A High-Speed QR Decomposition Processor for Carrier-Aggregated LTE-A Downlink Systems

Gangarajaiah, Rakesh; Liu, Liang; Stala, Michal; Nilsson, Peter; Edfors, Ove

2013

A High-Speed QR Decomposition Processor for Carrier-Aggregated LTE-A Downlink Systems

Rakesh Gangarajaiah, Liang Liu, Michal Stala, Peter Nilsson, and Ove Edfors Department of Electrical and Information Technology, Lund University, Sweden
Email: {rakesh.gangarajaiah,liang.liu,michal.stala,peter.nilsson,ove.edfors}@eit.lth.se

***Abstract*—This paper presents a high-speed QR decomposition (QRD) processor targeting the carrier-aggregated 4×4 Long Term Evolution-Advanced (LTE-A) receiver. The processor provides robustness in spatially correlated channels with reduced complexity by using modifications to the Householder transform, such as decomposing-target redefinition and matrix real-valued decomposition. In terms of hardware design, we extensively explore flexibilities in systolic architectures using a high-level synthesis tool to achieve area-power efficiency. In a 65 nm CMOS technology, the processor occupies a core area of 0.77 mm^2 and produces 72 MQRD per second, the highest reported throughput. The power consumed in the proposed processor is 127 mW.**

I. Introduction

The requirement of high speed wireless connections over limited spectrum has made the use of Multiple-Input Multiple-Output (MIMO) technique a necessity. To fully utilize the potential of MIMO systems, sophisticated signal processing is required at the receiver. QR decomposition (QRD) is one of the key operations used to correctly decode multiple streams of data affected by noise and interference [1].

Several standards have been introduced to meet requirements of high data rate applications. For example, the 3GPP Long Term Evolution-Advanced (LTE-A) delivers rates of over 1 Gbps using techniques such as enhanced MIMO and Carrier Aggregation (CA). This poses critical design challenges on the implementation of baseband processing algorithms. In one of the extreme use cases of LTE-A, where five frequency bands are aggregated into a 100MHz data bandwidth, the QRD processor needs to compute up to 72 MQRD/s under fluctuating channel conditions. Insufficient antenna spacing in hand-held devices creates further complications such as spatial correlation resulting in ill conditioned channel matrix $\mathbf{H}$ in the baseband. Numerical stability of algorithms working on such matrices is critical and previous studies have proved that the fixed point implementation of the Householder Transform (HT) is more numerically stable than the Gram-Schmidt (GS) method [2]. Moreover, HT works with columns instead of scalar elements, and thus is better for data-level parallelism. However, previous studies have suggested that the computational complexity of HT is very high, preventing it from being used in hardware implementation [3].

In this work, we leverage the high numerical stability of the HT to produce accurate QRD even in correlated MIMO channels. Two techniques are used to reduce the complexity while achieving high throughput with reasonable hardware resources. First, we redefine the QRD target based on the requirement of a tree-search symbol detector and avoid unnecessary matrix multiplications. Later, methods to exploit the symmetry and orthogonality properties in the Real Valued Decomposition (RVD) of $\mathbf{H}$ to further reduce complexity are detailed. We develop a scalable systolic VLSI architecture to implement the modified HT and utilize Calypto's Catapult tool to obtain optimized designs. This high-level synthesis tool translates C++ code into Register Transfer Level (RTL) and enables the designer to explore the effects of word widths, folding and pipelining against area and power consumption. Post-synthesis simulation results using 65 nm CMOS technology show that the proposed QRD processor achieves 72 MQRD/s, the highest reported throughput, with a gate count of 378 k gates.

II. Background

Consider a MIMO system with N transmitter (Tx) and N receiver (Rx) antennas. If the transmit vector is represented as $\mathbf{x} = [x_1, x_2, ..., x_N]^T$, the receive vector as $\mathbf{y} = [y_1, y_2, ..., y_N]^T$ with a channel $\mathbf{H} \in \mathbb{C}^{N \times N}$, then the system affected by random noise $\mathbf{n}$, can be described by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}. \tag{1}$$

To achieve low Bit error rate (BER) the MIMO symbol detector has to minimize the error $\| \mathbf{y} - \mathbf{H}\tilde{\mathbf{x}} \|_2$, where $\tilde{\mathbf{x}}$ is the estimate of the transmit vector. Efficient symbol detectors require $\mathbf{H}$ to be decomposed into the product $\mathbf{QR}$, where $\mathbf{Q}$ is a unitary matrix and $\mathbf{R}$ is an upper triangular matrix [1]. The module which decomposes $\mathbf{H}$ into this product is called a QRD processor. These processors can be classified into two broad categories, one which works by rotating submatrices like the Given's rotation (GR) method and the other which works on columns, namely the GS method and the HT.

