A 1.58 Gbps/W 0.40 Gbps/mm^2 ASIC Implementation of MMSE Detection for 128 × 8 64-QAM Massive MIMO in 65 nm CMOS

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

***Abstract*— The minimum-mean-square error (MMSE) plays a significant role in the signal detection process of massive multiple-input-multiple-output (MIMO) systems. Matrix inversion, which is the major part of calculating the MMSE, suffers from high computing loads and low parallelism, especially in massive MIMO systems; as such, hardware implementation is difficult. This paper proposes a user-level parallelism-based fully pipelined very large-scale integration (VLSI) architecture of an MMSE detector for an uplink 128 × 8 64-QAM massive MIMO system. First, a diagonal-based systolic array with single-sided input is adopted; this array eliminates the throughput limitation. Second, a weighted Jacobi-iteration-based architecture is proposed to iteratively achieve matrix inversion, thereby reducing the computational load and exploiting the potential parallelism of the matrix inversion. Third, an approximated architecture is proposed to compute the log-likelihood ratio. This architecture is verified on an FPGA and fabricated onto a 2.57 mm^2 silicon with TSMC 65 nm CMOS technology, thereby achieving a 1.02 Gbps data rate at 680 MHz while dissipating 646 mW. The results indicate an energy efficiency of 1.58 Gbps/W and an area efficiency of 0.40 Gbps/mm^2, which are 2.93× and 2.86× that of state-of-the-art similar designs with CMOS technology, respectively.**

***Index Terms*— Massive MIMO, MMSE, soft-output detector, weighted Jacobi iteration, VLSI.**

I. Introduction

MASSIVE multiple-input multiple-output (MIMO) is currently one of the most promising technologies for future wireless communications systems such as 5G [1], [2]. However, a critical limitation of massive MIMO is signal detection in the uplink. Specifically, it is difficult to achieve an effective compromise amongst low computing load, high processing parallelism and high detection accuracy [1], [2]. A maximum likelihood detection technique [3] has been proposed as the optimal detection algorithm. However, its computing load exponentially increases with the number of users, which makes it impractical in massive MIMO systems. Various non-linear detection algorithms, such as sphere-decoding [4] and the K-Best detection algorithm [5], have also been proposed. Although these non-linear detection algorithms can achieve high detection accuracy, their computing loads remain unacceptable when the number of antennas at the base station (BS) is large (e.g., 128 or 256 antennas). The complicated signal detection process is difficult to efficiently implement in an actual massive MIMO system. To reduce the computing load, various linear detection algorithms have been proposed [6]–[8]. These algorithms have lower detection accuracy compared with non-linear algorithms and can achieve lower computing loads. Among these linear algorithms, the minimum mean square error (MMSE) is one of the most effective algorithms in reducing computing loads, having minimal detection accuracy loss; as such, it has significant potential for use in practical massive MIMO systems [1], [2].

However, MMSE detection faces the significant challenge of intensive matrix inversion in practical massive MIMO systems [9]. The computing load of matrix inversion increases as $\mathcal{O}(M^3)$ (where M is the number of users), which produces difficulties for hardware implementation with increasing numbers of users. This issue limits the application of hardware architectures (detectors) in massive MIMO systems [9], [10]. Various works have been proposed to reduce computing loads and optimize hardware architectures, including Neumann series approximation (NSA)-based detectors [9], [10] and Cholesky decomposition (CHD)-based detectors [11], [12]. These methods achieve high throughput but consume a significant amount of hardware resources. To reduce hardware resource consumption, architectures based on approximation methods have also been proposed such as the Gauss-Seidel (GS) [13], [14], Richardson (RI) [15], successive over-relaxation (SOR) [16], [17] and successive over-relaxation (SSOR) [18] methods. However, the computations in the GS, RI, SOR and SSOR methods are difficult to parallelize due to the high correlation between each compute step. To explore the parallelism between each step, implicit methods have been proposed, including conjugate gradient (CG) [19], optimized coordinate descent (OCD) [20] and intra-iterative interference cancellation (IIC) [21]. However, these implicit methods do not consider the unique properties of massive MIMO systems, such as channel hardening; therefore, the preprocessing results (e.g., Gram matrices) cannot be reused. Here, the same

Manuscript received June 23, 2017; revised August 24, 2017; accepted September 15, 2017. Date of publication September 26, 2017; date of current version April 2, 2018. This paper was recommended by Associate Editor G. Masera. *(Corresponding author: Leibo Liu.)*

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

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

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

Digital Object Identifier 10.1109/TCSI.2017.2754282

Gram matrix must be computed many times, which means that architectures based on implicit methods suffer from high energy consumption and latency. A detailed analysis of related works is presented in Section II.

To address the problems described above, this paper proposes a VLSI architecture for signal detection in an uplink massive MIMO system. Based on user-level parallelism, a fully pipelined technique is adopted to achieve higher energy and area efficiencies (i.e., throughput/power and throughput/area), where each processing element of the estimated vectors is computed per clock cycle in parallel. First, given that an MMSE filtering matrix is diagonally dominant for an uplink massive MIMO system, a diagonal-based systolic array with single-sided input is designed. This systolic array substantially reduces latency using limited hardware, and the architecture operates in a deeply pipelined manner. Second, per the weighted Jacobi iteration (WeJi) method, an architecture is proposed to scale down the high computing load of the exact matrix inversion. In addition, a dominant diagonal-component approximation is employed to generate an initial solution of the iteration. Third, an architecture is designed to approximately compute the log-likelihood ratio (LLR) by applying the approximated LLR processing method to the WeJi method. The proposed VLSI architecture is verified on an FPGA and fabricated into a chip with 65 nm technology in an uplink massive MIMO system. The measured results demonstrate that the proposed architecture possesses advantages with respect to energy efficiency (throughput/power) and area efficiency (throughput/area) compared with other state-of-the-art designs. Considering detection accuracy, this chip reduces the signal-to-noise ratio (SNR) by 0.2 dB (measured at a target FER of 10^{-2}), consisting of a 0.11 dB loss from the approximation algorithm itself and a 0.09 dB loss from the fixed-point error (i.e., the truncation error results from the limited word-length of the hardware, approximate reciprocal unit, and look-up table (LUT)). This detection accuracy loss is already acceptable for an actual massive MIMO system and is even lower than that of other state-of-the-art designs [9]–[21].

Notation: Boldface capital letters and lowercase letters stand for matrices and vectors, respectively; $(\cdot)^T$, $(\cdot)^H$, $(\cdot)^{-1}$, $\rho(\cdot)$ and $det(\cdot)$ denote transpose, conjugate transpose, inversion, spectral radius and determinant, respectively; $\mathbb{E}(\cdot)$ denotes the expectation; $(\cdot)^*$ denotes the conjugate operator; $\mathbf{I}_M$ stands for the $M \times M$ identity matrix; $||\cdot||_2$ and $||\cdot||_F$ stand for the l_2-norm and Frobenius norm of a matrix, respectively; and lowercase k and uppercase K are the current iteration and total number of iterations, respectively.

Outline: The remainder of this paper is organized as follows. Section II briefly introduces the system model and related work. Section III details the proposed massive MIMO detector. Section IV provides frame-error-rate (FER) simulations and comparisons of detection algorithms. Section V shows the silicon implementation results and comparisons. Section VI provides the conclusion.

II. System Model and Related Work

A. System Model

A massive MIMO systems has N antennas at the BS to simultaneously communicate with M single-antenna users, predominantly $N \gg M$ [1]. The parallel transmit bit-streams of M users are encoded by utilizing channel encoders; the results are then mapped to constellation symbols to obtain a sequence of transmit vectors ($\mathbf{s}$) by taking symbols from a set of a constellation alphabet, $\mathcal{Q}$. Let $\mathbf{s} \in \mathcal{Q}$ denote the $M \times 1$ transmitted signal vector of all M users, where the vector $\mathbf{y}$ denotes the $N \times 1$ received signals at the BS. Now, the baseband input-output relationship can be described by

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

where $\mathbf{H} \in \mathbb{C}^{N \times M}$ stands for the flat Rayleigh fading channel matrix, whose elements are independent and identically distributed (i.i.d.) following $\mathbb{N}(0, 1)$, and all elements of $\mathbf{n}$ denote $N \times 1$ i.i.d. zero-mean complex additive white Gaussian noise.

The transmitting signal ($\mathbf{s}$) by MMSE can be calculated as

$$\hat{\mathbf{s}} = \left(\mathbf{H}^H\mathbf{H} + N_0 {E_s}^{-1}\mathbf{I}_M\right)^{-1}\mathbf{H}^H\mathbf{y} = \mathbf{A}^{-1}\mathbf{y}^{\mathrm{MF}}, \tag{2}$$

where $\mathbf{A} = \mathbf{H}^H\mathbf{H} + N_0 {E_s}^{-1}\mathbf{I}_M$ is the MMSE filtering matrix and $\mathbf{y}^{\mathrm{MF}} = \mathbf{H}^H\mathbf{y}$ is the matched-filter vector. In addition, N_0 and E_s denote the power spectral density of the noise and the power of the transmitted vector, respectively. Per the definition of the matched filter $\mathbf{y}^{\mathrm{MF}}$, the vector $\mathbf{s}$ can be described by

$$\hat{\mathbf{s}} = \mathbf{U}\mathbf{s} + \mathbf{v}, \tag{3}$$

where $\mathbf{U} = \mathbf{A}^{-1}\mathbf{H}^H\mathbf{H}$ and $\mathbf{v} = \mathbf{A}^{-1}\mathbf{H}^H\mathbf{n}$ are the equivalent channel matrices. Let σ_{eq}^2 denote the variance of the post-equalization noise plus interference (NPI); here, b is the bit index of the LLR of the i-th user. The max-log LLR satisfies

$$L_{i,b} = \frac{U_{ii}^2}{\sigma_{\mathrm{eq}}^2}\left(\min_{s \in S_b^0}\left|\frac{\hat{s}_i}{U_{ii}} - s\right|^2 - \min_{s \in S_b^1}\left|\frac{\hat{s}_i}{U_{ii}} - s\right|^2\right) = \zeta_i^2 \varphi_b(\hat{s}_i), \tag{4}$$

where ζ_i^2 is the signal-to-interference-plus-noise ratio (SINR) for the i-th user, $\varphi_b(\hat{s}_i)$ is a piecewise linear function for Gray mappings, and S_b^0 and S_b^1 denote the sets of modulation constellation symbols, where the i-th bit is 0 and 1, respectively.

In massive MIMO systems, MMSE detection can achieve near-optimal performance [1]. However, the computing loads of the matrix inversion $\mathbf{A}^{-1}$ in (3) and the LLR in (4) are high ($\mathcal{O}(M^3)$), especially in systems with a large number of antennas [9]. Furthermore, considering hardware implementation, the computing of the matrices $\mathbf{A}$ and $\mathbf{A}^{-1}$ restricts the system parallelism due to the high correlation of each computation.

B. Related Work

To achieve complicated matrix inversions in MMSE, three types of architectures have been proposed that can approximate or even accurately implement matrix inversion. These methods include the following: the exact matrix inversion method (CHD [11], [12]), explicit approximation methods (NSA [9], [10], GS [13], [14], SOR [16], [17] and SSOR [18]) and implicit approximation methods (CG [19], OCD [20], and IIC [21]).

