



LOW COMPLEXITY ARITHMETIC MEAN DECOMPOSITION BASED PRE-CODING FOR MIMO SYSTEMS AND ITS FPGA IMPLEMENTATION

Bharani dharan N., Chinna thambi M. and Rajaram S.

Department of Electronics and Communication, Thiagarajar College of Engineering, Madurai, India

E-Mail: bharani2410@gmail.com

ABSTRACT

This paper presents the novel approach for pre-coding in multiple input and multiple output systems using arithmetic mean decomposition. This proposed decomposition scheme has low complexity than popular geometric mean decomposition. CORDIC based Givens rotation is used for making the channel matrix as bi-diagonal matrix and then 2×2 singular value decomposition and 2×2 arithmetic mean decomposition operations are performed recursively on the bi-diagonal matrix. As a result upper triangular matrix with equal diagonal elements whose value equal to their arithmetic mean is achieved. The total system is implemented in Xilinx virtex-7 FPGA and accuracy analysis is also made.

Keywords: FPGA, CORDIC algorithm, givens rotation, matrix decomposition.

1. INTRODUCTION

Multiple Input and Multiple Output (MIMO) is an efficient technology in wireless communications for spatial multiplexing and it will increase spectral efficiency for a given total transmit power. There are number of advantages in MIMO systems over single-antenna-to-single-antenna communication such as reduction in sensitivity to fading by the spatial diversity provided by multiple spatial paths. On assuming flat fading MIMO channel, the input-output relationship can be mathematically modeled as,

$$y = Hx + z \quad (1)$$

where the matrix H is the channel impulse response, the vectors x and y are the discrete-time transmitted and received components, respectively, and z is Additive White Gaussian Noise (AWGN).

Pre-coding is an important concept in MIMO systems as it converts complex MIMO channel matrix into simple parallel Single Input and Single Output (SISO) channel matrix. Earlier Singular Value Decomposition (SVD) is used widely for pre-coding in MIMO systems. SVD decomposes the channel matrix (H) into three matrices [1] as follows:

$$H = U \Sigma V^H \quad (2)$$

where, Σ is the diagonal matrix with singular values as diagonal, U and V are both unitary matrices such that $UU^H = I$ and $VV^H = I$.

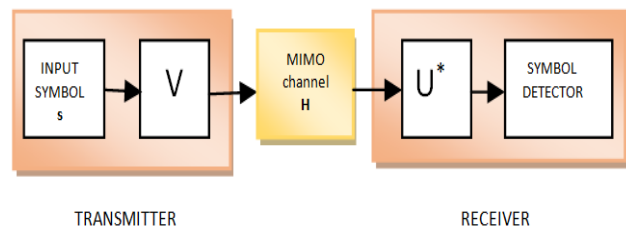


Figure-1. Pre-coder concept in MIMO systems.

As in Figure-1, V is used as the pre-coder matrix and U^H is used as the combiner matrix. Then the output of combiner will be,

$$y = \Sigma x + U^H z \quad (3)$$

Equation (3) is simply a set of parallel scalar sub-channel I/O relationships. The main problem with SVD is its non-uniform diagonal elements i.e., if some of the singular values are very small, then throughput and performances are affected. The Geometric Mean Decomposition (GMD) solves this problem by making all diagonal elements of Σ equal to the Geometric Mean (GM) of the channel singular values [2-6]. Geometric mean is found such that,

$$GM = \left(\prod_{i=1}^n a_i \right)^{\frac{1}{n}} \quad (4)$$

But the GMD scheme needs complex square root operations for finding its GM i.e., n^{th} root operation is needed for finding the GM of 'n' numbers. Hence it is more complex and it will occupy more area on FPGA. Arithmetic Mean Decomposition (AMD) does not involve any square root operations to find its Arithmetic Mean (AM). AM is simply the addition of all elements and then dividing it by number of elements.

$$AM = \frac{\sum_{i=1}^n a_i}{n} \quad (5)$$



In AMD scheme, the channel matrix H is decomposed into three matrices such that,

$$H = Q R P^H \tag{6}$$

where, R is an upper triangular matrix with equal diagonal elements; the diagonal value will be equal to AM. Q and P are both unitary matrices such that $Q Q^H = I$ and $P P^H = I$. Though R is an upper triangular matrix, it can be easily converted into diagonal matrix by back-substitution method. AMD scheme can be implemented by converting the channel matrix into bi-diagonal matrix using CORDIC based Givens rotation. 2x2 SVD and 2x2 AMD operations are applied to the bi-diagonal matrix to convert it into upper triangular matrix with equal diagonal elements.

