IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 7, 2019, accepted February 14, 2019, date of publication February 27, 2019, date of current version April 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2901944

A Cost-Effective Adaptive Overlapped Cluster-Based MIMO Detector in a Frequency Domain Reconfigurable Modem

YUAN-TE LIAO AND TERNG-YIN HSU

Department of Computer Science, College of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan

Corresponding author: Terng-Yin Hsu (tyhsu@ cs.nctu.edu.tw)

This work was supported by the Center for Open Intelligent Connectivity from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan.

ABSTRACT Not only to reduce the candidates but also to maintain detection performance in multiple-input multiple-output (MIMO) detection, an adaptive overlapped cluster (AOC) scheme to balance detection error and computing cost is built for 4×4 and 8×8 MIMO-OFDM systems with up to 256 quadrature amplitude modulation (QAM). The constellations are partitioned into several clusters. A cluster with size decided by channel status is chosen for signal decoding. Different partition schemes are combined to minimize the numbers of clusters required to cover a candidate symbol as the pre-estimated signal falls at cluster edges, namely overlapped clustering. The simulations of a 4×4 MIMO OFDM with 64 QAM and 8×8 MIMO OFDM with 256 QAM hint that the AOC-based detection requires an additional 0.57 dB and 1.02 dB compared to maximum likelihood (ML). Compared with K-best sphere decoding (SD), it is reduced the computing complexity to 24.50% $\sim$ 56.25% in 4×4 MIMO OFDM and, 35.00% $\sim$ 56.25% in 8×8 MIMO OFDM. In addition, the proposed scheme is ported to a reconfigurable frequency-domain (FD) modem, which is designed and implemented via TSMC 45-nm technology, with multi-rate clocking and processing elements (PEs) upgrading for supporting the proposed MIMIO detection. The results show that the throughput is 1077.8 Mbps with 4×4 64-QAM modulations.

INDEX TERMS Low complexity, MIMO detection, multi-rate clock, cluster, reconfigurable modem.

I. INTRODUCTION

Wireless communication techniques have developed significantly in recent decades. Multi-input multi-output orthogonal frequency division multiplexing (MIMO-OFDM) systems as well as large number of antennas and high-order quadrature amplitude modulation (QAM) have been widely used to support increasing data rates. However, these increase the computing complexity of the MIMO detection, leading to the challenge of how to design a MIMO detection algorithm with acceptable computing complexity and performance.

In uncoded MIMO-OFDM systems, MIMO detection can be mainly divided into two categories [1], [2]: linear detection, such as the zero forcing detection (ZFD) and the minimum mean squared error detection (MMSED), and non-linear detection such as maximum likelihood (ML). Linear detection methods are simple to implement, but have poor performance, especially in systems with the same numbers of transceiver and receiver antennas [2]. In contrast, non-linear detection methods can reach the optimal performance mathematically, but has unacceptable computing complexity [2]–[4]. Maximum likelihood post-detection processing (ML-PDP) [5] is employed to reduce computing complexity of the ML method, and numerous studies discuss compromises between operation complexity and performance.

The sphere decode (SD) method is a well-known MIMO detection method. The main idea of SD detection [6] is to confine the number of searching constellation points, although the associated algorithm still needs to span redundant searching sets exponentially in the worst case. Many studies have focused on decreasing the searching set. For example, the 256-QAM LR-Aided Fixed-Complexity Tree Searching (LRA FCTS) [7] is proposed to make computation

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

cost effectively via fixing tree size of the full expansion in the fixed-complexity sphere decoder (FSD). And the positive diagonal values (PSV) technique [8] was proposed to reduce the nodes visited times in node-pruning processing. The literature also discusses how to implement SD-based MIMO detection under hardware resource limitations [9]–[12]. The computing complexity of SD has been shown to decrease as the performance decrease [13], but there is a 16-QAM K-best Schnorr-Euchner (KSE) [10] to reduce computational complexity with near-ML performance which modifies the K-best SD via Schnorr-Euchner strategy. To balance the link performance and computational complexity, an optimized SD algorithm and the corresponding architecture by using a state-machine is reported [14], [15]. Similar to SD, Layered ORthogonal lattice Detector (LORD) is proposed to reduce the computational complexity. A enhanced LORD reduces the computing effort via choosing the constellation subset according to the received symbol [16]. And, the concept of LORD algorithm is suitable for high-QAM MIMO detection [17] Similar to SD algorithm, cluster algorithms are employed to shrink the searching space. Our previous works employed overlapped clusters to avoid performance loss which is caused by estimated signal allocating on clusters' edge [18]. For reducing the search space, an efficint low-complexity detector via dividing adjacent decision regions with horizontal or vertical edge [19].

Vertical bell laboratories layered space-time (V-BLAST) is a low-complexity MIMO detection whose basic concept is to decode signal starting from the received signal with the strongest power [20]. After the first signal is determined, the error between this signal and the strongest received signal is employed to identify the signal with second strongest power, and so on until the last signal is received. The literature notes that sorted QR decomposition can be employed in the V-BLAST algorithm to simplify successive detection processes in V-BLAST [21]. In addition, methods to improve the successive interference cancellation (SIC) efficiency have also been proposed [22]–[26] for V-BLAST. For instance, a 16-QAM MIMO detection [22], which is combined with ZF, ML, and SIC processing to minimize the computing cost.

The LR-aided MIMO detection using a coordinate transformation by adapting the Lenstra-Lenstra-Lovász (LLL) [27] algorithm has become well-known. The concept of the LR algorithm is to search a short and near-orthogonal path according to a lattice basis. However, the LLL algorithm is not efficient because the execution time is decided by column swap operations [28]. Hence, several studies have proposed different strategies to optimize the LR-aided algorithm [28]–[37], a LR-Aided K-best detection [38] combines hardware-optimized LLL (HOLLL) and K-best algorithm to approach ML performance. And a 64-QAM lattice-reduction-aided minimum-mean-squared-error-based fixed-complexity sphere decoder (LRA-MMSE-FSD) [39] makes FSD low computational cost via reducing the number of full expansion (FD) stage. In 8 × 8 MIMO, a 64-QAM MIMO detection base on fixed-complexity Effective Lenstra-Lenstra-LovÃąsz (fcELLL) algorithm [37] is introduced to improve the computational efficiency of Effective Lenstra-Lenstra-Lovász (ELLL).

In hardware implementation, there are two architectures: application-specific integrated circuit (ASIC) and application-specific instruction-set processor (ASIP). For ASIC implementation, a soft-output MIMO detector [40] that employs a lattice-reduction-aided soft-output K-best detector to remove the iteration loops for enhanced throughput has been reported. An optimal soft MIMO detector [41] combines the advantages of depth-first approaches and breadth-first approaches to provide a soft ML solution without exhaustive searching. A scalable K-best MIMO detector's [42] critical path length is constant and independent of the antenna number. For reducing the computing complexity of preprocessing in MIMO detection with high antenna number, a K-best detector combines Cholesky sorted QR decomposition and partial iterative lattice reduction [43]. Finally, a 2-D sorting detector [44] is proposed to parallel the K-best procedure for enhancing the throughput and reducing the storage elements. Besides the ASIC architecture, a MIMO detector is designed by using a reconfigurable ASIP (rASIP), which is constructed by coarse-grained reconfigurable architecture that supports different matrix operations [45].

Many algorithms have been proposed to reduce the complexity of the searching procedure in MIMO detection algorithms. For instance, Bayesian theory [46], [47] or maximum a posterior (MAP) [48] techniques have been employed to find a statistical property to make MIMO detection more efficient. Also, it is benefited to practical implementation by slicing the MIMO detection system into several 2-by-2 MAP detection cores and paralleling them [49]. The branch and bound algorithm has been combined with the M-algorithm to reduce the complexity of the ML algorithm [50]. A QL decomposition has been employed to avoid the matrix inversion operation, thus reducing complexity [51], and allow to speed up the procedure of MIMO detection via breaking the QL decomposition into subset streams. Graph theory has also been used as a strategy to analyze the optimal path [52], [53].

In this work, the adaptive overlapped cluster (AOC)-based MIMO detection is proposed which try to save computing complexity via partitioning constellations into several clusters to limit the searching space. Several partition schemes can be employed to cluster constellations to avoid extra computing complexity when the pre-estimated signal falls at clusters' edge. The concept of the proposed algorithm is illustrated in Fig. 1. At first, several clusters are chosen according to the location of the pre-estimated signal. After that, the size of the chosen cluster is narrowed or broadened according to channel status. Simulations of 4 × 4 and 8 × 8 MIMO detection hint that the AOC technique is an acceptable compromise between computing complexity and performance. Furthermore, the proposed detection can be ported to a 4 × 4 reconfigurable modem by upgrading processing elements (PEs) and employing multi-rate clock in the arithmetic logic unit (ALU). Additionally, a 454.8 KGE reconfigurable

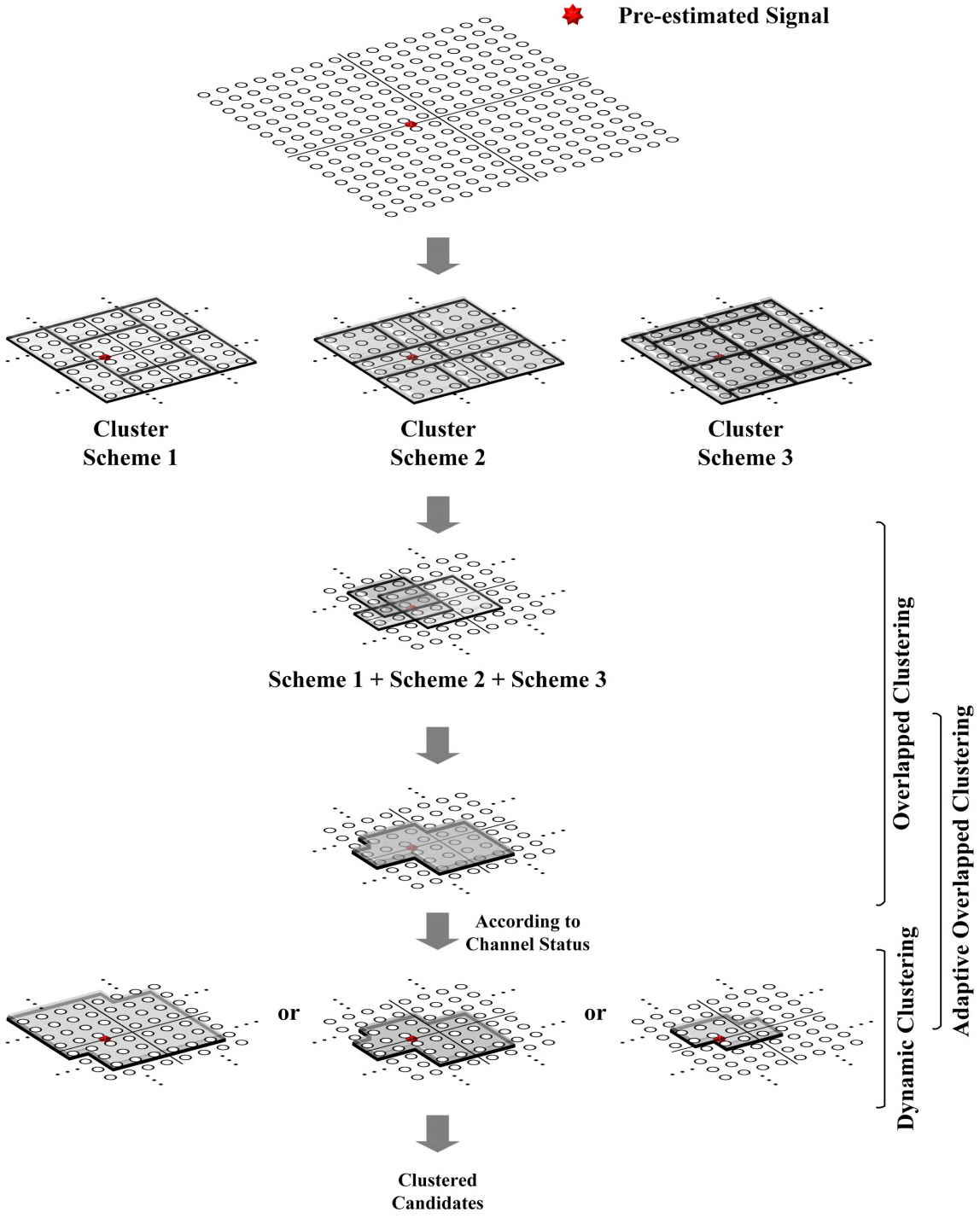

FIGURE 1. Concept of adaptive overlapped clustering.

frequency-domain (FD) modem with 1077.8 Mbps throughput is proposed for porting the proposed AOC MIMO detection.