In [11], [12], VLSI architectures were proposed based on the CHD method, which decompose the matrix $\mathbf{A}$ into two parts:

a diagonal matrix $\mathbf{P}$ and an off-diagonal matrix $\mathbf{Q}$. The exact matrix inversion in MMSE detection is achieved as

$$\hat{\mathbf{A}}^{-1} = \mathbf{L}^{-1}\mathbf{P}^{-1}\mathbf{L}^{-1}, \tag{5}$$

where $\mathbf{P}$ is a diagonal matrix and $\mathbf{L}$ is a lower unit triangular matrix ($\mathbf{Q} = \mathbf{L} + \mathbf{L}^{-1}$). In this architecture, there are three important computing blocks: triangular multiplication systolic arrays, substitution blocks and accumulation blocks. This architecture achieves precise matrix inversion, but certain deficiencies, such as the low parallelism between each computation step and the high computing load of the matrix inversion ($\mathcal{O}(U^3)$), remain. Hence, this architecture is unsuitable for massive MIMO systems because of the strict hardware requirements [9].

Explicit method (NSA) architectures have been proposed to achieve high throughput for massive MIMO detection [9], [10]. The NSA method rewrites the inversion of the MMSE filtering matrix, $\mathbf{A}$, with the following expression:

$$\hat{\mathbf{A}}_k^{-1} = \sum\nolimits_{n=0}^{k-1} \left(-\mathbf{P}^{-1}\mathbf{Q}\right)^n \mathbf{P}^{-1} \tag{6}$$

where k is the number of iterations. When k is not large, the computing load is low; however, the approximate error of the matrix inversion is unacceptable. When k is large, the computing load increases as $\mathcal{O}(M^3)$. Furthermore, there are numerous matrix multiplications. Hence, this method can partially reduce the number of multiplications . Although the number of real multiplications is partially scaled down in this algorithm, undesired large-scale matrix multiplications and large detection accuracy errors are present. In this architecture, there are eight systolic arrays, which are used to compute large-scale matrix multiplications and matrix inversion together. This architecture can achieve high throughput, but the area and power consumptions remain unacceptable; in addition, the approximate error is large in the signal detection process. The GS-method-based architecture has been proposed to iteratively achieve matrix inversion for signal detection [13], [14]. The GS method estimates the transmitting signal vector ($\mathbf{s}$) as

$$\hat{\mathbf{s}}^{(k)} = (\mathbf{P} + \mathbf{L})^{-1}\left(\mathbf{y}^{\mathrm{MF}} - \mathbf{L}^H\hat{\mathbf{s}}^{(k-1)}\right). \tag{7}$$

Considering the definitions of $\mathbf{P}$ and $\mathbf{L}$, the solution can also be presented as

$$\hat{s}_i^{(k)} = \frac{1}{A_{ii}}\left(y_i^{\mathrm{MF}} - \sum_{i<j} A_{ij}\hat{s}_j^{(k)} - \sum_{j>i} A_{ij}\hat{s}_j^{(k-1)}\right), \tag{8}$$

where $\hat{s}_i^{(k)}$, $\hat{s}_i^{(k-1)}$ and y_i^{MF} denote the i-th elements of $\hat{\mathbf{s}}^{(k)}$, $\hat{\mathbf{s}}^{(k-1)}$ and $\mathbf{y}^{\mathrm{MF}}$, respectively, and A_{ij} denotes the i-th row and j-th column of $\mathbf{A}$. This method effectively scales down the number of matrix multiplications of the matrix inversion. However, this method suffers from low parallelism because of the strong correlation between each element in the transmitting signal. The computation of $\hat{s}_i^{(k)}$ uses the elements of $\hat{\mathbf{s}}^{(k)}$ in the current iteration and the elements of $\hat{\mathbf{s}}^{(k-1)}$ in the previous iterations. In addition, the SOR and SSOR methods [16]–[18] also suffer problems regarding low parallelism because these methods have similar computations in each iteration. Considering its corresponding hardware implementation, the computations for each transmitting signal in this method cannot be simultaneously performed. So, the GS- and SOR-based architectures in [14], [17] cannot achieve high throughput.

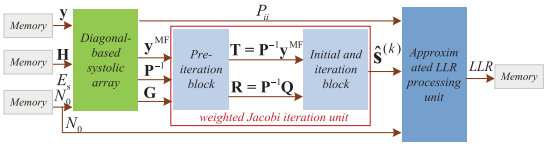


Fig. 1. Top-level block diagram of VLSI architecture. (LLR: log-likelihood ratio)

Implicit method architectures, such as CG [19], OCD [20] and IIC [21], have been proposed to explore the parallelism between matrix multiplication and inversion. In these explicit methods, the multiplication of the Gram matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ is first computed and then followed by the computation of the multiplication of the Gram matrix by the estimated vector $\hat{\mathbf{s}}$. However, for implicit methods, the above computations can be transformed to first compute $\mathbf{H}\hat{\mathbf{s}}$. Then, the matrix $\mathbf{H}^H$ is multiplied by the result vector. These methods can reduce the computing load of the Gram matrix multiplication, allowing the matrix multiplication and inversion computations to be performed in parallel. However, these implicit methods ignore the unique properties of a massive MIMO system (e.g., channel hardening). Therefore, the same Gram matrix needs to be calculated multiple times, which means that implicit method architectures suffer from higher energy consumption and latency compared to explicit methods.

Although these algorithms and architectures partially solve the problems of MMSE detection in massive MIMO systems, an efficient trade-off across computing load, processing parallelism and detection accuracy is difficult to achieve. In addition, the unique massive MIMO channel properties are not considered. These issues should be considered for hardware designs aiming to achieve high throughput with low area and energy consumptions.

III. Proposed Massive MIMO Detector

In this section, a VLSI architecture is designed to achieve massive MIMO detection based on a modified MMSE signal detection algorithm. The architecture was designed for a 64-QAM, 128×8 massive MIMO system case study. Fig. 1 shows the top-level block diagram for the proposed massive MIMO detector. To achieve a higher throughput with limited hardware resources, the top-level architecture is fully pipelined. The VLSI architecture is divided into three main components. In the first preprocessing unit (diagonal-based systolic array), the Gram matrix $\mathbf{G}$, $\mathbf{P}^{-1}$ and the matched-filter $\mathbf{y}^{\mathrm{MF}}$ are computed using inputs of the detector such as the received vector $\mathbf{y}$, the channel matrix $\mathbf{H}$, N_0 and E_s. These input data are stored in different memories of the architecture. A total of 32 static random access memories (SRAM) are used to store the complex values of the channel matrix $\mathbf{H}$ and the received vector $\mathbf{y}$. A total of 8 elements of the matrix $\mathbf{H}$ and vector $\mathbf{y}$ are read during each clock cycle. The memory sizes of the matrix $\mathbf{H}$ and vector $\mathbf{y}$ are 3 KB and approximately 0.34 KB, respectively. In addition, various parameters, such as N_0 and E_s, are stored in memory. In the second unit, the result matrices $\mathbf{G}$, $\mathbf{P}^{-1}$ and the vector $\mathbf{y}^{\mathrm{MF}}$ are used to iteratively achieve the matrix inversion with the WeJi method. The WeJi

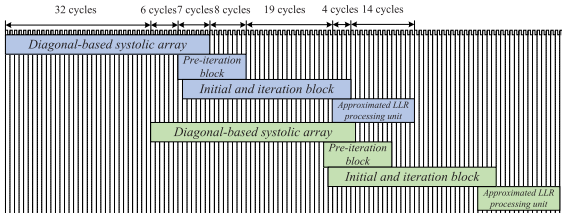


Fig. 2. Timing schedule of the proposed VLSI architecture.

unit includes various blocks. The pre-iteration block is used to compute the iterative matrix $\mathbf{R}$ ($\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$) and vector $\mathbf{T}$ ($\mathbf{T} = \mathbf{P}^{-1}\mathbf{y}^{\mathrm{MF}}$). The results of the pre-iteration block are transmitted to the initial and iterative block to achieve the computation of the final vector, $\hat{\mathbf{s}}^{(k)}$. Based on the simulation results and analyses (Section IV), $K = 2$ is chosen as the iteration number of the WeJi method for implementation, therein being sufficient to achieve high detection accuracy with low resource consumption. In the third unit, the final vector, $\hat{\mathbf{s}}^{(k)}$; the diagonal element of the MMSE filtering matrix, P_{ii}; and the parameter N_0 are computed to obtain the outputs (LLRs). The outputs are stored in 16 SRAMs, which is approximately 0.1 KB.

Fig. 2 shows the timing schedule of the proposed detector for a 128×8 MIMO system. In the diagonal-based systolic array, 45 clock cycles are used to compute all the results. The 45 clock cycles include 32 clock cycles for complex-valued multiplications, 5 clock cycles for performing accumulations to compute the matrix $\mathbf{P}$, and 8 clock cycles for computing the reciprocal of the matrix $\mathbf{P}$. After 38 clock cycles, the results of the diagonal-based systolic array can be obtained and are used in the pre-iteration block (the first block of the WeJi unit). In the pre-iteration block, the computations of the matrix $\mathbf{R}$ and vector $\mathbf{T}$ require 15 and 8 clock cycles, respectively. The initial and iteration block (the second block of the WeJi unit) can start to compute the first element of the initial solution; then, the other elements of the initial solution can be computed immediately. After 11 clock cycles, the first iteration can be started. Similar to the first iteration, the second iteration can be stared after 11 clock cycles when the first iteration is stared. As a summary, the initial and iteration blocks consume 37 clock cycles in total. Finally, after 11 clock cycles from the start of the second iteration, the approximated LLR processing unit can use the first element of the result vector $\hat{\mathbf{s}}^{(k)}$ to achieve the computation of the LLR values. After 3 clock cycles, the LLR values can be computed and then stored in the output memory. Then, the following 15 LLR values are obtained consecutively. The LLR unit consumes 18 clock cycles in total. In the proposed VLSI architecture, the average utilization rates of the diagonal-based systolic array and the initial and iteration block both approach 100%. The two models are complex (compared to the pre-iteration block and approximated LLR processing unit) and have much higher area compensations. To precisely transfer data, the input and output data of the pre-iteration block and the approximated LLR processing unit have to be matched with the data of the two major models. As a result, the average utilization rate of the pre-iteration block and the approximated LLR processing unit is approximately 60%. The following section presents each of the units in detail.

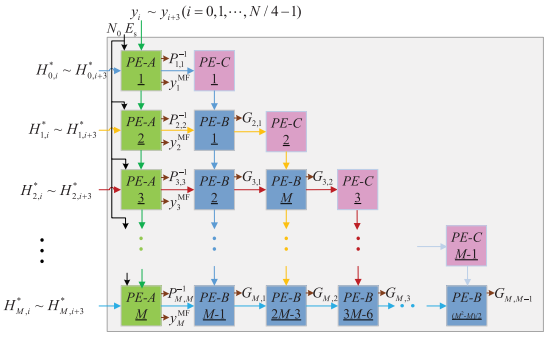


Fig. 3. Architecture of diagonal-based systolic array (PE: processing element).

