IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 June 2024, accepted 31 July 2024, date of publication 14 August 2024, date of current version 26 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3443536

RESEARCH ARTICLE

High-Throughput Accelerator for Exact-MMSE Soft-Output Detection in Open RAN Systems

THOMAS JAMES THOMAS, (Member, IEEE),
AND KONSTANTINOS NIKITOPOULOS, (Senior Member, IEEE)
5G and 6G Innovation Center, Institute for Communication Systems, University of Surrey, GU2 7XH Guildford, U.K.

Corresponding author: Konstantinos Nikitopoulos (k.nikitopoulos@surrey.ac.uk)

This work was supported in part by the Highly Intelligent, Highly Performing RAN (HiPer-RAN) Project, a winner of the U.K.'s Department for Science, Innovation and Technology (DSIT) Open Networks Ecosystem Competition; and in part by the Non-linear Processing for High Performing Communications (NL-COMM) Project, a winner of Innovate U.K.'s and U.K.'s DSIT Future Telecommunications Challenge Competition.

ABSTRACT Open Radio Access Networks (Open RANs), realized fully in software, require excessive computing resources to support time-sensitive signal-processing algorithms in the physical layer. Among them, multiple-input-multiple-output (MIMO) processing is a key functionality used to drive higher connectivity in the uplink, but it is computationally intensive, triggering the need for hardware acceleration to overcome the processing inefficiency of software-based solutions. Additionally, energy efficiency is becoming a key focus in Open RAN to enable sustainable deployments that utilize available resources efficiently. Because channel-inversion complexity increases polynomially with the number of users in linear detectors, such as zero-forcing (ZF) and minimum-mean-square-error (MMSE), acceleration based on channel-inverse approximations has gained significant attention. However, they unnecessarily multiply the number of base station (BS) antennas to ensure accurate detection, leading to a drastic increase in power consumption owing to the additional radio frequency (RF) chains employed. In contrast, linear detectors achieve a sufficiently good performance with only twice the number of BS antennas as users. This work introduces an exact-MMSE and soft-output hardware accelerator that includes an inversion-free, highly-parallel QR decomposition (QRD) architecture and a low-complexity detector stage with per-cycle soft-output generation, significantly improving the processing latency and throughput. The proposed architecture is fully scalable to support diverse MIMO configurations. Implementation evaluations on a Xilinx Virtex Ultrascale+ field-programmable gate array (FPGA) demonstrate that the proposed exact solution can achieve more than 2× improvement in hardware throughput over existing approximate designs. Moreover, the peak throughput can be increased around 10-fold in slowly fading channels.

INDEX TERMS Open RAN, MIMO, matrix inversion, QR decomposition, FPGA, hardware acceleration.

I. INTRODUCTION

A. BACKGROUND

The Open Radio Access Network (Open RAN) [1] has kick-started a radical transformation in the landscape of wireless network infrastructure by enabling a growing ecosystem of scalable, cost-efficient and standards-based components [2]. The underlying principle of open interfaces ensures interoperability and lends flexibility to mobile network operators in choosing the best-of-breed components from a wide pool of solution providers to optimize performance. This has also triggered research into leveraging reconfigurable platforms, such as general-purpose processors (GPPs), field programmable gate arrays (FPGAs), graphics processing units (GPUs) [3], or an amalgamation of these, to realize the computationally challenging functions of the physical layer (PHY), as opposed to monolithic radio access networks (RANs). These implementation options have varying degrees of performance with respect to throughput, latency, cost, flexibility and power consumption, leading to a requirement-based implementation approach for each use case. Nonetheless, the existing Open RAN solutions still fail

The associate editor coordinating the review of this manuscript and approving it for publication was Ronald Chang.

to achieve the absolute performance of traditional baseband units.

Multiple-input multiple-output (MIMO) [4], which is particularly instrumental in improving the throughput and connectivity gains of traditional RANs by spatially multiplexing several users on the same spectrum resources, is computationally intensive. This presents a significant challenge to purely software-based Open RAN deployments owing to their inherent inefficiency in performing advanced MIMO computations for a higher number of users and bandwidths. For example, standard-compliant solutions such as OpenAirInterface and srsRAN can support only up to two users, while efforts such as [5] and [6] utilize several system-level relaxations to support a higher number of users, such as having the same channel realization for 16 and 32 subcarriers, respectively. In addition to the resultant detrimental effects on performance, these approaches noticeably sway from fifth generation new radio (5GNR) standards and support only a fraction of the bandwidth. Recently, a 5GNR and Open RAN compliant GPP-based implementation was proposed in [7], which facilitates power-efficient MIMO systems with a large number of information streams but still leaves substantial room for fully exploiting the channel capacity. Alternatively, cloud-based servers can also operationalize software-based implementations of the open distributed unit (O-DU), but the considerable costs related to providing adequate front-haul bandwidth for such deployments leads to impractical scenarios [8]. This primarily results in O-DU deployments on site, in close proximity to the open radio unit (O-RU), which restricts the number of servers that can be used to improve the processing capability owing to physical size constraints.

From an algorithmic perspective, most existing Open RAN solutions adopt linear MIMO processing approaches such as zero-forcing (ZF) and minimum mean square error (MMSE) [9], which simplify the detection problem by translating the corresponding MIMO channel into several non-interfering single-antenna channels. They also enable easier soft output calculations in the form of log-likelihood ratios (LLRs) and a simplified mechanism to adapt the modulation and coding scheme (MCS) employed by each user according to the channel conditions. This efficient radio resource management (RRM) enabled by linear detectors simplifies their integration in standards-based systems such as 5GNR. Although non-linear techniques exist that can better exploit the spatial characteristics of MIMO channels by transforming the maximum likelihood (ML) detection problem into a tree search [10], they are exceedingly computationally demanding. Fixed-complexity decoders [11] and K-best detectors [12] reduce hardware complexity by curtailing the search space and attaining near-optimal performance for small-scale MIMO systems. However, their complexity and processing latency (PL) are significantly higher than those of linear approaches for an increasing number of users and modulation orders. Additionally, adapting the MCS of each user is not straightforward in non-linear receivers, which makes their integration into 5GNR more challenging [13]. Therefore, linear approaches are generally preferred over non-linear methods because of their easier integration and efficient radio resource management. Additionally, they can attain near-optimal performance when the number of receive antennas at the base station (BS), N_R, exceeds twice that of the user equipment (UE), N_T [14], [15]. However, linear methods require the direct inversion of channel matrices, which scale polynomially in complexity with the number of users, $\mathcal{O}(N_T^3)$. Conventional matrix factorization approaches such as QR decomposition [16], Gauss-Jordan elimination [17], and Cholesky [18] can reduce the complexity of inversion, but these are not trivial. Most detector implementations based on field-programmable gate arrays (FPGAs) or application-specific integrated circuits (ASICs) have focused on linear approaches for small-scale MIMO systems that support up to four users [18], [19], [20], [21]. There has been limited interest in pursuing exact-linear detectors for a large number of users due to the tremendous resources needed for implementation.

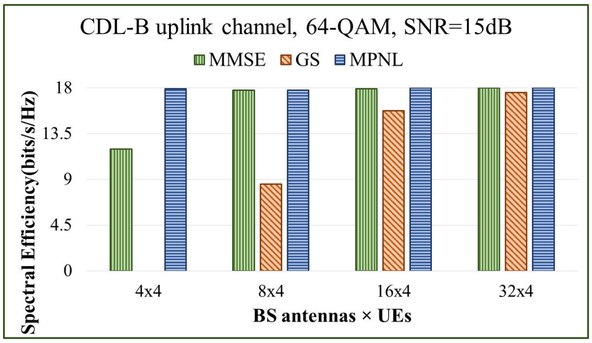


FIGURE 1. Detection performance comparison in a CDL-B uplink channel with 64-quadrature amplitude modulation (64-QAM), signal to noise ratio (SNR) per receive antenna 15dB, UE speed 5km/h and low-density parity check (LDPC) code rate 3/4.

Subsequently, iterative algorithms [22], [23], [24] have emerged, particularly in the context of massive MIMO, to circumvent costly inversions by converging to the MMSE solution in a series of adaptations, with a complexity of $\mathcal{O}(N_T^2)$. Due to the much higher number of receive antennas than users considered in these methods (i.e. $N_R \gg N_T$), they can generally obtain near-MMSE solutions in few iterations. As a result, there have been significant efforts to develop efficient hardware architectures based on iterative techniques such as the Neumann series expansion (NSE) [25], Gauss-Siedel (GS) [26], Jacobi, and Steepest Descent (SD) [27], [28]. However, in contrast to linear approaches that guarantee near-optimal detection in MIMO systems with $N_R = 2N_T$, approximation strategies have slow convergence and low reliability. Even when N_R is moderately higher than N_T, more iterations are required to attain convergence, which can affect the overall throughput. Fig. 1 shows a comparison of the spectral efficiency achieved by linear MMSE, approximation-based GS approach, and the

TABLE 1. High-level comparison of MIMO detection approaches.

Aspect	Linear	Approximate	Non-linear
Performance	Good	Worse than linear	Near-optimal
Computational complexity	Moderate	Lower than Linear	High
Ease of integration	High	High	Low
Power Efficiency	Moderate	Low	High

massively parallel non-linear (MPNL) [29] in a scenario with 4 user equipment (UEs) and an increasing number of BS antennas. It can be seen that MPNL achieves the maximum capacity when $N_R = N_T$ whereas GS fails completely. When N_R is twice N_T, MMSE matches the performance of MPNL, whereas GS attains a spectral efficiency of only 50%. GS can attain near maximum capacity only when N_R is increased to eight times N_T. Increasing the BS antenna to user ratio (i.e., N_R/N_T) in this manner to enable the complexity gains of these approximate-inversion detectors can adversely impact the energy efficiency, owing to the considerable radio frequency (RF) power required. This is further emphasized by the fact that energy-efficiency is a priority in Open RAN systems to achieve sustainable solutions [30]. Table 1 provides a high-level summary of linear, approximation-based, and non-linear approaches. Based on these considerations, linear solutions can realize highly practical system deployments and are better suited for MIMO processing owing to their easier integration and relative simplicity compared to non-linear approaches, and their power efficiency gains over approximate solutions.

