LUND UNIVERSITY

Energy Efficient Group-Sort QRD Processor with On-line Update for MIMO Channel Pre-processing

Zhang, Chenxin; Prabhu, Hemanth; Liu, Yangxurui; Liu, Liang; Edfors, Ove; Öwall, Viktor

Link to publication

Energy Efficient Group-Sort QRD Processor with On-line Update for MIMO Channel Pre-processing

Chenxin Zhang, *Student Member, IEEE,* Hemanth Prabhu, *Student Member, IEEE,* Yangxurui Liu, Liang Liu, *Member, IEEE,* Ove Edfors, *Member, IEEE,* and Viktor Öwall *Member, IEEE,*

Abstract—**This paper presents a Sorted QR-Decomposition (SQRD) processor for 3GPP LTE-A system. It achieves energy-efficiency by co-optimizing techniques, such as heterogeneous processing, reconfigurable architecture, and dual-supply voltage operation. At algorithm level, a low-complexity hybrid decomposition scheme is adopted, which switches, depending on the energy distribution of spatial channels, between the traditional brute-force SQRD and a proposed group-sort QR-update strategy. A reconfigurable vector processor is accordingly developed to support the adaptive processing with high hardware efficiency. Furthermore, on-chip power management technique is also integrated to obtain real-time power-saving by adapting the voltage supply based on the instantaneous workload. As a proof-of-concept, we implemented the processor using a $65\,\mathrm{nm}$ CMOS technology and conducted post-layout simulation. The proposed SQRD processor occupies $0.71\,\mathrm{mm}^2$ core area and has a throughput of up to $69\,\mathrm{MQRD/s}$. Compared to the brute-force approach, an energy reduction of $10 \sim 61.8\%$ is achieved.**

Index Terms—**QR decomposition, sorting, channel pre-processing, MIMO, reconfigurable processor**

I. INTRODUCTION

Most of modern wireless systems have adopted Multiple-Input Multiple-Output (MIMO) technique to improve bandwidth and transmitting power efficiency [1], [2]. One important processing in MIMO systems is channel matrix pre-processing that enables efficient signal detection at the receiver side. A widely-adopted pre-processing approach is QR-Decomposition (QRD) [3], which is a key prerequisite in many advanced signal detectors such as sphere decoder [4] and K-Best detection [5], [6]. To further improve detection performance, sorted QRD algorithm [7] has been introduced to relieve detection error propagation by optimizing the spatial processing order.

Despite their importance, channel pre-processing units are often excluded from conventional signal detectors [8], as their computations are considered to be less frequent with the assumption that channel is block-stationary [9]. Nevertheless, outdated Channel State Information (CSI) will introduce severe interference and degrade detection performance, especially in high-mobility scenarios. In these cases, in-time CSI update and the corresponding matrix pre-processing are highly desirable in maintaining link quality. On the other hand, SQRD is a quite computationally intensive procedure,

e.g., state-of-the-art SQRD processors consume even more energy than signal detectors [8], around 30 times in [10]. As a consequence, frequent computation of SQRD (e.g., at each CSI update and for each sub-carrier) will introduce unacceptable hardware and energy consumption overhead, making the efficient implementation of SQRD processors a critical challenge, especially for battery-powered mobile devices.

In this study, we tend to tackle this implementation challenge by conducting a co-optimization among different design layers, including decomposition algorithm, hardware architecture, and on-chip power management technique. More specifically, a hybrid decomposition scheme is developed to enable wide trade-off in complexity and performance. Depending on instantaneous channel realizations, an on-line SQRD computation scheme switches adaptively between brute-force SQRD and a very low-complexity QR-Update (QRU). This is performed by fully exploiting the property of LTE-A pilot pattern. The proposed QRU is able to produce exact $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices using only one Givens rotation, reducing the complexity significantly. To obtain the low-complexity benefit of QRU in the context of optimal detection ordering, we further propose an effective group-sort technique which approximates the precise sorting with a two-step (i.e., intra- and inter-antenna group) sorting.

From the hardware-design perspective, the implementation of the proposed hybrid SQRD algorithm is based on a flexible array architecture, providing run-time reconfigurability for adaptive switching between the aforementioned two SQRD schemes. The processor is vector enhanced to effectively leverage the data-level parallelism in MIMO-OFDM systems, e.g., the data stream of different sub-carriers and/or receiving antennas can be processed simultaneously to improve the processing throughput. Effectively harvesting the potential energy savings at run-time requires appropriate power management scheme. This work employs a dual-supply voltage technique, which is a result of coherently considering and optimizing power-saving capability, processor architecture, and workload variance. To demonstrate the effectiveness of the solution, we evaluated the proposed SQRD processor using a $65\,\mathrm{nm}$ CMOS technology. Post-layout simulation results demonstrate a 69 MQRD/s processing throughput. The power consumption is $195.3\,\mathrm{mW}$ for brute-force SQRD with $1.1\,\mathrm{V}$ supply and $74.5\,\mathrm{mW}$ for group-sort QRU with $0.9\,\mathrm{V}$ voltage.

The remainder of this paper is organized as follows: Section II briefly introduces the background, which includes the system model and previous works on QRD update/tracking schemes. Section III describes and evaluates the hybrid SQRD

C. Zhang, H. Prabhu, Y. Liu, L. Liu, O. Edfors, and V. Öwall are with Department of Electrical and Information Technology, Lund University, Lund, Sweden. The email address of authors are {Chenxin.Zhang, Hemanth.Prabhu, Yangxurui.Liu, Liang.Liu, Ove.Edfors, Viktor.Owall}@eit.lth.se

algorithm. Section IV presents the design of vector processor. The implementation results and performance comparison are shown in Section V, and conclusions are drawn in Section VI.

II. BACKGROUND

A. System Model

Considering an equivalent baseband model of a spatial-multiplexing $N \times N$ MIMO system, the $N \times 1$ received complex signal vector $\boldsymbol{y}$ is expressed as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n}, \tag{1}$$

where $\boldsymbol{x}$ is the $N \times 1$ transmit vector, $\boldsymbol{n}$ is the i.i.d. complex Gaussian noise vector with zero mean and variance of σ^2, and $\boldsymbol{H}$ denotes the $N \times N$ channel matrix containing uncorrelated complex-valued channel coefficients. Each component of $\boldsymbol{x}$ is obtained by mapping a set of information bits, encoded by error-correcting codes (ECC), onto a Gray-labelled complex constellation like M-QAM. In this work, we use 3GPP LTE-A downlink as a testbed and resort to Frame-Error-Rate (FER) to evaluate algorithms.

In LTE-A, data transfers are carried out using resource blocks (RBs), each containing 12 consecutive sub-carriers and 7 OFDM-symbol slots allocated in a time-frequency grid. To help with tracking channel changes in both time and frequency domain, scattered orthogonal pilots are inserted. LTE-A pilot pattern for four antenna ports are shown in Fig. 1. One special property to be noticed here is that the pilot tones allocated in the middle slot of each RB are only available for antenna ports 0 and 1. This corresponds to an update of half of columns (e.g., 2 out of 4) in the channel matrix $\boldsymbol{H}$, denoted as half-H renewal with respect to the full-H counterpart (e.g., all elements in $\boldsymbol{H}$ are changed) that takes place at the beginning of each RB. Such channel updating property paves way to various schemes for exploiting pre-processing, see Sec. II-B and Sec. III.

B. Channel Pre-processing - QRD

For channel matrix $\boldsymbol{H}$ of each time-frequency resource grid, one pre-processing approach involves decomposition of the channel matrix into

$$\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}, \tag{2}$$

where $\boldsymbol{Q}$ matrix contains the orthogonal unit vectors and $\boldsymbol{R}$ is an upper-triangular matrix. QRD of a matrix can be implemented efficiently using different algorithms, such as Givens rotation [10], Householder [11], Gram-Schmidt [12], by exploiting the intricate mechanisms of each of these algorithms. However, the computation involved is expensive and has to be performed for each sub-carrier. To tackle this, various approximative approaches are proposed to reduce the complexity at the price of some performance degradation, such as hold-$\boldsymbol{Q}$ [13], [14], interpolating-$\boldsymbol{Q}$ [15].