The conventional HT converts $\mathbf{H}$ into a product of N unitary $\mathbf{Q}$ matrices and an upper triangular $\mathbf{R}$ as shown in

$$\mathbf{H} = \mathbf{Q_1}\mathbf{Q_2}...\mathbf{Q_{N-1}}\mathbf{Q_N}\mathbf{R}, \tag{2}$$

where each of the $\mathbf{Q}_i$ matrices are of the form

$$\mathbf{Q}_i = \left(\mathbf{I} - \frac{\mathbf{v}_i \mathbf{v}_i^*}{\mathbf{v}_i^* \mathbf{z}_i} \right) \tag{3}$$

and $\mathbf{z}_i$ is the vector to be transformed with $\mathbf{v}_i$ being the difference vector from $\mathbf{z}_i$ to one of the columns of the identity matrix $\mathbf{I}$ [4]. Unfortunately the method of multiplying all the components $\mathbf{Q_1}\mathbf{Q_2} \cdots \mathbf{Q_N}$ to produce $\mathbf{Q}$ would lead to high computational complexity in the order of N^4 and a straight forward implementation would lead to unnecessary high effort.

III. Proposed QR decomposition

In this section we present techniques to reduce the complexity of the HT. First we look at the modified representation of the linear system. Then we discuss the RVD and detail the methods of exploiting symmetry and orthogonality to reduce complexity. Later we discuss the gains obtained by implementing HT using these properties.

A. Modified linear system

Using the QRD of $\mathbf{H}$, the system in (1) can be written as

$$\mathbf{Q}^*\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{Q}^*\mathbf{n}. \tag{4}$$

As mentioned before, tree search based detectors accept $\mathbf{QR}$ instead of $\mathbf{H}$ and work by solving equations of the form (4), hence the task of the QRD processor working with a tree based detector can be relaxed to that of producing $\mathbf{Q}^*\mathbf{y}$ and $\mathbf{R}\mathbf{x}$. One of the requirements for (4) to hold good is that the error $\| (\mathbf{Q}^*\mathbf{Q}) - \mathbf{I} \|_2$ is minimal, or in other words, $\mathbf{Q}$ is highly unitary. Since the $\mathbf{Q}$ produced by the HT is of the form shown in (2) and using the property that $\mathbf{Q}$ is unitary, the product $\mathbf{Q}^*\mathbf{y}$ can be rewritten as

$$\mathbf{Q}^*\mathbf{y} = \mathbf{Q_N}\mathbf{Q_{N-1}}...\mathbf{Q_1}\mathbf{y}. \tag{5}$$

Using the structure of the component $\mathbf{Q}_i$ matrices from (3), the above equation can be rewritten as

$$\mathbf{Q}^*\mathbf{y} = \mathbf{Q_N}\mathbf{Q_{N-1}}...\left(\mathbf{y} - \frac{\mathbf{v_1}\left(\mathbf{v_1^*}\mathbf{y}\right)}{\mathbf{v_1^*}\mathbf{z_1}}\right). \tag{6}$$

By calculating $\mathbf{Q}_i\mathbf{y}$ at each stage, the problem of computing $\mathbf{Q}^*\mathbf{y}$ by N full rank matrix multiplications followed by a matrix-vector computation, reduces to a vector-vector multiplication at each stage of the transform. The fact that the HT is inherently an iterative process computing $\mathbf{Q_1}$ before $\mathbf{Q_2}$ enables us to produce a highly pipelined and hardware efficient QRD processor. The complexity of computing $\mathbf{Q}^*\mathbf{y}$ is in the order of N^3 as compared to N^4 for direct implementation, which results in a huge reduction in computational cost, especially as N, the number of antennas, increases. Once the product $\mathbf{Q_1^*}\mathbf{y}$ is computed, the $\mathbf{v_1}$ vector can be discarded or, in hardware implementation, the same registers can be reused to store and process the ensuing $\mathbf{v}_i$ vectors, resulting in reduced storage area.

