas such as matrix computation, digital signal processing, digital image processing, communication, robotics and graphics [4]. The key concept of the algorithm is based on rotation of a two dimensional (2-D) (x,y) vector in circular, hyperbolic, and linear coordinate systems. J.E. Volder described and implemented the iterative formulation of a computational algorithm for the CORDIC in trigonometric functions, division and multiplication. It became more and more attractive when J. Walther showed that by modifying a few parameters, it can be used to perform a single algorithm for unified implementation of a wide range of elementary transcendental functions related to Logarithms, Exponentials, Square Root, etc. Meanwhile, D. Cochran [5] gave examples of various algorithms which present that the CORDIC technique is a better choice for scientific calculating applications. Its popularity

was much more increased thereafter primarily due to its potential for efficient and low-cost implementation. Some of the popular applications are: (1) direct digital frequency synthesis [6], digital modulation and coding for speech/music synthesis and communication; (2) direct and inverse kinematics computation for robot manipulation; and (3) planar and three-dimensional (3-D) vector rotation for graphics and animation [7].

The development of the CORDIC algorithm and architectures [8] has taken place for achieving the highest throughput rate and reduction of hardware-complexity as well as the computational latency of implementation. Some of the typical approaches for reducing-complexity implementation are targeted on minimization of using the scaling-operation and complexity of barrel-shifters and adders in the CORDIC engine. The inherent drawback of the conventional CORDIC algorithm is computational latency. Angle recording schemes, mixed-grain rotation and higher radix CORDIC have been developed to reduce the latency. Parallel and pipelined techniques have been suggested for high-throughput computation. In this paper, we focus on two subjects, i.e. (1) reduction of computational latency and improving the computational accuracy of the CORDIC algorithm based-on micro-rotation duplication and triplication methods, and (2) design and architecture of a high precision CORDIC architecture.

The rest of the paper is organized as follows. The related works are discussed in Section 2. Section 3 explains the contribution of this paper. The rotation-extension CORDIC method, i.e. the double-rotation and triple-rotation methods, are proposed in Section 4. The unified CORDIC algorithm and a high precision CORDIC algorithm are introduced and discussed in Section 5. Section 6 considers the algorithm and time investigation of the high precision CORDIC core. Finally, concluding remarks are presented in Section 7.

2. Related Works

There are various proposed methods for the high performance CORDIC algorithm, reducing the computational latency and increasing the computational accuracy, i.e. (1) high radix method, (2) parallel rotation method, (3) redundant number representation, and (4) rotation extension method.

2.1. High Radix CORDIC method

In 1996, E. Antelo et al. [9] proposed the mixed radix-2 and radix-4 method for a redundant CORDIC processor implemented in pipelining architecture. The combination of the two radix can reduce latency and area of the pipelining stats. The full radix-4 and very-high CORDIC method with on-line scaling factor computation for high performance rotation architectures [10] [11] were introduced. Meanwhile, C.C. Li [12] suggested the fast on-line scaling factor compensation for redundant CORDIC algorithm in radix-4 and simplified the on-line decomposition algorithm in hardware. J.Villalba [13] introduced an extension version of the radix-4 CORDIC method in vectoring mode, where the selection method is applied for non-redundant and redundant computation. The constant scaling factor for radix 2-4-8 CORDIC algorithm was investigated by T. Aoki et al. [14]. The algorithm dynamically changed from radix-2 to radix-4 and from radix-4 to radix-8 depending on the specified sequence of CORDIC algorithm. The constant scaling factor of each radix was expressed as $K_c = \prod_{i=1}^{N} \sqrt{1 + \delta_i^2 \cdot r^{-2 \cdot i}}$, where K_c is a constant scaling factor, N is the maximum number of iterations, δ_i is rotation direction of micro-rotation of CORDIC, r_i is the radix-based determined by i^{th} iteration. The articles dealing with the high radix CORDIC method shows that the scaling factor used for computation and compensation becomes a main problem. They attempt to solve this problem by proposing the on-line scaling

factor computation method and by finding a range of the computation in order to constant the scaling factor on radix-2-4-8 CORDIC algorithm. However, the CORDIC method used for the constant scaling factor is too complex and hard for hardware implementation.

2.2. Parallel CORDIC rotation method

The parallel CORDIC method was proposed by T.B. Juang et al. [15]. The algorithm predicts the rotation direction δ_i from binary value of initial angle. The original sequential CORDIC rotations is divided into two phase, where the rotation in each phase is executed in parallel. S.F. Hsiao [16] applied the method to implement a Sine/Cosine generator with memory-efficient concept; meanwhile W.Han et al. [17] implemented a digital down converter based on the parallel CORDIC algorithm. Thereby, the method can provide a constant scaling factor because the two parallel processing cores are based on the conventional CORDIC. Although, the scaling factor has been solved by using the parallel technique, the overhead of summation and prediction of the two parallel CORDIC engines and rotation direction are aggravated.

2.3. Redundant Number Representation Method

The third efficient way to accelerate the CORDIC algorithm is to use redundant number representation, such as singed-digit (SD) representation and carry-save representation. In 1990, M.D. Ercegovac [18] presented redundant (carry-free) addition instead of a conventional (carry-propagate) one and on-line scaling factor computation method in order to minimize area and to increase bandwidth. But, the on-line computation method is too complex which is very difficult in practice. To ignore the on-line scaling factor computation method, N. Takagi et al. [19] introduced the redundant CORDIC method with a constant scaling factor. Thereby, the scaling factor is investigated by minimizing the error from the CORDIC computation.

2.4. Rotation Extension Method

From the reviewed articles, the reduction of the CORDIC computational latency by the high radix method makes the scaling factor in instability situation. The unstable scaling factor problem can be solved by the on-line computation method which is too complex for hardware practice. The parallel technique is used to accelerate the computation. The method is based on the two conventional CORDIC cores processing in parallel. Thereby, the scaling factor problem is solved, but the prediction overhead of the rotation direction and the combination of the two conventional CORDIC results are added. The extension-rotation method based on radix-2 accelerates the micro-rotation angle ϕ with $n \cdot \phi$, where n is an integer value. The double-rotation method performs the computational results with micro-rotation angle 2ϕ. Its scaling factor is first calculated by on-line computation method and then by optimizing the errors of the computational result in order to constant a scaling factor.

This paper is focused on the reducing computational latency and the improving computational accuracy of CORDIC with the constant scaling factor based on radix-2. The radix-2 is used because the scaling factor can be found mathematically. The design and architecture of the CORDIC algorithm on the radix-2 is straightforward for hardware implementation. We propose both the rotation-extension CORDIC methods, i.e. non-redundant double-rotation and non-redundant triple-rotation. They are analyzed, investigated, and simulated to evaluate the constant scaling factors and the input domain capability for elementary functions. The initial parameter values and the compensation parameter values for each elementary function, which can be performed in circular, hyperbolic, and linear

coordinate systems are also proposed. We compare the computational accuracy of the elementary functions with Matlab/Simulink in order to show that the proposed CORDIC methods provide high computational accuracy. Finally, a high accuracy CORDIC algorithm and its architecture is introduced for VLSI hardware implementation, and also the time and area efficiency based on FPGA and Silicon technology are compared with the existing CORDICs.

3. Contribution

The reducing computational latency and the increasing computational accuracy of CORDIC become a challenge for engineers and scientists. Recently, they extremely attempt to improve the computational performance and accuracy by proposing several methods such as the parallel and prediction CORDIC (PCORDIC) [16] [15], the double-rotation CORDIC (DRCORDIC), etc. The double-rotation method with redundant method was proposed by Takagi et al. [19] in rotation mode only; thereby, the convergence of CORDIC parameters is accelerated by duplicate rotation angle θ to be 2θ. They used few MSB for prediction of rotation angle and attempted to constant the scaling factor. Similarly, we apply Takagi's idea for our double-rotation CORDIC method, but the difference is that the non-redundant method is utilized in order to constant the scaling factor, and the operation on vectoring mode is extended. In addition, we propose the triple-rotation CORDIC method where the micro-rotation angle is triplicated from the conventional CORDIC angle θ to be 3θ.

This paper provides contributions as follow:

- **Improvement of performance and accuracy of CORDIC computation:** the convergence of CORDIC parameters is accelerated by increasing the micro-rotation angle to be 2θ and 3θ for the double-rotation and triple-rotation CORDIC methods. The non-redundant method is applied in order to constant the scaling factor mathematically.
- **Analysis of the double-rotation and triple-rotation CORDIC methods:** the unified micro-rotation equations for the double-rotation and triple-rotation CORDIC methods based on radix-2 are proposed, where they are targeted for hardware simplification. The convergence ranges and the initial parameters which will be used to perform elementary functions on rotation and vectoring modes in the circular mode such as sine and cosine functions, in hyperbolic mode such as sine and cosine hyperbolic functions, and linear mode such as multiplication and division functions are examined and comprehensively summarized for VLSI implementation.
- **Design and time investigation of high accuracy CORDIC arithmetic units:** an algorithm and a computational latency model of a high precision CORDIC arithmetic unit is introduced for simple VLSI implementation. The model can be applied to estimate the computational delay and accuracy of the high precision CORDIC core conforming to the expected absolute error.

