A Low-Cost MMSE-SIC Detector for the MIMO System: Algorithm and Hardware Implementation

Tsung-Hsien Liu, Jin-Yi Jiang, and Yuan-Sun Chu

***Abstract*—We consider the minimum mean-squared error with successive interference cancellation (MMSE-SIC) detection of a frame of data in the spatially multiplexed multiple-input–multiple-output (MIMO) system. A complete MMSE-SIC detector needs to compute at both the preprocessing and SIC-detection stages. Since the SIC detection, which can be regarded as a backward substitution, is inevitable, we develop a computationally efficient preprocessing algorithm that relies on the backward substitution. We then propose a low-cost hardware architecture with a commonly shared backward-substitution module to work at both the preprocessing and detection stages. The very-large-scale-integration implementation results of our architecture for the four-by-four MIMO system using the 0.18-μm complementary metal–oxide–semiconductor technology reveal that our architecture requires the fewest 79-K gates, provides the high throughput rate of 416 Mb/s, and works with the smallest preprocessing latency of 64 clock cycles. Our MMSE-SIC detector is a cheap solution for MIMO detection.**

***Index Terms*—Bell Laboratories layered space–time (BLAST) architecture, minimum mean-squared error (MMSE) detection, multiple-input–multiple-output (MIMO) system, spatially multiplexed multiple-input–multiple-output (MIMO) system, successive interference cancellation (SIC).**

I. Introduction

As IN [1]–[3], we consider the detection of a frame of data by the minimum mean-squared error with successive interference cancellation (MMSE-SIC) detector in the spatially multiplexed multiple-input–multiple-output (MIMO) system. Each frame of data contains a preamble followed by its payload. The MIMO receiver utilizes the preamble to estimate its channel state information. The MMSE-SIC detector follows to perform two different stages of computation. At the preprocessing stage, the MMSE-SIC detector uses the channel state information to compute a feedforward matrix and a feedback matrix that are associated with the optimal detection order. Then, at the detection stage, the MMSE-SIC detector applies the SIC procedure to each received sample vector in the payload to recover the transmitted bits.

Reported algorithms for the MMSE-SIC detection mainly consider the fast computation performed at the preprocessing stage. These preprocessing algorithms roughly fall into two categories. The first category of algorithms [2], [4], [5] are based on the QR decomposition of the augmented channel matrix. The second category of algorithms [3], [6], [7] rely on multiplications and divisions; they require fewer multiplications than those algorithms in [2], [4], and [5]. In contrast, the implementation algorithm executed at the detection stage of the MMSE-SIC detector is simple and straightforward.

Hardware architectures for realizing the MMSE-SIC detection are studied. The architectures in [8]–[11] follow from the QR decomposition algorithms in [2] and [4]; thus, they are concentrated on the computation for the preprocessing stage. Their MMSE-SIC detectors require a separate backward-substitution module to perform the SIC procedure at the detection stage in order to fulfill the complete detection of a frame of data.

To our knowledge, no architecture based on the fastest preprocessing algorithms in [3] and [7] is proposed. We propose in this brief a variant algorithm of our modified recursive fast algorithm [3]. This new preprocessing algorithm is comprised of multiple backward/forward substitutions; more importantly, it requires as many multiplications as the fastest preprocessing algorithms do in [3] and [7]. Also, recall that the inevitable SIC procedure at the detection stage of the MMSE-SIC detector can be regarded as a backward substitution. We therefore have the preprocessing and detection algorithms both relying mainly on the backward substitution. Furthermore, we propose a hardware architecture with a backward-substitution module as the core module. Since the preprocessing and detection stages do not overlap in time, the backward-substitution module independently works at either stage. Compared with the architectures in [8]–[11], where separate preprocessing and SIC-detection modules are used, our proposed architecture contains a shared backward-substitution module and can be therefore implemented with fewer 79-K gates for the four-by-four MIMO system. Our architecture also works with the short preprocessing latency of 64 clock cycles and provides the high detection throughput rate of 416 Mb/s.

Recently, hardware architectures for the sphere decoder [12] and the K-best decoder [13]–[15] seem to be attractive. These architectures are concentrated only on the computation at the detection stage. The average computational complexity required by the sphere decoder to process one sample vector in the payload is roughly proportional to M^3 [16], where M is the number of transmit antennas. In comparison, the number of multiplications required by the SIC detection to process one sample vector is proportional to M^2 [2], [3]. The MMSE-SIC detector is important for being able to provide a cheaper detector than either the sphere decoder or the K-best detector. Either the sphere decoder or the K-best decoder is appropriate for the scenario where the number of transmit antennas is small

Manuscript received March 12, 2010; revised May 31, 2010 and July 30, 2010; accepted October 22, 2010. Date of current version January 19, 2011. This work was supported by the National Science Council, Taiwan, under Contract NSC97-2221-E-194-033 and Contract NSC98-2221-E-194-019-MY2. This work was recommended by Associate Editor Z. Wang.