B. Complexity reduction due to Real Valued decomposition

Tree search based detectors prefer RVD due to the easy enumeration of possible child nodes [5]. Any matrix in $\mathbb{C}^{N\times N}$ can be represented by an equivalent matrix in $\mathbb{R}^{2N\times 2N}$. One of the methods to do this is to represent each complex valued entry by an equivalent 2×2 real valued entry as shown in Fig. 1. It has to be noted that each of the 2×2 submatrices in $\mathbb{R}^{2\times 2}$ are not only orthogonal but also that the columns of the transformed matrix are pairwise orthogonal, as highlighted in Fig. 1. Applying the HT on the real valued matrix results in reducing the first column into a real valued entry $\boldsymbol{\alpha}_1$, which is the length of the first column, along with modifying all the other columns as indicated in Fig. 1. Due to the property

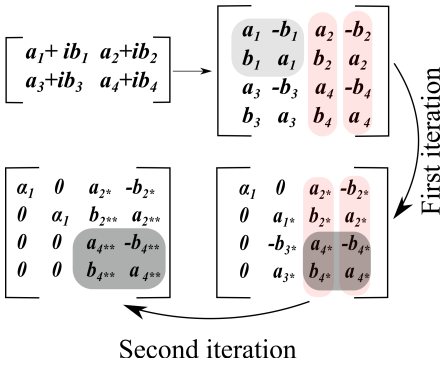

Fig. 1: Householder Real valued decomposition

of the transform, the first element in the second column is also reduced to zero. It should be noted that, since the HT is equivalent to multiplication by an orthonormal matrix, the orthogonal properties of the columns and the 2×2 submatrices remain unchanged. The second iteration of the HT only modifies the smaller 3×3 submatrix in the example shown above and changes the first element in the second column into a real entry representing the length of the second column. By construction, the second column is also the same length as the first column of the original matrix. Hence the second iteration also produces the real element $\boldsymbol{\alpha}_1$ which does not need to be computed again. Utilizing these properties, only half the number of columns in the real valued representation of the matrix need to be transformed.

C. Algorithm analysis

The algorithm to perform the QRD of a real matrix $\mathbf{H}_R$ in $\mathbb{R}^{2N\times 2N}$ using the HT is shown in Table I along with the number of real domain operations required. The first two columns of $\mathbf{H}_R$ are essentially the same data, repeated in a systematic way and hence the Householder vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ corresponding to the first two columns can be computed in parallel. These parallel computations enable two iterations of the HT can be performed in one run, thereby enabling $2N$ columns to be processed in N runs.

1) Operation count analysis: Using these modifications to the algorithm, the number of multiplications required for one QRD using the modified HT is in the order of $\frac{8N^3}{3}$ whereas the GS method requires more than $4N^3$ operations while the direct HT implementation requires N^4 operations [5]. The total number of operations including square roots, divisions and additions required to implement the transform compared to the GS method and the direct HT for different matrix sizes is shown in Fig. 2. It can be seen that the computational effort required to perform QRD using the proposed HT is not only lower than the direct HT method but also significantly lower than the corresponding GS method for matrices with large N.

2) Stability analysis in Correlated channels: Insufficient diversity in the channel or small antenna spacing creates correlated channels, resulting in a nearly rank deficient $\mathbf{H}$. The ability of the QRD processor to orthonormalize a channel under such conditions determines the performance of the whole MIMO system. Fixed point simulations with different