4. Rotation-Extension CORDIC Algorithm

The algorithm is based on the rotation of a vector on the xy plane. As shown in Figure 1, a vector x_{in}, y_{in} having angle A is rotated by angle B, resulting in a new vector x_{out}, y_{out}. This rotation is described by the expressions shown in Equ. 1 [20]

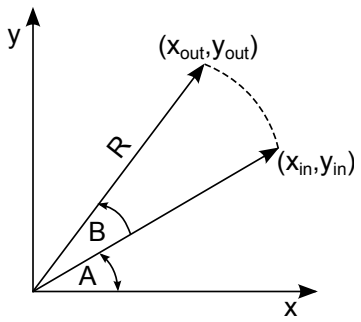


Figure 1: Vector rotation

$$\begin{aligned} x_{out} &= R \cdot cos(A+B) = x_{in} \cdot cos\, B - y_{in} \cdot sin\, B \\ y_{out} &= R \cdot sin(A+B) = x_{in} \cdot sin\, B - y_{in} \cdot cos\, B, \end{aligned} \tag{1}$$

where R is the modulus of the vector and A is the initial angle. The rotation can be described in matrix form as

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = \begin{bmatrix} cos\, B & -sin\, B \\ sin\, B & cos\, B \end{bmatrix} \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{2}$$

Instead of B as presented in Equ. (2), we use notation ϕ. The basic idea dealing with the rotation-extension CORDIC computation method is described in Equ. (3), where x_{in}, y_{in}, and ϕ are input parameters and r is a rotation degree. For the sake of simplicity, we derive and consider all equations based on the circular coordinate system.

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = \begin{bmatrix} cos\, \phi & -sin\, \phi \\ sin\, \phi & cos\, \phi \end{bmatrix}^r \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{3}$$

4.1. Conventional CORDIC

The algorithm performs a sequence of rotations by elementary angles. For the conventional CORDIC method, the rotation ϕ on the xy plan can be decomposed into a product of n elementary rotations when $r = 1$ as:

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = cos\, \phi \begin{bmatrix} 1 & -tan\, \phi \\ tan\, \phi & 1 \end{bmatrix} \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix}, \tag{4}$$

where $cos\, \phi$ is a constant scaling factor. The elementary rotation matrix $R(\delta_i)$ describes the elementary rotation with angle $\theta_i = tan^{-1}(2^{-i})$ while i is iteration step. $\delta_i \in \{1, -1\}$ determines the rotation direction.

$$\prod_{i=0}^{n-1} R(\delta_i) = \prod_{i=0}^{n-1} \frac{1}{\sqrt{1+2^{-2i}}} \begin{bmatrix} 1 & -\delta_i 2^{-i} \\ \delta_i 2^{-i} & 1 \end{bmatrix} \tag{5}$$

Since the rotation angle θ_i is always a constant value (based on a non-redundant method [21]), the constant scaling factor $cos\ \phi$ will be $\prod_{i=0}^{n-1} \frac{1}{\sqrt{1+2^{-2i}}}$, which approximately equals to 0.60725. The maximum and minimum rotation angle are defined as $\pm \sum_{i=0}^{n-1} tan^{-1}(2^{-i})$, which is in range of [-1.74329,+1.74329]. The micro-rotation equation of the conventional CORDIC method is presented in Equ. (6).

$$\begin{aligned} x_{i+1} &= x_i - \delta_i \cdot 2^{-i} \cdot y_i \\ y_{i+1} &= y_i + \delta_i \cdot 2^{-i} \cdot x_i \\ z_{i+1} &= z_i - \delta_i \cdot \theta_i \end{aligned} \tag{6}$$

4.2. Double-Rotation CORDIC

The aim of the double-rotation method is to accelerate the rotation computation of the CORDIC algorithm by duplicating the elementary angle to be $2\theta_i$. Thus, the degree of parameter r in Equ. (3) is equal to 2.

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = cos^2\,\phi \begin{bmatrix} 1 & -tan\,\phi \\ tan\,\phi & 1 \end{bmatrix}^2 \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{7}$$

The decomposition of the production of n elementary rotation is shown in Equ. (8) with $\theta_i = tan^{-1}(2^{-i-1})$.

$$\prod_{i=1}^{n} R(\delta_i) = \prod_{i=1}^{n} \frac{1}{(1+2^{-2i-2})} \begin{bmatrix} 1 & -\delta_i \cdot 2^{-i-1} \\ \delta_i \cdot 2^{-i-1} & 1 \end{bmatrix}^2 \tag{8}$$

To stabilize the scaling factor, the non-redundant number, where $\delta_i \in \{-1, 1\}$, is applied for the double-rotation CORDIC method. Thereby, the optimal constant scaling factor is formulated by $\prod_{i=1}^{n} \frac{1}{(1+2^{-2i-2})}$ which approximately equals to 0.9219. The maximum and minimum rotation angle $\sum_{i=1}^{n} 2 \cdot tan^{-1}(2^{-i-1})$ is in range of [-0.9885,+0.9885]. The micro-rotation equation of the double-rotation CORDIC method is shown in Equ. (9).

$$\begin{aligned} x_{i+1} &= x_i - \delta_i \cdot 2^{-i} \cdot y_i - 2^{-2i-2} \cdot x_i \\ y_{i+1} &= y_i + \delta_i \cdot 2^{-i} \cdot x_i - 2^{-2i-2} \cdot y_i \\ z_{i+1} &= z_i - \delta_i \cdot 2 \cdot \theta_i \end{aligned} \tag{9}$$

4.3. Triple-Rotation CORDIC

Like the conventional and double-rotation methods, the triple-rotation method can accelerate a micro rotation θ_i by triplicating the elementary angle θ_i to be $3\theta_i$.

$$\begin{bmatrix} x_{out} \\ y_{out} \end{bmatrix} = cos^3\,\phi \begin{bmatrix} 1 & -tan\,\phi \\ tan\,\phi & 1 \end{bmatrix}^3 \begin{bmatrix} x_{in} \\ y_{in} \end{bmatrix} \tag{10}$$

The decomposition of the product of n elementary rotation is described in Equ. (11), where $r = 3$, representing the triplicated rotation.

$$\prod_{i=2}^{n+1} R(\delta_i) = \prod_{i=2}^{n+1} \frac{1}{(1+2^{-2i-4})^{\frac{3}{2}}} \begin{bmatrix} 1 & -\delta_i 2^{-i-2} \\ \delta_i 2^{-i-2} & 1 \end{bmatrix}^3 \tag{11}$$

The micro-rotation angle is $\theta_i = tan^i(2^{-i-2})$. The non-redundancy for rotation direction is also applied. Thereby, the scaling factor is approximately constant at 0.9922, which is calculated from $\prod_{i=2}^{n+1} \frac{1}{(1+2^{-2i-4})^{\frac{3}{2}}}$. The maximum and minimum rotation angle can be found by $\sum_{i=2}^{n+1} 3 \cdot tan^{-1}(2^{-i-2})$, which is in range of [-0.3747, +0.3747]. The micro-rotation equation of the triple-rotation CORDIC method being simplified for VLSI hardware implementation is shown in Equ. (12).

$$\begin{aligned} x_{i+1} &= x_i(1 - 2^{-2i-3} - 2^{-2i-4}) - \delta_i(2^{-i-1} + 2^{-i-2} - 2^{-3i-6})y_i \\ y_{i+1} &= \delta_i x_i(2^{-i-1} + 2^{-i-2} - 2^{-3i-6}) + (1 - 2^{-2i-3} - 2^{-2i-4})y_i \\ z_{i+1} &= z_i - \delta_i \cdot 3 \cdot \theta_i \end{aligned} \tag{12}$$

4.4. Accuracy Evaluation

The computational accuracy can be considered on either rotation mode or vectoring mode. In this paper, the iterative CORDIC in rotation mode on the circular coordinate system is taken into account, where parameter z_i will be driven to 0. The summation formula of parameter z on each CORDIC methods are shown as following:

$$\begin{aligned} z_{out-CV} &= z_{in} - \sum_{i=0}^{n-1} \delta_i \cdot tan^{-1}(2^{-i}) \\ z_{out-DR} &= z_{in} - \sum_{i=1}^{n} \delta_i \cdot 2 \cdot tan^{-1}(2^{-i-1}) \\ z_{out-TR} &= z_{in} - \sum_{i=2}^{n+1} \delta_i \cdot 3 \cdot tan^{-1}(2^{-i-2}) \end{aligned} \tag{13}$$

The condition that the triple-rotation and double-rotation CORDIC methods have more precision than the conventional one can be expressed as:

$$\begin{aligned} \left| z_{in} - \sum_{i=2}^{n+1} \delta_i \cdot 3 \cdot tan^{-1}(2^{-i-2}) \right| &> \left| z_{in} - \sum_{i=1}^{n} \delta_i \cdot 2 \cdot tan^{-1}(2^{-i-1}) \right| \\ &> \left| z_{in} - \sum_{i=0}^{n-1} \delta_i \cdot tan^{-1}(2^{-i}) \right| \end{aligned} \tag{14}$$

From Equ. 14 the rotation direction δ on each iteration depends on an intermediate parameter z_i of each method. Analysis based on mathematic assumption could not guarantee the computational accuracy due to nonlinear equation. Therefore, the accuracy evaluation by using simulation based on statistical analysis in mathematic which is Mean Absolute Percent Error and standard statistic can be applied in practice.

