Algorithm and Architecture of a Low-Complexity and High-Parallelism Preprocessing-Based *K*-Best Detector for Large-Scale MIMO Systems

Guiqiang Peng, Leibo Liu, *Senior Member, IEEE*, Sheng Zhou, *Member, IEEE*, Yang Xue, Shouyi Yin, *Member, IEEE*, and Shaojun Wei, *Member, IEEE*

***Abstract*—As a branch of sphere decoding, the *K*-best method has played an important role in detection in large-scale multiple-input-multiple-output (MIMO) systems. However, as the numbers of users and antennas grow, the preprocessing complexity increases significantly, which is one of the major issues with the *K*-best method. To address this problem, this paper proposes a preprocessing algorithm combining Cholesky sorted QR decomposition and partial iterative lattice reduction (CHOSLAR) for *K*-best detection in a 64-quadrature amplitude modulation (QAM) 16 × 16 MIMO system. First, Cholesky decomposition is conducted to perform sorted QR decomposition. Compared with conventional sorted QR decomposition, this method reduces the number of multiplications by 25.1% and increases parallelism. Then, a constant-throughput partial iterative lattice reduction method is adopted to achieve near-optimal detection accuracy. This method further increases parallelism, reduces the number of matrix swaps by 45.5%, and reduces the number of multiplications by 67.3%. Finally, a sorting-reduced *K*-best strategy is used for vector estimation, thereby, reducing the number of comparators by 84.7%. This method suffers an accuracy loss of only approximately 1.44 dB compared with maximum likelihood detection. Based on CHOSLAR, this paper proposes a fully pipelined very-large-scale-integration architecture. A series of different systolic arrays and parallel processing units achieves an optimal tradeoff among throughput, area consumption, and power consumption. This architectural layout is obtained via TSMC 65-nm 1P9M CMOS technology, and throughput metrics of 1.40 Gbps/W (throughput/power) and 0.62 Mbps/kG (throughput/area) are achieved, demonstrating that the proposed system is much more efficient than state-of-the-art designs.**

***Index Terms*—Large-scale MIMO, K-best, sorted QR decomposition, lattice reduction, Cholesky decomposition, VLSI.**

I. Introduction

LARGE-SCALE multiple-input-multiple-output (MIMO) systems have proven to be an efficient solution for coping with rapidly increasing data rates and the exhaustion of spectral resources for future wireless communication systems, e.g., 5G [1]. Although large-scale MIMO systems have many advantages that attract substantial attention, multiple obstacles to implementation remain. One of the essential challenges due to co-channel interference is symbol detection for large-scale MIMO uplinks [2]. Therefore, a detector with low complexity, high accuracy and high processing parallelism is needed. The maximum likelihood (ML) algorithm [3], [4] provides optimal detection accuracy, but it suffers from complexity that increases exponentially with the number of users. Linear detection algorithms, such as zero forcing (ZF) and minimum mean square error (MMSE), have the advantage of low complexity [5], [6]. A variety of low-complexity linear detection algorithms have recently been proposed [7]–[11]. A few implementations using field-programmable gate arrays (FPGAs) [9] and application-specific integrated circuits (ASICs) [10]–[13] have been designed based on linear detectors. However, these types of linear algorithms suffer from non-negligible losses in detection accuracy, especially when the number of users is comparable (or equal) to the number of base station (BS) antennas in a realistic system [9]. Sphere decoding (SD) [14]–[16] and K-best detectors [17]–[19] are two variants of ML detectors. They can achieve attractive trade-offs between complexity and performance through suitable control of the number of nodes on each search level. Best-first tree search [20] combines depth-first and breadth-first search approaches to decide on the traversal direction to reach the shortest path within a reduced search space. However, QR decomposition in these nonlinear detectors leads to high computational complexity and low parallelism because element elimination and other unfavorable matrix operations are involved.

QR decomposition is the decomposition of an estimated channel matrix $\mathbf{H}$ into a unitary matrix $\mathbf{Q}$ and an upper triangular matrix $\mathbf{R}$ [17]. The Gram-Schmidt (GS) method [21]–[23], Householder transformation (HT) [24], [25] and Givens rotation (GR) [26]–[28] are three widely used methods for QR decomposition (see the analysis in Section II). A simplified GS method (WLD) [29]–[33] of matrix decomposition has been proposed to achieve permutation-robust QR decomposition. In GR, the computation of $\mathbf{Q}$ can be substituted with the same rotation operation as that for the upper triangular matrix $\mathbf{R}$. However, in large-scale MIMO systems, as the scale of $\mathbf{H}$ grows, eliminating one element at a time becomes inefficient. Another drawback

Manuscript received July 26, 2017; revised December 1, 2017, January 16, 2018, and January 17, 2018; accepted January 17, 2018. Date of publication January 30, 2018; date of current version February 15, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Subhrakanti Dey. *(Corresponding author: Leibo Liu.)*

G. Peng, L. Liu, Y. Xue, S. Yin, and S. Wei are with the Institute of Microelectronics, Tsinghua University, Beijing 100084, China (e-mail: pgq13@mails.tsinghua.edu.cn; liulb@tsinghua.edu.cn; xueyangthu@163.com; yinsy@tsinghua.edu.cn; wsj@tsinghua.edu.cn).

S. Zhou is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: sheng.zhou@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2018.2799191

of the GR method is that rotation can only be performed once the left two elements have already been eliminated. Hence, the parallelism of the GR method is limited, which results in low throughput in large-scale MIMO systems. To reduce the complexity while fully maintaining the desired diversity gain, lattice reduction (LR) [34], [35] has been proposed as a preprocessing technique for adjusting the estimated channel matrix in accordance with the Lenstra-Lenstra-Lovász (LLL) algorithm for polynomial time [34] (see the analysis in Section II). In addition, LR-based MMSE [21], [35], [36] and LR-based zero-forcing decision feedback (ZF-DF) [37]–[39] methods have been proposed to achieve higher performance compared with conventional linear detection algorithms. However, LR requires enormous numbers of condition checks and column swaps, resulting in uncertain throughput, low parallelism, and long latency in hardware implementations, especially as the MIMO system is scaled up. QR decomposition of the channel matrix is also required in the LR-based ZF-DF method, which is sensitive to layer ordering and scales up efficiently to large dimensions. Therefore, as the numbers of users and antennas increase, the preprocessing (including QR decomposition and LR) performed as part of the K-best algorithm becomes one of the major issues hindering the performance of the entire detector because of the high complexity and low parallelism [34], especially when the channel is not slowly varying. After the preprocessing, the detection routine itself is also a very complicated part of the entire process. The complexity of the K-best algorithm is determined by the number of child nodes in each stage. Finding the optimal solution in a smaller vector space range can greatly reduce the number of child nodes. Hence, the complexity of the detection routine itself can be decreased when the matrix obtained after preprocessing has more advantageous properties (when the preprocessing is more progressive orthogonal).

This paper proposes a preprocessing algorithm that combines Cholesky sorted QR decomposition and partial iterative lattice reduction (CHOSLAR) for K-best detection in a large-scale MIMO system. The CHOSLAR algorithm bypasses the calculation of a specific matrix $\mathbf{Q}$ during QR decomposition. Since the sorted QR decomposition pre-adjusts the matrix $\mathbf{R}$, the scale of the LR operation is also greatly reduced. A partial iterative lattice reduction (PILR) method is applied to obtain a matrix $\mathbf{R}$ with adequate properties. With this preprocessing scheme, a modified K-best detector with reduced sorting and expanding branches can achieve near-ML detection accuracy. Based on the proposed CHOSLAR algorithm, as an example, a very-large-scale-integration (VLSI) architecture is designed to realize a preprocessor for K-best detection in a 64-quadrature amplitude modulation (QAM) 16×16 MIMO system. To achieve the optimal trade-off among throughput, area consumption and power consumption, three types of diagonal-preferential systolic arrays are proposed to perform the initial matrix computation, LR, and matrix inversion. Antenna-level deep pipelined architectures are proposed to achieve high throughput and low latency in the sorted QR and post vector computations. Experiments demonstrate that the proposed architecture achieves a throughput of 3.528 Gbps and a latency of 1.2 μs with 5681 kG gate counts and 2.513 W of power consumption for preprocessing. The architecture possesses substantial advantages in throughput, latency, energy efficiency (throughput/power) and area efficiency (throughput/gate count) compared with state-of-the-art designs.

The remainder of this paper is organized as follows. Section II briefly introduces the system model and related works. Section III specifies the proposed CHOSLAR preprocessing algorithm for K-best detection. The VLSI architecture is illustrated in Section IV. Section V presents the implementation results and their comparison with state-of-the-art designs. Conclusions are drawn in section VI.

Notation: Bold uppercase, bold lowercase, and lowercase letters denote matrices, vectors, and scalars, respectively; $(\cdot)_{i,j}$ denotes the element in the ith row and jth column of a matrix; $(\cdot)_i$ denotes the ith element of a vector; $(\cdot)_{m:n,j}$ denotes the elements from the mth to nth rows and jth column of a matrix; $\mathbf{I}_M$ denotes the $M \times M$ identity matrix; $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^{-1}$, and $(\cdot)^*$ denote transposition, conjugate transposition, inversion, and conjugation, respectively; $\lceil \cdot \rfloor$ denotes a rounding operation; and $|\cdot|$ and $\|\cdot\|$ denote the scalar norm and the Euclidean norm of a vector, respectively .

II. System Model and Related Work

A. System Model

In an $N \times N$ MIMO system with N transmitters on the user side and N antennas on the BS side, the uplink system can be modeled as [40]

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

where $\mathbf{H} \in \mathbb{C}^{N \times N}$ represents a Rayleigh flat-fading channel matrix; $\mathbf{s}$ denotes the transmitted signal vector, which is based on the 64-QAM modulation constellation set Ω in this paper; $\mathbf{n}$ is an additive white Gaussian noise vector with zero mean and variance σ^2; and $\mathbf{y}$ is the received signal vector at the BS. This paper considers a complex-valued system with equal numbers of antennas at the transmitting and receiving ends and assumes that the channel matrix has already been estimated [17], [21], [26], [34], [40].

The K-best detector is a variant of the ML detector. For optimal signal detection, the ML detector solves a finite-set constrained least-squares optimization problem, which can be formulated as

$$\hat{\mathbf{s}}_{\mathrm{ML}} = \arg\min_{\mathbf{s}\in\Omega} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 . \tag{2}$$

This method can also be interpreted in terms of the minimum Euclidean distance (MED) criterion. After QR decomposition of the channel matrix $\mathbf{H}$, (2) can be transformed into

$$\hat{\mathbf{s}}_{\mathrm{ML}} = \arg\min_{\mathbf{s}\in\Omega} \left\|\mathbf{Q}^H\mathbf{y} - \mathbf{R}\mathbf{s}\right\|^2 = \arg\min_{\mathbf{s}\in\Omega} \|\hat{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 , \tag{3}$$

where $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$ and $\mathbf{H} = \mathbf{Q}\mathbf{R}$, with $\mathbf{Q}$ being a unitary matrix and $\mathbf{R}$ being an upper triangular matrix. ML MIMO detectors generate soft outputs[1] in the form of log-likelihood ratios (LLRs) by finding other symbol vectors that are closest to $\hat{\mathbf{y}}$. According

[1]In this paper, the detector generates a hard output, as we focus on the computational complexity, parallelism and hardware practicability during preprocessing (QR decomposition and LR). A soft output is not considered for the detector of this paper but will be considered in future work.

to [29]–[32], the LLR of bit b for symbol i can be computed as follows:

$$LLR\left(s_{i,b}\right) = \frac{1}{\sigma^2}\left(\min_{\mathbf{s}\in\mathbf{S}_{i,b}^{0}}\left\|\hat{\mathbf{y}}-\mathbf{R}\mathbf{s}\right\|^2 - \min_{\mathbf{s}\in\mathbf{S}_{i,b}^{1}}\left\|\hat{\mathbf{y}}-\mathbf{R}\mathbf{s}\right\|^2\right), \tag{4}$$

where $\mathbf{S}_{i,b}^{0}$ and $\mathbf{S}_{i,b}^{1}$ denote the modulation constellation symbol sets from the modulation constellation Ω with $s_{i,b}$ equal to 0 and 1, respectively.

Compared with brute-force-search-based ML decoding, tree-search-based detection significantly reduces the search area for potential solutions. In each layer, the tree expands each parent node into a few child nodes, sorts all these nodes by their accumulated partial Euclidean distances (PEDs), and then selects the K-best candidates as the parent nodes for the next layer. The accumulated PEDs for the ith layer can be expressed as

$$\mathrm{PED}_i = \mathrm{PED}_{i+1} + \left|\hat{y}_i - \sum_{j=i}^{N} r_{i,j}s_j\right|^2, \quad 1\le i\le N. \tag{5}$$

The tree search starts from the last element of $\mathbf{s}$ since $\mathbf{R}$ is an upper triangular matrix. The scale of the channel matrix $\mathbf{H}$ increases as the size of the MIMO system increases (e.g., from 2 $\times$ 2 to 16 $\times$ 16 or more). Hence, QR decomposition is difficult to implement in hardware because of the high complexity and data independency, especially in large-scale MIMO systems. Therefore, the complexity and parallelism of the QR decomposition computation are the main factors constraining large-scale MIMO detection throughput.

