Iterative QR Decomposition Architecture Using the Modified Gram-Schmidt Algorithm

Kuang-Hao Lin, Chih-Hung Lin, Robert Chen-Hao Chang, Member, IEEE, Chien-Lin Huang, and Feng-Chi Chen***

*Department of Electronic Engineering, National Chin-Yi University of Technology, Taichung, Taiwan
khlin66@gmail.com
Department of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan
**SoC Technology Center, Industrial Technology Research Institute, Hsinchu, Taiwan

Abstract—**Implementation of iterative QR decomposition (QRD) architecture based on the modified Gram-Schmidt (MGS) algorithm is proposed in this paper. In order to achieve computational efficiency with robust numerical stability, a triangular systolic array (TSA) for QRD of large size matrices is presented. Therefore, the TSA architecture can be modified into iterative architecture for reducing hardware cost that is called iterative QRD (IQRD). The IQRD hardware is constructed by the diagonal process (DP) and the triangular process (TP) with fewer gate counts and lower power consumption than TSAQRD. For a 4×4 matrix, the hardware area of the proposed IQRD can reduce about 76% of the gate counts in TSAQRD. For a generic square matrix of order *n* IQRD, the latency required is 2*n*-1 time units, which is based on the MGS algorithm. Thus, the total clock latency is only *n*(2*n*+3) cycles.**

I. Introduction

A recent surge of research on wireless local area networks (WLANs) has given us new challenges and opportunities. With the increasing usage of wireless communication systems, reliability requirements for high data rate have become more critical. In the current decade multiple input multiple output (MIMO) systems have generated tremendous research interest as they offer high reliability and high throughput [1]. To exploit the full potential of gains offered by MIMO, computationally efficient design of a wireless baseband communication receiver has become difficult and challenging. A detection circuit involved in a MIMO receiver has to be designed for high data throughput. The computational accuracy of the MIMO detection has direct consequence on the throughput and reliability achieved in the receiver.

QR decomposition (QRD) for MIMO detection pre-processor is an essential component of all MIMO receiver [2], [3]. The QRD process the channel response **H** first, and then decompose it into **Q** and **R** matrices to produce **H**=**QR**. The detected vector $\hat{\mathbf{s}}$ is computed based on the maximum likelihood (ML) algorithm with QRD as given in Eq. (1)

$$\begin{aligned}\hat{\mathbf{s}} &= \arg\min \left\|\mathbf{Y}-\mathbf{H}\mathbf{X}\right\|^2 = \arg\min \left\|\mathbf{Q}^H\mathbf{Y}-\mathbf{R}\mathbf{X}\right\|^2 \\ &= \arg\min \left\|\hat{\mathbf{y}}-\mathbf{R}\mathbf{X}\right\|^2\end{aligned} \tag{1}$$

where **Q** denotes a *Nr*×*Nt* orthogonal matrix and **R** represents an upper triangular matrix.

Related studies on the QRD architecture can be classified into two major hardware implementation categories. The first category is the triangular systolic array (TSA), based on the coordinate rotation digital computer (CORDIC) algorithm [4], [5]. The other category is a parallel architecture based on the modified Gram-Schmidt (MGS) algorithm [6], [7]. However, within the extensive literature on QRD, relatively few studies focus on improving the clock latency and hardware area.

This paper proposes an iterative QR decomposition (IQRD) hardware architecture based on the MGS method. From a hardware point of view, the IQRD method is constructed using the modified TSA architecture with lower gate count and clock latency than TSA structures. The QRD architecture uses an iteration operation to reduce hardware complexity via diagonal process (DP) and triangular process (TP) circuits. Simulation results also show the performance of QRD at different wordlength numbers.

This paper is organized as follows. Section II discusses the QRD algorithms. In section III, we present the efficient QRD architecture design. In Section IV, the simulation and implementation results are presented. Conclusions are drawn in Section V.

II. QR Decomposition

The common QRD algorithms are Givens rotation [8], MGS orthogonalization [9], Householder transformation [10], etc. Since Householder transformation is much more complex in hardware implementation than the other two algorithms, this paper only discusses the Givens rotation and the MGS orthogonalization.

