VLSI Implementation of a High-Speed Iterative Sorted MMSE QR Decomposition

P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber and W. Fichtner
Integrated Systems Laboratory, ETH Zurich, Switzerland
{luethi,apburg,haene,perels,felber,fw}@iis.ee.ethz.ch

***Abstract*— The QR decomposition is an important, but often underestimated prerequisite for pseudo- or non-linear detection methods such as successive interference cancellation or sphere decoding for multiple-input multiple-output (MIMO) systems. The ability of concurrent iterative sorting during the QR decomposition introduces a moderate overall latency, but provides the base for an improved layered stream decoding. This paper describes the architecture and results of the first VLSI implementation of an iterative sorted QR decomposition preprocessor for MIMO receivers. The presented architecture performs MIMO channel preprocessing using Givens rotations in order to compute the minimum mean squared error QR decomposition.**

I. Introduction

Multiple-input multiple-output (MIMO) is considered to be one of the key technologies for enabling next-generation high-speed wireless communication. MIMO systems employ multiple antennas at both the transmitter *and* the receiver in order to increase data rate and link reliability. In spatial multiplexing mode, MIMO systems reach higher throughputs by transmitting multiple data streams in parallel in the same frequency band without additional expense in bandwidth or transmit power. Consequently, numerous upcoming wireless communication standards such as IEEE 802.11n or IEEE 802.16 will employ MIMO technology. Unfortunately, the considerable throughput improvements enabled by MIMO systems entail a significant increase in signal processing complexity, especially for separating the multiple, parallel data streams.

QR decomposition (QRD) is one of the key algorithms employed in MIMO systems, since numerous MIMO detection methods require QR decomposition of the channel matrix as starting point. The application of QRD ranges from linear MIMO detection methods to ordered successive interference cancellation (OSIC) and to tree-search algorithms [1], [2] with close-to maximum likelihood bit error rate (BER) performance. Sorted QR decomposition (SQRD) in conjunction with successive interference cancellation (SIC) provides a considerable improved BER performance compared to unsorted SIC, while consuming only little additional hardware resources.

Contribution: In this work, we present - to the best of our knowledge - the first VLSI implementation of a 4×4 matrix preprocessor performing iterative sorted minimum mean squared error (MMSE) QR decomposition for OSIC or for tree-search algorithms. The described VLSI architecture incorporates optimized fixed-point arithmetic and shows how a combination of CORDIC circuits and complex-valued multipliers allows to achieve a very high throughput with low silicon area. The implemented reference design in a 0.25 μm technology processes 1.56 million complex-valued 4×4 channel matrices per second maintaining close-to floating-point BER performance for OSIC detection up to a signal to noise ratio (SNR) of 40 dB.

Outline: The remainder of this section introduces the system model and motivates the development of a high-speed QR decomposition circuit. Sec. II introduces the sorted MMSE QR decomposition algorithm under consideration. Sec. III describes the high-level VLSI architecture of the SQRD unit and the corresponding micro-architectural choices for the implementation of the arithmetic units. Complexity/performance trade-offs associated with different sorting strategies and fixed-point considerations are discussed in Sec. IV and implementation results are presented in Sec. V.

Notation: Bold uppercase and lowercase letters represent matrices and column vectors, respectively. The ith column vector of $\mathbf{H}$ is denoted by $\mathbf{h}_i$, and $\mathbf{H}_{i,j}$ stands for the element in row i and column j of $\mathbf{H}$.

A. System Model

The system under consideration is a MIMO system with M_T transmit and M_R receive antennas. The matrix $\mathbf{H}$ describes the MIMO channel, the $M_T \times 1$ transmit signal vector is denoted by $\mathbf{s} = [s_1 \dots s_{M_T}]^T$, and the vector $\mathbf{n}$ represents the additive zero-mean i.i.d. Gaussian noise with variance σ_n^2 per complex dimension. The transmitted vector symbol is normalized such that $E\{ss^H\} = \mathbf{I}_{M_T}$. The corresponding receive signal vector $\mathbf{y} = [y_1 \dots y_{M_R}]^T$ is given by

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

and the signal to noise ratio is defined as $\mathrm{SNR} = 1/\sigma_n^2$.

