International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Efficient Implementation of Reconfigurable MIMO Decoder Architecture

Teena Philip, S. Suresh Babu

Department of Electronics and Communication Engineering, Sree Buddha College of Engineering, India

Abstract: ***Multiple Input Multiple Output (MIMO) and Orthogonal Frequency Division Multiplexing (OFDM) are two dominant technologies in emerging wireless communication. MIMO transmission increases the capacity and reliability o a wireless system without increasing its bandwidth. OFDM divides a wideband channel into multiple narrowband subchannels via a computationally efficient fast fourier transform (FFT) operation, but it requires identical baseband processing for each of the subchannel. Combining OFDM with MIMO technique increase spectral efficiency, throughput and improves the link reliability. The MIMO decoder is one of the most complex blocks in MIMO transceiver. MIMO decoding is essentially an inversion of a complex matrix channel and it can be achieved by using a variety of MIMO decoding algorithms. The choice of algorithm and antenna configuration depends on the available resources, channel conditions, power budget and throughput requirement. The existing MIMO decoder design needs to be upgraded in order to allow new MIMO decoding algorithm and wireless communication standards. This necessitates the need for a programmable accelerator block to implement the MIMO decoder. This thesis proposes an area efficient implementation of a programmable MIMO decoder accelerator that targets MIMO decoding tasks of OFDM system.***

Keywords: Multiple Input Multiple Output (MIMO), Orthogonal Frequency Division Multiplexing (OFDM), Reconfigurable MIMO decoder, Rotation unit, Coordinate Rotation Digital Computers (CORDIC).

1. Introduction

Recently, MIMO technology had a great attention in wireless communication technologies, since it offers significant increase in data throughput and link range without additional bandwidth or transmit power. MIMO is a wireless technology that uses multiple transmitters and receivers to transfer more data at the same time. It takes the advantages of multipath propagation, where transmitted information undergo several path and reaching the receiving antenna multiple times via different angles and at slight different times.

MIMO technology is currently used in broadband systems that exhibit frequency selective fading and it results in intersymbol interference (ISI). ISI is avoided by combining MIMO with OFDM. OFDM modulation turns the frequency selective channel into set of parallel flat fading channels and is an attractive way of eliminates ISI in MIMO system. MIMO wireless technology in combination with OFDM is an attractive air interface solution for next generation Wireless Local Area Networks (WLANs), Wireless Metropolitan Area Networks (WMANs) and fourth generation mobile cellular wireless systems.

In MIMO communication environment, MIMO decoder is the most complex blocks in a MIMO transceiver. MIMO decoding is an inversion of complex channel matrix with a low latency requirement for each subcarrier and it can be achieved by using a variety of algorithms. The choice of algorithms and antenna configuration depends on the channel condition, available resources, power budget and throughput requirement. A MIMO decoder design process for a certain application is hard and time consuming. This motivates the need for a programmable accelerator block to implement the MIMO decoder.

Various hardware design and implementation for MIMO decoders are available. But these decoder designs use single MIMO decoding algorithm such as zero forcing (ZF), Maximum likelihood (ML), or one of the many sphere decoding (SD) variants. Also several reconfigurable MIMO decoders have been available. But these designs are neither flexible enough to incorporate new standards and algorithms.

Advanced MIMO decoding algorithms and wireless communication standards are emerging and upgrading of the existing system is required to meet the newly defined standards. Hence, there is a scope for design of a flexible and efficient MIMO decoder, which could be able to serve multiple standards simultaneously without compromising throughput, area and power requirement.

2. Multiple Input Multiple Output

In conventional wireless communication, a single antenna is used at source and destination and it lead to the problems with multipath effects. In radio communication, MIMO is a method for multiplying the capacity of a radio link using multiple transmit and receive antennas to exploit multipath propagation. MIMO technology takes advantages of multipath behaviour by using multiple smart transmitters and receivers with an added spatial dimension to increase the performance and range.

MIMO technology can be implemented in various ways such as spatial multiplexing and spatial diversity technique. Spatial multiplexing is used in MIMO concept for increasing the capacity. This need to send different set of data at same time through different MIMO antenna. If need to use advantage of MIMO diversity to overcome the fading, then need to send the same signal through the different MIMO antenna and at the receiver end, the different antenna will receive the same signal travelled through diverse paths.

2.1 MIMO OFDM

MIMO wireless technology meet the key challenges faced by future wireless communication systems by offering increased spectral efficiency through spatial multiplexing gain and improved link reliability due to antenna diversity gain. But the main disadvantages of MIMO technology is that exhibit frequency selective fading, which results in intersymbol interference (ISI). ISI in MIMO system can be avoided by combining MIMO with OFDM. OFDM modulation converts the frequency selective MIMO channel into a set of flat fading channel. MIMO OFDM system model is shown in Figure 1.

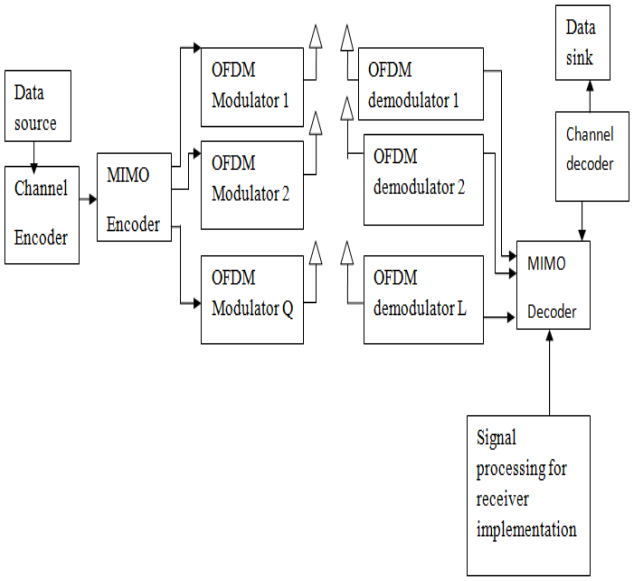

Figure 1: MIMO system model

2.2 MIMO Channel

Consider a MIMO system with N transmitter antennas and M receiver antennas. The mathematic model can be written as

$r=Hs+n$

where s is the transmitted symbol vector, $[s_1\ s_2,\ldots.s_N]^T$, H is a channel matrix with M X N dimensions, and the entries of the H matrix are all independent and identically distributed (i.i.d.) complex zero-mean Gaussian random variables, n is a complex white noise vector $[n_1\ n_2\ \ldots.n_M]^T$ with zero mean and variance N_0 per complex entry and $r=[r_1\ r_2\ \ldots.r_M]^T$is the received symbol vector.