A. Diagonal-Based Systolic Array

In the first preprocessing unit, a diagonal-based systolic array with single-sided input is designed to perform the computation of the Gram matrix and matched filter. Fig. 3 details the proposed architecture of the systolic array. Considering the scale of a massive MIMO system, this unit includes three different processing elements (PE). There are M PE-As, $\frac{M^2-M}{2}$ PE-Bs and $M-1$ PE-Cs in a deep pipeline. For example, in this 128×8 MIMO system, there are 8 PE-As, 28 PE-Bs and 7 PE-Cs. We take the first PE-A, PE-B and PE-C as an example to show the architectures in detail in Fig. 4. The PE-A is used to compute the matched filter $\mathbf{y}^{\mathrm{MF}}$, the diagonal elements of the Gram matrix, $\mathbf{G}$, and their inversion, $\mathbf{P}^{-1}$. The PE-A includes four groups of arithmetic logical units (ALUs), three accumulators (ACCs) and a reciprocal unit (RECU). The ALU-A and ALU-B are used to compute the real and imaginary parts of each element of the result matrix, respectively (Fig. 4-(a)). The outputs $P_{i,i}^{-1}$ and both the real and imaginary parts of y_i^{MF} are transmitted to the next block to compute. In the RECU, the reciprocal of the diagonal element of the matrix $\mathbf{P}$ is obtained from an LUT. The LUT stores the reciprocal of $\mathbf{P}$ from 72-200, which has minimal influence on the detection accuracy because the value of each element of $\mathbf{P}$ is near 128 (the number of antennas at the BS) [1]. Fig. 4-(b) shows the details of the PE-B, which performs the computation of the off-diagonal elements of the matrix $\mathbf{A}$. The PE-C is used to perform input data conjugation (Fig. 4-(c)). Note that the different types of calculations included in the PEs (all PEs in this massive MIMO detector) are achieved through multiple pipelines, and there are pipeline registers storing data between every calculation. For example, in the ALU-A of Fig. 4, the results achieved by the multiplier are stored in pipeline registers and serve as an input for the adder in the next step. It requires multiple periods to realize the ALU-A results serving as an input for the ACC during the add calculations. The results of every period are stored in pipeline registers. All other PEs follow the same type of multiple-period pipeline architecture. For this systolic array, four elements of the transposed input data matrix $\mathbf{H}^H$ and matrix $\mathbf{y}$ are simultaneously transmitted to the PE-A. To ensure that each PE processes the correct set of operands, the values in the i-th row of $\mathbf{H}^H$ are delayed by $(i-1)$ clock cycles. First, each value of $\mathbf{H}^H$ is transmitted from the PE-A to the PE-B and then to the PE-C (by row); they are then transmitted from the PE-C to the PE-B (by column). Given that the inversion of the matrix $\mathbf{P}$ is used to compute the matrix $\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$ and

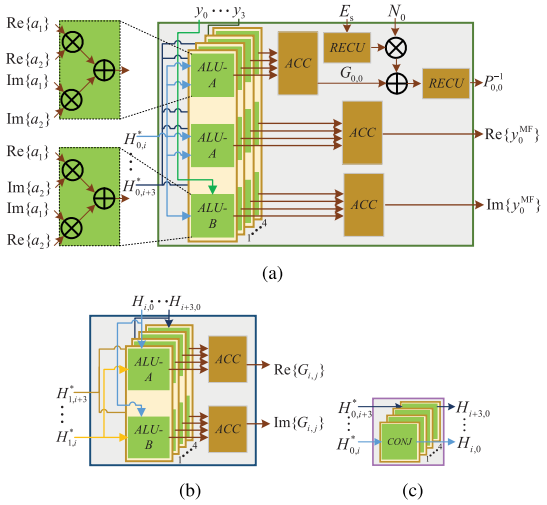


Fig. 4. Architecture of the PE-A, PE-B, and PE-C (MU: multiplied unit, ALU: arithmetic logical unit; ACC: accumulator; RECU: reciprocal unit, CONJ: conjugate). In the ALU-A, the adder symbol is a placeholder for both addition and subtraction. (a) PE-A. (b) PE-B. (c) PE-C.

vector $\mathbf{T} = \mathbf{P}^{-1}\mathbf{y}^{\mathrm{MF}}$ in the next unit (WeJi unit), the inversion of the diagonal elements has to be computed as soon as possible. This is why all the PE-As are the first processing elements in each row, being on the left side of the array. The outputs of the PE-A are transmitted to the next unit per clock cycle after the initial latency. Hence, this diagonal-based single-sided input systolic array can achieve high throughput and high hardware utilization.

Similar systolic arrays can be found in [9], [10] and [14]. In these architectures, the PE-A is not the first processing element, being in the diagonal part of the systolic array. Hence, the computations of the diagonal elements of the Gram matrix $\mathbf{G}$ are delayed, thereby consuming 15 clock cycles. This architecture reduces the number of clock cycles needed to compute the matrix $\mathbf{P}$ by half, i.e., it doubled the throughput. In [9], [10] and [14], double-sided inputs for the PE-A are used; however, in the proposed architecture, single-sided inputs are used, which reduces the number of registers in the input side by half due to the existence of the conjugate processing elements. The cost of this block results from the PE-C and is acceptable. As discussed in Section II-B, the architectures for the implicit method does not include the Gram matrix compute unit [19]–[21] because the Gram matrix is transmitted to two vector multiplications. The throughput of these architectures is high, and the hardware resource usage and power consumption are low . However, in an actual system, the same Gram matrix results as those in the implicit architectures are computed many times when considering the unique property (i.e., channel hardening) of a massive MIMO system (as discussed in Section II-B). Hence, the energy consumption and latency of these implicit architectures are very high in actual massive MIMO systems. In contrast, the energy consumption latency of the proposed systolic array for the Gram matrix computation is lower because of the reusability of the Gram matrix result.

B. Weighted Jacobi Iteration Unit

1) Signal Detection Based on Weighted Jacobi Iteration: In a massive MIMO system, the signal detection method is used to solve linear equations such as (2). As discussed in Section II, the complicated matrix inversion of $\mathbf{A}$ suffers from a high computing load and is difficult for parallel computation. Hence, such an inversion is not an ideal approach to solve the equation considering hardware implementations. In a massive MIMO system, the matrix $\mathbf{H}$ is asymptotically orthogonal, which means that the Gram matrix $\mathbf{G}$ and consequently $\mathbf{A}$ are Hermitian positive definite [1]. Therefore, the matrix $\mathbf{A}$ is split into two parts, $\mathbf{A} = \mathbf{P}+\mathbf{Q}$, where $\mathbf{P}$ includes the diagonal components of $\mathbf{A}$ and $\mathbf{Q}$ includes diagonal elements equal to zero and off-diagonal elements equal to those of $\mathbf{A}$. To exploit the weighted Jacobi iteration (WeJi) to approximately solve the linear equation in (2), the signal can be estimated as

$$\begin{aligned}\hat{\mathbf{s}}^{(k)} &= \mathbf{B}_W\hat{\mathbf{s}}^{(k-1)} + \mathbf{F} \\ &= \left((1-\omega)\,\mathbf{I} - \omega\mathbf{P}^{-1}\mathbf{Q}\right)\hat{\mathbf{s}}^{(k-1)} + \omega\mathbf{P}^{-1}\mathbf{y}^{\mathrm{MF}},\end{aligned} \tag{9}$$

where $\mathbf{B}_W = (1-\omega)\mathbf{I} - \omega\mathbf{P}^{-1}\mathbf{Q}$ and $\mathbf{F} = \omega\mathbf{P}^{-1}\mathbf{y}^{\mathrm{MF}}$ represent the iterative matrices, k is the number of iterations, and $\hat{\mathbf{s}}^{(0)}$ is the initial solution (discussed in the next paragraph). The parameter ω, which plays an important role in both the convergence and the convergence rate, satisfies the expression $0 < \omega < 1$. The proposed iteration algorithm is convergent because ω satisfies $0 < \omega < 2/\rho(\mathbf{P}^{-1}\mathbf{A})$ [22]. The matrix multiplication is achieved by a vector multiplication when utilizing this algorithm. Here, $\mathbf{P}$ is the main diagonal of the matrix $\mathbf{A}$; therefore, the diagonal matrix can be used to solve (9) to obtain the vector $\hat{\mathbf{s}}^{(k)}$. In addition, the computing load required to calculate $\mathbf{P}^{-1}$ is very low because $\mathbf{P}$ is a diagonal matrix; this is another reason why this algorithm achieves reduced computing load.

The initial solution of the iteration affects the detection accuracy and computing load when the number of iterations is limited. The next task in the signal detection method determines the initial solution, which traditionally is set as a zero vector because no a priori information of the final solution is available. For uplink massive MIMO systems, the matrix $\mathbf{A}$ becomes diagonally dominant. Based on this property, a low-computing-load initial solution is proposed using the Neumann series approximation. The matrix $\mathbf{A}^{-1}$ can be described as (6). Consequently, the vector $\hat{\mathbf{s}}^{(0)}$ can be computed as

$$\hat{\mathbf{s}}^{(0)} = \left(\mathbf{I} - \mathbf{P}^{-1}\mathbf{Q}\right)\mathbf{P}^{-1}\mathbf{y}^{\mathrm{MF}} = (\mathbf{I} - \mathbf{R})\,\mathbf{T}. \tag{10}$$

The approximation error of $\hat{\mathbf{s}}^{(0)}$ can be small because the matrix $\mathbf{P}$ includes the diagonal components of the matrix $\mathbf{A}$. This indicates that the initial solution converges faster than the traditional zero-vector solution. Hence, the proposed method can reduce hardware resource consumption and increase throughput.

Per the definition of $\hat{\mathbf{s}}^{(k)}$ in (9), the approximation error of the WeJi algorithm can be described with (11) because $\hat{\mathbf{s}}^{(\infty)}$ equals $\mathbf{s}$ when the iteration number $k \to \infty$ [9].

$$\Delta = \mathbf{s} - \hat{\mathbf{s}}^{(k)} \approx \hat{\mathbf{s}}^{(\infty)} - \hat{\mathbf{s}}^{(k)} = \mathbf{B}_W^k\left(\mathbf{s} - \hat{\mathbf{s}}^{(0)}\right) \tag{11}$$

Per the convergence rate definition [22], the convergence rate of the WeJi method is

$$R\left(\mathbf{B}_W\right) = -\ln\left(\lim_{k\to\infty}\left\|\mathbf{B}_W^k\right\|^{1/k}\right) = -\ln\left(\rho\left(\mathbf{B}_W\right)\right), \tag{12}$$

where $\rho(\mathbf{B}_W)$ is the spectral radius of the iteration matrix $\mathbf{B}_W$. For different methods, a smaller $\rho(\mathbf{B})$ leads to a higher convergence rate.

Lemma 1: In massive MIMO systems, $\rho(\mathbf{B}_W) \leq \omega\rho(\mathbf{B}_N)$, where $\rho(\mathbf{B}_W) = \rho\left((1-\omega)\mathbf{I} - \omega\mathbf{P}^{-1}\mathbf{Q}\right)$ and $\rho(\mathbf{B}_N) = \rho(\mathbf{P}^{-1}\mathbf{Q})$ are the iterative matrices of the WeJi and NSA methods, respectively.

