



FPGA-Based Implementation of Low Complexity CORDIC-Based Scalable Complex QR Decomposition for MIMO-OFDM Systems

F. Asghariyehlou* and J. Javidan*(C.A.)

Abstract: This paper deals with the optimization of the CORDIC-based modified Gram-Schmidt (MGS) algorithm for QR decomposition (QRD) and presents a scalable algorithm with maximum throughput, the least possible latency, and hardware resources. The optimized algorithm is implemented on Xilinx Virtex 6 FPGA using ISE software as a fixed point with selected accuracy based on the results of MATLAB simulation. Using the loop unrolling technique with different coefficients, an attempt is made to reduce the latency and increase the throughput. In contrast, increasing the unrolling factor leads to a decrease in the frequency of the CORDIC unit as well as a decrease in the number of resources. As a result, there is a trade-off between the unrolling factor and the frequency of the CORDIC unit. By investigating the different unrolling factors, it is shown that the loop unrolling technique with a factor of 4 has the highest throughput with the value of 5.777 MQRD/s and the lowest latency with the value of 173 ns. Moreover, it is shown that throughput and latency are improved by 42.52% and 73.74% respectively compared to the not optimized case. The proposed method is also scalable for different sizes of $m \times m$ complex channel matrices, where $\log_2 m \in \mathbb{N}$.

Keywords: CORDIC Algorithm, MIMO Detection, QR Decomposition, Unrolling Technique.

1 Introduction

NOWADAYS, Multiple-Input-Multiple-Output (MIMO) technology has become an essential part of wireless communication since it improves spectral efficiency and offers diversity gain [1, 2]. Moreover, Orthogonal Frequency Division Multiplexing (OFDM) along with MIMO can boost spectral efficiency. Nevertheless, signal detection in large-scale MIMO systems is a challenging task. Hence, the computationally efficient implementation of signal detection algorithms for large-scale MIMO systems is of great importance [3].

The QR decomposition is an integral part of MIMO detection algorithms due to its robust numerical stability. However, its complexity is exponentially increasing with the number of antennas. Therefore, proposing a novel and efficient implementation of QR decomposition that is scalable would be of great interest.

Any QRD algorithm decomposes a matrix with linearly independent columns (channel state information matrix in this work) into an orthonormal matrix and an upper triangular matrix. There are several conventional QRD approaches in literature [4]: Givens rotation (GR) method which suffers from relatively long latency because it makes zero the matrix entries one by one. Meanwhile, the Gram-Schmidt algorithm (GS) is known for its low latency compared to the Givens rotation method, because it decomposes matrices column-wise. However, since the Gram-Schmidt algorithm involves square roots and multiple divisions, it leads to higher complexity and makes it impractical in terms of latency, especially for large-scale MIMO. On the other hand,

Iranian Journal of Electrical and Electronic Engineering, 2022.

Paper first received 06 June 2021, revised 31 January 2022, and accepted 10 February 2022.

* The authors are with the Department of Electrical Engineering, University of Mohaghegh Ardabili, Ardabil, Iran.

E-mails: mohaghegh92@gmail.com and javidan@uma.ac.ir.

Corresponding Author: J. Javidan.

<https://doi.org/10.22068/IJEEE.18.2.2206>

Householder Transformation (HT) has the highest complexity which makes it not suitable for implementation.

To overcome the implementation complexity of square roots and divisions used in the QRD process, these operations can be realized by the popular CORDIC algorithm [5]. It just uses the basic shift-add operations. Hence, this simplicity of hardware implementation is attractive for most applications. Due to the iterative nature of CORDIC, one of the main challenges is to reduce the number of iterations required, which ultimately leads to a reduction in latency. Some improvement techniques have better performance and throughput, however, a larger area is needed for implementation, and vice versa. One of the methods used is increasing the radix, which increases hardware complexity and power consumption [6]. In [7-9], different methods have been proposed to reduce the latency caused by CORDIC iterations, although, the hardware and area are increased.

In this paper, a scalable algorithm with the least possible latency, hardware, and maximum throughput is presented. The proposed architecture has a slight latency and is also efficient in terms of the area using effective mapping of square root and division operations into CORDIC units. Also, during this operation, the required upper triangular matrix to solve the detection equations of MIMO systems is generated using recursive equations.

