A GENERALIZED MATRIX-DECOMPOSITION PROCESSOR FOR JOINT MIMO TRANSCEIVER DESIGN

Yu-Chi Wu and Pei-Yun Tsai

Department of Electrical Engineering, National Central University, Taoyuan, Taiwan

ABSTRACT

A generalized matrix-decomposition processor is designed and implemented, which supports QR decomposition (QRD), eigenvalue decomposition (EVD), and geometric-mean decomposition (GMD), to accelerate computations in MIMO precoding/beamforming systems. The processor adopts memory-based architecture with 16 processing elements (PEs) each consisting of one CORDIC module. An improved GMD algorithm is proposed, which reduces 13.2% complexity and can be implemented by homogeneous CORDIC operations. The EVD adopts the Rayleigh quotient shift and deflation technique to accelerate convergence. The basis computations can be accomplished by mirrored operations during channel matrix decomposition. From the implementation results, the generalized processor achieves decomposition throughput of 10M, 0.99M, 2.96M matrixes per second for 4×4 complex QRD, EVD and GMD.

Index Terms **—** MIMO precoding, QRD, EVD, SVD, GMD.

1. INTRODUCTION

Multiple-input multiple-output (MIMO) techniques are regarded as one of the promising solutions to bring capacity gain, diversity gain and beamforming gain. Recent wireless communication systems, such as 3GPP Long-term evolution (LTE)/LTE-A and IEEE 802.11n/ac, often incorporate open-loop MIMO and close-loop MIMO to exploit these gains so that the transmission efficiency in terms of either spectral efficiency or power efficiency can be enhanced.

Matrix decomposition is often required in close-loop MIMO systems. QRD can triangularize the channel matrix [1], and thus with the help of Tomlinson-Harashima precoding (THP) at the transmitter or successive interference cancellation (SIC) at the receiver, the inter-antenna-interference (IAI) can be eliminated. Singular value decomposition (SVD) is also an essential technique to decompose the spatially coupled channels into parallel subchannels, which is adopted in IEEE 802.11n/ac [2]. However, the subchannel with the worst channel gain usually deteriorates the system bit-error-rate performance. Hence, the GMD then has been proposed for the transceiver design [3], which results in identical signal-to-noise ratios for all the subchannels. The EVD is employed for searching the subspace to achieve interference alignment [4] or to maximize signal-to-interference-and-noise ratio [5]. Consequently, it is clear that a generalized matrix-decomposition processor will be a key component for supporting advanced MIMO precoding or beamforming techniques.

Dedicated hardware implementations related with matrix decomposition for MIMO systems have been proposed in [6]-[10]. In [6], QRD for 8×8 and 4×4 MIMO precoding is designed with the sorting capability. In [7] and [8], high-throughput and pipelined architecture is developed for 4×4 GMD. The authors also show that bidiagonalization plus successive 2×2 SVD and 2×2 GMD can be more efficient in computation than the full SVD-based approach, which needs iterative SVD preprocessing. In [8], a divide-and-conquer approach is proposed to further reduce the complexity and to increase the parallelism for throughput enhancement. A generalized triangular decomposition (GTD) processor is designed in [9] based on systolic array for computing 1×1 to 8×8 GTD. SVD pre-processing is adopted and thus the architecture also supports SVD. In [10], 4×4 SVD is realized by a pipelined architecture for high throughput applications. An eigen-solver is implemented in a semi-systolic array in [11]. It is worth noting that all these implementations use Givens rotation algorithm computed by coordinated rotation digital computer (CORDIC).

In this paper, the architecture of a generalized matrix-decomposition processor is designed. The processor aims to compute the results of QRD, EVD, and GMD for accelerating signal processing in base-station supporting various MIMO precoding techniques. Since SVD can be calculated by EVD, it implies that our processor also supports SVD-based precoding. To offer configurability and flexibility, the memory-based architecture is used instead of pipelined architecture. The processing element (PE) is composed of the CORDIC module that is configured for different operations, such as circular rotation, circular vectoring, hyperbolic rotation, etc. To achieve efficient computation, bidiagonalization and Hessenburg reduction are adopted for GMD and EVD in the first phase. Then, a new parallel processing approach for GMD processing is proposed and we will show the reduction in computational complexity as opposed to the prior method. The Rayleigh quotient shift and deflation are adopted for EVD to accelerate the convergence. Floating-point representation is used to cover the wide dynamic range of different matrix decomposition schemes. Implementation results and comparisons are provided to show the capability of our matrix-decomposition processor.