Proof: See Appendix A. ∎

Hence, Lemma 1 indicates that the proposed WeJi detection achieves a higher convergence rate than the NSA detection. In addition, without loss of generality, the l_2-norm is used to evaluate the approximation error in (11) as

$$\|\Delta\|_2 \leq \left\|\mathbf{B}_W^k\right\|_F \left\|\mathbf{s} - \hat{\mathbf{s}}^{(0)}\right\|_2 \leq \|\mathbf{B}_W\|_F^k \left\|\mathbf{s} - \hat{\mathbf{s}}^{(0)}\right\|_2. \quad (13)$$

According to (13), if the Frobenius norm of the iteration matrix $\mathbf{B}_W$ of the WeJi method satisfies $\|\mathbf{B}_W\|_F < 1$, the approximation error of the proposed WeJi method is exponentially close to zero with increasing iteration number K.

Lemma 2: In massive MIMO systems, the probability of $\|\mathbf{B}_W\|_F < 1$ satisfies

$$\Pr\{\|\mathbf{B}_W\|_F < 1\} \geq 1 - \omega\sqrt[4]{\frac{(N+17)(M-1)M^2}{2N^3}}, \quad (14)$$

where $\|\mathbf{B}_W\|_F$ is the Frobenius norm of the iteration matrix $\mathbf{B}_W$ of the WeJi method.

Proof: See Appendix B. ∎

Lemma 2 indicates that when the number of users (M) is fixed, an increase in N will cause $\Pr\{\|\mathbf{B}_W\|_F < 1\}$ to increase, which means that the probability of $\|\mathbf{B}_W\|_F < 1$ will increase. For massive MIMO systems, the number of antennas at the BS is much larger than the number of users ($N \gg M$), which indicates that the probability of $\|\mathbf{B}_W\|_F < 1$ is close to 1. According to (13), the approximation error of the proposed WeJi detection is close to zero, thereby approximating the exact inversion MMSE. In this paper, the WeJi method does not directly achieve matrix inversion; the WeJi method is an iterative method that relies on matrix inversion. The WeJi method combines matrix inversion with matrix-vector multiplication and uses an iterative method to estimate the transmitted signal.

2) Summary and Analyses of WeJi: The WeJi algorithm first performs the computation of the matrices $\mathbf{R}$ ($\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$) and $\mathbf{T}$ ($\mathbf{T} = \mathbf{P}^{-1}\mathbf{y}^{\mathrm{MF}}$). To facilitate WeJi detection, the initial solution should be computed as soon as possible. Thus, the vector $\mathbf{T}$ and the matrix $\mathbf{R}$ should be prepared within an allotted time. Second, the initial solution $\hat{\mathbf{s}}^{(0)}$ is computed according to (10). Note that when considering the hardware design, the architecture for the initial solution can be reused in the next iteration block. Finally, the iterations in (9) for the final value $\hat{\mathbf{s}}^{(k)}$ are performed. In the iteration part, the matrix multiplication was achieved by a vector multiplication, and all the elements of the estimated vector can be determined in parallel. The computing load of the complex matrix inversion is reduced using iterations. In addition, the weighted parameter results in small iteration numbers toward achieving a similar detection performance, which also reduces the computing load of the detector. Now, the proposed WeJi algorithm is compared with recently developed algorithms in terms of computing load, parallelism, and hardware-design realizability.

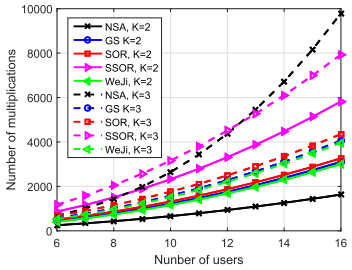


Fig. 5. The numbers of multiplications under the WeJi algorithm and the other methods.

Because both the MMSE algorithm and the proposed WeJi-based signal detection method need to compute the matrices $\mathbf{G}$ and $\mathbf{y}^{\mathrm{MF}}$, the current work focuses on the computing load of the matrix inversion and LLR computations, as in [9]–[14]. The computing load can be evaluated in terms of the required number of real multiplications, with each complex multiplication requiring four real multiplications. The first calculation comes from the computation of the multiplications of the $M \times M$ diagonal matrix $\mathbf{P}^{-1}$ with the $M \times M$ matrix $\mathbf{Q}$ and with the $N \times 1$ vector $\mathbf{y}^{\mathrm{MF}}$, which are $2M(M-1)$ and $2M$ in number, respectively. The second calculation comes from the multiplications of the iteration matrices $\mathbf{B}$ and $\mathbf{F}$, which include $4M$ real multiplications. The third calculation comes from the computation of the initial solution. The final calculation comes from the computation of the channel gain, NPI variance and LLRs. Hence, the total number of multiplications required by the WeJi algorithm is $(4K+4)M^2 - (4K-4)M$. Fig. 5 shows the numbers of real multiplications performed by the WeJi algorithm and the compared methods. The WeJi algorithm has a lower computing load than the GS, SOR and SSOR methods. The NSA method presents a much lower computing load when the iteration number satisfies $K = 2$. Generally, according to the analysis in [14], K should not be less than 3 in the NSA method to ensure satisfactory detection accuracy. When $K = 3$, the NSA method exhibits a higher computing load of $\mathcal{O}(M^3)$. Therefore, the reduction in the computing load of the NSA method is marginal.

Further important aspects for the hardware implementation of this signal detection algorithm must still be considered. The WeJi algorithm can be performed in parallel. As in (8), the solution of the WeJi algorithm in (9) can be rewritten as

$$\hat{s}_i^{(k)} = \frac{\omega}{A_{ii}} y_i^{\mathrm{MF}} + \frac{\omega}{A_{ii}} \sum_{j \neq i} \left(A_{ij}\hat{s}_j^{(k-1)} + (1-\omega)\hat{s}_j^{(k-1)}\right). \quad (15)$$

The calculation of $\hat{s}_i^{(k)}$ only requires the elements of the previous iterations. Therefore, each computation for all elements of $\hat{\mathbf{s}}^{(k)}$ can be performed in parallel. However, for the GS, SOR and SSOR methods, the iteration has a strong correlation for each transmitting signal (as discussed in Section II). According to (8), the computation of $\hat{s}_i^{(k)}$ uses the elements of $\hat{s}_j^{(k)}$ for $j = 1, 2, \ldots, i-1$ in the current k-th iteration and the elements of $\hat{s}_j^{(k-1)}$ for $j = i, i+1, \ldots, M$ in the previous $(k-1)$-th iteration. This means that the computations for each element cannot be performed in parallel. For this reason, the GS and SOR method architectures in [14], [17] cannot achieve high throughput, being much lower than that of the WeJi detection.

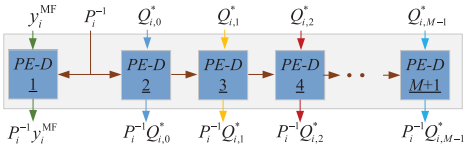


Fig. 6. Pre-iteration block architecture.

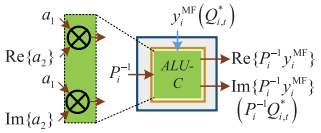


Fig. 7. PE-D architecture.

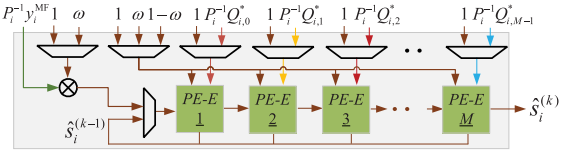


Fig. 8. Initial and iteration block architecture.

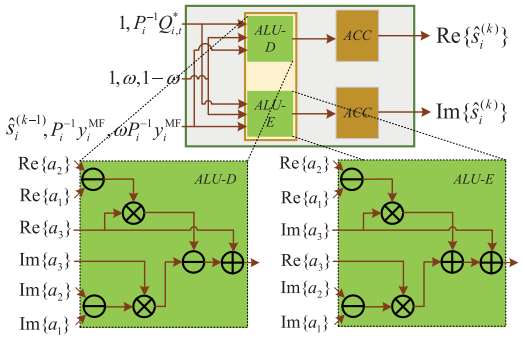


Fig. 9. PE-E architecture.

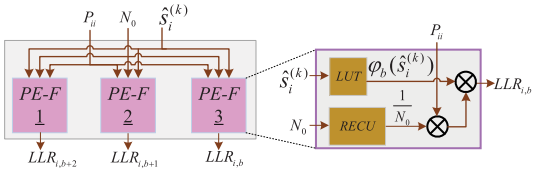


Fig. 10. Architecture of the approximated LLR processing unit. (LUT: look-up table)

Note that [23] proposes a Jacobi-based detection method. Compared to this method, the proposed WeJi method achieves a better performance in the following three regards. First, the proposed WeJi method is a method based on hardware architecture design considerations, which means that hardware implementation is fully considered in the process of algorithm optimization and improvement. In the process of algorithm design, the considered aspects include the detection accuracy, computing load, parallelism, and hardware reusability. In contrast, [23] does not consider hardware implementation aspects such as parallelism and hardware reusability. Second, WeJi and [23] adopt different initial iteration solution calculation methods. The initial value in [23] is constant (it is related to the modulation mode); this constant initial solution is quite different from the final result. In contrast, the proposed method considers massive MIMO system features and includes a method to compute the initial solution. By following (10), the iterative initial solution is close to the final result. As a result, the subsequent iteration number can be reduced, and unnecessary hardware consumption can be minimized. Moreover, the proposed method for computing the initial solution is similar to that of the following iteration, and the hardware resources can be reused for both the initial and final solution calculations. Because the Gram matrix computation before the iterative calculation would occupy much of a clock cycle, the reuse of hardware resources does not affect the throughput of the system. Third, compared with [23], the WeJi algorithm introduces a weight factor, as shown in (9). Hence, the accuracy can be improved. Therefore, the hardware resource consumption will be decreased. Moreover, the same unit can be reused to improve the unit usability rate when performing pre-iteration and iteration. In addition, this reuse does not affect the throughput.

3) Architecture for WeJi: An architecture based on the WeJi method is proposed. There are two blocks in the WeJi unit: the pre-iteration block and the initial and iteration block. The pre-iteration block is proposed to satisfy the requests of the input data in the initial and iteration block. Fig. 6 presents the details of the pre-iteration block, where $M+1$ PE-Ds compute in parallel in a deep pipeline (9 PE-Ds for a 128×8 MIMO system). There are two main operations for this block: the computation of the vector $\mathbf{T} = \mathbf{P}^{-1}\mathbf{y}^{\mathrm{MF}}$ and the iteration matrix $\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$. The calculations of these two parts are simultaneously processed, and the result of each PE-E is computed per clock cycle, resulting in high parallelism. Fig. 7 shows the architecture of the PE-D, which includes one ALU-C (real-complex multiplication). Here, the input data $\mathbf{P}^{-1}$ are a real matrix; therefore, the computations can be simplified.