In the proposed architecture, the output of CORDIC vectoring mode which is the rotation angle is used as an input of CORDIC rotation mode. This reduces the number of operations needed to find rotation angle for division operation. The proposed architecture

remarkably minimizes the hardware cost of division operations by effectively implementing the vector norm and normalization operation by a pair of CORDIC vectoring and rotation modes.

In summary, our contribution in this paper is twofold:

- First, the optimized coefficient of loop unrolling technique is proposed in order to reduce the overall latency.
- Second, the scalability of the proposed approach for channel matrix dimensions up to 32×32 is shown.

2 Modified Gram-Schmidt-Based QRD

As it is mentioned, most of the QRD approaches are based on the GR, HT, and GS algorithms. By applying a modification to GS, The Modified Gram-Schmidt algorithm [4] has been proposed which introduces smaller errors in finite-precision arithmetic compared to the conventional GS algorithm. The MGS algorithm in each step uses the available information of the previous step instead of the common method. Generally, it is good to use the updated information that is computed in the previous step. Comparison of Implementation results of different QRD algorithms for $m \times m$ complex channel matrix (where $\log_2 m \in \mathbb{N}$) is shown in Table 1. As it can be seen, the modified Gram-Schmidt algorithm needs a smaller number of adders, multipliers, division, and sqrt operations compared to other conventional algorithms. Hence, the modified Gram-Schmidt algorithm is preferred. Table 2 also shows the pseudo-code of the MGS algorithm for the channel matrix $\mathbf{H} = [\mathbf{h}_1 \ \mathbf{h}_2 \ \dots \ \mathbf{h}_m]$.

Table 1 Comparison of Implementation results of different QRD algorithms for $m \times m$ complex channel matrix.

QRD algorithm	Add	Mul	Div	Sqrt
Givens Rotation [10]	$\frac{8}{3}m^3 - \frac{3}{2}m^2 - \frac{7}{6}m$	$4m^3 - m^2 - 3m$	$\frac{5}{2}m^2 - \frac{5}{2}m$	$m^2 - m$
Householder Transform [10]	$\frac{8}{3}m^3 - 7m^2 - \frac{2}{3}m$	$\frac{8}{3}m^3 + 6m^2 + \frac{16}{3}m$	$m^2 + 3m$	$3m$
Modified Gram-Schmidt	$4m^3 - 3m^2$	$4m^3 - 2m^2$	$2m^2$	m

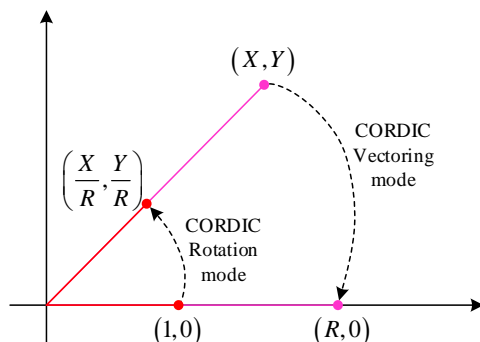


Fig. 1 Division operation using CORDIC unit.

Table 2 Modified Gram-Schmidt pseudocode.

1	For $j = 1:m$
2	$\mathbf{v}_j = \mathbf{h}_j$
3	For $i = 1:j-1$
4	$r_{ij} = \mathbf{q}_i^H \mathbf{h}_j$
5	$\mathbf{v}_j = \mathbf{v}_j - r_{ij} \mathbf{q}_i$
6	End
7	$r_{jj} = \ \mathbf{v}_j\ $
8	$\mathbf{q}_j = \mathbf{v}_j / r_{jj}$
9	End

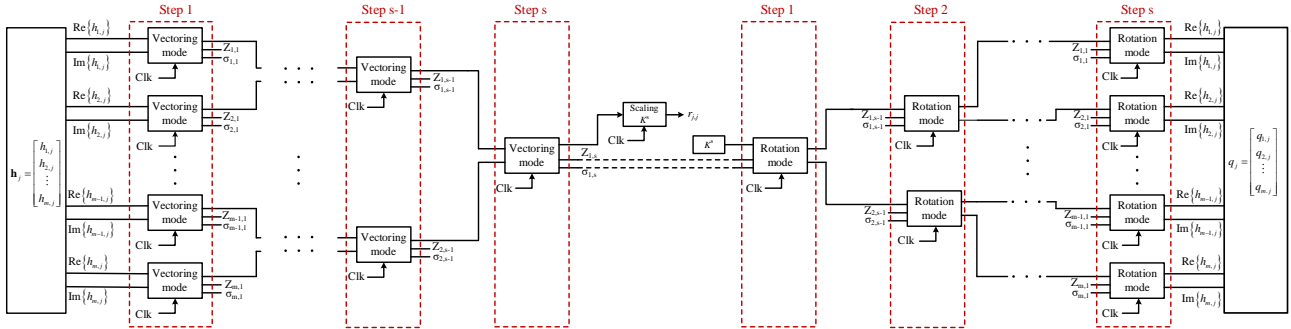


Fig. 2 Proposed CORDIC used in the implemented QRD architecture.