Table 1: Probability of rotation direction δ of the conventional, double-rotation, and triple-rotation CORDIC methods, where z_{in} is varied from 0.0 to 0.3

CORDIC method	δ		
	-1	0	1
Conventional	0.5020	0	0.4980
Proposed Double-rotation	0.5051	0.1434	0.4734
Proposed Triple-rotation	0.5051	0	0.4949

The Mean Absolute Percent Error (MAPE) mathematical factor is employed for accuracy evaluation in available range [-1.74329, +1.74329], [-0.9885,+0.9885], and [-0.3747, 0.3747] of the conventional, double-rotation, and triple-rotation CORDIC methods. The simulation results show that the proposed CORDIC methods provide the smallest MAPE at the same number of iterations, which implies better accuracy as illustrated in Table 2. The MAPE is defined by the formula:

$$M = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - F_i}{\bar{A}}\right|, \tag{15}$$

where A_i is the actual value and F_i is the forecast value. The difference between A_i and F_i is divided by the average actual value $\bar{A}$ again. The absolute value of this calculation is summed for every forecast point in time and divided again by the number of fitted points n.

Table 2: The MAPE comparison of x_I and y_I of the conventional, double-rotation and triple-rotation CORDIC methods, where the iteration step I equals to 8, 10, and 16.

	Mean absolute percent error (MAPE)		
CORDIC	I=8	I=10	I=16
x_I, $\cos(\phi)$			
Conventional	0.0597%	0.0153%	2.2425E-4%
Double-rotation	0.0297%	0.0086%	1.1735E-4%
Triple-rotation	0.0225%	0.0060%	8.0427E-5%
y_I, $\sin(\phi)$			
Conventional	0.0481%	0.0131%	1.6301E-4%
Double-rotation	0.0232%	0.0057%	9.2696E-5%
Triple-rotation	0.0180%	0.0050%	9.3906E-5%

Four standard statistical indicators are employed for accuracy evaluation of the proposed CORDIC computational methods, i.e. maximum absolute error ($Max.|error|$), minimum absolute error ($Min.|error|$), average absolute error ($Ave.|error|$), and standard deviation absolute error ($Ave.\,Dev.|error|$). All formulas are expressed as follow:

Maximum absolute error ($Max.|error|$)

$$Max.|error| = Max.\{|A_0 - F_0|, \cdots, |A_N - F_N|\} \tag{16}$$

Minimum absolute error ($Min.|error|$)

$$Min.|error| = Min.\{|A_0 - F_0|, \cdots, |A_N - F_N|\} \tag{17}$$

Average absolute error ($Ave.|error|$)

$$Ave.|error| = \frac{1}{N}\sum_{i=0}^{N}|A_i - F_i| \tag{18}$$

Standard deviation absolute error ($Ave.\,Dev.|error|$)

$$\begin{aligned} X_i &= |A_i - F_i| \\ \bar{X} &= \frac{1}{N}\sum_{i=0}^{N}|A_i - F_i| \\ Ave.\,Dev.|error| &= \sqrt{\frac{1}{N}\sum_{i=0}^{N}(X_i - \bar{X})^2} \end{aligned} \tag{19}$$

The analysis results with the four statistical indicators are shown in Table 3 and 4. Both tables show the comparison of the computational error of the conventional, double-rotation and triple-rotation CORDIC methods based on Matlab simulation in the various number of iterations(N) at 8, 10, 16, 32, and 64. For measuring environment, since we want to measure the computational accuracy on the three parameters, i.e. x_{out}, y_{out}, and z_{out}, the CORDIC methods have to be performed on both rotation mode and vectoring mode. Then, the circular coordinate system is selected for our test case. Table 3 reports the analysis result of the CORDIC methods in rotating mode performed by function number 1 in Table 5. Similarly, the analysis result in vectoring mode of the CORDIC method performed by function number 3 is illustrated in Table 4. From the tables, the double-rotation and triple-rotation CORDIC methods still provide higher accuracy than the conventional one when they are analyzed by the statistical measurements.

Table 3: Show the analysis of computational accuracy of CORDIC methods in rotation mode on the circular coordinate system.

	Iteration (N)	CORDIC	Statistical Analysis			
			$Max.\|error\|$	$Min.\|error\|$	$Ave.\|error\|$	$Std.Dev.\|error\|$
x_{out}	8	Conventional	9.4966E-3	3.2120E-5	4.5125E-3	2.6722E-3
		Double-rotation	4.8971E-3	1.5988E-5	2.3296E-3	1.3431E-3
		Triple-rotation	3.6940E-3	2.2748E-5	1.7411E-3	1.0119E-3
	10	Conventional	2.3952E-3	4.7574E-6	1.1589E-3	6.6409E-4
		Double-rotation	1.2320E-3	6.9294E-6	5.7707E-4	3.3635E-4
		Triple-rotation	9.1174E-4	5.6572E-7	4.3358E-4	2.5278E-4
	16	Conventional	3.7365E-5	1.2253E-7	1.8090E-5	1.0543E-5
		Double-rotation	1.8925E-5	8.1072E-9	9.1003E-6	5.3043E-6
		Triple-rotation	1.4181E-5	2.6530E-8	6.7987E-6	3.9405E-6
	32	Conventional	5.8888E-10	4.6108E-13	2.7506E-10	1.6137E-10
		Double-rotation	2.9457E-10	2.9358E-12	1.3889E-10	8.0351E-11
		Triple-rotation	2.1399E-10	1.0505E-12	1.0593E-10	5.9295E-11
	64	Conventional	9.9920E-16	3.7453E-15	5.8693E-15	3.6346E-16
		Double-rotation	3.4417E-15	1.5543E-15	2.4920E-15	2.1890E-16
		Triple-rotation	1.5543E-15	1.1102E-16	7.1996e-16	2.8387E-16
y_{out}	8	Conventional	6.748E-3	1.7481E-5	2.9117E-3	1.7723E-3
		Double-rotation	3.3532E-3	8.9682E-6	1.4952E-3	8.7957E-4
		Triple-rotation	2.5376E-3	1.5211E-5	1.1163E-3	6.5960E-4
	10	Conventional	1.6862E-3	3.3777E-6	7.4364E-4	4.3493E-4
		Double-rotation	8.2529E-4	3.9484E-6	3.7026E-4	2.1735E-4
		Triple-rotation	6.1626E-4	4.1980E-7	2.7813E-4	1.6486E-4
	16	Conventional	2.6937E-5	9.8153E-8	1.1614E-5	6.8945E-6
		Double-rotation	1.3301E-5	1.6685E-8	5.8318E-6	3.4532E-6
		Triple-rotation	9.6708E-6	4.4125E-9	4.3576E-6	2.5698E-6
	32	Conventional	4.0018E-10	1.9198E-12	1.7623E-10	1.0478E-10
		Double-rotation	2.0294E-10	5.1292E-13	8.9197E-11	5.2357E-11
		Triple-rotation	1.5010E-10	2.9110E-13	6.8012E-11	3.8826E-11
	64	Conventional	5.7732E-15	4.4580E-15	4.0651E-15	5.3280E-16
		Double-rotation	2.4425E-15	2.6645E-15	9.8142E-16	5.0489E-16
		Triple-rotation	1.7764E-15	1.0012E-15	4.2967E-16	3.1539E-16

4.5. Convergence & Accuracy Trade-Off

The performance and time efficiency of the double-rotation and triple-rotation CORDIC methods can be investigated and evaluated by two methods based on the conventional CORDIC and Matlab simulation. First, determination of absolute error constraint method: thereby, the absolute error is given as a different expected error ΔE of a CORDIC computational result and a Matlab simulation result. If the actual different error Δe of a CORDIC

