

# A Maximum Logarithmic Maximum *a Posteriori* Probability Based Soft-Input Soft-Output Detector for the Coded Spatial Modulation Systems

Tsung-Hsien Liu<sup>1</sup>, Member, IEEE, Ting-Xu Jiang<sup>2</sup>, Ching-Che Chung<sup>3</sup>, Senior Member, IEEE, and Yuan-Sun Chu<sup>4</sup>

**Abstract**—Soft-input soft-output (SISO) detector for the iterative detection and decoding of coded spatial modulation (SM) signals needs to compute the extrinsic bit log-likelihood ratio (LLR) associated with each detected bit. The maximum logarithmic maximum a posteriori (max-log-MAP) detector is known to perform closely to the optimal logarithmic maximum a posteriori (log-MAP) detector with much lower computational complexity. In this paper, we first apply several mathematical tricks to derive a new and low-complexity algorithm for the max-log-MAP detector. The proposed algorithm features a new tree pruning technique to visit few lattice points and a smart list administration technique to retain a max-log-MAP detector. By following the proposed algorithm, we then design a hardware architecture for SISO detection of coded SM signals over the scenario of 64-QAM symbols, 8 transmit antennas, and 4 receive antennas. Under the TSMC 40 nm CMOS technology, the VLSI implementation results reveal that our architecture requires 46.4K gates and leads to detection throughput 450 Mbps, while operating at clock frequency 400 MHz and consuming power 46 mW. Being the first hardware architecture for SISO detection of coded SM signals, our proposed low-cost max-log-MAP detector is very attractive.

**Index Terms**—Spatial modulation, maximum a posteriori detection, log-likelihood ratio, maximum logarithm approximation, iterative detection and decoding, soft-input soft-output detection.

Manuscript received 6 September 2021; revised 26 January 2022 and 7 March 2022; accepted 1 June 2022. Date of publication 14 June 2022; date of current version 30 August 2022. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST 109-2221-E-194-032-MY2; and in part by the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan. This article was recommended by Associate Editor S. Yin. (Corresponding author: Tsung-Hsien Liu.)

Tsung-Hsien Liu is with the Department of Communications Engineering, National Chung Cheng University, Chiayi 621301, Taiwan, and also with the Advanced Institute for Manufacturing with High-tech Innovations (AIM-HI), National Chung Cheng University, Chiayi 621301, Taiwan (e-mail: comtshliu@ccu.edu.tw).

Ting-Xu Jiang is with the Department of Electrical Engineering, National Chung Cheng University, Chiayi 621301, Taiwan.

Ching-Che Chung is with the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 62102, Taiwan.

Yuan-Sun Chu is with the Department of Electrical Engineering, National Chung Cheng University, Chiayi 621301, Taiwan, and also with the Advanced Institute for Manufacturing with High-tech Innovations (AIM-HI), National Chung Cheng University, Chiayi 621301, Taiwan.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TCSI.2022.3181629>.

Digital Object Identifier 10.1109/TCSI.2022.3181629

## I. INTRODUCTION

THE spatial modulation (SM) scheme [1], [2] maps each block of data bits to one antenna index and one amplitude-phase modulated symbol. The symbol is passed through the single radio frequency (RF) chain and the indexed antenna for transmission over the multiple antenna system. Due to the additional information of antenna indices, the SM leads to higher transmission rate than the counterpart amplitude-phase modulation scheme.

Detection scheme at the receiver side of SM multiple input multiple output (MIMO) systems needs to recover both the antenna index and modulated symbol information associated with each block of data bits. Not considering the channel coding, the maximum likelihood detection (MLD) [3], [4] or nearly MLD scheme [5], [6] delivers hard-output antenna indices and symbols.

In practical wireless communication systems, channel coding and modulation schemes are employed together to enhance the transmission quality and reliability. Thus, trellis coded bits were mapped to the antenna index and modulated symbol associated with each SM data block [7]–[11]. Coded bits at the outputs of low density parity check (LDPC) encoder were sent to the SM mapper [12], [13]. The SM with two layers of Markov superposition transmission codes was studied in [14].

For the above-mentioned coded SM systems, iterative detection and decoding (IDD) [15]–[19] at the receiver side is a practical approach to provide reliable decoded bits. The soft-output (SO) or soft-input soft-output (SISO) detector for the IDD receiver computes the extrinsic log-likelihood ratios (LLRs) associated with the antenna index bits and symbol bits. The optimal logarithmic maximum *a posteriori* (log-MAP) detector [7] to compute the *a posteriori* extrinsic bit LLRs is of high computational complexity. Instead, the sub-optimal SO detectors computed bit LLRs using the outputs from a set of matched filters [20] or zero-forcing equalizers [12], [21] to reduce detection complexity. The low-complexity SO detection scheme [22] sorted the computed metrics associated with all possible antenna indices and selected only finite antenna indices with small metrics for further computation. Other relevant works include the SISO detection of multi-user coded SM MIMO signals [23] and

the SO detection of coded generalized SM signals [24]–[28], where multiple amplitude-phase symbols are transmitted over the multiple indexed antennas.

The sub-optimal maximum logarithmic MAP (max-log-MAP) detector for the coded SM signals is known to outperform nearly all the sub-optimal detectors [12], [20]–[22], but its computational complexity is very high. Nevertheless, the max-log-MAP detection algorithms and hardware architectures for SISO detection of the spatially multiplexed MIMO (SMX-MIMO) signals have been demonstrated to be efficient [15]–[18], [23], [29], [30]. We also reported [4], [31] that fast and efficient hard-output detectors for the SM and generalized SM signals are possible by exploiting multiple times of tree search based SMX MIMO detectors. Therefore, for the comparable scenario of SO/SISO detection of coded SM MIMO signals here, we have potential to obtain low-complexity SO/SISO detectors by exploiting those tree search techniques for the SO/SISO detection of coded SMX-MIMO signals [15], [29], [32], [33]. While developing new and fast algorithms, we also consider the computational steps that are easy to be converted to hardware architectures. In the end, we will arrive at a low-complexity and fast algorithm; and, following the computational steps in the algorithm, we design efficient hardware architectures for SO/SISO detection of the coded SM MIMO signals.