Therefore, enabling O-DU to support linear MIMO processing for higher connectivity is of great practical interest. Although specialized software acceleration solutions targeting 5GNR systems exist [31], they are not adequate for increasing connectivity and bandwidth considerations, because of the significant processing power needed to reduce the PL. To address this challenge, hardware acceleration has emerged as a key solution to offload compute-centric operations of the O-DU to FPGAs, ASICs or graphics processing units (GPUs) [32]. However, the major focus has so far remained on small-scale MIMO systems [3] or accelerating only fixed functions, such as channel decoding [33]. Since matrix inversion complexity scales polynomially with the number of users, this easily overtakes the complexity of low density parity code (LDPC) decoding. Hence, significant attention should be given to developing hardware accelerators that can perform linear detection in MIMO systems with a large number of users by striking a good balance between hardware complexity and energy efficiency, which will be the focus of this work.

B. MAIN CONTRIBUTIONS

The main contributions of this paper are highlighted below

1) A new highly parallel inversion-free QR decomposition (QRD) architecture was proposed based on the MMSE criterion, which can directly yield the inverse of the triangular ($\mathbf{R}^{-1}$) and the near-orthogonal component of $\mathbf{Q}$.
2) A reduced-complexity architecture was designed to avoid matrix-matrix multiplication and compute the equalized vector in tandem with the post-detection signal-to-interference-noise ratio (SINR).
3) A novel highly efficient architecture is proposed that performs log likelihood ratio (LLR) computation in two stages, which enables significant resource sharing and full unfolding of distance computations for each user bit.

C. ORGANIZATION OF THE WORK

The remainder of this paper is organized as follows. The preliminaries including a brief description of the MIMO system model and aspects related to the soft-output MMSE detection are discussed in Section II. Subsequently, in Section III, the inversion-free parallel QR algorithm is presented in the soft-output MMSE equalization framework, along with computational complexity analyses and performance evaluations with respect to state-of-the-art linear and approximation-based methods. The results demonstrate the diminished computational gains of approximation-based methods in contrast to exact linear approaches when reliability is jointly considered and justifies the need to accelerate linear MIMO processing. Section IV discusses the hardware architecture of the proposed accelerator in detail, with an emphasis on careful optimization of the critical processing blocks. The implementation results of the proposed architecture on a Xilinx XCVU9P device are presented in Section V with detailed discussions on scalability, processing latency, throughput, power consumption, and software integration, along with some insights on future directions. The conclusion is presented in Section VI.

II. PRELIMINARIES

A. NOTATION

Boldface uppercase letters represent matrices and boldface lowercase letters denote column vectors. Given matrix $\mathbf{A}$, the Hermitian transpose of $\mathbf{A}$ is represented by $\mathbf{A}^H$. Given column vector $\mathbf{a}$, a_i denotes the i^{th} element of $\mathbf{a}$ and $\|\mathbf{a}\|_2$ denotes the l_2-norm of $\mathbf{a}$ which can be defined as $\|\mathbf{a}\|_2 = \sqrt{\sum_i |a_i|^2}$. $|\mathcal{A}|$ denotes the cardinality of set $\mathcal{A}$.

B. MIMO SYSTEM MODEL

Consider a multi-user MIMO-OFDM based uplink system consisting of N_T users served by an N_R-antenna base station. The user terminals individually encode their data streams and map the encoded bits onto a finite set of constellation symbols described by $\mathcal{O}$, corresponding to $\mathcal{B} = \log_2 |\mathcal{O}|$ bits of information per symbol. Assuming perfect channel state information (CSI) is available, the MIMO input-output relation can be mathematically represented as

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

where $\mathbf{y}$ is the N_R-dimensional receive vector belonging to the complex space (i.e., $\mathbf{y} \in \mathcal{C}^{N_R}$), $\mathbf{H}$ models the complex $N_R \times N_T$ channel matrix, and $\mathbf{s}$ represents the transmit symbol vector containing entries belonging to $\mathcal{O}$. $\mathbf{n}$ denotes the N_R-dimensional i.i.d. complex Gaussian noise with a variance $2\sigma^2$ per element.

C. MMSE DETECTION

The optimal detection strategy that minimizes the hard-symbol error rate is the ML formulation, as shown below.

$$\tilde{\mathbf{s}}^{\mathrm{ML}} = \arg \min_{\mathbf{s}\in\mathcal{O}^{N_T}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2 \tag{2}$$

However, the prohibitive complexity of solving Equation (2) does not make it employable in practice. Linear equalization methods determine approximate solutions to the ML problem by relaxing the finite constellation constraint $\mathbf{s} \in \mathcal{O}^{N_T}$ in Equation (2) to the N_T-dimensional complex space $\mathbf{s} \in \mathcal{C}^{N_T}$. This enables the low-cost computation of an estimate $\tilde{\mathbf{s}}$ that is potentially close to the optimal solution, which can then be sliced onto the closest constellation symbol in $\mathcal{O}$ to obtain the hard-output. Alternatively, $\tilde{\mathbf{s}}$ can be used to calculate the reliability information in terms of LLR values, which yields the soft outputs used by the channel decoder.

The zero-forcing formulation can be expressed simply as

$$\tilde{\mathbf{s}}^{\mathrm{ZF}} = \arg \min_{\mathbf{s}\in\mathcal{C}^{N_T}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2 \tag{3}$$

which is quadratic in $\mathbf{s}$ and has a closed-form solution obtained by multiplying $\mathbf{y}$ with the pseudo-inverse of $\mathbf{H}$. MMSE equalization is then formulated by including a penalty term that considers the noise amplification of the ZF solution, as shown below.

$$\tilde{\mathbf{s}}^{\mathrm{MMSE}} = \arg \min_{\mathbf{s}\in\mathcal{C}^{N_T}} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2 + 2\sigma^2\|\mathbf{s}\|_2^2 \tag{4}$$

The closed-form solution of the MMSE minimization problem is given by

$$\tilde{\mathbf{s}}^{\mathrm{MMSE}} = (\mathbf{H}^H\mathbf{H} + \sigma^2\mathcal{I}_{N_T})^{-1}\mathbf{H}^H\mathbf{y} \tag{5}$$

Thus, the ZF and MMSE equalization correspond to a transformation $\mathbf{T}$ of the receive vector $\mathbf{y}$ to yield the estimated vector $\tilde{\mathbf{s}}$, that is, $\tilde{\mathbf{s}} = \mathbf{T}\mathbf{y}$. Assuming $\mathbf{A} = \mathbf{H}^H\mathbf{H} + \sigma^2\mathcal{I}_{N_T}$ is the MMSE filter matrix and $\tilde{\mathbf{y}} = \mathbf{H}^H\mathbf{y}$ is the matched filter output, Equation (5) can be simplified to $\tilde{\mathbf{s}}^{\mathrm{MMSE}} = \mathbf{A}^{-1}\tilde{\mathbf{y}}$, whereas the linear transformation matrix $\mathbf{T}_{\mathrm{MMSE}}$ is given by

$$\mathbf{T}_{\mathrm{MMSE}} = (\mathbf{H}^H\mathbf{H} + \sigma^2\mathcal{I}_{N_T})^{-1}\mathbf{H}^H \tag{6}$$

Although this solution is efficient for conventional small-order MIMO systems, the complexity of computing $\mathbf{H}^H\mathbf{H}$ and the inverse of $\mathbf{A}$ rapidly scales up with the increasing order of the MIMO system.

D. SOFT-OUTPUT DETECTION

Hard-output detection can be achieved by directly slicing the MMSE estimates onto the nearest point in the constellation map. However, systems employing forward error correction (FEC) can exploit soft information to improve detection accuracy. This necessitates computing the LLR values which is typically performed using max-log approximation [18].

$$\mathcal{L}_{i,b} = \rho_i \left(\min_{a\in\mathcal{O}_b^0} \left| \frac{(\tilde{\mathbf{s}})_i}{\mu_i} - a \right|^2 - \min_{a\in\mathcal{O}_b^1} \left| \frac{(\tilde{\mathbf{s}})_i}{\mu_i} - a \right|^2 \right) \tag{7}$$

where $\mathcal{O}_b^0$ and $\mathcal{O}_b^1$ correspond to the sets of constellation points for which the b^{th} bit is 0 and 1 respectively, and ρ_i corresponds to the post-equalization signal-to-interference and noise-ratio (SINR). The channel gain μ_i can be computed as $\mu_i = [\mathbf{A}^{-1}]_i^H[\mathbf{G}]_i$, where $[\mathbf{A}^{-1}]_i$ and $[\mathbf{G}]_i$ represent the i^{th} row and column of $\mathbf{A}^{-1}$ and the gram matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ respectively [28].

III. INVERSION-FREE SOFT-MMSE EQUALIZATION

The exact solution of MMSE (4) demands huge computational resources particularly as the number of antennas and users increases. As discussed earlier, approximate solutions using GS [34] and conjugate gradient (CG) can work at moderate complexity, but the excessively high N_R required to ensure convergence leads to energy-efficiency concerns in practical scenarios. Exact-MMSE detection can work sufficiently with a relatively lower N_R but requires algorithmic optimizations that facilitate lower hardware complexity. In this regard, a hardware-friendly version of an inversion-free parallel QRD-based soft-output exact-MMSE detector is proposed.