A. Givens Rotation

The Givens rotation rotates in the plane expended by two coordinate axes. The rotation matrix can be expressed as

This work was supported in part by the National Science Council (NSC), Taiwan, R.O.C. under Grant NSC 97-2221-E-005-086-MY2 and in part by the Ministry of Education, Taiwan, R.O.C. under the ATU plan. The authors would like to thank the National Chip Implementation Center (CIC) of Taiwan for technical support.

978-1-4244-3828-0/09/$25.00 ©2009 IEEE

$$\mathbf{G}(i,k,\theta)=\begin{bmatrix}1 & \cdots & 0 & \cdots & 0 & \cdots & 0\\ \vdots & \ddots & \vdots & & \vdots & & \vdots\\ 0 & \cdots & \cos\theta_{i,i} & \cdots & -\sin\theta_{i,k} & \cdots & 0\\ \vdots & & \vdots & \ddots & \vdots & & \vdots\\ 0 & \cdots & \sin\theta_{k,i} & \cdots & \cos\theta_{k,k} & \cdots & 0\\ \vdots & & \vdots & & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1\end{bmatrix} \tag{2}$$

Here, $\mathbf{G}(i,k,\theta)^{\mathrm{T}}\cdot\mathrm{X}$ denotes that vector X counterclockwise rotates an angle θ in (i,k) plane. When the rotation matrix **G** multiplies another matrix **A**, it only affects the i-th and k-th row of matrix **A**. Therefore, the rotation matrix **G** can be simplified as

$$\mathbf{G}=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix} \tag{3}$$

The mathematical form can be expressed as

$$\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}\mathrm{r}\\ 0\end{bmatrix} \tag{4}$$

where $\mathrm{r}=\sqrt{(x)^2+(y)^2}$, cos$\theta$=$x$/r, sin$\theta$=(-$y$/r). The rotation matrix **G** value can be obtained from r.

Based on Eq. (4), if QRD is to be executed on any matrix, an up-triangular matrix **R** can be obtained only through successive multiplication of the rotation matrix **G**. The orthogonal matrix **Q** can also be obtained by executing the same operation on identity matrix **I**. When the QRD is achieved by the Givens rotation method, the hardware complexity can be reduced by the CORDIC operation core with vectoring and rotating modes.

B. *Modified Gram-Schmidt*

The QRD for any matrix can be realized by the MGS orthogonalization algorithm [6]. This procedure reduces the memory hardware consumption. The MGS orthogonalization is described as follows.

a_{ij} denotes row i and column j of the matrix **A,** and $\mathbf{a}_i$ denotes the column vector of the matrix **A**. The QRD of matrix **A**, **A**=**QR**, can be obtained by the following steps. For simplicity, the 3×3 matrix **A** is utilized to explain the procedure. First, the first column vector $\mathbf{a}_1$ is normalized to obtain $r_{11}=\|\mathbf{a}_1\|_2=\sqrt{(a_{11})^2+(a_{21})^2+(a_{31})^2}$, which represents the element in row 1 and column 1 of **R**. The value of $\mathbf{q}_1$, which is the 1st column vector of **Q**, can be calculated from r_{11}.

$$q_{11}=\frac{a_{11}}{r_{11}},q_{21}=\frac{a_{21}}{r_{11}},q_{31}=\frac{a_{31}}{r_{11}} \tag{5}$$

Second, the r_{12} and r_{13} can be calculated using column vector 1 of the matrix **Q** ($\mathbf{q}_1$) and the 2nd and 3rd column vectors of matrix **A**.

$$\begin{aligned}r_{12}&=\mathbf{q}_1{}^T\mathbf{a}_2=q_{11}a_{12}+q_{21}a_{22}+q_{31}a_{32}\\ r_{13}&=\mathbf{q}_1{}^T\mathbf{a}_3=q_{11}a_{13}+q_{21}a_{23}+q_{31}a_{33}\end{aligned} \tag{6}$$

After obtaining $\mathbf{q}_1$, r_{12} and r_{13} from Eq. (5) and Eq. (6), the matrix **A** can be converted to matrix A^{p1}.

$$\begin{aligned}&a_{11}^{p1}=0 \quad a_{12}^{p1}=a_{12}-r_{12}q_{11} \quad a_{13}^{p1}=a_{13}-r_{13}q_{11}\\ &a_{21}^{p1}=0\,,\ a_{22}^{p1}=a_{22}-r_{12}q_{21}\,,\ a_{23}^{p1}=a_{23}-r_{13}q_{21}\\ &a_{31}^{p1}=0 \quad a_{32}^{p1}=a_{32}-r_{12}q_{31} \quad a_{33}^{p1}=a_{33}-r_{13}q_{31}\end{aligned} \tag{7}$$