2. CORDIC ALGORITHM

The COordinate Rotation DIgital Computer (CORDIC) algorithm is a iterative method used to compute some mathematical functions that include multiplication, division and elementary trigonometric functions by means of planar rotation and vectoring [7-8]. The final equations of CORDIC algorithm are given in equations (1) and (2).

$$\begin{bmatrix} X_{n+1} \\ Y_{n+1} \end{bmatrix} = \cos\theta_n \begin{bmatrix} 1 & -S_n 2^{-n} \\ S_n 2^{-n} & 1 \end{bmatrix} \tag{7}$$

Here, $\cos\theta_n$ can be treated as a constant K and computed at the end and multiplications are replaced with shift operations. Residue Z which defines the angle difference between the expected rotation and the iterative rotations is:

$$Z_{n+1} = Z_n - s_n \arctan\left(\frac{1}{2^n}\right) \tag{8}$$

There are two different modes of CORDIC algorithm as rotation and vector mode. In the rotation mode, CORDIC is used for converting a vector from polar form to rectangular form and vector mode makes the reverse operation. The parameter of rotation S_n in rotation mode is given as,

$$S_n = \begin{cases} -1 & \text{if } Z_n < 0 \\ 1 & \text{if } Z_n \geq 0 \end{cases} \tag{9}$$

The parameter of rotation S_n in vector mode is given as,

$$S_n = \begin{cases} 1 & \text{if } y_n < 0 \\ -1 & \text{if } y_n \geq 0 \end{cases} \tag{10}$$

With proper selection of initial values of X_i , Y_i and Z_i , the required function is performed using above equations. In rotation mode, Z is driven to zero. With

initial values $X_i=1$, $Y_i=0$, and $Z_i=0$, at the end of the final iteration:

$$\begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} \tag{11}$$

In vector mode, Y is driven to zero instead of Z and with $X_i=X$, $Y_i=Y$, and $Z_i=0$ as initial values, the final iteration result will be,

$$\begin{bmatrix} X_n \\ Y_n \\ Z_n \end{bmatrix} = \begin{bmatrix} K \sqrt{X_i^2 + Y_i^2} \\ 0 \\ \arctan Y_i/X_i \end{bmatrix} \tag{12}$$

Initially, $X_i=1$ and $Y_i=0$ should be the value of sine and cosine value generation. If the initial conditions are $X_i=1$ and $Y_i=0$ then the resulting discrete sine and cosine values will vary from -1 to 1. These fractional values are not realizable in FPGA easily. Hence to make the discrete sine and cosine values to vary from -100 to 100, the initial values are multiplied by 100 i.e., $X_i=100$ and $Y_i=0$.

For representing sine and cosine values in the above range 8 bit CORDIC is used as it can represent -128 to 127. Generally the constant K as in the equation (1) should be multiplied to the final results after end of iteration. But to improve the accuracy, it is multiplied to initial conditions itself. $K=0.611$ for six stages of CORDIC and so initial values are given as $X_i=61(100 \times 0.611)$, $Y_i=0$. These initial values are shifted by i bits, where i is the integer {0, 1, 2, 3, 4, 5} which makes division of x and y by 1, 2, 4, 8, 16, 32 for each stage. In this rotation mode, the given vector is iteratively rotated to form new vectors at the intermediate stages to get the desired angle, Z_i .

