

QRD Architecture Using the Modified ILMGS Algorithm for MIMO Systems

Cang Liu¹(✉), Chuan Tang¹, Zuocheng Xing¹, Luechao Yuan¹, Yu Wang¹,
Lirui Chen¹, Yang Zhang¹, Suncheng Xiang¹, Wangfeng Zhao¹, Xing Hu¹,
and Jinsong Xu²

¹ National Laboratory for Parallel and Distributed Processing,
National University of Defense Technology, Changsha, China
{liucang, tc8831, zcxing, yuan.luechao, wangyu16, chenlirui14,
zhangyang, xiangsuncheng14, zhaowangfeng}@nudt.edu.cn, 348883881@qq.com,
351749562@qq.com

² School of Cryptography Engineering,
Information Engineering University, Zhengzhou, China
pine_xu@sohu.com

Abstract. QR decomposition (QRD) is one of the performance bottlenecks of transceiver processor in the multiuser multiple-input-multiple-output (MU-MIMO) systems. This paper proposes a QRD algorithm based on the existing modified Gram-Schmidt (MGS) algorithm and iteration look-ahead MGS (ILMGS) algorithm, which is named modified ILMGS (MILMGS) algorithm. A corresponding hardware architecture based on the proposed MILMGS algorithm is designed in 0.13 μm CMOS technique to decompose the 4×4 real matrix. The implementation results show that the gate count of the designed hardware architecture is 250.2K, the throughput and the critical path are 95.2M/s and 3.5 ns respectively.

Keywords: QRD · MU-MIMO · MGS · ILMGS · MILMGS

1 Introduction

The multiuser multiple-input-multiple-output (MU-MIMO) technique has attracted considerable interests, because of the continuously increasing demand for high data rates and high spectrum efficiency. And the MU-MIMO technique has been adopted in several wireless communication standards, such as IEEE 802.11n [1], IEEE 802.11ac [2] and IEEE 802.16e [3]. In order to simultaneously serve multiple users in the same frequency band, various precoding algorithms are proposed. Most of this algorithms are complicated and difficult to implement in MIMO systems [4, 5]. Consequently, exploiting the hardware architecture for this algorithms has important significance.

For presubtraction of the multiuser interference (MUI), Costa [6] proposed the dirty paper coding (DPC) algorithm in his representative literature “writing

on dirty paper”. The DPC algorithm is the optimal precoding algorithm, however, it is very difficult to implement in practical communication systems, due to its unreachable computational complexity [4, 5]. Some practical DPC algorithms have been proposed to reduce the computational complexity. Tomlinson-Harashima precoding (THP) is one of the most popular practical algorithms, and has gained great attentions in some communication research communities [7–9]. And QR decomposition (QRD) is the primary part in most THP algorithms.

The existing QRD hardware architectures of MIMO systems mainly base on the Givens Rotation (GR) [10–12] algorithm and the modified Gram-Schmidt (MGS) [13–17] algorithm. Due to using the coordinate rotation digital computer (CORDIC) algorithm in the GR algorithm, it can significantly reduce the corresponding hardware overhead. However, because the GR algorithm annihilates the elements one by one, it leads to longer processing latency. Compared to the GR algorithm, the MGS algorithm annihilates the elements one column by one column. Therefore, it significantly reduces the processing latency. To further reduce the processing latency, the iteration look-ahead MGS (ILMGS) algorithm is propose in [18]. Due to involving several high computation complexity operations (square root and division) in ILMGS algorithm, the corresponding hardware architecture costs large hardware overhead.

In this paper, a modified ILMGS (MILMGS) algorithm is proposed for the QRD of MIMO systems, and a new triangular systolic array (TSA) architecture is designed based on the proposed MILMGS algorithm. The Newton-Raphson (NR) algorithm is used in the proposed MILMGS algorithm to reduce the hardware overhead. Moreover, the designed hardware architecture is implemented to decompose a 4×4 real matrix, the implementation results show that the throughput performance and the latency performance of the designed TSA hardware architecture are 95.2 MHz and 53 clock cycles respectively, the gate count and the critical path of the designed TSA architecture are 250.2 K and 3.5 ns respectively.

The rest of the paper is organised as the follows. In Sect. 2, we illustrate the traditional QRD algorithms and propose the MILMGS algorithm. In Sect. 3, the designed hardware architecture for QRD with MILMGS algorithm is presented. In Sect. 4, we provide the implementation results and some comparisons with the existing works. Finally, Sect. 5 draws the conclusions.

2 QR Decomposition Algorithm

The existing QRD hardware architectures for MIMO systems mainly base on the GR algorithm and the MGS algorithm respectively. The proposed QRD algorithm (MILMGS) is based on the modified MGS algorithm (ILMGS). Hence, we only focus on the MGS algorithm and the ILMGS algorithm in the following, the proposed MILMGS algorithm is also provided in this section.

A $n \times n$ real matrix \mathbf{A} can be decomposed to an upper triangular matrix \mathbf{R} and an unitary matrix \mathbf{Q} by QRD algorithm. In order to simplify the description, the size of real matrix \mathbf{A} is set to 4×4 , \mathbf{x}_i and $a_{i,j}$ represent the elements in i th column and the (i, j) element of matrix \mathbf{X} respectively in this paper.