B. QR Decomposition

A detailed comparison of the computational complexities of the GS, HT and GR methods was presented in [17] based on the numbers of real-valued operations. For complex-valued systems, the GS and GR methods require approximately $\mathcal{O}(4N^3)$ real-valued multiplications, whereas HT requires many more multiplications due to the computation of the unitary matrix $\mathbf{Q}$ [24], [25]. Therefore, although HT is able to eliminate elements in a by-column manner, its hardware consumption is undesirable. In [22], a modified GS method was proposed for the implementation of 4 $\times$ 4 QR decomposition. The expressions for all elements in the matrices $\mathbf{Q}$ and $\mathbf{R}$ and the intermediate values were fully expanded for the sake of parallel design; nevertheless, the multiplication burden was not reduced compared with the conventional GS method. In [29]–[33], a simplified GS method (WLD) for matrix decomposition was proposed to achieve permutation-robust QR decomposition with low complexity. In [41], a time-correlated channel model was considered in which only portions of the matrix columns are renewed over time, and a hybrid method combining an approximate Hold-Q scheme and an exact QR update scheme was proposed to reduce the computational complexity. In [17], GR was adopted for QR decomposition using a complex coordinate rotation digital computer (CORDIC) unit to simplify complicated arithmetic units by means of iterative shift and add operations. Since detection based on a real-valued system was considered in [17], a new decomposition scheme was proposed to further reduce the number of arithmetic operations. However, simple QR decomposition with K-best detection suffers from heavy accuracy loss compared with ML detection. Other nonlinear signal detection methods, such as triangular approximate semidefinite relaxation (TASER) [40] and probabilistic data association (PDA) [42], are also not suitable for high-order modulation systems because of accuracy losses and implementation difficulties.

Sorted QR decomposition was proposed to reduce the search range and further improve the accuracy of K-best detection while incurring low complexity. During the execution of K-best detection, each element of $\hat{\mathbf{s}}_{\mathrm{ML}}$ can be estimated as

$$\hat{s}_i = \left(\hat{y}_i - \sum_{j=i+1}^{N} r_{i,j}s_j\right)\Big/ r_{i,i}, \quad 1\le i\le N. \tag{6}$$

Since $\hat{y}_i$ contains noise, a large diagonal element $r_{i,i}$ results in better immunity to noise and signal interference. QR decomposition is performed column by column, which means that the matrix $\mathbf{R}$ is generated row by row. Because the absolute value of the matrix $\mathbf{H}$ is fixed, the product of all diagonal elements of $\mathbf{R}$ is also constant. During the decomposition of the ith column, we have

$$r_{i,i} = \left\|h_{i:N,i}\right\| = \sqrt{\sum\nolimits_{j=i}^{N} h_{j,i}^2}, \quad 1\le i\le N, \tag{7}$$

in which $r_{i,i}$ decreases as i increases. Sorting and selecting the column with the minimum norm as the next column to be decomposed ensures that the product of the remaining diagonal elements will be as large as possible. In [43], a detector was proposed that combined sorted QR decomposition with the GS method and pairwise column symmetrization for a 64-QAM 4 $\times$ 4 MIMO system. The detector had a complexity of $\mathcal{O}(4N^3)$ with a slight loss of accuracy. In [26], sorting was combined with GR for preprocessing. In addition, a VLSI architecture was proposed to achieve flexibility for 64-QAM $1\times 1 \sim 4\times 4$ MIMO systems. In this architecture, the sorted QR decomposition is implemented with a long chain of duplicated CORDIC arrays, which results in long latency. When the size of the MIMO system increases, the problem of long latency becomes more serious.

C. Lattice Reduction

LR is adopted in the proposed method to pursue sufficient detection performance. When LR is applied to the matrix $\mathbf{H}$, an approximately orthogonal channel matrix is obtained:

$$\bar{\mathbf{H}} = \mathbf{H}\mathbf{T}, \tag{8}$$

where $\mathbf{T}$ is a unimodular matrix. $\bar{\mathbf{H}}$ is much better conditioned than $\mathbf{H}$; therefore, $\bar{\mathbf{H}}$ leads to less noise enhancement. The LLL algorithm is a well-known LR method by virtue of its polynomial time complexity [34]. This algorithm checks and adjusts the matrix $\mathbf{R}$ to satisfy two conditions. This paper adopts various LLL algorithms that use the Siegel condition [34] and a size reduction condition, which are expressed in (9) and (10), respectively. The parameter δ satisfies $0.25 < \delta < 1$. The Siegel condition maintains the difference between two neighboring diagonal elements

within a small range to prevent excessively small diagonal elements. The size reduction condition ensures that the diagonal elements are slightly dominant to achieve approximate orthogonality. As mentioned before, the adjusted channel matrix or the upper triangular matrix $\mathbf{R}$ can suppress interference between different antennas.

$$\delta \left| r_{k-1,k-1} \right| > \left| r_{k,k} \right|, \quad k = 2, 3, \ldots N \tag{9}$$

$$\frac{1}{2} \left| r_{k-1,k-1} \right| > \left| r_{k-1,j} \right|, \quad 2 \le k \le j \le N \tag{10}$$

However, LR requires enormous numbers of condition checks and column swaps, resulting in uncertain throughput, low parallelism and long latency. In [34], 3 pairs of CORDIC processors were used to perform LR via an odd-even method. The detector achieved near-ML accuracy for a 64-QAM 8×8 MIMO system. Considering a VLSI implementation, a few CORDIC pairs were designed for both QR decomposition and LR; these pairs were responsible for most of the latency, according to the timing schedule.

These two techniques (sorted QR decomposition and LR) are essentially similar; they adjust the matrix $\mathbf{R}$ to possess more advantageous properties. Hence, K-best detection achieves near-ML accuracy with a small amount of branch expansion. However, as the size of the MIMO system increases, the computational complexity for preprocessing becomes unmanageable. In GR-based QR decomposition, only one element a time is removed, resulting in low parallelism. For example, a trade-off was established in [34] in which throughput was sacrificed for hardware utilization. Moreover, sorted QR decomposition and LR both result in higher computational complexity, especially LR, which requires an unstable number of low-parallelism operations.

III. Proposed Signal Detection Algorithm for a Large-Scale MIMO System

In this section, a CHOSLAR preprocessing algorithm for K-best detection is proposed . Then, a modified K-best detector with reduced sorting is presented. The detection performance is discussed intermixed with the algorithms for different combinations of antenna numbers and modulation types. Finally, the complexity reduction and parallelism are analyzed.

A. CHOSLAR Preprocessing

1) Cholesky Sorted QR Decomposition: In K-best detection, the matrix preprocessing must enable high detection accuracy and calculation efficiency. The first preprocessing step in K-best detection is the QR decomposition of the channel matrix $\mathbf{H}$. In this decomposition process, the calculation of the unitary matrix $\mathbf{Q}$ results in great complexity. The proposed QR decomposition scheme attempts to avoid the calculation of the matrix $\mathbf{Q}$ by exploiting the properties of the matrices $\mathbf{Q}$ and $\mathbf{R}$. From the definition of QR decomposition, $\mathbf{H} = \mathbf{QR}$, where $\mathbf{Q}$ is a unitary matrix. Hence,

$$\mathbf{H}^H \mathbf{H} = (\mathbf{QR})^H \mathbf{QR} = \mathbf{R}^H \left(\mathbf{Q}^H \mathbf{Q}\right) \mathbf{R} = \mathbf{R}^H \mathbf{R}. \tag{11}$$

Algorithm 1: Cholesky decomposition.

```
for i = 1; i ≤ N − 1; i + 1 do
    r_{i,i} = √a_{i,i}
    for j = i + 1; j ≤ N; j + 1 do
        r_{i,j} = a_{i,j} / r_{i,i}
    end for
    for m = i + 1; m ≤ N; m + 1 do
        for n = m; n ≤ N; n + 1 do
            a_{m,n} ← a_{m,n} − r^H_{i,m} r_{i,n}
        end for
    end for
end for
r_{N,N} = √a_{N,N}
```

In addition, the elements of the channel matrix $\mathbf{H}$ are assumed to be subject to uncorrelated Rayleigh flat fading; i.e., they are independent and identically distributed (i.i.d) complex Gaussian variables with zero mean and unit variance. Therefore, the matrix $\mathbf{H}$ is non-singular, which means that the matrix $\mathbf{A} = \mathbf{H}^H \mathbf{H} = \mathbf{R}^H \mathbf{R}$ is positive definite. Thus, $\mathbf{A}$ is a Hermitian positive-definite matrix. Cholesky decomposition is a method of decomposing a Hermitian positive-definite matrix into the product of a lower triangular matrix and its conjugate transpose, i.e., $\mathbf{A} = \mathbf{L}\mathbf{L}^T$. When Cholesky decomposition is applied, the matrix $\mathbf{R}$ is equal to the upper triangular matrix $\mathbf{L}^T$, and the computation of a specific unitary matrix $\mathbf{Q}$ is avoided. Although a shortcut is taken when obtaining matrix $\mathbf{R}$, according to (3), the matrix $\mathbf{Q}$ is still required to compute $\mathbf{Q}^H \mathbf{y}$. Since $\mathbf{H} = \mathbf{QR}$, $\mathbf{Q}$ can be directly computed as $\mathbf{Q} = \mathbf{HR}^{-1}$. Therefore, the calculation of $\mathbf{Q}^H \mathbf{y}$ is transformed as follows:

$$\mathbf{Q}^H \mathbf{y} = \left(\mathbf{HR}^{-1}\right)^H \mathbf{y} = \left(\mathbf{R}^{-1}\right)^H \mathbf{H}^H \mathbf{y}, \tag{12}$$

where inversion of the upper triangular matrix $\mathbf{R}$ and two matrix-vector multiplications are performed in place of $\mathbf{Q}^H \mathbf{y}$. The complexity of computing the upper triangular matrix is $\mathcal{O}(N^3)$, which is significantly reduced compared with the complexity of computing $\mathbf{Q}$ directly. In GR, during the elimination of elements in the channel matrix $\mathbf{H}$, applying the same transformation to the vector $\mathbf{y}$ can also eventually lead to $\mathbf{Q}^H \mathbf{y}$, where the calculation of $\mathbf{Q}$ is implicit. However, the calculation of $\mathbf{R}$ itself requires $\mathcal{O}(4N^3)$ multiplications. A quantitative comparison will be presented in Section III-C.

The next problem is how to combine the sorting operation with QR decomposition. In the proposed Cholesky-based method, QR decomposition is realized using the Gram matrix $\mathbf{A}$. Based on Cholesky decomposition, the matrix $\mathbf{R}$ is also generated row by row. The pseudo-code for Cholesky decomposition is presented in Algorithm 1. Here, a 4×4 matrix is used as an example. When $i = 1$ (the first round of sorting), as a result of $\mathbf{A}$, the values compared during sorting are

$$v_k = h_k^H h_k = |h_k|^2 = a_{k,k}, \quad k = 1, 2, 3, 4, \tag{13}$$

which means that each diagonal element $a_{k,k}$ (in the kth row and kth column of $\mathbf{A}$) is the square norm of the corresponding column of the matrix $\mathbf{H}$. Hence, the elements of the row and

column with the smallest diagonal element in $\mathbf{A}$ are directly exchanged with the elements of the first row and column, and then, the first round of decomposition can begin. When $i = 2$ (the second round of sorting), the values v_k satisfy

$$\begin{aligned} v_k &= |h_k|^2 - r_{i-1,k}^H r_{i-1,k} \\ &= a_{k,k} - r_{i-1,k}^H r_{i-1,k}, \quad k = 2, 3, 4. \end{aligned} \tag{14}$$

According to (14) and line 8 in Algorithm 1, the values v_k are the diagonal elements of the matrix $\mathbf{A}$ (updated in the previous round of decomposition). Thus, when $i = 2$, the diagonal elements of the (updated) matrix $\mathbf{A}$ can again serve as the basis for sorting. For $i = 3, 4$, the analysis is similar. Therefore, the diagonal elements of the Gram matrix $\mathbf{A}$ can always be used as the norm for each column for the sorting operation. In the conventional GR method, when $i = 2$ (after the first round of decomposition), the matrix $\mathbf{H}$ can be written as

$$\begin{bmatrix} h_{1,1}, h_{1,2}, h_{1,3}, h_{1,4} \\ h_{2,1}, h_{2,2}, h_{2,3}, h_{2,4} \\ h_{3,1}, h_{3,2}, h_{3,3}, h_{3,4} \\ h_{4,1}, h_{4,2}, h_{4,3}, h_{4,4} \end{bmatrix} \rightarrow \begin{bmatrix} r_{1,1}, r_{1,2}, r_{1,3}, r_{1,4} \\ 0, \ h'_{2,2}, h'_{2,3}, h'_{2,4} \\ 0, \ h'_{3,2}, h'_{3,3}, h'_{3,4} \\ 0, \ h'_{4,2}, h'_{3,4}, h'_{4,4} \end{bmatrix}, \tag{15}$$

where $h'_{i,j}$ is the updated element in the ith row and jth column of the matrix $\mathbf{H}$. Since the GR method does not change the norm of each column, the values compared during the sorting of the kth column are

$$v_k = \sum_{j=i}^{4} \left(h'_{j,k}\right)^H h'_{j,k}, \quad k = 2, 3, 4, \tag{16}$$

when performing the decomposition of the second column. Hence, to achieve correct sorting, the values compared during each round of sorting must be computed in accordance with the updated matrix $\mathbf{H}$. This calculation costs $\frac{2}{3}N^3$ real-valued multiplications. Fig. 1 illustrates the difference between the proposed Cholesky sorted QR decomposition method and the conventional method. It is assumed that the fourth column has the smallest norm value in the second round of decomposition. In the proposed algorithm, sorting is performed by exchanging the rows and columns after the matrix update based on only the diagonal elements, shown in red in Fig. 1–(a). By contrast, in the conventional method, the square norm values of all three column vectors are required before sorting and column exchange can be performed, as indicated by the gray dashed circles in Fig. 1–(b).