3 Proposed Approach

To overcome the implementation complexity of square roots and divisions used in the QRD process, these operations can be realized by the popular CORDIC algorithm. The CORDIC algorithm is a type of shift and addition-based algorithm that allows a wide range of basic functions to be computed iteratively using fixed point computational operations. This algorithm has been used in many applications, not only because of the cost-effectiveness of hardware but also because of the importance of numerical stability in fixed-point applications. However, the iterative nature of the algorithm introduces a data-dependent latency.

Fig.1 shows the Norm and Normalization process using vectoring and rotation modes for a 2D vector (X, Y). The output of vectoring mode is (R, 0) in which $R = \sqrt{X^2 + Y^2}$. To do the normalization process, the unit vector (1, 0) with an angle Z which is acquired from vectoring mode leads to the output $(\frac{X}{R}, \frac{Y}{R})$.

Note that due to the iterative nature of the algorithm, doing n iterations in the CORDIC unit, the angle between the output of vectoring mode and the x-axis did not exactly equal to zero. This error is less than or equal to the n-th iteration as follows:

$$Z_{err, vectoring} \leq \tan^{-1}(2^{-(n-1)}) \tag{1}$$

Because of the error, the acquired angle from vectoring mode is slightly different from $\tan^{-1}(Y/X)$. To make the effect of vectoring mode error negligible in rotation mode, $(1, \sigma_n \times 2^{-n})$ is used instead of (1, 0) for the input of rotation mode, where the sign of vectoring mode error defines the value of σ_n which is equal to ± 1 . Hence, the rotation mode error would be as follows:

$$Z_{err, rotation} \leq \left| \tan^{-1}(2^{-(n-1)}) - \tan^{-1}(2^{-n}) \right| \tag{2}$$

It should be noted that $Z_{err, rotation}$ with n iterations, in this case, is equivalent to the $Z_{err, rotation}$ of rotation mode with the (1, 0) as input and n + 1 iterations. The proposed scheme could be implemented for processing

the $m \times m$ complex channel matrix where $\log_2 m \in N$. Since the proposed scheme is based on the MGS algorithm, it decomposes matrices column-wise. Therefore, it consists of m CORDIC blocks and m (non-CORDIC) projection blocks. The diagonal entries of both the upper triangular matrix and the normalized vector of the orthonormal matrix ($r_{jj} = \|v_j\|$, $q_j = v_j/r_{jj}$) are processed by the CORDIC blocks. Moreover, the off-diagonal of the triangular matrix and the updated middle vector ($r_{i,j} = q_j^H h_j$, $v_j = v_j - r_{i,j} q_j$) are processed by the non-CORDIC blocks.

Fig. 2, also shows the general CORDIC steps used in the QRD architecture of this paper. For a complex vector of size $m \times 1$, the norm and normalization process can be done in s-steps where $s = \log_2 2m$. Each of the CORDIC blocks consists of some vectoring and rotation sub-modules. It is evident in Fig. 2 that the number of vectoring mode sub-modules in each step is the half of the previous step. However, the number of rotation sub-modules in each step is two times of the previous step.

The scaling factor (K) used in the vectoring and rotation modes of Fig. 2, is achieved from the number of CORDIC iterations (n) as follows:

$$\begin{aligned} K &= \prod_{i=0}^{n-1} \cos(\tan^{-1}(2^{-i})) \\ &= \prod_{i=0}^{n-1} \frac{1}{\sqrt{1 + \tan^2(\tan^{-1}(2^{-i}))}} \\ &= \prod_{i=1}^{n-1} \frac{1}{\sqrt{1 + 2^{-2i}}} \cong 0.6073 \end{aligned} \tag{3}$$

It is clear that for $n \geq 7$, the scaling factor is approximately constant. On the other hand, the number of CORDIC iterations is chosen equal to 7 which is based on the existed tradeoff between the error and latency.