2.1 MGS Algorithm

The MGS algorithm performs QRD one column by one column. Firstly, The element $r_{1,1}$ of upper triangular matrix and the first column elements of unitary matrix \mathbf{Q} can be computed as follows

$$r_{1,1} = \|\mathbf{a}_1\|_2 = \sqrt{(a_{1,1})^2 + (a_{2,1})^2 + (a_{3,1})^2 + (a_{4,1})^2} \quad (1)$$

$$\begin{aligned} \mathbf{q}_1 &= \frac{\mathbf{a}_1}{r_{1,1}} \\ q_{1,1} &= \frac{a_{1,1}}{r_{1,1}}, q_{2,1} = \frac{a_{2,1}}{r_{1,1}}, q_{3,1} = \frac{a_{3,1}}{r_{1,1}}, q_{4,1} = \frac{a_{4,1}}{r_{1,1}} \end{aligned} \quad (2)$$

The elements $r_{1,j}, 1 < j \leq 4$ are obtained by elements \mathbf{q}_1 and the j th column vector of matrix \mathbf{A} as Eq. 3

$$\begin{aligned} r_{1,j} &= \mathbf{q}_1^T \cdot \mathbf{a}_j \\ &= q_{1,1}a_{1,j} + q_{2,1}a_{2,j} + q_{3,1}a_{3,j} + q_{4,1}a_{4,j}, \\ &\quad 2 \leq j \leq 4; j \in N \end{aligned} \quad (3)$$

To obtain the other elements of upper triangular matrix \mathbf{R} and unitary matrix \mathbf{Q} , the matrix \mathbf{A} should be converted to the next iterative matrix \mathbf{A}^1 as Eq. 4

$$\begin{aligned} \mathbf{a}_1^1 &= 0 \\ \mathbf{a}_2^1 &= \mathbf{a}_2 - r_{1,2}\mathbf{q}_1 \\ \mathbf{a}_3^1 &= \mathbf{a}_3 - r_{1,3}\mathbf{q}_1 \\ \mathbf{a}_4^1 &= \mathbf{a}_4 - r_{1,4}\mathbf{q}_1 \end{aligned} \quad (4)$$

Secondly, the element $r_{2,2}$ of upper triangular matrix \mathbf{R} and the second column vector of unitary matrix \mathbf{Q} can be obtained as follows

$$r_{2,2} = \|\mathbf{a}_2^1\|_2 = \sqrt{(a_{1,2}^1)^2 + (a_{2,2}^1)^2 + (a_{3,2}^1)^2 + (a_{4,2}^1)^2} \quad (5)$$

$$\begin{aligned} \mathbf{q}_2 &= \frac{\mathbf{a}_2^1}{r_{2,2}} \\ q_{1,2} &= \frac{a_{1,2}^1}{r_{2,2}}, q_{2,2} = \frac{a_{2,2}^1}{r_{2,2}}, q_{3,2} = \frac{a_{3,2}^1}{r_{2,2}}, q_{4,2} = \frac{a_{4,2}^1}{r_{2,2}} \end{aligned} \quad (6)$$

The second row vector $r_{2,j}, 2 < j \leq 4$ of upper triangular matrix \mathbf{R} can be computed by Eq. 7

$$\begin{aligned} r_{2,j} &= \mathbf{q}_2^T \cdot \mathbf{a}_j^1 \\ &= q_{1,2}a_{1,j}^1 + q_{2,2}a_{2,j}^1 + q_{3,2}a_{3,j}^1 + q_{4,2}a_{4,j}^1, \\ &\quad 3 \leq j \leq 4; j \in N \end{aligned} \quad (7)$$

To further calculate the third row vector of upper triangular matrix \mathbf{R} and the third column vector of unitary matrix \mathbf{Q} , the next iterative matrix \mathbf{A}^2 can be calculated as Eq. 8

$$\begin{aligned} \mathbf{a}_1^2 &= 0 \\ \mathbf{a}_2^2 &= 0 \\ \mathbf{a}_3^2 &= \mathbf{a}_3^1 - r_{2,3}\mathbf{q}_2 \\ \mathbf{a}_4^2 &= \mathbf{a}_4^1 - r_{2,4}\mathbf{q}_2 \end{aligned} \quad (8)$$

Thirdly, the elements $r_{3,3}$, \mathbf{q}_3 and $r_{3,4}$ can be computed as follows

$$r_{33} = \|\mathbf{a}_3^2\|_2 = \sqrt{(a_{1,3}^2)^2 + (a_{2,3}^2)^2 + (a_{3,3}^2)^2 + (a_{4,3}^2)^2} \quad (9)$$

$$\mathbf{q}_3 = \frac{\mathbf{a}_3^2}{r_{3,3}} \quad (10)$$

$$q_{1,3} = \frac{a_{1,3}^2}{r_{3,3}}, q_{2,3} = \frac{a_{2,3}^2}{r_{3,3}}, q_{33} = \frac{a_{3,3}^2}{r_{3,3}}, q_{4,3} = \frac{a_{4,3}^2}{r_{3,3}}$$

$$r_{3,4} = \mathbf{q}_3^T \mathbf{a}_4^2 \quad (11)$$

$$= q_{1,3}a_{1,4}^2 + q_{2,3}a_{2,4}^2 + q_{3,3}a_{3,4}^2 + q_{4,3}a_{4,4}^2$$

Similarly, the next iterative matrix \mathbf{A}^3 can be obtained as Eq. 12 to calculate the elements $r_{4,4}$ and \mathbf{q}_4 .

$$\begin{aligned} \mathbf{a}_1^3 &= 0 \\ \mathbf{a}_2^3 &= 0 \\ \mathbf{a}_3^3 &= 0 \\ \mathbf{a}_4^3 &= \mathbf{a}_4^2 - r_{3,4}\mathbf{q}_3 \end{aligned} \quad (12)$$

Finally, the fourth column vector of unitary matrix \mathbf{Q} and the element $r_{4,4}$ of upper triangular matrix \mathbf{R} can be obtained by Eqs. 13 and 14. The results (upper triangular matrix \mathbf{R} and unitary matrix \mathbf{Q}) of MGS algorithm are obtained by the aforementioned steps.

$$r_{4,4} = \|\mathbf{a}_4^3\|_2 = \sqrt{(a_{1,4}^3)^2 + (a_{2,4}^3)^2 + (a_{3,4}^3)^2 + (a_{4,4}^3)^2} \quad (13)$$

$$\mathbf{q}_4 = \frac{\mathbf{a}_4^3}{r_{4,4}} \quad (14)$$

$$q_{1,4} = \frac{a_{1,4}^3}{r_{4,4}}, q_{2,4} = \frac{a_{2,4}^3}{r_{4,4}}, q_{3,4} = \frac{a_{3,4}^3}{r_{4,4}}, q_{4,4} = \frac{a_{4,4}^3}{r_{4,4}}$$

2.2 ILMGS Algorithm

To reduce the processing latency of QRD with MGS algorithm, the ILMGS algorithm is proposed in [18] for the MIMO systems. The QRD with ILMGS algorithm is presented in the following.