In [9] and [10], after computing the Gram matrix and matched-filter block, the computations of the matrices $\mathbf{R}$ and $\mathbf{T}$ are performed in a systolic array; as such, the hardware consumption is small. However, considering the throughput of the entire system, the computations of the second $\mathbf{G}$ and $\mathbf{y}^{\mathrm{MF}}$ are delayed due to the matrices $\mathbf{T}$ and $\mathbf{R}$; hence, the throughput is reduced. To maintain a high throughput, the computations of the matrices $\mathbf{T}$ and $\mathbf{R}$ are performed in another systolic array, which requires more processing elements. In this architecture, the pre-iteration block uses a pipelined mechanism to compute the matrices $\mathbf{T}$ and $\mathbf{R}$ within an exact time limit. Compared with [9] and [10], this architecture effectively utilizes the processing elements considering the time limitations. The throughput has no influence, and lower area and power consumptions are achieved.

An initial and iteration block is proposed to achieve high throughput and hardware processing speeds with limited area consumption (Fig. 8). This block is used to compute iterations with fully pipelined architectures because of the time limitations of the previous blocks. To fit the high frequency, this block has M processing elements, called PE-Es. For example, there are 8 PE-Es in a 128×8 MIMO system. Fig. 9 shows the details of a PE-E, which includes two ALUs (ALU-D and ALU-E) for computing the real and imaginary parts of the $\hat{s}_i^{(k)}$. There are eight pipeline registers in each input side of the PE-E. The input vector is transmitted to the PE-E element by

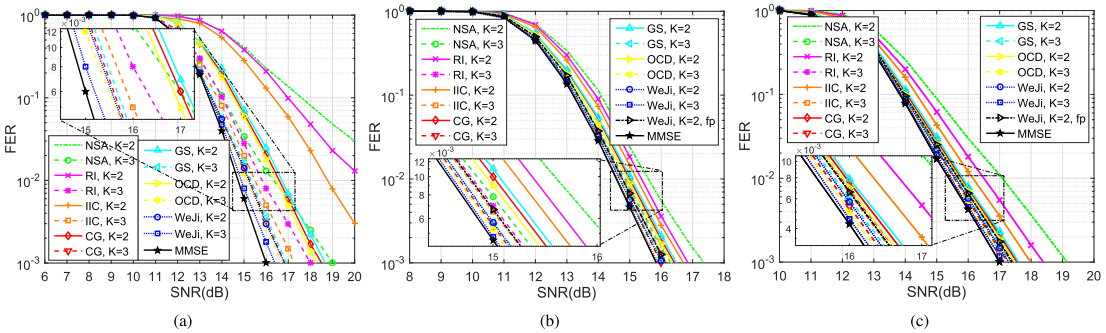


Fig. 11. FER performance comparison between the proposed algorithm and other algorithms under different MIMO configurations and code rates. (FER: frame error rate, SNR: signal-to-noise ratio) (a) $N = 64$, $M = 8$, 1/2 code rate. (b) $N = 128$, $M = 8$, 1/2 code rate. (c) $N = 128$, $M = 8$, 3/4 code rate.

element (from left to right), and the input matrix $\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$ is transmitted to each PE-E when the matrix is computed. In the first phase, the PE-Es are used to compute the initial solution based on the WeJi algorithm. The PE-Es compute the total time from when the first data were received. In the second phase, the PE-Es compute the iterations needed to obtain the vector $\hat{\mathbf{s}}^{(k)}$. The accumulation results, $\hat{s}_m^{(k)}$, of the PE-Es are stored in the input pipeline registers of the PE-Es.

In [9] and [10], these computations are also performed in a systolic array because they are needed to perform the multiplications of the matrix. Hence, additional processing elements are required, and each element of the systolic array is constantly performing calculations; this means that there are significant area and power consumptions. Compared with [9] and [10], this architecture block can perform the vector multiplications in place of the matrix multiplications at lower area and power consumptions. Compared with [14], this block shows that the proposed algorithm can achieve an $8\times$ higher parallelism than the proposed GS algorithm. The user-level parallelism unit can be utilized in this block; therefore, this block can be used to obtain a fully pipelined architecture that has no influence on the throughput of the complete system.

C. Approximated LLR Processing Unit

The approximated LLR processing unit is used to compute the LLR values for each transmitted bit based on the proposed algorithm. Following [9], [10], [13], the approximated LLR processing method is applied to the WeJi method and used to design an architecture. The NPI variance σ_{eq}^2 and the SINR ζ_i^2 can be computed as [9], [10], [13]

$$\sigma_{eq}^2 = E_s U_{ii} - E_s U_{ii}^2,$$
$$\zeta_i^2 = \frac{1}{E_s} \cdot \frac{U_{ii}}{1 - U_{ii}} \approx \frac{1}{E_s} \cdot \frac{\frac{P_{ii}}{P_{ii} + N_0 E_s^{-1}}}{1 - \frac{P_{ii}}{P_{ii} + N_0 E_s^{-1}}} = \frac{P_{ii}}{N_0}. \quad (16)$$

Fig. 10 shows the block diagram of the approximated LLR processing unit, which includes $\frac{1}{2}\log_2 \mathcal{Q}$ PE-Fs in this architecture for the $\mathcal{Q}$-QAM modulation. The first step calculates the SINR ζ_i^2 using P_{ii} and N_0 per (16). The authors note that the SINR value can be used for the same i-th user. The linear equation $\varphi_b(\hat{s}_i)$ is calculated with different $\hat{s}_i$. Following (4), the linear function $\varphi_b(\hat{s}_i)$ for the Gray mappings can be efficiently achieved in the hardware architecture. Next, the bit LLR values, $L_{i,b}$, are calculated using the SINR ζ_i^2. Fig. 10 also shows the details of the processing element, PE-F. Here, each of the linear equation coefficients, $\varphi_b(\hat{s}_i)$, are stored in a correction LUT; in addition, the effective channel gain, P_{ii}, is transmitted from the Gram and matched-filter block. In the RECU, the reciprocal of N_0 is achieved by the LUT. The block facilitates the computations of the LLRs to be simplified, which increases the processing speed and reduces both the area and power consumptions in the block. Although this method increases the number of LUTs, the increase is minimal; as such, the method is acceptable.

IV. Frame-Error-Rate Simulation

The FER simulation results of the proposed signal detection algorithm and the recently proposed algorithms are provided in this section. The FER performance of the classical MMSE algorithm with exact matrix inversion (CHD method) is also provided for comparison. The 64-QAM modulation scheme is used. The rate-1/2 industry standard convolutional code with a $[133_o\ 171_o]$ polynomial along with a random interleaver is adopted. The coding is performed over 120 symbols, and the channels are assumed to be i.i.d. Rayleigh fading is assumed across time/subcarriers, and the number of frames is 10,000. The outputs (LLRs) are used in the Viterbi decoding. At the receiver, the LLRs are Viterbi decoding soft input. Following [9], the SNR is defined at the receiver. These simulation settings are quite practical for use in, for example, long-term evolution (LTE), LTE-Advanced, digital video broadcasting for satellite and shortwave communication [24], [25]. A type of parallel concatenated convolutional code, the turbo code, has been discussed for 4G and 5G [24], [25]. The currently used turbo scheme in 4G is also an important coding scheme for 5G and sees extensive usage. In addition, these simulation settings are frequently used in many massive MIMO detection algorithms and architectures for 5G [13]–[18].

Fig. 11 shows the FER performance comparison between the proposed WeJi, NSA [9], [10], RI [15], IIC [21], CG [19], GS [13], [14], OCD [20], and MMSE algorithms [11], [12]. In Fig. 11-(a), these algorithms are simulated in a 64×8 massive MIMO system with a 1/2 code rate. Fig. 11-(b) shows the FER performance results for a 128×8 massive MIMO system with a 1/2 code rate. To prove that the proposed method also possesses advantages in terms of higher

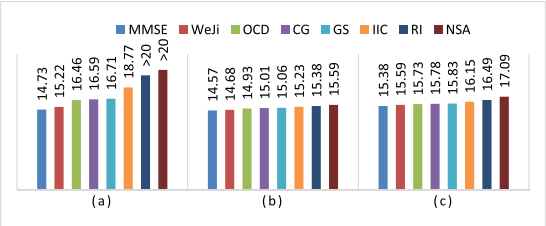


Fig. 12. SNR (dB) requirements for achieving an FER of 10^{-2}.

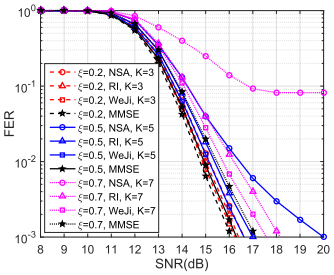


Fig. 13. FER performance comparison for the Kronecker channel model.

code rates, Fig. 11-(c) shows the performance results for a 128×8 massive MIMO system with a 3/4 code rate. The comparisons of these simulation results show that the WeJi method can achieve near-optimal performance under different MIMO configurations and code rates. To achieve the same FER, the WeJi method requires an SNR that is almost identical to those needed for the MMSE algorithm but lower than those required by the OCD, CG, GS, IIC, RI, and NSA methods. For example, Fig. 12 shows the performance comparison for achieving an FER of 10^{-2}. This value is appropriate for regular LTE data channels [1], [25]. In addition, 10^{-2} FER is the most commonly used value in comparing different algorithms in recent studies, which indicates that this paper can be compared fairly with other references [1], [11]. According to Fig. 12, the proposed WeJi method achieves a better FER performance than state-of-the-art methods in different MIMO configurations.

The authors note that previous simulation results were based on the Rayleigh fading channel model. To prove that the proposed algorithm is also superior in more realistic channel models, Fig. 13 shows the influences of the large-scale fading and spatial correlation of the MIMO channel. The Kronecker channel model has previously been used to evaluate performance because it is more practical than the i.i.d. Rayleigh fading channel model [26]. The Kronecker channel model assumes that the transmit and receive correlations are separable. Measurements have shown that the Kronecker model is a good approximation for non-line-of-sight scenarios. Hence, this model is widely used in the literature [13], [26]. In this channel model, the elements of the channel matrix satisfy $\mathcal{CN}(0, d(\mathbf{z})\mathbf{I}_B)$, where $d(\mathbf{z})$ is an arbitrary function that accounts for channel attenuation (such as shadowing and path loss). The classic path loss model is considered with the channel attenuation variance $d(\mathbf{z}) = \frac{C}{\|\mathbf{z}-\mathbf{b}\|^\kappa}$, in which $\mathbf{z} \in \mathbb{R}^2$, $\mathbf{b} \in \mathbb{R}^2$, κ and $\|\cdot\|$ denote the locations of the user and BS, the path loss exponent and the Euclidean norm, respectively. The independent shadow fading C satisfies $10\lg C \sim \mathcal{N}(0, \sigma_{sf}^2)$. Combining with the correlation matrix ($\dot{\mathbf{R}}$), the Kronecker channel matrix $\mathbf{H}$ can be expressed as

$$\mathbf{H} = \dot{\mathbf{R}}^{1/2}\mathbf{H}_{i.i.d}\sqrt{d(\mathbf{z})}\dot{\mathbf{R}}^{1/2}, \tag{17}$$

where $\mathbf{H}_{i.i.d}$ is a random matrix whose entries are i.i.d., with a complex Gaussian distribution of zero mean and unit variance. The exponential correlation is a model that is used to generate the correlation matrix and is explained in greater detail in [24]. The elements of the correlation matrix $\dot{\mathbf{R}}$ can be written as

$$\dot{r}_{ij} = \begin{cases} \xi^{j-i}, & i \le j \\ (\xi^{j-i})^*, & i > j \end{cases} \tag{18}$$

where ξ is the correlation factor between the neighbouring branches. Users in the same cell are uniformly distributed in a hexagon with a radius of $r = 500$ meters. The following assumptions are adopted for the simulation: $\kappa = 3.7$, $\sigma_{sf}^2 = 5$ and transmit power $\rho = r^\kappa/2$. The correlated factor ξ is 0.2, 0.5 and 0.7, sequentially. Fig. 13 shows that to achieve the same FER, the SNR required by the proposed algorithm is also smaller than that in the CG, RI and NSA methods, which proves that the proposed algorithm can maintain its advantages in a realistic model.