2. MATRIX DECOMPOSITION ALGORITHMS

In MIMO systems, received signal $\mathbf{y} \in \mathbb{C}^{N\times 1}$ of N receive antennas can be expressed as

$$\mathbf{y} = \mathbf{HFx} + \mathbf{n} \tag{1}$$

where $\mathbf{x} \in \mathbb{C}^{N\times 1}$ is the transmitted signal from N transmit antennas; $\mathbf{H} \in \mathbb{C}^{N\times N}$ is the MIMO channel response; $\mathbf{n}$ is the noise vector. The precoding matrix $\mathbf{F}$ of size $N \times N$ can be obtained by various matrix decomposition schemes depending on the requirements.

- For QRD, $\mathbf{H}^H = \mathbf{Q}_{QR}\mathbf{R}_{QR}$. $\mathbf{F} = \mathbf{Q}_{QR}$.
- For SVD, $\mathbf{H} = \mathbf{U}_{SVD}\mathbf{\Sigma}_{SVD}\mathbf{V}_{SVD}^{\boldsymbol{H}}$. $\mathbf{F} = \mathbf{V}_{SVD}$.
- For GMD, $\mathbf{H} = \mathbf{Q}_{GMD}\mathbf{R}_{GMD}\mathbf{P}_{GMD}^H$. $\mathbf{F} = \mathbf{P}_{GMD}$.

- For EVD, $\mathbf{C} = \mathbf{U}_{EVD}\mathbf{\Lambda}_{EVD}\mathbf{U}_{EVD}^{\boldsymbol{H}}$, where $\mathbf{C}$ is the covariance matrix related with signal, interference, and noise.

In these decomposition schemes, matrixes $\mathbf{R}_{QR}$ and $\mathbf{R}_{GMD}$ are upper triangular; matrixes $\mathbf{\Sigma}_{SVD}$ and $\mathbf{\Lambda}_{EVD}$ are diagonal; matrixes $\mathbf{Q}_{QR}$, $\mathbf{U}_{SVD}$, $\mathbf{V}_{SVD}$, $\mathbf{Q}_{GMD}$, $\mathbf{P}_{GMD}$, $\mathbf{U}_{EVD}$ are all unitary. All the decompositions mentioned above can be accomplished by Givens rotation algorithms, and thus our general matrix decomposition processor then utilizes the CORDIC-based Givens rotations to support these matrix decompositions. The QRD algorithm can be seen in [12]. Here, we provide the algorithms for GMD and EVD.

2.1. GMD

Implementations of GMD/GTD have been studied in [7]-[9]. In [9], the SVD of the input matrix is first obtained, and then the triangularization procedure follows. Parallel operations can be executed in the triangularization procedure given the known diagonal elements for acceleration. In [7] and [8], the authors show that bidiagonalization procedure can be adopted for pre-processing to replace the iterative singular-value diagonalization procedure and significant computation complexity can be saved. However, unknown geometric mean results in a series of equalization steps. A recursive conversion is adopted in [7] and a divide-and-conquer method is used in [8]. In light of the above, our processor combines the advantages of the prior implementations and designs two-phase operation for GMD. In the first phase, the bidiagonalization pre-processing is employed. Meanwhile, the geometric mean is derived. In the second phase, geometric-mean (GM) equalization is performed. Thus, the iterative singular-value diagonalization and recursive equalization operations can be avoided. The steps are summarized in algorithm 1.