Firstly, the first column vector \mathbf{q}_1 of unitary matrix \mathbf{Q} and the first row vector $r_{1,j}, 1 \leq j \leq 4$ of upper triangular matrix \mathbf{R} can be obtained as the MGS algorithm.

$$r_{1,1} = \|\mathbf{a}_1\|_2 = \sqrt{(a_{1,1})^2 + (a_{2,1})^2 + (a_{3,1})^2 + (a_{4,1})^2} \quad (15)$$

$$\mathbf{q}_1 = \frac{\mathbf{a}_1}{r_{1,1}} \quad (16)$$

$$q_{1,1} = \frac{a_{1,1}}{r_{1,1}}, q_{2,1} = \frac{a_{2,1}}{r_{1,1}}, q_{3,1} = \frac{a_{3,1}}{r_{1,1}}, q_{4,1} = \frac{a_{4,1}}{r_{1,1}}$$

$$r_{1,j} = \mathbf{q}_1^T \cdot \mathbf{a}_j \quad (17)$$

$$= q_{1,1}a_{1,j} + q_{2,1}a_{2,j} + q_{3,1}a_{3,j} + q_{4,1}a_{4,j},$$

$$2 \leq j \leq 4; j \in N$$

Then, the matrix \mathbf{A} need to be converted to the next iterative matrix \mathbf{A}^1 as Eq. 18. The converted method is the main difference between the ILMGS algorithm and the MGS algorithm.

$$\begin{aligned} \mathbf{a}_1^1 &= 0 \\ \mathbf{a}_2^1 &= \mathbf{a}_2 - r_{1,2}\mathbf{q}_1 = \mathbf{a}_2 - \mathbf{q}_1^T \mathbf{a}_2 \mathbf{q}_1 = \mathbf{a}_2 - \frac{\mathbf{a}_1^T \mathbf{a}_2 \mathbf{a}_1}{r_{1,1}^2} \\ \mathbf{a}_3^1 &= \mathbf{a}_3 - \frac{\mathbf{a}_1^T \mathbf{a}_3 \mathbf{a}_1}{r_{1,1}^2} \\ \mathbf{a}_4^1 &= \mathbf{a}_3 - \frac{\mathbf{a}_1^T \mathbf{a}_4 \mathbf{a}_1}{r_{1,1}^2} \end{aligned} \quad (18)$$

Equation 18 shows that the next iterative matrix \mathbf{A}^1 can be computed after getting the value of $r_{1,1}^2$. Moreover, the value of $r_{1,1}^2$ is computed with $\mathbf{a}_1^T \cdot \mathbf{a}_1$, and the time overhead of $\mathbf{a}_1^T \cdot \mathbf{a}_1$ is equal to the time overhead of $\mathbf{a}_1^T \cdot \mathbf{a}_2$. Hence, the value of \mathbf{A}^1 can be calculated simultaneously with the calculation of \mathbf{q}_1 and $r_{1,j}, 1 \leq j \leq 4$, and not need to wait the results of \mathbf{q}_1 and $r_{1,j}, 1 \leq j \leq 4$.

Secondly, to compute the second column vector of \mathbf{Q} and the second row vector of \mathbf{R} , we need to repeat the similar steps as shown in the following

$$r_{2,2} = \|\mathbf{a}_2^1\|_2 = \sqrt{(a_{1,2}^1)^2 + (a_{2,2}^1)^2 + (a_{3,2}^1)^2 + (a_{4,2}^1)^2} \quad (19)$$

$$\begin{aligned} \mathbf{q}_2 &= \frac{\mathbf{a}_2^1}{r_{2,2}} \\ q_{1,2} &= \frac{a_{1,2}^1}{r_{2,2}}, q_{2,2} = \frac{a_{2,2}^1}{r_{2,2}}, q_{3,2} = \frac{a_{3,2}^1}{r_{2,2}}, q_{4,2} = \frac{a_{4,2}^1}{r_{2,2}} \end{aligned} \quad (20)$$

$$\begin{aligned} r_{2,j} &= \mathbf{q}_2^T \cdot \mathbf{a}_j^1 \\ &= q_{1,2}a_{1,j}^1 + q_{2,2}a_{2,j}^1 + q_{3,2}a_{3,j}^1 + q_{4,2}a_{4,j}^1, \\ &\quad 3 \leq j \leq 4; j \in N \end{aligned} \quad (21)$$

Then, the value of next iterative matrix \mathbf{A}^2 is similar to Eq. 18.

$$\begin{aligned} \mathbf{a}_1^2 &= 0 \\ \mathbf{a}_2^2 &= 0 \\ \mathbf{a}_3^2 &= \mathbf{a}_3 - r_{2,3}\mathbf{q}_2 = \mathbf{a}_3 - \mathbf{q}_2^T \mathbf{a}_3 \mathbf{q}_2 = \mathbf{a}_3 - \frac{\mathbf{a}_2^{1T} \mathbf{a}_3 \mathbf{a}_2^1}{r_{2,2}^2} \\ \mathbf{a}_4^2 &= \mathbf{a}_4 - \frac{\mathbf{a}_2^{1T} \mathbf{a}_4 \mathbf{a}_2^1}{r_{2,2}^2} \end{aligned} \quad (22)$$