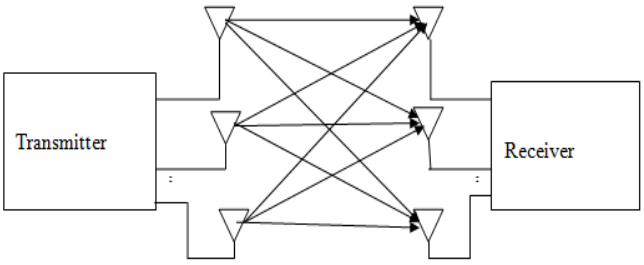

Figure 2: MIMO Channel

The MIMO channel is shown in Figure 2. The number of data streams that can be transmitted in parallel over the MIMO channel is given by min {Nt, Nr} and is limited by the rank of the matrix H. The transmission quality degrades significantly in case the singular values of matrix H are not sufficiently strong. The channel (for a specific delay) can thus be described by the following channel matrix H:

$$H=\begin{bmatrix} h11 & h12 \cdots & h1Nr \\ \vdots & \ddots & \vdots \\ hNt\,1 & hNt\,2 \cdots & hNt\,Nr \end{bmatrix}\begin{bmatrix} h11 & h12 \cdots & h1Nr \\ \vdots & \ddots & \vdots \\ hNt\,1 & hNt\,2 \cdots & hNt\,Nr \end{bmatrix}$$

3. MIMO Decoder Architecture

A MIMO decoder is the receiver component that separates the Nss transmitted data streams from the signals received on the Nrx receives antennas. Most, if not all, of the MIMO decoding operation is matrix and vector intensive. For an OFDM system, this processing is repeated for every subchannel. The result is a processing bottleneck at the receiver. MIMO decoding is essentially an inversion of a complex matrix channel. This can be achieved using a variety of algorithms with a range of complexity and performance. The choice of algorithm and antenna configuration depends on the expected power budget, throughput requirements channel conditions and available resources.

3.1 Programmable MIMO Decoder accelerator

Two prominent trends in wireless communication are the use of multiple input multiple output (MIMO) processing, and orthogonal frequency division multiplexing (OFDM) to improve data rate and reliability. All trends point to the convergence of multiple MIMO-OFDM standards on a single platform. This motivates an accelerator-like approach to efficiently deliver on the computation intensive elements of the system. The MIMO decoder is one such component. MIMO processing is computationally intensive due to the need to invert a channel matrix with very low latency. Moreover over time, systems are expected to incorporate a higher number of antennas and more advanced algorithms.

The accelerator needed to ensure that all major algorithms can be supported on the accelerator. Here approach to addressing this problem is to identify the set of primitive processing elements that form the basis of all major MIMO decoding algorithms. With such a set in hand, the realization of a specific decoder algorithm will translate into the proper sequencing of data among these primitive elements through a program. The major decoding algorithms fall into three categories: Maximum Likelihood solutions (ML) including Sphere Decoding (SD), Singular Value Decomposition (SVD) as an arithmetic aid to linear decoding or as a beam forming tool, and linear decoding algorithms such as MMSE and Zero Forcing (ZF). Matrix decomposition is critical to all these algorithms. The algorithms can be supported by four classes of arithmetic operations: complex multiplication, various unitary transformations, complex addition, and division.

The accelerator allows the programmer to define and implement MIMO decoders at will. The accelerator has a processor-like architecture with most of the controls

derived from a memory-stored program. The processing core is designed to support a range of complex operations necessary to enable the realization of major MIMO decoding algorithms. The accelerator core accepts very wide complex matrix operands and produces complex matrix results. The high access rate required to support this is made possible by a memory map that exploits the matrix/vector nature of the operands in MIMO decoding. The memory map is augmented by sorting circuits at the inputs and outputs of memory that allow the programmer to redefine input and output order without using extra processing cycles. The processing cycle uses properties of OFDM decoding to optimize its flow, and through the use of predecoded instructions and proper compiler positioning of critical control signals, the accelerator ensures that the processing pipeline is continually engaged. A programmable dynamic scaling circuit automatically handles intermediate word length issues for high dynamic range operations.

3.2 Building blocks of programmable MIMO decoder

MIMO decoding is based on extensive matrix processing. The MIMO accelerator is a complex number vector-based processor that works on complex vector operands of length Nrx where Nrx is the number of receive antennas used in the MIMO system. The basic building blocks of the MIMO accelerator are shown in Figure 3. The MIMO-accelerator in Figure 3 consists of a processing unit that supports highly flexible vector coarse operations, connected to a data memory designed to utilize properties of matrix processing in order to allow flexible and highly efficient access.

1) **Processing unit:** The processing core, which is the main data path of the MIMO accelerator, specifically targets the MIMO decoding tasks. It consists of four powerful processing units that are chosen based on the minimum set of primitive operations needed to fully implement most (if not all) MIMO decoding algorithms. The processing unit consists of four cores: An inner product core, a scalar division core, a coordinate rotation core, and a vector addition core. The four processing units are shown in Figure 3 First is the addition unit, which is an adder/subtractor that can process two pairs of Nrx complex vectors simultaneously. Examples of its uses are the formulation of the MMSE matrix and the calculation of the SD metrics.