As it is shown in Fig. 2, in order to hardware overhead optimization, instead of scaling in each step of norm calculation, the k^s is applied at the end of s-th step. Moreover, in the first step of the normalization process, the k^s is applied instead of the scalar 1. To reduce the hardware complexity, the k^s is implemented using some

shift and add or subtraction operations.

Due to the iterative nature of the CORDIC algorithm used in the system design, the loop unrolling technique is investigated to reduce the iteration latency. In this method, by performing I iterations in each step, the number of iterations of the loop is reduced from n iterations to n/I . I is the coefficient of the loop-unrolling technique. This type of optimization can have a significant effect on some loops but depends on the type of loop and its content [11]. The design proposed in this paper is implemented and evaluated by applying the introduced technique with $I = 1, \dots, 7$.

Fig. 3 shows the flowchart of the CORDIC vectoring and rotation modes. It should be noted that the correction rotation step takes one clock and the remaining part is also done in one clock. However, the iterations in each clock of the remaining part depend on the loop unrolling factor (I). For example, if the loop unrolling factor is equal to 2, then there are 2 iterations per clock. Moreover, the processing of non-CORDIC block is done in 4 clocks. It is also worthy to note that Z and σ are shared between vectoring mode and rotation mode which is called Z -sharing and σ -sharing techniques.

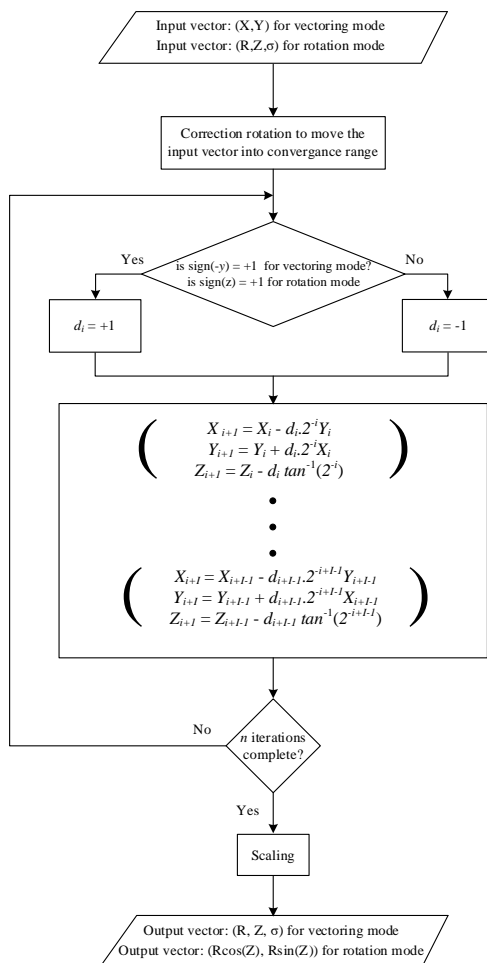


Fig. 3 Flowchart of the CORDIC vectoring and rotation modes.

4 Simulation Results

In this section, our aim is to investigate the simulation results of the proposed method. Due to the hardware considerations, the proposed design was coded using the Verilog hardware language and implemented on Xilinx Virtex 6 XC6VSX315T FPGA using ISE software as a fixed point with selected accuracy based on the results of MATLAB simulation. We also consider a 4×4 complex channel matrix. In order to numerically evaluate the QRD results, two criteria of Normalized Dissimilarity Error (NDE), and Orthogonality Error (OE) [12] are suggested as follows:

$$NDE = \frac{\mathbf{H} - \mathbf{Q}\mathbf{R}_F}{\mathbf{H}_F} \tag{4}$$

$$OE = \mathbf{I}_n - \mathbf{Q}^H \mathbf{Q}_F \tag{5}$$

In the ideal case, these criteria should be zero. A QRD algorithm must have high numerical performance, controllable complexity, and be easy to implement hardware. To evaluate the performance of hardware implementation, the criteria of latency, Throughput, frequency as well as used resources are evaluated.

In general, latency measures the time difference between the start and end of QRD processing on a matrix. Considering that in systems with parallel processing capability, it is possible to process several matrices simultaneously, two different definitions can be introduced as initial latency and steady-state latency. The time required to obtain the first output is the initial latency and the time required to obtain the next outputs after the first output is defined as the steady-state latency. Due to the specific frequency of the implemented architecture, the latency can be expressed in terms of the number of used clocks instead of the usual time units.