A. INVERSION-FREE MMSE REFORMULATION

Traditional QRD based on modified Gram Schmidt (MGS) was used in [16] to compute the Q and R matrices. However, this requires additional circuitry to invert R that also includes resource- and latency-intensive division operations. The QRD-based detection extended to the MMSE criterion in [35] yields an interesting observation. Consider an $(N_R+N_T)\times N_T$ dimensional matrix $\overline{\mathbf{H}}$ which is constructed by appending a scaled identity matrix to the bottom of $\mathbf{H}$ as follows

$$\overline{\mathbf{H}} = \begin{bmatrix} \mathbf{H} \\ \sigma\mathbf{I}_{N_T} \end{bmatrix} \tag{8}$$

This augmented channel matrix $\overline{\mathbf{H}}$ can be decomposed as

$$\overline{\mathbf{H}} = \overline{\mathbf{Q}}\overline{\mathbf{R}} = \begin{bmatrix} \mathbf{Q}^1 \\ \mathbf{Q}^2 \end{bmatrix} \overline{\mathbf{R}} = \begin{bmatrix} \mathbf{Q}^1\overline{\mathbf{R}} \\ \mathbf{Q}^2\overline{\mathbf{R}} \end{bmatrix} \tag{9}$$

where $\mathbf{Q}^1$ and $\mathbf{Q}^2$ are obtained by partitioning the $(N_r+N_t)\times N_t$ matrix $\overline{\mathbf{Q}}$. Therefore, $\mathbf{A}$ can now be rewritten as

$$\begin{aligned} \mathbf{A} &= (\mathbf{H}^H\mathbf{H} + \sigma^2\mathcal{I}_{N_T}) \\ &= \overline{\mathbf{H}}^H\overline{\mathbf{H}} \end{aligned} \tag{10}$$

Consequently, employing the QRD of the augmented channel matrix simplifies the computationally complex direct matrix inversion to a simple linear MMSE weight matrix $\mathbf{T}$ shown below.

$$\begin{aligned}\mathbf{T} &= (\overline{\mathbf{H}}^H\overline{\mathbf{H}})^{-1}\mathbf{H}^H \\ &= (\overline{\mathbf{R}}^H\overline{\mathbf{Q}}^H\overline{\mathbf{QR}})^{-1}\overline{\mathbf{R}}^H\mathbf{Q}^{1H} \\ &= (\overline{\mathbf{R}})^{-1}(\overline{\mathbf{R}}^H)^{-1}\overline{\mathbf{R}}^H\mathbf{Q}^1 \\ &= \overline{\mathbf{R}}^{-1}\mathbf{Q}^{1H} \qquad (11)\end{aligned}$$

In particular, computing $\mathbf{T}$ involves an upper triangular matrix inversion, which only entails the complexity of $\mathcal{O}(N_T^2)$ compared with that of a full matrix inversion, i.e, $\mathcal{O}(N_T^3)$. Furthermore, it can be directly inferred from (9) that $\sigma\mathbf{I}_{N_t} = \mathbf{Q}^2\overline{\mathbf{R}}$ as $\overline{\mathbf{H}} = \overline{\mathbf{QR}}$ must be satisfied. This implies that the inverse of the upper triangular matrix $\overline{\mathbf{R}}$ is a by-product of the augmented channel QR decomposition with a simple scaling factor, as shown below.

$$\overline{\mathbf{R}}^{-1} = \frac{1}{\sigma}\mathbf{Q}^2 \qquad (12)$$

This avoids the explicit inversion of $\overline{\mathbf{R}}$ and leads to further simplification in computing $\mathbf{T}$ and the MMSE equalized vector computation as follows

$$\mathbf{T} = \frac{1}{\sigma}\mathbf{Q}^2\mathbf{Q}^{1H} \qquad (13)$$

$$\tilde{\mathbf{s}}^{\text{MMSE}} = \mathbf{T}\mathbf{y} \qquad (14)$$

Furthermore, it can also be observed that as $\sigma \rightarrow 0$, the corresponding MMSE estimate approaches the solution of the ZF problem in (3). This implies that the parameter σ determines the nature of the detection. Though employing the QRD of the extended channel matrix $\overline{\mathbf{H}}$ instead of $\mathbf{H}$ leads to additional computations in the ZF case, it provides flexibility in choosing either of the two methods by simply varying the σ value, without requiring additional circuitry to invert $\overline{\mathbf{R}}$. However, setting $\sigma = 0$ leads to $\mathbf{Q}^2 = 0$ which causes the approach to fail. Instead, setting σ to an infinitesimal value such that $\frac{1}{\sigma}\mathbf{Q}^2$ can be implemented with simple shift operations, leads to a non-zero $\mathbf{Q}^2$, which ensures that the ZF solution can be computed.

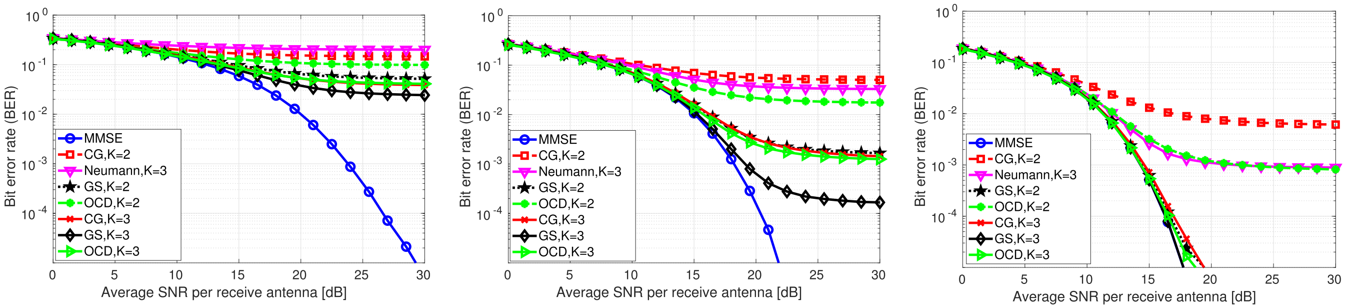
Even with these hardware-friendly optimizations, an efficient QRD of the extended channel matrix $\overline{\mathbf{H}}$ is important to overcome the fairly complex structure, strict data dependencies and high resource consumption of exact MMSE detectors for higher MIMO orders. Traditional QRD based on the modified Gram Schmidt (MGS) algorithm is widely popular owing to its column-based approach which invokes a good degree of parallelism. Both systolic array and iterative architectures have been pursued for QRD implementation [36]. However, the MGS algorithm entails a significant data dependency between the diagonal and triangular processes that are serially run for the duration of the algorithm. This data dependency can induce significant latency if a straightforward implementation of MGS is sought. The diagonal process (DP) in each iteration is responsible for three major tasks: i.) computing the squared norm of the currently selected column of $\overline{\mathbf{H}}$ in iteration i, ii.) determining the square-root of the squared l_2 norm term to yield $r_{i,i}$, and iii.) producing the orthonormal columns $\overline{\mathbf{q}}_i$ by dividing $\overline{\mathbf{H}}$ by $r_{i,i}$. The triangular process in turn, has to carry out two major tasks: a.) compute the projection of DP's orthonormal column $\overline{\mathbf{q}}_i$ on the remaining columns of $\overline{\mathbf{H}}$, and b.) deduct the corresponding component of $\overline{\mathbf{q}}_i$, that is, $r_{i,j}\overline{\mathbf{q}}_i$ from the respective column $\overline{\mathbf{h}}_i$. Lookahead techniques have been employed to speed up MGS processing for small-scale real-valued matrices [37], [38], but these still require significant additional processing to invert R, particularly for an increasing number of users.

To tackle the PL challenges of these data dependent computations from a hardware perspective, a highly parallel QRD algorithm is proposed, which effectively parallelizes the diagonal and triangular processes throughout the run of the algorithm and thus leads to significant latency savings over the traditional approach. This algorithm is discussed in detail below with the pseudo code provided in Algorithm 1.

Algorithm 1 Parallel QR Decomposition Algorithm

```
 1: procedure Parallel-QR( $\mathbf{H}, \sigma$ )
 2:     Initialization: $\overline{\mathbf{H}} = [\mathbf{H}^H \;\; \sigma\mathbf{I}_{N_T}]^H$
 3:     for $i = 1, \cdots, N_T$ do
 4:         Compute the dot product of $\overline{\mathbf{h}}_i$ with itself, i.e., $\overline{\mathbf{h}}_i^H\overline{\mathbf{h}}_i$
 5:         Compute the reciprocal square root of $\overline{\mathbf{h}}_i^H\overline{\mathbf{h}}_i$ given by rsr
 6:         for $j = i+1, \cdots, N_T$ do
 7:             Compute the inner product of $\overline{\mathbf{h}}_i$ with $\overline{\mathbf{h}}_j$ ($r_{i,j}^{\text{parallel}}$)
 8:         end for
 9:         Compute current $\mathbf{q}_i$ by multiplying $\overline{\mathbf{h}}_i$ with rsr
10:         for $j = i+1, \cdots, N_T$ do
11:             Compute $r_{i,j}$ by multiplying $r_{i,j}^{\text{parallel}}$ with rsr
12:             Deduct the contribution of $r_{i,j}\mathbf{q}_i$ from $\overline{\mathbf{h}}_j$
13:         end for
14:     end for
15:     Outputs: $\mathbf{Q} = \overline{\mathbf{H}}|_{1:N_R,:}$, $\mathbf{R}^{-1} = \frac{1}{\sigma}\overline{\mathbf{H}}|_{N_R+1:N_R+N_T,:}$
16: end procedure
```

Beginning with the first iteration, the diagonal process initiates the complex dot product of $\overline{\mathbf{h}}_1$ with itself and computes the inverse square root quantity rsr, shown on Step 4 of Algorithm 1. Simultaneously, an inner loop begins parallel computation of $r_{1,j}^{\text{parallel}}$ by the complex dot product of $\overline{\mathbf{h}}_1$ with $\overline{\mathbf{h}}_j$ where j varies from 2 to N_T, shown on steps 5-7 of Algorithm 1. Once rsr is computed, it is multiplied with $\overline{\mathbf{h}}_1$ to obtain $\overline{\mathbf{q}}_1$. Simultaneously, rsr is also shared with the parallel triangular processes to produce the respective coefficients $r_{i,j}$ that are needed subsequently to eliminate the contribution of $\overline{\mathbf{q}}_1$ from each of the remaining columns, as shown in steps 9-12 of Algorithm 1. This parallelization of diagonal and triangular processes continues until algorithm termination. The final $\overline{\mathbf{Q}}$ can simply be partitioned to obtain $\mathbf{Q}^1$ and $\mathbf{Q}^2$ matrices that will be used to determine the MMSE equalized vector according to equations (13) and (14). Compared to traditional QR decomposition strategies that require inversion of $\mathbf{R}$ to yield the MMSE estimate, this