T.-H. Liu and J.-Y. Jiang are with the Department of Communications Engineering, National Chung Cheng University, Chia-Yi 62101, Taiwan (e-mail: comtsliu@ccu.edu.tw; sp967673@gmail.com).

Y.-S. Chu is with the Department of Electrical Engineering, National Chung Cheng University, Chia-Yi 62101, Taiwan (e-mail: chu@ee.ccu.edu.tw).

Digital Object Identifier 10.1109/TCSII.2010.2092819

and the number of the constellation size of transmitted symbols is small [12]. The MMSE-SIC detector is also important because it can be used by the sphere decoder to find the initial lattice point to determine the initial radius at the detection stage. The proper initial point and radius are helpful in reducing the complexity of the sphere decoder [12], [14], [16]. Compared with the architectures in [12]–[15], our proposed architecture in this brief is a complete solution, which includes both the computations at the preprocessing and detection stages, for the MIMO detection.

The organization of this brief is as follows. We introduce our preprocessing and detection algorithms for the MMSE-SIC detector in Section II. We then propose our hardware architecture for the MMSE-SIC detector in Section III. The very-large-scale-integration (VLSI) implementation results of our proposed architecture and comparisons with other architectures are reported in Section IV. Conclusions are drawn in Section V.

Notations $(\ )^*$, $(\ )^H$, $(\ )^T$, and $(\ )^{-1}$ are used to denote the conjugate, Hermitian, transpose, and matrix inverse operations, respectively. Notation $\mathbf{I}_M$ denotes the $M \times M$ identity matrix. Notation $\mathbb{E}\{\cdot\}$ denotes the expectation operation.

II. MMSE-SIC Detection

Consider a spatially multiplexed MIMO system with M transmit and N receive antennas, $N \geq M$, in a flat-fading environment. The received signal of the jth antenna at time instant k is

$$x_j(k) = \sum_{i=1}^{M} H_{j,i} s_i(k) + w_j(k), \qquad j = 1, 2, \ldots, N \quad (1)$$

where $H_{j,i}$ denotes the complex-valued channel gain between the ith transmit and the jth receive antennas, $s_i(k)$ denotes the symbol transmitted from the ith transmit antenna, and $w_j(k)$ denotes the white Gaussian noise with variance σ^2 at the jth receive antenna. The transmitted symbols are normalized with $\mathbb{E}\{|s_i(k)|^2\} = 1$. By defining $\mathbf{x}(k) \triangleq [\, x_1(k)\ x_2(k)\ \cdots\ x_N(k)\,]^T$, $\mathbf{s}(k) \triangleq [\, s_1(k)\ s_2(k)\ \cdots\ s_M(k)\,]^T$, $\mathbf{w}(k) \triangleq [\, w_1(k)\ w_2(k) \cdots w_N(k)\,]^T$, $\mathbf{h}_i \triangleq [H_{1,i}\ H_{2,i}\ \cdots\ H_{N,i}]^T$, and $\mathbf{H}_M \triangleq [\, \mathbf{h}_1 \quad \mathbf{h}_2 \quad \cdots \quad \mathbf{h}_M\,]$, the signal model in vector form becomes

$$\mathbf{x}(k) = \mathbf{H}_M \mathbf{s}(k) + \mathbf{w}(k). \quad (2)$$

The MMSE-SIC detector needs to compute at the preprocessing stage and then at the detection stage. Our preprocessing algorithm for the MMSE-SIC detector is listed as follows. This algorithm is obtained by modifying the algorithm reported in [3].

1) Compute $\mathbf{R}_M = \mathbf{H}_M^H \mathbf{H}_M + \sigma^2 \mathbf{I}_M$ through direct multiplications and accumulations.
2) Let $\mathbf{L}_1 = [1]$ and $\mathbf{D}_1$ be equal to the (1, 1)th entry of $\mathbf{R}_M$.
3) Repeat steps 4–6 for $m = 1, 2, \ldots, M-1$.
4) Let $\mathbf{r}_{m+1}$ be the vector that contains the $(1, m+1)$th down to the $(m, m+1)$th entries of $\mathbf{R}_M$, and let r_{m+1} be equal to the $(m+1, m+1)$th entry of $\mathbf{R}_M$.
5) Solve the lower triangular system, i.e.,

$$\mathbf{L}_m \mathbf{D}_m \boldsymbol{\ell}_{m+1} = \mathbf{r}_{m+1} \quad (3)$$

for $\boldsymbol{\ell}_{m+1}$ through forward substitution, and compute d_{m+1} by

$$d_{m+1} = r_{m+1} - \boldsymbol{\ell}_{m+1}^H \mathbf{D}_m \boldsymbol{\ell}_{m+1}. \quad (4)$$