The stated throughput measure shows the number of matrices processed per second. Eq. (6) shows the relationship between latency and throughput:

$$Throughput = \frac{1}{\tau_s \times 1/f} = \frac{f}{\tau_s} \tag{6}$$

where τ_s and f represent steady-state latency (using number of clocks) and frequency respectively. To evaluate the numerical analysis, MATLAB simulations are provided. The average value of each quantity is obtained by averaging the corresponding quantity over 100 thousand realizations of channel matrices.

The modified Gram-Schmidt algorithm initially was implemented based on the CORDIC algorithm. The number of iterations used in each of the vectoring and rotation modes of the CORDIC algorithm has a direct effect on the value of investigated errors as well as the number of used bits. Fig. 4 illustrates the analysis results of the effect of different CORDIC iterations on the considered errors. For this purpose, the default

accuracy of MATLAB software is used for the experimental results of the proposed algorithm. Obviously, for a given number of iterations of the vectoring mode, by increasing the number of iterations of the rotating mode, the dissimilarity error decreases. Also, by increasing the number of iterations of the vectoring mode for the specified number of iterations of the rotating mode, the dissimilarity error is reduced.

The iteration value of 7 is selected for the CORDIC algorithm with a dissimilarity error of 1.29% and an orthogonality error of 3.65%. In the next step, the proposed algorithm is investigated for the iteration value of 7 vectoring modes and different iterations of CORDIC rotating mode with three different accuracies of floating point, fixed point, and σ -sharing technique. Fig. 5 illustrates the normalized dissimilarity and orthogonality errors, respectively. It should be noted that the fixed-point representation is depicted by (m, n)

in which m and n show the integer and fractional bits, respectively. Comparison of the results shows that the fixed-point representation with 11 fractional bits and the σ -sharing technique is very close to the floating-point representation with the default accuracy of MATLAB software. Fig. 6 is the results of the simulation for different accuracy of fixed points with an iteration value of 7 for vectoring mode and different iterations of CORDIC rotating mode. In the last step of numerical analysis, the scalability of the proposed method is investigated. For this purpose, errors are provided for different sizes of the channel matrix. Fig. 7 demonstrates the dissimilarity error and the orthogonality error, respectively. It is obvious that as the size of the channel matrix increases, the error values increase. To improve the error caused by increasing the channel matrix size, the number of fractional bits can be increased.

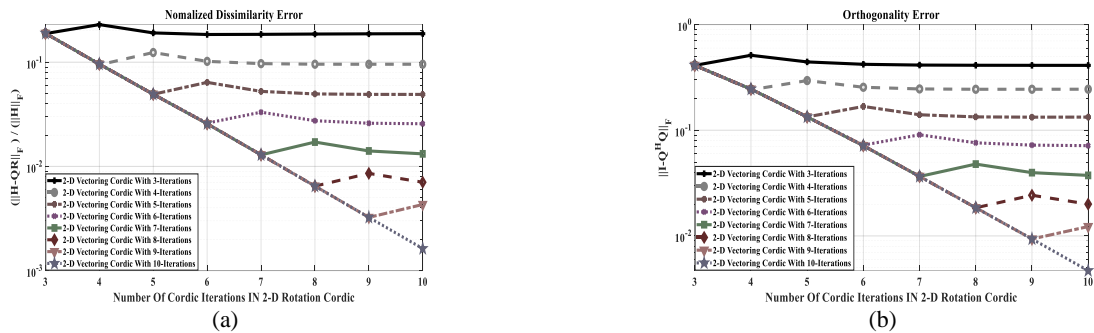


Fig. 4 Different errors versus CORDIC rotation mode iterations for different CORDIC iterations of vectoring mode; a) Normalized dissimilarity error and b) Orthogonality error.