Assume that after bidiagonalization

$$\mathbf{H} = \mathbf{Q}_B\mathbf{B}\mathbf{P}_B^H, \tag{2}$$

where $\mathbf{Q}_B$ and $\mathbf{P}_B$ are unitary matrixes and $\mathbf{B}$ is a bidiagonal matrix. Because $\mathbf{H} = \mathbf{Q}_{GMD}\mathbf{R}_{GMD}\mathbf{P}_{GMD}^H$, we have

$$\begin{aligned} det(\mathbf{Q}_{GMD}\mathbf{R}_{GMD}\mathbf{P}_{GMD}^H) &= det(\mathbf{R}_{GMD}) = \prod_{i=0}^{N-1} r_{i,i} = r_{GMD}^N \\ &= det(\mathbf{Q}_B\mathbf{B}\mathbf{P}_B^H) = det(\mathbf{B}) = \prod_{i=0}^{N-1} b_{i,i}, \end{aligned} \tag{3}$$

where $r_{i,i}$ and $b_{i,i}$ are the ith diagonal element of $\mathbf{R}_{GMD}$ and $\mathbf{B}$, respectively. Thus, we can obtain the geometric mean as

$$r_{GMD} = \sqrt[N]{\prod_{i=0}^{N-1} b_{i,i}}. \tag{4}$$

Eq. (4) can be computed by CORDIC in hyperbolic mode recursively. Let $\phi_{N+i} = b_{i,i}$. Define

$$\phi_i = \sqrt{\phi_{2i}\phi_{2i+1}} = \frac{1}{2}\sqrt{(\phi_{2i} + \phi_{2i+1})^2 - (\phi_{2i} - \phi_{2i+1})^2} \tag{5}$$

After recursive calculations from ϕ_4~ϕ_7 for $\log_2(N)$ times, $\phi_1 = r_{GMD}$.

The 2×2 diagonal submatrix $\mathbf{B}_{2i-1:2i,2i-1:2i}$ is processed by 2×2 SVD first in the second phase. Then, GM equalization for the diagonal terms is performed. There are four cases depending on the values of the diagonal terms and r_{GMD}. The top-down operation performs GM equalization from upper-left corner sequentially while the bottom-up mode equalizes the diagonal term from the lower-right corner sequentially. The swap mode is applied to the two center rows as show in Fig. 1 until that the 2×2 corner submatrix contains two elements satisfying

$$\min(\mathbf{\Sigma}_{2i-1,2i-1}, \mathbf{\Sigma}_{2i,2i}) \le r_{GMD} \le \max(\mathbf{\Sigma}_{2i-1,2i-1}, \mathbf{\Sigma}_{2i,2i}) \tag{6}$$

Algorithm 1: GMD algorithm by Givens rotation

Given matrix $\mathbf{H} \in \mathbb{C}^{4\times4}$

// First phase

1. $[\mathbf{Q}_B, \mathbf{B}, \mathbf{P}_B^H]$=Bidiagonalization($\mathbf{H}$)
2. $[r_{GMD}]$=GeometricMean(diag($\mathbf{B}$))
3. $\mathbf{U} = \mathbf{Q}_B, \mathbf{V}^{\boldsymbol{H}} = \mathbf{P}_B^H$

// Second phase

4. for $i = 1{:}2$
5. $[\mathbf{G}_{l,i}, \mathbf{G}_{r,i}]$=SVD2×2($\mathbf{B}_{2i-1:2i,2i-1:2i}$)
6. $\mathbf{\Sigma}_{2i-1:2i,start:end} = \mathbf{G}_{l,i}\mathbf{B}_{2i-1:2i,start:end}$
7. $\mathbf{\Sigma}_{start:end,2i-1:2i} = \mathbf{B}_{start:end,2i-1:2i}\mathbf{G}_{r,i}$
8. $\mathbf{U}_{start:end,2i-1:2i} = \mathbf{U}_{start:end,2i-1:2i}\mathbf{G}_{l,i}^H$,
9. $\mathbf{V}_{2i-1:2i,start:end} = \mathbf{G}_{l,i}^H\mathbf{V}_{2i-1:2i,start:end}$