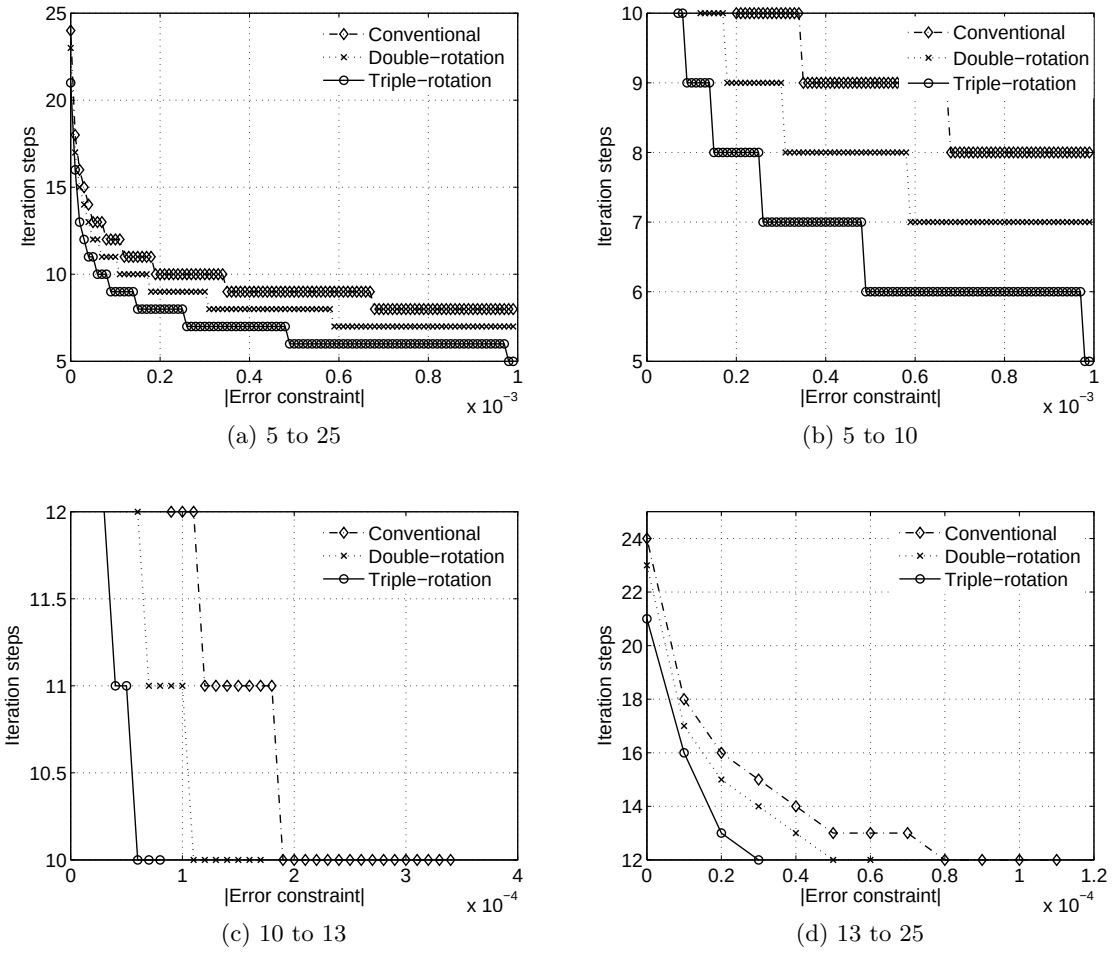
Table 4: The analysis of computational accuracy of CORDIC methods in vectoring mode on the circular coordinate system.

	Iteration (N)	CORDIC	Statistical Analysis			
			$Max.\|error\|$	$Min.\|error\|$	$Ave.\|error\|$	$Std.Dev.\|error\|$
x_{out}	8	Conventional	2.1886E-5	5.8874E-8	9.577E-6	9.2267E-6
		Double-rotation	6.0890E-6	3.2259E-6	3.5329E-6	2.0376E-6
		Triple-rotation	2.6975E-6	1.3162E-7	1.3317E-6	9.2651E-7
	10	Conventional	1.296E-6	8.4752E-9	5.9492E-7	5.9646E-7
		Double-rotation	3.0917E-7	6.8938E-8	1.9876E-7	1.0249E-7
		Triple-rotation	2.5691E-7	2.5071E-9	9.0108E-8	1.0678E-7
	16	Conventional	3.3458E-10	4.3225E-12	1.2284E-10	1.4963E-10
		Double-rotation	6.9108E-11	2.7643E-12	4.7638E-11	1.6298E-11
		Triple-rotation	4.5727E-11	1.3323E-14	1.9549E-11	2.3043E-11
	32	Conventional	5.5511E-15	2.2204E-15	3.3307E-16	1.3597E-16
		Double-rotation	4.1078E-15	3.2196E-16	3.7970E-15	3.6316E-16
		Triple-rotation	1.4433E-15	3.1307E-16	9.5479E-16	4.4823E-16
	64	Conventional	5.5511E-15	2.2204E-15	3.3307E-16	1.3597E-16
		Double-rotation	4.1078E-15	3.2196E-16	3.7970E-15	3.6316E-16
		Triple-rotation	1.4433E-15	3.1307E-16	9.5479E-16	4.4823E-16
z_{out}	8	Conventional	6.6160E-3	3.4315E-4	3.7572E-3	2.5093E-3
		Double-rotation	3.1040E-3	2.9878E-4	1.8719E-3	1.1285E-3
		Triple-rotation	2.3227E-3	8.0323E-4	1.5465E-3	5.8293E-4
	10	Conventional	1.6100E-3	1.1742E-4	9.3258E-4	6.3258E-4
		Double-rotation	6.7778E-4	1.0275E-4	4.2626E-4	2.6665E-4
		Triple-rotation	7.1681E-4	7.0811E-5	3.4929E-4	2.6975E-4
	16	Conventional	2.5868E-5	2.9403E-6	1.2778E-5	1.0149E-5
		Double-rotation	9.9709E-6	4.0604E-6	7.2306E-6	2.2905E-6
		Triple-rotation	9.5632E-6	1.6408E-7	4.7247E-6	4.5793E-6
	32	Conventional	3.2805E-10	1.0026E-10	2.3369E-10	9.6639E-11
		Double-rotation	1.7855E-10	7.2290E-11	9.8846E-11	4.5243E-11
		Triple-rotation	1.5460E-10	2.4420E-11	8.5503E-11	5.5706E-11
	64	Conventional	1.1102E-16	2.7756E-17	6.6613E-17	3.1646E-17
		Double-rotation	1.1102E-16	2.7756E-17	6.6613E-17	3.1646E-17
		Triple-rotation	1.1102E-16	2.7756E-17	4.1633E-17	4.3885E-17

computational result and a Matlab simulation result is equal or less than ΔE, then the number of iterations will be kept in record.

Figure 2 illustrates the relationship of the given absolute error with the number of iterations of the conventional, double-rotation, and triple-rotation CORDIC methods based on Matlab/Simulink. As the figure, at the same number of iterations, the triple-rotation and double-rotation CORDIC methods provide higher accuracy than the conventional one. In turn, at the same absolute error, the triple-rotation and double-rotation CORDIC methods require the smaller number of iterations.

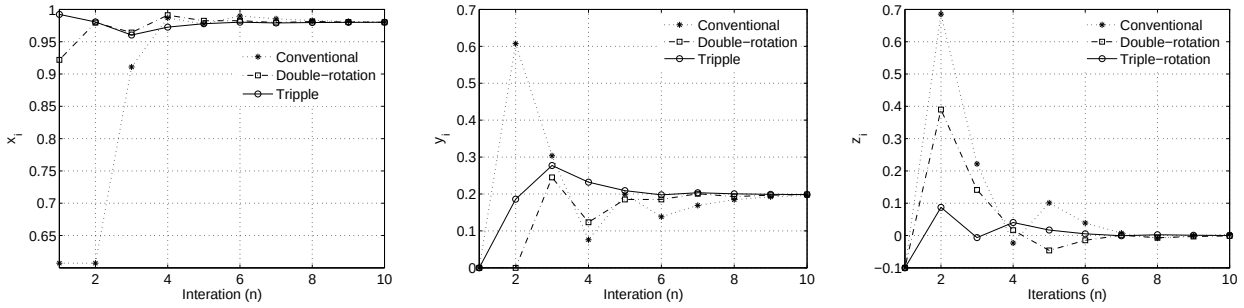
Second, observation of the converging parameters x, y, z of the micro-rotation: a Sine-Cosine function is performed by setting the initial parameters x, y, and z in rotating mode, where the parameter z is driven to be zero $z \longrightarrow 0$ by the CORDIC algorithm. Based on our experiment with several values of input z, the triple-rotation CORDIC method converges faster than the double-rotation and conventional ones for all three parameters. The converging of the three parameters is captured and shown in Figure 3 as an example, where z is initialized with $\phi = -0.1$ radian, y is set to 0, and x is defaulted by the constant scaling factor of each CORDIC method. The proposed CORDIC method provides better performance with the fewer number of iterations compared to the conventional and the double-ration method.