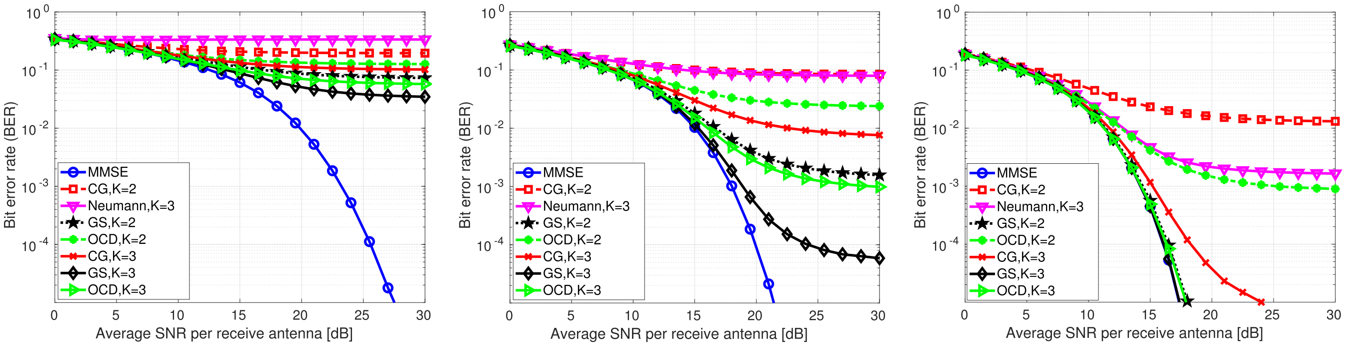


(a) 8 BS antennas

(b) 16 BS antennas

(c) 32 BS antennas

FIGURE 2. Bit error rate versus average SNR per receive antenna for 4 User Equipment (UE) and increasing number of BS antennas.



(a) 16 BS antennas

(b) 32 BS antennas

(c) 64 BS antennas

FIGURE 3. Bit error rate versus average SNR per receive antenna for 8 User Equipment (UE) and increasing number of BS antennas.

approach ensures hardware-friendliness by circumventing explicit inversion circuitry. While the advantages of this parallel QRD method can be understood from an algorithmic perspective, a careful scheduling of the individual operations and suitable trade-off between area and latency also need to be exploited, as will be laid out in the architecture discussion in Section IV.

Since MMSE detection inherently involves a trade-off between noise amplification and interference, the computation of post-detection signal-to-interference and noise ratio (SINR) for each layer is essential to generate correct soft outputs. To this extent, the error covariance matrix needs to be computed which generally involves the inversion of $\mathbf{A}$, this is simplified by incorporating some of the earlier identities, as shown below.

$$\begin{aligned}\mathbf{\Phi}_{\text{MMSE}} &= \mathbb{E}\{(\tilde{\mathbf{s}}^{\text{MMSE}} - \mathbf{s})(\tilde{\mathbf{s}}^{\text{MMSE}} - \mathbf{s})\} \\ &= \sigma^2(\mathbf{H}^H\mathbf{H} + \sigma^2\mathcal{I}_{N_T})^{-1} \\ &= \sigma^2(\overline{\mathbf{H}}^H\overline{\mathbf{H}})^{-1} \qquad \text{(from (10))} \\ &= \sigma^2\overline{\mathbf{R}}^{-1}\overline{\mathbf{R}}^{-H} = \mathbf{Q}^2\mathbf{Q}^{2H} \end{aligned} \tag{15}$$

Since the diagonal entries of the error covariance matrix yield the post-detection SINR quantities per stream, the upper triangular structure of $\mathbf{Q}^2$ leads to much fewer computations.

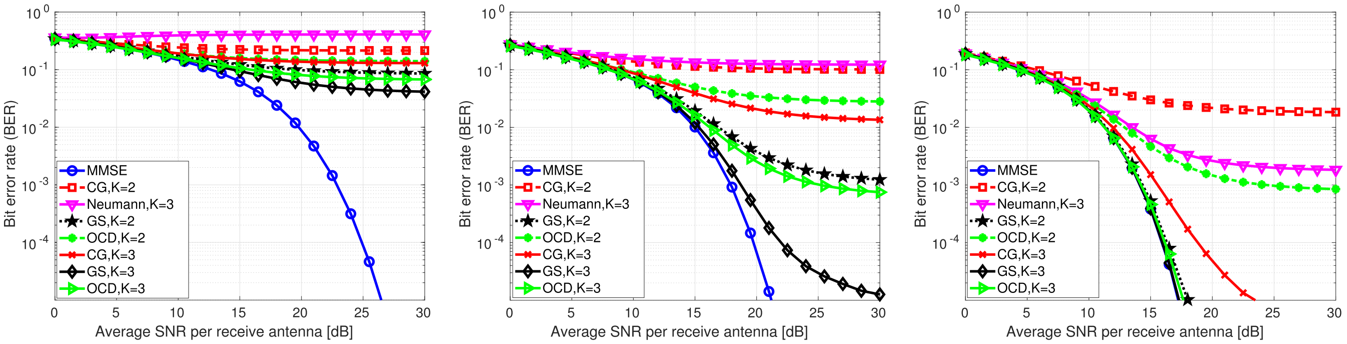
$$\eta = \text{diag}(\mathbf{Q}^2\mathbf{Q}^{2H}) \tag{16}$$

This simplification leads to a significant reduction in the hardware complexity to find the per-stream SINRs and can be performed as soon as the parallel QRD is completed.

To efficiently extract soft-information from the equalized output, the following simplified expression of the LLR computation in Equation (7) is given from [18].

$$LLR(b^i_{k,j}|\mathbf{s}^i, \mathbf{H}) \approx \frac{1}{\eta_i}\left(\min_{\mathbf{s}\in\mathcal{S}^0}\|\tilde{\mathbf{s}} - \mathbf{s}^i\|^2 - \min_{\mathbf{s}\in\mathcal{S}^1}\|\tilde{\mathbf{s}} - \mathbf{s}^i\|^2\right) \tag{17}$$

where $\tilde{\mathbf{s}}$ is the MMSE equalized output, $\mathcal{S}^0$ and $\mathcal{S}^1$ represent the subsets of constellation points in which the j^{th} bit of the corresponding symbol is 0 and 1 respectively, and η_i is the post-detection SINR for the i^{th} stream. This involves the calculation of squared Euclidean distances and minimizer functions which can unravel with higher modulation orders and number of UEs. Reference [39] exploits the Gray mapping of constellation symbols to propose a low-complexity LLR computation scheme that avoids multipliers by using a piecewise linear mapping [18]. In this paper, the max-log approximation in (17) is adopted for LLR calculations and a hardware reuse scheme in presented in Section IV that significantly curtails the resource consumption, in contrast to straightforward implementations.



(a) 32 BS antennas (b) 64 BS antennas (c) 128 BS antennas

FIGURE 4. Bit error rate versus average SNR per receive antenna for 16 User Equipment (UE) and increasing number of BS antennas.

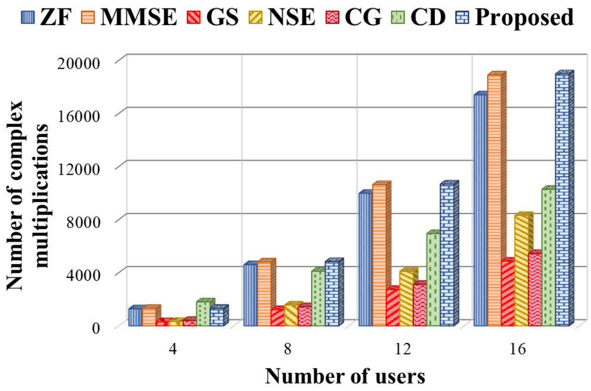


FIGURE 5. Number of complex multiplications for ZF, MMSE and approximation-based detectors for 64 BS antennas and different number of users.

TABLE 2. Computational complexity in terms of complex multiplications and divisions.

Algorithm	Multiplications	Divisions
ZF-BS	$N_R N_T + N_R N_T^2$	$2N_T$
MMSE-BS	$\frac{N_T^3}{3} + N_R N_T^2 + \frac{N_T^2}{2} + N_R N_T + \frac{N_T}{6}$	$2N_T$
GS	$2N_R N_T^2 + KN_T^2 + \frac{N_T}{2}$	N_T
NSE	$2N_R N_T^2 + (K-1)(\frac{N_T^3}{2}) + \frac{N_T^2}{2} - \frac{N_T}{2}$	N_T
CG	$2N_R N_T^2 + (K+1)(N_T^2 + 5N_T)$	N_T
CD	$2N_R N_T^2 + K(2N_R N_T + N_T)$	N_T
Proposed	$\frac{N_T^3}{3} + N_R N_T^2 + \frac{3N_T^2}{4} + N_R N_T + \frac{11N_T}{12}$	1

B. ERROR-RATE PERFORMANCE

In order to evaluate the detection error-rate performance, various antenna-user configurations are tested using Monte-Carlo simulations assuming Rayleigh fading channels. Fig. 2 plots the achieved uncoded bit error rate (BER) of the MMSE and existing approximation-based algorithms versus the average signal to noise ratio (SNR) per receive antenna for different number of BS antennas, when the number of user equipment (UE) is 4. It can be observed that for twice the number of UEs, most approximation algorithms cannot minimize the BER even with higher SNR and more iterations, while their performances only improve when the number of BS antennas is at least eight times that of the UEs. It can be seen that in this case, GS and optimized coordinate descent (OCD) with $K = 3$ approach the MMSE performance only when the number of receive antennas is 32 (i.e. 8 times N_T). Even in this setting, CG with $K = 2, 3$, OCD with $K = 2$ and Neumann with $K = 3$, fail to meet the MMSE performance for higher SNR and require more iterations to converge if at all. Figs. 3 and 4 capture a similar trend with increasing number of BS antennas when that of the UEs is 8 and 16 respectively.

The observation that MMSE performs well with N_R just twice that of N_T in many practical cases is very significant from the energy efficiency perspective as the approximative algorithms trade-off the inherent MMSE capability for lower hardware complexity by unnecessarily increasing the number of BS antennas. This gives rise to a corresponding huge increase in RF power which is often ignored, but has severe cost and environmental limitations.