1) Approximative Hold-$\boldsymbol{Q}$ (Tracking-$\boldsymbol{R}$) Scheme: The approximative method proposed in [13] utilizes the time-domain correlation property of $\boldsymbol{H}$ by assuming that the orthogonal vectors of $\boldsymbol{Q}$ remains unchanged during successive CSI updates, e.g., at the half-H renewal slot. The corresponding $\boldsymbol{R}$
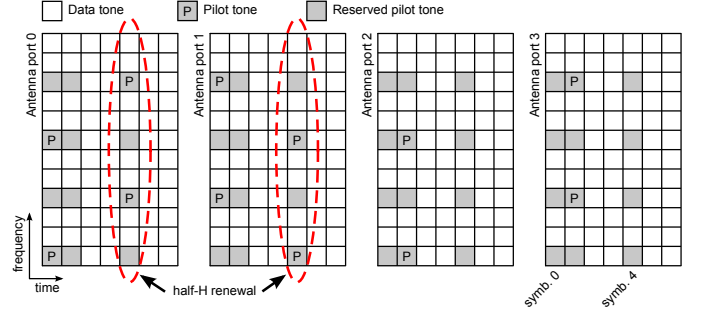

Fig. 1. Scattered pilot pattern for four antenna ports in one LTE-A resource block. Symbol positions of half-H renewals are circled in dashed lines.

matrix absorbs all the variations of the channel and thus is not a pure upper-triangular matrix. The corresponding computation in the hold-$\boldsymbol{Q}$ scheme can be simplified to

$$\tilde{\boldsymbol{H}}_{\text{new}} = \boldsymbol{Q}_{\text{new}}\boldsymbol{R}_{\text{new}} \approx \boldsymbol{Q}_{\text{old}}\tilde{\boldsymbol{R}}_{\text{new}}. \tag{3}$$

Here the subscript "old" refers to the QRD result at previous channel update, e.g., $\boldsymbol{Q}_{\text{old}}$ is the corresponding orthogonal matrix, precisely computed at the full-H renewal slot. The subscript "new" refers to the current updated channel. Different strategies can be employed to perform the tracking-$\boldsymbol{R}$ (or hold-$\boldsymbol{Q}$) scheme, such as least mean square [14], Kalman filter and threshold schemes. As mentioned in [13], a threshold scheme based on the inaccuracy of the outdated $\boldsymbol{Q}_{\text{old}}$ (energy of non-zero elements below the diagonal of $\boldsymbol{R}_{\text{new}}$) is used as the criterion of evoking the approximation in (3). Furthermore, various design trade-offs can be performed to adjust the corresponding performance and computational complexity. However, these methods are bound to introduce errors and in-turn degrade performance, particularly in cases where $\boldsymbol{H}$ changes fast.

2) Exact $\boldsymbol{Q}\boldsymbol{R}$ Update Scheme: To avoid the potential performance loss in aforementioned hold-$\boldsymbol{Q}$ scheme while still keeping the overall complexity low, we investigate a step further and propose an post-processing procedure after the computation of (3), referred to as exact $\boldsymbol{Q}\boldsymbol{R}$ update hereafter. The scheme can be employed in certain scenarios, where only a few columns of the matrix gets changed during CSI update. For example, if only one (n^{th}) column of $\boldsymbol{H}$ matrix changes

$$\boldsymbol{H}_{\text{new}} = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_{n-1}, \boldsymbol{h}_{\text{new},n}, \boldsymbol{h}_{n+1}, \cdots \boldsymbol{h}_N], \tag{4}$$

a low complexity method to compute QR decomposition is deduced by the fact that the multiplication of

$$\boldsymbol{Q}_{\text{old}}^H \boldsymbol{H}_{\text{new}} = [\boldsymbol{Q}_{\text{old}}^H \boldsymbol{h}_1, \cdots, \boldsymbol{Q}_{\text{old}}^H \boldsymbol{h}_{\text{new},n}, \cdots \boldsymbol{Q}_{\text{old}}^H \boldsymbol{h}_N]$$

$$=
\begin{bmatrix}
x & x & \cdots & x & x & x & \cdots & x & x \\
0 & x & \cdots & x & x & x & \cdots & x & x \\
. & 0 & \cdots & . & x & . & \cdots & . & . \\
. & . & \cdots & x & x & . & \cdots & . & . \\
. & . & \cdots & 0 & x & x & \cdots & . & . \\
. & . & \cdots & . & x & x & \cdots & . & . \\
. & . & \cdots & . & x & 0 & \cdots & . & . \\
. & . & \cdots & . & x & 0 & \cdots & x & . \\
0 & 0 & \cdots & 0 & x & 0 & \cdots & 0 & x
\end{bmatrix}, \tag{5}$$

with the column indices $1 \; 2 \; \cdots \; n{-}1 \; n \; n{+}1 \; \cdots \; N{-}1 \; N$ and the n-th column labeled as spike.

resulting in a spike in n^{th} column. The unwanted sub-diagonal elements can be cleared (zeroed) by performing a sequence of Givens rotation as

$$\boldsymbol{G}_{n+1}\cdots\boldsymbol{G}_{N-1}(\boldsymbol{Q}_{\text{old}}^H\boldsymbol{H}_{\text{new}}) = \boldsymbol{R}_{\text{new}}, \qquad (6)$$

where $\boldsymbol{G}_n$ is the Givens matrix for rotation in the planes n and $n+1$, as described in [16].

The key advantage of this method, apart from still being low complex, is that the accurate $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrix can be obtained and hence performs better than hold-$\boldsymbol{Q}$ schemes. In the next section, we apply this scheme in the framework of LTE-A.

3) Application in LTE-A Framework: As mentioned earlier, in case only parts of the matrix columns alter over time, QRD of the new matrix can be performed in a more efficient way than a brute-force computation (referred to as Case-I). Inspired by this, we apply the aforementioned method in the framework of LTE-A downlink receiver and propose a low-complexity QRU scheme during half-H (Fig. 1) renewals. Specifically, the proposed scheme starts with the brute-force SQRD during full-H renewals. During half-H renewals, $\tilde{\boldsymbol{H}}_{\text{new}}$ is obtained by updating two columns of $\tilde{\boldsymbol{H}}_{\text{old}}$. Although orthogonal vectors in $\boldsymbol{Q}_{\text{old}}$ may no longer triangularize $\tilde{\boldsymbol{H}}_{\text{new}}$, it can still have vectors pointing in the correct directions. However, due to the outdated $\boldsymbol{Q}_{\text{old}}$, $\tilde{\boldsymbol{R}}_{\text{new}}$ is no longer an upper-triangular matrix, but may still reveal some upper-triangular properties depending on the positions of the two renewed columns. Specifically, in cases where column changes take place at the right-most of $\tilde{\boldsymbol{H}}_{\text{new}}$, only one element in the lower triangular part of $\tilde{\boldsymbol{R}}_{\text{new}}$ (i.e., $\tilde{r}_{\text{new}}(4,3)$, 4-th row 3-rd column of $\tilde{\boldsymbol{R}}_{\text{new}}$) becomes non-zero. This implies that triangularization of $\tilde{\boldsymbol{R}}_{\text{new}}$ can be significantly simplified by nulling the single non-zero element instead of operating on all columns afresh as

$$\boldsymbol{G}\tilde{\boldsymbol{R}}_{\text{new}} = \boldsymbol{G}\boldsymbol{Q}_{\text{old}}^H\tilde{\boldsymbol{H}}_{\text{new}}, \qquad (7)$$

where

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c & s^* \\ 0 & 0 & -s^* & c^* \end{bmatrix}. \qquad (8)$$