The remaining sections of this paper are organized as follows. The problem of IDD and SISO detection of the coded SM signals is defined in Sections II. Our proposed algorithms for the SISO detection together with their error rate performance are introduced in Section III. The proposed hardware architectures that follow from the proposed low-complexity algorithm are given in Section IV. Finally, conclusions about our algorithms and architectures are delivered in Section V.

Notations  $()^H$ ,  $|\cdot|$ ,  $\Re\{\cdot\}$ ,  $\Im\{\cdot\}$ ,  $\|\cdot\|$ , and  $\mathbb{E}\{\cdot\}$  denote the conjugate transpose (Hermitian), absolute value, real-part, imaginary-part, 2-norm, and expectation operations, respectively. Bold-faced lowercase and uppercase alphabets denote vectors and matrices, respectively.

## II. PROBLEM DEFINITION

Let us consider the baseband transmitter and receiver in Fig. 1 [18], [34] for the coded SM system. At the transmitter side, information bits are generated and passed through the channel encoder and interleaver to become coded bits. Assume that there are  $N_t$  (which is a power of 2) transmit antennas and that  $M$ -ary QAM modulation is used. Every  $\log_2(N_t M)$  coded bits are SM mapped to become one SM symbol. The logical 0 or 1 of each bit  $x_\ell$  is represented by  $-1$  or  $1$ , respectively. Specifically, the bits  $x_\ell, \ell \in \mathcal{I}^A = \{1, 2, \dots, \log_2 N_t\}$ , are mapped to antenna index  $k \in \mathcal{K} = \{1, 2, \dots, N_t\}$  (see Table I). In the meanwhile, the bits  $x_\ell, \ell \in \mathcal{I}^I = \{\log_2 N_t + 1, \dots, \log_2 N_t + (1/2) \log_2 M\}$ , and bits  $x_\ell, \ell \in \mathcal{I}^Q = \{\log_2 N_t + (1/2) \log_2 M + 1, \dots, \log_2 N_t + \log_2 M\}$ , are mapped to the real- and imaginary-parts, respectively, of an  $M$ -ary QAM symbol  $s \in \mathcal{S} = \{a + jb : a \text{ and } b \in \{-\sqrt{M}+1, -\sqrt{M}+3, \dots, \sqrt{M}-3, \sqrt{M}-1\}\}$ . One example of the mapping from bits to  $\Re\{s\}$  or  $\Im\{s\}$  is illustrated in

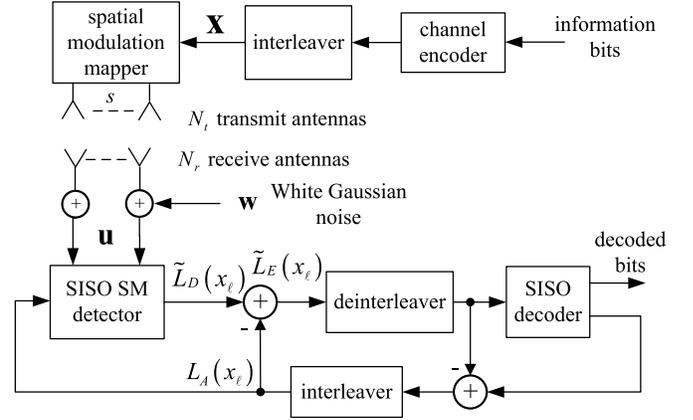


Fig. 1. The block diagram of the baseband transmitter and iterative receiver [18].

TABLE I  
BIT TO ANTENNA INDEX MAPPING FOR  $N_t = 8$

Bits	Antenna Index $k$	Channel Vector $\mathbf{h}_k$
000	1	$\mathbf{h}_1$
001	2	$\mathbf{h}_2$
010	3	$\mathbf{h}_3$
011	4	$\mathbf{h}_4$
100	5	$\mathbf{h}_5$
101	6	$\mathbf{h}_6$
110	7	$\mathbf{h}_7$
111	8	$\mathbf{h}_8$

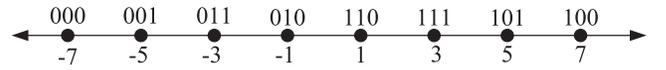


Fig. 2. Bit to  $\Re\{s\}$  or  $\Im\{s\}$  mapping for 64-QAM signals.

TABLE II  
THE EQUIVALENT NOTATIONS TO REPRESENT THE SAME INFORMATION

Information	Equivalent notations
Antenna Index	$k, \mathbf{x}^A, \{x_\ell : \ell \in \mathcal{I}^A\}$
Real-part of QAM Symbol	$\Re\{s\}, \mathbf{x}^I, \{x_\ell : \ell \in \mathcal{I}^I\}$
Imaginary-part of QAM Symbol	$\Im\{s\}, \mathbf{x}^Q, \{x_\ell : \ell \in \mathcal{I}^Q\}$
SM Symbol	$(k, s), \mathbf{x}, \{x_\ell : \ell \in \mathcal{I}^A \cup \mathcal{I}^I \cup \mathcal{I}^Q\}$

Fig. 2. We also let  $\mathbf{x}$  be the row vector that contains bits  $x_\ell, \forall \ell$ . Thus,  $\mathbf{x}$  can be decomposed into three sub-vectors of bits that are mapped to  $k$ ,  $\Re\{s\}$ , and  $\Im\{s\}$ , respectively, i.e.,  $\mathbf{x} \triangleq [\mathbf{x}^A \ \mathbf{x}^I \ \mathbf{x}^Q]$ . The QAM symbol  $s$  is passed through the single RF chain and then to only the  $k$ -th antenna for transmission. At every time instant, one SM lattice point  $(k, s)$  or equivalent bit vector  $\mathbf{x}$  is transmitted. For notational simplicity, the equivalent notations to denote the same symbol- and bit-level information are given in Table II and will be used interchangeably in this paper. It is possible that symbol-level  $k$  and bit-level  $\mathbf{x}^A$  appear in one equation to denote the same antenna index information.