C. COMPUTATIONAL COMPLEXITY

Like the reliability of a MIMO detection approach, computational complexity is also an equally important characteristic that needs to be evaluated. In this subsection, the computational complexity of algorithms are described in terms of the number of expensive floating-point operations like complex multiplications and divisions. Though hardware implementations based on fixed-point require an in-depth consideration of suitable optimizations for resource- and latency-intensive operations, the computational effort required for performing complex multiplications and divisions can serve as a reasonable estimate of an algorithm's complexity.

Table 2 compares the computational complexity in these terms for traditional ZF and MMSE; approximation algorithms like GS [25], NSE [26], Conjugate Gradient (CG) and Coordinate Descent [28]; and the proposed method. It should be noted that the computational effort of finding the Gram matrix (given by $\mathbf{G} = \mathbf{H}^H\mathbf{H}$) in the approximation techniques

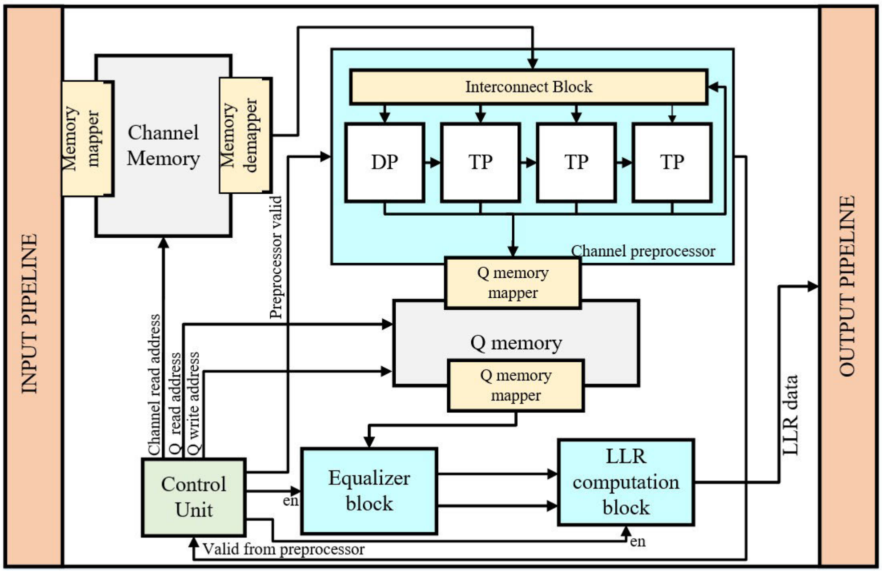


FIGURE 6. Top-level architecture of MMSE MIMO detector.

is also included for fair comparison. K refers to the number of iterations that are employed in the approximation algorithms. It can be seen that the proposed parallel-QR based MMSE detector entails slightly more multiplications than traditional MMSE but reduces the number of divisions from $2N_T$ to just 1.

Fig. 5 shows the rise in complex multiplications with the number of users while fixing the number of BS antennas as 64 and $K = 3$ for all approximation algorithms. It can be seen that exact linear detectors namely, ZF, MMSE and the proposed method have the highest complexity with ZF requiring slightly fewer multiplications. The approximate inversion based techniques can curtail the hardware complexity by a factor of roughly 4 in case of GS and CG for any number of users. NSE can also attain a four-fold reduction in complexity when $N_T = 4, 8$, but this does not scale as the number of users rises. CD needs slightly more computations than GS and CG in all cases.

However, if the detection reliability is also taken into account while evaluating the computational complexity, it was observed that MMSE reaches near-optimal performance when $N_R = 2N_T$ while the approximation algorithms generally require $N_R = 8N_T$. Fig. 7 shows the computational effort required under these practical considerations. It can be observed that the computational gains of the approximate inverse-based methods almost disappear in these settings, which further enhance the importance of exact-inversion methods.

IV. HARDWARE ARCHITECTURE DESIGN

In this section, the microarchitectural design of the reformulated MMSE detector is described in detail. The architecture was designed to conveniently enable parameterization of various MIMO configurations. The principal components of the soft-MMSE detector are the parallel QRD block, MMSE filter or equalizer block, and the LLR computation block. Fig. 6 shows a high-level overview of the proposed architecture. The channel matrix $\mathbf{H}$, receive vector $\mathbf{y}$ and regularization parameter σ are assumed as inputs to the architecture. The memory mapper and demapper modules are simple binary counters whose behaviours are controlled by a control unit (CU) to store and retrieve the corresponding augmented channel and Q matrices when necessary. The CU is also responsible for issuing appropriate control signals to facilitate seamless interaction between and within all blocks. The channel preprocessor block consists of parallel functioning diagonal process (DP) and triangular process (TP) blocks, which will be discussed later, and are internally pipelined to support multi-carrier processing. The interconnect block, composed of several multiplexers, behaves like a switch

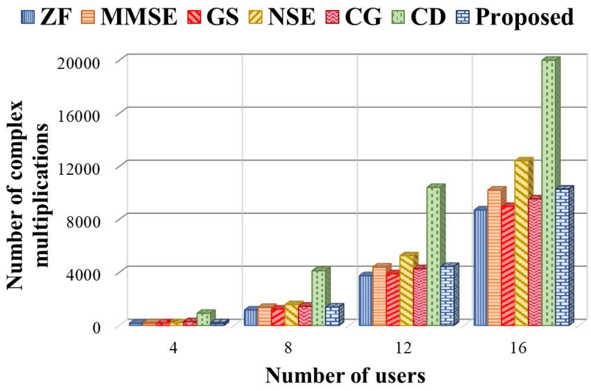


FIGURE 7. Number of complex multiplications for ZF, MMSE and the proposed algorithm with number of BS antennas = $2N_T$ and approximation-based detectors with number of BS antennas = $8N_T$.

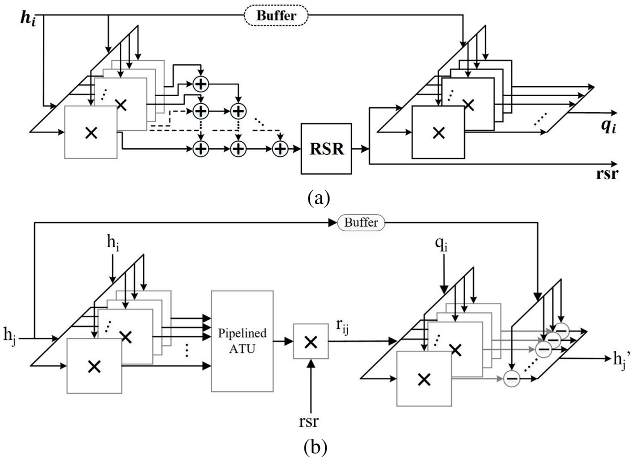


FIGURE 8. Hardware architecture of (a) Diagonal Process and (b) Triangular Process blocks.

matrix, which is responsible for apportioning the input data to the respective DP and TP blocks based on the current iteration of each subcarrier.

A. PARALLEL QRD BLOCK

As soon as a valid channel matrix is available at the input, the memory mapper block uses σ to generate the extended channel matrix $\overline{\mathbf{H}}$, which is written to the channel memory. When the channel preprocessor is ready to start performing the decomposition, the demapper retrieves the channel matrices sequentially from storage. The preprocessor comprises hardware blocks that execute the diagonal and triangular processes in parallel. Fig. 8 shows the internal architecture of the diagonal process (DP). The first stage of $(N_R + N_T)$ complex multipliers performs conjugate multiplication of $\overline{\mathbf{h}}_i$ with itself using two real multipliers for each product term. The generated terms are then passed to an adder tree that can be pipelined to reduce critical path delay. The number of pipeline stages is dependent on the desired frequency of operation and number of adder inputs, $N_R + N_T$. The successive adder stages should have a sufficient bit width to avoid overflows.

The next processing step in the pipeline is reciprocal square root computation. This can be implemented using an optimized Newton-Raphson block without the requirement of any scaling operations [40]. Two Newton-Raphson iterations were implemented in a pipelined manner to improve accuracy. This quantity is then passed to the other triangular processes as well as the second set of $(N_R + N_T)$ complex multipliers in DP to produce the orthonormal column $\overline{\mathbf{q}}_i$.

The triangular processes (TP) in each iteration can be parallelized by a maximum factor of $N_T - 1$ at most, with the number of processes declining as the algorithm proceeds to subsequent iterations. With the proposed reformulations in Algorithm 1, the parallel TPs begin execution at the start of every iteration, unlike the traditional case, where DP execution is required to be completed. Fig. 8b shows the internal hardware architecture of the TP block. The first stage of complex multipliers must perform the multiplication of $\overline{\mathbf{h}}_i^H$ with $\overline{\mathbf{h}}_j$, which typically requires four real multipliers and two adders. Owing to the replication of the TP block for parallel processing, an optimized stage is employed that uses only three multipliers and five adders to save hardware resources [41], as shown below.

$$(a + bi)(c + di) = ac - bd + ((a + b)(c + d) - ac - bd)i \tag{18}$$

A fully pipelined version of this multiplier can be realized with a latency of 6 clock cycles. The complex product terms now need to be passed to a pipelined complex adder tree unit that pipelines the successive adder stages to yield the corresponding $r_{i,j}^{\text{parallel}}$ term with a latency of $\log_2(N_R + N_T)$. Upon receiving the reciprocal square root from the DP block, the triangular coefficient $r_{i,j}$ can be computed, at which time $\overline{\mathbf{q}}_i$ is available from the DP. The second stage of the optimized complex multipliers computes the product of $r_{i,j}$ with $\overline{\mathbf{q}}_i$ and subtracts the resulting entity from buffered channel coefficients $\overline{\mathbf{h}}_j$.

While all DP and TP blocks are active in the first iteration, a finite-state machine (FSM) within the preprocessor uses appropriate signals to incrementally disable TP blocks in subsequent iterations. The orthonormal columns generated in each iteration were stored in registers until the final orthonormal column was computed. At this point, the columns of $\overline{\mathbf{Q}}$ are stored into the memory. The proposed structure permits interleaved pipelining of channel subcarriers to considerably improve throughput in addition to reducing latency. For instance, a block of P subcarriers can be sequentially retrieved by the channel preprocessor, where P is related to the latency of parallel processes. After N_T iterations, the computed $\overline{\mathbf{Q}}$ matrices are stored sequentially into memory, whereas the preprocessor begins retrieving the next block of subcarriers from the channel storage.