Compared with other Cholesky decomposition methods used in MIMO detection [39], [44]–[46], the method proposed in this paper has a different purpose, and the details of its realization are also different. In ZF-DF and Successive Interference Cancellation (SIC) detection [39], [44], the diagonal elements of $\mathbf{R}$ need to be computed via QR decomposition. The Gram matrix is decomposed into a lower triangular matrix with unit elements on its diagonal ($\mathbf{L}$) and a diagonal matrix with real elements ($\mathbf{D}$). After the decomposition of $\mathbf{D}$, the diagonal elements of $\mathbf{R}$ can then be computed. However, in K-best detection, the entire upper triangular matrix $\mathbf{R}$ is required. Hence, the matrix $\mathbf{R}$ cannot be simply computed in the LDL manner; other matrix computations are required. Therefore, the subsequent K-best routine will be affected. In the proposed CHOSLAR method,

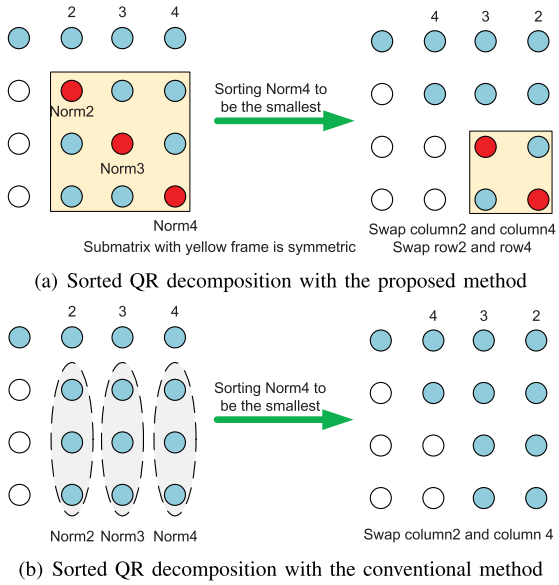
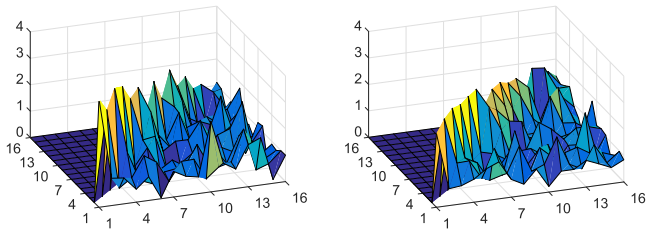


Fig. 1. Difference between the proposed Cholesky sorted QR decomposition method and the conventional method.

the matrices $\mathbf{R}$ and $\mathbf{Q}$ can be obtained directly through matrix decomposition, allowing the execution of the subsequent K-best routine to start in a short amount of time. In [45], [46], Cholesky decomposition was proposed for decomposing and inverting the Gram matrix in linear MMSE detection. Compared with these methods, in the proposed CHOSLAR approach, Cholesky QR decomposition is performed first, and LR and K-best detection are performed later. The proposed method can thus be adapted to nonlinear detection with higher performance. In summary, the proposed method can better facilitate the subsequent K-best search compared with other methods. In addition, the proposed method includes a sorting operation during the decomposition process, which increases the detection accuracy compared with the Cholesky decomposition methods presented in [39], [44]–[46]. Hence, the resulting matrix has flattened diagonal elements, which will greatly facilitate the subsequent LR and K-best computations and increase the detection accuracy. In traditional QR decomposition, sorting operations have been proposed to optimize the matrix $\mathbf{R}$ [21]–[23], [27], [28]. In the proposed sorted QR decomposition method, the norm of each column of the matrix $\mathbf{H}$ has already been computed when the matrix $\mathbf{A}$ is being adjusted. Consequently, the proposed sorted QR decomposition method does not require additional multiplications because the adjustment of $\mathbf{A}$ is performed as part of the Cholesky decomposition process. By contrast, conventional QR decomposition is realized directly using the matrix $\mathbf{H}$. The decomposition process is performed column by column, and the sorting operation requires an additional step to calculate the norms of all columns that remain undecomposed, whether GR [27], [28] or the GS method [21]–[23] is used.

2) Partial Iterative Lattice Reduction: This paper presents a PILR method with constant throughput to reduce the number of column swaps. The proposed PILR method proceeds from the last row to the $(N/2 + 1)$th row T times in an iterative



(a) Matrix $\mathbf{R}$ without sorting (b) Matrix $\mathbf{R}$ with sorting

Fig. 2. The values of the elements of the matrix $\mathbf{R}$ with and without sorting.

way. In the kth step of each iteration ($N/2+1 \leq k \leq N$), the method first checks the Siegel condition expressed in (9). If this condition is violated, the parameter μ is calculated as follows:

$$\mu = \lceil r_{k-1,k}/r_{k-1,k-1} \rfloor. \tag{17}$$

Then, the $(k-1)$th column of $\mathbf{R}$ is multiplied by μ and subtracted from the kth column as follows:

$$r_{1:k,k} \leftarrow r_{1:k,k} - \mu r_{1:k,k-1}. \tag{18}$$

The channel matrix $\mathbf{H}$ is used directly as follows:

$$h_{:,k} \leftarrow h_{:,k} - \mu h_{:,k-1}. \tag{19}$$

This step is a one-time size reduction to ensure that the Siegel reduction is processed correctly. After this one-time size reduction, the elements in columns k and $(k-1)$ are swapped in $\mathbf{R}$ and $\mathbf{H}$ to obtain $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$. This swap operation breaks the triangularity of the matrix $\mathbf{R}$; consequently, a 2×2 GR matrix θ is needed:

$$\theta = \begin{bmatrix} a^H, & b^H \\ -b, & a \end{bmatrix}, \tag{20}$$

where

$$a = \frac{\hat{r}_{k-1,k-1}}{\sqrt{\hat{r}^2_{k-1,k-1} + \hat{r}^2_{k,k-1}}}, b = \frac{\hat{r}_{k,k-1}}{\sqrt{\hat{r}^2_{k-1,k-1} + \hat{r}^2_{k,k-1}}}. \tag{21}$$

Finally, rows k and $(k-1)$ are updated by multiplying by matrix θ, which recovers the triangularity of the matrix $\mathbf{R}$, as follows:

$$\begin{bmatrix} \hat{r}_{k-1,k-1}, \hat{r}_{k-1,k}, \dots \\ 0, \quad \hat{r}_{k,k}, \dots \end{bmatrix} \leftarrow \theta \times \begin{bmatrix} \hat{r}_{k-1,k}, \hat{r}_{k-1,k-1}, \dots \\ \hat{r}_{k,k}, \quad 0, \dots \end{bmatrix}. \tag{22}$$

The full size reduction process is similar to the one-time size reduction step but is performed for the entire matrix $\hat{\mathbf{R}}$. When performing full size reduction, the proposed algorithm proceeds from $\hat{r}_{k,k}$ to $\hat{r}_{1,1}$, element by element. As an example, Fig. 2 shows the values of the elements of the matrix $\mathbf{R}$ with and without sorting for a 16×16 MIMO system. The values of the diagonal elements after sorted QR decomposition are already flattened, indicating that it is sufficient to apply LR to the lower right 8×8 sub-matrix of $\mathbf{R}$. The proposed method iterates from column 16 to column 9. The complete preprocessing algorithm, incorporating Cholesky sorted QR decomposition and PILR, is presented as Algorithm 2.

In [47], [48], the traditional LR and partial LR operations were proposed for optimizing the matrix $\mathbf{R}$. First, the conventional methods [47], [48] use a unit matrix $\mathbf{T}$ as a trace of all

Algorithm 2: Cholesky sorted QR decomposition and partial lattice reduction for K-best detection.

Inputs: $N \times N$ channel matrix $\mathbf{H}$; $N \times 1$ received signal $\mathbf{y}$; $N \times 1$ constant vector $\mathbf{v}$ with all elements equal to $1+1j$; parameter δ;
Outputs: matrix $\mathbf{R}$; vector $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$;
//Initialization:
  $\mathbf{A} = \mathbf{H}^H\mathbf{H}$; $\dot{\mathbf{y}} = 0.5 \times (\mathbf{y} - \mathbf{H}\mathbf{v})$;
//Sorted QR decomposition:
for $k = 1; k \leq N-1; k+1$ **do**
  Sort i as $a_{i,i} = \underset{d=k,k+1,k+2\dots N}{argmin}(a_{d,d})$;
  Swap the ith and kth columns of the matrix $\mathbf{H}$ to obtain $\dot{\mathbf{H}}$; Swap the ith and kth rows and the ith and kth columns of the matrix $\mathbf{A}$;
  $r_{k,k} = \sqrt{a_{k,k}}$;
  for $j = k; j \leq N; j+1$ **do**
    $r_{k,j} = a_{k,j}/r_{k,k}$;
  end for
  for $m = k+1; m \leq N; m+1$ **do**
    for $n = m; n \leq N; n+1$ **do**
      $a_{m,n} \leftarrow a_{m,n} - r^H_{k,m} r_{k,n}$;
    end for
  end for
end for
$r_{N,N} = \sqrt{a_{N,N}}$; //matrix $\mathbf{R}$ is obtained
//Partial iterative lattice reduction:
for $t = 1; t \leq T; t+1$ **do**
  for $k = N; k \geq N/2+1; k-1$ **do**
    if $\delta\,|r_{k-1,k-1}| > |r_{k,k}|$ **then**
      $\mu = r_{k-1,k}/r_{k-1,k-1}$;
      $r_{1:k,k} \leftarrow r_{1:k,k} - \mu r_{1:k,k-1}$; $\dot{h}_{:,k} \leftarrow \dot{h}_{:,k} - \mu \dot{h}_{:,k-1}$;
      Swap the kth and $(k-1)$th columns of the matrices $\mathbf{R}$ and $\dot{\mathbf{H}}$ to obtain $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$;
      $a = \frac{\hat{r}_{k-1,k-1}}{\sqrt{\hat{r}^2_{k-1,k-1}+\hat{r}^2_{k,k-1}}}$; $b = \frac{\hat{r}_{k,k-1}}{\sqrt{\hat{r}^2_{k-1,k-1}+\hat{r}^2_{k,k-1}}}$;
      $\theta = \begin{bmatrix} a^H & b^H \\ -b & a \end{bmatrix}$;
      Update the k and $k-1$ rows of the matrix $\hat{\mathbf{R}}$, $\begin{bmatrix} \hat{r}_{k-1,k-1} & \hat{r}_{k-1,k} & \dots \\ 0 & \hat{r}_{k,k} & \dots \end{bmatrix} \leftarrow \theta \times \begin{bmatrix} \hat{r}_{k-1,k} & \hat{r}_{k-1,k-1} & \dots \\ \hat{r}_{k,k} & 0 & \dots \end{bmatrix}$;
    end if
  end for
end for
//Full size reduction:
for $m = N-1; m \geqslant 2; m-1$ **do**
  for $n = m+1; n \leq N; n+1$ **do**
    $\mu = \lceil \hat{r}_{m,n}/\hat{r}_{n,n} \rfloor$;
    $\hat{r}_{1:m,n} \leftarrow \hat{r}_{1:m,n} - \mu \hat{r}_{1:m,m}$;
    $\hat{h}_{:,n} \leftarrow \hat{h}_{:,n} - \mu \hat{h}_{:,m}$;
  end for
end for
//Inversion of matrix $\hat{\mathbf{R}}$:
for $i = N; i \geqslant 2; i-1$ **do**
  $r^{\text{inv}}_{i,i} = 1/\hat{r}_{i,i}$;

Algorithm 2: Countinue.

for $j = i+1; j \le N; j+1$ **do**
  $r^{\text{inv}}{}_{i,j} = -\left(\sum_{k=j}^{N} \hat{r}_{i,k} r^{\text{inv}}{}_{k,j}\right) r^{\text{inv}}{}_{i,i}$;
end for
end for
//Post vector computation:
$\hat{\mathbf{y}} = (\mathbf{R}^{\text{inv}})^H \left(\hat{\mathbf{H}}^H \dot{\mathbf{y}}\right)$.

column adjustments, whereas here, the channel matrix $\mathbf{H}$ can be used directly. Second, when performing full size reduction, the original LR [35]–[39] and partial LR [47], [48] methods proceed from $\hat{r}_{1,1}$ to $\hat{r}_{k,k}$, element by element. The proposed algorithm works in the opposite direction, which makes it possible to operate row by row for improved parallelism. The high throughput achieved in the ASIC-based realization of the hardware architectural design presented later in Sections IV and V proves the high parallelism of the proposed method. Third, in conventional LR, adjustments are applied throughout the entire matrix $\hat{\mathbf{R}}$. By contrast, this paper combines the concepts of LR and sorted QR decomposition, which are both methods for adjusting the matrix $\hat{\mathbf{R}}$ to possess more advantageous properties. The proposed partial LR method proceeds from the last row to the $(N/2+1)$th row T times in an iterative way. Therefore, the proposed algorithm exploits this fact and combines both techniques to potentially reduce the total number of column swaps.