end

// GM equalization

10. **if** ($\min(\mathbf{\Sigma}_{1,1}, \mathbf{\Sigma}_{2,2}) \le r_{GMD} \le \max(\mathbf{\Sigma}_{1,1}, \mathbf{\Sigma}_{2,2})$) **&&** ($\max(\mathbf{\Sigma}_{3,3}, \mathbf{\Sigma}_{4,4}) \le r_{GMD}$)//Case 1
11. $[\mathbf{Q}_{GMD}, \mathbf{R}_{GMD}, \mathbf{P}_{GMD}^H]$=TopDown ($\mathbf{U}, \mathbf{V}^{\boldsymbol{H}}, \mathbf{\Sigma}$)
12. **else if** ($\min(\mathbf{\Sigma}_{3,3}, \mathbf{\Sigma}_{4,4}) \le r_{GMD} \le \max(\mathbf{\Sigma}_{3,3}, \mathbf{\Sigma}_{4,4})$) **&&** ($\min(\mathbf{\Sigma}_{1,1}, \mathbf{\Sigma}_{2,2}) \ge r_{GMD}$)//Case 2
13. $[\mathbf{Q}_{GMD}, \mathbf{R}_{GMD}, \mathbf{P}_{GMD}^H]$=BottomUp($\mathbf{U}, \mathbf{V}^{\boldsymbol{H}}, \mathbf{\Sigma}$)
14. **else if** ($\max(\mathbf{\Sigma}_{3,3}, \mathbf{\Sigma}_{4,4}) \le r_{GMD} \le \min(\mathbf{\Sigma}_{1,1}, \mathbf{\Sigma}_{2,2})$)//Case 3
15. $[\mathbf{Q}_{\mathbf{GMD}}, \mathbf{R}_{GMD}, \mathbf{P}_{GMD}^H]$=SwapThenParallel ($\mathbf{U}, \mathbf{V}^{\boldsymbol{H}}, \mathbf{\Sigma}$)
16. **else** //Case 4
17. $[\mathbf{Q}_{\mathbf{GMD}}, \mathbf{R}_{GMD}, \mathbf{P}_{GMD}^H]$=Parallel ($\mathbf{U}, \mathbf{V}^{\boldsymbol{H}}, \mathbf{\Sigma}$)

end

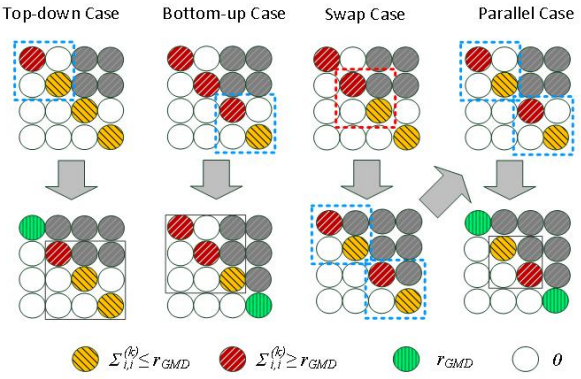

Fig. 1 Cases in GM equalization.

for $i = 1$ and $i = 2$. In that case, parallel operation begins for throughput acceleration. Fig. 1 depicts the four operations for different cases. Note that 2×2 equalization and 2×2 SVD [9] are carried out alternatively in all the cases to achieve GM equalization. However, they can be accomplished by CORDIC in circular vectoring, circular rotation and hyperbolic rotation modes.

2.2. EVD

EVD can also be accomplished by QR decomposition. The practical QR algorithm with shift has been shown to have good convergence [11]. The EVD procedure is given in Algorithm 2. Similarly, it contains two phase. In the first phase, Hessenberg reduction is performed. In the second phase, an iterative QR process is adopted. After Hessenberg reduction, input matrix $\mathbf{C}$ is transformed into upper Hessenberg matrix $\mathbf{S}^{(0)}(= \mathbf{U}_{EVD}^{(0)H}\mathbf{C}\mathbf{U}_{EVD}^{(0)})$, a tri-diagonal matrix for Hermitian symmetric matrix $\mathbf{C}$, where $\mathbf{U}_{EVD}^{(0)}$ is a unitary matrix.