2)

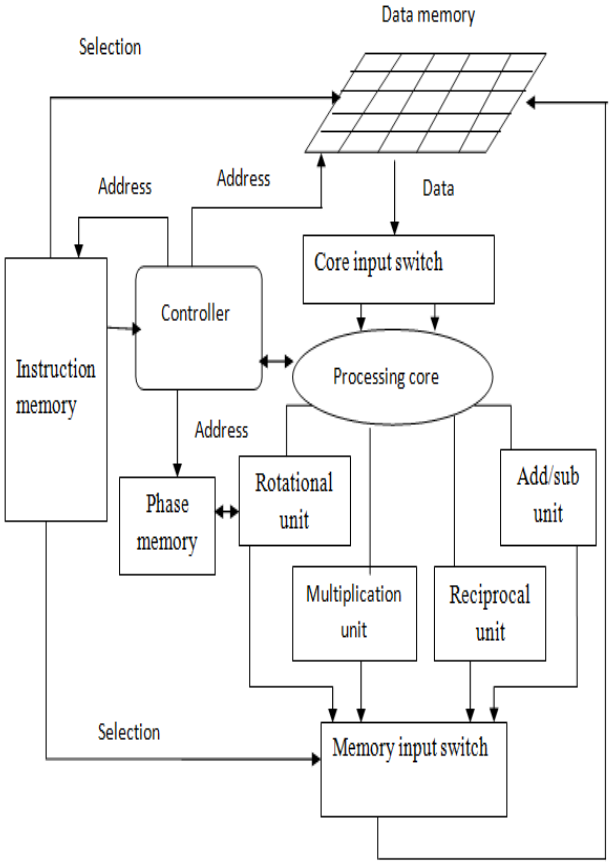

Figure 3: Basic block diagram of programmable MIMO decoder

Second is a multiplication unit that contains four dot product blocks. Each of which computes a single complex number that results from a dot product of two complex Nrx vectors. This allows the multiplication unit to perform a complete vector-matrix multiplication in a single processor cycle. This multiplication unit is necessary for many MIMO decoding algorithms. Third is a reciprocal unit that computes a reciprocal of Nrx real numbers. It is mainly used for scaling the signal power. The fourth processing unit is the rotation unit. It consists of a group of coordinate rotation (CORDIC) blocks.

3) **Instruction memory:**

The instruction memory is used to store predecoded instructions that run on the MIMO accelerator. An instruction is a wide control word that dictates: 1) which processing unit is to be used for a given operation; 2) the required configuration for this particular unit; 3) the data memory locations for the input vector operands for the processing unit; 4) the data memory locations that will be used to store the results (outputs) of the processing unit; and 5) flow controls that are used by the controller.

4) **Data memory:**

The data memory is the source for all operands as well as the target for the processing core results. Each memory location contains the data for a single OFDM subchannel. This data is logically divided into a number of complex matrix variables of size Nrx by Nrx. When an instruction is executed for a subchannel, the chunk of data associated

with the subchannel is retrieved and then delivered to the core-input switch.

5) **Core input switch:**

The core-input switch is a two level multiplexing circuit that selects and properly arranges the complex vectors needed by the processing core, whether they are row vectors, column vectors, matrix diagonals, or a combination thereof.

6) **Memory input switch:**

The memory-input switch performs the same task, but in the reverse direction. It takes the outputs of the processing units and properly packages them so as to write all data associated with the given OFDM subcarrier into the appropriate memory location. The instruction word provides the programmer with complete control of the two switching circuits, thus delivering a significant amount of flexibility.

4. CORDIC Rotation Unit

COordinate Rotation DIgital Computer (CORDIC) is an iterative algorithm for calculating trigonometric functions including sine, cosine, magnitude and phase. It is particularly suited to hardware implementations because it does not require any multiplies. CORDIC, also known as the digit-by-digit method and Volder's algorithm, is a simple and efficient algorithm to computing hyperbolic and trigonometric functions, using only basic arithmetic (addition, subtraction and shifts). It is commonly used when no hardware multiplier is available (e.g. in simple microcontrollers and FPGAs) as the only operations it requires are addition, subtraction, bitshift and table lookup.

The CORDIC algorithm is an iterative family of equations that is used to calculate vectors or angles, depending on the mode in which they are used. The CORDIC algorithm is classified as a linear convergence algorithm, requiring n-iterations for n-bits of accuracy. The CORDIC algorithm provides an iterative method of performing vector rotation by an arbitrary angle using only shift and add operation. All the trigonometric functions can be evaluated from functions using vector rotations.
In CORDIC algorithm, rotate a vector $(U, V)^T$ by angle θ

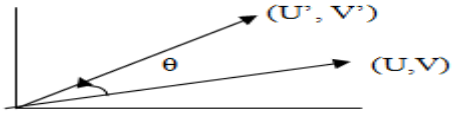

Figure 4: CORDIC Rotation unit

The algorithm is derived using the general rotation transform:

$U'=U\cos\theta - V\sin\theta$
$V'=V\cos\theta + U\sin\theta$

Where (U', V') are the coordinates of the resulting vector after rotation of a vector with coordinates (U,V) through an angle θ of in the rectangular plane. These equations can be:

$U'=\cos\theta\,[U - V\tan\theta]$

$V'=\cos\theta\,[V + U\tan\theta]$

If the rotation angles are restricted, $\tan(\phi)=\pm 2^{-i}$.multiplication by tangent term is reduced to simple shift operation. Hence angles of rotation can be found by doing continuously smaller elementary rotations. Therefore vector is iteratively rotate until angle is θ. Then the above equation for rotation can be expressed as:

$U_{i+1}=K_i\,[U_i - V_i.d_i.2^{-i}]$

$V_{i+1}=K_i\,[V_i + U_i.d_i.2^{-i}]$

$\theta_{i+1}=\theta_i - d_i.\tan^{-1}2\tan^{-1}2^{-i}$

Where $K_i=\cos(\tan^{-1}2\tan^{-1}2^{-i})=1/\sqrt{1+2^{-2i}}$ $1/\sqrt{1+2^{-2i}}$, $d_i=\pm 1$.

If the decision at each iteration i, is which direction to rotate rather than whether or not to rotate, then cos(θi) term become constant. θ is a variable to keep track of the total rotation.

CORDIC operates mainly in two modes for computation of different functions. These modes are known as rotation mode and vector mode. In rotation mode, the co-ordinate components of a vector and an angle of rotation is given and the co-ordinate component of original vector, after rotation through given angle are computed. In vector mode, the coordinate component of a given vector is given and the magnitude and angular argument of original vector are computed.

4.1 Vector CORDIC

Vectoring modes rotates the input vector through an angle necessary to align the input vector to the x-axis. The result of the vectoring operation is a rotation and scaled magnitude of the original vector, which is the x-component of the result. In this mode, each rotation works in order to minimize the y-component of the residual vector. The direction of rotation depends on the sign of the residual y-component. Initially the accumulator is initialized with zero and at the end of the vectoring operation; it will contain the traversed angle. Figure 5 represent the CORDIC vectoring mode.