In (8), $(\cdot)^*$ is the complex conjugation, c and s are defined as

$$\begin{aligned} c &= \tilde{r}_{\text{new}}^*(3,3)/z, \\ s &= \tilde{r}_{\text{new}}^*(4,3)/z, \\ z &= \left(|\tilde{r}_{\text{new}}(3,3)|^2 + |\tilde{r}_{\text{new}}(4,3)|^2\right)^{1/2}. \end{aligned} \qquad (9)$$

After triangularizing $\tilde{\boldsymbol{R}}_{\text{new}}$, exact $\boldsymbol{Q}_{\text{new}}$ and $\boldsymbol{R}_{\text{new}}$ in the proposed QRU scheme are obtained, expressed as

$$\boldsymbol{Q}_{\text{new}} = (\boldsymbol{G}\boldsymbol{Q}_{\text{old}}^H)^H, \qquad (10)$$

$$\boldsymbol{R}_{\text{new}} = \boldsymbol{G}\tilde{\boldsymbol{R}}_{\text{new}} = \boldsymbol{G}(\boldsymbol{Q}_{\text{old}}^H\tilde{\boldsymbol{H}}_{\text{new}}). \qquad (11)$$

Note that the lower-right diagonal element of $\boldsymbol{R}_{\text{new}}$ in (11), i.e., $r_{\text{new}}(4,4)$, has been transformed from real to complex-valued domain during this exact QR update. This can be easily resolved by performing an additional real-domain transformation, e.g., using Givens rotation matrix [3], in case if real-valued diagonal elements are required. In the next section, we incorporate the optimal channel ordering in the QRU schemes

and describe a group-sorting mechanism to further exercise this complexity reduction. Moreover, a hybrid scheme will be introduced to explore the advantages of different QRD schemes under varying channel conditions

III. HYBRID SQRD UPDATE ALGORITHM

SQRD is capable of improving detection performance by optimizing processing order based on energy of spatial channels. SQRD starts with a column permutation of the original channel matrix $\boldsymbol{H}$,

$$\tilde{\boldsymbol{H}} = \boldsymbol{H}\boldsymbol{P}, \qquad (12)$$

where $\tilde{\boldsymbol{H}}$ and $\boldsymbol{P}$ denote the sorted channel and the permutation matrix, respectively. The optimal sorting strives to achieve $|r(i,i)| > |r(i-1,i-1)|$, for $i = 2,\cdots,N$, which can be computed using an iterative sorting Gram-Schmidt algorithm [7]. After sorting, a QRD is performed on $\tilde{\boldsymbol{H}}$ to obtain the orthogonal matrix $\boldsymbol{Q}$ and upper-triangular matrix $\boldsymbol{R}$. In the following, we discuss how to apply the low-complexity QRU in the context of SQRD.

A. Sorted QRU Scheme

By combining the traditional brute-force approach (i.e., computing QRD from scratch during half-H renewals) and the QRU scheme, a hybrid decomposition algorithm is formed which dynamically switches between the two schemes to reduce the computational complexity, depending on run-time conditions of the channel reordering. Evidently, the complexity reduction depends on the applicability of the QRU scheme. Intuitively, we could fix the position of antenna ports 0 and 1 to the right-most part of $\tilde{\boldsymbol{H}}_{\text{new}}$ in order to obtain a maximum complexity gain, since it completely avoids brute-force computation during half-H renewals. However, the advantage of channel reordering (for improving detection performance) is lost and we refer to this as Case-II. On the other hand (Case-III), where channel columns are permuted based on the optimal detection order without considering the position of renewed channel columns, the applicability of the QR-update is dramatically reduced. For example, considering the 4×4 MIMO LTE-A, only $(2!2!)/4! = 1/6$ of sorting combinations meet the required update condition, thus limiting the complexity reduction. As a consequence, a smart scheduling strategy is needed to explore the low-complexity potential of the QR-update, while still retaining the performance gain of the optimal channel reordering.

B. Group-sort Algorithm

To fulfil the aforementioned requirement, we propose a group-sort algorithm for channel reordering. Instead of operating on individual columns, sorting of $\boldsymbol{H}$ is applied on two virtual groups, wherein columns associated with antenna ports 0 and 1 are tied together. This way, combinations of "columns" is reduced from 4! to 2!. Consequently, the probability of having both altered columns at the right-most part of $\tilde{\boldsymbol{H}}_{\text{new}}$ increases 3 times, i.e., from $1/6$ to $1/2$. To reduce errors due to the sub-optimal sorting sequences, a two-step sorting scheme

TABLE I
CASE-I−IV OF THE PROPOSED HYBRID SQRD ALGORITHM.

	Channel ordering	Brute-force	QRU
Case-I	Ordering with precise sorting	100%	0%
Case-II	Fixed order for antenna 0 and 1	0%	100%
Case-III	Ordering with precise sorting	83.33%	16.67%
Case-IV	Group-sort	50%	50%

is adopted. First, the sorting between groups is based on the total energy of bundled columns as

$$\mathcal{I} = \arg \max_{i=\{0,1\},\{2,3\}} \sum_i \|\boldsymbol{h}_i\|^2, \tag{13}$$

where $\mathcal{I}$ contains inter-group indexes, e.g., $\mathcal{I} = \{0,1\}$ if antenna ports 0 and 1 has the strongest channels. Second, the two columns within each group, e.g., indexes within $\mathcal{I}$, are intra-group based on the energy of individual columns. To conclude, Table I summaries four cases of the hybrid SQRD scheme and their applicability, wherein we denote the group-sort method as Case-IV. A significant increase to the applicability of the low-complexity QRU is obtained, comparing to the precise sorting scheme in Case-III.

C. Algorithm Evaluation

Before evaluating the overall system performance, an evaluation of the optimal sorting order between full-H and half-H update was performed. In Fig. 2, the probability of the change in the sorting order is shown. The probability of change in the sorting order increases with the increase of Doppler frequency, especially for ETU models. However, it should be noted that the probability of change for the strongest antenna is much lower, which in-turn leads to a good initial (starting) coordinate, critical in detectors. As a consequence, we perform sorting only during full-H updates and maintain the sorting order for half-H update slots, since the probability of change is quite low. This further reduces hardware overhead, e.g., sorting complexity and the storage of permutation matrix.

For simulating the FER of the proposed algorithms, the 3GPP Extended Vehicular A (EVA) channel model with a maximum Doppler frequency of $70\,$Hz is used. Operating at a $2.6\,$GHz carrier frequency, this corresponds to a speed of $29\,$km/h. Channel models with higher mobility, such as Extended Typical Urban (ETU) model with $300\,$Hz Doppler frequency, are not considered in this study, as smaller MIMO configurations (e.g., 2×2) or lower modulation schemes (e.g., QPSK) are expected to be used to mitigate serious interference induced by fast channel variations. In each FER simulation, 5000 LTE-A subframes are transmitted and decoded using a fixed-complexity sphere decoder [17]. The ECC scheme adopted in simulations is a rate 1/2 parallel concatenated turbo code with the coding block length of 5376 and 6 decoding iterations. Furthermore, we assume the receiver has perfect channel knowledge. Performance of the proposed group-sort QR-update and aforementioned cases (Table I) are shown in Fig. 3. Note that Case-III has the same performance as the brute-force approach and is used as a reference for FER
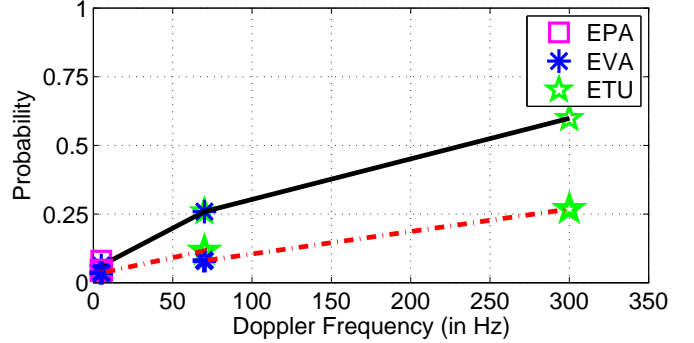