B. MIMO Detection based on QR Decomposition

The task of the MIMO receiver is to recover $\mathbf{s}$ from $\mathbf{y}$, assuming knowledge of the channel $\mathbf{H}$. To this end, many computationally efficient MIMO detection algorithms start by decomposing $\mathbf{H}$ into a unitary matrix $\mathbf{Q}$ and an upper-triangular matrix $\mathbf{R}$ using QR decomposition such that $\mathbf{H} = \mathbf{QR}$. The transformation $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$ then transforms the problem in (1) into

$$\hat{\mathbf{y}} = \mathbf{R}\mathbf{s} + \tilde{\mathbf{n}}, \tag{2}$$

which is fully equivalent, but can be solved with significantly reduced computational complexity (for example through back-substitution or sphere decoding).

The MMSE QR decomposition is a slight modification to the QR decomposition of $\mathbf{H}$ required for example for zero-forcing linear or OSIC detection. The basic idea is to take the additive noise into account by considering an augmented channel matrix $\bar{\mathbf{H}} = \left[\begin{array}{cc} \mathbf{H}^T & \sigma_n \mathbf{I} \end{array} \right]^T$ to obtain $\mathbf{Q} = \mathbf{Q}_a$ and $\mathbf{R}$ such that

$$\left[\begin{array}{cc} \mathbf{Q}_a & \mathbf{Q}_c \\ \mathbf{Q}_b & \mathbf{Q}_d \end{array} \right] \left[\begin{array}{c} \mathbf{R} \\ \mathbf{0} \end{array} \right] = \left[\begin{array}{c} \mathbf{H} \\ \sigma_n \mathbf{I} \end{array} \right]. \tag{3}$$

The QR decomposition of the augmented channel matrix has a computational complexity that is roughly 50% higher compared to the QR decomposition of $\mathbf{H}$ [3]. However, the algorithm achieves an improvement in BER performance with linear detection and a significant complexity reduction with tree-search based MIMO detection algorithms.

II. Sorted MMSE QR Decomposition Algorithm

The MMSE QR decomposition of $\bar{\mathbf{H}}$ can be performed through the Gram-Schmidt [4] orthogonalization or through a sequence of unitary transformations. The main advantage of unitary transformations resides in the fact that one can employ vector rotations as atomic operations which preserve the total power of the operands. Hence, the dynamic range of all variables is tightly bounded and the algorithm is well suited for fixed-point arithmetic.

To simplify the notation, the composite matrix

$$\mathbf{Z}^{(0)} = \left[\begin{array}{cc} \mathbf{H} & \mathbf{I}_{M_R} \\ \sigma_n \mathbf{I}_{M_T} & 0 \end{array} \right] \tag{4}$$

is introduced. Each Givens rotation, described by the matrix Θ_i, is designed to selectively zero a single entry of $\mathbf{Z}^{(i)}$. In order to upper-triangularize the left half of $\mathbf{Z}^{(0)}$, the iteration $\mathbf{Z}^{(i)} = \mathbf{\Theta}_i \mathbf{Z}^{(i-1)}$ is applied, which ultimately yields

$$\mathbf{Z}^{(N)} = \Theta_N \dots \Theta_1 \mathbf{Z}^{(0)} = \left[\begin{array}{cc} \mathbf{R} & \mathbf{Q}_a^H \\ 0 & \mathbf{Q}_c^H \end{array} \right], \tag{5}$$

where the nulling proceeds first row-by-row and then column-by-column as illustrated in Fig. 1 for $i = 1, 2, \dots, N$. The relevant parts of $\mathbf{Z}^{(N)}$ for further MIMO processing are $\mathbf{R}$ and $\mathbf{Q}_a^H$. The sorting follows the original SQRD algorithm proposed in in [5]. However, while the original description was based on a modified Gram-Schmidt procedure, it is adapted in Alg. 1 to be used with Givens rotations.