Thirdly, the elements $r_{3,j}, 3 \leq j \leq 4$ of upper triangular matrix \mathbf{R} and the third column vector \mathbf{q}_3 of unitary matrix \mathbf{Q} can be computed as follows

$$r_{33} = \|\mathbf{a}_3^2\|_2 = \sqrt{(a_{1,3}^2)^2 + (a_{2,3}^2)^2 + (a_{3,3}^2)^2 + (a_{4,3}^2)^2} \quad (23)$$

$$\begin{aligned} \mathbf{q}_3 &= \frac{\mathbf{a}_3^2}{r_{3,3}} \\ q_{1,3} &= \frac{a_{1,3}^2}{r_{3,3}}, q_{2,3} = \frac{a_{2,3}^2}{r_{3,3}}, q_{3,3} = \frac{a_{3,3}^2}{r_{3,3}}, q_{4,3} = \frac{a_{4,3}^2}{r_{3,3}} \end{aligned} \quad (24)$$

$$\begin{aligned} r_{3,4} &= \mathbf{q}_3^T \mathbf{a}_4^2 \\ &= q_{1,3}a_{1,4}^2 + q_{2,3}a_{2,4}^2 + q_{3,3}a_{3,4}^2 + q_{4,3}a_{4,4}^2 \end{aligned} \quad (25)$$

The next iterative matrix \mathbf{A}^3 is calculated as Eq. 26

$$\begin{aligned} \mathbf{a}_1^3 &= 0 \\ \mathbf{a}_2^3 &= 0 \\ \mathbf{a}_3^3 &= 0 \\ \mathbf{a}_4^3 &= \mathbf{a}_4^2 - r_{3,4}\mathbf{q}_3 = \mathbf{a}_4^2 - \mathbf{q}_3^T \mathbf{a}_4^2 \mathbf{q}_3 = \mathbf{a}_4^2 - \frac{\mathbf{a}_3^{2T} \mathbf{a}_4^2 \mathbf{a}_3^2}{r_{3,3}^2} \end{aligned} \tag{26}$$

Finally, repeating the similar operations in the matrix \mathbf{A}^3 , the value of $r_{4,4}$ and column vector \mathbf{q}_4 can be obtained as follows

$$r_{4,4} = \|\mathbf{a}_4^3\|_2 = \sqrt{(a_{1,4}^3)^2 + (a_{2,4}^3)^2 + (a_{3,4}^3)^2 + (a_{4,4}^3)^2} \tag{27}$$

$$\begin{aligned} \mathbf{q}_4 &= \frac{\mathbf{a}_4^3}{r_{4,4}} \\ q_{1,4} &= \frac{a_{1,4}^3}{r_{4,4}}, q_{2,4} = \frac{a_{2,4}^3}{r_{4,4}}, q_{3,4} = \frac{a_{3,4}^3}{r_{4,4}}, q_{4,4} = \frac{a_{4,4}^3}{r_{4,4}} \end{aligned} \tag{28}$$

2.3 MILMGS Algorithm

The ILMGS algorithm significantly reduces the processing latency. However, compared to the QRD hardware architecture based on the other algorithm, the main drawback of the MGS algorithm and the ILMGS algorithm is that they need more hardware overhead. To reduce the hardware overhead of QRD, this paper proposes the MILMGS algorithm based on the existing ILMGS algorithm. We will present the proposed MILMGS algorithm in the following.

The aforementioned description about the ILMGS algorithm shows than the value of $r_{i,i}, 1 \leq i \leq 4$ and each column vector of unitary matrix \mathbf{Q} can be rewritten as $x \cdot \frac{1}{r_{i,i}}$. And the division in the calculation of the next iterative matrix \mathbf{A}^m can also be rewritten as $x \cdot \frac{1}{r_{i,i}} \cdot \frac{1}{r_{i,i}}$. Due to the value of $\frac{1}{r_{i,i}}$ is equal to the value of $\frac{1}{\sqrt{r_{i,i}^2}}$, the value of $r_{i,i}^2$ is firstly calculated as Eq. 29

$$r_{1,1}^2 = (a_{1,1})^2 + (a_{2,1})^2 + (a_{3,1})^2 + (a_{4,1})^2 \tag{29}$$

Then, the reciprocal square root (RSR) operation is used to obtain the value of $\frac{1}{\sqrt{r_{i,i}^2}}$ as follows

$$RSR(r_{1,1}^2) = \frac{1}{\sqrt{r_{1,1}^2}} \tag{30}$$

Therefore, the value of $r_{1,1}$, the first column vector \mathbf{q}_1 of unitary matrix \mathbf{Q} and the next iterative matrix \mathbf{A}^1 can be represented as follows

$$\begin{aligned} q_{1,1} &= a_{1,1} \cdot RSR(r_{1,1}^2), \\ q_{2,1} &= a_{2,1} \cdot RSR(r_{1,1}^2), \\ q_{3,1} &= a_{3,1} \cdot RSR(r_{1,1}^2), \\ q_{4,1} &= a_{4,1} \cdot RSR(r_{1,1}^2), \\ r_{1,1} &= r_{1,1}^2 \cdot RSR(r_{1,1}^2) \end{aligned} \tag{31}$$

$$\begin{aligned}
\mathbf{a}_1^1 &= 0 \\
\mathbf{a}_2^1 &= \mathbf{a}_2 - \mathbf{a}_1^T \mathbf{a}_2 \mathbf{a}_1 \cdot RSR(r_{1,1}^2) \cdot RSR(r_{1,1}^2) \\
\mathbf{a}_3^1 &= \mathbf{a}_3 - \mathbf{a}_1^T \mathbf{a}_3 \mathbf{a}_1 \cdot RSR(r_{1,1}^2) \cdot RSR(r_{1,1}^2) \\
\mathbf{a}_4^1 &= \mathbf{a}_4 - \mathbf{a}_1^T \mathbf{a}_4 \mathbf{a}_1 \cdot RSR(r_{1,1}^2) \cdot RSR(r_{1,1}^2)
\end{aligned} \tag{32}$$

The values of $r_{1,j}$, $2 \leq j \leq 4$ can be obtained as the ILMGS method, and is presented in Eq. 17.