6) Update $\mathbf{L}_{m+1}$ and $\mathbf{D}_{m+1}$ by

$$\mathbf{L}_{m+1} = \begin{bmatrix} \mathbf{L}_m & \mathbf{0} \\ \boldsymbol{\ell}_{m+1}^H & 1 \end{bmatrix} \quad (5)$$

and

$$\mathbf{D}_{m+1} = \mathrm{diag}\{\mathbf{D}_m, d_{m+1}\} \quad (6)$$

respectively.
7) Repeat steps 8 and 9 for $j = 1, 2, \ldots, M$.
8) Solve the triangular system, i.e.,

$$\mathbf{L}_M \mathbf{y}_j = \mathbf{e}_j \quad (7)$$

for $\mathbf{y}_j$ through forward substitution, where $\mathbf{e}_j$ denotes the $M \times 1$ vector with 1 in its jth element and with 0's elsewhere.
9) Solve the triangular system, i.e.,

$$\mathbf{D}_M \mathbf{L}_M^H \mathbf{q}_j = \mathbf{y}_j \quad (8)$$

for $\mathbf{q}_j$ through backward substitution.
10) Construct matrix $\mathbf{Q}_M \triangleq [\, \mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_M\,]$. Let the index vector $\mathbf{f}_M \triangleq [\, 1 \quad 2 \quad \cdots \quad M\,]^T$, $\tilde{\mathbf{R}}_M = \mathbf{R}_M$, and $\tilde{\mathbf{H}}_M = \mathbf{H}_M$.
11) Repeat steps 12–15 for $m = 1, 2, \ldots, M-1$.
12) Find the minimum value on the diagonal of $\mathbf{Q}_{M-m+1}$; let the (κ_m, κ_m)th entry be of the minimum value. Permutations are applied to $\mathbf{f}_{M-m+1}$ as in

$$\begin{bmatrix} \mathbf{f}_{M-m} \\ p_m \end{bmatrix} = \mathbf{J}_m \mathbf{f}_{M-m+1} \quad (9)$$

where $\mathbf{J}_m$ is the permutation matrix that is associated with interchanging the κ_mth and $(M-m+1)$th rows or columns of a matrix.
13) Partition the permuted $\mathbf{Q}_{M-m+1}$ as in

$$\begin{bmatrix} \tilde{\mathbf{Q}}_{M-m} & \tilde{\mathbf{q}}_{M-m} \\ \tilde{\mathbf{q}}_{M-m}^H & \tilde{q}_{M-m} \end{bmatrix} = \mathbf{J}_m \mathbf{Q}_{M-m+1} \mathbf{J}_m \quad (10)$$

where $\tilde{\mathbf{Q}}_{M-m}$, $\tilde{\mathbf{q}}_{M-m}$, and $\tilde{q}_{M-m}$ are of sizes $(M-m) \times (M-m)$, $(M-m) \times 1$, and 1×1, respectively.
14) Compute $\mathbf{Q}_{M-m}$ by

$$\mathbf{Q}_{M-m} = \tilde{\mathbf{Q}}_{M-m} - \frac{\tilde{\mathbf{q}}_{M-m} \tilde{\mathbf{q}}_{M-m}^H}{\tilde{q}_{M-m}}. \quad (11)$$

15) Interchange the κ_mth and $(M-m+1)$th columns and rows of $\tilde{\mathbf{R}}_{M-m+1}$, and denote the permuted $M \times M$ matrix by $\tilde{\mathbf{R}}_{M-m}$. Interchange the κ_mth and $(M-m+1)$th columns of $\tilde{\mathbf{H}}_{M-m+1}$, and denote the permuted matrix by $\tilde{\mathbf{H}}_{M-m}$.
16) Let $p_M = \mathbf{f}_1$.
17) Perform a LDL^H decomposition to $\tilde{\mathbf{R}}_1$ as in steps 2–6 to obtain the unit upper triangular $\tilde{\mathbf{U}}$ and diagonal $\tilde{\mathbf{\Sigma}}$, i.e., $\tilde{\mathbf{R}}_1 = \tilde{\mathbf{U}}^H \tilde{\mathbf{\Sigma}} \tilde{\mathbf{U}}$.
18) Solve the upper triangular system $\tilde{\mathbf{\Sigma}} \tilde{\mathbf{U}} \tilde{\mathbf{G}}^H = \tilde{\mathbf{H}}_1^H$ for each column of $\tilde{\mathbf{G}}^H$. $\tilde{\mathbf{G}}$ is obtained.