So far, these steps have derived the values of $\mathbf{q}_1$, r_{11}, r_{12}, r_{13} and the matrix A^{p1}. The above steps are then repeated with A^{p1} to compute $\mathbf{q}_2$, r_{22}, r_{23} and the matrix A^{p2}. The 2nd column vector of the matrix A^{p1} is normalized to obtain the r_{22} i.e., $r_{22}=\|\mathbf{a}_2^{p1}\|_2=\sqrt{(a_{12}^{p1})^2+(a_{22}^{p1})^2+(a_{32}^{p1})^2}$. The $\mathbf{q}_2$ and r_{23} can then be calculated.

$$q_{12}=\frac{a_{12}^{p1}}{r_{22}},q_{22}=\frac{a_{22}^{p1}}{r_{22}},q_{32}=\frac{a_{32}^{p1}}{r_{22}} \tag{8}$$

$$r_{23}=\mathbf{q}_2{}^T\mathbf{a}_3^{p1}=q_{12}a_{13}^{p1}+q_{22}a_{23}^{p1}+q_{32}a_{33}^{p1} \tag{9}$$

The matrix A^{p2} is obtained from the following equation.

$$\begin{aligned}&a_{11}^{p2}=0 \quad a_{12}^{p2}=0 \quad a_{13}^{p2}=a_{13}^{p1}-r_{23}q_{12}\\ &a_{21}^{p2}=0\,,\ a_{22}^{p2}=0\,,\ a_{23}^{p2}=a_{23}^{p1}-r_{23}q_{22}\\ &a_{31}^{p2}=0 \quad a_{32}^{p2}=0 \quad a_{33}^{p2}=a_{33}^{p1}-r_{23}q_{32}\end{aligned} \tag{10}$$

Similarly, by repeating the steps in the matrix A^{p2}, r_{33} can be computed by normalizing the 3rd column vector of matrix A^{p2}, i.e., $r_{33}=\|\mathbf{a}_3^{p2}\|_2=\sqrt{(a_{13}^{p2})^2+(a_{23}^{p2})^2+(a_{33}^{p2})^2}$, thus giving

$$q_{13}=\frac{a_{13}^{p2}}{r_{33}},q_{23}=\frac{a_{23}^{p2}}{r_{33}},q_{33}=\frac{a_{33}^{p2}}{r_{33}} \tag{11}$$

Finally, **A**=**QR** is obtained, in which

$$\mathbf{Q}=\begin{bmatrix}q_{11} & q_{12} & q_{13}\\ q_{21} & q_{22} & q_{23}\\ q_{31} & q_{32} & q_{33}\end{bmatrix},\quad \mathbf{R}=\begin{bmatrix}r_{11} & r_{12} & r_{13}\\ 0 & r_{22} & r_{23}\\ 0 & 0 & r_{33}\end{bmatrix}.$$

III. Architecture Design

The demonstration above shows that the QRD can be separated into two steps. The first step computes the diagonal line elements of the upper triangular matrix **R** and the unitary matrix **Q**. A diagonal line element (r_{jj}) indicates the square root of the summation of the square of each element in the column j of **a** ($\mathbf{a}_j$). To save hardware, $\mathbf{a}_j$ elements are sequentially imported to diagonal process (DP), and only a squarer, an adder, and a square root (sqrt) operator are adopted, as shown in Fig. 1. Additionally, the $\mathbf{a}_j$ elements are delayed by buffer register (B) and sequentially divided by the r_{jj}. Therefore, the $\mathbf{q}_j$ of the unitary matrix **Q** can be derived.

In the second step, $\mathbf{q}_j$ values of DP are sequentially input to triangular process (TP), and the matrix **R** of the non-diagonal line elements (r_{ij}) with new matrix $A\left(\mathbf{a}_i^p\right)$ is computed. Fig. 2 illustrates the block II hardware architecture, which uses two multipliers, an adder, and a subtractor. TP can be modified by multiplexers, and the hardware area is reduced by eliminating a multiplier, as shown in Fig. 3. The select signals of the multiplexer and demultiplexer in modified TP are "0" so that the $\mathbf{a}_j$ and $\mathbf{q}_j$ inner product accumulates. Then, select signals are "1" so that $A\left(\mathbf{a}_i^p\right)$ is computed.