Similarly, to obtain the second row vector of upper triangular matrix \mathbf{R} and the second column vector of unitary matrix \mathbf{Q} , the value of $r_{2,2}^2$ and $RSR(r_{2,2}^2)$ need to be obtained firstly as follows

$$r_{2,2}^2 = (a_{1,2}^1)^2 + (a_{2,2}^1)^2 + (a_{3,2}^1)^2 + (a_{4,2}^1)^2 \tag{33}$$

$$RSR(r_{2,2}^2) = \frac{1}{\sqrt{r_{2,2}^2}} \tag{34}$$

The second column vector \mathbf{q}_2 of unitary matrix \mathbf{Q} , the element $r_{2,2}$ and the next iterative matrix \mathbf{A}^2 can be calculated as follows

$$\begin{aligned}
q_{1,2} &= a_{1,2}^1 \cdot RSR(r_{2,2}^2), \\
q_{2,2} &= a_{2,2}^1 \cdot RSR(r_{2,2}^2), \\
q_{3,2} &= a_{3,2}^1 \cdot RSR(r_{2,2}^2), \\
q_{4,2} &= a_{4,2}^1 \cdot RSR(r_{2,2}^2), \\
r_{2,2} &= r_{2,2}^2 \cdot RSR(r_{2,2}^2)
\end{aligned} \tag{35}$$

$$\begin{aligned}
\mathbf{a}_1^2 &= 0 \\
\mathbf{a}_2^2 &= 0 \\
\mathbf{a}_3^2 &= \mathbf{a}_3^1 - \mathbf{a}_2^{1T} \mathbf{a}_3^1 \mathbf{a}_2^1 \cdot RSR(r_{2,2}^2) \cdot RSR(r_{2,2}^2) \\
\mathbf{a}_4^2 &= \mathbf{a}_4^1 - \mathbf{a}_2^{1T} \mathbf{a}_4^1 \mathbf{a}_2^1 \cdot RSR(r_{2,2}^2) \cdot RSR(r_{2,2}^2)
\end{aligned} \tag{36}$$

The values of $r_{2,j}$, $3 \leq j \leq 4$ can be obtained by Eq. 21.

Next, the values of $r_{3,3}^2$ and $RSR(r_{3,3}^2)$ are computed as Eqs. 37 and 38.

$$r_{3,3}^2 = (a_{1,3}^2)^2 + (a_{2,3}^2)^2 + (a_{3,3}^2)^2 + (a_{4,3}^2)^2 \tag{37}$$

$$RSR(r_{3,3}^2) = \frac{1}{\sqrt{r_{3,3}^2}} \tag{38}$$

The column vector \mathbf{q}_3 , the elements $r_{3,3}$ and the next iterative matrix \mathbf{A}^3 can be computed as follows

$$\begin{aligned}
q_{1,3} &= a_{1,3}^2 \cdot RSR(r_{3,3}^2), \\
q_{2,3} &= a_{2,3}^2 \cdot RSR(r_{3,3}^2), \\
q_{3,3} &= a_{3,3}^2 \cdot RSR(r_{3,3}^2), \\
q_{4,3} &= a_{4,3}^2 \cdot RSR(r_{3,3}^2), \\
r_{3,3} &= r_{3,3}^2 \cdot RSR(r_{3,3}^2)
\end{aligned} \tag{39}$$

$$\begin{aligned}
\mathbf{a}_1^3 &= 0 \\
\mathbf{a}_2^3 &= 0 \\
\mathbf{a}_3^3 &= 0 \\
\mathbf{a}_4^3 &= \mathbf{a}_4^2 - \mathbf{a}_3^{2T} \mathbf{a}_4^2 \mathbf{a}_3^2 \cdot RSR(r_{3,3}^2) \cdot RSR(r_{3,3}^2)
\end{aligned} \tag{40}$$

Finally, the values of $r_{4,4}^2$ and $RSR(r_{4,4}^2)$ are shown as follows

$$r_{4,4}^2 = (a_{1,4}^3)^2 + (a_{2,4}^3)^2 + (a_{3,4}^3)^2 + (a_{4,4}^3)^2 \quad (41)$$

$$RSR(r_{4,4}^2) = \frac{1}{\sqrt{r_{4,4}^2}} \quad (42)$$

Hence, the element $r_{4,4}$ and the fourth column vector \mathbf{q}_4 of \mathbf{Q} can be get as Eq. 43

$$\begin{aligned} q_{1,4} &= a_{1,4}^3 \cdot RSR(r_{4,4}^2), \\ q_{2,4} &= a_{2,4}^3 \cdot RSR(r_{4,4}^2), \\ q_{3,4} &= a_{3,4}^3 \cdot RSR(r_{4,4}^2), \\ q_{4,4} &= a_{4,4}^3 \cdot RSR(r_{4,4}^2), \\ r_{4,4} &= r_{4,4}^2 \cdot RSR(r_{4,4}^2) \end{aligned} \quad (43)$$

To compute the value of $RSR(x)$, a NR based method is proposed in [19]. This method is made up of the following steps:

Firstly, x should be scaled into the working range $wr = [0.5, 2]$ by Eq. 44. The value of z is in the closed interval and n represents the scaling factor. Therefore, the computation of $RSR(x)$ is converted to the computation of $RSR(z)$, and then de-scale the computation result by $\frac{1}{2^{\frac{n}{2}}}$.

$$RSR(x) = \frac{1}{2^{\frac{n}{2}} \sqrt{\frac{x}{2^n}}} = \left(\frac{1}{2^{\frac{n}{2}}} \right) \left(\frac{1}{\sqrt{z}} \right) \quad (44)$$

Secondly, the piecewise polynomial approximation is used to compute the approximation result of $RSR(z)$ in the wr range, that is shown as Eq. 45. Where w_0 is used to be the seed of the NR processing, and a_i is the polynomial coefficients.

$$w_0 = \sum_{i=0}^2 a_i \cdot z_i \quad (45)$$

Thirdly, using the seed (w_0) performs one iteration with the NR method as shown in Eq. 46.

$$w_1 = \frac{w_0}{2} (3 - zw_0^2) \quad (46)$$

Finally, the value of w_1 should be de-scaled to the original data range by Eq. 47.

$$RSR(x) = \left(\frac{1}{2^{\frac{n}{2}}} \right) w_1 \quad (47)$$

The aforementioned description is the proposed the MILMGS algorithm, which is completed by multiplier, adder and shifter. Compared to the existing MGS algorithm and the ILMGS algorithm, the proposed MILMGS algorithm eliminates the square-root and the division operations.