V. Silicon Implementation and Comparison

The proposed hardware architecture is verified on an FPGA platform (Xilinx Virtex-7) and implemented using TSMC 65 nm 1P8M CMOS technology. Details of the hardware features, derived from the silicon implementation, are compared to state-of-the-art designs. In addition, the fixed-point design and its implementation performance with respect to detection accuracy is also presented. Note that the throughput Θ of the detector is formulated as

$$\Theta = \frac{\log_2 \mathcal{Q} \times M}{T_s} \times f_{\text{clk}}, \tag{19}$$

where f_{clk} is the clock frequency, $\mathcal{Q}$ is the constellation size, M is the number of users, and T_s is the number of clock cycles needed for the calculations per symbol vector. According to (19), the throughput of the architecture in this paper is closely related to clock frequency, user number, constellation size and processing cycles. In addition, the number of iterations, scale of hardware resources and numbers of antennas and users will affect the processing cycle. In this architecture, the number of clock cycles is designed to satisfy $T_s = \frac{N}{4}$.

A. Fixed-Point Design

To reduce the hardware resource consumption, fixed-point arithmetic is used throughout the design. Based on extensive simulations, the associated fixed-point parameters are determined. Note that the word widths refer to the real or imaginary part of a complex-valued number. The inputs of the architecture are all quantized to 14 bits, including the received signals $\mathbf{v}$, the flat Rayleigh fading channel matrix $\mathbf{H}$, and the power spectral density of the noise N_0. Hence, the multiplications are quantized to 14 bits, and the results are transmitted to the accumulator in the diagonal-based systolic array, which is set to 20 bits. The LUT for achieving the reciprocal of an

TABLE I
COMPARISON OF RESOURCE USAGE ON A XILINX VIRTEX-7 FPGA

	This Work	[9]		[14]	[19]	[20]	[21]	[27]	
MIMO System	128×8 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	128×8 BPSK	128×8 QPSK
Inversion method	WeJi	CHD	NS	GS	CG	OCD	IIC	TASER	TASER
Preprocessing	Included (explicit)	Included (explicit)	Included (explicit)	Included (explicit)	Included (implicit)	Included (implicit)	Included (implicit)	Not included	Not included
LUT slices	20454	208161	168125	18976	3324	23914	72231	4790	13779
FF slices	25103	213226	193451	15864	3878	43008	151531	2108	6857
DSP48	697	1447	1059	232	33	774	1245	52	168
Frequency [MHz]	205	317	317	309	412	258	305	232	225
Throughput [Mb/s]	308	603	603	48	20	376	915	38	50
Throughput/slices [b] [Mbps/K slices]	6.76	1.43	1.67	1.38	2.78	5.62	4.09	5.51	2.42

[a] According to the simulation results, the number of iterations assumed is $K = 2$, which achieves near-optimal performance under the WeJi method.
[b] Summation of LUT and FF slices.

TABLE II
COMPARISON OF ASIC IMPLEMENTATION RESULTS

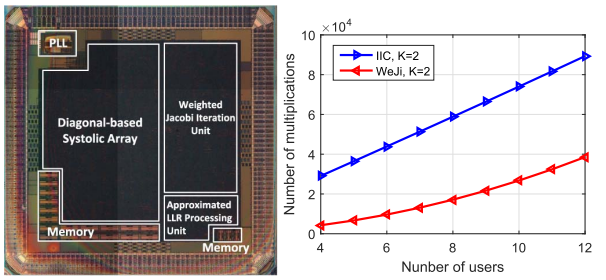
	This Work[a]	[28]	[21]	[27]		[29]	[12]
Technology	65 nm CMOS	45 nm CMOS	65 nm CMOS	40 nm CMOS		40 nm CMOS	28 nm FD-SOI
MIMO System	128×8 64-QAM	128×8 64-QAM	128×8 64-QAM	128×8 BPSK	128×8 QPSK	128×32 256-QAM	128×8 256-QAM
Inversion method	WeJi	NSA	IIC	TASER	TASER	MPD	CHD
Silicon proof	Yes	No (Layout)	No (Layout)	No (Layout)	No (Layout)	Yes	Yes
Preprocessing	**Included (explicit)**	**Included (explicit)**	**Included (implicit)**	**Not included**	**Not included**	**Not included**	**Not included**
Logic [M Gates]	1.07	6.65	4.3	0.142	0.448	-	0.148
Memory [KB]	3.52	15.00	-	-	-	-	-
Area [mm^2]	2.57	4.65	9.6	0.15	0.483	0.58	1.1
Frequency [GHz]	0.68	1.00	0.60	0.598	0.56	0.425	0.30
Power [W] (@ Voltage)	0.65 (@ 1.00V)	1.72 (@ 0.81V)	1.00 (-)	0.041 (@ 1.1V)	0.087 (@ 1.1V)	0.221 (@ 0.9V)	0.018 (@ 0.9V)
Throughput [Gbps]	1.02	2.0	3.6	0.099	0.125	2.76	0.3
Energy efficiency[b] [Gbps/W]	1.58	1.16	3.6	2.41	1.44	12.49	16.67
Area efficiency[b] [Gbps/mm^2]	0.40	0.43	0.375	0.66	0.26	4.76	0.27
Normalized[c] Energy efficiency [Gbps/W]	**1.58 (2.93[d]×)**	**0.54**	**3.6**	**1.11**	**0.66**	**3.83**	**2.51**
Normalized[c] Area efficiency [Gbps/mm^2]	**0.40 (2.86[d]×)**	**0.14**	**0.375**	**0.15**	**0.06**	**1.11**	**0.022**

[a] According to the simulation results, the number of iterations assumed is $K = 2$, which achieves near-optimal performance under the WeJi method.
[b] Energy and area efficiencies are defined as throughput/power and throughput/area, respectively.
[c] Technology normalized to 65 nm CMOS technology assuming the following: $f_{\text{clk}} \sim s$, $A \sim 1/s^2$, and $P_{\text{dyn}} \sim (1/s)(V_{\text{dd}}/V'_{\text{dd}})^2$.
[d] Energy and area efficiency ratios between this work and previous work [28].

element in the matrix **P** consists of 128 addresses with 12-bit outputs. The preprocessing block uses a 14-bit input, which indicates that the multiplications are quantized to 14 bits. In addition, the outputs of the preprocessing block are quantized to 14 bits, which are the inputs of the initial and iteration block. In the initial and iteration block, the multiplications are quantized to 14 bits, and the results are transmitted to accumulators, which are quantized to 18 bits to achieve sufficient accuracy at low hardware consumption. In addition, the outputs are set to 16 bits and transmitted to the LLR preprocessing block. In this block, the multiplications are set to 14 bits, and the outputs are set to 12 bits. The resulting fixed-point performance is shown in Fig. 11 (labeled 'fp') for a 128×8 MIMO system. In this architecture, the SNR loss needed to achieve an FER of 10^{-2} is 0.2 dB, which includes an approximately 0.11 dB error from the algorithm and an approximately 0.09 dB fixed-point error from the silicon implementation. In the hardware realization process, the entire architecture adopts fixed-point arithmetic to reduce the hardware resource consumption. Thus, as a result of the error produced by the fixed-point parameters, the hardware realization will increase the SNR loss compared to the software simulation, for example, from the truncation error resulting from the limited word length of the hardware, approximate reciprocal unit, LUT, etc. Note that the fixed-point error should increase during the iteration process because the proposed architecture is an iterative architecture. However, according to the simulation results in Fig. 11, the detection accuracy increases with increasing iteration number for the algorithm. As a result, the detection accuracy increases with increasing number of iterations. Fig. 11 shows the FER performance of the exact MMSE detector, the proposed algorithm, the fixed-point implementation and comparisons with other algorithms. Compared with state-of-the-art methods, the error-rate performance loss of the WeJi method (0.2 dB) is less than that under the NSA (0.36 dB) [9], [10], RI (0.43 dB) [15], IIC (0.49 dB) [21], CG (0.66 dB) [19], GS (0.81 dB) [13], [14], and OCD (1.02 dB) [20] methods.

B. FPGA Implementation

Table I summarizes the key implementation results on an FPGA platform (Xilinx Virtex-7). The results are compared with other architectures [9], [14], [19]–[21], [27] and represent the best solutions for massive MIMO detectors with FPGA implementations. Compared with the CHD-based architecture, the throughput/slice of the proposed architecture is 4.72× higher. This architecture scales down the throughput to 64.67% but reduces the (LUT+FF) slice consumption in the NSA-based detector by 87.40%; the DSP consumption is also reduced. Given the same resources, the throughput under



(a) (b)

Fig. 14. (a) Die micrograph. (b) The numbers of multiplications of the explicit (WeJi) and implicit (IIC) architectures with $T_c = 7$.

the proposed work would be 4.05× higher than that of the NAS-based architecture [9], which can be attributed to the proposed algorithm and its VLSI architecture. The GS method architecture [14] achieves a low hardware resource consumption compared with the WeJi architecture. However, the low throughput (48 Mb/s) is the limitation of the architecture of the GS method. This is due to the low parallelism of the computation of each element in the estimated vector (as discussed in Section II). Therefore, the WeJi-based architecture achieves 4.90× higher throughput/slice compared to the architecture of the GS method [14]. In addition, the proposed WeJi architecture is compared with the implicit architectures. The CG-based architecture [19] achieves a low hardware consumption, but the throughput is only 20 Mb/s, which is substantially smaller than that of the WeJi method. Considering throughput/slice, the WeJi method achieves a 2.43× higher throughput compared to the CG-based architecture [19]. Compared with the OCD-based architecture [20], the WeJi architecture achieves 1.20× throughput/slice. The IIC-based architecture [21] achieves a high throughput but consumes many slices and DSP. Hence, the WeJi method maintains its advantage in terms of throughput/slice, which is 1.65× higher than the IIC-based architecture [21]. Finally, the authors note that architectures with FPGA implementations for massive MIMO detectors with nonlinear algorithms, such as the two architectures in [27], have been developed. The detection accuracy of the nonlinear detection algorithm, the triangular approximate semidefinite relaxation (TASER) algorithm, is better than that of linear detection algorithms such as MMSE-based algorithms (WeJi, CHD, NAS, CG, IIC, etc.). Similar to GS- and CG-based architectures, these two TASER-based architectures achieve low throughput (38 Mb/s and 50 Mb/s). This low throughput limits the utilization of these architectures. The WeJi architecture achieves 1.23× and 2.79× higher throughput/slice compared with the two TASER-based architectures.