B. EQUALIZER BLOCK

The stored $\overline{\mathbf{Q}}$ can be partitioned to yield $\mathbf{Q}^1$ and $\mathbf{Q}^2$ which need to be multiplied according to Equation (13) to generate the MMSE filter matrix. The complexity of this matrix-matrix multiplication is given by $\mathcal{O}(N_R N_T^2)$ complex multiplications and $\mathcal{O}(N_R N_T)$ complex divisions. This straightforward implementation leads to significant hardware complexity, but this can easily be overcome by a simplified application of $\mathbf{T}$ to the receive vector.

First, a transformed receive vector denoted by $\tilde{\mathbf{y}}$ can be computed as $\tilde{\mathbf{y}} = {\mathbf{Q}^1}^H \mathbf{y}$. This can be accomplished by first passing these quantities into a network of N_R complex multipliers. Subsequently, N_T number of N_R-input adder trees is employed to deliver $\tilde{\mathbf{y}}$ to the next stage. Subsequently, $\mathbf{Q}^2$ is multiplied by $\tilde{\mathbf{y}}$ in a similar manner to deliver the N_T-dimensional vector. Because $1/\sigma$ can be pre-computed, this is succeeded by a layer of N_T constant multipliers to yield the MMSE equalized vector.

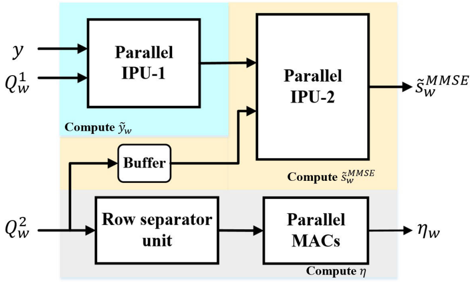


FIGURE 9. Equalizer block architecture.

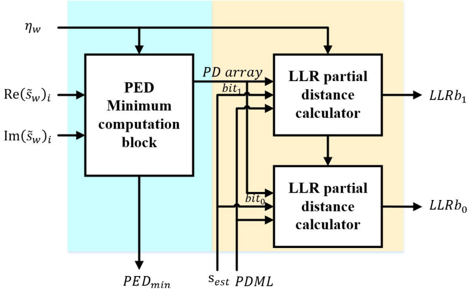

η_w
$\mathrm{Re}(\tilde{s}_w)_i$
$\mathrm{Im}(\tilde{s}_w)_i$
PED Minimum computation block
PD array
bit_1
LLR partial distance calculator
$LLRb_1$
bit_0
LLR partial distance calculator
$LLRb_0$
PED_{min}
s_{est}
PDML

FIGURE 10. LLR computation block architecture for the i_{th} equalized symbol assuming 4-QAM with 2 bits per user.

Simultaneously, the equalizer block is also tasked with computing the post-detection SNR according to Equation (16). Because this needs to be computed only once for every channel matrix, N_T complex multiply-and-accumulate (MAC) units are used to determine the SINR estimates for each layer. A high-level overview of the equalizer block architecture is shown in Fig. 9.

C. LLR COMPUTATION ARCHITECTURE

To significantly reduce the resource consumption of the traditional max-log approximation-based design from Equation (17), the LLR computation block is constructed using two stages, as shown in Fig. 10. In the first stage, the partial Euclidean distance (PED) between the i^{th} equalized symbol of $\tilde{\mathbf{s}}$ and the corresponding quadrature amplitude modulation (QAM) constellation symbols is computed. The minimum PED is extracted to determine the overall PED across all the equalized symbols given by *PDML*. The second stage consists of $\mathcal{O}(\log_2(QAM))$ partial-distance calculators with respect to the number of bits per user. These blocks use intermediate computations from the first stage, as shown in Fig. 10 as *PDarray*, which leads to significant computation reuse and hardware savings. The entire process is highly pipelined and properly synchronized to deliver LLR values in every clock cycle after the pipeline is flushed.

The LLR computation block is highly reconfigurable for various numbers of users and modulation orders. The reduced-complexity design made possible by the significant reuse of PED computations enables FPGA implementations

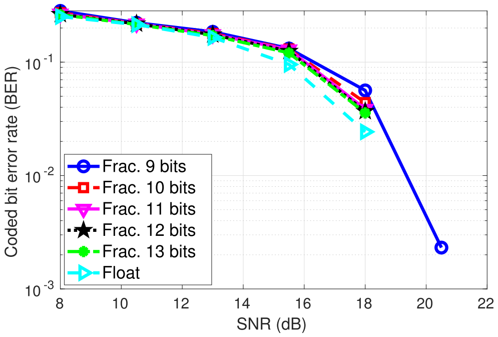


FIGURE 11. Coded BER versus SNR for varying fixed-point precisions in a 32 × 12 MIMO uplink detection.

by limiting the number of digital signal processing (DSP) slices needed for higher MIMO dimensions.

V. IMPLEMENTATION RESULTS AND DISCUSSION

This section evaluates the FPGA implementation of the proposed accelerator and compares the design with recent approximative detectors for massive MIMO in the literature. The parameterized design enables us to analyze the performance of the accelerator for different MIMO settings and modulation orders.

A. FPGA IMPLEMENTATION RESULTS

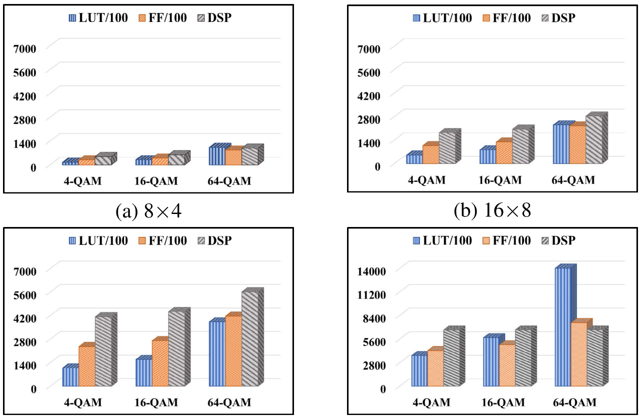
First, a fixed-point analysis was performed to evaluate the impact of different quantization formats on the detection performance of the architecture. Clustered Delay Line (CDL) channel models with the ‘B’ profile are adopted for realizing a 32 × 12 MIMO uplink system with a carrier frequency of 3.5 GHz, a subcarrier spacing of 30 kHz and a delay spread of 300 ns. Low-density parity code (LDPC) decoding with code rates of 5/6 and 64-QAM modulation were employed. Fig. 11 shows the degradation in the coded BER for different fractional precisions with respect to the floating-point implementation. It is observed that a fractional precision of 13 bits achieves the closest possible performance to the floating-point curve and is thus chosen for our implementation.

The proposed architecture is written using the Verilog hardware description language (HDL) in a parameterized manner to support any fixed MIMO configuration. The design is first implemented for two MIMO configurations supporting four and eight number of users with 64 BS antennas to compare with state-of-art approximation-based detectors that also support four and eight users respectively. The design is implemented on a Xilinx Virtex Ultrascale+ XCVU9P FPGA device considering a MIMO system with 64-QAM modulation order and the performance metrics are listed in Table 3. One of the key performance measures used in the literature is hardware throughput (HT) in megabits per second (Mbps), which can be calculated as follows

$$\mathrm{HT} = \frac{\mathrm{SC} \times N_T \log_2(\mathrm{QAM}) \times f_{\mathrm{clk}}}{\text{Latency of computing equalized output}} \tag{19}$$

TABLE 3. Comparison of the SQRD preprocessing accelerator with state-of-art FPGA-based designs.

Reference	Wu [25] [JSTSP'14]		Wu [26] [ISCAS'16]	Zhang [24] [TCAS-I'21]	Zhang [42] [TVT'19]	Wu [28] [TCAS-I'16]	Chen [43] [TCAS'I'16]	This Work	
MIMO system	64×4 64-QAM	128×8 64-QAM	128×8 64-QAM	128×8 64-QAM	128×8 64-QAM	128×8 64-QAM	128×8 64-QAM	64×4 64-QAM	64×8 64-QAM
FPGA	Virtex-7		Virtex-7	Virtex-7	Virtex-7	Virtex-7	Virtex-7	Virtex US+	
Detector	Cholesky		NSE	IGS	TMA	OCD	IIC	MMSE	
LUTs	78756 (26%)	208161 (69%)	148797 (49%)	105135 (35%)	91353 (30%)	23914 (8%)	72231 (24%)	143829 (12%)	291026 (25%)
FFs	39602 (7%)	213226 (35%)	161934 (27%)	73130 (12%)	69784 (12%)	43008 (7%)	151531 (25%)	180157 (8%)	355263 (15%)
DSPs	329 (12%)	1447 (52%)	1016 (36%)	1850 (66%)	2000 (71%)	774 (28%)	1245 (45%)	2440 (36%)	5017 (73%)
Frequency[MHz]	317	317	317	308	225	258	305	400	400
Hardware Throughput[Mbps]	301	603	621	732	630	376	915	**1028**	**1745**
Throughput/LUTs [bps/LUT]	3822	3587	4173	6962	6896	15597	12668	7147	5996
Throughput/FFs [bps/FF]	7600	3117	3835	10010	9027	8743	6038	5706	4912



(a) 8×4

(b) 16×8

(c) 24×12

(d) 32×16

FIGURE 12. VCU118 resource utilization trends for different modulation orders and MIMO sizes in terms of LUTs/100, FFs/100 and DSPs.

where the architecture-specific multiple subcarrier interleaved processing is denoted by SC (e.g., 24 in [28], 18 for the proposed design, and 1 in other cases), N_T is the number of users, QAM is the modulation order, and f_{clk} denotes the maximum clock frequency of the design. The hardware efficiency was measured in terms of the throughput (Mbps) divided by the number of look-up tables (LUTs) and flip-flops (FFs). Wu et al. [25] presented Cholesky decomposition-based detectors for 64 × 4 and 128 × 8, which were able to achieve hardware data throughputs of 301 Mbps and 603 Mbps respectively. Reference [26] presented an NSE-based detector for 128 × 8 MIMO which achieved 621 Mbps, while [24] used an improved GS method to improve the throughput to 732 Mbps. Reference [42] proposed a tridiagonal matrix inversion (TMA) detection that achieved 630 Mbps data rates. Reference [28] demonstrated a high-throughput optimized CD (OCD) detector that attained 376 Mbps throughput with comparatively fewer resources. Reference [43] proposed an intra-iterative interference cancellation (IIC) scheme that achieved a higher hardware throughput of 915 Mbps.