Fig. 2. Sorting order between a full-H and half-H update of different channel models, the solid line represents the probability for any column to change and the dashed line represents the probability of the strongest column changing.
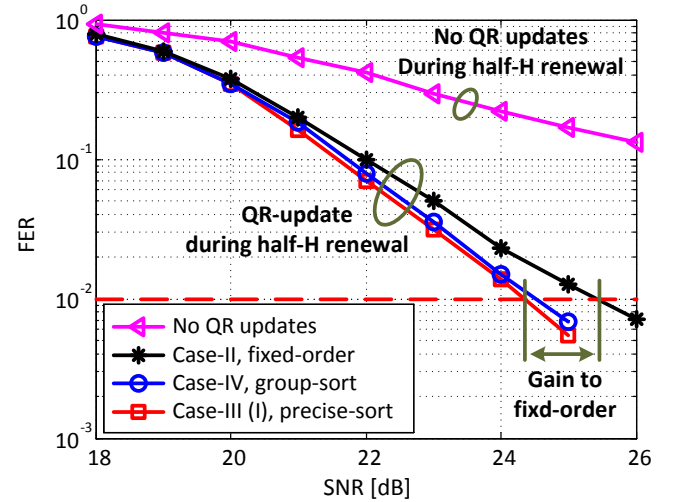

Fig. 3. Simulated FERs in a 4×4 MIMO LTE-A downlink using 3GPP EVA-70 channel model with 64-QAM modulation.

measurements. Compared to the one where no QRDs are performed during half-H renewals (upper curve in Fig. 3), it clearly shows the importance of performing CSI and QR updates even for channels with moderate Doppler shifts. Additionally, the adoption of channel reordering during QR decomposition improves performance to that of the fixed-order approach, e.g., $1.1\,$dB difference between Case-II and III at FER $= 10^{-2}$. Furthermore, the group-sort approach has only small performance degradation of about $0.2\,$dB compared to Case-III, however, with a large complexity reduction as analyzed in the following.

D. Complexity Analysis

Table II summaries the complexity ($\mathcal{C}$) of computations $(2)-(7)$ for an $N \times N$ MIMO system. To perform the brute-force decomposition (2), we consider Gram-Schmidt algorithm [16] that has a total complexity of $\mathcal{C}_1$. Computations required for both (3) and (7) have a total complexity of $\mathcal{C}_2 + \mathcal{C}_3$, which is significantly lower than $\mathcal{C}_1$, e.g., by about 42% for $N = 4$. Note that the product of $\boldsymbol{Q}_{\text{old}}^H \tilde{\boldsymbol{H}}_{\text{new}}$ in (3) requires only half of the matrix computations during QR updates, since only two columns change in $\tilde{\boldsymbol{H}}_{\text{new}}$. The complexity of sorting in both precise- and group-sort approach is denoted as $\mathcal{C}_4$. Based on this analysis and in reference to Case-I,

TABLE II
COMPLEXITY OF COMPUTATIONS IN THE HYBRID SQRD
ALGORITHM.

Complexity	Computation	Multiplication	Others*
$\mathcal{C}_1$	QRD (2)	$N^3 + N^2$	$2N$
$\mathcal{C}_2$	$\boldsymbol{Q}_{\text{old}}^H \tilde{\boldsymbol{H}}_{\text{new}}$ (3)	$\frac{1}{2}N^3$	0
$\mathcal{C}_3$	Triangularization (7)	$N^2 + 2N$	3
$\mathcal{C}_4$	Group Sorting (13)	$N^2 - N$	0
$\mathcal{C}_5$	Iterative Sorting (12)	$\frac{1}{2}N^3 - \frac{1}{2}N^2$	0

* Others include division, square-root, and CORDIC, where the latter
one is often used for generating GR matrices.

TABLE III
COMPLEXITY AND PERFORMANCE COMPARISONS OF CASE-I−IV.

	Complexity	Complexity reduction	SNR loss
Case-I	$\mathcal{C}_1 + \mathcal{C}_5$	− (ref.)	− (ref.)
Case-II	$\mathcal{C}_2 + \mathcal{C}_3$	50%	1.1 dB
Case-III	$\frac{5}{6}\mathcal{C}_1 + \frac{1}{6}(\mathcal{C}_2 + \mathcal{C}_3) + \mathcal{C}_4$	6%	0 dB
Case-IV	$\frac{1}{2}\mathcal{C}_1 + \frac{1}{2}(\mathcal{C}_2 + \mathcal{C}_3) + \mathcal{C}_4$	18%	0.2 dB

Table III shows the complexity reduction versus performance
degradation of Case-II−IV for a 4×4 system. It shows that a
50% complexity reduction is obtained for Case-II. Moreover,
combining the group-sort and the QR-update schemes results
in more palatable trade-offs, i.e., 18% complexity reduction
for only 0.2 dB performance degradation.

To further evaluate the hardware friendliness of the proposed
algorithm, operations required in the four computations (Ta-
ble II) are profiled. In Table IV, most operations are at vector
level, representing a high degree of data level parallelism that
can be exploited to achieve high processing throughput. In
addition, most of them are shared among all computations,
implying that extensive hardware reuse is possible.

IV. VLSI ARCHITECTURE OF THE VECTOR PROCESSOR

Based on the operation analysis, we present an Application-
Specific Instruction-set Processor (ASIP) for performing the
proposed hybrid SQRD algorithm. The hardware architecture
is optimized jointly for flexibility, implementation cost, and
processing speed. Regarding the flexibility requirements in
contemporary system designs for coping with content switch-
ing (e.g., to adopt an appropriate algorithm based on the run-
time update conditions in Table I), a run-time reconfigurable
computing architecture is used. The adopted instruction set
is tailored for baseband processing in MIMO-OFDM systems
with special focus on the efficient computation of the opera-
tions listed in Table IV. The reconfigurable processor employs
the vector processing concept with Single Instruction Multiple
Data (SIMD) [18] architecture to parallelize the processing
with high throughput.

A. Vector Processor

The implementation of the vector processor is based on the
reconfigurable cell array framework presented in [19]. Fig. 4

TABLE IV
OPERATION PROFILE OF THE HYBRID SQRD ALGORITHM FOR $N = 4$

	Vector operations			Scalar+
	$\boldsymbol{A} \cdot \boldsymbol{B}$	$\boldsymbol{A} \odot \boldsymbol{B}^*$	$\boldsymbol{A} \pm \boldsymbol{B}$	
QRD (2)	17	4	6	4
$\boldsymbol{Q}_{\text{old}}^H \tilde{\boldsymbol{H}}_{\text{new}}$ (3)	8	0	0	0
Triangularization (7)	10	0	0	3
Group Sorting (13)	4	0	0	0
Iterative Sorting (12)	9	0	0	0

* Element-wise vector multiplication.
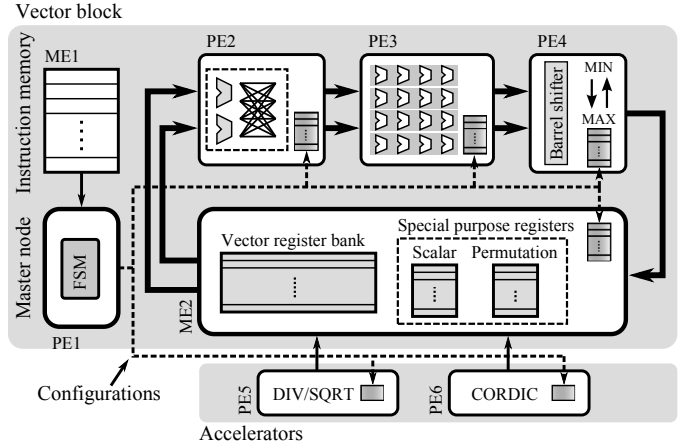+ Scalar operations include division, square-root, and CORDIC.