C. Silicon Implementation

The proposed MMSE detector is implemented onto a 2.57 mm^2 silicon chip with TSMC 65 nm 1P8M CMOS technology (Table II). Fig. 14-(a) shows the die micrograph of the chip. Note that the energy and area efficiencies are defined as throughput/power and throughput/area, respectively. [10] provides a state-of-the-art approach and efficiently solves the large-scale problems of ASIC implementations in massive MIMO systems (as presented in Section III). The authors note that the detector in [10] includes additional processing (e.g., inverse fast Fourier transform processing). To ensure a fair comparison, the same architecture without additional processing (from the author's doctoral dissertation [28]) is used as a comparison with the proposed architecture. In addition, for the different technologies, the energy and area efficiencies (Table II) are normalized to the 65 nm technology and with a 1 V supply voltage as

$$f_{\text{clk}} \sim s, \quad A \sim 1/s^2, \quad P_{\text{dyn}} \sim (1/s)(V_{\text{dd}}/V'_{\text{dd}})^2, \qquad (20)$$

where s, A, P_{dyn}, and V_{dd} denote the technology rate, area, power and voltage, respectively. This scaling method is widely used for comparing different architectures of different technologies such as [27]–[29]. The architecture [28] achieves a 0.54 Gbps/W normalized energy efficiency and a 0.14 Gbps/mm^2 normalized area efficiency. The comparison shows that the energy and area efficiencies are 2.93× and 2.86× those in [28], respectively.

[27] proposes two architectures based on TASER algorithms that can achieve high detection accuracies. However, both of their throughputs are very low (0.099 Gbps and 0.125 Gbps). The proposed WeJi architecture achieves a throughput of 1.02 Gbps, which is 10.3× and 8.16× that of the two architectures in [27]. In addition, the architectures from [27] can only be used for binary phase-shift keying (BPSK) or quadrature phase-shift keying (QPSK). These architectures are not appropriate for higher order modulation, which limits their application and development. For comparison, the results are normalized to 65 nm technology, as shown in Table II. The WeJi architecture exhibits a better performance in terms of normalized energy and area efficiencies compared with [27]. Specifically, compared to the two TASER detectors for BPSK and QPSK, the WeJi detector exhibits an increase in normalized energy efficiency of 1.42× and 2.39× as well as an increase in normalized area efficiency of 2.67× and 6.67×. Note that the preprocessing part is not included in the detectors [27]. According to Fig. 14-(a), the preprocessing part occupies a large portion of the chip (more than 50%). Thus, the preprocessing part will consume a significant amount of the power of the chip. As a result, if considering including the preprocessing part in the TASER detectors, the proposed WeJi detector should have a substantially better performance as a result of the increases in the energy and area efficiencies.

[12] proposes a detector design based on the CHD method. This detector has a relatively low throughput (0.3 Gbps), thereby limiting its application. The proposed WeJi architecture in this paper can realize a throughput of 1.02 Gbps (approximately 3.4×). The area consumption in [12] is 1.1 mm^2, which is smaller than the WeJi method. However, the area efficiency of the proposed WeJi method is 1.48× that in [12]. Considering that [12] adopted a 28 nm FD-SOI technology, the result of [12] is normalized to 65 nm technology. The normalized area efficiency of the WeJi method is approximately 18.18× that in [12]. The normalized energy efficiency achieved under the architecture of [12] is 1.58× that of the WeJi architecture. Note that the FD-SOI technology is adopted in the chip developed by [12], and the power of the FD-SOI technology is lower than CMOS technology when normalized to 65 nm [30]. Meanwhile, the results of [12] did not include the preprocessing part (i.e., the power consumed

by preprocessing is not included), while the results obtained under the WeJi architecture include the preprocessing part. Therefore, based on the above two reasons, the energy efficiency of the architecture in [12] should decrease remarkably. [29] proposes a message-passing detector (MPD) that can achieve very high throughput and normalized energy and area efficiencies. Note that [29] processes a 128×32 MIMO system, which shows an evident improvement in throughput when compared to a 128×8 MIMO system. The architecture in [29] does not include the preprocessing part, and according to the computing load analysis, in a 128×8 MIMO system, the proportion of the resource consumption in the preprocessing part is larger than that in a 128×32 MIMO system. Thus, considering preprocessing in the architecture, to ensure a high throughput of 2.76 Gbps, the area and power requirements are significant. Hence, the normalized energy and area efficiencies of the WeJi architecture are comparable to the architectures in [12], [29].

The ASIC implantation results in Table II are from [21], who recently proposed an IIC detector under an implicit method architecture with an ASIC implementation. The architecture achieves a normalized area efficiency of 0.37 Gbps/mm^2, which is lower than that of the proposed architecture. The energy efficiency of the IIC detector is higher than that of the WeJi detector. When the channel frequency is flat and slowly changing, with an obvious channel hardening effect, the results of the preprocessing part of the explicit methods can be reused. In an actual system, when considering the unique property (i.e., channel hardening) of a massive MIMO system, the implicit architecture [21] is required to compute the same Gram matrix T_{c} times, and the explicit architectures (including the proposed architecture and [28]) only need to compute the same Gram matrix one time. For example, when considering typical system parameters in the current LTE-Advanced standard [25], the channel coherence time satisfies $T_{\mathrm{c}} = 7$. Fig. 14-(b) shows the number of multiplications performed under the explicit (WeJi) and implicit (IIC) architectures with $T_{\mathrm{c}} = 7$. The implicit architecture (IIC) [21] suffers from a very high computing load and energy consumptions (approximately T_{c} times) in actual massive MIMO systems. Hence, the energy consumption of the IIC detector increases significantly (approximately T_{c} times). As such, the energy efficiency in the IIC (3.6 Gbps/W) is T_{c} times lower. Therefore, when considering the reusability of the Gram matrix, the energy efficiency of the WeJi detector is higher than that of the implicit IIC architecture [21]. Note that slowly changing channels results in T_{c} times as many buffers to store the channels, which is the limitation of the WeJi architecture.

According to the design, when the numbers of antennas or users increase, by adopting a similar algorithm and architecture design, the PE numbers in Fig. 3, Fig. 6 and Fig. 8 should all be increased correspondingly. For example, for an $N \times M$ MIMO system, the numbers of PE-A, PE-B, PE-C, PE-D, PE-E and PE-F should be M, $\frac{M^2-M}{2}$, $M-1$, $M+1$, M, and $\frac{1}{2}\log_2 \mathcal{Q}$, respectively. If this architecture is to be adopted, the throughput would satisfy (19); in addition, the area and power consumptions would increase according to the numbers of PEs. The PE numbers would also be increased to achieve scalable MIMO systems in recent architectures such as NSA [28], IIC [21], and CHD [12]. Considering the silicon reuse issue, when the numbers of antennas or users increase, reuse can be realized for the chip because the large-scale channel matrix and received vector could be decomposed into a matrix and vector of a lesser scale that can be implemented by this chip. Note that other chips are required for control reasons and for intermediate data storage. Compared with a new silicon implementation, this chip could suffer from only slight throughput and efficiencies losses. Comprehensively considering the efficiencies, time and human effort, it is not necessary to utilize new silicon. The conceptual and silicon reuses can be realized under the proposed architecture for increasing numbers of antennas or users.

VI. Conclusion

This paper proposes an ASIC implementation of a signal detector for a massive MIMO system. A fully pipelined diagonal-based systolic array with single-sided input, a weighted Jacobi iteration unit and an approximate LLR compute architecture are designed. This architecture achieves high energy and area efficiencies. This technology may have applications in future communication technologies such as 5G. Future work will focus on the development of reconfigurable coarse-grain hardware architectures in uplink massive MIMO systems.

Appendix A
Proof of Lemma 1

The spectral radius of $\mathbf{B}_W$ is described as

$$\rho(\mathbf{B}_W) = \rho\left((1-\omega)\mathbf{I} - \omega\mathbf{P}^{-1}\mathbf{Q}\right). \tag{21}$$

In the WeJi method, the parameter ω approaches 1 and satisfies $0 < \omega < 1$ [22], which means that $0 < 1-\omega < 1$. Hence, the $\rho(\mathbf{B}_W)$ in (21) satisfies

$$\rho(\mathbf{B}_W) = \omega\rho(\mathbf{P}^{-1}\mathbf{Q}) - (1-\omega) \leq \omega\rho(\mathbf{P}^{-1}\mathbf{Q}). \tag{22}$$

Because the iteration matrix of the NSA method is $\rho(\mathbf{B}_N) = \rho(\mathbf{P}^{-1}\mathbf{Q})$ [9], combining with (22), the Frobenius-norm of $\mathbf{B}$ in the proposed method satisfies $\rho(\mathbf{B}_W) \leq \omega\rho(\mathbf{B}_N)$. ■

Appendix B
Proof of Lemma 2

Inspired by [9], according to Markov's inequality, there are

$$\begin{aligned}\Pr\{\|\mathbf{B}_W\|_F < 1\} &\geq 1 - \Pr\{\|\mathbf{B}_W\|_F \geq 1\} \\ &\geq 1 - \mathbb{E}(\|\mathbf{B}_W\|_F).\end{aligned} \tag{23}$$

Note that in the WeJi method, the parameter ω approaches 1 and satisfies $0 < \omega < 1$ [22]; hence, $0 < 1-\omega < 1$, which is very small. Now, the influence of $(1-\omega)\mathbf{I}$ can be omitted, which indicates that the probability in (14) can almost satisfy

$$\Pr\{\|\mathbf{B}_W\|_F < 1\} \geq 1 - \mathbb{E}\left(\left\|\omega\mathbf{P}^{-1}\mathbf{Q}\right\|_F\right). \tag{24}$$

Hence, the value of $\Pr\{\|\mathbf{B}_W\|_F < 1\}$ is related to the value of $\mathbb{E}\left(\left\|\omega\mathbf{P}^{-1}\mathbf{Q}\right\|_F\right)$. Next, for $\left\|\mathbf{P}^{-1}\mathbf{Q}\right\|_F$, each element on the i-th row and j-th column of the matrix $\mathbf{A}$ can be described as

$$a_{ij} \rightarrow \begin{cases} \sum_{t=1}^{N} h_{ti}^{*} h_{tj}, & i \neq j \\ \sum_{t=1}^{N} |h_{ti}|^2 + N_0 E_{\mathrm{s}}^{-1}, & i = j. \end{cases} \tag{25}$$

Thus, the $\mathbb{E}\left(\left\|\omega\mathbf{P}^{-1}\mathbf{Q}\right\|_F\right)$ can now be described as

$$\begin{aligned}\mathbb{E}&\left(\left\|\omega\mathbf{P}^{-1}\mathbf{Q}\right\|_F\right)\\ &=\mathbb{E}\left(\sqrt{\sum_{i=1}^{M}\sum_{j=1,i\neq j}^{M}\left|\omega\cdot\frac{a_{ij}}{a_{ii}}\right|^2}\right)\\ &=\mathbb{E}\left(\sqrt{\left(\sum_{i=1}^{M}\sum_{j=1,i\neq j}^{M}\left(\omega^2\cdot|a_{ij}|^2\cdot|a_{ii}|^{-2}\right)\right)^2}\right).\end{aligned}\quad(26)$$