Although FPGA implementations of traditional MMSE detection for higher number of users are non-existent to the best of our knowledge, the proposed accelerator can support these higher MIMO configurations on a single FPGA with certain tradeoffs. The proposed 64 × 4 detector achieved more than 3 times the throughput achieved by [25] for supporting four users. It also has better hardware efficiency in terms of throughput per LUT. The proposed 64 × 8 detector achieved a hardware throughput of 1.745 Gbps, which is almost twice that of the best-performing detector for $N_T = 8$ in [43]. Although the hardware efficiency is not high compared to [28] and [43], it is comparable to that of the other detectors. Furthermore, the proposed detector can potentially attain superior BER performance compared to approximative detectors, particularly when the number of BS antennas is lower (Figs. 3a and 3b).

B. DISCUSSION

1) SCALABILITY

The parameterized architecture described in this work enables the rapid evaluation of diverse MIMO configurations to ensure scalability of the architecture. In particular, the hardware description can support any N_R, N_T and modulation order without affecting the design functionality. This facilitates the synthesis and evaluation of multiple configurations to understand the effect of the number of users and modulation orders on the hardware complexity of the proposed architecture. Vivado synthesis was performed for the XCVU9P FPGA device on the VCU118 evaluation board. Fig. 12 shows the LUT/100, FF/100, and DSP utilization for 8 × 4, 16 × 8, 24 × 12 and 32 × 16 detectors. It is observed that the 8 × 4 and 16 × 8 architectures sit very well in hardware with less than 40% DSP, 25% LUT, and 15% FF utilization. For the 24 × 12 case, DSP usage crosses 80% for 64-QAM, whereas LUT and FF usage are approximately 30% and 20% respectively. When the architecture is extended to 32 × 16,the DSP usage reaches a maximum, whereas the FF utilization is around 35-40%. It must be noted that the LUT utilization in Fig. 12 increases for 64-QAM because DSP resources on VCU118 are exhausted and LUTs are utilized to implement multiplications. It can be observed that for a fixed number of users, the hardware consumption increases linearly for

TABLE 4. Latency analysis of the design stages for different N_R, N_T and modulation orders.

Stage	Latency (in clock cycles)
Parallel-QRD	$(16 + \lceil \log_2(N_R + N_T) \rceil) N_T$
Equalizer block	$12 + \lceil \log_2(N_R + N_T) \rceil$
LLR comp. block	$15 + 2\log_2(QAM) + 2\lceil \log_2(N_T) \rceil$

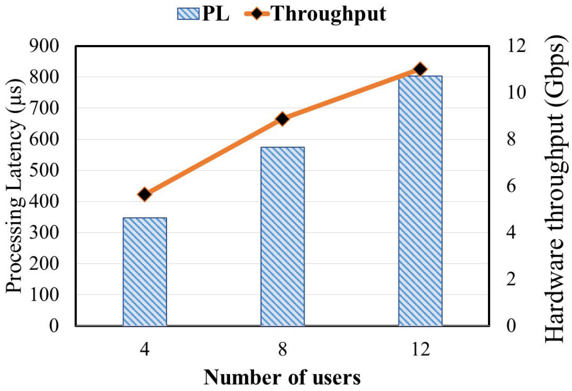


FIGURE 13. Processing latency and hardware throughput versus the number of users for a 64-QAM MIMO system with 32 BS antennas.

higher modulation orders, namely, 4-QAM, 16-QAM and 64-QAM. This was mainly due to the complexity of the LLR computation block which increases significantly for higher bits per user because of the extensive parallelism required. Alternatively, for a fixed modulation order, increasing the number of users also scales up hardware consumption owing to the highly parallel QRD and equalizer architectures.

2) PROCESSING LATENCY AND THROUGHPUT EVALUATION

The processing latency for each stage of the proposed design was examined carefully, and is tabulated in Table 4. It can be observed that the parallel-QRD block consumes the most clock cycles and is linearly proportional to N_T. The equalizer and LLR computation blocks are relatively inexpensive in terms of the clock cycles. It should be noted that because of the semi-pipelined nature of the parallel-QRD architecture, a block of $16 + \log_2(N_R + N_T)$ subcarriers can be processed iteratively and sequentially stored into memory at the end of the first stage, while the next block starts being processed.

Assuming slowly varying MIMO uplink channels with 32 BS antennas and 64-QAM, one channel estimate per slot is sufficient to reliably decode the transmitted bits [44]. In this scenario, a single instance of the LLR computation block accepting a new equalized vector in every clock cycle can attain a peak data rate of 2.4 Gbps. This is because, while the next block of channel estimates is being processed, the equalizer and LLR blocks can operate on multiple data symbols corresponding to the previous sub-carrier block. Fig. 13 shows the variation in the processing latency and hardware throughput in such a channel for an increasing number of users, considering 10 data symbols per subcarrier. The high hardware throughput arises from the highly pipelined equalizer and LLR computation blocks. Reference [34] achieved an improved throughput of 127 Mbps to 607 Mbps considering 10 received vectors per subcarrier in a slowly fading channel. Under similar considerations, the proposed design can achieve significantly higher throughputs of approximately 11 Gbps, which is greater than that in [34] considering the highest level of parallel processing. This also implies that the hardware efficiency of the proposed architecture in terms of throughput per LUT and FF can be significantly improved compared with the reported figures in Table 3.

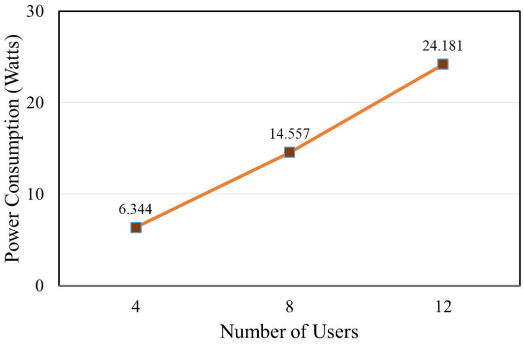


FIGURE 14. Power consumption in Watts versus the number of users for a 32 BS antenna MIMO system.

3) FPGA POWER CONSUMPTION

To understand the impact of the design parameters on the energy consumption, Fig. 14 shows the increase in dynamic power with an increasing number of users in a MIMO system with 32 BS antennas and 64-QAM modulation. The dynamic power was estimated using the Vivado Power Estimator. It is observed that the dynamic power scales linearly with the number of users.

4) SOFTWARE INTEGRATION

Fig. 15 shows the high-level hardware overview of the interaction between the host software and the Gen 3 × 16-based FPGA accelerator considering a 16 × 8 MIMO uplink system operating at 100 MHz with 30 kHz subcarrier spacing, per-slot channel estimates, and 12 data symbols. In our tests, we used the Xilinx DMA (XDMA) subsystem configured in memory-mapped mode to interact with our top-level wrapper. Xilinx DMA drivers running on an Ubuntu 20.04 OS were used to establish communication with the FPGA. The host-to-card (H2C) interface responsible for writing channel data and received vectors to the FPGA and the card-to-host (C2H) interface responsible for reading LLR data from the FPGA have measured latencies of 300μs and 175μs respectively for 50 MHz bandwidth, while the FPGA processing time is approximately 45μs. It is worth mentioning that the PCIe data transfer burden is several-fold higher for approximate

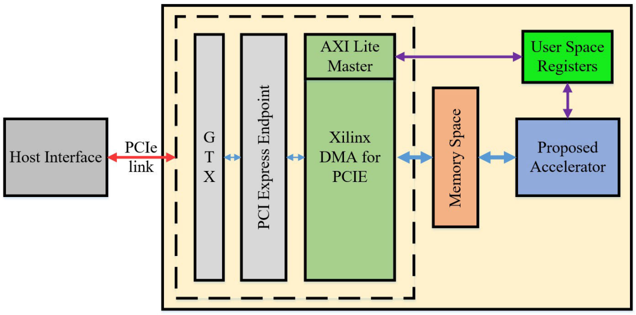


FIGURE 15. **Hardware overview of PCIe-based interfacing.**

inverse-based methods owing to the much higher N_R needed to facilitate the detection of eight users. This unnecessarily increases the PCIe latency, affecting real-time capabilities and does not make them suitable for lookaside acceleration.

5) FUTURE DIRECTIONS

Some interesting directions for further research to extend the capabilities of the proposed accelerator are briefly discussed. Complexity-reduction techniques for simplifying LLR calculations are being investigated to save DSP resources and potentially extend the accelerator to support higher MIMO dimensions. Dynamic partial reconfiguration (DPR) is also being actively investigated to make the accelerator run-time flexible to different MIMO dimensions.

There is also a great deal of interest in implementing computationally complex massively parallel non-linear (MPNL) detectors [29], [45], [46], [47], which can greatly boost the energy efficiency of Open RAN systems by supporting the same number of users as exact-MMSE solutions with half the number of BS antennas.

VI. CONCLUSION

In this paper, a hardware-friendly, parallel QRD-based soft-output exact-MMSE accelerator is proposed that achieves low processing latency and high throughput. The design scales well to support different MIMO configurations and modulation orders. The implementation results on a Xilinx Virtex Ultrascale+ device demonstrate significant throughput improvements over state-of-the-art approximation-based detectors supporting eight users, in addition to the superior BER performance achieved when the number of BS antennas is lowered. The achievable throughput of the proposed architecture can be further improved from 1.745 Gbps to 11 Gbps by considering slowly fading channels with multiple transmitted symbols per channel state information.

REFERENCES

[1] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "OpenRAN: A software-defined ran architecture via virtualization," in *Proc. ACM SIGCOMM Conf. SIGCOMM*, Aug. 2013, pp. 549–550.