3 Hardware Architecture Design Based on the Proposed MILMGS Algorithm

The proposed MILMGS algorithm has been presented in Sect. 2.3. A new TSA architecture based on the proposed MILMGS for 4×4 real matrix will be presented in this section.

3.1 The TSA Architecture with MILMGS Algorithm

Figure 1 is the TSA hardware architecture based on the proposed MILMGS algorithm for the 4×4 real matrix. The designed architecture consists of four Block1, six Block2 and three Block3. Block1, Block2 and Block3 are the basic module to implement a basic operation. Block1 is used to calculate the value of $RSR(r_{i,i}^2)$, $r_{i,i}$ and the column vector \mathbf{q}_i of unitary matrix \mathbf{Q} . Block2 mainly computes the next iterative matrix by the current matrix and the value of $RSR(r_{i,i}^2)$. Block3 is used to obtain the value of $r_{i,j}$, $i < j \leq 4$.

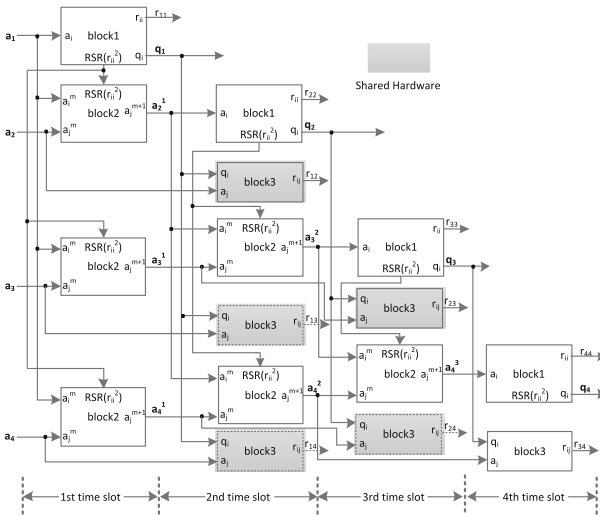


Fig. 1. The TSA architecture for 4×4 real matrix

The primary advantage of the TSA architecture is the simple and regular characteristic, and can easily implement high throughput. However, the TSA architecture needs more hardware overhead than the iterative architecture. As shown in Fig. 1, the hardware sharing technique is used to reduce the hardware overhead. In the second time slot, only one full pipelined Block3 is used to compute the value of $r_{1,j}$, $2 \leq j \leq 4$. Similarly, in the third time slot, we use one full pipelined Block3 to compute $r_{2,3}$ and $r_{2,4}$.

3.2 The Basic Blocks

In this section, the detailed design of the basic blocks will be presented. Figure 2 is the hardware architecture of Block1, which is used to compute the value of $r_{i,i}$, $RSR(r_{i,i}^2)$ and \mathbf{q}_i . Block1 contains the RSR module, and the kernel of RSR is shown in Fig. 3.

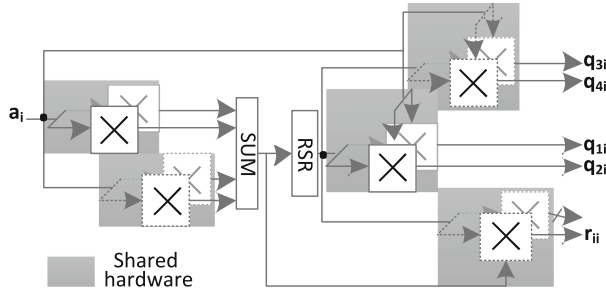


Fig. 2. The hardware architecture of Block1

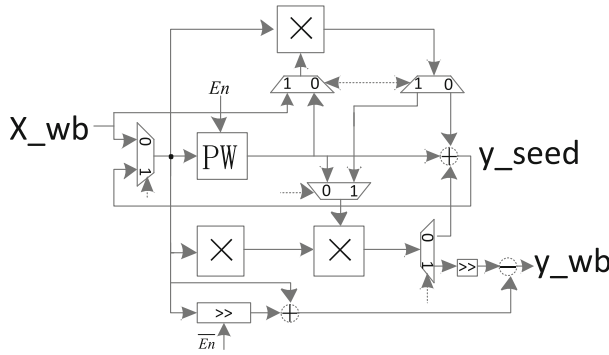


Fig. 3. The Kernel hardware architecture of RSR

The Fig. 3 shows that the designed RSR hardware consists of three multipliers and five multiplexers. The processing latency and throughput are four clock cycles and three clock cycles respectively. Therefore, the hardware sharing technique is used to reduce the hardware overhead in the Block1, and the throughput of Block1 is also three clock cycles.

Figure 4 is the hardware architecture of Block2, which mainly consists five multipliers, three adders (module SUM consists three adders), two subtractors. The aforementioned description shows that the throughput of RSR is three clock cycles. Because we can obtain a new $RSR(r_{i,i}^2)$ value every three clock cycles, the multipliers are shared in the designed Block2. The throughput of Block2 is also three clock cycles as the throughput of RSR architecture.

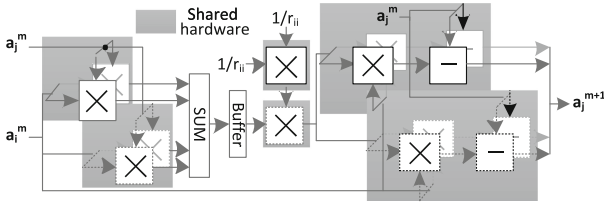


Fig. 4. The hardware architecture of Block2

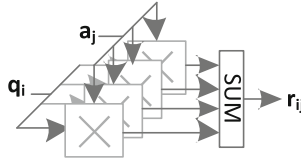


Fig. 5. The hardware architecture of Block3

Figure 5 shows the hardware architecture of Block3, which consists four multipliers and three adders (module *SUM* consists three adders). The full pipelined architecture is used to design the Block3, hence, the Block3 can be shared in the TSA architecture, that is shown in Fig. 1.