The CORDIC elementary rotation equation in the vectoring mode is

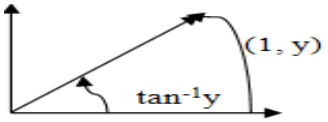

Figure 5: vectoring mode

$U_{i+1}= U_i – d_i.V_i.2^{-i}$

$V_{i+1}= V_i + d_i.U_i.2^{-i}$

$\theta_{i+1}= \theta_i – d_i.\tan^{-1}(2^{-i})$

where d_i= - sign(V_i). Once the vectoring operation is completed, the final results are

$$U_n= T_n \sqrt{U_0^2 + V_0^2}\sqrt{U_0^2 + V_0^2}$$

$V_n= 0$

$\theta_n= \tan^{-1}(V_0/U_0)$

$$T= \pi\sqrt{1+2^{-2i}}\ \pi\sqrt{1+2^{-2i}}$$

Where $[U_0\ V_0]^T$ and $[U_n\ V_n]^T$ represent the input and output vectors of vectoring process, respectively. T_n represent the processing gain of CORDIC algorithm and n represents the number of CORDIC algorithm iterations.

4.2 Rotation CORDIC

In the rotation mode, the vector magnitude and an angle of rotation are known and the coordinate (X-Y) components are computed after rotation. The CORDIC rotation mode algorithm begins by initializing an angle accumulator with the desired rotation angle. Next, the rotation decision at each CORDIC iteration is done in a way that decreases the magnitude of the residual angle accumulator. The rotation decision is based on the sign of the residual angle in the angle accumulator after each iteration. CORDIC rotation mode is shown in Figure 6.

d_i= sign (θ_i).

Once the rotation operation is completed the CORDIC equation for the rotation modes are

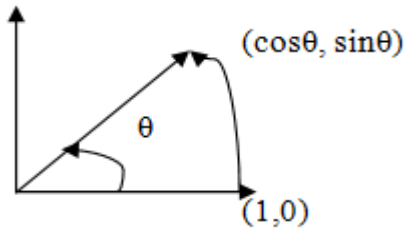

Figure 6: Rotation Mode

$U_{i+1}= U_i – V_i.d_i.2^{-i}$

$V_{i+1}= V_i + U_i.d_i.2^{-i}$

$\theta i+1 = \theta_i – d_i.\tan^{-1}(2^{-i})$

where d_i=-1 if θ_i<0, +1 otherwise.

i=0,1,2,……,n-1,and n is the total number of iteration and which provides the following result as N approaches to ∞:

$U_n = T_n\ [U_0 \cos\theta_0 – V_0 \sin\theta_0]$

$V_n= T_n\ [U_0 \cos\theta_0 – V_0 \sin\theta_0]$

$\theta_n = 0$

$$T_n = \pi \prod_{i=0}^{n-1} \pi \prod_{i=0}^{n-1} \sqrt{1+2^{-2i}}\sqrt{1+2^{-2i}}$$

In rotation mode, the CORDIC algorithm is limited to rotation angles between $–\pi/2$ to $\pi/2$ and to support angles outside of that range, quadrant correction is often used.

Figure 10 shows the CORDIC circuit for rotation mode operation. U_0 and V_0 are fed as set/reset input to a pair of input registers and the successive feedback values U_i and V_i at the i^{th} iteration are fed in parallel to the input registers.

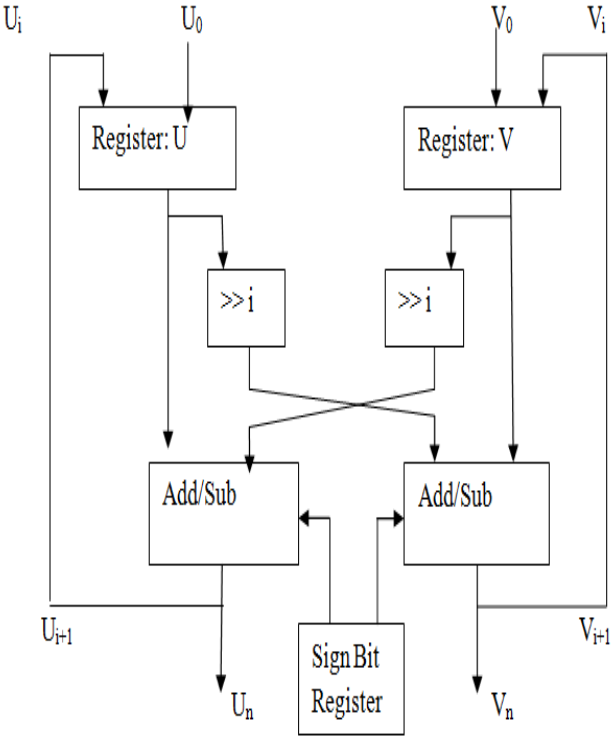

Figure 7: CORDIC Circuit for fixed rotation

5. Proposed System

In programmable MIMO Decoder architecture latency and area utilization by the processing core is large due to the presence of CORDIC rotation unit. The speed of operation of CORDIC unit is limited by the use of large number of iterations. Each rotation of CORDIC rotation takes several iterations to complete a single rotation. This thesis presents a simpler angle selection function for the Original CORDIC method which does not require the cycle time to be increased and achieves the advantages of the reduction in iteration count.

To overcome the disadvantages of the Conventional CORDIC algorithm of rotation unit in MIMO Decoder architecture, a new CORDIC algorithm with reduced the number of iterations is presented. In conventional CORDIC, each elementary angle needs to be performed sequentially so as to Complete the microrotation phase.

This new CORDIC algorithm removes sequential constraint in the microrotation phase as well as decreases the number of CORDIC iterations.

In programmable MIMO Decoder architecture, the input to a CORDIC rotation unit is a fixed and a known angle. Here present a optimization schemes for reducing number of rotations and scaling factor in a rotation mode of a CORDIC circuit. In the rotation mode, the components of a vector and an angle of the rotation are given and the coordinate components of the original vector, after rotation through the given angle, are computed. In every micro-rotation i, fixed angles of the value arctan (2^{-i}) are subtracted or added from/to the angle remainder θi, so that the angle remainder approaches zero.

5.1 Optimization of Micro Rotation

In rotation mode CORDIC, rotation of a vector through a known and fixed angle of rotation can be represented as a set of small number of predetermined elementary angle $\{\theta_i$, for $0 \le i \le r-1\}$,where $\theta_i = \arctan(2^{-M(i)})$ is the elementary angle of rotation used for i^{th} micro rotation in the CORDIC algorithm and r is the minimum necessary number of micro rotations.