The channel gain between the  $k$ -th transmit and  $n$ -th receive antennas is  $h_{n,k}$ . Each of the receive antennas is connected to

a receive RF chain. The received baseband signal is

$$\underbrace{\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N_r} \end{bmatrix}}_{\mathbf{u}} = \underbrace{\begin{bmatrix} h_{1,k} \\ h_{2,k} \\ \vdots \\ h_{N_r,k} \end{bmatrix}}_{\mathbf{h}_k} s + \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N_r} \end{bmatrix}}_{\mathbf{w}}, \quad (1)$$

where  $w_n$  is the white Gaussian noise with variance  $N_0$ . The signal-to-noise ratio (SNR) associated with the received  $\mathbf{u}$  is  $10 \log_{10}((2/3)(M-1)/N_0)$  dB, where the factor  $(2/3)(M-1)$  is used due to the fact that  $\mathbb{E}\{|s|^2\} = (2/3)(M-1)$ .

At the receiver side, the IDD procedure in Fig. 1 is performed as follows. The *a posteriori* bit LLR associated with each bit  $x_\ell$  is defined by

$$\tilde{L}_D(x_\ell) \triangleq \ln \left( \frac{P[x_\ell = +1|\mathbf{u}]}{P[x_\ell = -1|\mathbf{u}]} \right). \quad (2)$$

By exploiting the statistical properties for the random  $\mathbf{x}$  and  $\mathbf{u}$ , the  $\tilde{L}_D(x_\ell)$  in (2) can be written as [18]

$$\tilde{L}_D(x_\ell) = \ln \frac{\sum_{\mathbf{x} \in \mathcal{X}_\ell^{+1}} [p(\mathbf{u}|\mathbf{x}) \exp(\frac{1}{2} \sum_i x_i L_A(x_i))]}{\sum_{\mathbf{x} \in \mathcal{X}_\ell^{-1}} [p(\mathbf{u}|\mathbf{x}) \exp(\frac{1}{2} \sum_i x_i L_A(x_i))]}, \quad (3)$$

where  $L_A(x_\ell) \triangleq \ln \left( \frac{P[x_\ell = +1]}{P[x_\ell = -1]} \right)$  represents the *a priori* bit LLR associated with bit  $x_\ell$ . The  $\mathcal{X}_\ell^{+1}$  or  $\mathcal{X}_\ell^{-1}$  denotes the set of  $\mathbf{x}$ 's, where each  $\mathbf{x}$  having bit  $x_\ell$  equal to  $+1$  or  $-1$ , respectively. The SISO detector computes the extrinsic bit LLRs, e.g.,

$$\tilde{L}_E(x_\ell) = \tilde{L}_D(x_\ell) - L_A(x_\ell), \quad \forall \ell, \quad (4)$$

and send them to subsequent deinterleaver and SISO channel decoder. By regarding the  $\tilde{L}_E(x_\ell)$ ,  $\forall \ell$ , as *a priori* information, the SISO decoder delivers the decoded information bits and computes the extrinsic information for the coded bits. The extrinsic information is next sent back to the SISO detector as the *a priori* bit LLRs,  $L_A(x_\ell)$ ,  $\forall \ell$ . In this way, one iteration of the detection and decoding of information bits is completed. The detection and decoding may proceed for several iterations until the decoded information bits at the decoder output are reliable. Note that in the first iteration, the SISO detection proceeds with  $L_A(x_\ell) = 0$ ,  $\forall \ell$ . When only one iteration of the detection and decoding is conducted, the SISO detection is reduced to the SO detection.

For the coded SM-MIMO system, the SISO detector needs to compute  $\tilde{L}_D(x_\ell)$  and  $\tilde{L}_E(x_\ell)$ ,  $\forall \ell$ , from the observed  $\mathbf{u}$  and  $L_A(x_\ell)$ ,  $\forall \ell$ , according to (3) and (4). From the model of (1), the conditional probability  $p(\mathbf{u}|\mathbf{x})$  or  $p(\mathbf{u}|(k, s))$  is

$$p(\mathbf{u}|\mathbf{x}) = (\pi N_0)^{-N_r} \exp \left( -\frac{1}{N_0} \|\mathbf{u} - \mathbf{h}_k s\|^2 \right). \quad (5)$$

By applying (5) to (3), one obtains

$$\begin{aligned} & \tilde{L}_D(x_\ell) \\ &= \ln \frac{\sum_{\mathbf{x} \in \mathcal{X}_\ell^{+1}} \exp \left( -\frac{1}{N_0} \|\mathbf{u} - \mathbf{h}_k s\|^2 + \frac{1}{2} \sum_i x_i L_A(x_i) \right)}{\sum_{\mathbf{x} \in \mathcal{X}_\ell^{-1}} \exp \left( -\frac{1}{N_0} \|\mathbf{u} - \mathbf{h}_k s\|^2 + \frac{1}{2} \sum_i x_i L_A(x_i) \right)}. \end{aligned} \quad (6)$$

The computational challenge for obtaining the  $\tilde{L}_D(x_\ell)$ ,  $\forall \ell$ , is the need to evaluate  $2^{N_r M}$  times of  $\exp(\cdot)$  operations. One common approach to avoid the  $\exp(\cdot)$  operation is to exploit the maximum logarithm approximation [18], i.e.,  $\ln(e^a + e^b) \approx \max(a, b)$ . By repeatedly applying the maximum logarithm approximation to (6), one arrives at

$$\begin{aligned} \tilde{L}_D(x_\ell) &\approx L_D(x_\ell) \\ &\triangleq \max_{\mathbf{x} \in \mathcal{X}_\ell^{+1}} \left\{ -\frac{1}{N_0} \|\mathbf{u} - \mathbf{h}_k s\|^2 + \frac{1}{2} \sum_i x_i L_A(x_i) \right\} \\ &\quad - \max_{\mathbf{x} \in \mathcal{X}_\ell^{-1}} \left\{ -\frac{1}{N_0} \|\mathbf{u} - \mathbf{h}_k s\|^2 + \frac{1}{2} \sum_i x_i L_A(x_i) \right\}. \end{aligned} \quad (7)$$