Algorithm 2: EVD algorithm by Givens rotation

Given Hermitian symmetric matrix $\mathbf{C} \in \mathbb{C}^{4\times4}$

// First phase

1. $[\mathbf{U}_{EVD}^{(0)}, \mathbf{S}^{(0)}]$=HessenbergReduction($\mathbf{C}$)

// Second phase

i =0,

2. **while** (!converged)
3. $\mathbf{T}^{(i)} = \mathbf{S}^{(i)} - \mu_i \mathbf{I}$
4. $[\mathbf{Q}^{(i)}, \mathbf{R}^{(i)}] = QRD(\mathbf{T}^{(i)})$
5. $\mathbf{T}^{(i+1)} = \mathbf{R}^{(i)}\mathbf{Q}^{(i)}$
6. $\mathbf{S}^{(i+1)} = \mathbf{T}^{(i+1)} + \mu_i \mathbf{I}$
7. $\mathbf{U}_{EVD}^{(i+1)} = \mathbf{U}_{EVD}^{(i)}\mathbf{Q}^{(i)}$
8. $i = \mathrm{i} + 1$

end

In step 4 of algorithm 2, a series of left multiplications of Givens rotation matrixes that constitute $\mathbf{Q}^{(i)H}(= \mathbf{G}_1^{(i)}\mathbf{G}_2^{(i)} \dots \mathbf{G}_M^{(i)})$ is performed. On the other hand, in step 5 and 7, right multiplication of matrix $\mathbf{Q}^{(i)}$ is required. We can simply handle these two steps by using column-wise operations, namely

$$\mathbf{T}^{(i+1)} = \mathbf{R}^{(i)}\mathbf{Q}^{(i)} = \mathbf{R}^{(i)}\mathbf{G}_M^{(i)H} \dots \mathbf{G}_2^{(i)H}\mathbf{G}_1^{(i)H}. \quad (12)$$

Thus, the directions of micro-rotations during row-wise operations are reserved for column-wise operations.

3. PROPOSED MEMORY-BASED ARCHITECTURE FOR MATRIX-DECOMPOSITION PROCESSOR

The proposed architecture for the matrix decomposition processor which adopts memory-based design is depicted in Fig. 2. The main memory contains three memory modules for saving the decomposed channel matrix, left unitary matrix and right unitary matrix. There are 16 processing elements (PEs), eight for value computation and eight for basis computation. The left unitary matrix memory and right unitary matrix memory are accessed by PE9 to PE16 for basis computation while the decomposed channel matrix memory is accessed by PE1 to PE8 for value computation. Each PE is composed of one CORDIC which can be configured to circular vectoring, circular rotation, hyperbolic rotation, and idle mode. The circular vectoring mode is used to nullify the undesired component and the circular rotation mode is employed to duplicate the same operation to the associated elements. The hyperbolic mode is required to compute geometric mean in (5) and to generate rotation angles for 2×2 equalization.

Since a wide dynamic range is needed to cover various decomposition schemes, fixed-point representation for the datapath is not sufficient. Consequently, we adopt an external floating-point internal fixed-point strategy. The floating-point representation is used for the data stored in the memory. From the simulation results for the wordlength of mantissa in Fig. 3, GMD is more sensitive to finite precision than QRD and EVD. Consequently, 16 bits are reserved for the mantissa, 5 bits are allocated for the exponent with offset -27. Together with one sign bit, a total of 22 bits are used. Compared to [11] which uses 48-bit fixed-point representation, floating-point representation is efficient. The quantization results is less than 0.2 dB SNR degradation at bit-error rate about 10^{-3} for 16-QAM when the CORDIC contains 10 stages. On the other hand, the internal datapath of the CORDIC adopts fixed-point representation. Thus, a pre-alignment module and a post-alignment module are inserted at the input and output of the PE, respectively, as shown in Fig. 4.

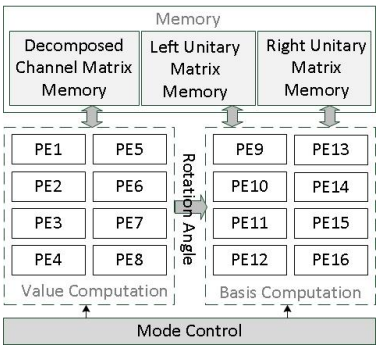

Fig. 2 Proposed memory-based architecture for the matrix decomposition processor.