Step 1: The initial optimization step perform the rotation mapping because the rotation through any angle, $0 < \theta < 2\pi$ can be mapped into a positive rotation through $0 < \phi < \pi/4$ without any extra arithmetic operations.

Step 2: The Number of elementary angle is minimized in the set$\{\theta_i\}$ according to the accuracy requirements.

The rotation mode CORDIC algorithm can be modified as

$$U_{i+1} = U_i - V_i.d_i.2^{-M(i)}$$

$$V_{i+1} = V_i + U_i.d_i.2^{-M(i)}$$

$$\begin{bmatrix} Ui+1 \\ Vi+1 \end{bmatrix}\begin{bmatrix} Ui+1 \\ Vi+1 \end{bmatrix} = \begin{bmatrix} 1 & -di.2-M(i) \\ di.2-M(i) & 1 \end{bmatrix} \begin{bmatrix} 1 & -di.2-M(i) \\ di.2-M(i) & 1 \end{bmatrix} \begin{bmatrix} Ui \\ Vi \end{bmatrix}\begin{bmatrix} Ui \\ Vi \end{bmatrix}$$

Such that for a minimum number of rotation r

$$\begin{bmatrix} U' \\ V' \\ \emptyset' \end{bmatrix}\begin{bmatrix} U' \\ V' \\ \emptyset' \end{bmatrix} = T \begin{bmatrix} Ur \\ Vr \\ 0 \end{bmatrix}\begin{bmatrix} Ur \\ Vr \\ 0 \end{bmatrix}.$$

Here the scale factor T depends on the θ_i. The CORDIC algorithm accuracy depends on how closely resultant rotation due to all micro rotation approximates to the desired angle of rotation θ, which in turn determines the change of actual rotation vector from the estimated value.

ALGORITHM I: Algorithm for micro rotation optimization

Step1**:** r=1
Step 2: do
Step 3: $\Delta\theta = \min | \theta - \sum_{i=0}^{r-1} arctan \sum_{i=0}^{r-1} arctan\ d_i\ 2^{-M(i)}|$, $d_i \varepsilon \pm 1$.M(i) is nonnegative integer.
Step 4: r= r+1
Step 5: while ($\Delta\theta > \varepsilon_\phi$)
End while

Algorithm for optimization of micro rotation start with the single micro rotation(r=1). The optimization algorithm minimize the objective function $\Delta\theta$ by searching for the appropriate parameters M(i) and di. If the micro rotation results in larger angle of deviation than ε_ϕ, then the micro rotation count is increased by one and optimization algorithm is repeated. To search the entire parameter for all combination of M(i) and d_i, a exhaustive search is employed in the optimization algorithm.

Table I: Optimization of Full Rotation with Four Microrotations

θ	M(0),d_0	M(1),d_1	M(2),d_2	M(3),d_3	Δθ
1	6,1	9,1	-----	----	0.007
2	4,1	5,0	9,0	-----	0.010
3	4,1	7,0	9,0	-----	0.017
4	3,1	5,0	9,1	9,1	0.003
5	3,1	5,0	9,1	9,1	0.001
6	3,1	5,0	6,0	8,0	0.006
7	3,1	9,0	----	9,0	0.013
8	3,1	5,1	6,1	9,1	0.020
9	3,1	5,1	9,1	----	0.027
10	2,1	4,0	7,0	10,1	0.011
11	3,1	4,1	8,1	10,1	0.019
12	2,1	2,0	5,1	8,0	0.026
13	1,1	2,0	7,1	----	0.024
14	2,1	10,1	----	----	0.035
15	2,1	6,1	10,1	----	0.013
16	1,1	2,0	4,1	9,1	0.007
17	1,1	2,0	4,1	6,1	0.000
18	1,1	3,0	7,0	----	0.008
19	1,1	3,0	7,0	----	0.008
20	1,1	3,1	7,1	9,1	0.000
21	2,1	2,1	3,0	10,1	0.003
22	1,1	4,0	6,0	9,1	0.018
23	1,1	4,0	8,1	----	0.011
24	1,1	5,0	6,0	9,1	0.008
25	1,1	5,0	8,1	----	0.001
26	1,1	7,0	9,0	10,0	0.004
27	1,1	7,1	----	-----	0.013
28	2,1	5,1	8,0	9,0	0.019
29	2,1	2,1	6,1	----	0.032
30	1,1	4,1	9,0	----	0.029
31	1,1	4,1	6,1	----	0.037
32	1,1	3,1	5,0	9,1	0.02
33	1,1	3,1	7,0	8,0	0.019
34	1,1	3,1	7,1	9,0	0.026
35	2,1	2,1	2,1	3,0	0.016
36	0,1	3,0	6,0	----	0.020
37	0,1	3,0	6,0	----	0.020
38	0,1	3,0	9,1	10,1	0.012
39	0,1	3,0	6,1	8,1	0.006
40	0,1	4,0	7,0	----	0.024
41	0,1	4,0	7,0	----	0.024
42	0,1	4,0	7,1	9,1	0.016
43	0,1	5,0	8,0	----	0.014
44	0,1	5,0	6,0	9,0	0.007
45	0,1	----	----	----	0.000

In experiments with the maximum input angular deviation ε_ϕ=0.04, only a set of four micro rotation is enough to complete the desired angle of rotation. In TABLE I, it is shown that rotations through any angle in the range $0 \le \theta \le 45^o$could be achieved with maximum angular deviation $\Delta\theta = 0.037^o$, where $\Delta\theta = |\theta - \theta_A|$. From the micro rotation

optimization table shows that a few elementary angles are sufficient to have a CORDIC rotation in the range [0, π/4], and different sets of elementary angles can be chosen according to the accuracy requirement.

The rotations through $0.1^0 \le |\theta| \le 2.0^0$ in an interval of 0.1^0 could be obtained by four micro rotations with angular deviation, $\sim 0.003^0$ is shown in the TABLE II.