The sub-optimal SISO detector that computes  $L_D(x_\ell)$  according to (7) and then sends

$$L_E(x_\ell) = L_D(x_\ell) - L_A(x_\ell), \quad \forall \ell, \quad (8)$$

to subsequent channel decoder is referred to as the max-log-MAP detector. In comparison, the optimal log-MAP detector computes the exact  $\tilde{L}_D(x_\ell)$  according to (6) and then  $\tilde{L}_E(x_\ell)$  according to (4). Other sub-optimal detectors [12], [20]–[22] compute approximate values to the  $\tilde{L}_E(x_\ell)$ ,  $\forall \ell$ , in (4).

Our work in this paper is to study the low-complexity algorithm and hardware architecture for computing  $L_D(x_\ell)$  and  $L_E(x_\ell)$  according to (7) and (8), respectively. Since the computation of  $L_E(x_\ell)$  from  $L_D(x_\ell)$  and  $L_A(x_\ell)$  is trivial, our emphasis will be on the computation of  $L_D(x_\ell)$ .

### III. PROPOSED SISO DETECTION ALGORITHMS

In this section, we introduce our proposed SISO detection algorithms. The simulation results for our proposed algorithms together with the required complexity are also discussed.

#### A. Proposed Exhaustive SISO Detection Algorithm

We start to derive our proposed algorithms for computing  $L_D(x_\ell)$ ,  $\forall \ell$ , according to (7). First, by applying

$$\mathbf{q}_k = \mathbf{h}_k / \|\mathbf{h}_k\|, \quad y_k = \mathbf{q}_k^H \mathbf{u}, \quad \text{and } r_k = \|\mathbf{h}_k\|, \quad (9)$$

we have [4, eq. (3)]

$$\|\mathbf{u} - \mathbf{h}_k s\|^2 = \|\mathbf{u}\|^2 - |y_k|^2 + |y_k - r_k s|^2. \quad (10)$$

Substituting (10) into (7), we obtain

$$\begin{aligned} & L_D(x_\ell) \\ &= \min_{\mathbf{x} \in \mathcal{X}_\ell^{-1}} \left\{ \frac{1}{N_0} \left( -|y_k|^2 + |y_k - r_k s|^2 \right) - \frac{1}{2} \sum_i x_i L_A(x_i) \right\} \\ &\quad - \min_{\mathbf{x} \in \mathcal{X}_\ell^{+1}} \left\{ \frac{1}{N_0} \left( -|y_k|^2 + |y_k - r_k s|^2 \right) - \frac{1}{2} \sum_i x_i L_A(x_i) \right\}, \end{aligned} \quad (11)$$

where each max is replaced by -min. The  $\|\mathbf{u}\|^2$  is cancelled in (11), because it appears in both the two minimizations. The computation of  $L_D(x_\ell)$  according to (7) is more complex, because it involves the norm of  $N_r \times 1$  vector  $\mathbf{u} - \mathbf{h}_k s$ . In contrast, the computation of  $L_D(x_\ell)$  according to (11) is less complex, because it involves only norm of scalar  $y_k - r_k s$ .

Besides, an alternative expression for  $L_D(x_\ell)$  can facilitate the derivation of our algorithms [16]. For each bit  $x_\ell$ , one of the two minima in (11) is

$$\lambda^{\text{MAP}} = \min_{\mathbf{x} \in \mathcal{X}_\ell^{-1} \cup \mathcal{X}_\ell^{+1}} \rho(k, s), \quad (12)$$

where the metric associated with SM lattice point  $(k, s)$  is

$$\rho(k, s) \triangleq \frac{1}{N_0} \left( -|y_k|^2 + |y_k - r_k s|^2 \right) - \frac{1}{2} \sum_i x_i L_A(x_i). \quad (13)$$

The  $\mathcal{X}_\ell^{-1} \cup \mathcal{X}_\ell^{+1}$  in (12) actually denotes the set of all  $\mathbf{x}$ 's. Also, let  $\mathbf{x}^{\text{MAP}}$  be the bit vector  $\mathbf{x}$  that leads to  $\lambda^{\text{MAP}}$ , i.e.,

$$\mathbf{x}^{\text{MAP}} = \arg \min_{\mathbf{x} \in \mathcal{X}_\ell^{-1} \cup \mathcal{X}_\ell^{+1}} \rho(k, s). \quad (14)$$

The other minimum for bit  $x_\ell$  in (11) can be written as

$$\lambda_\ell^{\overline{\text{MAP}}} = \min_{\mathbf{x} \in \mathcal{X}_\ell^{x_\ell^{\overline{\text{MAP}}}}} \rho(k, s), \quad (15)$$

where  $x_\ell^{\text{MAP}}$  denotes the  $\ell$ -th bit of  $\mathbf{x}^{\text{MAP}}$ , and,  $x_\ell^{\overline{\text{MAP}}}$  denotes the bit inverse (counter hypothesis) of  $x_\ell^{\text{MAP}}$ . With the two minima, i.e.,  $\lambda^{\text{MAP}}$  and  $\lambda_\ell^{\overline{\text{MAP}}}$ , the  $L_D(x_\ell)$  in (11) can alternatively be written as

$$L_D(x_\ell) = \begin{cases} \lambda_\ell^{\overline{\text{MAP}}} - \lambda^{\text{MAP}}, & x_\ell^{\text{MAP}} = +1 \\ \lambda^{\text{MAP}} - \lambda_\ell^{\overline{\text{MAP}}}, & x_\ell^{\text{MAP}} = -1. \end{cases} \quad (16)$$

Based on the expressions in (12)-(16), we now have the proposed Algorithm 1. Our algorithm includes two parts of computations, preprocessing and payload processing. In the preprocessing, the channel vectors  $\mathbf{h}_k, \forall k$ , are available from channel estimation. In lines 2-4, the algorithm computes  $\mathbf{q}_k$  and  $r_k$  from each  $\mathbf{h}_k$ . In the payload processing, the algorithm processes each payload vector  $\mathbf{u}$  to deliver bit LLR information  $L_D(x_\ell), \forall \ell$ . The payload processing of our algorithm is motivated by the single tree search (STS) [16], [32] for the SISO detection of SMX MIMO signals. Recall that the STS traverses a tree to visit each leaf node and computes the metric associated with each SMX symbol vector. A list administration technique is then applied to update the required MAP information for computing bit LLRs. The STS is guaranteed to achieve the max-log-MAP detection of SMX MIMO signals.