4 Implementation Results and Comparisons

To verify the performance of the designed hardware architecture based on the MILMGS algorithm for 4×4 real matrix, the TSA hardware architecture shown in Fig. 1 is implemented in $0.13 \mu\text{m}$ CMOS technology. We will demonstrate the implementation results and some comparisons in the following.

The 16-bit word is used in our design. In order to choice the better fractional bits, we simulate the proposed MILMGS algorithm with various configurations in MATLAB. The mean error of upper triangular matrix \mathbf{R} and the mean error of unitary matrix \mathbf{Q} are presented in Figs. 6 and 7 respectively. The X-axle is the fractional bits of *RSR* module, and curves represent the different fractional bits of data. According to the simulation results, the data format of *RSR* module and the other modules are set to be one sign bit, two integer bits and thirteen fractional bits.

Table 1 shows the implementation results of the designed hardware architecture. To decompose a 4×4 real matrix, the designed hardware architecture needs 250.2K gate count (GC). The critical path (CP) and throughput are 3.5 ns and 95.2M/s respectively. The processing latency (PL) is defined as the number of clock cycles between the matrix feeding into the TSA architecture and the results outputting from the designed architecture. The PL of designed architecture is 53 clock cycles.

Compared to the existing hardware architectures based on the MGS algorithm [14] and the ILMGS algorithm [18], the designed hardware architecture based on the proposed MILMGS algorithm adopts the architecture of TSA,

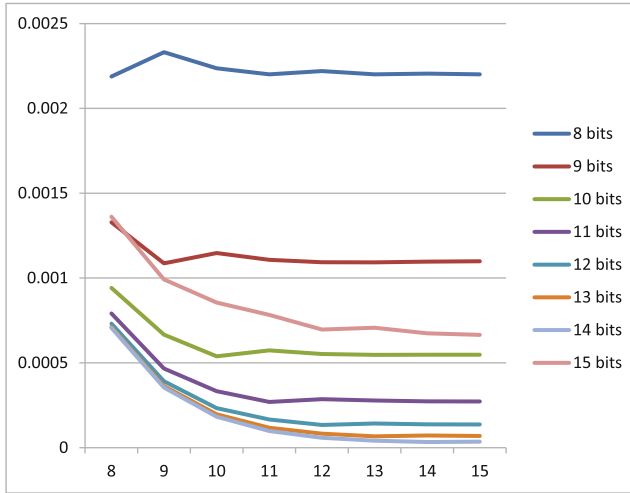


Fig. 6. The obtained \mathbf{R} matrix performance by our designed hardware architecture versus word-length settings

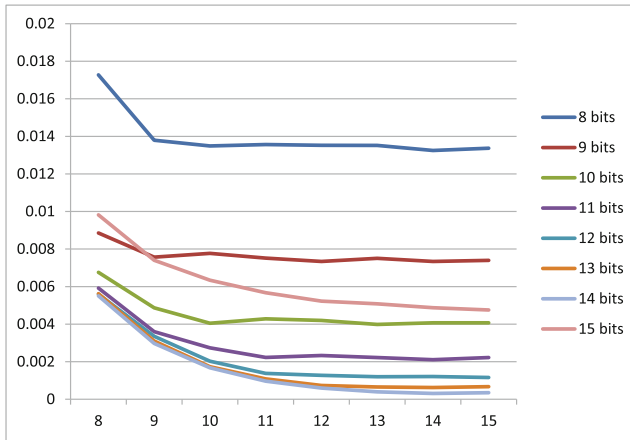


Fig. 7. The obtained \mathbf{Q} matrix performance by our designed hardware architecture versus word-length settings

which is easy to implement higher throughput. Hence, the throughput of architecture designed in this paper is superior to the architecture designed in [14, 18]. The drawback of TSA architecture is that it needs more area overhead than the iterative architecture, so our architecture needs more gate count than the architectures in [14, 18]. Compared to the existing architectures proposed in [20, 21], the throughput performance and the processing latency performance of our designed architecture are better. Moreover, the performance of gate count, throughput and processing latency are all better than the architectures designed in [22, 23].

Table 1. Performance comparison of QRD.

	Algorithm	Technology	Matrix size	WL	PL	GC	CP	Throughput
This work	MILMGS	0.13 μm	Real 4×4	16	53	250.2 k	3.5 ns	95.2 M/s
Liu et al. [18]	ILMGS	0.13 μm	Real 4×4	14	31	52.3 k	1.5 ns	21.5 M/s
Chang et al. [14]	MGS	0.18 μm	Real 4×4	14	35	32.6 k	2.5 ns	11.4 M/s
Shabany et al. [20]	Hybrid	0.13 μm	Complex 4×4	16	150	36 k	3.6 ns	6.95 M/s
Lin et al. [21]	GR	0.18 μm	Complex 4×4	16	96	192.1 k	5 ns	25 M/s
Chen et al. [22]	GR	90 nm	Complex 8×8	15	–	1098 k	13.2 ns	9.46 M/s
Zhang et al. [23]	CGRA	65 nm	Complex 4×4	16	–	1055 k	2 ns	39.6 M/s

5 Conclusion

QRD has been a vital component in the transceiver processor for the MU-MIMO wireless communication systems. This paper proposes a novel MILMGS algorithm based on the existing MGS algorithm and ILMGS algorithm for MIMO systems. To eliminate the square root and division operations, the NR algorithm is used in the proposed MILMGS algorithm. A TSA architecture based on the proposed MILMGS algorithm is also implemented in 0.13 μm CMOS technology. The results show that the designed architecture needs 250.2 gate count, the critical path and throughput are 3.5 ns and 95.2 M/s respectively. Compared to the architectures designed in [14, 18, 20–23], the throughput of our architecture is superior. Moreover, although only the hardware architecture for 4×4 real matrix is presented in this paper, the designed architecture is regular, and can be easily extended to the common size of MIMO wireless communication systems.