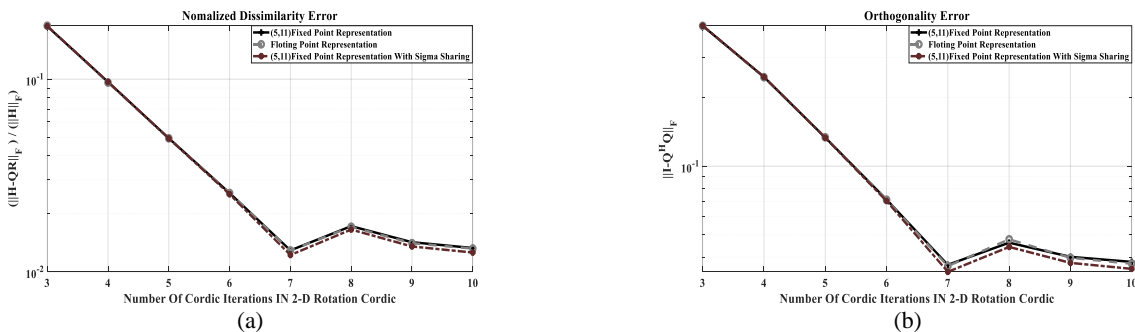


Fig. 5 Different errors versus CORDIC rotation mode iterations for different representations; a) Normalized dissimilarity error and b) Orthogonality error.

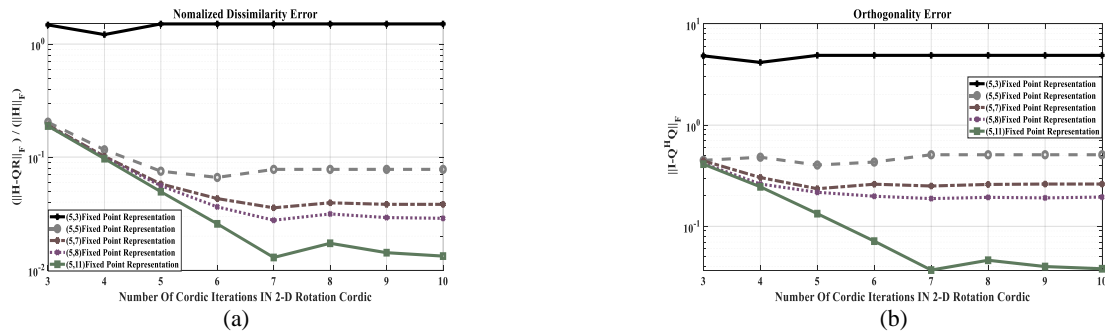


Fig. 6 Different errors versus CORDIC rotation mode iterations for different accuracy of fixed-point representation; a) Normalized dissimilarity error and b) Orthogonality error.

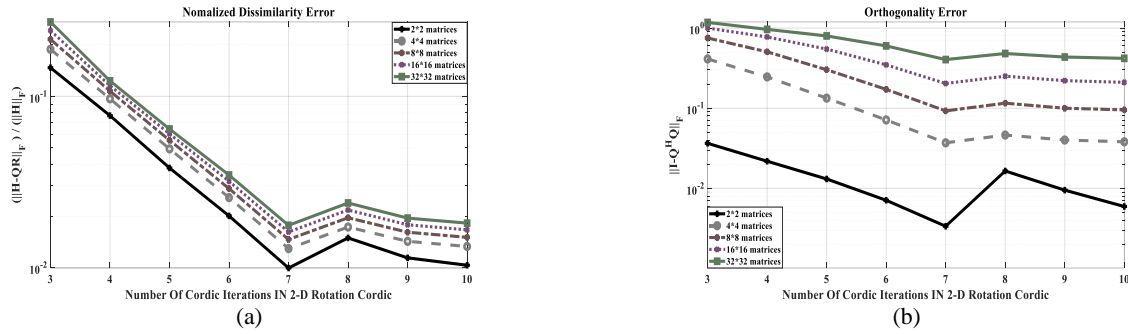


Fig. 7 Different errors versus CORDIC rotation mode iterations for different channel matrix sizes; a) Normalized dissimilarity error and b) Orthogonality error.

Table 3 Implementation results for different unrolling factors.

Unrolling factor (<i>I</i>)	1	2	3	4	5	6	7
CORDIC unit frequency	360.776	197.936	136.385	103.980	83.178	69.848	60.200
QRD unit frequency	159.586	159.586	136.385	103.980	83.178	69.848	60.200
Initial processing clocks	204	132	108	84	84	84	60
Initial latency [ns]	1278	827	792	808	101	1203	997
Steady state processing clocks	48	30	24	18	18	18	12
Steady state latency [ns]	301	188	176	173	216	258	199
Throughput [MQRD/s]	3.325	5.319	5.683	5.777	4.621	3.880	5.017
Slice registers	15740	9734	7874	6173	6173	5858	4061
Slice LUTs	18302	17819	17729	17639	17639	17639	17459
LUT-FF pairs	14612	8666	6923	5123	5123	4829	1835
BUFG/BUFGCTRLs	16	16	16	16	16	16	32
DSP48E1S	240	240	240	240	240	240	240

Table 4 Comparative results with other FPGA Implementations.