III. VLSI Architecture

The application field of the circuit developed in this paper are MIMO-OFDM systems where SQRD must be performed on a large number of channel matrices in a short time [6]. The subsequent architectural considerations target therefore the high-speed region of the design-space.

A. Resource Allocation and Scheduling

For the design of a suitable high-level VLSI architecture, we start by identifying the two types of atomic operations required for Givens rotations, in order to finally constitute the QR decomposition algorithm described in the previous section: *Vectoring* subsumes the computation of Θ_i and the associated nulling of the corresponding entry in $\mathbf{Z}^{(i)}$. *Rotation* refers to the application of Θ_i to an individual column of $\mathbf{Z}^{(i)}$, in which only two entries are affected by the transformation.

Algorithm 1 MMSE-SQRD based on Givens rotations

1: $\mathbf{Z} = \mathbf{Z}^{(0)}$, $p = [1 \dots M_T]$
2: **for** $j = 1, \dots, M_T$ **do**
3: $\quad \xi_j = \|\mathbf{h}_j\|^2$ *initial column norms*
4: **end for**
5: **for** $i = 1, \dots, M_T$ **do**
6: $\quad k = \arg\min_{j=i,\dots,M_T} \xi_j$
7: $\quad$ exchange columns i and k in order array p and in the first $M_R + i - 1$ rows of $\mathbf{Z}$
8: $\quad$ compute a series of Givens rotations Θ_u such that rows $i + M_R, \dots, i + 1$ of column $\mathbf{z}_i$ become zero: $\mathbf{Z} = \left(\prod_{u=(i-1)M_R+1}^{iM_R} \Theta_u \right) \mathbf{Z}$
9: $\quad$ **for** $j = i, \dots, M_T$ **do**
10: $\quad\quad \xi_j = \xi_j - \|\mathbf{Z}_{i,j}\|^2$ *update column norms*
11: $\quad$ **end for**
12: **end for**

As can be seen from the illustration in Fig. 1a, each vectoring operation is followed by multiple rotation operations. Hence, the number of vector rotations exceeds by far the number of vectoring operations, even if explicit rotation is avoided when the affected entries of $\mathbf{Z}^{(i)}$ are a-priori known to be zero. Since a VLSI architecture for high-speed matrix processing needs super-scalar execution units, dedicated hardware resources are allocated for vectoring and rotation, allowing both operations to be carried out concurrently as illustrated in Fig. 1b. The rotation circuit is optimized for speed since the number of operations to be carried out is large. The vectoring circuit needs to perform fewer operations, it can therefore be designed to require less silicon area by applying iterative decomposition. These dedicated VLSI optimizations do not affect the overall throughput of the QR decomposition unit, because the total processing time for the area-optimized vectoring operations can be hidden behind the large number of rotation operations.

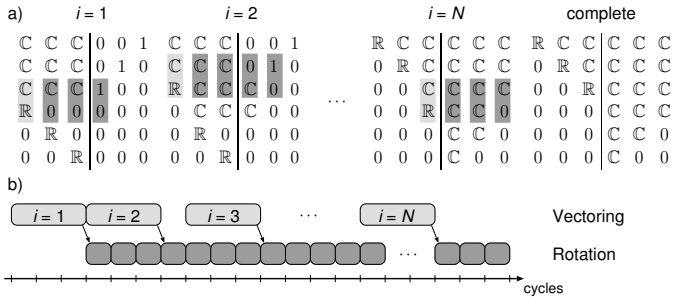

Fig. 1. a) Illustration of the MMSE QR decomposition sequence. The initial matrix $\mathbf{Z}^{(0)}$ is shown at the left, the final result $\mathbf{Z}^{(N)}$ of the decomposition at the right. b) Parallel processing of vectoring and rotation in the proposed VLSI architecture.