(a) 5 to 25 (b) 5 to 10

(c) 10 to 13 (d) 13 to 25

Figure 2: The expected number of iterations with the absolute error varied from 1.0E-8 to 1.0E-3.



(a) The convergence of parameter x (b) The convergence of parameter y (c) The convergence of parameter z

Figure 3: The convergence of CORDIC parameters where x is initialized with the constant scaling factors of each CORDIC method, $y = 0$, and $z = \phi = -0.1$ radian.

5. Unified CORDIC Algorithm

In this section, the unified micro-rotation of the double-rotation and triple-rotation CORDIC methods is introduced. It is described in the circular, hyperbolic, and linear coordinate systems in a unified manner by defining the parameter m so that

- **m=1**: for the circular/trigonometric coordinate system
- **m=0**: for the linear coordinate system
- **m=-1**: for the hyperbolic coordinate system

In that case, the unified micro-rotation of the double-rotation and triple-rotation CORDIC methods are:

The unified micro-rotation of the double-rotation CORDIC method

$$\begin{aligned} x_{i+1} &= x_i - m \cdot (\delta_i \cdot 2^{-i} \cdot y_i + 2^{-2i-2} \cdot x_i) \\ y_{i+1} &= y_i + \delta_i \cdot 2^{-i} \cdot x_i - m \cdot 2^{-2i-2} \cdot y_i \\ z_{i+1} &= \begin{cases} z_i - \delta_i \cdot 2 \cdot tan^{-1}(2^{-i-1}) & if\, m = 1\,, -1 \\ z_i - \delta_i \cdot 2^{-i} & if\, m = 0 \end{cases} \end{aligned} \tag{20}$$

The unified micro-rotation of the triple-rotation CORDIC method

$$\begin{aligned} x_{i+1} &= x_i \cdot (1 - m \cdot (2^{-2i-3} + 2^{-2i-4})) - \delta_i (m \cdot (2^{-i-1} + 2^{-i-2}) - 2^{-3i-6}) \cdot y_i \\ y_{i+1} &= \delta_i \cdot x_i \cdot (2^{-i-1} + 2^{-i-2} - m \cdot 2^{-3i-6}) + (1 - m \cdot (2^{-2i-3} + 2^{-2i-4})) \cdot y_i \\ z_{i+1} &= \begin{cases} z_i - \delta_i \cdot 3 \cdot tan^{-1}(2^{-i-2}) & if\, m = 1\,, -1 \\ z_i - \delta_i \cdot 3 \cdot 2^{-i-2} & if\, m = 0 \end{cases} \end{aligned} \tag{21}$$

With the non-redundant method the scaling factor for the double-rotation and triple-rotation CORDICs are $K_{dr} = \prod_{i=1}^{n} \frac{1}{(1+2^{-2i-2})}$ and $K_{tr} = \prod_{i=2}^{n+1} \frac{1}{(1+2^{-2i-4})^{\frac{3}{2}}}$ which are approximately 0.9219 and 0.992. The elementary functions being performed by the methods are listed in Table 5.

6. Algorithm and Investigation of a high precision CORDIC Core

6.1. Algorithm

The main characteristics of the double-rotation and triple-rotation CORDIC methods are: (1) reduction of computational latency, and (2) high computational accuracy with the small number of iterations. Thus, in this section we utilize such characteristics to design a high precision CORDIC core algorithm. The algorithm runs the double-rotation method for normal-accuracy mode, and the triple-rotation one for high-accuracy mode. It consists of three main parts, i.e. pre-processing, CORDIC processing, and post-processing as shown in Algorithm 1.

Table 5 is simultaneously considered with the Algorithm 1 due to its parameter relationship. The functional parameter ($func$) is in accordance with the function number shown in the table. The function number 1, 2 and 3 are used to implement circular/trigonometric functions such as sine, cosine, magnitude and phase functions, the function number 4, 5 and 6 are used to implement hyperbolic functions such as sine hyperbolic and cosine hyperbolic functions, and the function number 7 and 8 are used to implement linear functions, i.e. multiplication and division function, respectively.

Algorithm 1 $[x_{out}, y_{out}, z_{out}]$ = High-ACC-CORDIC(x_{in}, y_{in}, z_{in}, $Iter$, K, K^{-1}, $rmode$, m, $func$, hs)

{******** Pre-processing ********}
if ($hs = 1$) **then**
  $start = 2$, $end = Iter + 1$
else
  $start = 1$, $end = Iter$
end if
if ($func = 1$) or ($func = 4$) **then**
  x_{start}=K, y_{start}=0, z_{start}=z_i
else
  x_{start}=x_i, y_{start}=y_i, z_{start}=z_i
end if
{******** CORDIC processing ********}
for $i = start$ to end **do**
  if ($rmode = 0$) **then**
    if ($z_i \geq 0$) **then**
      $\delta_i = 1$
    else
      $\delta_i = -1$
    end if
  else
    if ($y_i \geq 0$) **then**
      $\delta_i = -1$
    else
      $\delta_i = 1$
    end if
  end if
  if ($hs = 1$) **then**
    $x_{i+1} = x_i \cdot (1 - m \cdot (2^{-2i-3} + 2^{-2i-4})) - \delta_i (m \cdot (2^{-i-1} + 2^{-i-2}) - 2^{-3i-6}) \cdot y_i$
    $y_{i+1} = \delta_i \cdot x_i \cdot (2^{-i-1} + 2^{-i-2} - m \cdot 2^{-3i-6}) + (1 - m \cdot (2^{-2i-3} + 2^{-2i-4})) \cdot y_i$
    if ($m = 0$) **then**
      $z_{i+1} = z_i - \delta_i \cdot 3 \cdot 2^{-i-2}$
    else
      $z_{i+1} = z_i - \delta_i \cdot 3 \cdot tan^{-1}(2^{-i-2})$
    end if
  else
    $x_{i+1} = x_i - m \cdot (\delta_i \cdot 2^{-i} \cdot y_i + 2^{-2i-2} \cdot x_i)$
    $y_{i+1} = y_i + \delta_i \cdot 2^{-i} \cdot x_i - m \cdot 2^{-2i-2} \cdot y_i$
    if ($m = 0$) **then**
      $z_{i+1} = z_i - \delta_i \cdot 2^{-i}$
    else
      $z_{i+1} = z_i - \delta_i \cdot 2 \cdot tan^{-1}(2^{-i-1})$
    end if
  end if
  $x_i = x_{i+1}$
  $y_i = x_{i+1}$
  $z_i = z_{i+1}$
end for
{******** Post-processing ********}
if ($rmode = 0$) **then**
  if ($func = 2$) or ($func = 5$) **then**
    $x_{out} = x_i \cdot K^{-1}$
    $y_{out} = y_i \cdot K^{-1}$
    $z_{out} = z_i$
  else
    $x_{out} = x_i$
    $y_{out} = y_i$
    $z_{out} = z_i$
  end if
else
  $x_{out} = x_i \cdot K^{-1}$
  $y_{out} = y_i$
  $z_{out} = z_i$
end if
return x_{out}, y_{out}, z_{out}

Table 5: The relationship of the elementary functions performed by the high precision CORDIC and all input arguments.