Fig. 4. Microarchitecture of the vector processor, consisting of 6 PEs and 2
MEs. Solid and dashed lines depict data and control bus, respectively.

shows a microarchitecture of the processor, consisting of 6
processing (PE1−6) and 2 memory (ME1−2) elements inter-
connected via high-bandwidth low-latency links. According
to the type of underlining operations, resource elements are
partitioned into two parts. The upper half performs compu-
tationally intensive vector operations, while the lower part
accelerates special operations such as division/square-root and
CORDIC. Operation modes of these elements are specified in
embedded configuration memories, which are reloadable in
every clock cycle. To ease run-time control of the whole pro-
cessor, a master node (PE1) is responsible for tracking overall
processing flow and controlling configuration memories based
on instructions stored in ME1.

1) Single-cycle Multi-stage Vector Computing: The vector
block has 3 processing (PE2−4) and 1 memory (ME2) el-
ements, functioning as a multi-stage computation path and
a register bank, respectively. PE3 performs all vector oper-
ations in Table IV. To concurrently compute multiple data
streams in $N \times N$ MIMO systems ($N = 4$ in this work),
it is constructed from N homogeneous parallel processing
lanes, each having four pipelined Complex-valued Multiply-
ACcumulate (CMAC) units. It can be seen from Table IV
that vector dot product is the most often used operation.
Thus, in order to reduce computation latency, a single-clock-
cycle vector dot product is supported by each processing
lane. This is accomplished by interconnecting adders in each
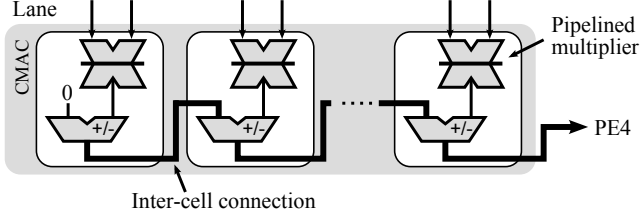row of CMACs to form an adder tree, which can add up

Fig. 5. Illustration of a processing lane in the SIMD core (PE3).

four multiplication results in one clock cycle (see Fig. 5), achieving four concurrent vector operations in each clock cycle. To assist these vector computations, pre- and post-processing elements (PE2 and PE4 in Fig. 4) are employed, which perform operations such as matrix Hermitian (7) and result sorting (12)(13). By combining these three processing elements (PE2−4), several consecutive data manipulations can be accomplished in one single instruction without storing and loading intermediate results. This execution scheme is similar to that of VLIW processors, but has additional flexibility for loading configurations into individual processing elements without affecting others, hence resulting in reduced control overhead. The vector processor is parameterizable and in this work we have used 16 bits computations with 18 bits internal precision for accumulations. The register bank (ME2) contains 16 general purpose vector registers and each configuration memory can buffer upto 16 hardware configurations. Moreover, the instruction memory (ME1) has a capacity of 4 Kbits.

2) Multi-level Zero-delay Inner Loop Control: Multi-level loops are widely used in MIMO-OFDM systems to, for example, process multiple sub-carriers per OFDM symbol, multiple spatial layers per sub-carrier, and multiple iterations per spatial layer. To reduce control overhead during loop operations, a multi-level inner loop control scheme is adopted in PE1. Fig. 6(a) illustrates the deployed stack-based loop controller, which contains a configuration stack used to store the address of the first instruction in a loop (link address) and the corresponding loop count. Compared to other memory structures, the stack has a simple control mechanism and natively supports the execution order of multi-level loops. With the help of a finite-state machine (FSM), link addresses of loops are pushed into the stack in a "last-in-first-out" manner during the execution of a program. Fig. 6(b) shows a snapshot of the stack when the entire loop hierarchy of the enclosed code fragment is pushed into the stack. The adoption of multi-level inner loop control eases program writing, speeds up loop processing, and reduces program size and control overhead.

B. Algorithm Mapping and Scheduling

The presented hybrid SQRD scheme, including brute-force SQRD and group-sort QRU, are mapped manually onto the developed vector processor. The primary objective of the algorithm mapping and task scheduling is to achieve high utilization of the processor and to suffice the stringent timing constraint of LTE-A systems. Moreover, data-level parallelism in MIMO-OFDM systems is exploited to enable high processing throughput and energy efficiency.

In Fig. 7, we detail the mapped operations for both brute-force SQRD and the proposed group-sort QRU. Most of
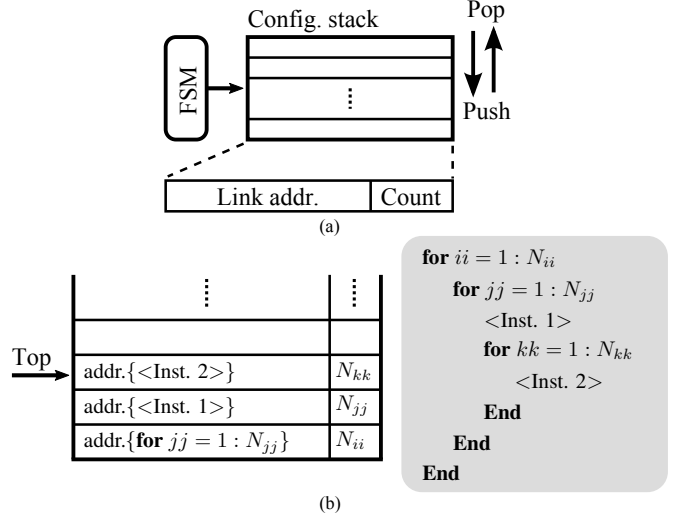

Fig. 6. (a) Multi-level inner loop control. (b) A snapshot of the configuration stack and an example of a code fragment.

TABLE V
PSEUDO-CODE OF THE PROPOSED QRU SCHEME.

for $i = 1 : 12$ **do**	% processing 12 tones in parallel
for $j = 1 : 4$ **do**	% Half-$\boldsymbol{H}$ renewals, Stage 1
for $k = 3 : 4$ **do**	
'$\boldsymbol{A} \cdot \boldsymbol{B}$' to compute $\tilde{\boldsymbol{r}}_{new(j,k)} = \boldsymbol{q}_k^H \boldsymbol{h}_j$	
end for	
end for	
CORDIC $\tilde{\boldsymbol{R}}_{new}$ to obtain $\boldsymbol{G}$	
for $j = 1 : 4$ **do**	% $\boldsymbol{Q}_{old}$ rotation, Stage 2
for $k = 3 : 4$ **do**	
'$\boldsymbol{A} \cdot \boldsymbol{B}$' to compute $\boldsymbol{q}_{k,j} = \boldsymbol{g}_j^H \boldsymbol{q}_k$	
end for	
end for	% $\tilde{\boldsymbol{R}}_{new}$ rotation, Stage 3
'$\boldsymbol{A} \cdot \boldsymbol{B}$' to compute $\boldsymbol{r}_{3,4} = \boldsymbol{g}_3^H \tilde{\boldsymbol{r}}_{new(4)}$	
'$\boldsymbol{A} \cdot \boldsymbol{B}$' to compute $\boldsymbol{r}_{4,4} = \boldsymbol{g}_4^H \tilde{\boldsymbol{r}}_{new(4)}$	
end for	