3. IMPLEMENTATION OF BI-DIAGONAL MATRIX

Generally the Givens rotation can be represented in matrix form as,

$$R(i, j, \theta) = \begin{pmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \dots & c & \dots & -s & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & s & \dots & c & \dots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \tag{13}$$

Here $c = \cos\theta$ and $s = \sin\theta$ which appear at the intersections of i^{th} and j^{th} rows and columns. The non-



zero elements of the above Givens rotation matrix is given by:

$$r_{ii} = c ; r_{jj} = c ; r_{ji} = -s; r_{ij} = s \text{ for } j \leq i ;$$

$$r_{kk} = 1 \text{ for } k \neq i, j$$

The product of $R(i, j, \theta).H$ represents the counterclockwise rotation of the vector H in the (i, j) plane for θ radians. By using this Givens rotation, zeros can be introduced into the matrix H for making it upper triangular matrix as follows:

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix} \tag{14}$$

The product of $R(i, j, \theta).H$ gives the row wise nullification while the product of $H.R(i, j, \theta)$ gives column wise nullification. For converting the channel square matrix into bi-diagonal matrix both row wise and column wise nullification must be used. In fig.2, black dots indicate non-zero elements and white dots indicate zero. Six row-wise nullification and three column wise nullifications are used for this process. CORDIC algorithm in both rotation mode and vector mode is used for each nullification.

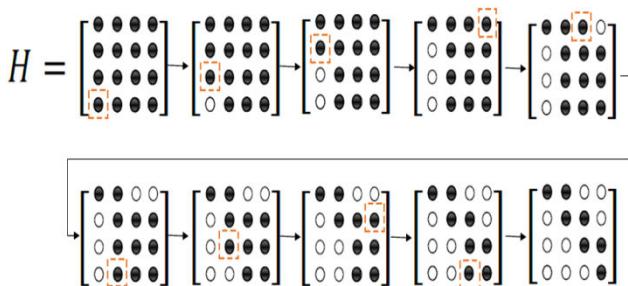


Figure-2. Conversion of square matrix into bi-diagonal matrix.

4. IMPLEMENTATION OF AMD FROM BI-DIAGONAL MATRIX

After matrix bi-diagonalization, the arithmetic mean of the diagonal elements (present in bi-diagonal matrix) is first calculated. AMD is then accomplished by equalizing all diagonal elements to this AM value using unitary matrix operations. To avoid the iteration problem, a scheme starting with the AMD of a 2×2 sub-matrix and recursively proceeding to the AMD of the final $N \times N$ matrix is proposed. In each recursion, a fixed number of 2×2 SVD and 2×2 AMD operations are applied, and

the total number of operations needed is deterministic with respect to N . In the first step, the 2×2 SVD computation transforms the matrix into a diagonal matrix using two unitary rotations. Singular values σ_1 and σ_2 are both positive. The 2×2 SVD scheme is given as,

$$\begin{bmatrix} \cos\theta_r & -\sin\theta_r \\ \sin\theta_r & \cos\theta_r \end{bmatrix} \begin{bmatrix} p & q \\ 0 & s \end{bmatrix} \begin{bmatrix} \cos\theta_c & \sin\theta_c \\ -\sin\theta_c & \cos\theta_c \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \tag{15}$$

θ_r and θ_c present in equation (15) is given by,

$$\theta_r = \frac{\tan^{-1}\left(\frac{q}{s-p}\right) - \tan^{-1}\left(\frac{-q}{s+p}\right)}{2} \tag{16}$$

$$\theta_c = \frac{\tan^{-1}\left(\frac{q}{s-p}\right) + \tan^{-1}\left(\frac{-q}{s+p}\right)}{2} \tag{17}$$

In the second step, the 2×2 AMD computation converts the leading diagonal element to a desired AMD value δ . The 2×2 AMD scheme is given as,

$$\begin{bmatrix} \cos\theta_a & \sin\theta_a \\ -\sin\theta_a & \cos\theta_a \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} \cos\theta_b & -\sin\theta_b \\ \sin\theta_b & \cos\theta_b \end{bmatrix} = \begin{bmatrix} \delta & x \\ 0 & y \end{bmatrix} \tag{18}$$