All of the **Q** and **R** matrix element values can be derived by the repetitive operation of these two blocks. Figure 4 illustrates the triangular systolic array QRD (TSAQRD) for a 4×4 matrix, and indicates that the hardware area substantially increases as the rank of the matrix increases. Therefore, the iterative QRD (IQRD) using the feedback control circuit and hardware sharing can significantly reduce the hardware area, as Fig. 5 shows.

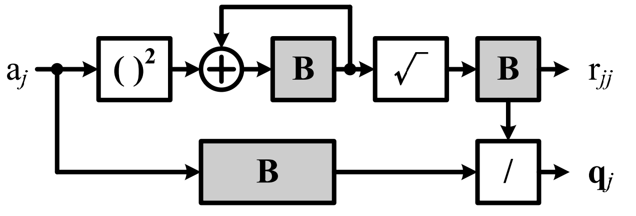

Figure 1. Diagonal process hardware architecture of QRD.

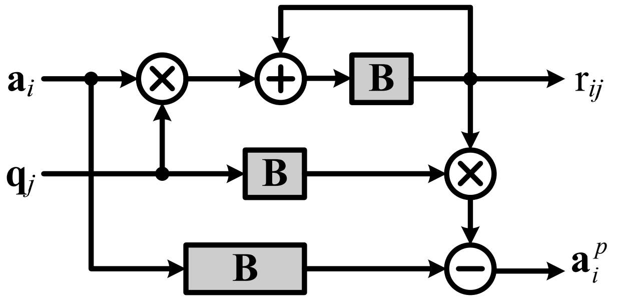

Figure 2. Triangular process hardware architecture of QRD.

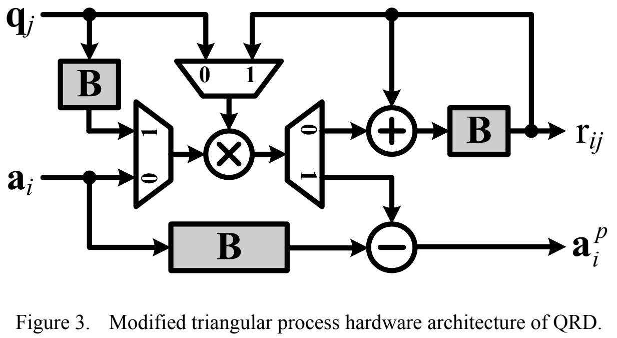

Figure 3. Modified triangular process hardware architecture of QRD.

A. Triangular Systolic Array

The TSA consists of a set of processing units. Each processing unit can perform some simple operations. The advantages of this architecture are a simple and regular design that can speed up computation flow. Figure 4 shows the TSA architecture for QRD with MGS algorithm. Using the TSA architecture for QRD hardware can offers high throughput, but the processing units increase with the dimension of **H**.

The TSAQRD sequential operation needs seven time slots from t_1 to t_7 when a 4×4 matrix is decomposed. DP executes in odd time slots (t_1, t_3, …, t_7) and TP executes in even time slots (t_2, t_4, t_6). In the first time slot t_1, a_1 operates pass through DP circuit, obtaining r_{11} and $\mathbf{q}_1$. $\mathbf{q}_1$ distributes pass through TP circuits. In addition, (a_2, …, a_4) passes through delay buffer (B) and waits for $\mathbf{q}_1$ to operate in TP circuit. In the next time slot t_2, (a_2, …, a_8) and $\mathbf{q}_1$ are computed by TP, and then (r_{12}, …, r_{14}) and $\left(\mathbf{a}_2^{p1}, \cdots, \mathbf{a}_4^{p1}\right)$ are generated, respectively. In time slot t_3, $\mathbf{a}_2^{p1}$ operates in DP, and $\left(\mathbf{a}_3^{p1}, \mathbf{a}_4^{p1}\right)$ sequentially feeds back to delay buffer (B). Therefore, repetitive operation using DP and TP accomplishes QRD. The hardware area is defined as $G_{DP}m + G_{TP}\sum_{i=1}^{m-1}(m-i)$ gate counts, where G_{DP} is gate count of a DP circuit and G_{TP} is gate count of a TP circuit.