The remainder of this paper is organized as follows. Section II presents assumptions of the system and the problem statement. Sections III introduces the concept of the AOC MIMO detection. The proposed algorithm is ported to a frequency domain (FD) reconfigurable modem in Section IV. In Section V, the proposed AOC scheme is simulated and compared with prior works. Finally, Section VI provides conclusions.

II. SYSTEM ASSUMPTIONS AND PROBLEM STATEMENT

A. SYSTEM DESCRIPTION

Assume an uncoded MIMO-OFDM system denoted as $N_T \times N_R$, where N_T is the number of transmitter antennas and

N_R is the number of receiver antennas. In this study, 4 × 4 and 8 × 8 systems are simulated. To maintain a focus on performance of the MIMO detection algorithm, we further assume perfect timing and frequency synchronization for avoid the additional errors which is caused by synchronizations. The rich-scattering environment is considered to be additive white Gaussian noise (AWGN). The received baseband signal for the $N_T \times N_R$ MIMO system can be modelled as eq. (1).

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

where $\mathbf{x}^j = \begin{bmatrix} x_1^j & x_2^j & \dots & x_N^j \end{bmatrix}^T$; $[*]^T$ means transpose; x_i^j is the j-th transmitted subcarrier modulated with N-QAM constellation, where N is 64 or 256 for both 4×4 system and 8×8 system, and transmitted by the i-th transmitted antenna; $\mathbf{y}^j = \begin{bmatrix} y_1^j & y_2^j & \dots & y_N^j \end{bmatrix}^T$ is the received symbol vector in the received signal space; and $\mathbf{n}^j = \begin{bmatrix} n_1^j & n_2^j & \dots & n_N^j \end{bmatrix}^T$ is an independent identical distributed (i.i.d.) complex zero-mean Gaussian noise vector with variance σ^2 per dimension. Moreover, the j-th $N_T \times N_R$ channel $\mathbf{H}^j$ is frequency-selective fading and expressed as eq. (2), where h_{ik}^j presents the channel state information (CSI) of the j-th subcarrier transmitted from the k-th transmitter antenna to i-th receiver antenna.

$$\mathbf{H}^j = \begin{bmatrix} h_{11}^j & \cdots & h_{1N_T}^j \\ \vdots & \ddots & \vdots \\ h_{N_R1}^j & \cdots & h_{N_RN_T}^j \end{bmatrix} \tag{2}$$

B. PROBLEM STATEMENT

The SD and LORD algorithm is an attractive MIMO detection algorithm to limit the amounts of searching constellations and thus reducing computing cost and memory requirement. However, it takes extra operations to decide if constellations allocate at a given search radius for each pre-estimated signal. Thus, it is advantageous to group constellations beforehand, that is, to partitions constellations into clusters. At least two clusters need to be chosen to ensure sufficient diversity when the pre-estimated signal falls at the edge of clusters. In different fading channels, the size of the chosen cluster may be insufficient, causing performance loss, or redundant, causing unnecessary computing cost. Therefore, it is important to minimize amounts of candidates without performance degrading.

In VLSI implementations, computing power is proportional to the clock rate multiplied by the computing resource. Preferred to supply various specifications, the proposed algorithm is ported to a reconfigurable modem [54] which must be upgraded to ensure sufficient computing power for the proposed detection porting. Because speeding up the clock rate makes chips hard to configured and increasing the computing resources leads to complex interconnection and large chip area, the impact of layout, interconnection, and area are additional important design issues.

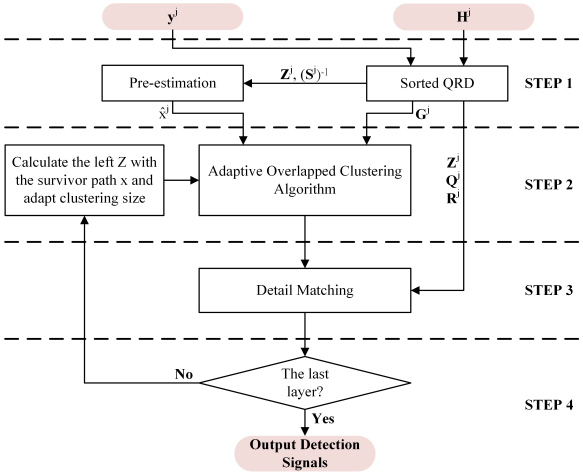

FIGURE 2. Flow chart of the proposed algorithm.

III. PROPOSED MIMO DETECTION

The basic concept of the AOC-based MIMO detection is to partition the constellations into several clusters. To address the diversity problem which is caused by pre-estimated signals falling at the edge of clusters, a multiple partition scheme allows more than one cluster to possess the same constellation point. The best cluster is chosen according to the allocation of an estimated symbol by detail matching, and the size of the chosen cluster is variable according to channel gain of each antenna to reduce the computing cost in a high signal-to-noise ratio (SNR) environment or to avoid performance loss in a low SNR environment. As illustrated in Fig. 2, the flow of the proposed MIMO detection can be divided into four major steps.

1) **Pre-processing.** Pre-processing is combined by Pre-estimated and Sorted QRD. In Sorted QRD, the channel matrix $\mathbf{H}^j$ is decomposed to unitary matrix $\mathbf{Q}^j$ and upper-triangular $\mathbf{R}^j$, where $\mathbf{H}^j = \mathbf{Q}^j \times \mathbf{R}^j$, via the sorted QR decomposition (SQRD) according to channel gain. The SQRD processing outputs the sorted received signal $\mathbf{Z}^j$, inverted sorted channel matrix $(\mathbf{S}^j)^{-1}$, sorted unitary matrix $\mathbf{Q}^j$,sorted upper-triangular matrix $\mathbf{R}^j$, and sorted channel gain $\mathbf{G}^j$. Simultaneously, the proposed algorithm exploits linear detectors to obtain pre-estimating signals $\hat{\mathbf{x}}^j$ in Pre-estimation, where $\mathbf{Z}^j = \mathbf{S}^j \times \hat{\mathbf{x}}^j$.
2) **AOC scheme.** The AOC algorithm partitions the constellations into a number of clusters using different partition schemes. Cluster sizes are increased or decreased according to the column norm of $\mathbf{H}^j$, which includes channel gain information. Then, the AOC algorithm chooses a cluster according to the pre-estimated signal $\hat{\mathbf{x}}^j$.
3) **Detail matching.** This step is adapted from breadth-first searching procedure [10]. The partial Euclidean distance (PED) of the received signal and the constellation points in the chosen cluster is computed. The PEDs

of candidate signals and the received signals in the searching space are compared and the signals with the K smallest PED are kept.

4) **Termination criterion.** The candidate signal with the lowest PED is selected as the estimated signal. If the decoding antenna level is not the last one, return to step 2. Otherwise, the estimated signal is reordered and output.

These steps are explained in more detail in the following subsections

A. PRE-PROCESSING

The purpose of pre-processing is to rearrange the columns of the channel matrix $\mathbf{H}^j$ according to the norm of the channel columns, and to determine the pre-estimated signal $\hat{\mathbf{x}}^j$ employed in the AOC algorithm. The pre-processing procedure includes two functions, Sorted QR Decomposition and Pre-Estimation.

1) SORTED QR DECOMPOSITION

We employ the MMSE-SQRD to decompose the channel matrix $\mathbf{H}^j$ into $\mathbf{Q}^j$ and $\mathbf{R}^j$ according to the channel gain $\mathbf{G}^{i,j}$, where $\mathbf{G}^i \equiv \|\mathbf{h}^i_{nor}\|^2$ and $\mathbf{H}_{nor} \equiv \left[\mathbf{h}^1_{nor}\ \mathbf{h}^2_{nor}\ \ldots\ \mathbf{h}^n_{nor}\right]$. Normalized channel matrix $\mathbf{H}_{nor}$ is defined in Eq. (3) where $h_{max} = \max\{h_{11}, \ldots, h_{N_R N_T}\}$. Thus, the decoding procedure of the detail matching step starts at the strongest signal [55].

$$\mathbf{H}_{nor} = \begin{bmatrix} \frac{h^j_{11}}{h_{max}} & \cdots & \frac{h^j_{1N_T}}{h_{max}} \\ \vdots & \ddots & \vdots \\ \frac{h^j_{N_R1}}{h_{max}} & \cdots & \frac{h^j_{N_R N_T}}{h_{max}} \end{bmatrix} \tag{3}$$

2) PRE-ESTIMATION

Pre-Estimating uses linear detection to produce the pre-estimated signal $\hat{\mathbf{x}}$ according to the sorted received signals $\mathbf{Z}$ and $\mathbf{H}$ [55]. The goal of pre-estimating is to reduce computing complexity in the AOC procedure. In this work, the MMSE approach is used to estimate $\hat{\mathbf{x}}$.

B. AOC SCHEME

To balance detection error and computing cost, the AOC scheme is divided into overlapped clustering and dynamic clustering. The best cluster is identified according to the pre-estimated signal $\hat{\mathbf{x}}$ via the following decoding procedure.

1) OVERLAPPED CLUSTERING

Overlapped clustering is composed of cluster choosing and boundary removal, as described in the following sections. A simplification of 256-QAM overlapped clustering in the first quadrant is illustrated in Fig. 3.

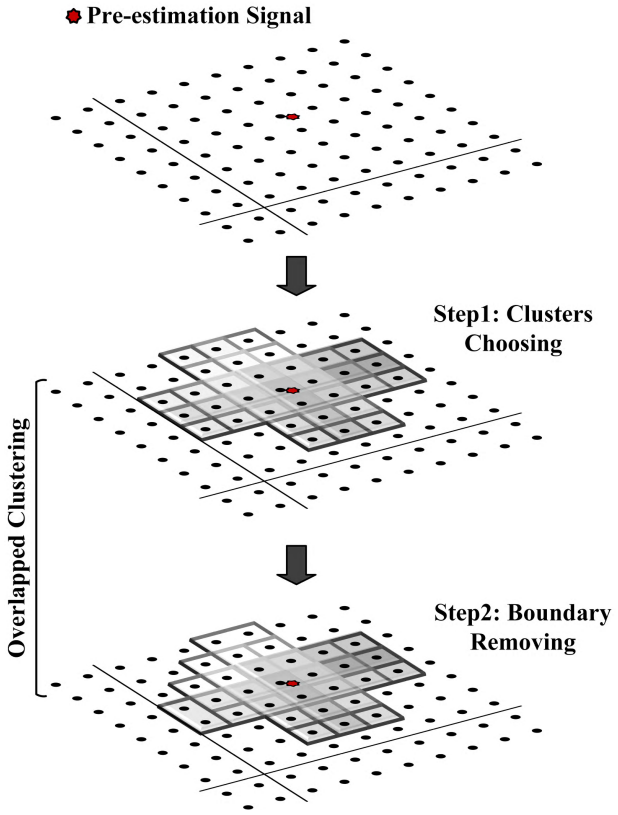

FIGURE 3. Overlapped clustering in first quadrant of 256 QAM.

Algorithm 1 Cluster Selection

Pre-estimated
Input: signal $\hat{\mathbf{x}}$, Pre-partitioned clusters $\mathbf{C}$
Output: Sets of the cluster, $\boldsymbol{\omega}^i$

```
for i = 1:N_R do
    ω^i ← ∅
    ω^i_temp ← ∅
    for each cluster C do
        if C ∋ x̂ then
            ω^i_temp = ω^i_temp ⋃ C
        else
            Do nothing
        end if
    end for
    ω^i ← ω^i_temp
end for
```

a: CLUSTER CHOOSING

In order to save memory in implementations, we pre-partition the constellations into several clusters $\mathbf{C}_k$ according to various schemes. To additionally ensure a diverse set of candidates, we select clusters which include the pre-estimated signal $\hat{\mathbf{x}}$. In $\boldsymbol{\omega}^i$, the resulting set of clusters, i indicates the layer index. Cluster selection is illustrated in Step 1 of Figure 3 and is described in Algorithm 1.

b: BOUNDARY REMOVAL

To reduce computational costs when the pre-estimated signal $\hat{\mathbf{x}}$ falls on the edge of a cluster, the cluster is dropped if

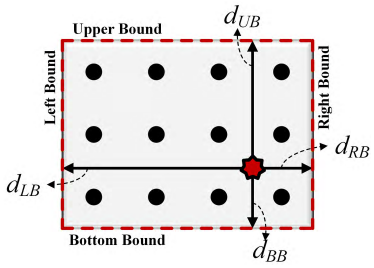

FIGURE 4. Expression for distance of estimated signal to cluster boundaries.