the operations in brute-force SQRD are vector operations and thus are mainly executed by PE3 (Fig. 4). The scalar division and square-root operations are out-sourced to PE5. As shown in Fig. 7(a), the GS-based SQRD performs matrix orthogonalization iteratively in a column-wise fashion. Each iteration starts with column-norm calculation of the matrix followed by a sorting operation, which is performed in PE4. Each iteration finishes with the energy-normalization of $\boldsymbol{Q}$ matrix and the update of $\boldsymbol{R}$ matrix. It should be pointed out that the workload of the straightforward operation mapping is unbalanced, e.g., calculating the column-norm of a 4×4 matrix occupies all four lanes in PE3 while only one lane is used when generating one column of the $\boldsymbol{Q}$ matrix. To improve the hardware utilization, we leverage the data-level parallelism in OFDM systems and process the channel matrix of multiple sub-carriers simultaneously. More specifically, the processor works with one sub-carrier when current operation occupies all four lanes of PE3, otherwise multiple sub-carriers are processed in parallel. In summary, the execution time of brute-force SQRD for 12 sub-carriers is 87 cycles, resulting in an average execution time of 7.25 cycles per SQRD.
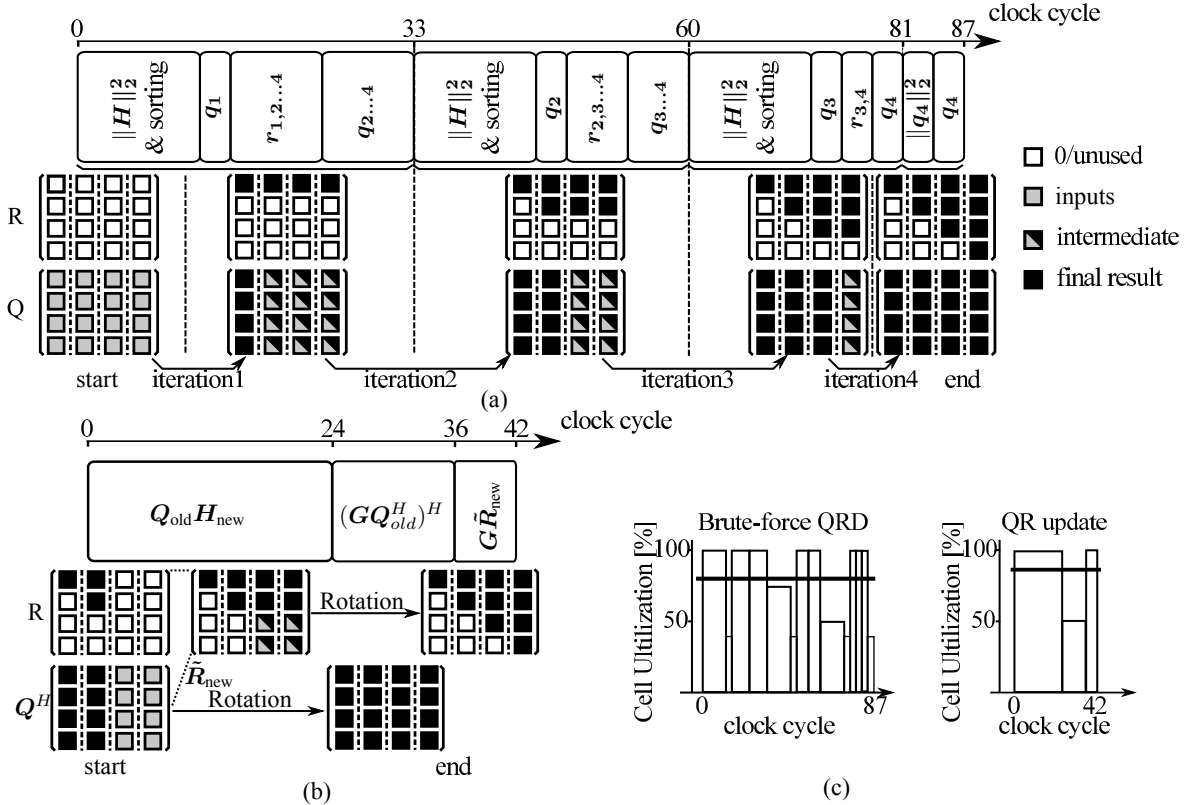
Fig. 7. Example of algorithm mapping, (a) brute-force SQRD, (b) QRU, and (c) hardware utilization of the SIMD core (PE3). Horizontal lines in (c) show the average core-utilization of the corresponding algorithm mapping.

Table V shows the pseudo-code of the proposed QRU scheme for 12 sub-carriers within one LTE-A resource block. The corresponding operations can be separated into three stages, shown in Fig. 7(b). The first stage is the computation of $\tilde{R}_{new} = Q_{old}^H H_{new}$. Since only 2 columns of H_{new} is updated during half-H renewals, column 3 and 4 of R need to be computed (the left half of R remains untouched). Updating each element requires a length-4 vector dot product. The Givens rotation matrix G is obtained using CORDIC (PE6). The second and third stage consist of rotating $\tilde{R}_{new}$ and Q_{old}^H with G. According to (8), G is sparse, i.e., most elements being zeros, which enables parallel computation for multiple sub-carriers. As a result, QRU of 12 sub-carriers requires 42 cycles, averaging 3.5 cycles per QRU. Fig. 7(c) shows the resulting hardware utilization of brute-force SQRD and QRU mapping. It can be seen that the adopted algorithm mapping and task scheduling achieves high utilization of the processor, e.g., resulting in an average core-utilization of 84%.

C. Flexibility

By loading different programs and configurations to the processing and memory cells shown in Fig. 4, the platform has the potential to perform different algorithms, tasks, and radio standards. In the presented case study, the flexibility is demonstrated by dynamically adopting different QRD algorithms to efficiently perform MIMO channel pre-processing. Other examples include mapping of different antenna setups and run-time adaptation of system performance, e.g., adjusting the frequency of channel estimation and MIMO signal detection.

V. IMPLEMENTATION RESULTS AND DISCUSSION

To verify the aforementioned solution (both the hybrid adaptive SQRD algorithm and the reconfigurable vector processor), we implement the proposed SQRD processor using STMicroelectronics 65 nm CMOS technology and evaluate the corresponding hardware cost, processing throughput, and energy consumption with post-layout results.

A. Dual-Supply Voltage

As previously discussed, the proposed QRU scheme is capable of providing accurate Q and R matrix with much lower computational complexity, which is also demonstrated in Fig. 7. Given that the QRU requires only 48.2% of the execution cycles of the brute-force SQRD, one efficient method of harvesting power saving at run-time is to reduce the power supply of the processor at QRU mode. Here, the precondition is to maintain the processing throughput during SQRD mode switching in order to simplify the overall scheduling. In accordance with the available supply voltage in the provided standard cell library, we use ST general purpose standard V_t (GPSVT) cell with 1.1 V voltage support for brute-force SQRD operations and 0.9 V for QRU operations. *Candence* common power format (CPF) design flow is utilized to enable multiple-supply voltage implementation.

B. Post-layout Simulation Results

The designed SQRD processor has a total core area of $0.71\,mm^2$ equivalent to 362 K two-input NAND gates (GE).
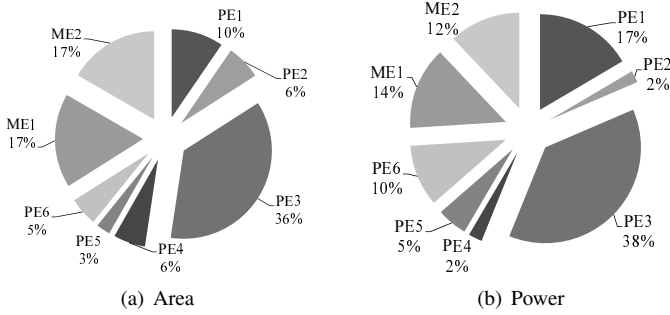
Fig. 8. Area and power breakdown of the vector processor.

TABLE VI
IMPLEMENTATION SUMMARY OF THE VECTOR PROCESSOR.

	Brute-force QRD	Proposed QRU
Gate count [KGE]	339	
Execution cycles	7.25	3.5
Frequency. [MHz]	500	241
Supply voltage [V]	1.1	0.9
Throughput [MQRD/s]	69	
Energy [nJ/QRD]	2.83	1.08

According to post-layout static timing analysis, the maximum clock frequency for brute-force SQRD mode is $500\,$MHz, corresponding to a processing throughput of $69\,$MQRD/S. The power consumption is evaluated using *Synopsis* PrimeTime with switching activities annotated. In the brute-force SQRD mode, the processor operates at $500\,$MHz clock frequency with $1.1\,$V voltage supply and consumes $195.3\,$mW. In the QRU mode, the maximum clock frequency is $241\,$MHz with $0.9\,$V supply and the processor consumes $74.5\,$mW. The corresponding energy consumption for the two QRD computation modes is $2.83\,$nJ and $1.08\,$nJ, respectively. Table VI summaries the implementation results.