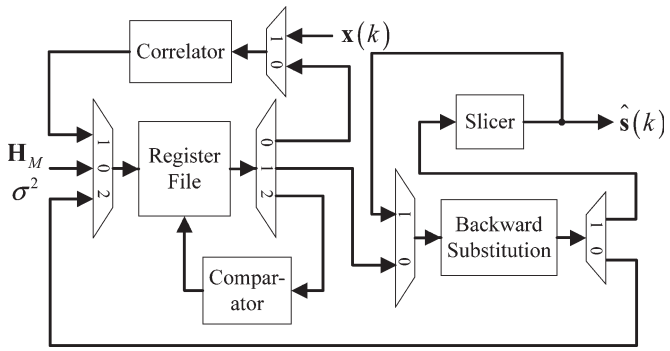

Fig. 1. Our proposed system architecture for the MMSE-SIC detection with optimal detection order in a four-by-four MIMO system.

When the above preprocessing algorithm ends, a feedforward matrix $\tilde{\mathbf{G}}$ and a feedback matrix $\tilde{\mathbf{U}}$ that are associated with the optimal detection order $\{p_1, p_2, \ldots, p_M\}$ are obtained. Note that $s_{p_1}(k)$ is to be detected first, and $s_{p_M}(k)$ is to be detected the last. Steps 2–6 are for the LDL^H decomposition of $\mathbf{R}_M$, i.e., $\mathbf{R}_M = \mathbf{L}_M \mathbf{D}_M \mathbf{L}_M^H$. Steps 7–9 are for computing the inverse of $\mathbf{R}_M$ (i.e., $\mathbf{Q}_M$) by utilizing the already computed $\mathbf{L}_M$ and $\mathbf{D}_M$. The number of complex-valued multiplications required by this preprocessing algorithm is $NM^2 + (5/6)M^3 + O(NM) + O(M^2)$. This algorithm is as computationally efficient as those in [3] and [7]. However, it is different in the fact that the forward and backward substitutions are used in steps 5, 8, 9, 17, and 18.

At the detection stage, the following SIC detection procedure [3] is applied to each received vector $\mathbf{x}(k), \forall k$, in the payload for the detection of $\{s_{p_1}(k), s_{p_2}(k), \ldots, s_{p_M}(k)\}$. The backward substitution is performed in step 5 below, which requires $(1/2)M^2 + O(M)$ multiplications. The linear transformation in steps 1 and 4 requires M^2 multiplications. Assume that $\tilde{\mathbf{G}} = [\,\tilde{\mathbf{g}}_{p_M} \quad \tilde{\mathbf{g}}_{p_{M-1}} \quad \cdots \quad \tilde{\mathbf{g}}_{p_1}\,]$.

1) Compute $z_{p_1}(k) = \tilde{\mathbf{g}}_{p_1}^H \mathbf{x}(k)$.
2) Obtain the first detected symbol by $\hat{s}_{p_1}(k) = \text{slice}(z_{p_1}(k))$.
3) Repeat steps 4 and 5 for $m = 2, 3, \ldots, M$.
4) Compute $z_{p_m}(k) = \tilde{\mathbf{g}}_{p_m}^H \mathbf{x}(k)$.
5) Slice the interference-cancelled output to obtain the detected symbol, i.e., $\hat{s}_{p_m}(k) = \text{slice}(\xi_{p_m}(k))$, where

$$\xi_{p_m}(k) = z_{p_m}(k) - \sum_{j=1}^{m-1} \hat{s}_{p_j}(k) \tilde{U}_{M-m+1, M-m+1+j} \qquad (12)$$

and $\tilde{U}_{M-m+1, M-m+1+j}$ is the $(M-m+1, M-m+1+j)$th entry of $\tilde{\mathbf{U}}$.