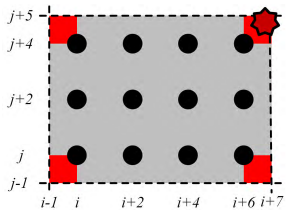

FIGURE 5. The smallest removal case.

Algorithm 2 Boundary Removal

Pre-estimated

Input: signal $\hat{\mathbf{x}}$, sets of cluster, $\boldsymbol{\omega}^i$, which include $\hat{\mathbf{x}}$, and boundary threshold T_B

Output: Candidate cluster sets $\boldsymbol{\omega}_c^i$

$\boldsymbol{\omega}_c^i \leftarrow \emptyset$
for each cluster $\mathbf{C} \in \boldsymbol{\omega}^i$ **do**
  Compute d_{UB}, d_{BB}, d_{LB}, and d_{RB}
  Compute ER according to Eq. (4)
  if $ER < T_B$ **then**
    Remove $\mathbf{C}$ from $\boldsymbol{\omega}^i$
  else
    Keep $\mathbf{C}$ in $\boldsymbol{\omega}^i$
  end if
end for
$\boldsymbol{\omega}_c^i \leftarrow \boldsymbol{\omega}^i$

the edge ratio ER, which is defined in Eq. (4), is smaller than a given boundary threshold T_B. The smallest removal case is illustrated in Fig. 5. If $\hat{\mathbf{x}}$ is allocated on the corners (red area in Fig. 5) of the 3×4 (or 4×3) cluster, such cluster is removed. The edge ratio of such case is $1/7 \times 1/5 = 1/35$ Thus, T_B is defined as $1/35$ in this work. Boundary removal is illustrated in Step 2 of Figure 3 and is described in Algorithm 2.

$$ER = \frac{min(d_{RB}, d_{LB})}{max(d_{RB}, d_{LB})} \times \frac{min(d_{UB}, d_{BB})}{max(d_{UB}, d_{BB})} \tag{4}$$

2) DYNAMIC CLUSTERING

To compensate for performance losses caused by the limited combinations of multiple clusters, the size of the candidate clusters is updated taking into account the channel status of each layer and antenna. In the AOC algorithm, we use dynamic clustering to balance detection error and computing costs. Dynamic clustering broadens the candidate cluster set $\boldsymbol{\omega}_c^i$ to reduce the probability of detection error when the channel gain G^i is too low, and trims $\boldsymbol{\omega}_c^i$ to reduce computing costs when G^i is high. Thresholds T_1 and T_2 are used to control whether $\boldsymbol{\omega}^i$ is enlarged or narrowed. If G^i is smaller than T_1, then $\boldsymbol{\omega}^i$ is enlarged by 1 constellation point. Else, if G^i is larger than T_2, then $\boldsymbol{\omega}^i$ is narrowed by 1 constellation point. A dynamic clustering example is illustrated in Fig. 6. As the ideal CFR matrix approximates a unit matrix, which corresponds to a channel gain close to 1. In Fig. 7 (a), the simulation shows that BER performance and T_1 are similar as $T_1 = 1$, 0.9, or 0.8. Thus, T_1 is set to 0.8 for balancing performance and computing complexity. In Fig. 7 (b), BER performance are similar as T_2 being infinite and 1.2. Infinite T_2 means that the candidate cluster is never narrowed. Hence, T_2 is set to 1.2. If G^i lies between T_1 and T_2, $\boldsymbol{\omega}^i$ is invariant. The pseudo code is presented in Algorithm 3. The inputs are the candidate cluster $\boldsymbol{\omega}^i$ and channel gain G^i for each antenna i of which $\boldsymbol{\omega}^i$ is broadened or narrowed according to G^i. Finally, the adjusted $\boldsymbol{\omega}^i$ is output and used in the detail matching step.

Algorithm 3 Dynamic Clustering

Input: Channel gain G^i, Candidate cluster $\boldsymbol{\omega}^i$

Output: Candidate set $\boldsymbol{\omega}^i$

Given threshold values $\{T_1, T_2\}$, where $T_1 < T_2$.
for $p = 1 : m$ **do**
  $q = p + 1$
  if $G^i < T_1$ **then**
    Enlarge $\boldsymbol{\omega}^i$
  else if $G^i > T_2$ **then**
    Narrow $\boldsymbol{\omega}^i$
  else
    $\boldsymbol{\omega}^i$ is not changed
  end if
end for
Output candidates set $\boldsymbol{\omega}^i$

IV. MULTI-RATE ARCHITECTURE

We revised the FD MIMO-OFDM reconfigurable modem [54] to provide enough computing power for the proposed AOC-based MIMO detection. The revised modem is implemented in HDL and synthesized using Synopsys's Design Compiler with TSMC's 45-nm technology. In the ALU of the revised modem, basic PEs such as adders, multipliers, and dividers are driven by an 80-MHz or 160-MHz clock and accelerators while the sorter and matrix inverter are driven by a 320-MHz clock as a compromise between computing power and routing complexity. With multi-rate clocking, PE counts are controlled to facilitate VLSI implementation. Below we present details on the modifications and port scheduling. There are 10 R/W ports between the SIMD-based ALU and the centralized memory pool. The block diagram of the modified design is shown in Fig. 8.

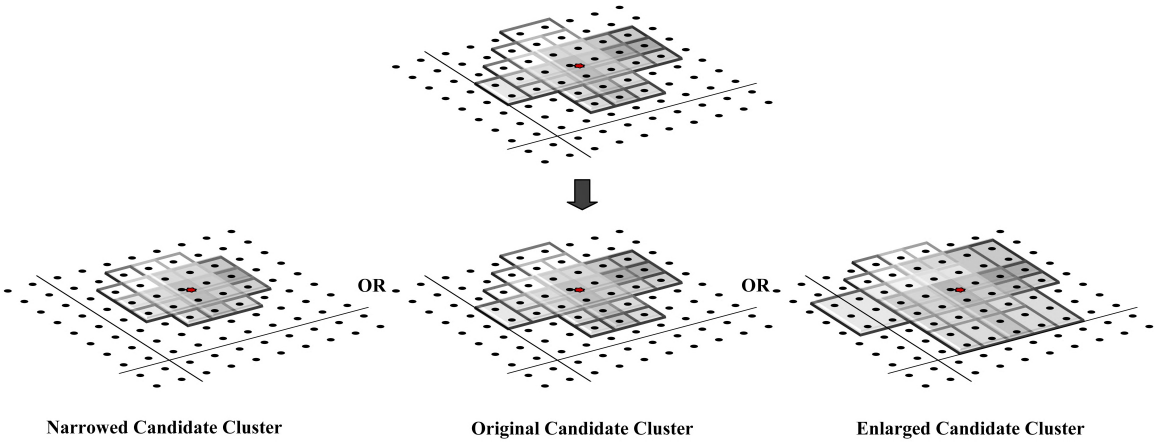

FIGURE 6. Dynamic clustering.

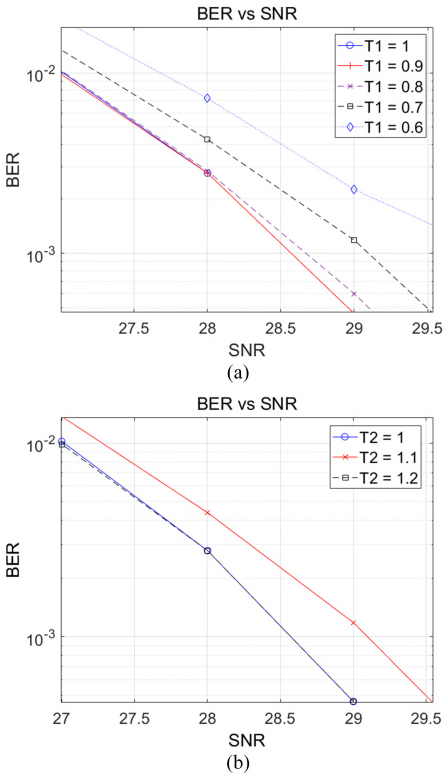

FIGURE 7. Thresholds T_1 and T_2 v.s. BER: (a) T_1; (b) T_2.

The basic system clock is 20 MHz and is applied to modules employed in the digital front-end. A multi-rate clock generator [56] is used to generate a 4× clock (80 MHz) and an 8× clock (160 MHz) based on the system clock.

A. IMPLEMENTATION LIMITS

Initially, the proposed method selects possible constellation points according to the estimated signal which is obtained by linear detectors. The performance and computing complexity is affected by the largest pre-set cluster. The relationship between BER performance and maximal cluster size with 4 × 4 64-QAM is illustrated in Fig. 9. The BER performance of 5×5 and 6×6 case are similar. However, 6×6 case takes more computing power. The largest cluster size for partitioning constellations is set to 5 × 5, as illustrated in Fig. 10. And, to maintain computational efficiency in implementations, the partitioned cases are fixed. Furthermore, the constellations are numbered, as illustrated in Fig. 11, to record the surviving candidates in MIMO detection. In this numbering system, we divide the constellation index *IdxNum* by 4, and then use the remainder to indicate the quadrant of the constellation via Eq. (5). The quotient is used to indicate complex values. After recognizing the quadrant of the allocated constellation, the constellation index $ConIdx = \lfloor IdxNum/4 \rfloor$ is employed to map the index to the complex value via Equations (7) and (6). The signs of the real and imaginary parts are indicated by the quadrant information. For example, a constellation with an index *IdxNum* of 37 is allocated on the second quadrant because $37 \mod 4 = 1$, yielding a constellation index *ConIdx* of 9, which in turn yields a complex value of $-3+5i$ because $9 \mod 4 = 1$ and $\lfloor 9/4 \rfloor = 2$.

$$IdxNum \mod 4 = \begin{cases} 0 & \text{First quadrant} \\ 1 & \text{Second quadrant} \\ 2 & \text{Third quadrant} \\ 3 & \text{Fourth quadrant} \end{cases} \quad (5)$$

$$ConIdx \mod 4 = \begin{cases} 0 & \text{Real part = 1} \\ 1 & \text{Real part = 3} \\ 2 & \text{Real part = 5} \\ 3 & \text{Real part = 7} \end{cases} \quad (6)$$

$$\lfloor ConIdx/4 \rfloor = \begin{cases} 0 & \text{Imaginary part = 1} \\ 1 & \text{Imaginary part = 3} \\ 2 & \text{Imaginary part = 5} \\ 3 & \text{Imaginary part = 7} \end{cases} \quad (7)$$

The set of possible constellation points is then broadened or narrowed according to the channel gain information. To reduce hardware costs, the maximum number of candidates in the clusters is limited to 16. Thus, the candidate signals

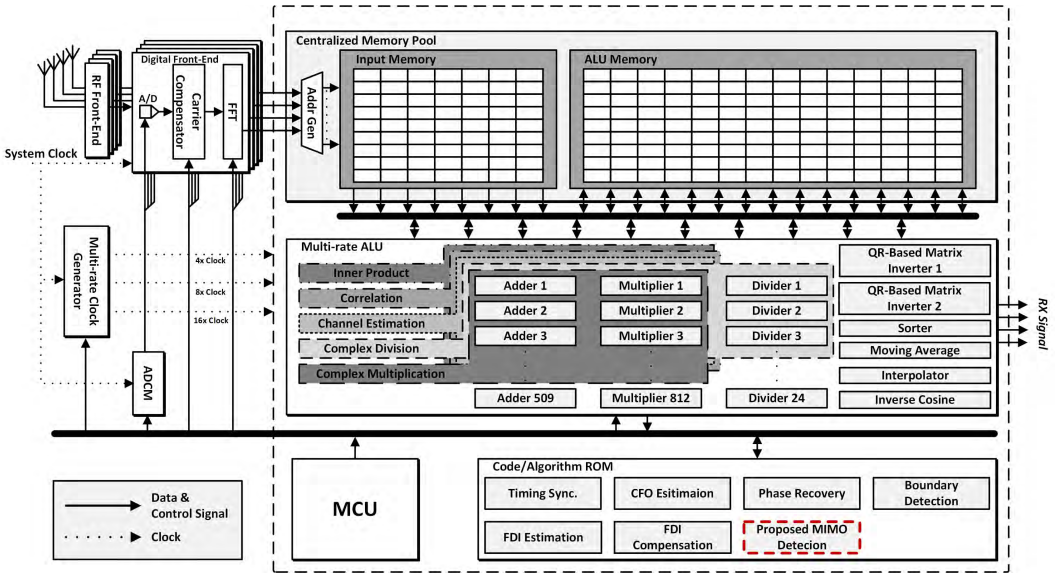

FIGURE 8. Block diagram of revised reconfigurable modem.