Here, our algorithm exhaustively visits each SM lattice point  $(k, s)$  and computes the associated metric  $\rho(k, s)$  in line 10 of Algorithm 1. For the first visited SM lattice point, the computed  $\rho(k, s)$  and  $\mathbf{x}$  in line 12 are saved as the initial  $\lambda^{\text{MAP}}$  and  $\mathbf{x}^{\text{MAP}}$ , respectively. For any other visited lattice point, the computed  $\rho(k, s)$  and  $\mathbf{x}$  are used to update the MAP information

$$\mathcal{L}^{\text{MAP}} = \left\{ \lambda^{\text{MAP}}, \mathbf{x}^{\text{MAP}}, \text{and } \lambda_\ell^{\overline{\text{MAP}}}, \forall \ell \right\}. \quad (17)$$

When  $\rho(k, s) < \lambda^{\text{MAP}}$ , the update of  $\mathcal{L}^{\text{MAP}}$  occurs as shown by lines 16-21. When  $\rho(k, s) < \lambda_\ell^{\overline{\text{MAP}}}$ , the update of  $\lambda_\ell^{\overline{\text{MAP}}}$  may occur as shown by lines 23-27. The list administration of  $\mathcal{L}^{\text{MAP}}$  in lines 15-28 is to retain the most recent MAP

---

**Algorithm 1** The Proposed Exhaustive Search Algorithm for SISO Detection of Coded SM MIMO Signals

---

**input** : Payload vector  $\mathbf{u}$ , channel vectors  $\mathbf{h}_k, \forall k$ , and  $L_A(x_\ell), \forall \ell$   
**output**:  $L_E(x_\ell), \forall \ell$

```

1 % Preprocessing
2 for  $k = 1$  to  $N_t$  do
3   | Compute  $\mathbf{q}_k$  and  $r_k$  from  $\mathbf{h}_k$  according to (9)
4 end
5 % One iteration of payload processing of  $\mathbf{u}$ 
6 for  $k = 1$  to  $N_t$  do
7   | Compute  $y_k = \mathbf{q}_k^H \mathbf{u}$  as in (9)
8   | for each  $s \in \mathcal{S}$  do
9     | % Visit lattice point  $(k, s)$ 
10    | Compute  $\rho(k, s)$  according to (13)
11    | if  $k = 1$  and  $s = -\sqrt{M} + 1 + j(-\sqrt{M} + 1)$ 
12    |   | then
13    |   |   |  $\lambda^{\text{MAP}} = \rho(k, s)$ ,  $\mathbf{x}^{\text{MAP}} = \mathbf{x}$ , and
14    |   |   |  $\lambda_\ell^{\overline{\text{MAP}}} = \infty, \forall \ell$ 
15    |   | else
16    |   |   | % List administration
17    |   |   | if  $\rho(k, s) < \lambda^{\text{MAP}}$  then
18    |   |   |   | for  $\ell = 1$  to  $\log_2(N_t M)$  do
19    |   |   |   |   | if  $x_\ell \neq x_\ell^{\text{MAP}}$  then
20    |   |   |   |   |   |  $\lambda_\ell^{\overline{\text{MAP}}} = \lambda^{\text{MAP}}$ 
21    |   |   |   |   | end
22    |   |   |   | end
23    |   |   |   |  $\lambda^{\text{MAP}} = \rho(k, s)$ ,  $\mathbf{x}^{\text{MAP}} = \mathbf{x}$ 
24    |   |   | else
25    |   |   |   | for  $\ell = 1$  to  $\log_2(N_t M)$  do
26    |   |   |   |   | if  $x_\ell \neq x_\ell^{\text{MAP}}$  and  $\rho(k, s) < \lambda_\ell^{\overline{\text{MAP}}}$ 
27    |   |   |   |   |   | then
28    |   |   |   |   |   |   |  $\lambda_\ell^{\overline{\text{MAP}}} = \rho(k, s)$ 
29    |   |   |   |   |   | end
30    |   |   |   |   | end
31    |   |   |   | end
32    |   |   | end
33    |   | end
34   | end
35 end
36 Compute  $L_D(x_\ell)$  and then  $L_E(x_\ell), \forall \ell$ , according
37 to (16) and (8), respectively.

```

---

information. When all the SM lattice points  $(k, s), \forall k$  and  $\forall s$ , are visited, the finally obtained  $\lambda^{\text{MAP}}, \mathbf{x}^{\text{MAP}}$ , and  $\lambda_\ell^{\overline{\text{MAP}}}, \forall \ell$ , satisfy (12), (14), and (15), respectively. In line 32, the  $L_D(x_\ell)$  and  $L_E(x_\ell), \forall \ell$ , are computed from  $\lambda^{\text{MAP}}, \mathbf{x}^{\text{MAP}}$ , and  $\lambda_\ell^{\overline{\text{MAP}}}, \forall \ell$ .

### B. Proposed Improved SISO Detection Algorithm

The complexity of Algorithm 1 to visit all  $N_t M$  lattice points is high. Some of the visits to the lattice points may lead to the update of  $\mathcal{L}^{\text{MAP}}$ , while some of the visits may not. To reduce the computational complexity, we should avoid visiting those SM lattice points that lead to no update of  $\mathcal{L}^{\text{MAP}}$ .