TABLE I: Complexity Analysis of RVD

Algorithm	Add.	Mul.
for $i = 2N : -2 : 0$		
$\mathbf{x1} = \mathbf{H_{i:1,i}}$		
$\alpha = sign(x_{i,i}) \parallel \mathbf{x_1} \parallel_2$	$i-1$	i
$\mathbf{v_1} = \alpha\mathbf{e_i} + \mathbf{x1_{i:1,i}}$	1	
$\parallel \mathbf{v_1^* v_1} \parallel^2 = \alpha^2 + \alpha x1_{i,i}$	1	1
$\boldsymbol{\beta_1} = \frac{\mathbf{v_1}}{\parallel \mathbf{v_1^* v_1} \parallel^2}$		
$\mathbf{\Gamma}_1 = \boldsymbol{\beta}_1(\mathbf{v_1}^* \mathbf{H_{i:1,i+2:1}})$	$(i-1)\left(\frac{i-2}{2}\right)$	$(i+i)\left(\frac{i-2}{2}\right)$
$\mathbf{H1} = \mathbf{H_{i:1,i+2:1}} - \mathbf{\Gamma}_1$	$i\left(\frac{i-2}{2}\right)$	
$G = \frac{-x1(2)}{\alpha + x1(1)}$		1
$\mathbf{x2} = \mathbf{H_{i:1,i+1}}$		
$\mathbf{v2} = \mathbf{x2} - \mathbf{x1}G + \alpha\mathbf{e_{i+1}}$	i	$i-1$
$\boldsymbol{\beta_2} = \frac{\mathbf{v_2}}{\parallel \mathbf{v_2^* v_2} \parallel^2} = \frac{\mathbf{v_2}}{\parallel \mathbf{v_1^* v_1} \parallel^2}$		
$\mathbf{\Gamma}_2 = \boldsymbol{\beta}_2(\mathbf{v_2}^* \mathbf{H1_{i+1:1,i+2:1}})$	$(i-2)\left(\frac{i-2}{2}\right)$	$2(i-1)\left(\frac{i-2}{2}\right)$
$\mathbf{H} = \mathbf{H1_{i+1:1,i+2:1}} - \mathbf{\Gamma}_2$	$(i-1)\left(\frac{i-2}{2}\right)$	
$\mathbf{y} - 2\boldsymbol{\beta}_1 \mathbf{v_1^* y}$	$2i-1$	$2i$
$\mathbf{y} - 2\boldsymbol{\beta}_2 \mathbf{v_2^* y}$	$2i-3$	$2(i-1)$
end for		
Total for each iteration	$2i^2+1$	$2i^2+i+1$
Corrected Total for each iteration	$8i^2+1$	$8i^2+2i+1$

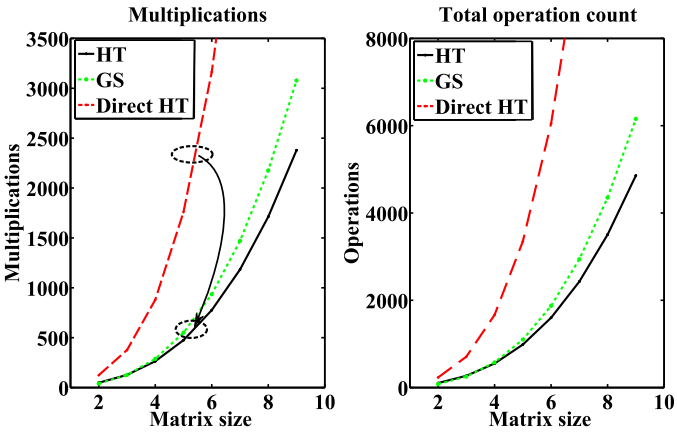

Fig. 2: Operation count for HT and GS

channel models show that 13 bits of normalized channel data is sufficient for the QRD processor to obtain near floating point performance in uncorrelated channel conditions. The Mean square error (MSE) in producing unitary **Q** using the HT and GS algorithms implemented using 13 bits for 4×4 complex valued **H** with different condition numbers is shown in Table II. The results show that HT is significantly better at producing orthonormalized **Q**, especially as the condition number of the matrix increases. Effects of channel correlation on the BER using both floating point and fixed point QRD processors along with the setup used for the experiment is

TABLE II: Gain in inversion accuracy

Condition number (**H**)	10	200	400	600	800
MSE of HT	0.0039	0.0040	0.0040	0.0041	0.0041
MSE of GS	0.0078	0.5482	1.4291	2.5380	3.7312