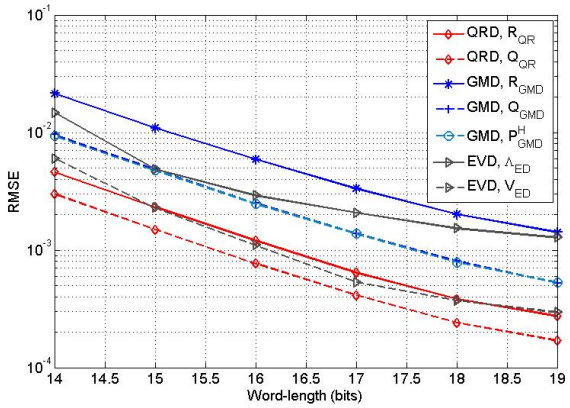

Fig. 3 Performance simulation for the wordlength of mantissa.

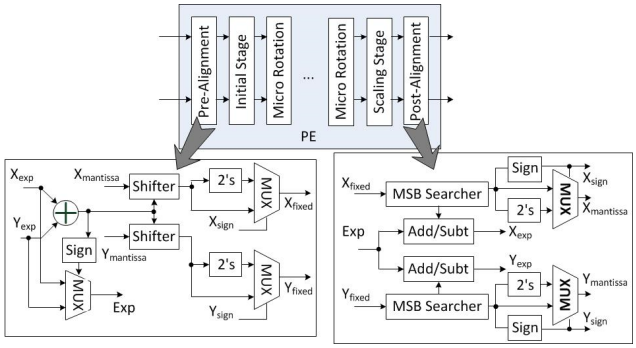

Fig. 4 Block diagram of the PE.

The control of the operation modes of all PEs are illustrated in Fig. 5. Take parts of the bidiagonalization process for GMD as an example. Two types of Givens rotations for dealing with complex-valued channel matrixes are involved [12]. The first type transforms the complex element to the real element, called C2R. The second type nullifies one real element by another real element, called R2Z. The styles of the circles in the red rectangle indicate the results after being processed by PEs. The PE configured in the vectoring mode delivers the rotation directions to the PEs configured in the rotation mode. The scheduling considers high hardware utilization and control regularity. The right and left unitary matrix memories are first initialized with the identity matrix. Then, they are accessed by PE9 to PE16 with mirrored operations to generate left and/or right basis matrixes. Thus, the row operation and column operation of PE9~PE16 are always opposite to PE1~PE8.

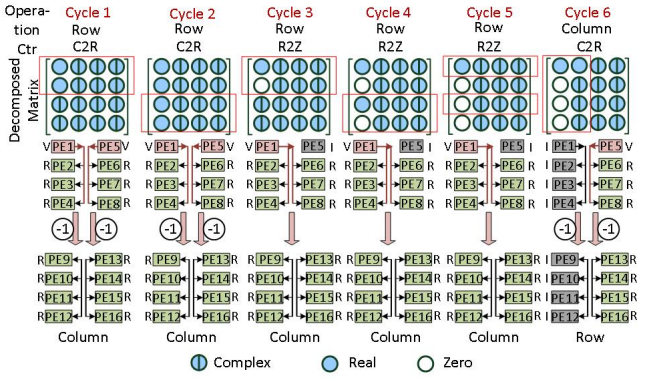

Fig. 5 Control of the operation modes of all PEs.