To achieve this goal, we may take advantage of the fact: if the metric  $\rho(k, s)$  associated with lattice point  $(k, s)$  is larger than the most recent  $\lambda^{\text{MAP}}$  and  $\lambda_{\ell}^{\text{MAP}}, \forall \ell$ , the lattice point leads to no update of  $\mathcal{L}^{\text{MAP}}$ . Please refer to lines 15-28 of Algorithm 1 for this fact.

A new expression for the  $\rho(k, s)$  may help us to derive our following algorithm. Each complex number is decomposed into its real- and imaginary-components. Hence, the lattice point information  $(k, s)$  is divided into three parts, i.e.,  $k$ ,  $\Re\{s\}$ , and  $\Im\{s\}$ . By exploiting the equivalent notations in Table II, the  $\rho(k, s)$  in (13) can be written as

$$\begin{aligned} \rho(k, s) &= \underbrace{\frac{-1}{N_0} |y_k|^2 - \frac{1}{2} \sum_{x_i, i \in \mathcal{I}^A} x_i L_A(x_i)}_{\rho^A(k, s)} \\ &+ \underbrace{\frac{1}{N_0} |\Re\{y_k\} - r_k \Re\{s\}|^2 - \frac{1}{2} \sum_{x_i, i \in \mathcal{I}^I} x_i L_A(x_i)}_{\rho^I(k, s)} \\ &+ \underbrace{\frac{1}{N_0} |\Im\{y_k\} - r_k \Im\{s\}|^2 - \frac{1}{2} \sum_{x_i, i \in \mathcal{I}^Q} x_i L_A(x_i)}_{\rho^Q(k, s)}. \end{aligned} \quad (18)$$

Now, we consider to reduce the complexity of Algorithm 1. We regard Algorithm 1 as an algorithm being comprised of  $N_t$  tree traversals. In each tree traversal, the Algorithm 1, between lines 8 and 30, visits a total of  $M$  lattice points associated with a fixed  $k$ , i.e.,  $(k, s), \forall s$ . The computed  $M$  metrics are used to update the  $\lambda^{\text{MAP}}$ ,  $\mathbf{x}^{\text{MAP}}$ , and  $\lambda_{\ell}^{\text{MAP}}, \forall \ell$ . Actually, among the  $M$  lattice points visited by the  $k$ -th tree traversal, there is one lattice point that is associated with the minimum metric

$$\beta^{\text{MAP}} = \min_s \rho(k, s). \quad (19)$$

The metric associated with any other  $M - 1$  lattice points is larger than  $\beta^{\text{MAP}}$  and cannot supersede  $\beta^{\text{MAP}}$  to be the  $\lambda^{\text{MAP}}$ . By applying (18) together with the equivalence between  $s$  and  $[\mathbf{x}^I \ \mathbf{x}^Q]$ , we have from (19) that

$$\beta^{\text{MAP}} = \rho^A(k, s) + \underbrace{\min_{\mathbf{x}^I} \rho^I(k, s)}_{\beta^{I, \text{MAP}}} + \underbrace{\min_{\mathbf{x}^Q} \rho^Q(k, s)}_{\beta^{Q, \text{MAP}}}. \quad (20)$$

The equality in (20) holds, because  $\rho^I(k, s)$  is independent of  $\mathbf{x}^Q$  and  $\rho^Q(k, s)$  is independent of  $\mathbf{x}^I$ . For further derivations, we let the  $\mathbf{x}^I$  and  $\mathbf{x}^Q$  that lead to  $\beta^{I, \text{MAP}}$  and  $\beta^{Q, \text{MAP}}$ , respectively, be

$$\begin{aligned} \mathbf{z}^{I, \text{MAP}} &= \arg \min_{\mathbf{x}^I} \rho^I(k, s) \quad \text{and} \\ \mathbf{z}^{Q, \text{MAP}} &= \arg \min_{\mathbf{x}^Q} \rho^Q(k, s). \end{aligned} \quad (21)$$

Accordingly, the SM lattice point that leads to the metric  $\beta^{\text{MAP}}$  in (20) is associated with the bit vector

$$\mathbf{z}^{\text{MAP}} = [\mathbf{z}^A \ \mathbf{z}^{I, \text{MAP}} \ \mathbf{z}^{Q, \text{MAP}}], \quad (22)$$

where  $\mathbf{z}^A$  contains the bits that are mapped to antenna index  $k$ . Please refer to Table I for the mapping from  $k$  to  $\mathbf{z}^A$ .

We also consider to visit the lattice points that may lead to the update of  $\lambda_{\ell}^{\text{MAP}}, \forall \ell$ , for the  $k$ -th tree traversal of Algorithm 1. Refer to (15). For fixed  $k$  and each  $\ell \in \mathcal{I}^I$ , we only need to compute

$$\begin{aligned} \beta_{\ell}^{\text{MAP}} &= \min_{\mathbf{x}^I, \mathbf{x}^Q, x_{\ell} = z_{\ell}^{I, \text{MAP}}} \rho(k, s) \\ &= \rho^A(k, s) + \underbrace{\min_{\mathbf{x}^I, x_{\ell} = z_{\ell}^{I, \text{MAP}}} \rho^I(k, s)}_{\beta_{\ell}^{I, \text{MAP}}} + \underbrace{\min_{\mathbf{x}^Q} \rho^Q(k, s)}_{\beta^{Q, \text{MAP}}}, \end{aligned} \quad (23)$$

where the last equality holds for the same reason as the one that the equality in (20) holds. Note that the identical  $\beta^{Q, \text{MAP}}$  is computed in both (20) and (23). Also note that  $z_{\ell}^{I, \text{MAP}}$  denotes the bit inverse of the  $\ell$ -th bit in vector  $\mathbf{z}^{I, \text{MAP}}$ . The  $\mathbf{x}^I$  and  $\mathbf{x}$  that lead to  $\beta_{\ell}^{I, \text{MAP}}$  and  $\beta_{\ell}^{\text{MAP}}$  in (23) are denoted respectively by

$$\mathbf{z}_{\ell}^{I, \text{MAP}} = \arg \min_{\mathbf{x}^I, x_{\ell} = z_{\ell}^{I, \text{MAP}}} \rho^I(k, s). \quad (24)$$

and

$$\mathbf{z}_{\ell}^{\text{MAP}} = [\mathbf{z}^A \ \mathbf{z}_{\ell}^{I, \text{MAP}} \ \mathbf{z}^{Q, \text{MAP}}], \quad \ell \in \mathcal{I}^I. \quad (25)$$