Acknowledgments. This work is supported by National Science Foundation of China (Grant No. 61170083) and Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20114307110001).

References

1. IEEE: IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications amendment 6: Wireless access in vehicular environments. IEEE Std 802.11p-2010 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, IEEE Std 802.11n-2009, and IEEE Std 802.11w-2009), pp. 1–51, July 2010
2. IEEE: IEEE standard for information technology - telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements - part 11: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications amendment 5: Television white spaces (TVWS) operation. IEEE Std 802.11af-2013 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012, IEEE Std 802.11aa-2012, IEEE Std 802.11ad-2012, and IEEE Std 802.11ac-2013), pp. 1–198, February 2014

3. IEEE: IEEE standard for local and metropolitan area networks part 16: air interface for broadband wireless access systems. IEEE Std 802.16-2009 (Revision of IEEE Std 802.16-2004), pp. 1–2080, May 2009
4. Acharya, J., Gao, L., Gaur, S.: Performance comparison of ZF-DPC to block diagonalization for quantized feedback. In: 2013 ASILOMAR Conference on Signals, Systems and Computers, pp. 1238–1242, November 2013
5. Li, H., Liu, S., Gudaitis, M.: Optimal interference pre-cancellation order in DPC-based broadcast and unicast hybrid network. In: 2013 47th Annual Conference on Information Sciences and Systems (CISS), pp. 1–6, March 2013
6. Costa, M.: Writing on dirty paper (corresp.). *IEEE Trans. Inf. Theory* **29**(3), 439–441 (1983)
7. Krikidis, I., Ottersten, B.: Diversity fairness in Tomlinson-Harashima Precoded multiuser MIMO through retransmission. *IEEE Sig. Process. Lett.* **20**(4), 375–378 (2013)
8. Tseng, F.-S., Wang, Y.-C., Hsu, C.-Y., Lin, S.-S.: Robust Tomlinson-Harashima Precoder design with random vector quantization in MIMO systems. *IEEE Commun. Lett.* **18**(2), 265–268 (2014)
9. Chen, C.-E., Cho, T.-W., Chung, W.-H.: Blockwise-lattice-reduction-aided Tomlinson-Harashima Precoder designs for MU-MIMO downlink communications with clusters of correlated users. *IEEE Trans. Veh. Technol.* **63**(3), 1146–1159 (2014)
10. Hwang, Y.-T., Chen, W.-D.: A low complexity complex QR factorization design for signal detection in MIMO OFDM systems. In: IEEE International Symposium on Circuits and Systems, ISCAS 2008, pp. 932–935, May 2008
11. Patel, D., Shabany, M., Gulak, P.: A low-complexity high-speed QR decomposition implementation for MIMO receivers. In: IEEE International Symposium on Circuits and Systems, ISCAS 2009, pp. 33–36, May 2009
12. Liu, T.H., Chiu, C.N., Liu, P.Y., Chu, Y.S.: Block-wise QR-decomposition for the layered and hybrid alamouti STBC MIMO systems: algorithms and hardware architectures. *IEEE Trans. Sig. Process.* **62**(18), 4737–4747 (2014)
13. Liu, C., Xing, Z., Yuan, L., Tang, C., Zhang, Y.: A novel architecture to eliminate bottlenecks in parallel tiled QRD algorithm for future MIMO systems. *IEEE Trans. Circ. Syst. II Express Briefs* **64**(1), 26–30 (2016)
14. Chang, R.-H., Lin, C.-H., Lin, K.-H., Huang, C.-L., Chen, F.-C.: Iterative QR decomposition architecture using the modified gram-schmidt algorithm for MIMO systems. *IEEE Trans. Circuits Syst. I Regul. Pap.* **57**(5), 1095–1102 (2010)
15. Singh, C., Prasad, S.H., Balsara, P.: VLSI architecture for matrix inversion using modified Gram-Schmidt based QR decomposition. In: 20th International Conference on VLSI Design, Held Jointly with 6th International Conference on Embedded Systems, pp. 836–841, January 2007
16. Lin, K.-H., Chang, R.-H., Lin, H.-L., Wu, C.-F.: Analysis and architecture design of a downlink M-Modification MC-CDMA system using the Tomlinson-Harashima precoding technique. *IEEE Trans. Veh. Technol.* **57**(3), 1387–1397 (2008)
17. Lin, K.-H., Lin, C.-H., Chang, R.-H., Huang, C.-L., Chen, F.-C.: Iterative QR decomposition architecture using the modified Gram-Schmidt algorithm. In: IEEE International Symposium on Circuits and Systems, ISCAS 2009, pp. 1409–1412, May 2009
18. Liu, C., Tang, C., Yuan, L., Xing, Z., Zhang, Y.: QR decomposition architecture using the iteration look-ahead modified Gram-Schmidt algorithm. *IET Circuits Devices Syst.* **10**(5), 402–409 (2016)

19. Pizano-Escalante, L., Parra-Michel, R., Castillo, J.V., Longoria-Gandara, O.: Fast bit-accurate reciprocal square root. *Microprocess. Microsyst.* **39**(2), 74–82 (2015)
20. Shabany, M., Patel, D., Gulak, P.: A low-latency low-power QR-decomposition ASIC implementation in 0.13 μm CMOS. *IEEE Trans. Circuits Syst. I Regul. Pap.* **60**(2), 327–340 (2013)
21. Lin, J.S., Hwang, Y.T., Fang, S.H., Chu, P.H.: Low-complexity high-throughput QR decomposition design for MIMO systems. *IEEE Trans. Very Large Scale Integr. Syst.* **23**(10), 2342–2346 (2015)
22. Chen, C.M., Lin, C.H., Tsai, P.Y.: Multi-mode sorted QR decomposition for 4×4 and 8×8 single-user/multi-user MIMO precoding. In: 2015 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2980–2983, May 2015
23. Zhang, C., Liu, L., Markovic, D., Owall, V.: A heterogeneous reconfigurable cell array for MIMO signal processing. *IEEE Trans. Circuits Syst. I Regul. Pap.* **62**(3), 733–742 (2015)