Table II: Optimization of Small Rotation with Four Micro Rotations

θ^0	M(0),d_0	M(1),d_1	M(2),d_2	M(3),d_3	$\Delta\theta$
0.1	9,1	12,0	----	----	0.0021
0.2	8,1	11,0	14,1	-----	0.0007
0.3	7,1	9,0	11,0	13,0	0.0007
0.4	7,1	10,0	13,1	----	0.0013
0.5	7,1	10,1	----	-----	0.0036
0.6	6,1	8,1	10,0	12,0	0.0014
0.7	7,1	8,1	11,1	----	0.0006
0.8	6,1	9,0	12,1	----	0.0027
0.9	6,1	14,1	----	----	0.0013
1.0	6,1	9,1	13,0	----	0.0001
1.1	6,1	8,1	12,0	14,0	0.0015
1.2	5,1	7,0	9,0	11,0	0.0024
1.3	5,1	7,0	11,0	12,0	0.0003
1.4	6,1	7,1	10,1	----	0.0013
1.5	1,0	2,1	2,1	13,0	0.0004
1.6	5,1	8,0	11,1	13,1	0.0011
1.7	5,1	9,0	12,1	13,1	0.0010
1.8	5,1	13,1	----	----	0.0031
1.9	5,1	9,1	-----	----	0.0018
2.0	5,1	8,1	12,0	-----	0.0003

The algorithm for optimized micro rotation eliminates the need for a angle estimation data path required in the CORDIC circuit for fixed and known rotation. This is possible due to predetermined calculation of the elementary angles and direction of micro rotations for the given angle of rotation.

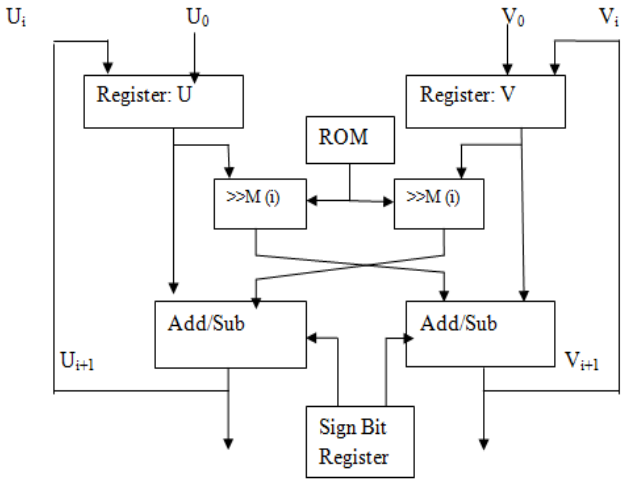

Figure 8: CORDIC circuit for optimized micro rotation

In micro rotation optimization algorithm, only a few elementary angles are used and the corresponding control bits are stored in a ROM of few words. The ROM memory contains the control bits for the number of shifts corresponding to the micro rotations to be implemented by the shifter unit and the direction of micro rotations stored in the sign bit register (SBR). The scaling factor depends on the elementary angle sets{ θ_i }. Based on the obtained micro rotations, different objective function is used for searching the parameters for scaling operation.

5.2 Optimization of Scaling Factor

In conventional CORDIC algorithm, a fixed scaling factor is used for scaling operation. Optimization in micro rotation results a different objective function for scaling factor calculation. Scale factor optimization is required in order to match with the optimized set of elementary angles for the micro rotations.

The generalized expression for scale factor is given by

$$T = \prod_{i=0}^{n-1}[\prod_{i=0}^{n-1}[(1+2^{-2\,M(i)})]^{-1/2}$$

This equation can be expressed for the selected set of r_1 micro rotations as

$$T = \prod_{i=0}^{r1-1}[\prod_{i=0}^{r1-1}[(1+2^{-2\,M(i)})]^{-1/2}$$

Where M(i) for $0 \le i \le r_1$ is the number of shifts in the i^{th} micro rotation. Expect for M(i) = 0(rotation by 45^o), by binomial expansion any term in equation can be written as

$$1 - \frac{x}{2}\frac{x}{2} + \frac{3x2}{8}\frac{3x2}{8} - \frac{5x35x3}{16\ 16} + \frac{35x435x4}{128\ 128} - \frac{63x563x5}{256\ 256} + \ldots\ldots\ldots$$

Where $x=2^{-2i}$, i is the number of shifts in a micro rotation, and can be expressed alternatively in terms of i as

$$1 - \frac{1}{2^{2i+1}}\frac{1}{2^{2i+1}} + \frac{1}{2^{4i+8}}\frac{1}{2^{4i+8}} - \frac{1}{2^{6i+4}}\frac{1}{2^{6i+4}} + \frac{1}{2^{8i+7}}\frac{1}{2^{8i+7}} - \frac{1}{2^{10i+8}}\frac{1}{2^{10i+8}} + \ldots..$$

Then the approximate scale factor as a product of shift add operation terms of the form

$$M_A = \prod_{i=0}^{r2-1}[1 + di\ 2^{-p(i)}]\prod_{i=0}^{r2-1}[1 + di\ 2^{-p(i)}]$$

Where p(i) is the number of shift performed for the i^{th} iteration of scaling, $d_i = \pm 1$, and r_2 is maximum number of scaling iterations required for the approximation.

ALGORITHM II: Algorithm for scaling optimization

Step1: $T = \prod_{i=0}^{r1-1}[\prod_{i=0}^{r1-1}[(1+2^{-2\,M(i)})]^{-1/2}$
Step 2: $r_2 = 1$
Step 3: do
Step 4: $\Delta T = \min \left| 1 - \frac{\prod_{i=0}^{r2-1}[1 + di\ 2^{-p(i)}]}{\prod_{i=0}^{r2-1}[1 + di\ 2^{-p(i)}]} / T \right|$, $d_i\ \varepsilon \pm 1$, p(i) is nonnegative integer.
Step 5: $r_2 = r_2 + 1$
Step 5: while ($\Delta T > \varepsilon_T$)
End while

ALGORITHM II is used to describe the scaling factor optimization scheme to the set of micro rotations obtained

by ALGORITHM I. The set of optimal micro rotation is obtained using ALGORITHM I, then ideal scaling factor T is calculated using equation. The objective function ΔT for the optimal scaling is defined as

$$\Delta T = |1 - T_A T|$$

The algorithm for scaling optimization starts with single term of scaling, i.e r_2=1. If the value of ΔT is larger than the given maximum deviation ε_T, then the number of scale factor term is increased by one and the optimization algorithm is run again. The value of ε_T need to be set as same value of ε_ϕ in the ALGORITHM I, since ΔT and Δθ contribute equally to the overall approximation error. The value of ε_T is set a 0.698 x 10^{-3}.