III. Hardware Architecture

Our proposed VLSI architecture for the MMSE-SIC detection in a four-by-four MIMO system is illustrated in Fig. 1. This architecture follows from the preprocessing and SIC-detection algorithms in Section II. At the preprocessing stage, our proposed architecture works as follows. Assume that the channel information, including $\mathbf{H}_M$ and σ^2, is stored in the register file. As indicated by step 1 of the preprocessing algorithm, the correlator module computes the correlation matrix $\mathbf{R}_M$, the result of which is stored back in the register file. The backward-substitution module then computes $\mathbf{L}_M$, $\mathbf{D}_M$, and $\mathbf{Q}_M$ according to steps 2–9; the computed $\mathbf{Q}_M$ is stored in the register file. The comparator module is for finding the MMSE as indicated by step 12 of the preprocessing algorithm. The recursive computation of $\mathbf{Q}_{M-m}, \forall m$, in step 14 is performed by the backward-substitution module. The computed ordering information $\{p_1, p_2, \ldots, p_M\}$ is stored in a control unit that is not plotted in Fig. 1. The backward-substitution module is then activated several times to perform the multiple backward/forward substitutions as required by steps 17 and 18. The finally computed feedforward and feedback matrix pair $(\tilde{\mathbf{G}}, \tilde{\mathbf{U}})$ is stored in the register file. Our register file provides a memory storage for values $\mathbf{H}_M$, σ^2, $\mathbf{R}_M$, $\mathbf{Q}_M$, $\mathbf{L}_M$, and $\mathbf{D}_M$; the finally computed $\tilde{\mathbf{\Sigma}}$, $\tilde{\mathbf{U}}$, and $\tilde{\mathbf{G}}$ overwrite the intermediate values $\mathbf{D}_M$, $\mathbf{L}_M$, and $\mathbf{Q}_M$, respectively. Each of the 14 bits is used to represent a real-valued number, and this register file provides a memory storage of 1.69-K bits. Additionally, the permutation of vectors and matrices and the signaling between all modules are controlled by the control unit.

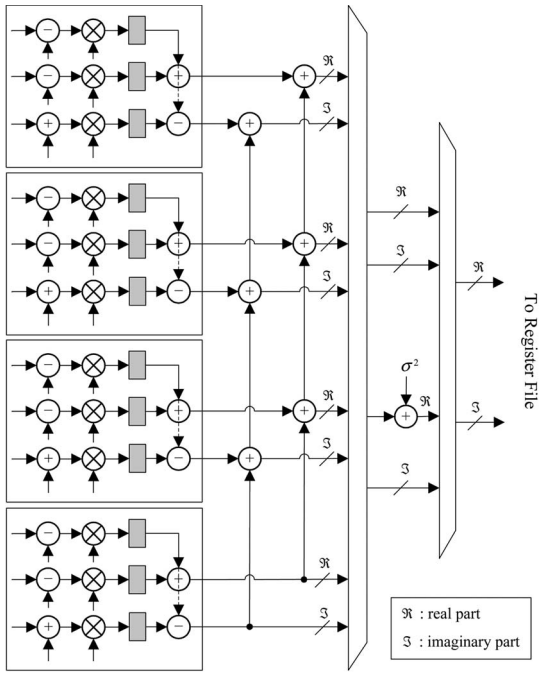

Fig. 2. Hardware architecture of the correlator module.

At the detection stage, for each of the sample vector $\mathbf{x}(k)$ in the payload, the correlator module computes the linearly transformed $z_{p_m}(k), \forall m$, as in steps 1 and 4 of the SIC-detection procedure. Each $z_{p_m}(k)$ is produced in a clock cycle and is then immediately passed to the backward-substitution module to produce output $\xi_{p_m}(k), \forall m$. The slicer module performs the slice$(\cdot)$ operation in steps 2 and 5; it can demodulate the binary phase-shift keying, quadrature phase-shift keying (QPSK), 16 quadrature amplitude modulation (QAM), and 64 QAM signals. Thus, four clock cycles are required by our architecture to process one sample vector $\mathbf{x}(k)$ and to yield the output vector $[\hat{s}_{p_M}(k), \hat{s}_{p_{M-1}}(k), \ldots, \hat{s}_{p_1}(k)]^T$. A very important fact is that both the correlator and backward-substitution modules work at the preprocessing and detection stages.

The correlator module, which is depicted in Fig. 2, is comprised of 12 real-valued multipliers and 27 real-valued adders. The three real-valued multipliers to perform a complex-valued multiplication are suggested by [17] for reducing the silicon