θ_a and θ_b present in equation (18) is given by,

$$\theta_a = \tan^{-1}\left(\sqrt{\frac{\sigma_1^2 - \delta^2}{\delta^2 - \sigma_2^2}}\right) \tag{19}$$

$$\theta_b = \tan^{-1}\left(\frac{\sigma_2 \cdot \sin\theta_a}{\sigma_1 \cdot \cos\theta_a}\right) \tag{20}$$

If δ is the geometric mean of σ_1 and σ_2 , the value y in (18) becomes δ as well. The cosine and sine values required in 2×2 SVD and AMD can be realized by using the CORDIC algorithm in rotation mode. The clear illustration of the conversion process from bi-diagonal matrix to AMD is shown in Figure-3. It is shown how the AMD is accomplished from a 2×2 sub-matrix extending to the complete 4×4 matrix. On comparing with the conventional SVD-based AMD scheme, this proposed scheme is completely free of the convergence problem in SVD computation.

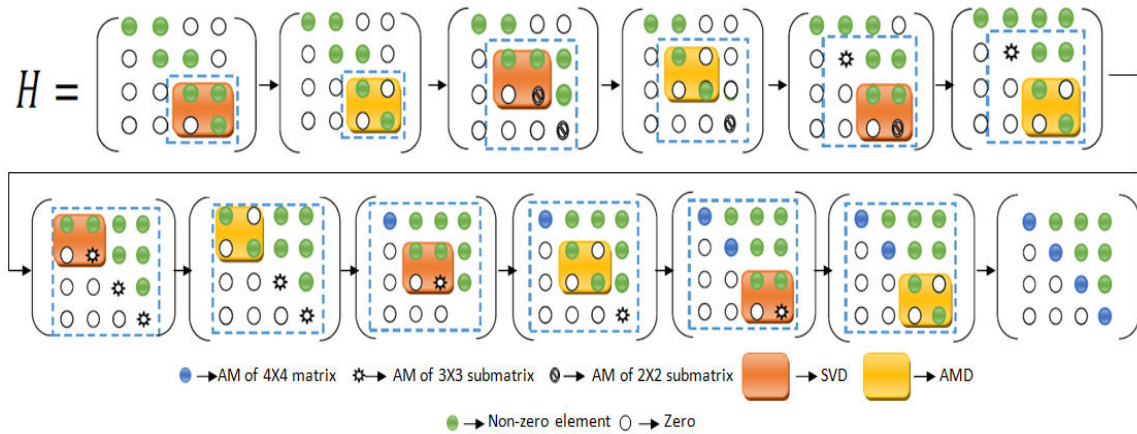


Figure-3. Implementation of AMD scheme from bi-diagonal matrix

This type of recursive approach gives a fixed computational complexity and, so constant throughput is achieved. The matrix Q is easily computed from the product of all row-wise rotations performed and taking transpose for it while P^H is computed from the product of all column-wise rotations performed.

The channel square matrix and AMD computation operations may contain fractional values. It is very important to realize those fractional values in FPGA and for this purpose floating point arithmetic can be used. But floating point arithmetic needs 32 bits for representing each value which will cost much more area. Hence integer arithmetic is used and input values are multiplied by 1000 and number of bits used for representing each value will be less than 13 bits.

5. RESULTS AND DISCUSSIONS

The total AMD scheme for 4x4 channel matrix is stimulated and implemented in Xilinx- xc7v2000t-2flg1925 FPGA. The simulation results for AMD scheme is shown in Figure-4. In the simulation diagram, 'H' is the 4x4 channel input matrix while 'Bid' is the bi-diagonal matrix computed and 'R' is the final upper diagonal matrix with equal diagonal elements. It is checked that, for all the numerical values, this scheme is working properly and more accurate. But there is a little degradation always in the final diagonal element (In R, the first three diagonal values are 2 but final diagonal value is 1). Pre-coder and combiner matrices are found easily from the transpose of product of rotation matrices.