To further understand the proposed architecture, the area and power breakdown of the processor is plotted, see Fig. 8. As can be seen, the vector block (PE1-4 and ME1-2) occupies 92% of the total area and consumes on average 85% of power. Among all, the homogeneous CMAC bank (PE3) consumes most of the area and power, and the master node (PE1) together with its instruction memory (ME1) take around 30%. It is worth emphasizing that the proposed low-complexity QRU offers 61.8% of energy saving in comparison to the brute-force SQRD. This is a result of the co-design among algorithm, architecture, and power-saving techniques.

Fig. 9 presents the resulting design trade-offs between energy and performance for Case-I$-$IV of the hybrid SQRD algorithm. Taking the brute-force QRD (Case-I) as a reference, the corresponding numbers on the horizontal axis measures the SNR degradation for reaching the target 10^{-2} FER, while the percentage of energy reduction is shown on the vertical axis. Accordingly, algorithms having their coordinates towards the bottom-left corner is desired. In Fig. 9, it clearly shows that the proposed group-sort QR-update scheme (Case-IV) achieves a good compromise, i.e., trading $0.2\,$dB performance for around 31% energy reduction. In case of energy-constrained systems, the fixed-order scheme (Case-II) can be adopted to further reduce the energy consumption, i.e., by 62% in total,
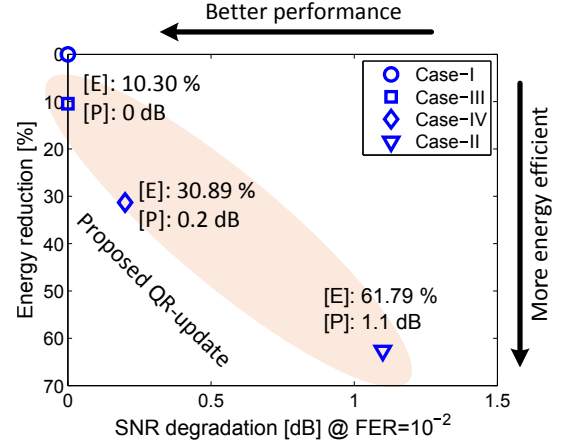

Fig. 9. Energy (E) and performance (P) trade-off for Case-I$-$IV of the hybrid SQRD algorithm.

whereas the precise-sort scheme (Case-III) is used when high performance is demanded. All four update schemes in Fig. 9 are supported by the implemented vector processor, which can be dynamically reconfigured to adopt to run-time channel conditions and performance requirements.

C. Result Discussion

Table VII shows the implementation results of some state-of-the-art MIMO channel pre-processors. The purpose here is not to conduct comparison saying one implementation is better than the other. Instead, we are interested in discussing techniques and strategies at different design layers that can be used to target various optimization objectives, such as system performance (algorithm selection), processing throughput, hardware efficiency, and flexibility. The discussion is supposed to serve as an overview and a design guideline on how to achieve balanced trade-off under different application scenarios. To ease the discussion we divide the table into two sections (columns), i.e., Application-Specific Integrated Circuit (ASIC) and ASIP, and cover 3 broad features (rows). Note that the 8×8 real-valued matrix supported by [11] and [3] in Table VII is an equivalent real domain transformation of the 4×4 complex-valued matrix. Therefore, it is fair to compare system performance with these works.

Top Level Features: ASIPs are reprogrammable within a certain application domain and hence are more flexible for algorithm adaptation and future evolution. In this study, we successfully leveraged this feature and mapped a hybrid SQRD scheme which switches adaptively between Group-sorting QRU and the brute force iterative Gram-Schmidt. Recently, the correlations in wireless channel have been utilized to reduce the computational complexity. For example, in [15], [20], an interpolation of QR in frequency domain is implemented, which can be merged with our proposed QRU scheme to utilize both the time and frequency correlations simultaneously. The reconfigurable processor has the capability to support future algorithm-level modifications.

Architecture Features: Advanced power management schemes like Dynamic Frequency Scaling (DFS) and Application-Driven Clock-Gating (AD-CG) [21] can be employed for low-power design [23]. In our design, the support

TABLE VII
IMPLEMENTATION RESULTS OF DIFFERENT QRD(SQRD) PROCESSORS.

		ASIC				ASIP		
		[11]	[3]	[12]	[20]	[21]	[22]	This Work
Top level Features	Sorting, Update, Interpolation	✗,✗,✗	✗,✗,✗	✓,✗,✗	✓,✗,✓	-	✗,✗,✗	✓,✓,✗
	Matrix Dimension	8×8	8×8	4×4	4×4	4×2	4×4	4×4
		Real	Real	Complex	Complex	Complex	Complex	Complex
Architecture Features	Technology	65 nm	180 nm	90 nm	90 nm	65 nm	65 nm	65 nm
	Supply voltage [V]	1.05	-	-	-	1.2	1.2V	1.1-0.9
	Power Scheme	No	No	No	No	DFS	No	Dual-Supply
	Internal Wordlength	12	16	18	13	32	16	16
Reported Numbers	Execution cycles	-	4	6	4	-	-	5.4
	Frequency. [MHz]	225	100	300	141	400	234	500 @1.1V
	Throughput [MQRD/s]	72	25	50	35	-	10.64	69
	Power [mW]	252	60	-	-	219[a]	169[b]	135
Technology Normalization	Gate Count [KGE]	264	152	334	318	5969[a]	824[b]	362[b]
	Normalized Throughput[+]	72	69.2	69.2	48.5	-	10.64	69
	Normalized Power[#]	228.5	51.2	-	-	152	117.3	135
	Hardware Efficiency[*]	272.7	454	207.2	152.5	-	12.8	191

[+] Normalized Throughput [MQRD/s] = Throughput $\times \frac{\text{Technology}}{65}$ [#] Normalized Power = Power $\times (\frac{\text{Technology}}{65\,nm} \times (\frac{1.1V}{\text{Voltage}})^2)$

[*] Hardware, Efficiency = Normalized Throughput/Gate Count   [a] Only relevant parts of the chip   [b] Data buffers excluded

for dual-supply voltage is implemented to reduce power during the low complexity QRU operation. One important aspect that needs to be pointed out here is the overhead in ASIP, e.g., the extra data storage for intermediate results and the corresponding memory bandwidth requirements (this is not included in this study). On the other side, ASIC implementations are highly streamlined and does not require any intermediate buffers. Optimized within the application domain, ASIPs can be deeply pipelined and parallelized to compensate for the overhead, e.g., running the same instruction for a large amount of data to reduce the instruction fetching overhead.

Internal wordlength is a critical aspect for efficient hardware implementation and system performance. For ASIC implementations, such as in [11] and [20], wordlength optimizations are performed to trade-off between hardware cost and performance. Their results show that QR pre-processing for LTE systems can be performed accurately using around 16 bits. However, performing wordlength optimization for ASIPs is not an easy task, since ASIPs are flexible platforms designed for mapping different algorithms and tasks. To suffice precision requirements for most of the baseband processing tasks, 16 bits computation with 18 bits internal precision for data accumulations is adopted in this work, similar to [22].

Hardware Efficiency: To get a rough idea of the efficiency of different design strategies, we also report the implementation efficiency of different works. The hardware efficiency of ASICs is very high (e.g., [3] with NHE of 454) compared to ASIPs, when considering the overheads involved (like memory and instruction reads). If only considering the processing core, the proposed ASIP achieves the same level of hardware efficiency as that of ASICs, thanks to the high hardware utilization (see Fig. 7).