Table III: Optimized Shifts to Implement Scaling for the Case of Rotation with Four Micro Rotation

θ^0	$s(0),d_0$	$s(1),d_1$	$s(2),d_2$	$s(3),d_3$	kA	Δk
1	13,0	----	----	0.9999	0.9999	0.0019
2	9,0	----	----	0.9975	0.9980	0.5012
3	9,0	----	----	0.9980	0.9980	0.0267
4	7,0	----	----	0.9917	0.9922	0.3387
5	7,0	---	----	0.9918	0.9922	0.4295
6	7,0	----	----	0.9916	0.9922	0.5359
7	7,0	----	----	0.9923	0.9922	0.0892
8	7,0	----	----	0.9916	0.9922	0.5359
9	7,0	----	----	0.9918	0.9922	0.3989
10	7,0	----	----	0.9917	0.9922	0.4103
11	7,0	9,0	----	0.9903	0.9903	0.0887
12	3,0	7,0	----	0.8676	0.8682	0.6915
13	3,0	7,0	----	0.8677	0.8682	0.5402
14	5,0	10,1	----	0.9700	0.9697	0.3092
15	5,0	10,1	----	0.9700	0.9697	0.3377
16	3,0	7,0	9,0	0.8659	0.8665	0.6929
17	3,0	7,0	9,0	0.8659	0.8665	0.6261
18	3,0	10,0	10,0	0.8874	0.8877	0.3380
19	3,0	10,0	10,0	0.8875	0.8878	0.3502
20	3,0	6,1	10,0	0.8874	0.8873	0.1126
21	4,0	8,0	----	0.9339	0.9338	0.0752
22	4,0	4,0	6,1	0.8925	0.8924	0.0337
23	4,0	4,0	6,1	0.8927	0.8926	0.0518
24	4,0	5,0	6,0	0.8938	0.8940	0.2237
25	4,0	5,0	6,0	0.8940	0.8940	0.0319
26	4,0	5,0	6,0	0.8943	0.8944	0.1118
27	4,0	5,0	6,0	0.8944	0.8940	0.4332
28	5,0	7,0	9,0	0.9696	0.9697	0.1031
29	4,0	8,1	----	0.9411	0.9412	0.1068
30	4,0	4,0	6,1	0.8926	0.8926	0.0703
31	4,0	4,0	6,1	0.8926	0.8926	0.0703
32	3,0	6,1	10,0	0.8870	0.8873	0.3382
33	3,0	6,1	10,0	0.8875	0.8878	0.3578
34	3,0	6,1	10,0	0.8874	0.8875	0.1126
35	3,0	5,1	8,1	0.9060	0.9059	0.1721
36	2,0	4,0	9,0	0.7016	0.7018	0.2721
37	2,0	4,0	9,0	0.7016	0.7017	0.2721
38	2,0	4,0	9,0	0.7016	0.7018	0.2721
39	2,0	4,0	9,0	0.7016	0.7018	0.2798
40	2,0	4,0	8,1	0.7057	0.7059	0.2315
41	2,0	4,0	8,1	0.7057	0.7059	0.2315
42	2,0	4,0	8,1	0.7057	0.7058	0.1417

In order to get scaling optimization for angle 43° and 45°with desired accuracy and less number of iterations with the above approach, the scaling factor can be expressed as

$$T = 1 - \frac{1}{2^2}\frac{1}{2^2} - \frac{1}{2^4}\frac{1}{2^4} + \frac{1}{2^6}\frac{1}{2^6} + \frac{1}{2^8}\frac{1}{2^8}$$

Equation can be expressed in recursive shift add terms

$$T = 1 - \frac{1}{2^2}\frac{1}{2^2}\left(1 + \frac{1}{2^2}\frac{1}{2^2}\right)\left(1 - \frac{1}{2^4}\frac{1}{2^4}\right)$$

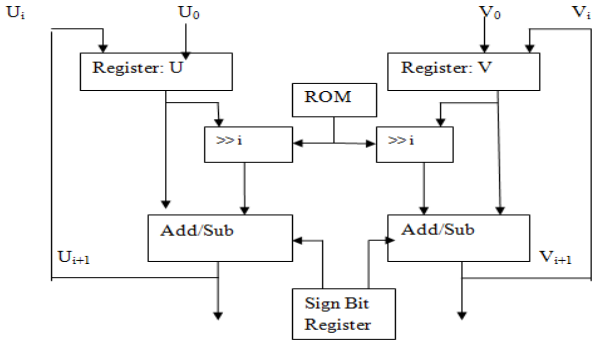

Figure 9: Shift-add scaling circuit

The Figure 9 shows the shift-add scaling circuit. To speed up the computation time of the CORDIC algorithm, either the number of iterations or the delay of each iteration has to be minimized. The proposed algorithm introduces a novel approach, in which reduce the number of micro rotation by encoding the angle of rotation as a linear combination of a set of four selected elementary angles of micro rotations. The rotation direction and elementary angle of rotations is precomputed for angles in the range $0 \leq \theta \leq \pi/4$ and variable scaling factor for each micro rotation is stored in a table. Hence, a significant speedup of the delay per iteration is obtained. The proposed architecture also eliminates the θ-datapath and reduces the area of the implementation by reduction in number of micro rotation.

6. Results and Discussion

The modules are modeled using Verilog in Xilinx ISE Design Suite 14.2 and the simulation of the design is performed using ISim simulator to verify the functionality of the design. The simulation results for design of a programmable MIMO Decoder with CORDIC as rotation unit is given below. The entire operations of the MIMO Decoder are divided into four modes of operation. Mode 0: Reciprocal operation, Mode 1: Multiplication operation, Mode 3: Addition/subtract operation, Mode 4: CORDIC Rotation. The input to the MIMO Decoder is a complex numbers. Two complex numbers are available at the processing core input in order to perform different modes of operation.

The input and output data is in complex type. The 3 bit mode [2:0] is used for selecting the different modes of operation.real_input_to_core_a,imagi_input_to_core_b,real_input_to_core_c,imagi_input_to_core_id are the real and imaginary part of two complex numbers.

real_output_a,imaginary_output_ib,sine_out_a,cosine_out_a,sine_out_b,cosine_out_b is the real and imaginary part of processing core outputs. The value given for all the 32 input sample. At 0th clock cycle, reset signal is enabled.At 1st clock cycle, reset signal is disabled and the first input sample is given and rest of the input sample is given at consecutive clock cycles. The entire MIMO Decoder is enable using enable_decoder signal. The write and read enable signal is used to enable write and read operation.

select input is used for selecting addition /subtract operation.