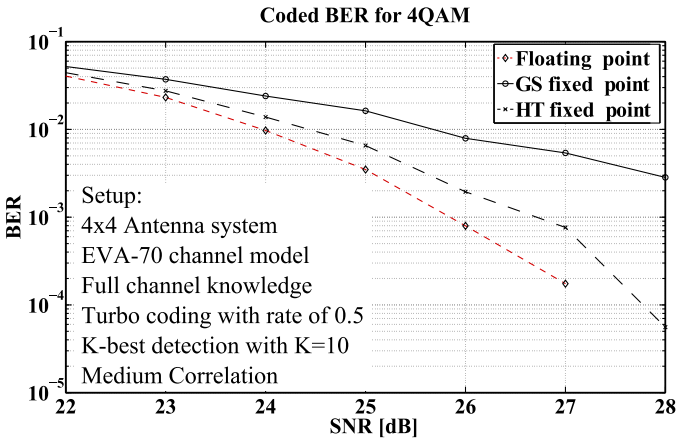

Fig. 3: BER curves for 4QAM in correlated channel

shown in Fig. 3. Due to degradation in BER performance in correlated channels, 4QAM is used as modulation alphabet along with coding to get an acceptable performance. It can be seen that the performance of the HT is within 1 dB of the full floating point QRD, whereas the GS method fails to achieve acceptable BER even with high signal to noise ratio.

IV. Hardware Implementation

In this section, the basic architecture of the HT is presented. Later the methodology used to obtain different implementations of the QRD processor using the high level synthesis tool is discussed. Finally the hardware synthesis and power results are presented and a comparison is done with previously published QRD processors.

A. Architecture

Fig. 4 shows a high level architecture of the HT. The transform contains multiple arithmetic units such as multipliers, adders, square root, and division units represented by **AU** in the architecture. Since the algorithm is sequential, the systolic array architecture is well suited for hardware implementation. Furthermore, the operations performed in each stage are essentially the same as explained in Table I and techniques such as folding and pipelining can be used to reduce area and increase throughput. Folding enables reuse of arithmetic units, reducing area, but increases power consumption as the circuit needs to run at a higher frequency to meet fixed throughput requirements. The number of arithmetic units required are reduced at later stages of the QRD as the HT operates on lower number of elements in each successive column. Choosing an optimal number of multipliers and other combinational units is not an easy task and a flexible solution which enables Power-Area trade-offs for different technologies and throughput requirements is needed.

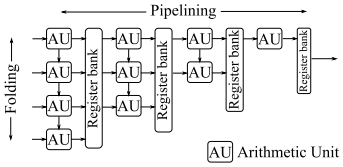

Fig. 4: High level architecture

B. Methodology

Coding of the algorithm is done in C++ and fixed point libraries are used to translate the code into RTL using Catapult. Constraints such as synthesis technology, clock frequency, area and latency requirements are provided to the tool. The tool finds a feasible schedule to implement the algorithm and optimizes the design to find the best combination of hardware resources to meet the constraints. Constraints on clock frequency, area and pipelining can be used to explore the design space to find an optimal solution to meet the design goal. The design is then synthesized using Design Compiler and power estimates are obtained using Primetime.

C. Experimental setup

Two designs of a 4×4 MIMO system were considered for implementing the QRD using the HT. The first design is synthesized to produce a throughput of 15 MQRD/s, which would correspond to an LTE-A system running at 20 MHz bandwidth without CA and another design to produce 72 MQRD/s, which corresponds to an LTE-A system running with a five band CA. The designs are taken through the flow described in the previous section and the resulting normalized values of Power, Area, and their product PA for different folding factors is shown in Fig. 5. The absolute values for these parameters can be obtained by using Table III. The design with a throughput requirement of 15 MQRD/s is synthesized for different frequencies ranging from 15 MHz to 135 MHz. Area reduces as folding increases since multipliers and other combinational units are reused, but the power consumption increases due to higher operating frequency. Similar trends are seen for the design producing 72 MQRD/s.