B. Implementation of Givens Rotations

The two basic operations for Givens rotations, vectoring and rotation, can both be implemented using *conventional arithmetic* or dedicated *CORDIC circuits* [7]. CORDICs are

a well-established method to implement Givens rotations in hardware. In short, the concept of the CORDIC algorithm is to decompose the rotation of a vector into a series of micro rotations by applying shift and add operations. This sequence of shift and add operations is first determined by the vectoring block, and afterwards executed similarly by the rotation block. A more detailed analysis shows that CORDICs are particularly well suited for the area-efficient implementation of vectoring using iterative decomposition, while fast rotation can be realized more efficient by using conventional complex-valued multipliers [3], but this implies the availability of the corresponding complex-valued rotation coefficients. A solution to this problem is to attach an area-optimized *slave* CORDIC in rotation mode to the vectoring CORDIC as shown in Fig. 2. The input to this slave CORDIC is a unit vector, prescaled by the CORDICs constant scaling factor κ. The corresponding output values are the coefficients required for the multipliers which carry out the vector rotation.

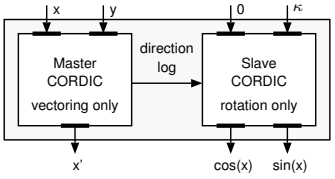

Fig. 2. Enhanced vectoring CORDIC, computes directly $cos(x)$ and $sin(x)$ for subsequent vector rotations using standard multiplications.

C. High-level Architecture

The overall VLSI architecture of the QR decomposition unit is shown in Fig. 3. The dedicated vectoring and rotation circuits (using CORDIC and conventional arithmetic, respectively) are extended to handle complex-valued matrix entries. The memory which stores the original and intermediate matrices $\mathbf{Z}^{(i)}$ is shown as *QR Cache* and is realized using RAMs with a dedicated read and write port. To satisfy the high memory bandwidth requirements of the rotation block (two read and two write accesses per cycle), the cache is split into two independent memory banks. One bank holds the even rows, the other holds the odd rows of $\mathbf{Z}^{(i)}$. Since the rotation block always requires the full memory bandwidth, the vectoring block is fed by a separate FIFO and an additional shadow memory. This solution prevents the rotation block from being stalled by memory access conflicts.

D. Iterative Sorting

The iterative sorting described in Alg. 1 is a key feature of this circuit. The sort metrics are the column norms of $\mathbf{Z}^{(i)}$, which are completely calculated for $\mathbf{Z}^{(0)}$ at the beginning of Alg. 1 in line 3, and which are then iteratively updated for $\mathbf{Z}^{(i)}$, $i \geq 1$ in line 10. The recursive column-norm update procedure requires few additional hardware resources[1] and reordering of the columns of $\mathbf{Z}^{(i)}$ can be implemented efficiently with simple address remappers shown in Fig. 3. However, the sorting occasionally hinders the parallel processing of

[1]In order to reduce hardware complexity, the squared ℓ^2-norm in Alg. 1 can be approximated for example by the ℓ^1-norm or by the ℓ^∞-norm.

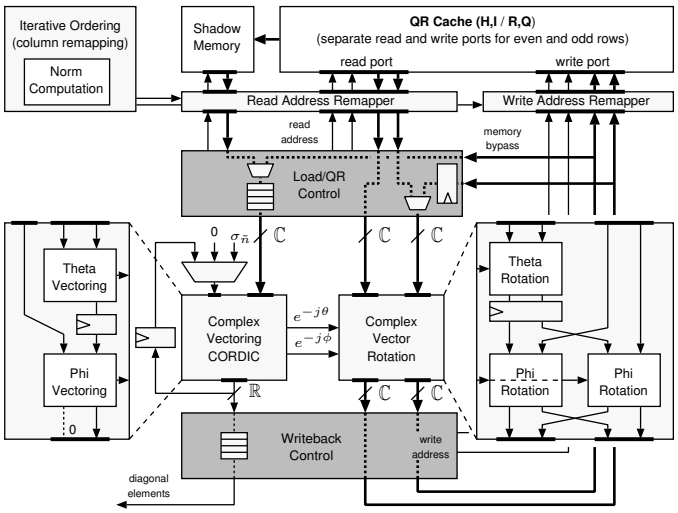

Fig. 3. VLSI architecture of the low latency iterative sorted MMSE QR decomposition with super-scalar vectoring and vector rotation.