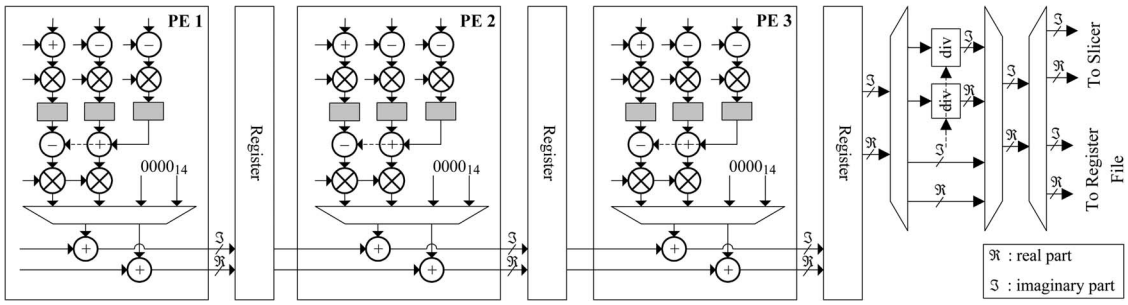

Fig. 3. Hardware architecture of the backward-substitution module.

area. When working at the preprocessing stage, this correlator module requires ten clock cycles to produce the upper triangular part of $\mathbf{R}_M$ in step 1 of the preprocessing algorithm. When working at the detection stage, it also requires four clock cycles to produce each $z_{p_m}(k)$, as indicated by steps 1 and 4 of the SIC-detection algorithm.

The backward-substitution module, the detailed architecture of which is illustrated in Fig. 3, executes the multiple forward/backward substitutions required by the preprocessing and SIC-detection algorithms. This module consists of three cascaded processing elements (PEs). Each PE includes three real-valued multipliers and five adders, as in the correlator module, to perform a complex-valued multiplication. The two additional real-valued multipliers in each PE are associated with the computation indicated by (3) and (4). The two real-valued two-stage pipelined dividers on the right part of Fig. 3 are associated with the divisions in (3), (8), and (11). The dividers work only at the preprocessing stage; they do not work at the detection stage. Clearly, our backward-substitution module is more complicated than the one in [8]. In addition, the correlator module is a parallel architecture, and this backward-substitution module is a multistage architecture. The detection throughput of our proposed architecture in Fig. 1 largely depends on this backward-substitution module. To overcome the throughput bottleneck, the three-stage pipelined operations are used in Fig. 3. Our design of the architectures in Figs. 2 and 3 are only for the four-by-four MIMO system. The architecture can be easily extended to the detection in the M-by-M MIMO system with $M > 4$.

IV. Implementation Results

The architecture of Fig. 1 was described in Verilog hardware description language at a register-transfer level and was synthesized by the Synopsys Design Compiler. We conducted a fixed-point simulation for our design, the results of which are given in Fig. 4. Each real-valued number in our computation was represented by a 14-bit data word. The architecture was implemented in a Taiwan Semiconductor Manufacturing Company 0.18-μm complementary metal–oxide–semiconductor technology, the results of which are compared with several architectures in Table I. The layout of our architecture is also illustrated in Fig. 5.

The architectures for the MMSE-SIC detector studied in [8]–[11] consider the computations at both the preprocessing and detection stages. They follow from the QR-decomposition-based preprocessing algorithms in [2] and [4]. Each architecture includes a coordinate-rotation-digital-computer supercell,

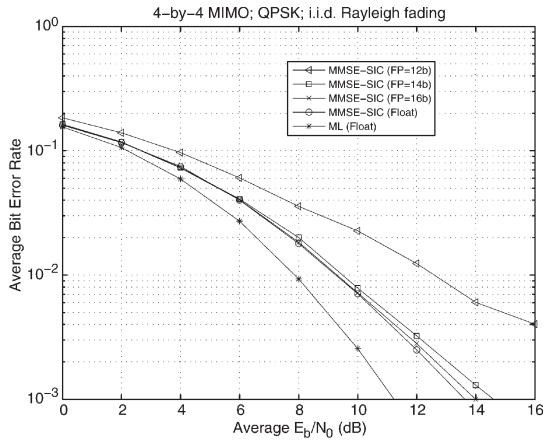

Fig. 4. Fixed-point simulation results of our proposed architecture for the four-by-four MIMO system with transmitted QPSK signals.

together with few multipliers, to execute their preprocessing computations; each architecture also includes a separate module to perform the detection-stage computations. Obviously, the architectures relying on the SIC-detection procedure work with the detection latency equal to four clock cycles, which is equal to the number of transmit antennas. Also, our architecture outperforms the architectures in [8]–[11] because it can work with a clock rate of 208 MHz and can provide a detection throughput rate of 416 Mb/s. The detection throughput rate is defined by the following formula [18]:

$$\text{Throughput} = \frac{\log_2(\text{const. size})}{\text{detection latency}}(\text{max. clk. rate})(\#\text{ of Tx Ant.}).$$

Note that the detection throughput is independent of the preprocessing latency; it is an important performance measure for an architecture working at the detection stage. We also observe that our proposed architecture works with the smallest preprocessing latency; this fact is not surprising because our architecture follows from the fastest preprocessing algorithm given in Section II. A detector with a small preprocessing latency is advantageous that only a small buffer is required for saving the incoming payload vectors [11], [20]. Finally, our architecture requires the fewest 79-K gates due to the commonly shared correlator and backward-substitution modules that work at both the preprocessing and detection stages.