No.	Functions	*rmode*	Initial values			Configuration values		
			x_{in}	y_{in}	z_{in}	K^{-1}		m
						$hs=0$	$hs=1$	
Circular/Trigonometric Coordinate System								
1	$x_{out}=cos(z_{in})$ $y_{out}=sin(z_{in})$	0	K	0	given value	0	0	1
2	$x_{out}=x_{in}\cdot cos(z_{in})-y_{in}\cdot sin(z_{in})$ $y_{out}=x_{in}\cdot sin(z_{in})+y_{in}\cdot cos(z_{in})$		given value	given value	given value	K_{dr}^{-1}	K_{tr}^{-1}	1
3	$x_{out}=\sqrt{x_{in}^2+y_{in}^2}$ $z_{out}=z_{in}+tan^{-1}\left(\frac{y_{in}}{x_{in}}\right)$	1	given value	given value	given value	K_{dr}^{-1}	K_{tr}^{-1}	1
Hyperbolic Coordinate System								
4	$x_{out}=cosh(z_{in})$ $y_{out}=sinh(z_{in})$	0	K	0	given value	0	0	-1
5	$x_{out}=x_{in}\cdot cosh(z_{in})+y_{in}\cdot sinh(z_{in})$ $y_{out}=x_{in}\cdot sinh(z_{in})+y_{in}\cdot cosh(z_{in})$		given value	given value	given value	K_{dr}^{-1}	K_{tr}^{-1}	-1
6	$x_{out}=\sqrt{x_{in}^2-y_{in}^2}$ $z_{out}=z_{in}+tanh^{-1}\left(\frac{y_{in}}{x_{in}}\right)$	1	given value	given value	given value	K_{dr}^{-1}	K_{tr}^{-1}	-1
Linear Coordinate System								
7	$x_{out}=x_{in}$ $y_{out}=y_{in}+x_{in}\cdot z_{in}$	0	given value	given value	given value	1	1	0
8	$x_{out}=x_{in}$ $z_{out}=z_{in}+\frac{y_{in}}{x_{in}}$	1	given value	given value	given value	1	1	0

In the pre-processing step, the computational mode (hs) is determined in either normal-accuracy mode or high-accuracy mode, where the start ($start$) and end (end) indexes will be set up. Afterwards, the input parameters x_{start}, y_{start}, and z_{start} will be initialized corresponding to ($start$) and $func$. In the CORDIC processing step, the micro-rotation of either double-rotation or triple-rotation methods will be executed iteratively, where the execution in either rotation mode or vectoring mode and the coordinate systems depend on the rotating parameter $rmode$ and the coordinate parameter m. Finally, the post-processing step will compensate the computed results with inversion of the constant scaling factor corresponding to $func$ in Table 5.

6.2. Computational Time Investigation

The block diagram of the high precision CORDIC core according to Algorithm 5 is shown in Figure 4a. Due to the small convergence range of the double-rotation and triple-rotation methods, the convergence extension module which could be realized by convergence extension methods is included. Generally, there are two types of the convergence extension methods introduced to solve the convergence range problem, i.e. mathematic identity method [19], [22] and expansion of the set of iterative method [23]. The mathematic identity method applies the mathematic properties such as trigonometric identities in order to compress inputs x_i, y_i, and z_i and to decompress outputs x_{out}, y_{out}, and z_{out} generated by the high precision CORDIC core. The expansion of the set of iterative method expands the linear CORDIC convergence range by choosing the set of iterative indexes to $i=-M,-M+1,\cdots,N$, where M and N are two integers applied to determine the convergence. Then, the convergence ranges are $|z_i|\leq 2^{M+1}$ and $\left|\frac{y_i}{x_i}\right|\leq 2^{M+1}$ for driving z or

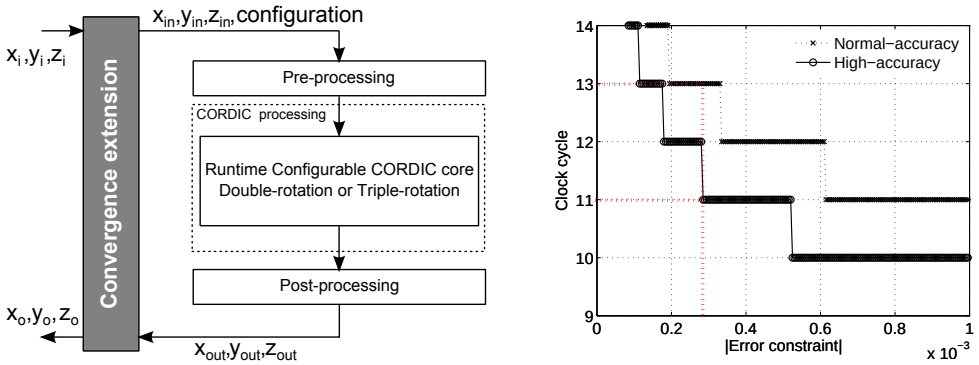


Figure 4: The block diagram of the high precision CORDIC core with the convergence extension module and its computational latency.

y toward zero. For the sake of simplicity the detail of the convergence extension methods will be ignored in this paper and its computational latency will be specified as a constant value.

From the block diagram in Figure 4a, the computational time investigation of the high precision CORDIC core with the convergence extension module can be examined as following: 1) suppose that T_{ext}, T_{pre}, T_{dr}, T_{tr}, and T_{post} are internal delay of the convergence extension, the pre-processing, the double-rotation, the triple-rotation and the post-processing, respectively. 2) T_{dr} and T_{tr} depending on the number of CORDIC iterations ($N_{iter-dr}$, $N_{iter-tr}$) corresponding to the expected accuracy as shown in Figure 2 are expressed as:

$$\begin{aligned} T_{dr} &= T_{micro-dr} \cdot N_{iter-dr} \\ T_{tr} &= T_{micro-tr} \cdot N_{iter-tr}, \end{aligned} \tag{22}$$

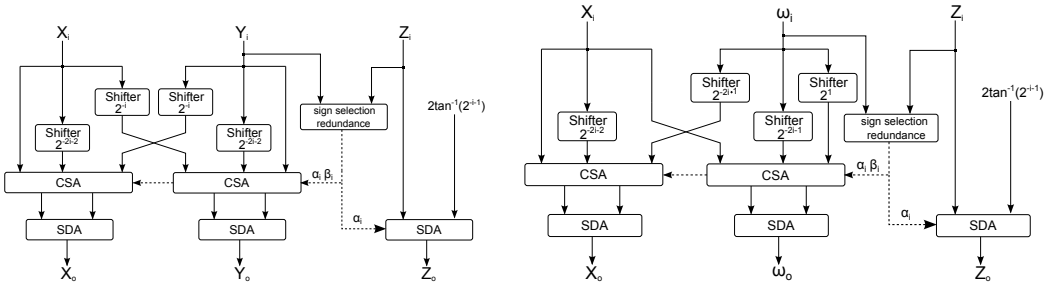
where $T_{micro-dr}$ and $T_{micro-tr}$ are the computational delay of the micro-rotation of the double-rotation and triple-rotation CORDICs. The number of iterations of the double-rotation and triple-rotation CORDIC methods is denoted as $N_{iter-dr}$ and $N_{iter-tr}$. Therefore, the computational delay investigation of the high precision CORDIC core with the convergence extension module can be described as:

$$\begin{aligned} T &= 2 \cdot T_{ext} + T_{pre} + T_{CORDIC} + T_{post}, \\ T_{CORDIC} &= \begin{cases} T_{dr} & hs = 0 \ (Normal\ accuracy\ mode), \\ T_{tr} & hs = 1 \ (High\ accuracy\ mode) \end{cases} \end{aligned} \tag{23}$$

Suppose that T_{ext}, T_{pre}, T_{post}, $T_{micro-dr}$, and $T_{micro-tr}$ equal to 1 clock cycle, then the delay of the high precision CORDIC core based on the double-rotation and triple-rotation CORDIC methods when the expected absolute error 3.0E-4 are at 11 clock cycle and 13 clock cycle. The graph in Figure 4b shows the relationship of the expected absolute computational error and the delay of the high precision CORDIC Core in normal-accuracy mode and high-accuracy mode with the absolute errors range from 0 to 1.0E-3.

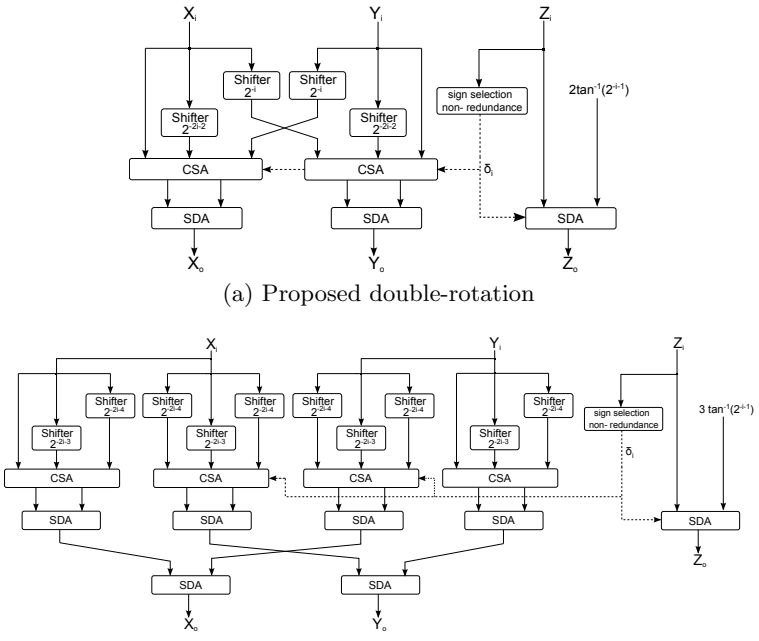
6.3. Performance Comparison

This section compares the proposed CORDIC methods with the existing ones. The micro-rotation in pipelined (unfolded) digit-parallel architecture has been brought up for



(a) Double-rotation [19] (b) 2D-Householder [24]

Figure 5: The existing constant scaling factor CORDIC based on redundant method.