Likewise, for fixed  $k$  and each  $\ell \in \mathcal{I}^Q$ , we only need to compute

$$\begin{aligned} \beta_{\ell}^{\text{MAP}} &= \min_{\mathbf{x}^I, \mathbf{x}^Q, x_{\ell} = z_{\ell}^{Q, \text{MAP}}} \rho(k, s) \\ &= \rho^A(k, s) + \underbrace{\min_{\mathbf{x}^I} \rho^I(k, s)}_{\beta^{I, \text{MAP}}} \\ &+ \underbrace{\min_{\mathbf{x}^Q, x_{\ell} = z_{\ell}^{Q, \text{MAP}}} \rho^Q(k, s)}_{\beta_{\ell}^{Q, \text{MAP}}}. \end{aligned} \quad (26)$$

The  $\mathbf{x}^Q$  and  $\mathbf{x}$  that lead to  $\beta_{\ell}^{Q, \text{MAP}}$  and  $\beta_{\ell}^{\text{MAP}}$  in (26) are denoted respectively by

$$\mathbf{z}_{\ell}^{Q, \text{MAP}} = \arg \min_{\mathbf{x}^Q, x_{\ell} = z_{\ell}^{Q, \text{MAP}}} \rho^Q(k, s). \quad (27)$$

and

$$\mathbf{z}_{\ell}^{\text{MAP}} = [\mathbf{z}^A \ \mathbf{z}^{I, \text{MAP}} \ \mathbf{z}_{\ell}^{Q, \text{MAP}}], \quad \ell \in \mathcal{I}^Q. \quad (28)$$

Based on the results in (19)-(28), we now have the proposed Algorithm 2. The processing of  $\mathbf{u}$  is still comprised of  $N_t$  tree traversals. For the  $k$ -th tree traversal, the results in (19)-(22) suggest us to compute  $(\mathbf{z}^{I, \text{MAP}}, \beta^{I, \text{MAP}})$ ,  $(\mathbf{z}^{Q, \text{MAP}}, \beta^{Q, \text{MAP}})$ , and  $(\mathbf{z}^{\text{MAP}}, \beta^{\text{MAP}})$ , although  $k$  is not appeared in these notations. Then, we compute the  $\beta_{\ell}^{I, \text{MAP}}$  in (23) and  $\beta_{\ell}^{Q, \text{MAP}}$  in (26) for each  $\ell \in \mathcal{I}^I$  and  $\ell \in \mathcal{I}^Q$ , respectively. The  $\beta_{\ell}^{\text{MAP}}$  for  $\ell \in \mathcal{I}^I$  and  $\ell \in \mathcal{I}^Q$  are computed as well according to (23) and (26), respectively. Thus, for the  $k$ -th tree traversal, only the list of the partial MAP information, i.e.,

$$\mathcal{L}^{\text{part}} = \left\{ \beta^{\text{MAP}}, \mathbf{z}^{\text{MAP}}, \text{and } \beta_{\ell}^{\text{MAP}}, \ell \in \mathcal{I}^I \cup \mathcal{I}^Q \right\}, \quad (29)$$

is computed. There are no computed values of  $\beta_\ell^{\overline{\text{MAP}}}$ ,  $\ell \in \mathcal{I}^A$ , because the bits  $x_\ell$ ,  $\ell \in \mathcal{I}^A$ , are associated with the single and fixed  $k$ . Some further comments about the processing of each payload vector  $\mathbf{u}$  are given below.

- In lines 5-13 of Algorithm 2, the  $k$ -th tree traversal is conducted to compute the  $k$ -th  $\mathcal{L}^{\text{part}}$ . As indicated in lines 15-19, the first computed  $\mathcal{L}^{\text{part}}$  is used to initialize the values in  $\mathcal{L}^{\text{MAP}}$ ; but, for any other generated  $\mathcal{L}^{\text{part}}$ , the computed values in  $\mathcal{L}^{\text{part}}$  are used to update the values in  $\mathcal{L}^{\text{MAP}}$ .
- For the  $k$ -th tree traversal, the Algorithm 1 needs to visit  $M$  SM lattice points. Now Algorithm 2 only needs to visit one lattice point that may lead to the update of all values in  $\mathcal{L}^{\text{MAP}}$ , and,  $\log_2 M$  lattice points that may lead to the update of  $\lambda_\ell^{\overline{\text{MAP}}}$ ,  $\forall \ell$ . Thus, Algorithm 2 only visits a total number of  $\log_2 M + 1$  lattice points and delivers  $\mathcal{L}^{\text{part}}$  for the  $k$ -th tree traversal.
- During the  $k$ -th tree traversal, some of the quantities are computed only once, but are used several times to achieve complexity reduction. The quantities include at least  $\beta^{I,\text{MAP}}$ ,  $\beta^{Q,\text{MAP}}$ ,  $\rho^A(k, s)$ ,  $\mathbf{z}^{I,\text{MAP}}$ , and  $\mathbf{z}^{Q,\text{MAP}}$ . For example,  $\beta^{I,\text{MAP}}$  is computed in line 6, and is used in lines 7 and 9.
- The list administration here in Algorithm 3 is to update  $\mathcal{L}^{\text{MAP}}$  from the  $\mathcal{L}^{\text{part}}$  that is obtained in the  $k$ -th tree traversal. This list administration is different from the one in Algorithm 1, where each  $\rho(k, s)$  is used to update  $\mathcal{L}^{\text{MAP}}$ . Note that the properties of the bits associated with antenna index and symbols are different; and, there is no  $\beta_\ell^{\overline{\text{MAP}}}$ ,  $\ell \in \mathcal{I}^A$ , contained in  $\mathcal{L}^{\text{part}}$ . The operations to update the  $\lambda_\ell^{\overline{\text{MAP}}}$ ,  $\ell \in \mathcal{I}^A$ , are listed separately in lines 2-6 and 22-26 of Algorithm 3. The operations to update the  $\lambda_\ell^{\overline{\text{MAP}}}$ ,  $\ell \in \mathcal{I}^I \cup \mathcal{I}^Q$ , are listed separately in lines 7-19 and 27-37. One important property for the values in  $\mathcal{L}^{\text{part}}$  is that  $\beta^{\text{MAP}} < \beta_\ell^{\overline{\text{MAP}}}$ ,  $\ell \in \mathcal{I}^I \cup \mathcal{I}^Q$ ; and, only  $\beta^{\text{MAP}}$  and associated  $\mathbf{z}^{\text{MAP}}$  are considered to update  $\lambda^{\text{MAP}}$  and  $\mathbf{x}^{\text{MAP}}$ , respectively. We only check whether  $\beta^{\text{MAP}} < \lambda^{\text{MAP}}$  is valid in line 1. Other comparisons among the values from  $\mathcal{L}^{\text{part}}$  and  $\mathcal{L}^{\text{MAP}}$  are made to determine the appropriate updates of values in  $\mathcal{L}^{\text{MAP}}$ .