B. K-Best Detector and Performance Simulation

Based on the proposed preprocessing algorithm, a modified K-best detector with $K = 10$ is adopted in this paper. The detector consists of N stages for solving for the output signal $\hat{\mathbf{s}}$. In stage one, an approximate solution $\hat{s}_N$ is calculated as follows:

$$\hat{s}_N = \hat{y}_N / \hat{r}_{N,N}. \tag{23}$$

Then, the four nearest Gaussian integers to $\hat{s}_N$ in the 64-QAM constellation are obtained in a certain order, as shown in Fig. 3. These four nodes are used as the parent nodes for stage two, in ascending order based on their PEDs. In stage two, an interference cancellation is performed for the four parent nodes. Then, the parent node with the smallest PED is expanded to obtain four child nodes using the same method as in stage one, and the remaining three parent nodes are each expanded to obtain two further child nodes in a certain order, as shown in Fig. 3. After the resulting ten child nodes have been sorted in ascending order based on their PEDs, they become the parent nodes for stage three. Stages three to $N-1$ are similar in structure. First, an interference cancellation is performed for all ten parent nodes. Then, four child nodes are obtained from the parent node with the smallest PED, and two child nodes are obtained from each of the remaining parent nodes. Next, the ten child nodes with the smallest PEDs are selected as the parent nodes for the next stage. In the final stage, after the interference cancellation, only one child node is obtained from each parent node. Finally, the child node with the smallest PED is chosen as the final solution, and the path corresponding to this child node is selected as the output signal $\hat{\mathbf{s}}$. The search tree is illustrated in Fig. 4, where a 16×16 MIMO system is considered as an example.

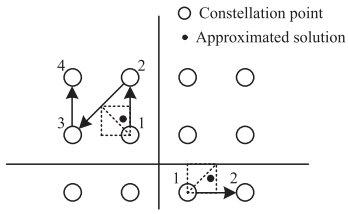


Fig. 3. Enumeration method for constellation points.

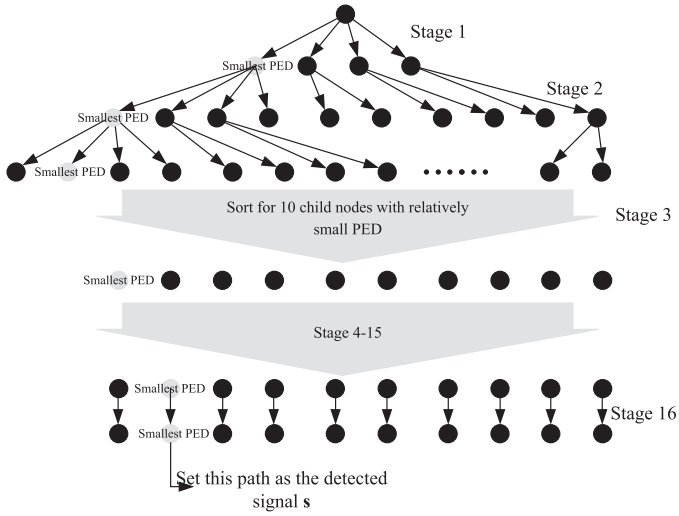
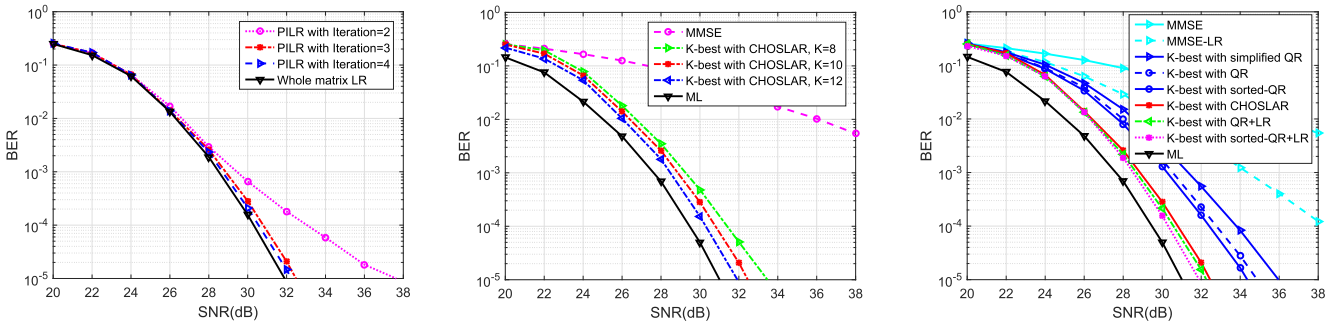


Fig. 4. Tree expansion for the K-best detector.

To evaluate the performance of the proposed method, simulations of bit error rate (BER) performance were conducted. In all simulations, large-scale MIMO systems were considered. In addition, the 64-QAM modulation scheme and a rate-1/2 industry standard convolutional code with a $[133_o\ 171_o]$ polynomial were employed, along with a random interleaver [7], [8], [11]. The coding was performed over 120 symbols, and the number of frames was 100,000. The channels were assumed to exhibit i.i.d. Rayleigh fading across the coded symbols, and the signal-to-noise ratio was defined at the receiver. Fig. 5–(a) compares the BER performances of the full matrix LR and PILR methods with different numbers of iterations for the case of a 16×16 MIMO system. According to Fig. 5–(a), three iterations are sufficient to achieve near-optimal performance with a sorting-reduced K-best algorithm with $K = 10$. Hence, when the proposed lower-right 8×8 PILR method is customized for a 16×16 MIMO system, the method proceeds from the last row to the ninth row three times in an iterative way. Fig. 5–(b) compares the BER performances for different values of K ($K = 8$, 10, and 12) in the proposed K-best algorithm with CHOSLAR preprocessing. According to Fig. 5–(b), the CHOSLAR ($K = 10$) algorithm suffers from acceptable performance losses of 1.44 dB and 0.53 dB (for a BER target of 10^{-5}) compared with the ML and CHOSLAR ($K = 12$) algorithms, respectively. However, when $K = 8$, there is a 3.46 dB performance loss compared with the ML algorithm. This is why



(a) (b) (c)

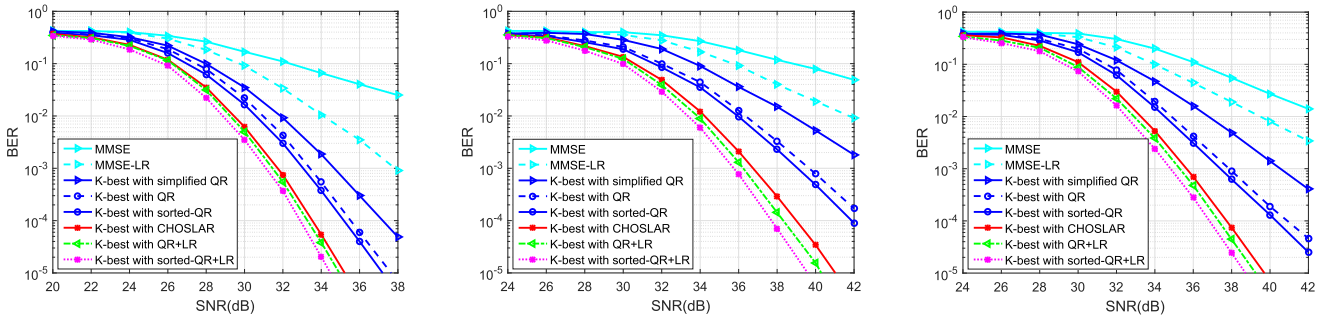
Fig. 5. BER performance comparisons for (a) PILR and full matrix LR (iteration numbers $t = 2, 3, 4$), (b) different values of K, and (c) different algorithms in a 64-QAM 16 $\times$ 16 MIMO system (K=10).

the value of $K = 10$ is adopted in this paper. Fig. 5–(c) compares the simulated BER results obtained for different detection algorithms ($K = 10$) in a 64-QAM 16 $\times$ 16 MIMO system. Compared with the ML algorithm, the performance loss of the proposed K-best algorithm with CHOSLAR preprocessing for a BER target of 10^{-5} is 1.44 dB, which is close to the performance losses for the K-best algorithm with sorted QR decomposition and LR (0.89 dB) and the K-best algorithm with QR decomposition and LR (1.2 dB) [34]. In addition, the performance losses of the proposed method are competitive with those of the K-best algorithm with sorted QR decomposition (3.37 dB) [23], [26], the K-best algorithm with QR decomposition (3.88 dB) [22], the K-best algorithm with simplified QR decomposition (WLD) (4.96 dB) [30], [32], the MMSE-LR method (more than 8 dB) [21], [35], [36], and the MMSE method (more than 8 dB). Note that an interference cancellation set of size two is adopted in the simulation of the WLD method. Moreover, Fig. 5–(c) demonstrates that the performance achieved with sorted QR decomposition and LR combined is much better than that achieved using only sorted QR decomposition. Note that the simulation results are based on 64-QAM, which is not supported by the TASER algorithm [40].

All simulations presented above are based on 16 $\times$ 16 MIMO systems and indicate that the proposed CHOSLAR algorithm is advantageous in terms of BER performance. To confirm that the proposed method maintains its advantages for higher-order MIMO systems and different modulation types, Fig. 6 shows the BER performances in different simulation settings. Fig. 6–(a) and Fig. 6–(b) compare the BER performances for 64 $\times$ 64 and 128 $\times$ 128 MIMO systems with 64-QAM. Note that the complexity of the ML detection algorithm is prohibitively high in 64 $\times$ 64 and 128 $\times$ 128 MIMO systems. Therefore, no ML detection results are presented in these figures. According to Fig. 6–(a), the proposed algorithm suffers an SNR loss of 0.77 dB compared with the K-best algorithm with sorted QR decomposition and LR in a 64 $\times$ 64 MIMO system. The corresponding value for a 128 $\times$ 128 MIMO system is 1.41 dB, as shown in Fig. 6–(b). In addition, the proposed method achieves a higher BER performance than those of the methods reported in [21], [23], [26], [30], [32], [35], [36]. These figures show that the proposed CHOSLAR algorithm maintains its advantages in higher-order MIMO systems. Fig. 6–(c) shows the BER performances for a high-order modulation scheme (256-QAM). The value of K used in the K-best algorithm was 14 in these simulations. According to Fig. 6–(c), the proposed CHOSLAR algorithm suffers only a 1.01 dB performance loss while maintaining its advantages.