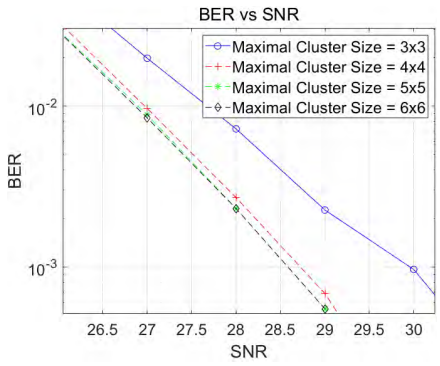

FIGURE 9. Maximal cluster size v.s. performance.

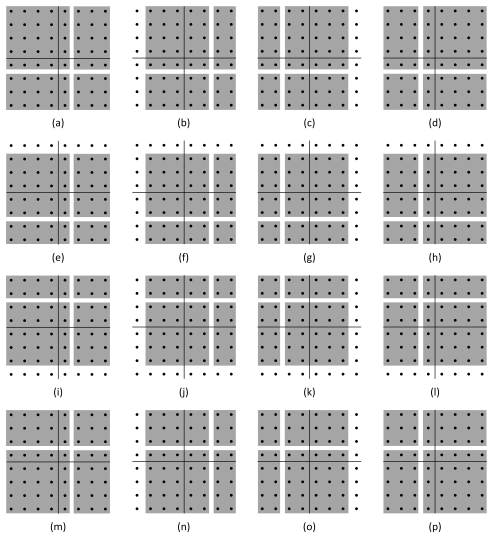

FIGURE 10. Overlapped clustering scheme in 64 QAM (fixed scale).

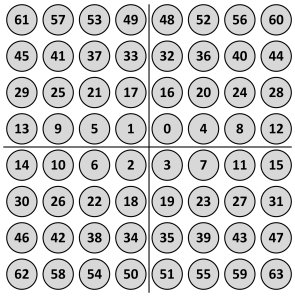

FIGURE 11. QAM mapping index.

are compared with the received signal via K-best ($K = 12$) in the SQRD search space. The output is the detection signal with the least cumulative square Euclidean distance. In this study, the reconfigurable modem is designed for 20-MHz 4×4 WLAN applications. Most of the mathematic operation units are shared. However, the original architecture does not perfectly support AOC MIMO detection: for instance, SQRD and matrix multiplications, which are used frequently in the proposed MIMO detection algorithm, are not provided in the original architecture.

Two limits must be balanced when porting AOC-based MIMO detection to the reconfigurable modem: the clock rate of the ALU and the PE resources in the ALU. To port the proposed MIMO detection, the reconfigurable modem is extended, as illustrated in Fig. 12.

1) CLOCK LIMIT

The main purpose of the single instruction multiple data (SIMD)-based ALU [54] is to maximize the utilization and efficiency of basic PEs such as adders and multipliers. Therefore, in the ALU we create advanced instructions as combinations of these basic operations. These are implemented by reconfiguring computing paths in the ALU. However,

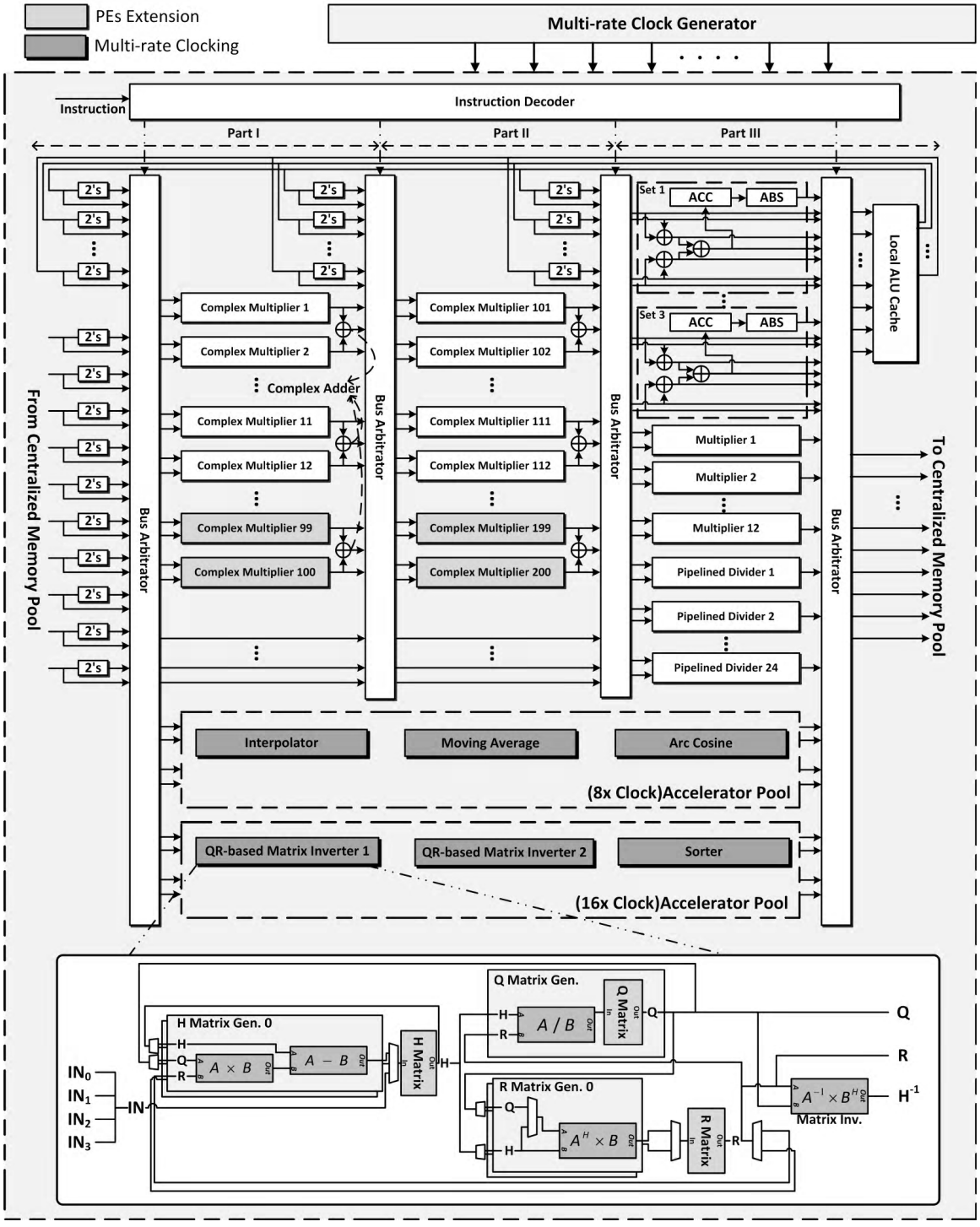

FIGURE 12. Multi-rate ALU architecture of PE resource extension.

as this leads to high routing complexity and bus connections, the clock rate of this part cannot be too high; that is, there must be a clock limit. To enhance throughput, PE resources must be increased with acceptable routing complexity and bus connections. Based on the advanced instructions supported in the original ALU, we slightly modify it to add advanced instructions to support one-cycle (160 MHz) 4×1 matrix multiplication.

2) PE LIMIT

Not every instruction can be implemented by reconfiguring computing paths. For example, matrix inversion is not suitable for this technique due to its high complexity in terms of computing paths. Thus, we use a dedicated accelerator to support such operations. As PE utilization in dedicated accelerators is inefficient, we take into account PE resources as a trade-off between utilization and throughput. In this work,

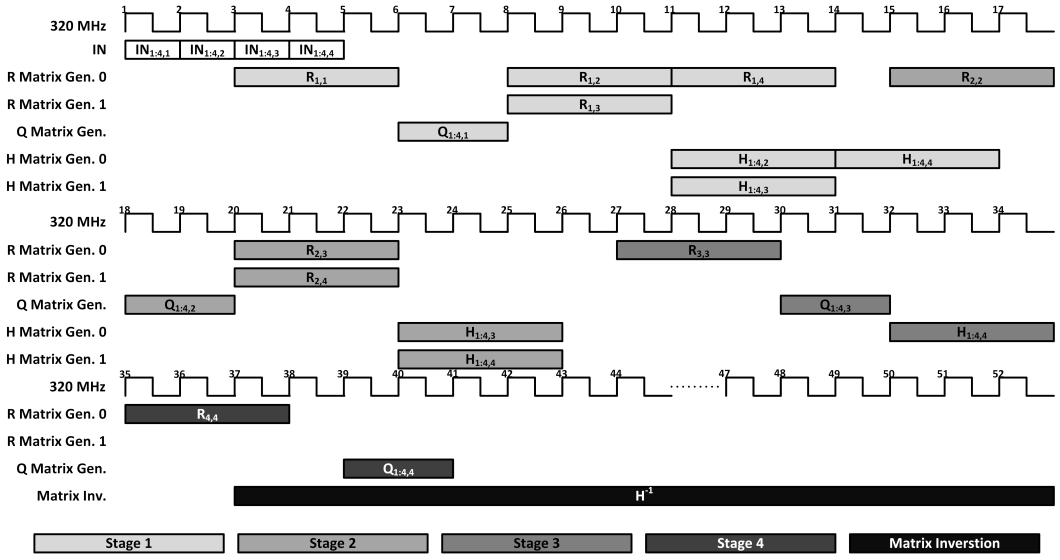

FIGURE 13. Timing diagram of QR-based matrix inverter.

we adopt a multi-clock rate strategy to balance utilization and throughput in a dedicated accelerator. To support AOC-based MIMO detection, key operations in the algorithm such as matrix inversion and QR decomposition must be added into the original ALU architecture. However, these operations have high computing complexity, especially in a 4 × 4 or more antenna system. In this work, to reduce hardware costs, QR decomposition and matrix inversion share a hardware resource: a QR-based matrix inverter. The dedicated accelerator proposed in [54] is also integrated in the multi-rate ALU. Furthermore, we support matrix multiplication, an advanced instruction. To increase flexibility in scheduling and control, we use a micro control unit (MCU) to make decisions in every algorithm, which is stored in an algorithm ROM in the modified reconfigurable modem. For increased efficiency, we propose a centralized memory pool to reduce data-access latency and maximize memory utilization.

B. HARDWARE AND INSTRUCTION ENHANCEMENT

1) QR-BASED MATRIX INVERTER

QR decomposition and matrix inversion are important operations in the proposed detection method. To meet the computing requirements of AOC-based MIMO detection, a QR-based matrix inverter is integrated in the SIMD-based ALU in sequence. The hardware design of the QR-based matrix inverter uses a modified Gram-Schmidt algorithm [57]. And the algorithm is described in Algorithm 4.