VI. CONCLUSION

This paper exploits the energy efficient design methodology for algorithm development and VLSI implementation of a SQRD processor for 3GPP LTE-A systems. At the algorithm level, a hybrid QR-decomposition algorithm is proposed to reduce the computational complexity by combining low-complexity QR-update scheme and the traditional brute-force SQRD method. Algorithmic analyses show that a complexity reduction of up to 50% is achieved. To leverage the flexible decomposition algorithm, a reconfigurable vector processor is developed which is able to dynamically switch between different QRD schemes based on energy distribution of spatial channels. Implementation with dual-supply voltage technique demonstrates upto $\sim 62\%$ energy reduction in comparison to the brute-force approach.

REFERENCES

[1] "3GPP TS 36.211 V10.2.0: Physical Channels and Modulation (Release 10)," Jun. 2011. [Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.211/36211-a20.zip

[2] Q. Li *et al.*, "MIMO techniques in WiMAX and LTE: a feature overview," *IEEE Commun. Mag.*, vol. 48, no. 5, pp. 86–92, May. 2010.

[3] Z.-Y. Huang and P.-Y. Tsai, "Efficient Implementation of QR Decomposition for Gigabit MIMO-OFDM Systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 10, Oct. 2011.

[4] A. Burg *et al.*, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2005.

[5] L. Liu *et al.*, "A 1.1-Gb/s 115-pJ/bit Configurable MIMO Detector Using 0.13-μCMOS Technology," *IEEE Trans. Circuits and Syst. II: Exp. Briefs*, vol. 57, no. 9, pp. 701–705, Sept. 2010.

[6] Z. Guo and P. Nilsson, "Algorithm and Implementation of the K-best Sphere Decoding for MIMO Detection," *IEEE J. Sel. Areas in Commun.*, vol. 24, no. 3, pp. 491–503, March 2006.

[7] D. Wubben, R. Bohnke, V. Kuhn, and K. D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *58th IEEE Veh. Technol. Conf.*, vol. 1, Oct 2003, pp. 508–512 Vol.1.

[8] L. Liu, J. Löfgren, and P. Nilsson, "Area-Efficient Configurable High-Throughput Signal Detector Supporting Multiple MIMO Modes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 9, pp. 2085–2096, 2012.

[9] S. Haene, A. Burg, N. Felber, and W. Fichtner, "OFDM channel estimation algorithm and ASIC implementation," in *European Conf. on Circuits and Syst. for Commun.*, July 2008, pp. 270–275.

[10] J. Y. Wang *et al.*, "A 2x2-8x8 sorted QR decomposition processor for MIMO detection," in *IEEE Asian Solid State Circuits Conf.*, 2010.

[11] R. Gangarajaiah *et al.*, "A high-speed QR decomposition processor for carrier-aggregated LTE-A downlink systems," in *European Conf. Circuit Theory and Design*, Sept 2013, pp. 1–4.

[12] Y. Miyaoka, Y. Nagao, M. Kurosaki, and H. Ochi, "Sorted QR decomposition for high-speed MMSE MIMO detection based wireless communication systems," in *IEEE Int. Symp. Circuits and Syst.*, May 2012, pp. 2857–2860.

[13] C. Zhang, H. Prabhu, L. Liu, O. Edfors, and V. Owall, "Energy efficient MIMO channel pre-processor using a low complexity on-line update scheme," in *NORCHIP*, 2012, pp. 1–4.

[14] L. Gor and M. Faulkner, "Power Reduction through Upper Triangular Matrix Tracking in QR Detection MIMO Receivers," in *IEEE 64th IEEE Veh. Technol. Conf.*, Sept 2006, pp. 1–5.

[15] D. Cescato and H. Bolcskei, "Algorithms for Interpolation-Based QR Decomposition in MIMO-OFDM Systems," *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1719–1733, April 2011.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 1st ed. Johns Hopkins University Press, 1983.

[17] L. Barbero and J. Thompson, "Fixing the Complexity of the Sphere Decoder for MIMO Detection," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2131–2142, June 2008.

[18] H. Lee, C. Chakrabarti, and T. Mudge, "A low-power DSP for wireless communications," *IEEE Trans. Very Large Scale Integr. Ssys.*, vol. 18, no. 9, pp. 1310–1322, Sept. 2010.

[19] C. Zhang, L. Liu, D. Marković, and V. Öwall, "A heterogeneous reconfigurable cell array for mimo signal processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, no. 99, pp. 1–10, 2014.

[20] P. L. Chiu *et al.*, "Interpolation-Based QR Decomposition and Channel Estimation Processor for MIMO-OFDM System," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 5, May 2011.

[21] F. Clermidy, C. Bernard, R. Lemaire, J. Martin, I. Miro-Panades, Y. Thonnart, P. Vivet, and N. Wehn, "A 477mW NoC-based digital baseband for MIMO 4G SDR," in *IEEE Inter. Solid-State Circuits Conf. Digest of Technical Papers*, Feb 2010, pp. 278–279.

[22] K. Mohammed and B. Daneshrad, "A MIMO Decoder Accelerator for Next Generation Wireless Communications," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no. 11, pp. 1544–1555, Nov 2010.

[23] J. Rabaey, *Low Power Design Essentials*, 1st ed. Springer Publishing Company, 2009.

Yangxurui Liu received the B.S and M.S degrees from National University of Defense Technology, Changsha, China, in 2010 and 2012, respectively. He is currently working toward the Ph.D. degree in Department of Electrical and Information Technology, Lund University, Lund, Sweden. His research interests include architecture design, VLSI implementation of digital systems and low power signal processing.
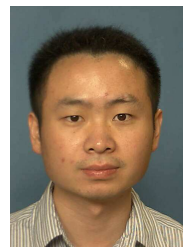
Liang Liu (S'10-M'12) is an Assistant Professor in the Department of Electrical and Information Technology at Lund University, Sweden. He received his B.S. and Ph.D. degree in the Department of Electronics Engineering (2005) and Microelectronics (2010) from Fudan University China. In 2010, he was with Rensselaer Polytechnic Institute, New York, USA as a visiting researcher. He joined Lund University as a Post-doc in 2010. His research interest includes wireless communication system and digital integrated circuits design.

Ove Edfors (M'91) is Professor of Radio Systems at the Department of Electrical and Information Technology, Lund University, Sweden. He graduated with a PhD in Signal Processing from Luleå University of Technology in 1996 and was appointed professor at Lund University in 2002. His research interests include statistical signal processing and low-complexity algorithms with applications in wireless communications. He has long experience with OFDM and MIMO systems and his current research focus is on how realistic propagation characteristics and hardware limitations influence system performance and baseband processing complexity.

Chenxin Zhang (S'09) received his M.S. degree in electrical engineering from Lund University, Sweden in 2009. He is currently working toward the Ph.D. degree in digital circuit design at the Department of Electrical and Information Technology at the same University. From Oct. 2012 to Feb. 2013, he was a visiting scholar at the Department of Electrical Engineering, University of California, Los Angeles. His research mainly focuses on developments of reconfigurable architectures for high computing performance and run-time flexible task mappings.

Hemanth Prabhu (S'11) received his B.E. and M.S. degrees in electronics and communication and System On Chip from VTU, India, and Lund University, Sweden, in 2006 and 2011 respectively. He has worked for 3 years as verification engineer in sasken communication, Bangalore, India. He is currently pursuing his Ph.D. degree in Lund University. His research interests are hardware implementation and signal processing for wireless communication systems such as LTE-A and Massive MIMO.

Viktor Öwall (M'90) received the M.Sc. and Ph.D. degrees in electrical engineering from Lund University, Sweden, in 1988 and 1994, respectively. During 1995 to 1996, he joined the Electrical Engineering Department, the University of California at Los Angeles as a Postdoc. Since 1996, he has been with the Department of Electrical and Information Technology, Lund University, Sweden. He is currently full Professor at the same department and since 2009 the Head of Department. He is the Director of the VINNOVA Industrial Excellence Center in System Design on Silicon (SoS). His main research interest is in the field of digital hardware implementation, especially algorithms and architectures for wireless communication and biomedical applications.