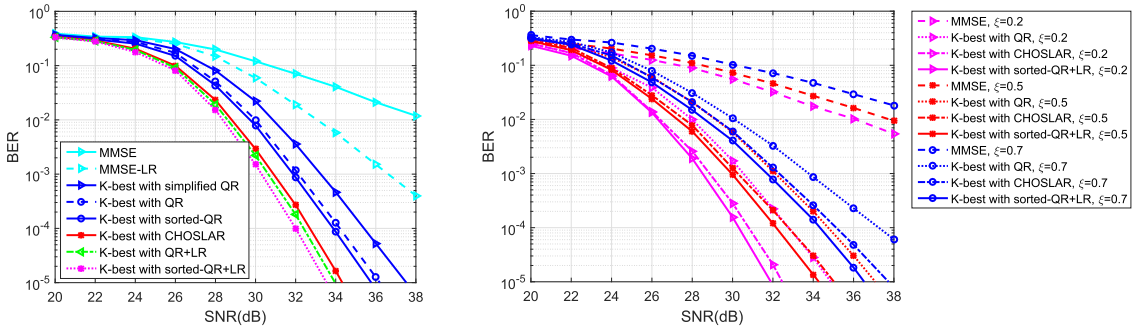
Notably, although only symmetric MIMO systems have been discussed thus far in relation to the proposed method, the method is also suitable for non-symmetric MIMO systems, in which the number of transmitters on the BS side is larger than that on the user side [1], [2]. Throughout the entire algorithm, QR decomposition is transformed into Cholesky decomposition, and the number of transmitters on the BS side affects only the initialization stage (line 4 of Algorithm 2) and the update and column swap processes for the matrix $\mathbf{H}$ (lines 8, 25, 26, and 38 of Algorithm 2). Hence, as the number of transmitters on the BS side increases (above the number of transmitters on the user side), these elements of the processing will be influenced. Initialization can still be achieved because of the simple matrix and vector multiplications. In addition, the update and column swap processes for the matrix $\mathbf{H}$ can also be achieved because these processes are based on single columns of the matrix $\mathbf{H}$. Thus, the proposed CHOSLAR algorithm is still suitable in the case of a non-symmetric MIMO system. A BER performance comparison for a non-symmetric (16 $\times$ 32) MIMO system is shown in Fig. 7–(a). The results indicates that the proposed CHOSLAR algorithm is suitable for a non-symmetric MIMO system and that its advantages are maintained in this case.

To better present how different channel characteristics influence the algorithm and the simulation results, a performance evaluation was also performed using the Kronecker channel model [11], [49]. In this model, the elements of the channel matrix follow a distribution of the form $\mathcal{CN}(0, d(\mathbf{z})\mathbf{I}_B)$, where $d(\mathbf{z})$ represents the channel attenuation (such as path loss and shadowing). The classical path loss model was used, with a channel attenuation variance of $d(\mathbf{z}) = \frac{C}{\|\mathbf{z}-\mathbf{b}\|^{\kappa}}$, where $\mathbf{z} \in \mathbb{R}^2$, $\mathbf{b} \in \mathbb{R}^2$, and κ denote the location of the user, the location of the BS, and the path loss exponent, respectively. The independent shadow fading represented by C satisfied $10\lg C \sim \mathcal{N}(0, \sigma_{sf}^2)$. Another distinct feature of the Kronecker channel model is its consideration of channel correlations; specifically, $\mathbf{R}_r$ and $\mathbf{R}_t$ represent



(a) 64×64 MIMO, 64-QAM (b) 128×128 MIMO, 64-QAM (c) 16×16 MIMO, 256-QAM

Fig. 6. BER performance comparison for different configurations in terms of the numbers of antennas and users (assuming a symmetric MIMO system) and different modulation types.



(a) (b)

Fig. 7. BER performance comparisons for (a) a non-symmetric (16 × 32) MIMO system and (b) the Kronecker channel model (16 × 16 MIMO, 64-QAM).

the channel correlations of the receiving and transmitting antennas, respectively. The exponential correlation model [11], [49] was adopted, with ξ representing the correlation factor. Hence, the channel matrix $\mathbf{H}$ could be modeled as

$$\mathbf{H} = \mathbf{R}_r^{1/2}\mathbf{H}_{i.i.d.}\sqrt{d(\mathbf{z})}\mathbf{R}_t^{1/2}, \tag{24}$$

where $\mathbf{H}_{i.i.d.}$ is a random matrix whose entries are i.i.d. with a complex Gaussian distribution of zero mean and unit variance. In this simulation, each hexagonal cell had a radius of $r = 500$ meters, with independent and random user locations of $\mathbf{z} \in \mathbb{R}^2$. The following assumptions were adopted for the simulation: $\kappa = 3.7, \sigma_{sf}^2 = 5$, and a transmission power of $\rho = r^\kappa/2$. Fig. 7–(b) compares the BER performances of the different algorithms for the Kronecker channel model with three values of the correlation factor ($\xi = 0.2$, 0.5, and 0.7). According to Fig. 7–(b), the CHOSLAR algorithm maintains its advantages for this realistic model. Note that throughout the paper, the performance simulations and computational complexity are compared with those of similar references in which no soft output was generated to achieve a fair comparison.

C. Summary and Analysis of the CHOSLAR Algorithm

This section compares the computational complexity and parallelism of CHOSLAR with those of other algorithms (GS, GR, and HT), which are summarized in detail in [17]. The analysis shows that the majority of the computational complexity is due to the QR decomposition and LR processes. A complex-valued system is considered in this paper; therefore, the computational complexity of the QR decomposition algorithms is evaluated in terms of the required number of real-valued operations. The computational complexity of QR decomposition is dominated by real-valued multiplications (RMUL) and real-valued additions (RADD). As in [29]–[32], it is assumed that real-valued division and square-root operations are each equivalent to a RMUL. In addition, a complex multiplication operation requires 4 RMUL and 2 RADD, whereas a complex addition operation requires 2 RADD. Table I shows the numbers of real-valued operations required for the GS, GR, HT and Cholesky sorted QR algorithms. The proposed Cholesky sorted QR algorithm includes two main components: the matrix multiplication of $\mathbf{H}^H\mathbf{H}$ and the decomposition of the Gram matrix $\mathbf{A}$. The matrix multiplication of $\mathbf{H}^H\mathbf{H}$ involves $2N^3 + 2N^2$ RMUL and $N^3 - N$ RADD because it requires conjugate symmetric matrix multiplication. The Cholesky decomposition of the matrix $\mathbf{A}$

TABLE I
COMPUTATIONAL COMPLEXITIES OF DIFFERENT DETECTION ALGORITHMS FOR LARGE-SCALE MIMO SYSTEMS

Algorithm	Real-valued additions	Real-valued multiplications
Gram-Schmidt	$4N^3 + N^2 - 2N$	$\frac{14}{3}N^3 + 4N^2 + N$
Givens Rotation	$4N^3 + \frac{15}{2}N^2 - \frac{23}{2}N$	$\frac{18}{3}N^3 + \frac{23}{2}N^2 - \frac{107}{6}N$
Householder Transformation	$2N^4 + 5N^3 + \frac{21}{2}N^2$	$\frac{8}{3}N^4 + \frac{22}{3}N^3 + 14N^2$
Cholesky Sorted QR	$2N^3 - 3N^2 + 2N$	$\frac{11}{3}N^3 + 5N$

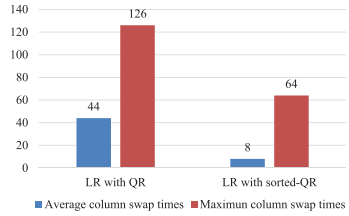


Fig. 8. Comparison of column swapping for LR with and without sorting.

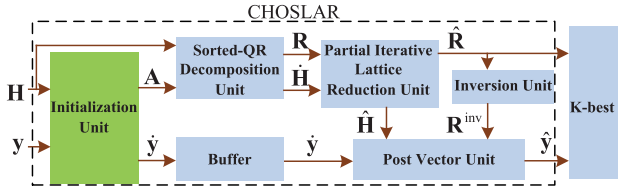


Fig. 9. High-level block diagram of CHOSLAR. The elements of the matrices and vectors take complex values.

requires $N^3 - 2N^2 + 5N$ RMUL and $N^3 - 3N^2 + 3N$ RADD. Note that the calculation of $\mathbf{Q}$ is implicit in GR; therefore, its contribution to the computational complexity is omitted in Table I. To perform the sorting operation in each of these methods, $\frac{2}{3}N^3$ additional RMUL are required [28]. As shown in Table I, compared with the other algorithms, the complexity of the proposed Cholesky sorted QR decomposition method is reduced. For example, when $N = 16$, the number of RMUL required is reduced by 25.1%, 44.6%, and 93.2% compared with the GS, GR, and HT methods, respectively. In addition, the number of RADD is reduced by 55.1%, 58.9%, and 95.2% compared with the GS, GR, and HT methods, respectively.

Fig. 8 presents simulated results for the average and maximum numbers of column swaps for LR with and without sorting in a 16 × 16 MIMO system and shows that the number of column swaps is reduced by applying sorted QR decomposition. The proposed constant-throughput PILR method requires three iterations, and each iteration requires 8 matrix swaps. Hence, there are 24 matrix swaps in total, which is less than the 44 matrix swaps required for whole-matrix LR (a 45.5% reduction). In addition, row updating requires fewer multiplications because the number of matrix swaps is reduced and PILR needs to be performed only in the lower right corner of the small triangular matrix. The average numbers of multiplications in the LR and PILR methods are 3960 and 1296, respectively, in a 16 × 16 MIMO system, which means that the proposed PILR method reduces the number of multiplications by 67.3%. In the K-best calculation, only 33 comparators are required in each stage of the proposed K-best method to achieve the same detection accuracy achieved with 216 comparators in each stage without sorting. Hence, the number of comparators in each stage is reduced by 84.7%.

Considering the parallelism of the hardware implementation, the Cholesky sorted QR decomposition method proposed in this paper can be used to eliminate an entire column of the matrix $\mathbf{A}$. By contrast, the commonly used CORDIC-pair-based GR method can only eliminate one element of the matrix $\mathbf{H}$ at a time. Moreover, before the elimination of a new column can begin, the elements in the column to the left must be eliminated. Hence, the correlation of these eliminations limits the parallelism, especially as the numbers of antennas and users increase in higher-order MIMO systems. For example, the CORDIC-pair-based GR method requires four rounds to eliminate all elements in the first column of a 16 × 16 matrix. Therefore, compared with conventional QR decomposition, the proposed Cholesky sorted QR decomposition method achieves higher parallelism and lower preprocessing latency. In addition, the parallelism of the size reduction process in LR is also improved because the column updating of the matrices $\mathbf{H}$ and $\mathbf{R}$ is achieved by row instead of by element.

IV. VLSI Architecture

This section describes a VLSI architecture for realizing the proposed CHOSLAR algorithm. As a concrete example, the architecture is designed for a 64-QAM 16 × 16 MIMO system. The circuit design method for other, larger-scale MIMO systems is similar.

Fig. 9 shows the high-level block diagram of the CHOSLAR algorithm, which consists of five main units, namely, the initialization unit, the sorted QR decomposition unit, the PILR unit, the inversion unit and the post vector unit. These units are fully pipelined to achieve high throughput. First, the initial results (the Gram matrix $\mathbf{A}$ and the vector $\dot{\mathbf{y}}$) listed on line 2 of Algorithm 2 are computed in the initialization unit. Second, the output of the initialization unit, the Gram matrix $\mathbf{A}$, is used to perform the sorted QR decomposition described on lines 2 to 2 of Algorithm 2. The contents of the channel matrix $\mathbf{H}$ are simultaneously swapped to obtain $\dot{\mathbf{H}}$, as described on line 2 of Algorithm 2. Third, the matrix $\mathbf{R}$ is used to perform PILR and obtain the resulting matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$ as described on lines 2 to 2 of Algorithm 2. The matrix $\hat{\mathbf{R}}$ is one of the outputs of CHOSLAR, and the matrix $\hat{\mathbf{H}}$ is passed to the post vector unit. Fourth, the matrix $\hat{\mathbf{R}}$ is inverted in the inversion unit as described on lines 2 to 2 of Algorithm 2. Finally, in the post vector unit, the outputs of previous steps (the matrices $\hat{\mathbf{H}}$ and $\mathbf{R}^{\text{inv}}$ and the vector $\dot{\mathbf{y}}$) are used to obtain the final output $\hat{\mathbf{y}}$ of the CHOSLAR algorithm by means of matrix-vector multiplications. The processing and architectural details of each unit will be discussed in the following subsections.

A. Initialization Unit

In the initialization unit, the main goal is to compute the Gram matrix $\mathbf{A}$ and the vector $\dot{\mathbf{y}}$, which are used in the subsequent units. To achieve high throughput, a systolic array that includes two types of processing elements (PEs) is designed. There are N PE-As and $(1/2N^2 - 1/2N)$ PE-Bs (e.g., 16 PE-As and 120 PE-Bs for a 16 × 16 MIMO system) in the systolic array. The next unit (the sorted QR decomposition unit) requires the diagonal elements of the matrix $\mathbf{A}$ for comparison. Hence, the PE-As, whose purpose is to obtain these diagonal elements, comprise the first block in each row, as shown in Fig. 10. In addition, $(N - 1)$ registers (REGs) store the elements of the

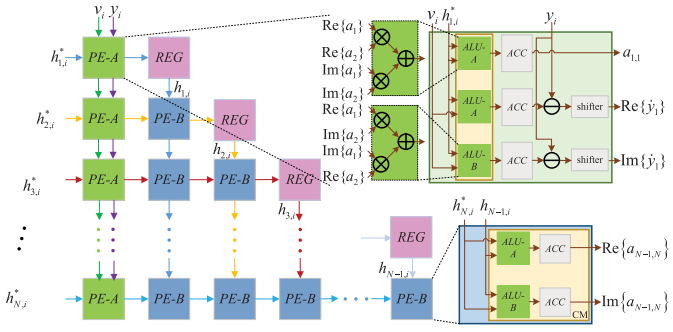


Fig. 10. Architecture of the initialization unit. The real and imaginary parts are individually depicted in this figure to show the details of the PE-A, PE-B and CM units. In other figures, the complex values are not divided into their real and imaginary parts for convenience.