vectoring and rotation. The problem arises when the update of one column norm needs to wait for the completion of outstanding rotation operations. In this case, also the next vectoring operation needs to be delayed, until the next column to be processed can be identified based on the updated norms. As a consequence, iterative sorting introduces an additional delay compared to QR decomposition with a fixed column order. Speculative vectoring of the first element in a new column of $\mathbf{Z}^{(i)}$ helps to reduce the associated performance penalty. However, as can be seen from the comparison in Tbl. I, a small increase in the number of cycles compared to unsorted or one-time (a-priori) sorted QR decomposition remains.

TABLE I
PROCESSING TIME OF DIFFERENT MODES

Mode	Processing time per 4×4 matrix
unsorted MMSE-QRD	67 cycles / 536 ns
one-time sorted MMSE-QRD	67 cycles / 536 ns
iterative sorted MMSE-QRD	80 cycles / 640 ns

IV. Implementation Trade-Offs

Implementation trade-offs comprise the *sorting strategy* and the choice of the *fixed-point parameters*. For the subsequent analysis, consider a MIMO system with $M_T = M_R = 4$, 16-QAM modulation and MMSE-(O)SIC detection.

A. Impact of the Sorting Strategy

The BER simulation results for different sorting strategies are shown in Fig. 4. Clearly, a simple one-time sorting as used in [6] already provides a significant performance gain compared to unordered SIC. The iteratively sorted QR decomposition closes the gap between the highly complex, but optimal V-BLAST [8] ordering and the very simple, but less effective one-time sorting.

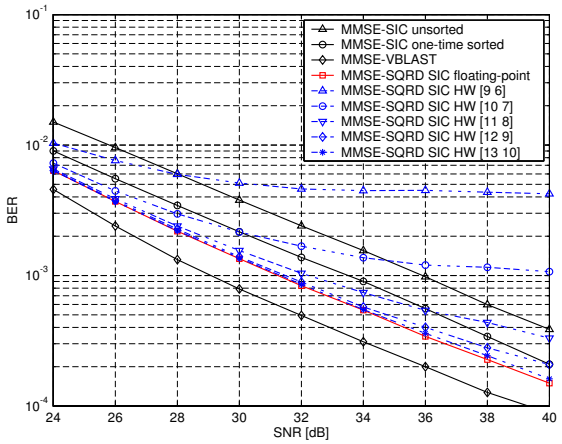

Fig. 4. BER performance of different MIMO detection methods for uncoded 16-QAM, $M_T = M_R = 4$, and perfectly known $\mathbf{H}$ and σ_n ([x y] denotes total number of bits including the sign bit, and fractional bits, respectively)

B. Fixed-Point Considerations

A critical aspect for the efficient implementation of the QR decomposition unit is a conservative choice of the fixed-point parameters. The ultimate performance measure is the implementation loss which relates the BER performance of a fixed-point receiver implementation to the BER achieved with a corresponding floating-point receiver. Unfortunately, analytical expressions for this implementation loss as a function of the number of integer and fractional bits used for the intermediate variables and for the number of CORDIC iterations used for the vectoring are not available. Hence, we must resort to Monte-Carlo simulations. Corresponding results are shown in Fig. 4, where only the QR decomposition has been implemented in fixed-point, while the detection stage performing SIC has been implemented using floating-point arithmetic to clearly separate the two units. Moreover, the input matrix is assumed to be scaled such that the maximum absolute value of real and imaginary parts does not exceed one (*block floating-point*).