4. IMPLEMENTATION RESULTS AND COMPARISON

The hardware of the proposed matrix-decomposition processor is implemented. It takes 12 clock cycles to complete QR decomposition and to produce matrix $\mathbf{R}_{QR}$ and $\mathbf{Q}_{QR}$ simultaneously, saved in decomposed-channel-matrix memory and left-unitary-matrix memory respectively. For 4×4 EVD, it takes 14 clock cycles to perform Hessenberg reduction. Although the conventional QR and RQ algorithm is adopted, the shift and deflation mechanisms are incorporated in the hardware. Both accelerates the convergence by rapidly shrinking matrix dimension. Hence, the processor uses 6, 4, 2 clock cycles to complete one QR and RQ iteration for 4×4, 3×3, 2×2 matrix, respectively. The average processing cycles for 4×4 EVD with shift and deflation are 120.8 assuming a maximum of 30 iterations. For 4×4 GMD, the bidiagonalization in the first phase needs 19 clock cycles and 6 clock cycles are reserved for geometric-mean calculation and 2×2 SVD. The occurrence probability of the GM equalization in four cases for randomly-generated channel matrixes is 5.95%, 18.57%, 4.51%, and 70.97%, respectively. It is clear that the probability of the parallel mode is higher than the others. Thus, the complexity reduction and throughput acceleration are both achieved. The time complexity and computation complexity in each process for various matrix decompositions are listed in Table 1. The first term in the rightmost column refers to the CORDIC operations for decomposing the channel matrix. The second (third) term associates with the computation complexity for the left (right)-unitary matrix.

Table 2 shows the complexity comparison, in terms of CORDIC operations, of different GMD algorithms for deriving $\mathbf{Q}_{GMD}, \mathbf{R}_{GMD}, \mathbf{P}_{GMD}^{H}$. With the proposed GM calculation in the first phase and GM equalization in the second phase, we can achieve 13.2% complexity reduction assuming that the complexity of the square-root operation is similar to one CORDIC operation. Note that a generalized matrix-decomposition processor prefers algorithms that can be implemented by homogeneous PEs. Table 3 lists some implementation results of this work and conventional works that can provide the complete basis matrixes and value matrix. The efficiency is obtained by dividing the decomposition rate to the number of CORDIC modules. The EVD in [11] process real input matrixes. It takes 15 clock cycles for one QR and RQ step to deal with a 4×4 matrix and 30 steps are configured [11]. If the same systolic array architecture is used to process complex input matrixes, 22 CORDIC modules will be required, just like [12]. From the table, we can see that our generalized matrix-decomposition processor is competitive to some dedicated hardware and has high hardware efficiency.

Table 1 Clock cycles and CORDIC operations in each process

	Process	Clock Cycles	No. of CORDIC operations
QRD		12	64 + 88
EVD	Hessenberg Reduction	14	78 + 48
	4×4 QR & RQ	6	16 + 24
	3×3 QR & RQ	4	10 + 16
	2×2 QR & RQ	2	4 + 8
GMD	Bidiagonalization	19	103 + 88 +48
	GM+2×2SVD	6	18 + 16 + 16
	GM Equalization	15.54	38 + 35 + 35

Table 2 Complexity comparison of GMD algorithms.

No. of CORDIC operations	[8]	This work
Bidiagonalization (Value Compuation)	103	103
Remaining GMD (Value Computation)	66	18 (GM + 2×2 SVD)
		38 (GM Equalization)
Square Root	8	-
Bidiagonalization (Basis Computation)	136	136
Remaining GMD (Basis Computation)	144	102
Total	457(100%)	397(86.8%)

Table 3 Comparison of implementations.

	This Work	[12]	This Work	[11]	This Work	[9]
Function	4 × 4 QRD	4 × 4 QRD	4 × 4 EVD	4 × 4 EVD (Real)	4 × 4 GMD	1 × 1 ~ 8 × 8 GMD
Output Matrix	$\mathbf{Q}_{QR}$ $\mathbf{R}_{QR}$	$\mathbf{Q}_{QR}^{H}$ $\mathbf{R}_{QR}$	$\mathbf{U}_{ED}$ $\mathbf{\Lambda}_{ED}$	$\mathbf{U}_{ED}$ $\mathbf{\Lambda}_{ED}$	$\mathbf{Q}_{GMD}$ $\mathbf{R}_{GMD}$ $\mathbf{P}_{GMD}^{H}$	$\mathbf{Q}_{GMD}$ $\mathbf{R}_{GMD}$ $\mathbf{P}_{GMD}^{H}$
No. of CORDIC modules	16	22	16	6	16	64
Processing Cycles(1)	12	8	120.8	450	40.54	578.2
Process(nm)	40	180	40	-	40	90
Frequency (MHz)	120	100	120	232	120	112.4
Rate(M/s)(1)	10	12.5	0.99	0.516	2.96	0.194
Efficiency(1)	0.625	0.568	0.062	0.086	0.185	0.003

(1): Consider the results of processing 4x4 complex matrixes except [11].