The block diagram and timing diagram are illustrated in the bottom of Figures 12 and 13, respectively. The matrix inverter takes 52 clock cycles (320 MHz) to compute and output each matrix. It employs 2 **H** matrix generators, 2 **R** matrix generators, and 1 **Q** matrix generator. **H** matrix generators are employed to subtract the columns of the **H** matrix by

Algorithm 4 Modified Gram-Schmidt Algorithms [57]

Input: Channel Frequency Response Matrix H

Output: Unitary Matrix Q, Upper-triangle Matrix R, and inverted Channel Frequency Response Matrix H^{-1}

for $k = 1 : n - 1$ **do**
  $R_{k,k} = \|H_{1:n,k}\|$
  $Q_{1:n,k} = H_{1:n,k} / R_{k,k}$
  for $j = k + 1 : n$ **do**
    $R_{k,j} = Q^{\dagger}_{1:n,k} \times H_{1:n,j}$
    $H_{1:n,j} = H_{1:n,j} - Q_{1:n,k} \times R_{k,j}$
  end for
end for
$H^{-1} = R^{-1} \times Q^{\dagger}$
Note: † means Hermitian conjugate.

the production of $\mathbf{Q}_{1:n,k}$ and $\mathbf{R}_{k,j}$ in 3 cycles (320 MHz), where **n** is the number of antennas, k is from 1 to $n - 1$, and j is from $k + 1$ to j. **R** matrix generators take 3 cycles (320 MHz) to compute the diagonal elements of **R** and the remaining elements of upper-triangle matrix **R**. The **Q** matrix generator takes 2 cycles (320 MHz) to compute the columns of the unitary matrix **Q**. To increase utilization of function, the pipeline architecture is also used to computing the matrix inversion of **H** as well. After completing QR decomposition, the $\mathbf{H}^{-1}$ matrix is computed via the following equation, $\mathbf{H}^{-1} = \mathbf{R}^{-1} \times \mathbf{Q}^{\dagger}$, where $\mathbf{R}^{-1}$ is defined in Eq.(8), as shown at the bottom of this page.

2) MATRIX MULTIPLICATION

In the proposed MIMO detection, where $\mathbf{Q}^T$ is a 4×4 matrix and $\mathbf{Z}^j$ is a 4×1 matrix, the sorted transposed unitary matrix $\mathbf{Q}^T$, decomposed from channel matrix $\mathbf{H}^j$, is multiplied by the sorted received signal $\mathbf{Z}^j$. The implementation

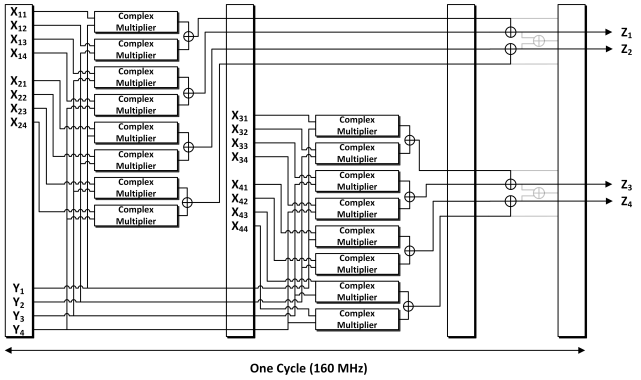

FIGURE 14. Matrix multiplication data path.

of this matrix multiplication requires 16 complex multipliers and 12 complex adders. To shorten the time needed for decoding and fetching instructions, matrix multiplication is designed as an advanced instruction, as it accounts for most of the operations. An equation for the matrix multiplication operation is given by (9). In this example, matrix multiplication requires 16 complex multipliers and 12 complex adders. As it is time-consuming to fetch these instructions, we use an advanced instruction. The equation shows that matrix multiplication is easily parallelized; thus it can be executed in a single 160-MHz cycle. As the I/O port operates at 160 MHz, the input data and computing result is read/written from/to memory in two 160-MHz cycles. Also, the ALU executes in one 160-MHz cycle. In the proposed ALU architecture, the matrix multiplication operation is parallelized. The ALU data path is illustrated in Fig. 14: in this ALU, 12 matrix multiplications can be executed simultaneously.

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X_{21} & X_{22} & X_{23} & X_{24} \\ X_{31} & X_{32} & X_{33} & X_{34} \\ X_{41} & X_{42} & X_{43} & X_{44} \end{bmatrix} \times \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} \tag{9}$$

3) PARTIAL EUCLIDEAN DISTANCE COMPUTING AND SORTING

In addition to the instructions mentioned above, it is also important to compute partial Euclidean distances (PEDs) and to choose the K-best PEDs in the implementation of the proposed detection approach. An advanced instruction is thus added to execute these two operations. Figure 15 illustrates the instructions for decoding different layers. The clustered

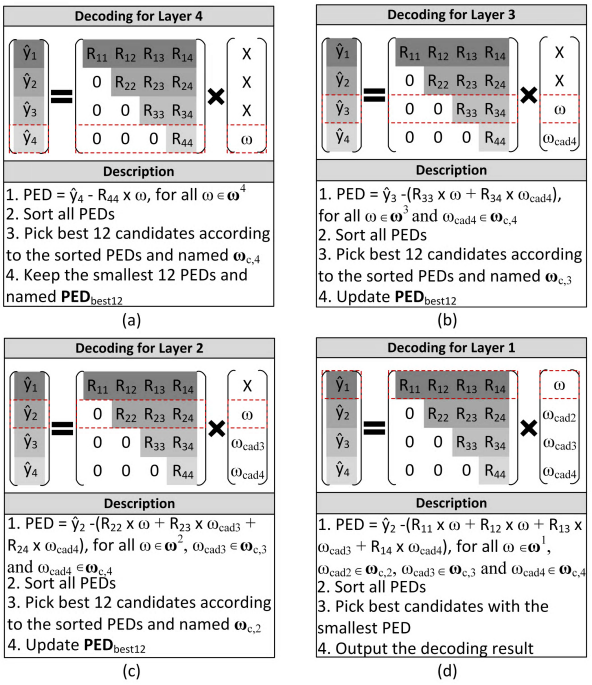

FIGURE 15. Instructions for decoding different layers: (a) Layer 4; (b) Layer 3; (c) Layer 2; (d) Layer 1.

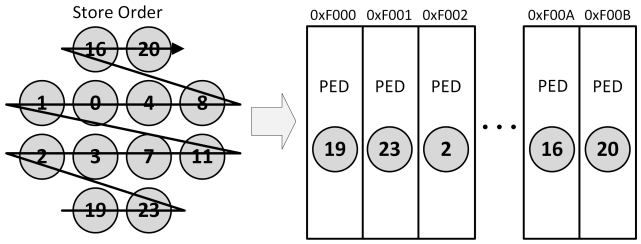

FIGURE 16. Memory mapping of clustered candidates to PED results.

candidates are recorded with *IdxNum*, and the PED results are stored in memory sequentially, as illustrated in Fig. 16. The initial values of this memory space are set to 0xFFFF. To reduce computing complexity, the decoding procedure starts from the 4th layer. The number of constellations in the clustered candidate sets is limited to 16; thus, 16 complex multipliers and 16 complex adders are required to compute the PEDs of the 4th layer. Thus 192 complex multipliers and 192 complex adders are required to compute the PEDs in

$$\mathbf{R}^{-1} = \begin{bmatrix} \frac{1}{r_{11}} & \frac{-r_{12}}{r_{11}r_{22}} & \frac{r_{12}r_{23} - r_{13}r_{22}}{r_{11}r_{22}r_{33}} & \frac{r_{13}r_{22}r_{44} + r_{12}r_{33}r_{24} - r_{12}r_{23}r_{44} - r_{22}r_{33}r_{14}}{r_{11}r_{22}r_{33}r_{44}} \\ 0 & \frac{1}{r_{22}} & \frac{-r_{23}}{r_{22}r_{33}} & \frac{r_{23}r_{34} - r_{33}r_{24}}{r_{22}r_{33}r_{44}} \\ 0 & 0 & \frac{1}{r_{33}} & \frac{-r_{34}}{r_{33}r_{44}} \\ 0 & 0 & 0 & \frac{1}{r_{44}} \end{bmatrix} \tag{8}$$

the 3rd layer, 382 complex multiplications and 382 complex additions in the 2nd layer, and 572 complex multipliers and 572 complex adders in the 1st layer. Therefore, the proposed multi-rate ALU takes 1, 1, 2, and 3 160-MHz cycles to compute the PEDs in the 4th, 3rd, 2nd, and 1st layers, receptively. After computing the PEDs in each layer, they are sorted and the results stored in local memory, 12 constellations with smaller PEDs are chosen as candidates. To accelerate the sorting process, the sorters are divided into 16 subgroups, each of which takes four 320-MHz cycles (or two 160-MHz cycles) to determine the best 12 PED candidates. In order to avoid placing approximate-value PEDs in the same sorting subgroup, candidate PEDs are spread and assigned to sorters uniformly as illustrated in Fig. 17. The address mapping algorithm is described in Algorithm 5. In this implementation, the maximum number of PED candidate combinations is $16 * 12 = 192$, where 12 is the surviving candidates from the last layer and 16 is the maximum number of clustered candidates in the current layer.

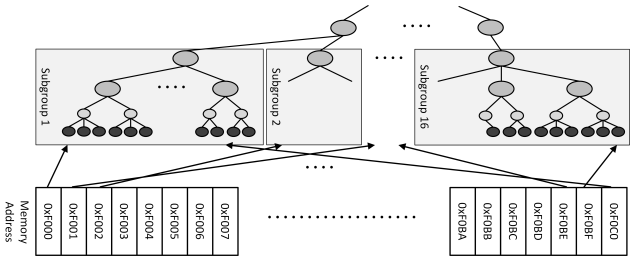

FIGURE 17. PED-to-sorter memory mapping.

The timing diagram of the PED computing and sorting procedure is shown in Fig. 18. To process one data symbol (52 subcarriers), the proposed ALU requires 106 160-MHz cycles to compute and sort PEDs in layers 3 and 4. Also, it requires 107 and 159 160-MHz cycles in layers 2 and 1, respectively.

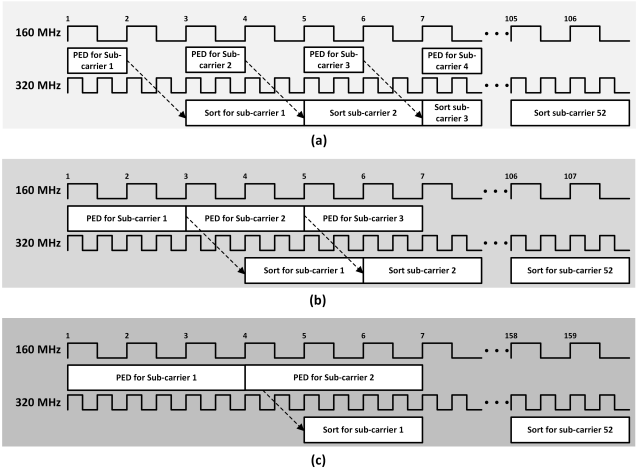

FIGURE 18. Timing diagram of PED computing and sorting instruction: (a) Layers 4 and 3; (b) Layer 2; (c) Layer 1.