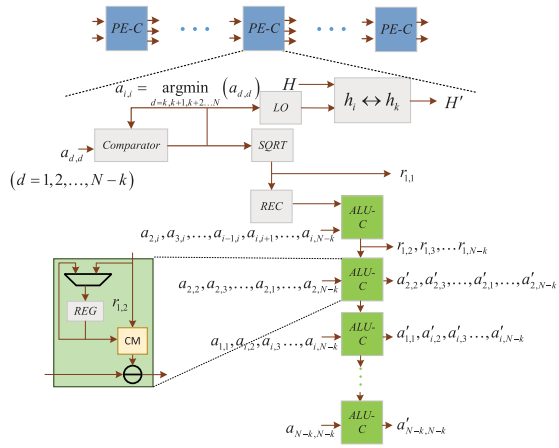


Fig. 11. Architecture of the sorted QR decomposition unit. The element $a'_{i,j}$ is the updated value of $a_{i,j}$, as shown on line 8 of Algorithm 2.

channel matrix and obtain the conjugate of each element during output. These explicit REGs are used to balance the schedule of the pipeline. The REGs balance the latency, which arises from the computations of the PE-Bs, among all parallel rows. The first processing elements, the PE-As, are used to compute the vector $\dot{\mathbf{y}}$, which consists of the diagonal elements of the Gram matrix $\mathbf{A}$. There are two types of arithmetic logical units (two ALU-As and an ALU-B), three accumulators (ACC), two subtractors and two shifters in each PE-A. One ALU-A performs the complex-valued multiplications of $h^*_{i,j}$ and $h_{i,j}$, and the result of each cycle is accumulated. The other ALU-A, in combination with the ALU-B, performs the complex-valued multiplications of the matrix $\mathbf{H}$ and the vector $\mathbf{v}$. Similarly to the computation of the elements of $\mathbf{A}$, the results of this ALU-A/ALU-B pair are accumulated. To perform line 4 of Algorithm 2, the results are subtracted from the elements of $\mathbf{y}$; then, the shifter completes the computation of the real and imaginary parts of the vector $\dot{\mathbf{y}}$. The subsequent processing elements, the PE-Bs, are used to compute the off-diagonal elements of the matrix $\mathbf{A}$. Each PE-B contains one complex multiplication (CM) unit, which includes one ALU-A and one ALU-B. To ensure that each processing element processes the correct set of operands, the values in the ith column of $\mathbf{H}^H$ are delayed by $(i-1)$ clock cycles. First, each value of $\mathbf{H}^H$ is transferred from a PE-A to the subsequent PE-B and then to the REG (by row); then, the corresponding conjugate values are transferred from the REGs to the PE-Bs (by column).

Similar systolic arrays were used to compute the Gram matrix for the linear detection methods applied in [9], [10]. Unlike in the proposed systolic array, the computations in these architectures do not begin with the PEs that compute the diagonal elements of $\mathbf{A}$ [9], [10]. Hence, the computations of the diagonal elements of the Gram matrix $\mathbf{G}$ are delayed, which means that the subsequent sorted QR decomposition unit must wait longer to receive its inputs. Therefore, the throughput of the overall architecture is reduced, and the latency is increased. In [9], [10], double-sided inputs were used for the PE-As, whereas in the proposed architecture, single-sided inputs are used. Thus, the number of ports of the systolic array used in [9], [10] is increased by half (on the input side).

B. Sorted QR Decomposition Unit

After the initialization unit, the output matrix $\mathbf{A}$ is transferred to the next unit, which performs sorted QR decomposition based on Cholesky decomposition to obtain the matrix $\mathbf{R}$ (as shown in Fig. 11). The channel matrix $\mathbf{H}$ is also updated in this unit. To achieve high parallelism, a deep pipelined architecture is proposed for this unit, which includes N similar processing elements, PE-Cs (e.g., 16 PE-Cs for a 16 $\times$ 16 MIMO system). All of the PE-Cs are similar, but the number of ALU-Cs in each PE-C is different. The number of ALU-Cs in each PE-C is reduced from the first PE-C to the final PE-C in the column. The kth PE-C is taken as an example to describe the architecture in detail. First, a comparator is used to compare all diagonal elements of the matrix $\mathbf{A}$ and to find the smallest $a_{i,i}$ and its location (LO) in $\mathbf{A}$. Second, in the square root (SQRT) unit, $a_{i,i}$ is used to compute $r_{1,1}$, which is one element of the output, namely, the kth column of the matrix $\mathbf{R}$. In addition, in accordance with the location of $a_{i,i}$, the ith and kth columns of $\mathbf{H}$ are swapped to obtain $\dot{\mathbf{H}}$, as shown on line 15 of Algorithm 2. Third, the reciprocal (REC) unit is used to compute the reciprocal of $r_{1,1}$, and the result is transferred to the first ALU-C to obtain further results for the kth column of $\mathbf{R}$, as shown on lines 2 to 2 of Algorithm 2. Finally, in accordance with the matrices $\mathbf{R}$ and $\mathbf{A}$, the elements of $\mathbf{A}$ are updated and transferred to the $(k+1)$th PE-C, as shown on line 2 of Algorithm 2. In the ALU-C, multipliers and subtractors perform the element computations for the matrix $\mathbf{A}$ that are described on lines 2 to 2 of Algorithm 2. Note that the diagonal elements of the matrix $\mathbf{A}$ are the first to be computed in this architecture, which means that the results can be used by the next PE-C. Hence, the latency is reduced in this sorted QR decomposition unit.

In [26], [34], VLSI architectures were proposed based on the GR method in combination with sorting algorithms. In [26], a flexible architecture was proposed for 64-QAM $1 \times 1 \sim 4 \times 4$ MIMO systems. In [34], an 8×8 sorting-reduced K-best detector with LR and QR decomposition was proposed. The sorted QR decomposition units in these architectures were each

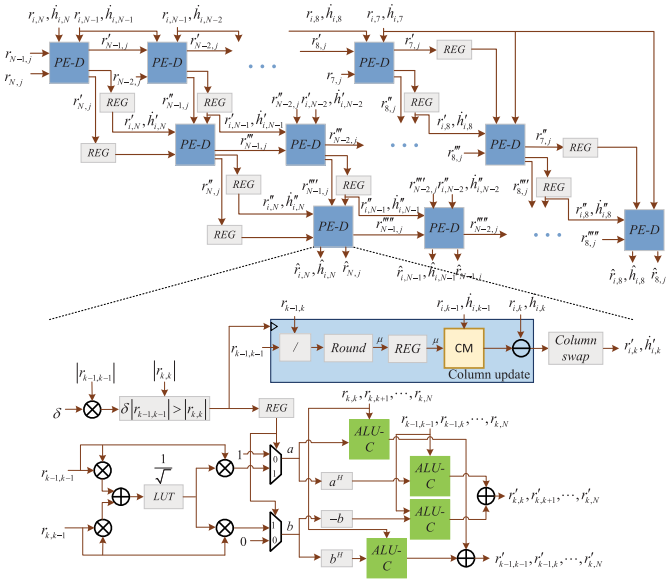


Fig. 12. Architecture for the updating of the matrix $\mathbf{R}$ based on the Siegel condition. The elements $r'_{i,j}$, $r''_{i,j}$, $r'''_{i,j}$, etc. are intermediate results from different iterations.

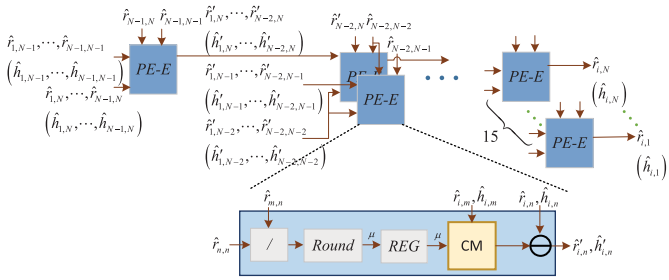


Fig. 13. Architecture for the full size reduction of the matrix $\mathbf{R}$. The elements $\hat{r}'_{i,j}$, $\hat{h}'_{i,j}$, etc. are intermediate results from different steps of the full size reduction.

constructed as a long chain of duplicated CORDIC arrays, resulting in excessive latency; this can be a serious problem, especially as the size of the MIMO system increases. The increased computation time required for sorted QR decomposition in turn influences the overall detector throughput. By contrast, the architecture proposed here does not require GR or HT for QR decomposition. The decomposition is achieved via a series of multiplications that can be performed by a deep pipeline to satisfy the high-throughput requirements of future wireless communication systems.

C. PILR Unit

The PILR unit has two main functions, as shown in Fig. 12 and Fig. 13. The first is to update the matrix $\mathbf{R}$ based on the Siegel condition, and the second is to perform the full size reduction of the matrix $\mathbf{R}$. The matrices $\mathbf{H}$ and $\mathbf{R}$ are updated simultaneously.

In the first component of the unit (Fig. 12), there are $3N$ PE-Ds (e.g., 48 PE-Ds for a 16×16 MIMO system). All of the PE-Ds are similar and operate in parallel. The kth PE-D is illustrated as an example. The input to the first PE-D is the kth and $(k-1)$th rows of the matrix $\mathbf{R}$, and the PE-D updates these two rows. Then, these two rows are used as the input to the next PE-D. Note that REGs ensure the schedule of the pipeline. The architecture of the PE-D array is shown in Fig. 12. Each PE-D has three main components, for the column updating of $\dot{\mathbf{H}}$ and $\mathbf{R}$, the column swapping of $\dot{\mathbf{H}}$ and $\mathbf{R}$, and the updating of $\mathbf{R}$. Before these components, there are processes for performing the comparison on line 2 in Algorithm 2. The result of this comparison is used as the enable signal for the subsequent processing. For the column updating process, the parameter μ is computed first. The division is achieved by means of a look-up table (LUT). Then, a CM unit is used to perform the multiplication on line 2 in Algorithm 2, and the resulting matrices $\mathbf{R}$ and $\mathbf{H}$ are swapped by column to obtain the matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$. Then, to update $\hat{\mathbf{R}}$, the parameters a and b in the matrix θ are computed. As part of this process, real-valued multipliers and adders and an LUT are used to perform the necessary multiplication, square root and reciprocal operations on the elements of the matrix $\hat{\mathbf{R}}$. Multiplexers, conjugates, and negative units are used to obtain θ, and the kth and $(k-1)$th rows of the matrix $\hat{\mathbf{R}}$ are updated.

In the second component of the PILR unit, there are $(1/2N^2 - 1/2N)$ identical processing elements, PE-Es (e.g., 120 PE-Es for a 16×16 MIMO system). The architecture for the size reduction of the matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$ is shown in Fig. 13. A single PE-E is illustrated as an example. The PE-E first computes the parameter μ; then, each element of the $(N-2)$th column in matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$ is multiplied by μ. The results are subtracted from each element of the $(N-1)$th column. In this size reduction architecture, the purpose of the first stage is to update the Nth column of matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$ (except for the element in the $(N-1)$th row of the Nth column). In the second stage, the results of the first stage are used to update the Nth column of matrices $\hat{\mathbf{R}}$ and $\hat{\mathbf{H}}$ (except for the elements in the $(N-1)$th and $(N-2)$th rows of the Nth column). In addition, the $(N-1)$th and $(N-2)$th columns are input into the PE-E to update the $(N-1)$th column. The size reduction architecture includes $(N-1)$ stages, and the calculation methods are the same in all subsequent stages.

One VLSI architecture has previously been proposed that includes a similar LR procedure as part of the detection preprocessing. In [34], 3 pairs of CORDIC processors were used to implement LR via an odd-even method. The detector achieved near-ML accuracy for a 64-QAM 8×8 MIMO system. Considering a VLSI implementation, a few CORDIC pairs were designed for both QR and LR, which contributed the majority of the latency according to the timing schedule. Hence, the throughput of the entire detector was decreased. By contrast, the PILR unit proposed here performs LR via two architectures, one for applying the Siegel condition and one for performing size reduction. These architectures are designed based on systolic arrays. All intermediate data are computed in the next PE-D, and all computations are deeply pipelined. The hardware utilization and throughput are higher than those of the CORDIC-processor-based architecture presented in [34].

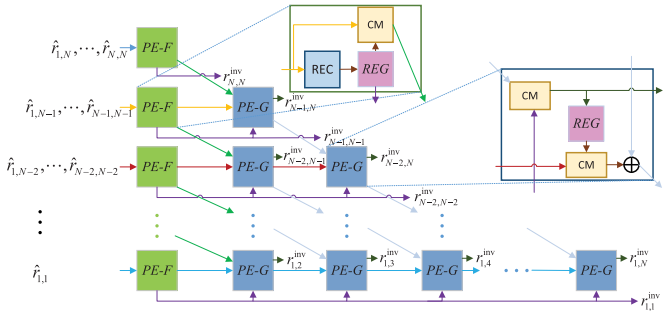


Fig. 14. Architecture of the inversion unit.