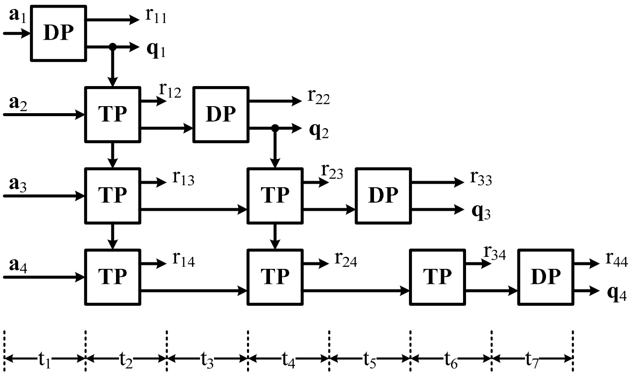

Figure 4. TSAQRD for a 4×4 matrix.

B. Iterative QR Decomposition

This paper proposes the IQRD architecture shown in Fig. 5. The IQRD sequential operation also needs seven time slots. For a generic square matrix of order n, the DP latency required is n time units and the TP latency required is $(n-1)$ time units. Thus, the total latency of matrix IQRD is $n+(n-1)=(2n-1)$ time units, which is better than the previously proposed architecture of Singh *et al.* [6]. The clock latency cycles are $(n+2)$, $(n+1)$ and $(n+1)$ for DP, TP, and the last output data, respectively. Therefore, the total latency clock is $n(n+2)+(n-1)(n+1)+(n+1)=n(2n+3)$ cycles. The IQRD structure is proposed based on TSA to reduce complexity and hardware area, which can be defined as G_{DP}+G_{TP}(m-1) gate counts. Compared with the TSAQRD, IQRD hardware can decrease $G_{DP}(m-1)+G_{TP}\sum_{i=2}^{m-1}(m-i)$ gate counts. The hardware area of the proposed IQRD reduces about 76% of the gate counts in TSAQRD for a 4×4 matrix.

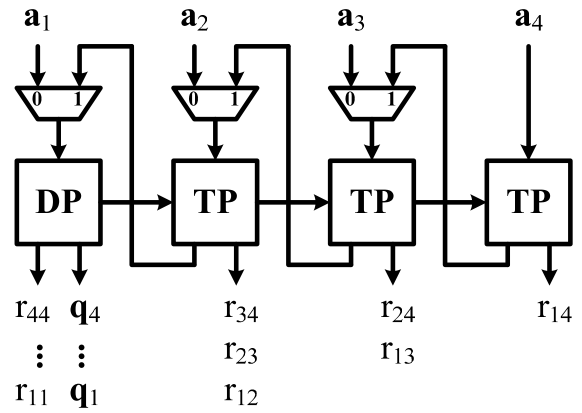

Figure 5. IQRD for a 4×4 matrix.

IV. Simulation and Implementation Result

This paper performs fixed-point simulation to determine the best number of bits for hardware implementation. Figure 6 shows the mean square error (MSE) versus fractional wordlength at IQRD structure. The MSE represents the error between the fixed-point and floating-point QR decomposition output. In Fig. 6, the curve is saturated with 10^{-8} MSE. Increasing the number of bits does not increase system performance. Therefore, a 5-bit fractional wordlength is sufficient in system hardware design.

Table 1 illustrates the comparison results of this paper with the folding structure [5] and the Gram-Schmidt algorithm [6]. Although the folding structure has similar gate count to the IQRD structure, the memory required is 2416 bits and clock latency is more than the proposed structure. As Table 1 shows, the proposed IQRD architecture has lower clock latency than Ref. [5] and a smaller hardware area (gate count) than the TSA structure. Compared with the TSA and IQRD of Gram-Schmidt approach, it reduces gate count at the same matrix order by 76%.

V. Conclusion

The hardware architecture design of QR decomposition is extensively discussed in current MIMO detection system studies on enhancing operational efficiency. The most popular architecture adopted is the TSA with processing elements based on CORDIC computing. In this paper, we adopted the TSA structure to implement QRD with MGS, and then it is modified by iterative structure to achieve better clock latency and chip area. The total latency clock of IQRD is $n(2n+3)$ cycles and area is $G_{DP}+G_{TP}(m\text{-}1)$ gate counts. The proposed architecture is implemented and verified by TSMC 0.18 μm CMOS technology.