Algorithm 5 Sorter Memory Mapping

```
for i = 1 : 192 do
    switch (address mod 16)
    case 0:
        Assign PED to subgroup 0;
    case 1:
        Assign PED to subgroup 8;
    case 2:
        Assign PED to subgroup 1;
    case 3:
        Assign PED to subgroup 9;
    case 4:
        Assign PED to subgroup 2;
    case 5:
        Assign PED to subgroup 10;
    case 6:
        Assign PED to subgroup 3;
    case 7:
        Assign PED to subgroup 11;
    case 8:
        Assign PED to subgroup 4;
    case 9:
        Assign PED to subgroup 12;
    case 10:
        Assign PED to subgroup 5;
    case 11:
        Assign PED to subgroup 13;
    case 12:
        Assign PED to subgroup 6;
    case 13:
        Assign PED to subgroup 14;
    case 14:
        Assign PED to subgroup 7;
    case 15:
        Assign PED to subgroup 15;
    end switch
end for
```

4) CENTRALIZED MEMORY POOL AND LOCAL ALU CACHE

As mentioned above, the centralized memory pool is used to make memory access more efficient. In this work, the pool is combined with latch cells to reduce access latency. After this operation, the channel matrices $\mathbf{H}_i$ are estimated and stored in the pool. In a 4×4 IEEE 802.11n system with 20 MHz bandwidth, there are 52 channel matrices for the data subcarrier; each $\mathbf{H}_i$ contains 16 I/Q data. In this system, both the real and imaginary parts of I/Q data are stored in 16-bit (1 word) format. Therefore, $52 \times 16 \times 2$ words are needed to store the estimated $\mathbf{H}_i$ without pilot positions. The algorithm's pre-processing yields the decomposed matrix unitary matrices $\mathbf{Q}_i$ and up-triangle matrices $\mathbf{R}_i$. Therefore, the centralized memory pool requires $52 \times 16 \times 2 \times 2$ words to store the data matrices of $\mathbf{Q}$ and $\mathbf{R}$. In addition, 52×2 words are required to store subcarriers for each OFDM symbol. The memory

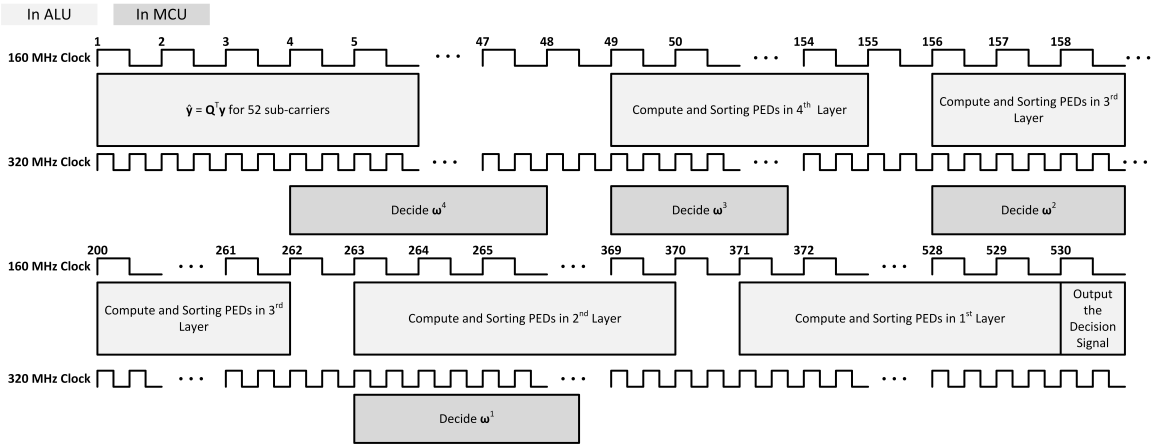

FIGURE 19. Scheduling of proposed MIMO detection.

requirements are summarized in Table 1. The system must provide at least 7280 words to store the channel matrices. Here, to store the information mentioned above and exchange data between the MCU, ALU, and accelerators, the centralized memory pool contains 16384 words of storage space. In addition to the centralized memory pool, a local ALU cache is used to store intermediate data generated during MIMO detection. As the maximum space requirement occurs when storing intermediate data for the computing and sorting of PEDs, the size of the local ALU cache is 9984 words. For throughput enhancement, the operating frequency of the centralized memory pool is 160 MHz. The characteristics of the centralized memory pool and local ALU cache are summarized in Table 2. All of the memory units are composed of 32-bit latches.

TABLE 1. Memory requirements for each subcarrier.

Estimated CFR H_i	16×2 words
Unitary matrix Q_i	16×2 words
Upper-triangle matrix R_i	16×2 words
Estimated signal E_i	4×2 words
$Q_i^T \times y_i$	4×2 words
Candidates	12×2 words
Gain table G_i	4 words
Total	140 words
Operating frequency	160 MHz

C. SCHEDULING

The scheduling of the proposed MIMO detection is illustrated in Fig. 19. The operating frequency of the proposed ALU and MCU is 160 MHz and 320 MHz, respectively. Starting off, the ALU computes $\hat{\mathbf{y}} = \mathbf{Q}^T \times \mathbf{y}$ to later obtain the PED. As the proposed ALU executes at most 12 matrix multiplications in one 160-MHz cycle, it requires 5 cycles to process the matrix multiplication for 52 subcarriers, after which an MCU is used to choose the cluster $\boldsymbol{\omega}^i$ according to the channel gain and the

TABLE 2. Summary of centralized memory pool and local ALU cache.

Specification of centralized memory pool	
Size of memory	16384 words
Minimum memory access unit	1 words/32-bit latch
Operating frequency	160 MHz
Specification of local ALU cache	
Size of memory	9984 words
Minimal memory access unit	1 words/32-bit latch
Operating frequency	160 MHz

estimated signals where i is layer 1, 2, 3, and 4. The MCU requires at most 88 320-MHz cycles (44 160-MHz cycles) for this procedure. In this implementation, to ensure reasonable hardware complexity and computing times, the number of clustered constellations in each $\boldsymbol{\omega}^i$ is kept to at most 16. After determining the clustered constellation set of the 4th layer $\boldsymbol{\omega}^4$, the ALU uses 106 160-MHz cycles to compute and decide the 12 best PED candidates. At the same time, $\boldsymbol{\omega}^4$ is clustered in the MCU. Similarly, the PEDs of layers 3, 2, and 1 are computed in 106, 107, and 159 160-MHz cycles. Finally, the procedure uses at most 530 160-MHz cycles to decide the decoding result of each subcarrier.

Fig. 20 is an overview timing diagram of the scheduling for one signal packet. A short training field (STF) is used to synchronize sample timing and frequency at the beginning of the received packet. A high-throughtput long training field (HT-LTF) is subsequently employed to estimate the channel frequency response and frequency-dependent I/Q imbalance (FDI) [54]. Once the former is estimated, the QR-based matrix inverter computes the $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{H}^{-1}$ for FDI estimation and further MIMO detection. After $\mathbf{H}^{-1}$ is computed, the channel gain and the estimated signals can be output in one 160-MHz cycle. The inverse channel matrix of the two corresponding subcarriers can be output every 26 160-MHz cycles, after which two matrix multiplications, 32 complex multipliers, and 24 complex adders are employed to compute

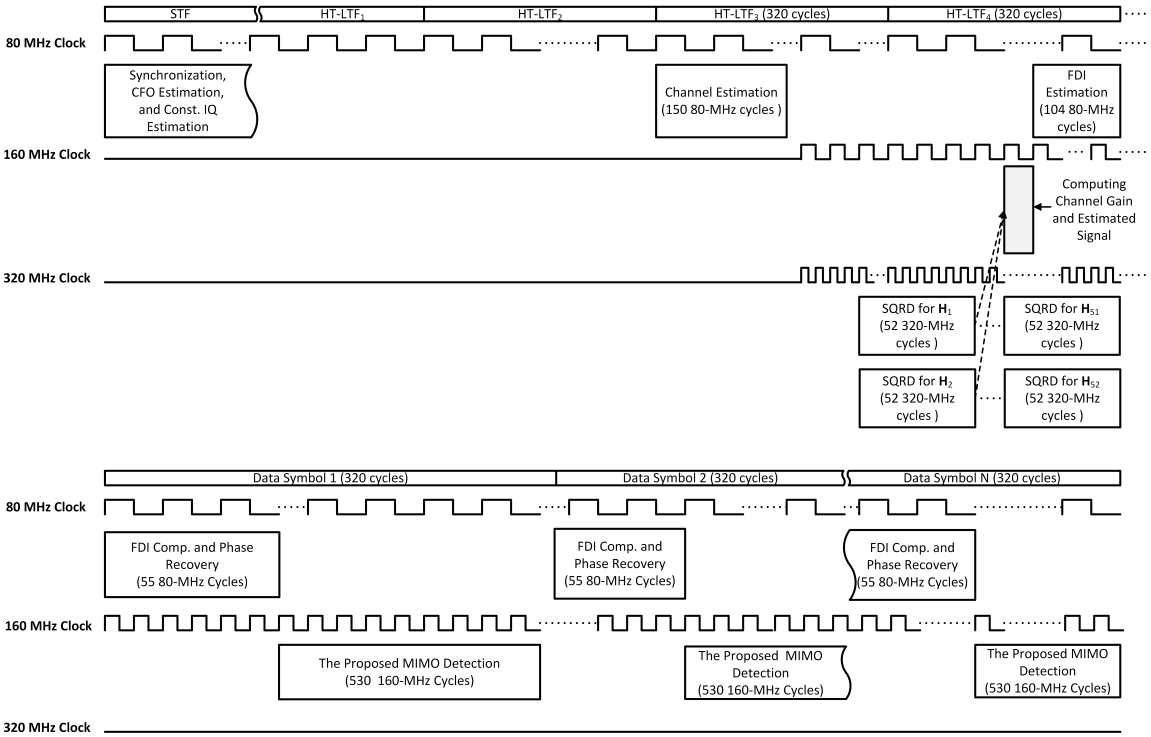

FIGURE 20. Scheduling for processing of single packet.

TABLE 3. Specifications of SIMD-based ALU.

Specification of SIMD-based ALU					
Input ports	10 16-bit complex values				
Output ports	10 16-bit complex values				
Interface freq.	160 MHz				
Operation freq.	80 MHz; 160 MHz; 320 MHz				
Total resources	509+, 812×,24÷				
Instruction resource requirements					
Instruction	Cycles			HW resource	Max. parallel
	80 MHz	160 MHz	320 MHz		
Complex mul.	1			2+,4×	60
Complex div.	3			4+,8×,2÷	12
Matrix mul.		1		56+,64×	12
QR-based inv.			52	QR-based inv.	2
PED computing and sort		4, 5, 6		Max. 572 +,572×, 1 sorter	1

the channel gain and the estimated signal for the corresponding subcarrier. The proposed ALU executes 12 matrix multiplications and at least 60 complex multiplications in one 160-MHz cycles; thus, it can compute the channel gain and estimated signal in one 160-MHz cycle for two subcarriers. Following the HT-LTF symbols, data symbols are received. Due to the 20-MHz specification of the implementation, each data symbol should be processed within 320 80-MHz cycles. At the beginning of every data symbol, 55 cycles (80 MHz) are needed to complete FDI compensation. The proposed MIMO detection procedure is completed in 530 160-MHz cycles (265 80-MHz cycles).

D. IMPLEMENTATION RESULTS

The ALU's hardware operation resources are summarized in the upper part of Table 3. The multi-rate ALU has 10 read ports and 10 write ports for 16-bit complex value access. The operating frequency of the I/O ports is 160 MHz and the operationing frequencies of the multi-rate ALU are 80, 160 MHz, and 320 MHz. The proposed ALU supplies a total of 509 adders, 812 multipliers, and 24 dividers. The multi-rate ALU takes one cycle (160 MHz) each to complete one complex multiplication and one complex division. Matrix multiplication, an advanced instruction, is used to compute $\hat{\mathbf{y}} = \mathbf{Q}^T \times \mathbf{y}$, where $\mathbf{Q}$ is

TABLE 4. Simulation specification.

Parameter	Value
Standard	OFDM w/o channel coding
Signal Bandwidth	40 MHz
Number of Subcarriers	108 subcarriers
FFT Size	128 points
Number of Antennas	4×4; 8×8
packet Size	1024 Bytes per antenna
Delay Spread	15 taps
RMS delay	100 ns
Noise	AWGN
Modulation	64 QAM; 256 QAM
Measurement PER	0.08

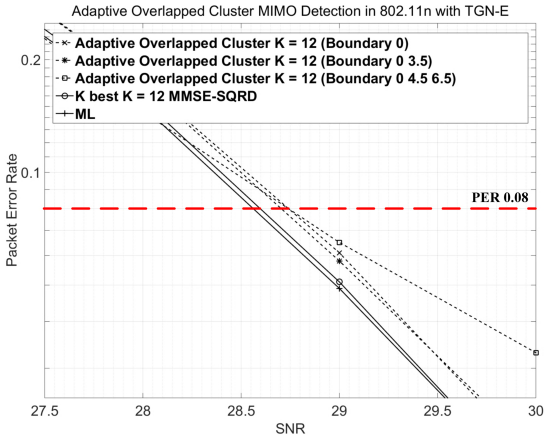

FIGURE 21. Performance in a 4 × 4 64-QAM MIMO-OFDM system.

a 4 × 4 matrix and **y** is a 4 × 1 vector in a 160-MHz cycle. The QR-based inverter takes 52 cycles (320 MHz) to compute the inverse channel matrix and decompose the channel matrix into a unitary matrix **Q** and an upper-triangle matrix **R**. Finally, the PED computing and sorting operation is also implemented as an advanced instruction which takes 4, 4, 5, and 6 160-MHz cycles to compute and sort the PED of the 4th, 3rd, 2nd, and 1st layer, respectively. Hence, the proposed SIMD-based ALU handles each OFDM symbol time.