D. Inversion Unit

A systolic array is proposed for the inversion of the matrix $\hat{\mathbf{R}}$ in the CHOSLAR architecture, as shown in Fig. 14. There are two types of PEs, namely, N PE-Fs and $(1/2N^2 - 1/2N)$ PE-Gs (e.g., 16 PE-Fs and 120 PE-Gs for a 16×16 MIMO system). The PE-Fs and PE-Gs are used to compute the diagonal and off-diagonal elements, respectively, of the matrix $\mathbf{R}^{\text{inv}}$. The PE-Fs comprise the first PE in each row because the diagonal elements of $\mathbf{R}^{\text{inv}}$ are required to compute the off-diagonal elements. To ensure that each processing element processes the correct set of operands, the values in each column of $\hat{\mathbf{R}}$ are delayed by $(i-2)$ clock cycles, and each value in the matrix $\mathbf{R}$ is transferred from a PE-F to the subsequent PE-G (by row). The third PE-F and PE-G are taken as examples. The PE-F contains one REC, one REG, and one CM unit, and the PE-G includes two CM units, one REG and one adder. The diagonal elements are computed by the REC and are output to the REG to be reused in the PE-Gs many times. In the next cycle, the off-diagonal elements of $\hat{\mathbf{R}}$ are transferred to the PE-F, and the diagonal elements of $\hat{\mathbf{R}}$ are transferred to the next PE-G in the row. The PE-F performs the multiplications of the diagonal elements of $\hat{\mathbf{R}}$ and the off-diagonal elements of $\mathbf{R}^{\text{inv}}$, and the results are transmitted to the PE-G to the lower right. The PE-G uses the results from the upper left PE, the diagonal elements of $\hat{\mathbf{R}}$ from the left PE and the diagonal elements of $\mathbf{R}^{\text{inv}}$ to compute the off-diagonal elements of $\mathbf{R}^{\text{inv}}$ as shown on line 2 of Algorithm 2.

Previously proposed linear detectors with VLSI architectures, such as those presented in [9], [10], have also included inversion units. The inversion units in these architectures were implemented in an approximate manner based on the Neumann series method. These inversion units are similar to the one proposed here (which is designed based on a systolic array). However, the proposed architecture exactly inverts the matrix $\mathbf{R}$, whereas the units presented in [9], [10] suffer from approximation errors. In addition, the proposed systolic array first inverts the diagonal elements of $\mathbf{R}$ in the PE-Fs, the first column of PEs in the unit. Hence, the results can subsequently be used in the PE-Gs. In the units proposed in [9], [10], the diagonal elements are computed only after a long delay because the computation does not begin with the PEs that compute these elements. Furthermore, the architectures presented in [9], [10] have more ports than the proposed architecture.

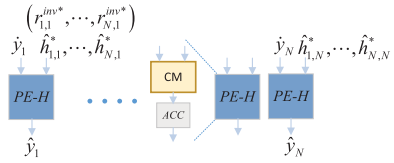


Fig. 15. Architecture of the post vector unit.

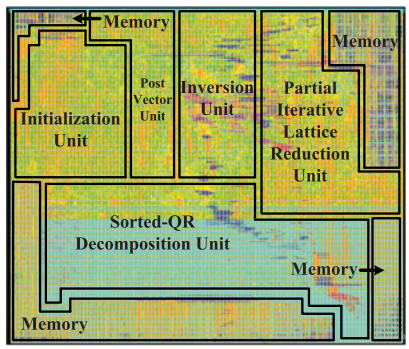


Fig. 16. The ASIC layout of the CHOSLAR implementation (for a 64-QAM 16×16 MIMO system).

E. Post Vector Unit

The post vector unit performs the multiplications of the matrices $(\mathbf{R}^{\text{inv}})^H$ and $\hat{\mathbf{H}}^H$ and the vector $\dot{\mathbf{y}}$ that are described on line 2 of Algorithm 2. Note that the matrix $\hat{\mathbf{H}}^H$ is output by the PILR unit, whereas the matrix $(\mathbf{R}^{\text{inv}})^H$ is obtained from the inversion unit. The multiplications are performed by first multiplying the matrix $\hat{\mathbf{H}}^H$ and the vector $\dot{\mathbf{y}}$. Then, the results are multiplied by the matrix $(\mathbf{R}^{\text{inv}})^H$ to obtain the result $\hat{\mathbf{y}}$. In addition, because $(\mathbf{R}^{\text{inv}})^H$ and $\hat{\mathbf{H}}^H$ are computed successively, the two matrix-vector multiplications can also be performed successively. Hence, the resources for matrix-vector multiplications can be reused. The architecture of the post vector unit is shown in Fig. 15. The unit contains N PE-Hs, each of which computes one element of the result vector. Each PE-H contains one CM unit and one ACC for complex-valued multiplications and accumulation, respectively.

V. Implementation Results and Comparison

The layout of the proposed VLSI architecture was prepared using TSMC 65 nm 1P8M technology. The experimental results obtained using this ASIC implementation are detailed in this section and compared with the results of various state-of-the-art nonlinear detectors. Fig. 16 shows the ASIC layout, and Table II presents a detailed comparison of post-layout simulation results based on the hardware features of the CHOSLAR architecture and the designs presented in [17], [22], [23], [26], [32], [34], [40], which are efficient, state-of-the-art ASIC architectures for preprocessing for nonlinear detection in either small-scale or large-scale MIMO systems.

The proposed architecture achieves a throughput of 3.528 Gbps, higher by 5.16×, 9.59×, 1.34×, and 4.81× than the throughputs of the designs presented in [22], [23], [26], and [32], respectively, which are designed for small-scale MIMO systems. The authors believe that the throughput achieved

TABLE II
ASIC IMPLEMENTATION RESULTS AND COMPARISON WITH OTHER REPORTED MIMO DETECTORS

	[34]	[26]	[22]	[23]	[32]	[40]			This work	
Antenna Size	Complex 8×8	Complex 4×4	Complex 4×4	Complex 4×4	Complex 4×4	Complex 4×4	Complex 8×8	Complex 16×16	Complex 16×16	
Modulation	64-QAM	64-QAM	64-QAM	64-QAM	256-QAM	QPSK			64-QAM	
Output	Hard	Soft	Hard	Hard	Soft	Hard			Hard	
Algorithm	GR+LR	Sorted GR	GS	Sorted QR	WLD	TASER			Sorted QR	Sorted QR+LR
SNR loss[a] [dB]	1.2	3.37	3.88	3.37	4.96	-			3.37	1.44
Process [nm]	90	65	90	65	90	40			65	
Voltage [V]	1.1	1.2	1	1.2	-	1.1			1.2	
Frequency [MHz]	65	550	114	500	275	598	560	454	588	
Throughput [Mbps]	585	2640	684	367.88	733	298	374	363	3528	
Latency [μs]	2.876	1.26	0.28	-	-	-	-	-	0.7	1.2
Gate Count [kG]	978	1529	505	1055	1580	148	471	1428	3720	5681
Power [mW]	37.1	429	56.8	307	320.56	41	87	216	1831	2513
Energy Efficiency[b] [Gbps/W]	15.77	6.15	12.04	1.20	2.29	7.27	4.30	1.68	1.93	1.40
Area Efficiency[b] [Mbps/kG]	0.60	1.73	0.73	0.35	0.46	2.01	0.79	0.25	0.95	0.62
Normalized Energy Efficiency [Gbps/W]	**6.35[cd]**	**0.38[d]**	**1.00[cd]**	**0.07[d]**	**0.27[cd]**	**0.14[cd]**	**0.34[cd]**	**0.54[c]**	**1.93**	**1.40**
Normalized Area Efficiency [Mbps/kG]	**0.21[cd]**	**0.11[d]**	**0.12[cd]**	**0.02[d]**	**0.04[cd]**	**0.08[cd]**	**0.12[cd]**	**0.16[c]**	**0.95**	**0.62**

[a] SNR loss compared with the ML detection algorithm in a 16×16 MIMO system with 64-QAM (BER target of 10^{-5}).
[b] Energy efficiency is defined as throughput/power; area efficiency is defined as throughput/gate count.
[c] Technology normalized to 65 nm CMOS technology under the full-scaling approach as follows: $f \sim s$ and $P_{dyn} \sim (1/s)(V_{dd}/V_{dd}^{'})^2$.
[d] Scaled to a 16×16 MIMO configuration: energy efficiency×$(N \times N)/(16\times16)$ and area efficiency×$(N \times N)/(16\times16)$.

using the CHOSLAR approach will be more suitable for meeting the requirements of future communication systems, such as 5G. However, this high throughput requires substantial hardware resources and power consumption. Hence, the area and energy efficiencies are also compared with those of recent designs. Note that the CHOSLAR algorithm is designed for application in large-scale MIMO systems, while the previous designs proposed in [17], [22], [23], [26], [32], [34] will incur much higher resource and power implementation costs in large-scale MIMO systems. In addition, different technologies and MIMO configurations are used in these designs. Hence, to ensure a fair comparison, the energy and area efficiencies have been normalized to 65 nm technology and a 16 × 16 MIMO configuration, as shown in Table II. Such normalization methods are widely used when comparing implementation results for different technologies and MIMO configurations, such as in [22], [23], [28], [34], [40], [41]. CHOSLAR achieves a normalized energy efficiency of 1.40 Gbps/W, which is 1.40×, 20.00×, 3.68×, and 5.19× higher than those in [22], [23], [26], and [32], respectively. Moreover, CHOSLAR achieves a normalized area efficiency of 0.62 Mbps/kG, which is 5.17×, 31.00×, 5.64×, and 15.50× higher than those in [22], [23], [26], and [32], respectively. In addition, the architectures proposed in [22], [23], [26], [32] do not perform LR. The proposed architecture with LR achieves high energy and area efficiencies of 1.93 Gbps/W and 0.95 Mbps/kG, respectively. Under these conditions, the proposed architecture achieves significant improvements in terms of energy and area efficiency. In terms of latency, CHOSLAR achieves a latency of 0.7 μs with no LR, which is 55.56% of the latency reported in [26]. The latency reported in [22] is lower than that of CHOSLAR, but the energy and area efficiencies of CHOSLAR are significantly improved compared with those in [22]. The architecture presented in [34] achieves a higher energy efficiency than that of CHOSLAR, but the area efficiency is lower. Notably, the throughput reported in [34] was determined under the assumption that the channel conditions remain fixed for a resource block of 72 MIMO symbols and, hence, that preprocessing is performed only once for multiple MIMO detections, which means the throughput and energy efficiency should be scaled down significantly for a fair comparison. Nevertheless, the proposed architecture achieves a much higher throughput (by 6.03×) with 41.72% of the latency reported in [34]. Notably, a 16 × 16 MIMO configuration is considered for CHOSLAR, whereas 4 × 4 or 8 × 8 MIMO configurations are considered in [22], [26], [32], and [34]. For higher-order MIMO configurations, the latency values reported in [22], [26], [32], and [34] will be significantly increased.

The architectures proposed in [40] are suitable for nonlinear detection in a large-scale MIMO system. The throughput achieved with CHOSLAR is 11.84×, 9.43× and 9.72× higher than those of the architectures reported for different MIMO configurations in [40]. The low throughputs achieved in [40] may not satisfy the data rate requirements of future communication systems. The normalized energy efficiency achieved with CHOSLAR is 10.00×, 4.12×, and 2.59× higher than those reported for the different MIMO configurations in [40]. Moreover, the area efficiency achieved with CHOSLAR is improved by 7.75×, 5.17×, and 3.88× compared with those reported in [40]. Furthermore, the architectures presented in [40] support only binary phase shift keying (BPSK) or quadrature phase shift keying (QPSK) transmission and are not suitable for high-order modulation. Hence, this limitation to low-order modulation is an additional disadvantage for the use of these architectures in future communication systems.

Notably, the architectures proposed in [9]–[13] are designed for linear detection algorithms and achieve near-MMSE performance. Hence, these linear detectors suffer from non-negligible losses in detection accuracy, especially in MIMO systems with comparable (or equal) numbers of users and BS antennas (as shown in Fig. 5), which is why Table II does not include results for these linear detectors.

VI. Conclusions

This paper proposes a preprocessing algorithm for K-best detection that combines Cholesky sorted QR decomposition and partial iterative lattice reduction (CHOSLAR). In this algorithm, Cholesky sorted QR decomposition is used in place of complex QR decomposition, constant-throughput partial iterative lattice reduction with high parallelism is adopted to achieve near-optimal detection accuracy, and a sorting-reduced K-best strategy is used for vector estimation. A corresponding VLSI architecture with low latency and high energy and area efficiencies is designed. This preprocessor represents an important contribution to next-generation MIMO communication systems, such as 5G. Future work will focus on different close-to-ML options for soft output purposes.