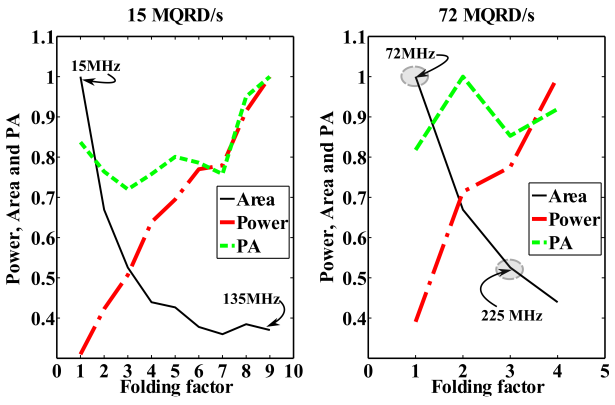

Fig. 5: PA analysis for different designs

D. Results

Two designs capable of producing a throughput of 72 MQRD/s, highlighted in Fig. 5 are presented in Table III along with previously published designs. One of the designs is synthesized from the fully unfolded RTL and the other one is obtained from a version with a folding factor of three. The power numbers are obtained by post synthesis simulations. The results are normalized to 65nm technology and a power supply of 1 Volt. The Normalized Hardware Efficiency (NHE) is defined as the ratio of normalized throughput over the gate count [5]. The energy consumption is calculated as the ratio of normalized power over the throughput. The current work has the highest reported throughput while consuming lower energy than the design presented in [5] in the fully unfolded configuration.

TABLE III: Comparison with previous works

Items	Patel [6]	Huang [5]	Miyaoka [7]	This Work	
Matrix type	4x4 complex	8x8 Real	4x4 complex	8x8 Real Unfolded	8x8 Real Folded
Technology	$130\,nm$	$180\,nm$	$90\,nm$	$65\,nm$	$65\,nm$
Max Freq	$270\,MHz$	$100\,MHz$	$300\,MHz$	$72\,MHz$	$225\,MHz$
Gate count	$36k$	$152\,k$	$334\,k$	$378\,k$	$264\,k$
Throughput [MQRD/s]	6.75	25	50	72	72
Normalized Throughput	13.5	69	69	72	72
N.H.E.	375	454	150	190	272
Normalized Power	–	$60\,mW$	–	$127\,mW$	$252\,mW$
Energy/MQRD	–	$2.4\,mJ$	–	$1.7\,mJ$	$3.36\,mJ$

V. Conclusion

In this paper, modifications to the standard Householder Transform (HT) are proposed which enable QRD to be performed with lower computational complexity than Gram-Schmidt (GS) method. The proposed design is able to meet the requirements of a full CA LTE-A system producing a throughput of 72 MQRD/s. Simulation results have also shown that using the HT instead of the GS method results in performance gain of over 2 dB at Signal to Noise Ratio (SNR) levels of around 25 dB in correlated channels. RTL implementation results shows that the high level synthesis tool is very effective in evaluating designs for Area-Power trade-offs. The implemented design has the highest reported throughput, while consuming comparable energy and area.

Acknowledgment

This work is a part of the DARE project and the authors would like to thank Lund University and the funding organization, Stiftelsen för Strategisk Forskning.

References

[1] M. Wenk, M. Zellweger, A. Burg, N. Felber, and W. Fichtner, "K-best MIMO detection VLSI architectures achieving up to 424 Mbps," in *Proc. IEEE Int. Symp. Circuits Syst., ISCAS 2006.*

[2] G. H. Golub and C. F. Van Loan, *Matrix computations (3rd ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 1996.

[3] Y. T. Hwang and W. D. Chen, "Design and implementation of a high-throughput fully parallel complex-valued QR factorisation chips," *IET Circuits, Devices Syst.*, vol. 5, no. 5, pp. 424–432, 2011.

[4] K.-L. Chung and W.-M. Yan, "The complex Householder transform," *IEEE Trans. Signal Process.*, vol. 45, no. 9, sep 1997.

[5] Z.-Y. Huang and P.-Y. Tsai, "Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems," *IEEE Trans. Circuits Syst. I*, vol. 58, no. 10, pp. 2531–2542, oct. 2011.

[6] D. Patel, M. Shabany, and P. Gulak, "A low-complexity high-speed QR decomposition implementation for MIMO receivers," in *Proc. IEEE Int. Symp. Circuits Syst., ISCAS 2009.*

[7] Y. Miyaoka, Y. Nagao, M. Kurosaki, and H. Ochi, "Sorted QR decomposition for high-speed MMSE MIMO detection based wireless communication systems," in *Proc. IEEE Int. Symp. Circuits Syst., ISCAS 2012.*