(a) Proposed double-rotation

(b) Proposed triple-rotation

Figure 6: The proposed constant scaling factor CORDIC based on non-redundant method.

consideration in speed and area performance, where pre-processing and post-processing are ignored. Basic components normally used to implement the CORDIC consist of 3-to-2 Carry-Save-Adder (CSA), Sign-Digit-Adder (SDA), redundant sign selection (SIGN-SEL), non-redundant sign selection (SIGN-SEL-NON), and right shifter (SHR). To obtain comparison results close to reality as much as possible, the basic components are implemented and analyzed in various data width at 16-bit, 32-bit, and 64-bit on 90-nm Faraday silicon technology. The synthesis results are individually normalized based on delay and consumed area of the targeted technology; afterward they are normalized again in the various data width as presented in Table 6.

The two existing constant scaling factor CORDIC methods, i.e. the redundant double-rotation [19] CORDIC and the 2D-Householder [24] CORDIC, have been put forward for comparison with the proposed CORDIC methods, whose architectures on rotation mode in the circular coordinate system are illustrated in Figure 5 and 6. Table 7 compares the speed and area performance of these CORDIC methods. Based on pipeline architecture, the computational operators corresponding to each CORDIC algorithm are performed as the delay models. Also the number of utilized basic computational operators, conforming

Table 6: Basic components synthesis results on 90-nm Faraday silicon technology.

Basic component	16-bit		32-bit		64-bit	
	D(ns)	A(μm^2)	D(ns)	A(μm^2)	D(ns)	A(μm^2)
3-to-2 CSA	0.13/0.0807	252.23/0.5708	0.13/0.0458	504.11/0.4836	0.13/0.0245	1,005.87/0.3978
SDA	1.61/1	442.96/1	2.84/1	806.74/0.7714	5.31/1	1,534.29/0.6068
SHR	0.70/0.4348	377.89/0.8533	0.62/0.2183	1,045.85/1	0.92/0.1733	2,528.40/1
SIGN-SEL	0.15/0.0932	13.33/0.0301	0.15/0.0528	13.33/0.0127	0.15/0.0282	13.33/0.0053
SIGN-SEL-NON	0.01/0.0063	2.35/0.0053	0.01/0.0035	2.35/0.0220	0.01/0.0019	2.35/0.0009

Table 7: The time and area performance of the CORDIC methods in pipeline (unfolded) digit-parallel architecture.

Double-rotation [19]	Delay	$y_i + \beta_i 2^{-2i-2} x_i - \alpha_i 2^i y_i \Longrightarrow D_{SHR} + D_{SIGN-SEL} + D_{CSA} + D_{SDA}$
	Area	$4 \cdot A_{SHR} + 2 \cdot A_{CSA} + 3 \cdot A_{SDA} + A_{SIGN-SEL}$
2D-Householder [24]	Delay	$2 \cdot \omega_i - \beta_i 2^{-2i-1} \omega_i - \alpha_i 2^i x_i \Longrightarrow D_{SHR} + D_{SIGN-SEL} + D_{CSA} + D_{SDA}$
	Area	$4 \cdot A_{SHR} + 2 \cdot A_{CSA} + 3 \cdot A_{SDA} + A_{SIGN-SEL}$
Double-rotation	Delay	$x_i + \delta_i 2^{-i} y_i - \delta_i 2^{-2i-2} x_i \Longrightarrow D_{SHR} + D_{SIGN-SEL-NON} + D_{CSA} + D_{SDA}$
	Area	$4 \cdot A_{SHR} + 2 \cdot A_{CSA} + 3 \cdot A_{SDA} + A_{SIGN-SEL-NON}$
Triple-rotation	Delay	$(x_i - 2^{-2i-3} x_i - 2^{-2i-4} x_i) - (\delta_i 2^{-i-1} y_i + \delta_i 2^{-i-2} y_i - \delta_i 2^{-3i-6} y_i)$ $\Longrightarrow D_{SHR} + D_{SIGN-SEL-NON} + D_{CSA} + 2 \cdot D_{SDA}$
	Area	$10 \cdot A_{SHR} + 4 \cdot A_{CSA} + 7 \cdot A_{SDA} + A_{SIGN-SEL-NON}$

to Figure 5 and 6, is modeled as the area models. In [19], the redundant double-rotation method with a constant scaling factor is applied to only the CORDIC rotation mode. The proposed double-rotation method extends to vectoring mode. The redundant rotation direction ($\alpha_i, \beta_i \in \{-1, 0, 1\}$) are employed in [19], but we apply the non-redundant rotation direction in our double-rotation CORDIC. From the delay and area models in Table 7, the delay and consumed area on 16-bit data width of the double-rotation CORDIC method of [19] can be evaluated as following: Delay : $0.4348 + 0.0932 + 0.0807 + 1 = 1.6087$, Area : $4 \times 0.8533 + 2 \times 0.5708 + 3 + 0.0301 = 7.5849$.

In [24], the redundant 2D-Householder CORDIC method is applied on both rotation mode and vectoring mode. By the way, its scaling factor is performed by the on-line computation which increases the complexity for VLSI implementation. However, for the sake of simplification, the on-line computation is neglect for this comparison. The delay and area on 16-bit data width of the 2D-Householder CORDIC method can be estimated as following: Delay : $0.4348 + 0.0932 + 0.0807 + 1 = 1.6087$, Area : $4 \times 0.8533 + 2 \times 0.5708 + 3 + 0.0301 = 7.5849$. By the same method the delay and area on 16-bit of the proposed double-rotation CORDIC method can be evaluated as follow: Delay : $0.4348 + 0.0063 + 0.0807 + 1 = 1.5218$, Area : $4 \times 0.8533 + 2 \times 0.5708 + 3 + 0.0053 = 7.5601$, and Delay : $0.4348 + 0.0063 + 0.0807 + 2 = 2.5218$, Area : $10 \times 0.8533 + 4 \times 0.5708 + 7 + 0.0053 = 17.822$ for the proposed triple-rotation CORDIC method. The time and area efficiency of these CORDIC methods in different data width can be illustrated in Table 8.

From the comparison, the proposed redundant double-rotation CORDIC method provides better time efficiency than the non-redundant double-rotation and 2D-Householder CORDIC methods. However, the proposed double-rotation one is extended to vectoring mode and also unemploy to use the on-line constant scaling factor computation. Area efficiency of the three CORDIC methods have similar values because they use the same number of basic components. The proposed triple-rotation CORDIC methods is also evaluated to demonstrate its time and area efficiencies. Although the time and area efficiency of the triple-rotation CORDIC method are lower than the double-rotation CORDIC meth-

Table 8: Normalized speed and area performance comparison of the proposed CORDIC methods and the existing CORDIC methods in different data width.

CORDIC methods	16-bit		32-bit		64-bit	
	Delay	Area	Delay	Area	Delay	Area
Double-rotation [19]	1.6087	7.5856	1.317	7.2916	1.226	6.6214
2D-Householder [24]	1.6087	7.5856	1.317	7.2916	1.226	6.6214
Proposed double-rotation	1.5217	7.5608	1.2677	7.2811	1.1996	6.6171
Proposed triple-rotation	2.5218	17.822	2.2678	17.332	2.1996	15.84

ods, but the method provides higher computational accuracy. Its time efficiency can be improved by increasing a pipeline stage or by enhancing the performance of Adder.

7. Conclusion

Design and analysis of the extension-rotation CORDIC methods, the double-rotation and the triple-rotation, to improve the performance and accuracy of the CORDIC computation have been presented and discussed. The paper can be summarized as follow:

1. The methods use non-redundant values to stabilize the constant scaling factor and to avoid the on-line scaling factor problem. The computational accuracy of the two CORDIC methods is measured by statistical measurements, i.e. MAPE, $Max.|error|$, $Min.|error|$, $Ave.|error|$, $Std.Dev.|error|$, and compared to the conventional CORDIC method and the Matlab Simulation results. The analysis results show the double-rotation and triple-rotation CORDIC methods provide higher accuracy than the conventional one with the same number of iterations. On the other hand, with the same computational accuracy, the double-rotation and triple-rotation can be achieved with smaller number of iteration.
2. The unified CORDIC algorithms of the double-rotation and triple-rotation CORDIC methods applied to perform elementary function in rotation mode and vectoring mode on the circular, hyperbolic, linear coordinate systems are come out. Afterward, they are utilized for algorithm of the high precision CORDIC core.
3. The high accuracy CORDIC core is introduced and investigated in order to show the performance and time efficiency in normal-accuracy and high-accuracy modes in various expected absolute error. Based on the pipeline (unfolded) digit-parallel architecture, the speed and area performance of the proposed CORDIC methods are compared with the existing CORDIC methods, where the proposed double-rotation CORDIC method provide better time efficiency with similar area efficiency to the existing constant scaling factor CORDIC methods.