References

[1] F. Rusek *et al.*, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[2] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tut.*, vol. 17, no. 4, pp. 1941–1988, Sep. 2015.

[3] W. Van Etten, "Maximum likelihood receiver for multiple channel transmission systems," *IEEE Trans. Commun.*, vol. COM-24, no. 2, pp. 276–283, Feb. 1976.

[4] M. Kisialiou, X. Luo, and Z. Q. Luo, "Efficient implementation of quasi-maximum-likelihood detection based on semidefinite relaxation," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4811–4822, Dec. 2009.

[5] Y. Jiang, M. K. Varanasi, and J. Li, "Performance analysis of ZF and MMSE equalizers for MIMO systems: An in-depth study of the high SNR regime," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2008–2026, Apr. 2011.

[6] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.

[7] L. Dai, X. Gao, X. Su, S. Han, C.-L. I, and Z. Wang, "Low-complexity soft-output signal detection based on Gauss-Seidel method for uplink multi-user large-scale MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4839–4845, Oct. 2015.

[8] X. Gao, L. Dai, Y. Hu, Z. Wang, and Z. Wang, "Matrix inversion-less signal detection using SOR method for uplink large-scale MIMO systems," in *Proc. IEEE Global Telecommun. Conf.*, Austin, TX, USA, Dec. 2014, pp. 3291–3295.

[9] M. Wu, B. Yin, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "Large-scale MIMO detection for 3GPP LTE: Algorithms and FPGA implementations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 916–929, Oct. 2014.

[10] G. Peng, L. Liu, S. Zhou, S. Yin, and S. Wei, "A 1.58 Gbps/W 0.40 Gbps/mm^2 ASIC implementation of MMSE detection for 128 $\times$ 8 64-QAM massive MIMO in 65 nm CMOS," *IEEE Trans. Circuits Syst. I, Reg. Papers*, to be published.

[11] G. Peng, L. Liu, P. Zhang, S. Yin, and S. Wei, "Low-computing-load, high-parallelism detection method based on chebyshev iteration for massive MIMO systems with VLSI architecture," *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3775–3788, Jul. 2017.

[12] H. Prabhu, J. N. Rodrigues, L. Liu, and O. Edfors, "A 60 pJ/b 300 Mb/s 128 $\times$ 8 massive MIMO precoder-detector in 28 nm FD-SOI," in *Proc. IEEE Int. Solid-State Circuits Conf. Dig.*, San Francisco, CA, USA, Feb. 2017, pp. 60–61.

[13] W. Tang, C.-H. Chen, and Z. Zhang, "A 0.58 mm^2 2.76 Gb/s 79.8 pJ/b 256-QAM massive MIMO message-passing detector," in *Proc. IEEE Symp. VLSI Circuits*, Honolulu, HI, USA, Jun. 2016, pp. 1–2.

[14] X. Chu and J. Mcallister, "Software-defined sphere decoding for FPGA-based MIMO detection," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 6017–6026, Nov. 2012.

[15] M. M. Mansour, S. P. Alex, and L. M. A. Jalloul, "Reduced Complexity soft-output MIMO sphere detectors—Part I: Algorithmic optimizations," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5505–5520, Nov. 2014.

[16] M. M. Mansour, S. P. Alex, and L. M. A. Jalloul, "Reduced complexity soft-output MIMO sphere detectors—Part II: Architectural optimizations," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5521–5535, Nov. 2014.

[17] 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, pp. 2531–2542, Oct. 2011.

[18] C. A. Shen, A. Eltawil, and K. N. Salama, "Evaluation framework for K-best sphere decoders," *J. Circuits, Syst. Comput.*, vol. 19, no. 5, pp. 975–995, Aug. 2010.

[19] S. Mondal, A. Eltawil, C. A. Shen, and K. N. Salama, "Design and implementation of a sort-free K-best sphere decoder," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 18, no. 10, pp. 1497–1501, Oct. 2010.

[20] C. A. Shen, A. M. Eltawil, K. N. Salama, and S. Mondal, "A best-first soft/hard decision tree searching MIMO decoder for a 4 $\times$ 4 64-QAM system," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 20, no. 8, pp. 1537–1541, Aug. 2012.

[21] T. Fujino, S. Wakazono, and Y. Sasaki, "A Gram-Schmidt based lattice-reduction aided MMSE detection in MIMO systems," in *Proc. IEEE Global Telecommun. Conf.*, Honolulu, HI, USA, Dec. 2009, pp. 1–8.

[22] P. L. Chiu, L. Z. Huang, L. W. Chai, and C. F. Liao, "A 684 Mbps 57 mW joint QR decomposition and MIMO processor for 4 $\times$ 4 MIMO-OFDM systems," in *Proc. IEEE Asian Solid State Circuits Conf.*, Jeju, South Korea, Nov. 2011, pp. 309–312.

[23] 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*, vol. 62, no. 3, pp. 733–742, Mar. 2015.

[24] I. H. Kurniawan, J. H. Yoon, and J. Park, "Multidimensional Householder based high-speed QR decomposition architecture for MIMO receivers," in *Proc. IEEE Int. Symp. Circuits Syst.*, Beijing, China, May 2013, pp. 2159–2162.

[25] J. M. Delosme and S. F. Hsiao, "Householder CORDIC algorithms," *IEEE Trans. Comput.*, vol. 44, no. 8, pp. 990–1001, Aug. 1995.

[26] Z. Yan, G. He, Y. Ren, W. He, J. Jiang, and Z. Mao, "Design and implementation of flexible dual-mode soft-output MIMO detector with channel preprocessing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 11, pp. 2706–2717, Nov. 2015.

[27] 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.*, Seoul, South Korea, May 2012, pp. 2857–2860.

[28] J. Y. Wang, R. H. Lai, C. M. Clien, and P. A. Ting, "A 2 $\times$ 2-8 $\times$ 8 sorted QR decomposition processor for MIMO detection," in *Proc. IEEE Asian Solid State Circuits Conf.*, Beijing, China, Nov. 2010, pp. 1–4.

[29] H. Sarieddeen, M. M. Mansour, L. Jalloul, and A. Chehab, "High order multi-user MIMO subspace detection," *J. Signal Process. Syst.*, vol. 87, no. 1, pp. 1–17, Apr. 2017.

[30] M. M. Mansour, "A near-ML MIMO subspace detection algorithm," *IEEE Signal Process. Lett.*, vol. 22, no. 4, pp. 408–412, Apr. 2015.

[31] H. Sarieddeen, M. M. Mansour, and A. Chehab, "Efficient near-optimal 8 $\times$ 8 MIMO detector," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Doha, Qatar, Apr. 2016, pp. 1–6.

[32] M. M. Mansour and L. M. A. Jalloul, "Optimized configurable architectures for scalable soft-input soft-output MIMO detectors with 256-QAM," *IEEE Trans. Signal Process.*, vol. 63, no. 18, pp. 4969–4984, Sep. 2015.

[33] H. Sarieddeen, M. M. Mansour, and A. Chehab, "Efficient subspace detection for high-order MIMO systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 1001–1005.

[34] C. F. Liao, J. Y. Wang, and Y. H. Huang, "A 3.1 Gb/s 8 $\times$ 8 sorting reduced K-best detector with lattice reduction and QR decomposition," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 22, no. 12, pp. 2675–2688, Dec. 2014.

[35] D. Wubben, R. Bohnke, V. KüHn, and K. D. Kammeyer, "MMSE-based lattice-reduction for near-ML detection of MIMO systems," in *Proc. ITG Workshop Smart Antennas*, Munich, Germany, Mar. 2008, pp. 106–113.

[36] Q. Li, J. Zhang, L. Bai, and J. Choi, "Lattice reduction-based approximate MAP detection with bit-wise combining and integer perturbed list generation," *IEEE Trans. Commun.*, vol. 61, no. 8, pp. 3259–3269, Aug. 2013.

[37] X. Wang, P. Gong, K. Niu, and W. Wu, "Decision feedback aided detection based on lattice reduction in MIMO systems," in *Proc. IEEE Veh. Technol. Conf.*, Melbourne, VIC, Australia, May 2006, pp. 1708–1712.
[38] S. JäRmyr, B. Ottersten, and E. A. Jorswieck, "Statistical framework for optimization in the multi-user MIMO uplink with ZF-DFE," *IEEE Trans. Signal Process.*, vol. 62, no. 10, pp. 2730–2745, Oct. 2014.
[39] T. Liu, J. K. Zhang, and K. M. Wong, "Optimal precoder design for correlated MIMO communication systems using zero-forcing decision feedback equalization," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3600–3612, Sep. 2009.
[40] O. Castaneda, T. Goldstein, and C. Studer, "Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2334–2346, Dec. 2016.
[41] C. Zhang, H. Prabhu, Y. Liu, L. Liu, O. Edfors, and V. Öwall, "Energy efficient group-sort QRD processor with on-line update for MIMO channel pre-processing," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 5, pp. 1220–1229, May 2015.
[42] S. Yang and L. Hanzo, "Exact Bayes' theorem based probabilistic data association for iterative MIMO detection and decoding," in *Proc. IEEE Global Telecommun. Conf.*, Atlanta, GA, USA, Dec. 2013, pp. 1891–1896.
[43] T. H. Kim, "Low-complexity sorted QR decomposition for mimo systems based on pairwise column symmetrization," *IEEE Trans. Wireless Commun.*, vol. 13, no. 3, pp. 1388–1396, Mar. 2014.
[44] Y. Chen, H. Halbauer, M. Jeschke, and R. Richter, "An efficient cholesky decomposition based multiuser MIMO detection algorithm," in *Proc. IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, Instanbul, Turkey, Sep. 2010, pp. 499–503.
[45] Y. Xue, C. Zhang, S. Zhang, and X. You, "A fast-convergent pre-conditioned conjugate gradient detection for massive MIMO uplink," in *Proc. IEEE Int. Conf. Dig. Signal Process.*, Beijing, China, Oct. 2016, pp. 331–335.
[46] Y. Xue, C. Zhang, S. Zhang, Z. Wu, and X. You, "Steepest descent method based soft-output detection for massive mimo uplink," in *Proc. IEEE Int. Workshop Signal Process. Syst.*, Dallas, TX, USA, Oct. 2016, pp. 273–278.
[47] W. Zhang, S. Qiao, and Y. Wei, "HKZ and Minkowski reduction algorithms for Lattice-reduction-aided MIMO detection," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5963–5976, Nov. 2012.
[48] W. Jiang, Y. Asai, and S. Kubota, "A novel detection scheme for MIMO spatial multiplexing systems with partial lattice reduction," in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, Tokyo, Japan, Apr. 2009, pp. 2524–2528.
[49] B. E. Godana and T. Ekman, "Parametrization based limited feedback design for correlated MIMO channels using new statistical models," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 5172–5184, Oct. 2013.

Guiqiang Peng received the B.S. degree from the School of Micro-Electronics and Solid-State Electronics, University of Electronic Science and Technology of China, Chengdu, China, in 2013. He is currently working toward the Ph.D. degree at the Institute of Microelectronics, Tsinghua University, Beijing, China. His current research interests include reconfigurable computing, mobile computing, and VLSI signal processing and wireless communications.

Leibo Liu (M'10–SM'16) received the B.S. degree in electronic engineering and the Ph.D. degree, both from the Institute of Microelectronics, Tsinghua University, Beijing, China, in 1999 and 2004, respectively. He is currently an Associate Professor with the Institute of Microelectronics, Tsinghua University. His current research interests include reconfigurable computing, mobile computing, and VLSI digital signal processing.

Sheng Zhou (S'06–M'12) received the B.E. and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2005 and 2011, respectively. From January to June 2010, he was a visiting student with the Wireless Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA, USA. He is currently an Associate Professor with the Department of Electronic Engineering, Tsinghua University. His research interests include cross-layer design for multiple-antenna systems, edge computing and caching, and green wireless communications.

Yang Xue received the B.S. degree from the School of Micro-Electronics and Solid-State Electronics, University of Electronic Science and Technology of China, Chengdu, China, in 2014. He is currently working toward the Master's degree at the Institute of Microelectronics, Tsinghua University, Beijing, China. His current research interests include reconfigurable computing, mobile computing, and VLSI signal processing and wireless communications.

Shouyi Yin (M'09) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 2000, 2002, and 2005, respectively. He was with the Imperial College London, as a Research Associate. He is currently with the Institute of Microelectronics, Tsinghua University, as an Associate Professor. His research interests include mobile computing, wireless communications, and SoC design.

Shaojun Wei (M'91) was born in Beijing, China, in 1958. He received the Ph.D. degree from La Faculté Polytechnique de Mons, Mons, Belgium, in 1991. He became a Professor at the Institute of Microelectronics, Tsinghua University, Beijing, China, in 1995. He is a senior member of the Chinese Institute of Electronics. His main research interests include VLSI SoC design, EDA methodology, and ASIC design for communications.