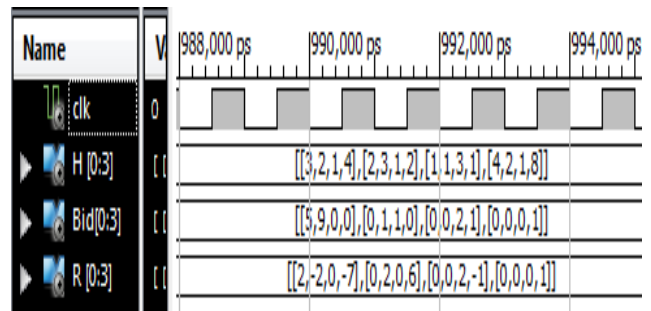


Figure-4. Simulation results of AMD scheme.

Table-1 shows the implementation summary of AMD scheme in Xilinx FPGA. It also includes the combinational path delay which is constant for varying values of H. It is the time taken for the finding the Q, R, P^H matrix from the H matrix. This time is an important parameter in MIMO pre-coding systems as the channel matrix H may vary quickly and pre-coding should be done before H varies to another value. For this scheme, combinational path delay is less and so it satisfies the requirement.

The usage of integer arithmetic instead of floating point arithmetic is the main reason for less area consumption. Accuracy analysis is also made and it is observed that there is very less percentage of error. The usage of eight bit CORDIC algorithm with only six stages is also the reason for less area consumption. In real time application, the values of channel matrix will be complex in nature but the same concept of Givens rotation using CORDIC can be first used to convert the complex value matrix into real value matrix and then AMD scheme can be performed.

**Table-1.** Implementation summary of amd scheme in xilinx.

Logic utilization	Used	Available	Utilization
Number of slice registers	5963	2443200	0%
Number of slice LUTs	12138	1221600	0%
Number of fully used LUT-FF pairs	5522	12579	43%
Number of bonded IOBs	513	1200	42%
Number of BUFG/BUFGCTRLs	1	128	0%
Number of DSP48E1s	1156	2160	53%
Combinational path delay	208 ns	-	-

6. CONCLUSIONS

In this paper, a detailed analysis of AMD scheme based on CORDIC is made. From the implementation results, it is well observed that area consumption and combinational path delay are very less because of low complexity AMD scheme. The percentage of error is also less and this design can be easily extended to the larger channel matrix sizes. In future, the percentage of error and combinational path delay can still be reduced and this scheme can be improvised further.

REFERENCES

- [1] Xinying Wang and Joseph Zambreno, "An FPGA Implementation of the Hestenes-Jacobi Algorithm for Singular Value Decomposition", IEEE 28th International Parallel and Distributed Processing Symposium Workshops 2014.
- [2] Wei-Da Chen and Yin-Tsung Hwang, "A Constant Throughput Geometric Mean Decomposition Scheme Design for Wireless MIMO Precoding", IEEE transactions on vehicular technology, vol. 62, no. 5, June 2013.
- [3] Chiao-En Chen and Chia-Hsiang Yang "An Iterative Geometric Mean Decomposition Algorithm for MIMO Communications Systems", IEEE transactions on wireless communications, Volume: 14, Issue: 1 August 2014.
- [4] Shaowei Lin, Winston W. L. Ho, and Ying-Chang Liang, "Block Diagonal Geometric Mean Decomposition (BD-GMD) for MIMO Broadcast Channels", IEEE transactions on wireless communications, VOL. 7, NO. 7, July 2008.
- [5] Feng Liu, Lingge Jiang and Chen He, "Advanced Joint Transceiver Design for Block-Diagonal Geometric-Mean-Decomposition-Based Multiuser MIMO Systems", IEEE transactions on vehicular technology, vol. 59, no. 2, February 2010.
- [6] Yin-Tsung Hwang, Wei-Da Chen, and Cheng-Ru Hong, "A Low Complexity Geometric Mean Decomposition Computing Scheme and Its High Throughput VLSI Implementation", IEEE transactions on circuits and systems-I: regular papers, vol. 61, no. 4, April 2014.
- [7] J.E. Volder, "The CORDIC Trigonometric Computing Technique", IRE Transactions on Electronic computer, vol. EC-8, pp. 330-334, 1959.
- [8] J. Walther, "A unified algorithm for elementary functions," proc. Spring joint comp. con & vol.38, pp.379-385, 1971.