References

[1] J. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electron. Compute.*, vol. EC-8, pp. 330–334, 1959.

[2] ——, "The birth of CORDIC," *The Journal of VLSI Signal Processing*, vol. 25(2), pp. 101–105, 2000.

[3] J. Walther, "A unified algorithm for elementary functions," *Spring Joint Computer Conf.*, pp. 379–385, 1971.

[4] P. Meher, J. Valls, J. Tso-Bing, K. Sridharan, and K. Maharatna, "50 Years of CORDIC: Algorithms, Architectures, and Applications," *IEEE Transactions, Circuits and Systems*, vol. 56, pp. 1893–1907, 2009.

[5] D. Cochrab, "Algorithms and accuracy in the HP-35," *HewlettPackard Journal*, pp. 1–11, June 1972.

[6] S. F.Hsiao and J. M. Delosme, "Parallel singular value decomposition of complex matrices using multidimensional CORDIC algorithms," *IEEE Trans. Signal Processing*, vol. 44, no. 3, pp. 685–697, 1996.

[7] ——, "Householder CORDIC algorithm," *IEEE Trans. Computers*, vol. 44, no. 8, pp. 990–1001, 1995.

[8] B. Lakshmi and A. S. Dhar, "CORDIC Architectures: A Survey," *Hindawi Publishing Corporation VLSI Design*, vol. 2010, pp. 1–19, 2010.

[9] E. Antelo, J. Bruguera, and E. Zapata, "Unified mixed radix 2-4 redundant CORDIC processor," *IEEE Transactions on Computers*, vol. 45, pp. 1068–1073, 1996.

[10] E. Antelo, J. Villalaba, D. Bruguera, and E. Zapata, "High performance rotaion architecture based on the radix CORDIC algorithm," *IEEE Trans. Comput.*, vol. 46, no. 46, pp. 855–870, August 1997.

[11] E. Antelo, T. Lang, and J. Bruguera, "Very-high radix cicular CORDIC: Vecotring and unified rotation/vectoring," *IEEE Trans. Comput.*, vol. 49, no. 7, pp. 727–739, July 2000.

[12] C.-C. Li and S.-G. Chen, "A radix-4 redundant CORDIC algorithm with fast on-line variable scale factor compensation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997, pp. 639–642.

[13] J. Villalba, E. Zapata, E. Antelo, and J. Bruguera, "Radix-4 Vectoring CORDIC Algorithm and Architectures," *The Journal of VLSI Signal Processing*, vol. 19, no. 2, pp. 127–147, 1998.

[14] T. Aoki, I. Kitaori, and T. Higuchi, "Radix-2-4-8 CORDIC for Fast Vector Rotation," *IEICE Transaction Fundamentals*, vol. E83-A, no. 6, pp. 1106–1114, 2000.

[15] J. Tso-Bing, H. Shen-Fu, and T. Ming-Yu, "Para-CORDIC: parallel CORDIC rotation algorithm," *IEEE Transactions on Circuits and Systems I, Regular Papers*, vol. 51, pp. 1515–1524, 2004.

[16] H. Shen-Fu, H. Yu-Hen, and J. Tso-Bing, "A memory-efficient and high-speed sine/cosine generator based on parallel CORDIC rotations," *IEEE Signal Processing Letters*, vol. 11, pp. 152–155, 2004.

[17] W. Han, Z. Yousi, and L. Xiaokang, "A Parallel Double-Step CORDIC Algorithm for Digital Down Converter," in *Communication Networks and Services Research Conference, 2009.*, 2009.

[18] M. D. Ercegovac and T. Lang, "Redundant and on-line CORDIC: Application to matrix triangularization and SVD," *IEEE Trans. Comput.*, vol. 39, no. 6, pp. 725–740, June 1990.

[19] N. Takagi, T. Asada, and S. Yajima, "Redundant CORDIC methods with a constant scale factor for sine and cosine computation," *IEEE Transactions on Computers*, vol. 40, pp. 989–995, September 1991.

[20] M. D. Ercegovac and T. Lang, *Digital Arithmetic*. Morgan Kaufmann, 2003.

[21] J. Muller, *Elementary Functions: Algorithms and Implementation*. Cambridge, MA: Birkhauser, 1997.

[22] P. Surapong and M. Glesner, ""Pipelined Floating-Point Architecture for a Phase and Magnitude Detector based on CORDIC"," in *International Conference on Field Programmable Logic and Application*, 2011, pp. 382–384.

[23] X. Hu, R. Harber, and S. Bass, "Expanding the range of convergence of the CORDIC algorithm," *IEEE Transactions on Computers*, vol. 40, pp. 13–21, 1991.

[24] S.-F. Hsiao and J.-Y. Chen, "Design, Implementation and Analysis of a New Redundant CORDIC Processor with Constant Scaling Factor and Regular Structure," *J. VLSI Signal Process. Syst.*, vol. 20, pp. 267–278, December 1998.

Authors

Pongyupinpanich Surapong was born in Prachinburi, Thailand. He received his Bachelor and Master of Engineering degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand in 1998 and 2002. Currently, he is working toward the PhD degree in Microelectronic Systems Research Group, Technische Universität Darmstadt, Darmstadt, Germany. His research interests include computer-aided VLSI design, design optimization algorithm, circuit simulation, digital signal processing, system-on-chip, all in the context of field-programmable gate-array devices and VLSI technology.

Faizal Arya Samman was born in Makassar, Indonesia. In 1999, he received his Bachelor of Engineering degree from Universitas Gadjah Mada, in Yogyakarta, Indonesia. In 2002, he received his Master of Engineering degree from Institut Teknologi Bandung, in Indonesia with Scholarship Award from Indonesian Ministry of National Education. In 2002, he was appointed to be a research and teaching staff at Universitas Hasanuddin, in Makassar, Indonesia. He received his PhD degree in 2010 at Technische Universität Darmstadt, in Germany with scholarship award from Deutscher Akademischer Austausch-Dienst (DAAD, German Academic Exchange Service). He is now working as a postdoctoral fellow in LOEWE-Zentrum AdRIA (Adaptronik-Research, Innovation, Application) within the research cooperation framework between Technische Universität Darmstadt and Fraunhofer Institut LBF in Darmstadt. His research interests include network-on-chip (NoC) microarchitecture, NoC-based multiprocessor system-on-chip, design and implementation of analog and digital electronic circuits for control system applications on FPGA/ASIC as well as energy harvesting systems and wireless sensor networks.

Manfred Glesner received the diploma degree and the Ph.D. degree from Universität des Saarlandes, Saarbrücken, Germany, in 1969 and 1975, respectively. His doctoral research was based on the application of nonlinear optimization techniques in computer-aided design of electronic circuits. He received three Doctor Honoris Causa degrees from Tallinn Technical University, Tallinn, Estonia, in 1996, Poly-technical University of Bucharest, Bucharest, Romania, in 1997, and Mongolian Technical University, Ulan Bator, Mongolia, in 2006. Between 1969 and 1971, he has researched work in radar signal development for the Fraunhofer Institute in Werthoven/Bonn, Germany. From 1975 to 1981, he was a Lecturer in the areas of electronics and CAD with Saarland University. In 1981, he was appointed as an Associate Professor in electrical engineering with the Darmstadt University of Technology, Darmstadt, Germany, where, in 1989, he was appointed as a Full Professor for microelectronic system design. His current research interests include advanced design and CAD for micro- and nanoelectronic circuits, reconfigurable computing systems and architectures, organic circuit design, RFID design, mixed-signal circuit design, and process variations robust circuit design. With the EU-based TEMPUS initiative, he built up several microelectronic design centers in Eastern Europe. Between 1990 and 2006, he acted as a speaker of two DFG-funded graduate schools. Dr. Glesner is a member of several technical societies and he is active in organizing international conferences. Since 2003, he has been the vice-president of the German Information Technology Society (ITS) in VDE and also a member of the DFG decision board for electronic semiconductors, components, and integrated systems. He was a recipient of the honor/decoration of "Palmes Academiques" in the order of Chevalier by the French Minister of National Education (Paris) for distinguished work in the field of education in 2007/2008.