Applying the Cauchy-Schwarz inequality, (26) can now be described as

$$\begin{aligned}\mathbb{E}&\left(\left\|\omega\mathbf{P}^{-1}\mathbf{Q}\right\|_F\right)\\ &\leq\omega\sqrt{\sum_{i=1}^{M}\sum_{j=1,i\neq j}^{M}\mathbb{E}\left(|a_{ij}|^4\right)\cdot\sum_{i=1}^{M}\mathbb{E}\left(|a_{ii}|^{-4}\right)}.\end{aligned}\quad(27)$$

Note the two critical values: $\mathbb{E}\left(|a_{ij}|^4\right)$ and $\mathbb{E}\left(|a_{ii}|^{-4}\right)$. In massive MIMO systems, the diagonal elements a_{ii} from (25) can be well approximated by N [1]. Hence, $\mathbb{E}\left(|a_{ii}|^{-4}\right)$ can be described as

$$\mathbb{E}\left(|a_{ii}|^{-4}\right)=\frac{1}{N^4}.\quad(28)$$

The following details are related to $\mathbb{E}\left(|a_{ij}|^4\right)$. According to (25), $\mathbb{E}\left(|a_{ij}|^4\right)$ can now be described as

$$\begin{aligned}\mathbb{E}\left(|a_{ij}|^4\right)=\mathbb{E}\left(\left|\sum_{t=1}^{N}h_{ti}^*h_{tj}\right|^4\right)&=\sum_{\substack{q_1+q_2\\+\cdots+q_N=N}}\binom{N}{q_1,q_2,\cdots,q_N}\\ &\times\mathbb{E}\left(\prod_{1\leq t\leq N}\left(h_{ti}^*h_{tj}\right)^{q_t}\right),\end{aligned}\quad(29)$$

which can be decomposed to effectively compute $\mathbb{E}\left(|a_{ij}|^4\right)$. Next, $\mathbf{X}=[h_{1i}^*h_{1j},h_{2i}^*h_{2j},\ldots,h_{Ni}^*h_{Nj}]^T$ and $\mu=[\mu_1,\mu_2,¡,\mu_N]^T$ are imported into the matrix to conveniently describe the properties, where μ_t is the mean of $h_{ti}^*h_{tj}$. Therefore, the N-dimensional normal distribution, with a joint probability density function, can be described as

$$\varphi\left(h_{1i}^*h_{1j},h_{2i}^*h_{2j},...,h_{Ni}^*h_{Nj}\right)=\frac{e^{-\frac{(\mathbf{X}-\mu)^T\mathbf{C}^{-1}(\mathbf{X}-\mu)}{2}}}{(2\pi)^{\frac{N}{2}}(\det\mathbf{C})^{\frac{1}{2}}},\quad(30)$$

in which the matrix $\mathbf{C}=(C_{ij})$ is the covariance matrix. Note that each element of the flat Rayleigh fading channel matrix $\mathbf{H}$ is independent and identically distributed (i.i.d.) following $\mathbb{N}(0,1)$, which means that h_{ti}^* is independent of h_{tj} when $i\neq j$ and that $h_{ti}^*h_{tj}$ is i.i.d. following $\mathbb{N}(0,1)$. Therefore, $h_{ti}^*h_{tj}$ and $h_{pi}^*h_{pj}$ for $t\neq p$ are also independent and follow $\mathbb{N}(0,1)$. Now, (30) can be simplified as

$$\varphi\left(h_{1i}^*h_{1j},h_{2i}^*h_{2j},...,h_{Ni}^*h_{Nj}\right)=\frac{1}{(2\pi)^{\frac{N}{2}}}e^{-\frac{\mathbf{X}^T\mathbf{X}}{2}},\quad(31)$$

which indicates that $h_{ti}^*h_{tj}h_{pi}^*h_{pj}$ for $t\neq p$ follows $\mathbb{N}(0,1)$ and that $(h_{ti})^2$ follows $\chi^2(1)$. Hence, the probability density function of the random variable $(h_{ti})^2$ can be described as

$$f(h_{ti};1)=\begin{cases}\dfrac{1}{2\Gamma\left(\frac{1}{2}\right)}\left(\dfrac{h_{ti}}{2}\right)^{-\frac{1}{2}}e^{-\frac{h_{ti}}{2}}, & h_{ti}>0\\ 0, & h_{ti}\leq 0,\end{cases}\quad(32)$$

where Γ is the gamma function. Here, we have $\mathbb{E}(|h_{ti}|^2)=1$ and $\mathbb{D}(|h_{ti}|^2)=2$. The expressions $\mathbb{E}(|h_{ti}|^2)=\mathbb{D}(|h_{ti}|^2)+[\mathbb{E}(|h_{ti}|^2)]^2=3$, $\mathbb{E}(|h_{ti}^*h_{tj}|^2)=\mathbb{E}(|h_{ti}^*|^2)\mathbb{E}(|h_{tj}|^2)=1$ and $\mathbb{E}(|h_{ti}^*h_{tj}|^4)=\mathbb{E}(|h_{ti}^*|^4)\mathbb{E}(|h_{tj}|^4)=9$ can now be derived because h_{ti}^* is independent of h_{tj} when $i\neq j$. So, by removing the zero terms, $\mathbb{E}\left(|a_{ij}|^4\right)$ in (29) can be computed as

$$\begin{aligned}\mathbb{E}\left(|a_{ij}|^4\right)&=N\mathbb{E}\left(\left(h_{ti}^*h_{tj}\right)^4\right)+\binom{N}{2}\left(\mathbb{E}\left(\left(h_{ti}^*h_{tj}\right)^2\right)\right)^2\\ &=\frac{1}{2}\left(N^2+17N\right).\end{aligned}\quad(33)$$

Finally, substituting (28) and (33) into (24) and (27), (14) in Lemma 2 can be obtained. ■

REFERENCES

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

[2] P. Harris *et al.*, "Performance characterization of a real-time massive mimo system with los mobile channels," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 6, pp. 1244–1253, Jun. 2017.

[3] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[4] C.-H. Liao, T.-P. Wang, and T.-D. Chiueh, "A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications," *IEEE J. Solid-State Circuits*, vol. 45, no. 2, pp. 411–421, Feb. 2010.

[5] M. Shabany and P. G. Gulak, "A 0.13 μm CMOS 655 Mb/s 4×4 64-QAM K-best MIMO detector," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2009, pp. 256–257.

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

[7] S. Jacobsson, G. Durisi, M. Coldrey, U. Gustavsson, and C. Studer, "Throughput analysis of massive MIMO uplink with low-resolution ADCs," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 4038–4051, Jun. 2017.

[8] S. Ozyurt and M. Torlak, "Exact joint distribution analysis of zero-forcing V-BLAST gains with greedy ordering," *IEEE Trans. Wireless Commun.*, vol. 12, no. 11, pp. 5377–5385, Dec. 2012.

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

[10] B. Yin, M. Wu, G. Wang, C. Dick, J. R. Cavallaro, and C. Studer, "A 3.8 Gb/s large-scale MIMO detector for 3GPP LTE-Advanced," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Florence, Italy, May 2014, pp. 3879–3883.

[11] D. Auras, R. Leupers, and G. H. Ascheid, "A novel reduced-complexity soft-input soft-output MMSE MIMO detector: Algorithm and efficient VLSI architecture," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 4722–4728.

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

[13] L. Dai *et al.*, "Low-complexity soft-output signal detection based on Gauss–Seidel method for uplink multi-user large-scale MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 64, no. 10, pp. 4839–4845, Oct. 2015.

[14] 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)*, Montreal, QC, Canada, May 2016, pp. 1886–1889.

[15] X. Gao, L. Dai, Y. Ma, and Z. Wang, "Low-complexity near-optimal signal detection for uplink large-scale MIMO systems," *Electron. Lett.*, vol. 50, no. 18, pp. 1326–1328, Sep. 2014.

[16] X. Gao, L. Dai, Y. Hu, and Z. Wang, "Matrix inversion-less signal detection using SOR method for uplink large-scale MIMO systems," in *Proc. IEEE Global Telecommun. Conf. (IEEE GLOBECOM)*, San Diego, CA, USA, Dec. 2015, pp. 3291–3295.

[17] P. Zhang, L. Liu, G. Peng, and S. Wei, "Large-scale MIMO detection design and FPGA implementations using SOR method," in *Proc. 8th IEEE Int. Conf. Commun. Softw. Netw.*, Beijing, China, Jun. 2016, pp. 206–210.

[18] J. Ning, Z. Lu, T. Xie, and J. Quan, "Low complexity signal detector based on SSOR method for massive MIMO systems," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast.*, Ghent, Belgium, Jun. 2015, pp. 1–4.

[19] B. Yin, M. Wu, J. R. Cavallaro, and C. Studer, "VLSI design of large-scale soft-output MIMO detection using conjugate gradients," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Lisbon, Portugal, May 2015, pp. 1498–1501.

[20] 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.

[21] 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.

[22] D. R. Kincaid and E. W. Cheney, Jr., *Numerical Analysis: Mathematics of Scientific Computing*, 3rd ed. Belmont, CA, USA: Wadsworth, 2002.

[23] B. Y. Kong and I.-C. Park, "Low-complexity symbol detection for massive MIMO uplink based on Jacobi method," in *Proc. IEEE 27th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Valencia, Spain, Sep. 2016, pp. 1–5.

[24] L. Chen, "Iterative soft decoding of reed-solomon convolutional concatenated codes," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4076–4085, Oct. 2013.

[25] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (EUTRA); Physical Layer Procedures (Release 10)*, document 3GPP Organizational Partners TS 36.213 version 10.10.0, Jul. 2013.

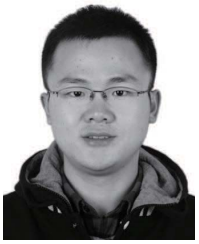
[26] B. E. Godana and T. Ekman, "Parametrization based limited feedback design for correlated MIMO channels using new statistical models," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 5172–5184, Oct. 2013.

[27] O. Castañeda, T. Goldstein, and C. Studer, "Data detection in large multi-antenna wireless systems via approximate semidefinite relaxation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2334–2346, Dec. 2016.

[28] B. Yin, "Low complexity detection and precoding for massive MIMO systems: Algorithm, architecture, and application," Ph.D. dissertation, Dept. Electr. Comput. Eng., Rice Univ., Houston, TX, USA, 2014.

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

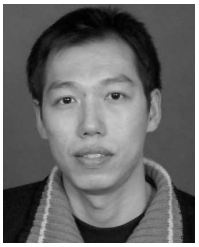
[30] G. Steffan *et al.*, "A 64 Gb/s PAM-4 transmitter with 4-tap FFE and 2.26 pJ/b energy efficiency in 28 nm CMOS FDSOI," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2017, pp. 66–67.



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

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

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



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

Shaojun Wei (M'91) was born in Beijing, China, in 1958. He received the Ph.D. degree from the Faculte Polytechnique de Mons, Belguim, in 1991. He became a Professor with the Institute of Microelectronics of Tsinghua University in 1995. He is currently a Senior Member with the Chinese Institute of Electronics. His main research interests include VLSI SoC design, EDA methodology, and communication ASIC design.