We also include the implementation results of other architectures [12]–[14], [18] in Table I for comparison. The detection schemes adopted by these architectures are the sphere decoder and the K-best decoder, which perform with lower

TABLE I
IMPLEMENTATION RESULTS OF OUR ARCHITECTURE AND OTHER ARCHITECTURES FOR A FOUR-BY-FOUR MIMO SYSTEM. THE ACRONYM NA IN THE TABLE STANDS FOR INFORMATION NOT AVAILABLE

Architecture	Proposed	[8] and [9]	[10] and [11]	[18]	[13]	[14]	[12]
Hardware platform	ASIC	ASIC	ASIC	ASIC	ASIC	ASIC	ASIC
Detection Scheme	MMSE-SIC with optimal order	MMSE-SIC with optimal order	MMSE-SIC with suboptimal order	V-BLAST with ML	K-best decoding	K-best decoding	Sphere decoding
Modulation	QPSK	QPSK	QPSK	64-QAM	64-QAM	16-QAM	16-QAM
Equivalent Gates (K)	79	190	92	205 +preproc.	114 +preproc.	91 +preproc.	50 +preproc.
Max. Clock Rate (MHz)	208	80	80	71	282	100	71
Preprocessing Latency (Clock Cycles)	64	≤ 640	80	NA	NA	NA	NA
Detection Latency (Clock Cycles)	4	4	4	15	10	30	NA
Detection Throughput (Mbps)	416	160	160	114	675	53.3	169 (at 20 dB)
CMOS Technology (μm)	0.18	0.35	0.25	0.13	0.13	0.13	0.25

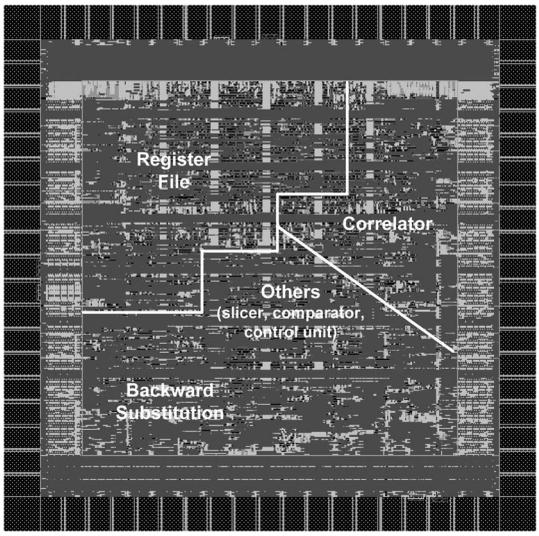

Fig. 5. Layout of our proposed architecture in Fig. 1.

bit-error rate than the MMSE-SIC detector. These architectures do not consider the preprocessing computation; therefore, their preprocessing latency is unavailable. Compared with these architectures, our proposed architecture is a complete solution, which includes computations at both the preprocessing and detection stages. Clearly, our architecture provides a cheap solution to the MIMO detection; the advantages of our solution include few gates, short preprocessing latency, and high detection throughput rate. The concept of hardware sharing for the MMSE detector was reported in [19].

V. CONCLUSION

Computationally efficient algorithms reported in [3] and [7] have been demonstrated to require the fewest complex-valued multiplications for the preprocessing of the MMSE-SIC detector. How to organize the multiplications properly to be performed by hardware architectures is a main challenge for the hardware implementation of the MMSE-SIC detector. In this brief, we have proposed a computationally efficient detection algorithm that uses backward substitution for both the preprocessing and the SIC detection of the MMSE-SIC detector. This new algorithm is suitable for hardware implementation. Unlike other hardware architectures where the preprocessing and the SIC detection are treated separately, our proposed architecture features commonly shared correlator and backward-substitution modules for the preprocessing and the SIC detection. Implementation comparisons in Table I have revealed that our architecture requires the fewest gates, provides high throughput rate, and works with the smallest preprocessing latency. The small preprocessing latency implies that only a small buffer is required for our MMSE-SIC detector to buffer incoming payload vectors [11], [20].

Our contribution in this brief is to provide a low-complexity hard-output MMSE-SIC detector for the general spatial multiplexed MIMO system [1], [2]. One future work is to study the soft-output MMSE-SIC detection [21], [22] to provide a better detection performance. We may also apply our architecture to the MIMO orthogonal frequency-division multiplexing system [23], where a special design for the preprocessing and detection computations is required.