References

[1] A.J. Paulraj, D.A. Gore, R.U. Nabar, and H. Bolcskei, "An overview of MIMO communications - a key to gigabit wireless," *Proceedings IEEE*, vol. 92, pp. 198–218, Feb. 2004.

[2] D. Rawal, and C. Vijaykumar, "QR-RLS based adaptive channel TEQ for OFDM wireless LAN," in *Proc. IEEE Int. Conf. Signal Processing Commun. Networking*, pp. 46–51, Jan. 2008.

[3] K.-H. Lin, R. C.-H. Chang, C.-L. Huang, F.-C. Chen, and S.-C. Lin, "Implementation of QR decomposition for MIMO-OFDM detection systems," *The 15th IEEE Int. Conf. on Electronics, Circuits and Systems*, Malta, Aug. 2008.

[4] A. Maltsev, V. Pestretsov, R. Maslennikov, and A. Khoryaev, "Triangular systolic array with reduced latency for QR-decomposition of complex matrices," in *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 385–388, May 2006.

[5] F. Sobhanmanesh, and S. Nooshabadi, "Parametric minimum hardware QR-factoriser architecture for V-BLAST detection," *IEE Proc. Circuits Devices Syst.*, vol. 153, pp. 433–441, Oct. 2006.

[6] C.K. Singh, S.H. Prasad, and P.T. Balsara, "VLSI architecture for matrix inversion using modified Gram-Schmidt based QR decomposition," in *Proc. Int. Conf. VLSI Design*, pp. 836–841, Jan. 2007.

[7] C. K. Singh, S. H. Prasad, and P. T. Balsara, "A fixed-point implementation for QR decomposition," in *Proc. IEEE Dallas Workshop Design Applicat. Integration Software*, pp. 75–78, Oct. 2006.

[8] S. Wang, and Jr. E. E., Swartzlander, "The critically damped CORDIC algorithm for QR decomposition," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.,* pp. 908–911, Nov. 1996.

[9] H. Sakai, "Recursive least-squares algorithms of modified Gram-Schmidt type for parallel weight extraction," *IEEE Trans. Signal Procss.*, vol. 42, pp. 429–433, Feb. 1994.

[10] S.-F. Hsiao and J.-M. Delosme, "Householder CORDIC algorithms," *IEEE Trans. Comput.*, vol. 44, pp. 990–1001, Aug. 1995

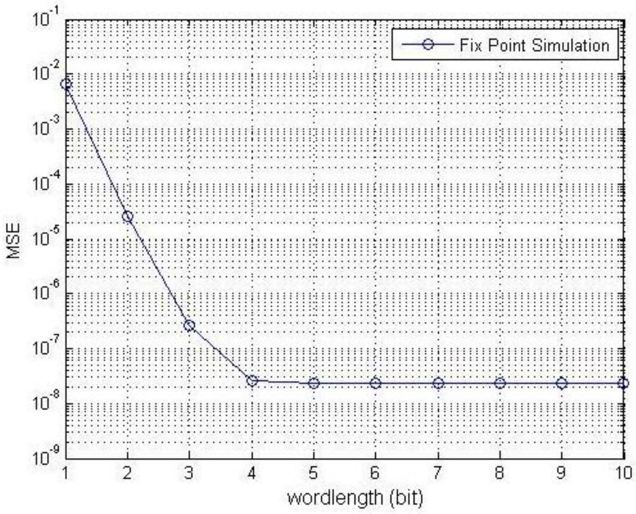

Figure 6. Fixed-point simulation of QRD with different wordlength.

TABLE I. Comparisons of the QRD Implementation Results

	This work		Ref.[5]	Ref.[6]
Algorithm	Gram-Schmidt		Givens Rotation	Gram-Schmidt
	TSA	IQRD		
Order	4×4 (real)	4×4 (real)	4×4 (complex)	4×4 (real)
Clock Latency	44	44	252	67
Technology	0.18-μm CMOS	0.18-μm CMOS	FPGA	0.18-μm CMOS
Gate Count	215k	51k	1528LC (about 32k)	72k
Memory	None	None	2416bits	None