V. SIMULATIONS AND DISCUSSION

A. ALGORITHM LEVEL

In this section, the required SNR between the standard ML, K-Best SD, and the proposed detection under a packet error rate (PER) of 0.08 and the complexity between K-best SD MIMO detection and the AOC-based MIMO detection are compared at first. And then, the difference between some prior arts and the proposed method are also discussed.

The simulation specifications are provided in Table 4, the proposed MIMO detection is simulated and referred to OFDM system without channel coding. In this work, the multi-path fast-fading frequency-selective channel with 15 taps, 100 ns Root Mean Square (RMS) delay, and AWGN is chosen. The distance between the transceiver and receiver is 20 meters. The standard deviation of shadow fading is 3 dB in line of sight (LOS) and 6 dB in non-line-of-sight NLOS propagation. The signal bandwidth is 40 MHz with 108 subcarriers, whose Fast Fourier Transform (FFT) length is 128. The antenna and modulation scheme are 4 × 4 and 8 × 8 with both 64 QAM and 256 QAM. The K-best parameter, K, and the cluster parameter are tuned to have nearly-performance performance for the purpose of comparison with K-best SD. Since K-best SD is suitable for a practical system, its computing cost is set as 100%. And the boundary condition is used to indicate different partition scheme for pre-partition scheme clusters **C**.

Fig. 21 illustrates the PER of ML-based detection, K-Best SD detection, and the AOC MIMO detection adapted in a 4 × 4 64 QAM MIMO-OFDM system. Table 5 summarizes

TABLE 5. Performance comparison for a 4 × 4 64-QAM system.

Method	*ML*	*K Best*	*This Work*		
	N/A	K = 12	K = 12		
Boundary Condition	N/A	N/A	0	0, 3.5	0, 4.5, 6.5
SNR[a]	28.55	28.6	28.73	28.7	28.73
SNR Loss	0	0.05	0.18	0.15	0.18
Average Case					
Operation reduction	N/A	100%	56.25%	36.96%	35.87%
Multiplicand	N/A	36.8k	20.7k	13.6k	13.2k
Addition	N/A	35k	19.7k	13.31k	12.6k
Worst Case					
Operation reduction	N/A	100%	56.25%	56.25%	56.25%
Multiplicand	N/A	36.8k	20.7k	20.7k	20.7k
Addition	N/A	35k	19.7k	19.7k	19.7k

[a] SNR is measured at PER 0.08.

the performance of Fig. 21, with the performance and complexity comparisons are normalized to ML-based detection. In the performance simulation, SNR degradation between ML-based detection and the proposed algorithm was within 0.57 dB. Compared to K-best SD, the algorithmic complexity can be reduced to 24.50% ∼ 56.25% in the average case and 39.06% ∼ 56.25% in the worst case.

Fig. 22 presents the PER of ML-based MIMO detection, the proposed algorithm, and the K-best SD in 8 × 8 MIMO OFDM with 256 QAM. It is observed that the performance loss between ML and the proposed AOC is within 1.02 dB. The performance and complexity comparisons are presented in Table 6. Compared to K-best SD, the algorithmic complexity can be reduced to 56.25% ∼ 35% in the average case and 56.25% in the worst case.

Tables 7 addresses the comparisons of 4 × 4 and 8 × 8 MIMO detection, i.e., the amount of floating operations (FLOPs), the required SNR at a bit error rate of 10^{-3} and channel type. In this comparison, we convert the measured performance@PER 0.08 into equivalent performance@BER

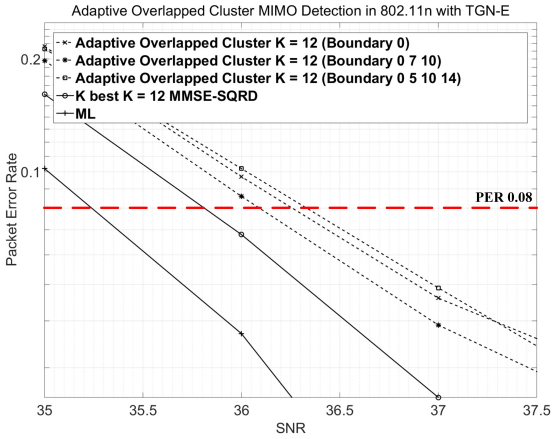

FIGURE 22. Performance in an 8 × 8 256-QAM MIMO-OFDM system.

TABLE 6. Performance comparison for an 8×8 256-QAM system.

Method	*ML*	*K Best*	*This Work*		
	N/A	K = 12	K = 12		
Boundary Condition	N/A	N/A	0	0, 7, 10	0, 5, 10, 14
SNR[a]	35.23	35.81	36.25	36.1	36.33
SNR Loss	0	0.58	1.02	0.87	1.1
Average Case					
Operation reduction	N/A	100%	56.25%	39.94%	35%
Multiplicand	N/A	294.9k	165.9k	117.8k	102.4k
Addition	N/A	289.5k	162.9k	115.7k	100.5k
Worst Case					
Operation reduction	N/A	100%	56.25%	56.25%	56.25%
Multiplicand	N/A	36.8k	165.9k	165.9k	165.9k
Addition	N/A	35k	162.9k	162.9k	162.9k

[a] SNR is measured at PER 0.08.

TABLE 7. Comparisons of 4×4 and 8×8 MIMO detection.

4×4 MIMO Detection				
	Ref. [10]	Ref. [19]	Ref. [39]	This work
Modulation	16 QAM	64 QAM	64 QAM	64 QAM
Detector	KSE	Slicing List	LRA-MMSE FSD	AOC
K value	5	16	8	12
Delay Spread	N/A	N/A	N/A	15 taps
RMS Delay	N/A	N/A	N/A	100 ns
SNR@BER 10^{-3}	29.8dB[a]	29.78 dB	26.78 dB	26.3 dB[b]
# of Real Multiplication	5040	N/A	N/A	52.8k
FLOPs	N/A	N/A	7937	71k[c]
Search Tree Spanning (Each Layer)	N/A	16^2	N/A	12^2
8×8 MIMO Detection				
	Ref. [7]	Ref. [19]	Ref. [37]	This work
Modulation	256 QAM	64 QAM	64 QAM	256 QAM
Detector	LRA FCTS	Slicing List	fcELLL	AOC
K value	N/A	16	N/A	12
Delay Spread	N/A	N/A	N/A	15 taps
RMS Delay	N/A	N/A	N/A	100 ns
SNR@BER 10^{-3}	38.3 dB	28.28 dB	34.5 dB	33.5 dB
# of Real Multiplication	N/A	N/A	N/A	471.2k
FLOPs for Matrix Reordering	N/A	N/A	275	120
FLOPs	850k	N/A	N/A	815k
Search space size (Each Layer)	N/A	16^2	N/A	12^2

[a] $(S/N)_{dB} = 10 \times \log(E_b/N_0 \times f)$, where $f = R_c \times S_b$ and R_c is the coding rate and S_b is the bit per Symbol.
[b] PER 0.6 is equivalent to BER 10^{-3} by $PER = 1-(1-BER)^{\#bits}$, where $\#bits$ refer to number of bits.
[c] FLOPs are counted 2 for complex addition or subtraction and 6 for a complex multiplication [37].

10^{-3} via eq. (10).

$$PER = 1 - (1 - BER)^{Bit\_Length\_of\_Packet} \tag{10}$$

Compared to 4 × 4 MIMO detections, the proposed algorithm takes 71k FLOPs to maintain SNR performance with 64 QAM in frequency-selective fading channel (with 100 ns RMS delay and 15 taps delay spread). If this AOC-based MIMO detection is worked in AWGN only, the required FLOPs, which is more than LRA-MMSE FSD [39] (7937), is around 3.97k in 64-QAM. Compared to KSE method [10], the AOC algorithm takes about 16k if 9 constellations are chosen in each layer with 16 QAM and K is set to 12. Finally, compared to a slicing list detector [19] reduce the number of visited node in the detection tree via mapping the points to decision lists, the proposed detection spans 43.75% fewer search tree between layers and has 3.48 dB enhancement. In the 8 × 8 systems, the proposed detection consumes 815k FLOPs for maintaining SNR performance with 256 QAM. Also, the channel condition is the same as 4 × 4 simulation. Compared to fcELLL detection [37], the proposed algorithm take fewer FLOPs for matrix reordering because we sort the channel according to the channel gain straightly. Then, compared to the slicing list detection [19], the tree spanning of the proposed detection is also 43.75% fewer than the slicing list detection [19]. Finally, compared to the MIMO detection [7] which is combined with LR-based processing, the proposed AOC-based MIMO detection balances computing efforts and required SNR in frequency-selective fading with an acceptable computational effort.

B. IMPLEMENTATION LEVEL

The implementation features are listed in Table 8. The revised reconfigurable modem is realized by HDL and synthesized by the Synopsys Design Compiler via TSMC's 45-nm technology. The hardware cost of the proposed algorithm is estimated under the resource utilization in the revised reconfigurable modem because the modem does not support only the MIMO detection function. Table 9 presents the comparisons of this

TABLE 8. Implement summarize of proposed AOC MIMO detection.

Implementation Level	
Antenna Config.	4×4
Modulation	64 QAM
Technology	45 nm
Gate Count	454.8 KGE
Clock Freq.	80 MHz, 160 MHz, 320 MHz
Throughput	1077.8 Mbps

TABLE 9. Comparison with related 64-QAM MIMO detectors.

	[43]	[44]	[45]	This Work		
Arch.	ASIC		rASIP	Reconfigurable Modem		
Antenna Config.	16×16	4×4	4×4[d]	4×4		
Algo.	CHOSLAR	Var. K-Best	LMMSE	AOC		
Tech. (nm)	65	90	65	45		
Gate Count (KGE)	5681	182	482	454.8		
Clock Freq. (MHz)	588	200	400	80[a]	160[b]	320[c]
Throughput (Mbps)[e]	5096	2400	866.67	1077.8		

[a] The clock rate of normal operation.
[b] The clock rate of memory accessing and processing the proposed MIMO detection.
[c] The clock rate of QR-based Matrix Inverter.
[d] The available antenna configurations are 1×2, 1×4, 2×2, 2×4, 4×4, and 4×8.
[e] These throughputs are normalized to 45-nm technology via Eq. (11).

work and other related MIMO detectors whose comparison indexes are antenna configuration, gate counts, clock frequency, and throughput. In this comparison, we estimate the equivalent hardware cost of the proposed MIMO detector according to the computing resources in the reconfigurable modem, which is used to accomplish the AOC detection function. It takes 454.8 kGE (kilo-gates equivalent) to implement the proposed MIMO detection. The two QR-based inverters consume 51.8% of the hardware cost. Excluding the cost of two QR-based inverters, the proposed MIMO detection requires 235.8 kGE to realize its function. The proposed detector throughput is 1077.8 Mbps. Further, the throughputs of the references are normalized via Eq. 11.

$$Througput_{norm} = Througput \times \frac{Tech.}{45nm} \tag{11}$$

The ASIC architecture, a 16×16 Cholesky sorted QR decomposition and partial iterative lattice reduction (CHOSLAR) detector [43] and a 4×4 variable K-best detector [44], which are realized respectively via the 65-nm and 90-nm technology, are reported. The 16×16 MIMO detector adopted Cholesky decomposition to perform QR decomposition. Compared to the QR-based inverter which is adopted in reconfigurable modem, the Cholesky-based QR decomposition performs better parallel ability. Thus, a MIMO detector with higher antenna number is implemented. If the proposed algorithm is designed in a 16×16 detector, it requires more than 2000 kGE to implement the AOC MIMO detection. Because the gate counts of QR-based Inverter is grown at least 10 times. On the other hand, a variable K-best MIMO detector [44], whose normalized throughput is 2400 Mbps, is implemented via 2-dimensioned paralleled sorter. Compared to this detector, the design purposed of the sorter in this work is to shorten the latency via combining the memory mapping. However, it takes additional operators to compute the PEDs of each candidate. A reconfigurable application-specific instruction set processor (rASIP) LMMSE detector [45], which is constructed by coarse-grained reconfigurable architecture that supports different matrix operations [45], is compared. The detector supports up to an 8×8 antenna configuration, which hints that the rASIP architecture is flexible. Compared to the rASIP MIMO detection, the hardware cost and throughput of the proposed reconfigure modern are acceptable.