[2] M. Bertuletti, Y. Zhang, A. Vanelli-Coralli, and L. Benini, "Efficient parallelization of 5G-PUSCH on a scalable RISC-V many-core processor," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Apr. 2023, pp. 1–6.

[3] A. Kelkar and C. Dick. *Introducing Nvidia Aerial Research Cloud for Innovations in 5G and 6G*. Accessed: Apr. 2, 2024. [Online]. Available: https://developer.nvidia.com/blog/introducing-aerial-research-cloud-for-innovations-in-5g-and-6g/

[4] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive MIMO for next generation wireless systems," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 186–195, Feb. 2014.

[5] J. Ding, R. Doost-Mohammady, A. Kalia, and L. Zhong, "Agora: Real-time massive MIMO baseband processing in software," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol.*, New York, NY, USA, 2020, pp. 232–244, doi: 10.1145/3386367.3431296.

[6] J. Gong, A. Kalia, and M. Yu, "Scalable distributed massive MIMO baseband processing," in *Proc. 20th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, Boston, MA, USA, Apr. 2023, pp. 405–417. [Online]. Available: https://www.usenix.org/conference/nsdi23/presentation/gong

[7] G. N. Katsaros, M. Filo, R. Tafazolli, and K. Nikitopoulos, "MIMO-SoftiPHY: A software-based PHY design and implementation framework for highly-efficient open-RAN MIMO radios," *IEEE Trans. Mobile Comput.*, early access, Jun. 10, 2024, doi: 10.1109/TMC.2024.3411788.

[8] L. Kundu, X. Lin, E. Agostini, V. Ditya, and T. Martin, "Hardware acceleration for open radio access networks: A contemporary overview," *IEEE Commun. Mag.*, early access, Nov. 6, 2023, doi: 10.1109/MCOM.023.2300281.

[9] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, Nov. 2010.

[10] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.

[11] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2131–2142, Jun. 2008.

[12] J. Yang, C. Zhang, X. You, and S. Xu, "Improved K-best algorithm for low-complexity MIMO detector," in *Proc. 6th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2014, pp. 1–6.

[13] K. Pavan Srinath and J. Hoydis, "Bit-metric decoding rate in multi-user MIMO systems: Theory," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7961–7974, Nov. 2023.

[14] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up MIMO: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, Jan. 2013.

[15] K. Nikitopoulos, J. Zhou, B. Congdon, and K. Jamieson, "Geosphere: Consistently turning MIMO capacity into throughput," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 631–642, Aug. 2014.

[16] C. K. Singh, S. Honnavara Prasad, and P. T. Balsara, "VLSI architecture for matrix inversion using modified gram-Schmidt based QR decomposition," in *Proc. 20th Int. Conf. VLSI Design Held Jointly 6th Int. Conf. Embedded Syst. (VLSID)*, Jan. 2007, pp. 836–841.

[17] J. Arias-García, R. Pezzuol Jacobi, C. H. Llanos, and M. Ayala-Rincón, "A suitable FPGA implementation of floating-point matrix inversion based on gauss-jordan elimination," in *Proc. VII Southern Conf. Program. Log. (SPL)*, Apr. 2011, pp. 263–268.

[18] 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.

[19] S. Haene, D. Perels, and A. Burg, "A real-time 4-stream MIMO-OFDM transceiver: System design, FPGA implementation, and characterization," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 877–889, Aug. 2008.

[20] L. Boher, R. Rabineau, and M. Helard, "FPGA implementation of an iterative receiver for MIMO-OFDM systems," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 6, pp. 857–866, Aug. 2008.

[21] D. Shin and J. Park, "A low-latency and area-efficient Gram–Schmidt-Based QRD architecture for MIMO receiver," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 8, pp. 2606–2616, Aug. 2018.

[22] F. Jiang, C. Li, and Z. Gong, "A low complexity soft-output data detection scheme based on Jacobi method for massive MIMO uplink transmission," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–5.

[23] 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. (SiPS)*, Oct. 2016, pp. 273–278.

[24] C. Zhang, Z. Wu, C. Studer, Z. Zhang, and X. You, "Efficient soft-output Gauss–Seidel data detector for massive MIMO systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 12, pp. 5049–5060, Dec. 2021.

[25] 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.

[26] Z. Wu, C. Zhang, Y. Xue, S. Xu, and X. You, "Efficient architecture for soft-output massive MIMO detection with Gauss–Seidel method," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1886–1889.

[27] M. Wu, C. Dick, J. R. Cavallaro, and C. Studer, "FPGA design of a coordinate descent data detector for large-scale MU-MIMO," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 1894–1897.

[28] M. Wu, C. Dick, J. R. Cavallaro, and C. Studer, "High-throughput data detection for massive MU-MIMO-OFDM using coordinate descent," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2357–2367, Dec. 2016.

[29] K. Nikitopoulos, "Massively parallel, nonlinear processing for 6G: Potential gains and further research challenges," *IEEE Commun. Mag.*, vol. 60, no. 1, pp. 81–87, Jan. 2022.

[30] G. N. Katsaros, R. Tafazolli, and K. Nikitopoulos, "On the power consumption of massive-MIMO, 5G new radio with software-based PHY processing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2022, pp. 765–770.

[31] M. Filo, Y. Xia, and K. Nikitopoulos, "SACCESS: Towards a software acceleration framework for 5G radio access networks," in *Proc. IEEE Int. Medit. Conf. Commun. Netw. (MeditCom)*, Sep. 2021, pp. 318–323.

[32] S. Stanley, "Heavy Reading's accelerating open RAN platforms operator survey," Heavy Reading, Light Reading, New York, NY, USA, White Paper, 2022.

[33] E. A. Papatheofanous, D. Reisis, and K. Nikitopoulos, "LDPC hardware acceleration in 5G open radio access network platforms," *IEEE Access*, vol. 9, pp. 152960–152971, 2021.

[34] J. Zeng, J. Lin, and Z. Wang, "An improved Gauss–Seidel algorithm and its efficient architecture for massive MIMO systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 9, pp. 1194–1198, Sep. 2018.

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

[36] R. C. Chang, C.-H. Lin, K.-H. Lin, C.-L. Huang, and F.-C. Chen, "Iterative *QR* decomposition architecture using the modified Gram–Schmidt algorithm for MIMO systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 5, pp. 1095–1102, May 2010.

[37] C. Liu, C. Tang, L. Yuan, Z. Xing, and Y. Zhang, "QR decomposition architecture using the iteration look-ahead modified Gram–Schmidt algorithm," *IET Circuits, Devices Syst.*, vol. 10, no. 5, pp. 402–409, 2016. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-cds.2015.0349

[38] C. Liu, C. Tang, Z. Xing, L. Yuan, Y. Wang, L. Chen, Y. Zhang, S. Xiang, W. Zhao, X. Hu, and J. Xu, "QRD architecture using the modified ILMGS algorithm for MIMO systems," in *Proc. Int. Wireless Internet Conf.*, 2016, pp. 164–178.

[39] I. B. Collings, M. R. G. Butler, and M. McKay, "Low complexity receiver design for MIMO bit-interleaved coded modulation," in *Proc. 8th IEEE Int. Symp. Spread Spectr. Techn. Appl.-Programme Book Abstr.*, Sep. 2004, pp. 12–16.

[40] E. Libessart, M. Arzel, C. Lahuec, and F. Andriulli, "A scaling-less Newton-raphson pipelined implementation for a fixed-point inverse square root operator," in *Proc. 15th IEEE Int. New Circuits Syst. Conf. (NEWCAS)*, Jun. 2017, pp. 157–160.

[41] D. E. Knuth, "Seminumerical algorithms," in *The Art of Computer Programming*, 3rd ed. Reading, MA, USA: Addison-Wesley, 1997.

[42] C. Zhang, X. Liang, Z. Wu, F. Wang, S. Zhang, Z. Zhang, and X. You, "On the low-complexity, hardware-friendly tridiagonal matrix inversion for correlated massive MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6272–6285, Jul. 2019.

[43] J. Chen, Z. Zhang, H. Lu, J. Hu, and G. E. Sobelman, "An intra-iterative interference cancellation detector for large-scale MIMO communications based on convex optimization," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 11, pp. 2062–2072, Nov. 2016.

[44] Y. Xia, C. Jayawardena, and K. Nikitopoulos, "Reduced complexity matrix inversions in slow time-varying MIMO channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2022, pp. 456–461.

[45] K. Nikitopoulos and R. Tafazolli, "Parallel processing of sphere decoders and other vector finding approaches using tree search," U.S. Patent 2 016 198 845 A1, Aug. 5, 2020.

[46] K. Nikitopoulos, "Massively parallel and flexible processing for MIMO systems," in *Wiley 5G Ref.* Hoboken, NJ, USA: Wiley, 2019, pp. 1–21.

[47] K. Nikitopoulos, G. Georgis, C. Jayawardena, D. Chatzipanagiotis, and R. Tafazolli, "Massively parallel tree search for high-dimensional sphere decoders," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 10, pp. 2309–2325, Oct. 2019.

THOMAS JAMES THOMAS (Member, IEEE) received the Ph.D. degree in VLSI signal processing from Indian Institute of Space Science and Technology, Thiruvananthapuram. He is currently a Research Fellow with the 5G/6G Innovation Center (5G/6GIC), University of Surrey, U.K. His research interests include the design of advanced signal processing architectures for real-time applications focused on future wireless and open RAN communication systems.

KONSTANTINOS NIKITOPOULOS (Senior Member, IEEE) is currently an Associate Professor (a Reader) with the Institute for Communication Systems, University of Surrey, U.K., and the Director of the Wireless Systems Laboratory. He is an active Academic Member of the 5G/6G Innovation Centre (5G/6GIC), where he leads the "Theory and Practice of Advanced Concepts in Wireless Communications" research area. He is also the Technical Lead of the Open RAN Implementation of the University of Surrey. His research interests include the physical layer (PHY) aspects of pragmatic, energy and latency-efficient wireless communication systems that "work in practice" and, in particular, on the intersection of advanced, highly efficient, and physical layer (PHY) processing design; advanced computing architectures; and system level design and demonstration. He was a recipient of the prestigious First Grant of the U.K.'s Engineering and Physical Sciences Research Council.