5. CONCLUSION

A generalized matrix-decomposition processor is presented to support QRD, GMD and EVD. The improved GMD algorithm saves 13.2% arithmetic complexity and can be accomplished by homogeneous PEs. EVD adopts the shift and deflation strategies to accelerate convergence. The memory-based architecture offers the configurability to support these decompositions. External floating-point and internal fixed-point representation is used for the datapaths to cover a wide dynamic range of different decompositions. It takes 12, 120.8, 40.54 averaged clock cycles to process 4×4 complex QRD, EVD and GMD. From the comparison, the implementation achieves good hardware efficiency.

REFERENCES

[1] X. Hou, Z. Zhang. H. Kayama, “QRD-based MU-MIMO transmission scheme towards evolved LTE-TDD system,” IEEE Globecom Workshops 2010, pp. 866-870.

[2] E. Perahia and R. Stacey, *Next Generation Wireless LANs 802.lln and 802.llac*. 2nd Ed., Cambridge University Press, 2013.

[3] Y. Jiang, J. Li, W. W. Hager, "Joint transceiver design for MIMO communications using geometric mean decomposition", IEEE Trans. Signal Process. , vol.53, no.10, pp.3791–3803, 2005.

[4] K. Gomadam, V. R. Cadambe and S. A. Jafar, “A distributed numerical approach to interference alignment and applications to wireless interference networks,” IEEE Transactions on Information Theory, vol. 57, no. 6, pp. 3309-3322, June 2011.

[5] C. Wilson and V. Veeravalli, "A convergent version of the Max SINR algorithm for the MIMO interference channel," IEEE Transactions on Wireless Communications, vol. 12, no. 6, pp. 2952-2961, June 2013.

[6] C. M. Chen, C. H. Lin and P. Y. Tsai, “Multi-mode Sorted QR Decomposition for 4x4 and 8x8 Single-user/Multi-user MIMO Precoding,” International Symposium on Circuits and Systems (ISCAS), May 2015, pp. 2980-2983.

[7] W. D. Chen and Y. T. Huang, “A Constant Throughput Geometric Mean Decomposition Scheme Design for Wireless MIMO Precoding,” IEEE Transactions on Vehicular Technology, vol. 62, iss. 5, pp. 2080-2090, May 2013.

[8] Y. T. Huang, W. D. Chen, and C. R. Hong, “A Low Complexity Geometric Mean Decomposition Computing Scheme and Its High Throughput VLSI Implementation,” IEEE Trans. Circuits and System I: Regular Papers, vol: 61, Issue: 4, pp.1170-1182, Apr. 2014

[9] C. H. Yang, C. W. Chou, C. S. Hsu, and C. E. Chen, “A systolic array based GTD processor with a parallel algorithm,” IEEE Transactions on Circuits and Systems I: Regular papers, vol. 62, Iss. 4, pp. 1099-1108, Apr. 2015.

[10] Y. T. Hwang, K. T. Chen, C. K. Wu, “A high-throughput unified SVD/QRD precoder design for MIMO OFDM systems,” IEEE International Conference on Digital Signal Processing (DSP), 2015, pp. 1148-1151.

[11] J. Guerrero-Ramirez, J. Velasco-Medina and J. Arce-Clavijo, “Hardware design of an eigensolver based on the QR method,” *IEEE Fourth Latin American Symposium on Circuits and Systems (LASCAS)*, 2013. Pp. 1-4.

[12] Zheng-Yu Huang and Pei-Yun Tsai, “Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems,” IEEE Transactions on Circuits and Systems I: Regular paper, vol. 58, pp. 2531-2542, Oct. 2011.