Simulation results of 8-microrotation CORDIC unit is shown in Figure 10 and corresponding optimized 4-microrotation CORDIC is shown in Figure 11.

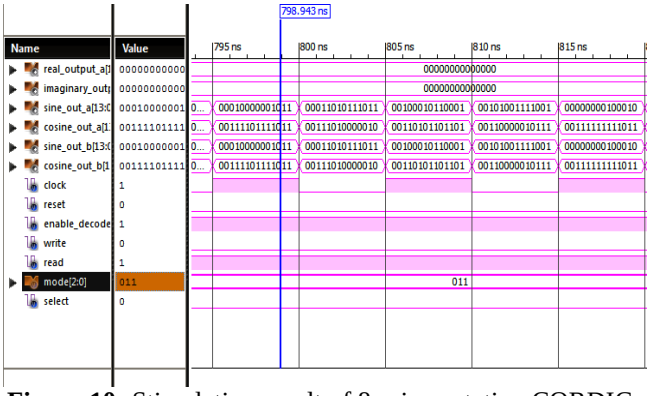

Figure 10: Stimulation result of 8-microrotation CORDIC unit

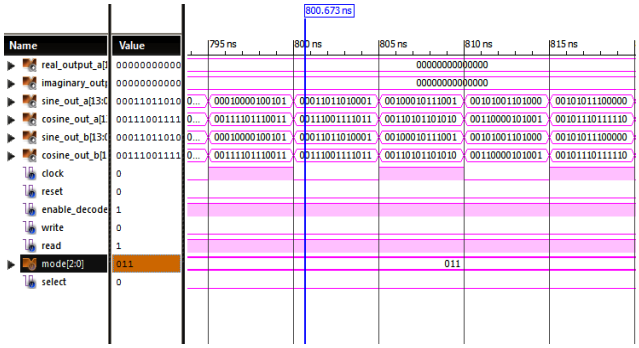

Figure 11: Stimulation result of optimized 4-microrotation CORDIC unit

Comparison of Area utilization in terms of number of slice LUTs and total processing time of 8-microrotation CORDIC and optimized 4-microrotation CORDIC in programmable MIMO Decoder Architecture by synthesize the design in Virtex 5 xc5vlx110T is shown in TABLE IV.

Table IV: Comparison of 8-microrotation CORDIC unit and optimized 4-microrotation CORDIC in programmable MIMO Decoder Architecture

Architecture	Number of slices	Processing time (ns)
8-micro rotation CORDIC unit	2759	43.614
Optimized 4-micro rotation CORDIC unit	1389	33.947

7. Conclusion

MIMO OFDM plays an important role in most of the emerging wireless communication technologies. This work proposes a programmable MIMO decoder architecture with reduced area and processing time by reducing the micro rotation of the CORDIC rotation unit in existing MIMO decoder architecture. Here an optimization scheme has been used for micro rotation and scaling in to the CORDIC unit in the programmable MIMO decoder architecture. The proposed method adaptively select the appropriate iteration steps and converges to the final results by executing only four micro rotation as compared with increased number of rotations in the conventional CORDIC. The proposed algorithm perform all the angle of rotation within the range of 0^0 to 45^0 by angle folding method and which results in a pre-computation of micro rotations for all the rotation angles in the range 0^0 to 45^0. This has eliminated the need for micro rotation calculation steps in the conventional CORDIC and thus entire processing time for rotation operation is reduced. Also, optimization scheme used in this work has reduced chip area requirement for the MIMO decoder, which in turn results in the reduction of power consumption.

References

[1] Mohamed I. A. Mohamed, Karim Mohammed and Babak Daneshrad "Energy Efficient Programmable MIMO Decoder Accelerator Chip in 65-nm CMOS " IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 7, pp. 1481–1490, July. 2014.

[2] Helmut Bolcskei and ETH Zurich," MIMO-OFDM Wireless System: Basics,Perspectives, and Challenges", IEEE wireless communications, August 2006,pp.31-37.

[3] C.-H. Yang and D. Markovic, "A flexible DSP architecture for MIMO sphere decoding," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 56, no. 10, pp. 2301–2314, Oct. 2009.

[4] C.-J. Huang, C.-W. Yu, and H.-P. Ma, "A power-efficient configurable low-complexity MIMO detector," IEEE Trans. Circuits Syst. I, Reg.Papers, vol. 56, no. 2, pp. 485–496, Feb. 2009.

[5] J. Eilert, D. Wu, and D. Liu, "Implementation of a programmable linear MMSE detector for MIMO-OFDM," in Proc. IEEE ICASSP, Apr. 2008, pp. 5396–5399.

[6] K. Mohammed, M. I. A. Mohamed, and B. Daneshrad, "A parameterized programmable MIMO decoding architecture with a scalable instruction set and compiler," IEEE Trans. Very Large Scale Integr. (VLSI) Syst.,vol. 19, no. 8, pp. 1485–1489, Aug. 2011.

[7] K. Mohammed and B. Daneshrad, "A MIMO decoder accelerator for next generation wireless communications," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 11, pp. 1544–1555, Nov. 2010.

[8] M. Ali, K. Mohammed, and B. Daneshrad, "MIMO accelerator: A design flow for a programmable MIMO decoder architecture," in Proc. Comput. Conf. Rec.

43rd Asilomar Conf. Signals, Syst., Nov. 2009, pp. 1292–1296.

[9] J. E. Volder, "The CORDIC trigonometric computing technique," IRE Trans. Electron. Comput., vol. EC-8, pp. 330–334, Sep. 1959.

[10] J.S Walther. "A Unified Algorithm for Elementary Functions", AIFS Spring Joint Computer Conference, pp.375-385, 1971.

[11] Y. H. Hu and S. Naganathan, "An angle recoding method for CORDIC algorithm implementation," IEEE Trans. Comput., vol. 42, no. 1, pp.99–102, Jan. 1993.

[12] 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.

[13] Pramod Kumar Meher and Sang Yoon Park, " CORDIC Designs for Fixed Angle of Rotation" IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 21, No. 2,pp.217-228 Feb. 2013