Work	[13]	[14]	[15]	Proposed
QRD dimension	4x4	4x4	4x4	4x4
Data type	Fixed point	Fixed point	Fixed point	Fixed point
Algorithm	GR	GR	GR	MGs
CORDIC iteration	-	8	10	7
Platform	Virtex 5	Virtex 6	Virtex 5	Virtex 6
Word length	16	16	16	16
Clock [MHz]	246	128	254	103.98
Latency (Clock cycle)	180	137	52	18
Latency [ns]	732	1070	200	173
Throughput [MQRD/s]	1.36	0.934	31.7	5.78

After numerical analysis, the QRD architecture is examined in terms of hardware for different modes such as not optimized case (technique with unrolling factor 1) and optimized cases with unrolling factor 2 to 7, and the important features of the implementation are examined. Hardware (such as frequency, latency, throughput, and used hardware resources) are evaluated. All the examined cases were coded using the Verilog hardware language and implemented on Virtex 6 using ISE software.

A comparison of frequency, initial latency, steady-state latency, and throughput for different unrolling factors is given in Table 3. The frequency of the non-CORDIC module in all modes is constant and equal to 227.376 MHz. In the non-optimized case (technique with a factor of 1), the CORDIC unit has the highest frequency value of 360.776 MHz. Therefore, in this case, the non-CORDIC module is the frequency limiter. In applications where the QRD unit does not limit the

frequency of the main system, the appropriate architecture can be used depending on the type of application and the required frequency. From the results, it is clear that the lowest steady-state latency and the highest throughput are related to the loop unrolling technique with a factor of 4. The used hardware resources in the structure of each of the different cases are also described in Table 3. It is also clear from the comparison of the results that the number of used registers, LUTs and LUT-FFs decreases with increasing unrolling factor. Since the hardware resources in the structure of FPGAs (such as LUTs) have more than two inputs, it is possible to implement a large number of functions. Due to the recursive nature of the used CORDIC algorithm in the QRD architecture, the number of hardware resources decreased in each case.

Finally, to evaluate the performance of the proposed method, Table 4 compares the results with some prior works [13-15]. In order to fair comparison, it has been

tried to compare our proposed method with works whose implementation parameters are similar to ours. As it can be seen, the proposed method is superior in terms of latency and throughput compared to the previous works [13, 14]. The proposed approach in [15], has high throughput compared to this work. It should be noted that the high throughput of [15] is achieved due to highly pipelined architecture.

5 Conclusion

To detect the MIMO signal, different aspects of the implementation of the CORDIC-based modified Gram-Schmidt algorithm for QR decomposition such as latency, throughput, and hardware resources are optimized. The proposed approach was implemented on Xilinx Virtex 6 XC6V SX315T FPGA using ISE software as a fixed point with selected accuracy based on the results of MATLAB simulation. It is shown that the loop unrolling technique with a factor of 4 has the highest throughput (processing speed) of 5.777 MQRD/s and the lowest latency of 173 ns, which is 42.52% and 73.74% better than the non-optimized case, respectively. Due to the recursive nature of the CORDIC algorithm used in the QRD architecture, increasing the unrolling factor resulted in decreasing the used hardware resources. On the other hand, the results of the numerical analysis show that the proposed method is scalable for different sizes of channel matrices that can be a power of 2. Due to the scalability of the proposed architecture, it can be used to detect signals in massive MIMO systems as well as multi-user MIMO systems.

Intellectual Property

The authors confirm that they have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property.

Funding

No funding was received for this work.

CRedit Authorship Contribution Statement

F. Asghariyehlou: Idea & conceptualization, Research & investigation, Software and simulation, Original draft preparation. **J. Javidan:** Idea & conceptualization, Project administration, Supervision, Revise & editing.

Declaration of Competing Interest

The authors hereby confirm that the submitted manuscript is an original work and has not been published so far, is not under consideration for publication by any other journal and will not be submitted to any other journal until the decision will be

made by this journal. All authors have approved the manuscript and agree with its submission to "Iranian Journal of Electrical and Electronic Engineering".