REFERENCES

[1] G. Golden, C. Foschini, R. Valenzuela, and P. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space–time communication architecture," *Electron. Lett.*, vol. 35, no. 1, pp. 14–16, Jan. 1999.
[2] B. Hassibi, "An efficient square-root algorithm for BLAST," in *Proc. IEEE ICASSP*, 2000, pp. II737–II740.
[3] T.-H. Liu and Y.-L. Yeh Liu, "Modified fast recursive algorithm for efficient MMSE-SIC detection of the V-BLAST system," *IEEE Trans. Wireless Commun.*, vol. 7, no. 10, pp. 3713–3717, Oct. 2008.
[4] D. Wübben, R. Böhnke, V. Kühn, and K. D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE Veh. Technol. Conf.*, Orlando, FL, Oct. 2003, pp. 508–512.
[5] T.-H. Liu, "Efficient and stable MMSE detection of the V-BLAST system," in *Proc. IEEE GLOBECOM*, San Francisco, CA, Nov. 2006, pp. 1–5.
[6] J. Benesty, Y. Huang, and J. Chen, "A fast recursive algorithm for optimum sequential signal detection in a BLAST system," *IEEE Trans. Signal Process.*, vol. 51, no. 7, pp. 1722–1730, Jul. 2003.
[7] Y. Shang and X.-G. Xia, "On fast recursive algorithms for V-BLAST with optimal ordered SIC detection," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 2860–2865, Jun. 2009.
[8] Z. Guo and P. Nilsson, "A VLSI implementation of MIMO detection for future wireless communications," in *Proc. IEEE Pers., Indoor, Mobile Radio Commun.*, Sep. 2003, pp. 2852–2856.
[9] Z. Guo and P. Nilsson, "An improved MIMO detector for high data rates wireless communications," in *Proc. IEEE Asia-Pac. Conf. Circuits Syst.*, Dec. 2004, pp. 213–216.
[10] P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, "VLSI implementation of a high-speed iterative sorted MMSE QR decomposition," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 1421–1424.
[11] A. Burg, S. Haene, D. Perels, P. Luethi, N. Felber, and W. Fichtner, "Receiver design for multi-antenna wireless communications," in *Proc. Ph.D. Res. Microelectron. Electron.*, Jul. 2005, pp. 35–38.
[12] A. Burg, M. Börgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, Jul. 2005.

[13] M. Shabany and P. G. Gulak, "A 0.13 μm CMOS 655 Mb/s 4×4 64-QAM K-best MIMO detector," in *Proc. IEEE ISSCC Dig. Tech. Papers*, Feb. 2009, pp. 256–257.

[14] 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.

[15] T. H. Im, I. Park, J. Kim, J. Yi, J. Kim, S. Yu, and Y. S. Cho, "A new signal detection method for spatially multiplexed MIMO systems and its VLSI implementation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 5, pp. 399–403, May 2009.

[16] O. Damon, 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.

[17] V. Oklobdzija, D. Villeger, and T. Soulas, "An integrated multiplier for complex numbers," *J. VLSI Signal Process.*, vol. 7, no. 3, pp. 213–222, May 1994.

[18] C.-J. Huang, C.-W. Yu, and H.-P. Ma, "A power-efficient configurable low-complexity MIMO detector," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 56, no. 2, pp. 485–496, Feb. 2009.

[19] A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, "Algorithm and VLSI architecture for linear MMSE detection in MIMO-OFDM system," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 4102–4105.

[20] S. Yoshizawa and Y. Miyanaga, "VLSI implementation of a 4×4 MIMO-OFDM transceiver with an 80-MHz channel bandwidth," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2009, pp. 1743–1746.

[21] C. Manchon, L. Deneire, P. Mogensen, and T. Sorensen, "On the design of a MIMO-SIC receiver for LTE downlink," in *Proc. IEEE Veh. Technol. Conf.*, 2008, pp. 1–5.

[22] T. Im, I. Park, S. Yu, and Y. Cho, "Novel signal detection methods using multiple ordering technique for spatially multiplexed MIMO systems and its VLSI implementation," in *Proc. IEEE Int. Symp. Pers., Indoor, Mobile Radio Commun.*, 2009, pp. 2424–2428.

[23] R.-H. Lai, C.-M. Chen, P.-A. Ting, and Y.-H. Huang, "A modified sorted-QR decomposition algorithm for parallel processing in MIMO detection," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2009, pp. 1405–1408.