V. Implementation Results

The presented circuit has been implemented in a 0.25μm 1P/5M CMOS technology. It supports all possible configurations deducted from $M_T \leq M_R \leq 4$. The sort mode can either be disabled, or set to one-time or iterative sorting. To achieve close to floating-point performance up to an SNR of 40 dB, an internal quantization setting of 3 integer and 10 fractional bits has been chosen together with 9 CORDIC iterations for vectoring. With iterative sorting based on the ℓ^2-norm, the corresponding design requires only 54k gates $(2.1\text{mm}^2, 0.25\mu\text{m})$ and a suitable detection unit [6] would occupy an additional 23k gates $(0.9\text{mm}^2, 0.25\mu\text{m})$.

In comparison, the MMSE V-BLAST described in [9] has a footprint of 9.0mm^2 in a 0.35μm technology which corresponds to roughly 190k gates. A first reason for this significant area penalty is the fact that the V-BLAST algorithm employs two sequential sets of unitary transformations instead of one, which translates either into a twofold area or into a twofold increase in processing time. A second reason lies in the higher sensitivity of the V-BLAST algorithm to quantization effects, which ultimately calls for more complex arithmetic units.

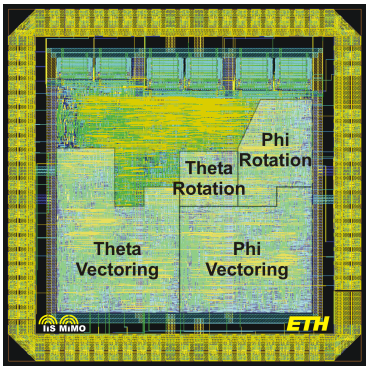

Fig. 5. Layout of the iterative sorted QR decomposition ASIC in 0.25μm 1P/5M CMOS technology: 125 MHz maximum clock frequency and final core area of 2.61 mm^2 at 83% core utilization.

VI. Conclusions

MMSE-SQRD is a key preprocessing step for many relevant MIMO detection algorithms, including successive interference cancellation and sphere decoding. The key to an area efficient, high-throughput VLSI architecture is the joint consideration of algorithmic and VLSI implementation aspects. The implemented MMSE-SQRD algorithm employs Givens rotations. The corresponding rotation matrices are obtained with CORDIC circuits and are applied through complex-valued multipliers. The iterative sorting adds only a small penalty in terms of silicon area and throughput, but provides a considerable BER performance improvement with OSIC and potential for complexity reductions in numerous advanced MIMO detection schemes.

Acknowledgment

This work is supported by the STREP project No. IST-026905 (MASCOT) within the sixth framework program of the European Commission.

References

[1] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoder algorithm," *IEEE Journal of Solid-State Circuits*, 2005.

[2] L. G. Barbero and J. S. Thompson, "Performance analysis of a fixed-complexity sphere decoder in high-dimensional mimo systems," in *Proc. IEEE ICASSP*, vol. 4, May 2006, pp. 557–560.

[3] A. Burg, *VLSI Circuits for MIMO Communication Systems*, Feb. 2006, Ph.D. dissertation, IIS/ETH-Zurich.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*. John Hopkins Univ. Press, 1996.

[5] D. Wübben, J. Rinas, R. Böhnke, V. Kühn, and K. Kammeyer, "Efficient algorithm for detecting layered space-time codes," in *Proc. ITG Conference on Source and Channel Coding*, Jan. 2002, pp. 399–405.

[6] D. Perels, S. Haene, P. Luethi, A. Burg, N. Felber, W. Fichtner, and H. Bölcskei, "ASIC implementation of a MIMO-OFDM transceiver for 192 mbps WLANs," in *Proc. IEEE ESSCIRC*, 2005, to appear.

[7] J. Volder, "The CORDIC trigonometric computing technique," *IRE Trans. Electronic Computers*, vol. EC-8, no. 3, pp. 330–334, Sept. 1959.

[8] P. Wolniansky, G. Foschini, G. Golden, and R. Valenzuela, "VBLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," in *Proc. IEEE ISSSE*, Oct. 1998, pp. 295–300.

[9] Z. Guo and P. Nilsson, "A VLSI implementation of MIMO detection for future wireless communications," in *Proc. IEEE PIMRC*, vol. 3, 2003, pp. 2852–2856.