References

- [1] A. Goldsmith, *Wireless communications*. Cambridge University Press, 2005.
- [2] R. W. Heath Jr and A. Lozano, *Foundations of MIMO communication*. Cambridge University Press, 2018.
- [3] L. Chen, Z. Xing, Y. Li, and S. Qiu, "Efficient MIMO preprocessor with sorting-relaxed QR decomposition and modified greedy LLL algorithm," *IEEE Access*, Vol. 8, pp. 54085–54099, 2020.
- [4] R. C. H. Chang, C. H. Lin, K. H. Lin, C. L. Huang, and F. C. Chen, "Iterative QR decomposition architecture using the modified Gram Schmidt algorithm for MIMO systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 57, No. 5, pp. 1095–1102, 2010.
- [5] P. K. Meher, J. Valls, T. Juang, K. Sridharan, and K. Maharatna, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 56, No. 9, pp. 1893–1907, 2009.
- [6] J. Rudagi and S. Subbaraman, "Comparative analysis of radix-2, radix-4, radix-8 CORDIC processors," in *International Conference on Inventive Computing and Informatics (ICICI)*, pp. 378–382, 2017.
- [7] D. Timmermann, H. Hahn, and B. J. Hosticka, "Low latency time CORDIC algorithms," *IEEE Computer Architecture Letters*, Vol. 41, No. 8, pp. 1010–1015, 1992.
- [8] R. Kunemund, H. Soldner, S. Wohlleben, and T. Noll, "CORDIC processor with carry-save architecture," in *16th European Solid-State Circuits Conference (ESSCIRC'90)*, pp. 193–196, 1990.
- [9] T. K. Rodrigues and E. E. Swartzlander, "Adaptive CORDIC: Using parallel angle recoding to accelerate CORDIC rotations," in *Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 323–327, 2006.
- [10] A. Awasthi, R. Guttal, N. Al-Dhahir, and P. T. Balsara, "Complex QR decomposition using fast plane rotations for MIMO applications," *IEEE Communications Letters*, Vol. 18, No. 10, pp. 1743–1746, 2014.
- [11] P. Desai, "FPGA implementation of QR decomposition algorithms using high-level synthesis on Zynq SoC," *M.Sc. Thesis, Illinois Institute of Technology*, 2017.

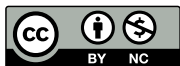
- [12] S. Aslan, E. Oruklu, and J. Saniie, "Realization of area efficient QR factorization using unified division, square root, and inverse square root hardware," in *IEEE International Conference on Electro/Information Technology*, pp. 245–250, Jun. 2009.
- [13] S. Aslan, S. Niu, and J. Saniie, "FPGA implementation of fast QR decomposition based on Givens rotation," in *IEEE 55th International Midwest Symposium on Circuits and Systems*, pp. 470–473, 2012.
- [14] W. Zhao, J. Lin, and S. C. Chan, "Throughput/area efficient FPGA implementation of QR decomposition for MIMO systems," in *IEEE International Conference on Digital Signal Processing (DSP)*, pp. 522–526, Oct. 2016.
- [15] S. D. Munoz and J. Hormigo, "High-throughput FPGA implementation of QR decomposition" *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 62, No. 9, pp. 861–865, 2015.



F. Asghariyehlou was born in Ardabil, Iran. She received the B.Sc. and M.Sc. degrees both in electrical engineering from the University of Mohaghegh Ardabili (UMA), Ardabil, Iran, in 2017 and 2020, respectively. She is currently a Ph.D. Student in Electrical Engineering at the University of Tabriz, Tabriz, Iran. Her research interest includes digital processing, FPGA Implementation, Optical Electronics.



J. Javidan received the B.Sc., M.Sc. and Ph.D. degree in Electrical Engineering all from Sharif University of Technology (SUT), Tehran, Iran, in 2001, 2003, and 2009, respectively. He received a one-year grant from HKUST University in Hong Kong for doing his research as an RA and Visiting Student in 2009. Currently, he is an Associate Professor in the Technical Engineering Department of the University of Mohaghegh Ardabili (UMA), Ardabil, Iran. He has published several technical journals and conference papers in the area of power electronics and mixed-signal integrated circuit design. His main research interests are Power Electronic, data converter, RF IC design, and VLSI.



© 2022 by the authors. Licensee IUST, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0) license (<https://creativecommons.org/licenses/by-nc/4.0/>).