VI. CONCLUSION

The proposed adaptive overlapped cluster (AOC)-based MIMO detection not only maximizes the hit probability of the chosen cluster but also minimizes the number of clustered constellations by combining several overlapped clusters, with the size determined by the channel status. This was ported to a frequency-domain (FD) MIMO-OFDM modem that supports 4×4 WLAN with a 20-MHz bandwidth. The modem includes a revised centralized memory pool, multi-rate ALU, and MCU. We extended PEs slightly in the multi-rate ALU to improve operation utilization with acceptable clock rates. The centralized memory pool prevents data copy and reduced access latencies. There is also a MCU for scheduling and controlling. In the implementation, the AOC scheme costs 454.8 kGE to yield a throughput of 1077.8 Mbps, which is sufficient to support the target specification. On the algorithm level, the simulations shows that the computing complexity of 4×4 64-QAM MIMO OFDM and 8×8 256-QAM MIMO OFDM saves 43.75% (100%–56.25%) in the worst case. Compared to ML detection, the proposed detection yields dB reductions of 0.57 in 4 64-QAM MIMO OFDM and 1.02 in 8×8 256-QAM MIMO OFDM.

REFERENCES

[1] T.-D. Chiueh and P.-Y. Tsai, *OFDM Baseband Receiver Design for Wireless Communications*. Hoboken, NJ, USA: Wiley, 2007.
[2] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, *MIMO Wireless Communications*. New York, NY, USA: Cambridge Univ. Press, 2007.
[3] X. Zhu and R. D. Murch, "Performance analysis of maximum likelihood detection in a MIMO antenna system," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 187–191, Feb. 2002.
[4] Y. L. C. de Jong and T. J. Willink, "Iterative tree search detection for MIMO wireless systems," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 930–935, Jun. 2005.
[5] K. Neshatpour, M. Shabany, and G. Gulak, "A high-throughput VLSI architecture for hard and soft SC-FDMA MIMO detectors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 3, pp. 761–770, Mar. 2015.
[6] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inf. Theory*, vol. 45, no. 5, pp. 1639–1642, Jul. 1999.
[7] H. Kim, J. Park, H. Lee, and J. Kim, "Near-ML MIMO detection algorithm with LR-aided fixed-complexity tree searching," *IEEE Commun. Lett.*, vol. 18, no. 12, pp. 2221–2224, Dec. 2014.

[8] M. Kim and J. Kim, "Applications of SDR exact-ML criterion to tree-searching detection for MIMO systems," in *Proc. 8th Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2014, pp. 1–7.
[9] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.
[10] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-Best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.
[11] X. Chen, G. He, and J. Ma, "VLSI implementation of a high-throughput iterative fixed-complexity sphere decoder," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 60, no. 5, pp. 272–276, May 2013.
[12] M. Mahdavi and M. Shabany, "Novel MIMO detection algorithm for high-order constellations in the complex domain," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 5, pp. 834–847, May 2013.
[13] M. Neinavaie and M. Derakhtian, "ML performance achieving algorithm with the zero-forcing complexity at high SNR regime," *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4651–4659, Jul. 2016.
[14] 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.
[15] 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.
[16] A. Tomasoni, M. Siti, M. Ferrari, and S. Bellini, "Hardware oriented, quasi-optimal detectors for iterative and non-iterative MIMO receivers," *EURASIP J. Wireless Commun. Netw.*, vol. 2012, no. 1, p. 62, Feb. 2012. doi: 10.1186/1687-1499-2012-62.
[17] H. Sarieddeen, M. M. Mansour, L. M. A. Jalloul, and A. Chehab, "Low-complexity MIMO detector with 1024-QAM," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2015, pp. 883–887.
[18] J. Y. Chu, Y. T. Liao, and T. Y. Hsu, "Low complexity MIMO detection by using overlapped cluster search," in *Proc. IEEE 5th Int. Conf. Consum. Electron. Berlin (ICCE-Berlin)*, Sep. 2015, pp. 401–404.
[19] S. Suh and J. R. Barry, "Reduced-complexity MIMO detection via a slicing breadth-first tree search," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1782–1790, Mar. 2017.
[20] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41–59, Feb. 1996.
[21] D. Wubben, R. Bohnke, V. Kuhn, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE Veh. Technol. Conf. (VTC-Fall)*, vol. 1, Oct. 2003, pp. 508–512.
[22] Y. Lee, H. C. Shih, C. S. Huang, and J. Y. Li, "Low-complexity MIMO detection: A mixture of ZF, ML and SIC," in *Proc. 19th Int. Conf. Digit. Signal Process.*, Aug. 2014, pp. 263–268.
[23] M. Mandloi and V. Bhatia, "Ordered iterative successive interference cancellation algorithm for large MIMO detection," in *Proc. IEEE Int. Conf. Signal Process., Inf., Commun. Energy Syst. (SPICES)*, Feb. 2015, pp. 1–5.
[24] M. Mandloi, M. A. Hussain, and V. Bhatia, "An improved multiple feedback successive interference cancellation algorithm for MIMO detection," in *Proc. 8th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2016, pp. 1–6.
[25] D.-C. Chang and D.-L. Guo, "Spatial-division multiplexing MIMO detection based on a modified layered OSIC scheme," *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, pp. 4258–4271, Sep. 2013.
[26] M. Mandloi, M. A. Hussain, and V. Bhatia, "Improved multiple feedback successive interference cancellation algorithms for near-optimal MIMO detection," *IET Commun.*, vol. 11, no. 1, pp. 150–159, 2017.
[27] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 515–534, Dec. 1982.
[28] Q. Wen and X. Ma, "An efficient Greedy LLL algorithm for MIMO detection," in *Proc. IEEE Military Commun. Conf.*, Oct. 2014, pp. 550–555.
[29] 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.
[30] Q. Zhou and X. Ma, "Element-based lattice reduction algorithms for large MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 274–286, Feb. 2013.
[31] U. Ahmad *et al.*, "Exploration of lattice reduction aided soft-output MIMO detection on a DLP/ILP baseband processor," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5878–5892, Dec. 2013.
[32] K. Zhao and S. Du, "Full-diversity approximated lattice reduction algorithm for low-complexity MIMO detection," *IEEE Commun. Lett.*, vol. 18, no. 6, pp. 1079–1082, Jun. 2014.
[33] S. Shahabuddin, J. Janhunen, Z. Khan, M. Juntti, and A. Ghazi, "A customized lattice reduction multiprocessor for MIMO detection," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 2976–2979.
[34] D. Mitsunaga, T. T. Y. Khine, and H. A. Zhao, "A low complexity ZF-based lattice reduction detection using curtailment parameter in MIMO systems," in *Proc. IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2016, pp. 1–5.
[35] S. Liu, Y. Yu, and H. Lv, "Hybrid pre-judging fix-LLL lattice reduction algorithm for MIMO detection," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng. (ICISCE)*, Jul. 2016, pp. 1337–1341.
[36] J. Liu, S. Xing, and L. Shen, "Lattice-reduction-aided sphere decoding for MIMO detection achieving ML performance," *IEEE Commun. Lett.*, vol. 20, no. 1, pp. 125–128, Jan. 2016.
[37] Q. Wen and X. Ma, "Fixed-complexity variants of the effective LLL algorithm with greedy convergence for MIMO detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 3826–3830.
[38] M. Shabany, A. Youssef, and G. Gulak, "High-throughput 0.13-μm CMOS lattice reduction core supporting 880 Mb/s detection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 5, pp. 848–861, May 2013.
[39] H. Kim, H. Lee, and J. Kim, "MMSE-based lattice-reduction-aided fixed-complexity sphere decoder for low-complexity near-ML MIMO detection," in *Proc. 21st IEEE Int. Workshop Local Metrop. Area Netw.*, Apr. 2015, pp. 1–6.
[40] J.-Y. Lin, J.-C. Chi, C.-F. Liao, and Y.-H. Huang, "A 6.4 g LLR/s 8×8 64-QAM soft-output MIMO detector with lattice reduction preprocessing," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2017, pp. 1–4.
[41] M. Shabany, R. Doostnejad, M. Mahdavi, and P. G. Gulak, "A 38 pj/b optimal soft-MIMO detector," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 9, pp. 1062–1066, Sep. 2017.
[42] I. A. Bello, B. Halak, M. El-Hajjar, and M. Zwolinski, "VLSI implementation of a scalable K-best MIMO detector," in *Proc. 15th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Oct. 2015, pp. 281–286.
[43] G. Peng, L. Liu, S. Zhou, Y. Xue, S. Yin, and S. Wei, "Algorithm and architecture of a low-complexity and high-parallelism preprocessing-based k-best detector for large-scale MIMO systems," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1860–1875, Apr. 2018.
[44] W. Fan and A. Alimohammad, "Two-dimensional sorting algorithm for high-throughput K-best MIMO detection," *IET Commun.*, vol. 11, no. 6, pp. 817–822, 2017.
[45] X. Chen, A. Minwegen, S. B. Hussain, A. Chattopadhyay, G. Ascheid, and R. Leupers, "Flexible, efficient multimode MIMO detection by using reconfigurable ASIP," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 10, pp. 2173–2186, Oct. 2015.
[46] S. Yang, T. Lv, R. G. Maunder, and L. Hanzo, "From nominal to true *a posteriori* probabilities: An exact Bayesian theorem based probabilistic data association approach for iterative MIMO detection and decoding," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2782–2793, Jul. 2013.
[47] C.-C. Cheng, S. Sezginer, H. Sari, and Y. T. Su, "Robust MIMO detection under imperfect CSI based on Bayesian model selection," *IEEE Wireless Commun. Lett.*, vol. 2, no. 4, pp. 375–378, Aug. 2013.
[48] R. Malladi, K. Kuchi, and R. D. Koilpillai, "Set-partitioning based forward/backward soft decision algorithms for MIMO detection," in *Proc. Int. Conf. Signal Process. Commun. (SPCOM)*, Jul. 2012, pp. 1–5.
[49] 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.
[50] A. Elghariani and M. Zoltowski, "Branch and bound with M algorithm for near optimal MIMO detection with higher order QAM constellation," in *Proc. Mil. Commun. Conf. (MILCOM)*, Oct. 2012, pp. 1–5.
[51] M. M. Mansour, "A near-ML MIMO subspace detection algorithm," *IEEE Signal Process. Lett.*, vol. 22, no. 4, pp. 408–412, Apr. 2015.
[52] S. Yoon and C.-B. Chae, "Low-complexity MIMO detection based on belief propagation over pairwise graphs," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2363–2377, Jun. 2014.
[53] B. Y. Kong and I.-C. Park, "Efficient tree-traversal strategy for soft-output MIMO detection based on candidate-set reorganization," *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1758–1761, Sep. 2013.

[54] S. Y. Yeh, Y. T. Liao, W. C. Lai, and T. Y. Hsu, "Cost-efficient frequency-domain MIMO–OFDM modem with an SIMD ALU-based architecture," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 12, pp. 2791–2803, Dec. 2015.
[55] L. Bai and J. Choi, *Low Complexity MIMO Detection*. New York, NY, USA: Springer, 2012.
[56] W. C. Lai, Y. T. Liao, and T. Y. Hsu, "A cost-effective preamble-assisted engine with skew calibrator for frequency-dependent I/Q imbalance in 4×4 MIMO-OFDM modem," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2199–2212, Aug. 2013.
[57] P. Salmela, A. Burian, H. Sorokin, and J. Takala, "Complex-valued QR decomposition implementation for MIMO receivers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2008, pp. 1433–1436.

YUAN-TE LIAO was born in Taipei, Taiwan, in 1985. He received the B.S. degree in mathematics from National Tsing Hua University, Hsinchu, Taiwan, in 2008. He is currently pursuing the Ph.D. degree with the Department of Computer Science, National Chiao Tung University, Hsinchu. His current research interests include MIMO-OFDM systems, system-on-a-chip design technology, and related VLSI architecture.

TERNG-YIN HSU received the Ph.D. degree from the Institute of Electronics, National Chiao Tung University, Hsinchu, Taiwan, in 1999, where he joined the Department of Computer Science, in 2003, and is currently a Professor. He was a co-founder and the CTO of amoesolu cooperation, from 2013 to 2014. His current research interests include GPU-based soft RAN, radio function split and virtualization, baseband algorithms, multi-spec transmissions, VLSI architectures, analog-like digital circuits, system-on-a-chip design technology, and related application-specified integrated circuits (ASIC) designs.