### C. Simulation Results

Simulations based on the transmitter and receiver structure in Fig. 1 were conducted. At the transmitter side, frames of information bits were transmitted. The convolutional encoder with generators  $133_o$  and  $171_o$  was used to produce rate 1/2 coded bits [35]. Coded bits with higher coding rate, e.g., 2/3, were obtained by applying puncturing to the rate 1/2 convolutional coded bits. The coded SM MIMO data frames were passed through slow Rayleigh fading channels. The channel was invariant over the duration that a frame was transmitted, but was variant over different frame durations. At the receiver side, each SISO/SO detection scheme worked together and iteratively with the SISO BCJR decoding [36] to yield decoded information bits. The proposed sub-optimal detector computes  $L_D(x_\ell)$  and  $L_E(x_\ell)$ ,  $\forall \ell$ , according to Algorithm 2. Because Algorithm 1 yields the same detection

---

**Algorithm 2** The Proposed Improved Tree Search Algorithm for SISO Detection of Coded SM MIMO Signals

---

**input** : Payload vector  $\mathbf{u}$ , channel vectors  $\mathbf{h}_k$ ,  $\forall k$ , and  $L_A(x_\ell)$ ,  $\forall \ell$   
**output**:  $L_E(x_\ell)$ ,  $\forall \ell$

```

1 % Preprocessing
2 Compute as in lines 2-4 of Algorithm 1
3 % One iteration of payload processing of  $\mathbf{u}$ 
4 for  $k = 1$  to  $N_t$  do
5   Compute  $y_k = \mathbf{q}_k^H \mathbf{u}$  as in (9)
6   Compute  $\beta^{I,\text{MAP}}$  and  $\beta^{Q,\text{MAP}}$  in (20) and obtain
   the corresponding  $\mathbf{z}^{I,\text{MAP}}$  and  $\mathbf{z}^{Q,\text{MAP}}$  in (21)
7   Compute the  $\beta^{\text{MAP}}$  in (20) and the  $\mathbf{z}^{\text{MAP}}$  in (22)
8   for each  $\ell \in \mathcal{I}^I$  do
9     Compute  $\beta_\ell^{I,\text{MAP}}$  and then  $\beta_\ell^{\overline{\text{MAP}}}$  according
   to (23)
10  end
11  for each  $\ell \in \mathcal{I}^Q$  do
12    Compute  $\beta_\ell^{Q,\text{MAP}}$  and then  $\beta_\ell^{\overline{\text{MAP}}}$  according
   to (26)
13  end
14  % List Administration
15  if  $k = 1$  then
16     $\lambda^{\text{MAP}} = \beta^{\text{MAP}}$ ,  $\mathbf{x}^{\text{MAP}} = \mathbf{z}^{\text{MAP}}$ ,
     $\lambda_\ell^{\overline{\text{MAP}}} = \infty$ ,  $\ell \in \mathcal{I}^A$ , and
     $\lambda_\ell^{\overline{\text{MAP}}} = \beta_\ell^{\overline{\text{MAP}}}$ ,  $\ell \in \mathcal{I}^I \cup \mathcal{I}^Q$ 
17  else
18    Execute the list administration in Algorithm 3
19  end
20 end
21 Compute  $L_D(x_\ell)$  and then  $L_E(x_\ell)$ ,  $\forall \ell$ , according
   to (16) and (8), respectively.

```

---

results as the Algorithm 2, each bit error rate (BER) curve in the figures that is associated with the proposed detector can be regarded as being associated with either Algorithm 1 or Algorithm 2. Performance measures for the iterative detection and decoding include the information BER and frame error rate (FER) at the decoder output.

The information BER curves associated with the log-MAP detector and proposed detector under the simulation Scenario #1 in Table III are illustrated in Fig. 3. The BER associated with either the log-MAP or proposed detector decreases as the number of detection and decoding iterations increases; and, the BER improvement after 3 iterations is small. Also, for each iteration, the optimal log-MAP detector outperforms the proposed detector by less than 0.3 dB at BER level  $10^{-5}$ . The BER curves for the one in [37], for the one in [4], for the QAM-based soft-output detector (QBSD) [21], and for the improved QBSD (IQBSD) [21] are provided for comparison. Either the detector in [37] or [4] produces hard-output results, which are sent to subsequent Viterbi decoder [35] to produce decoded information bits. The algorithm in [4] achieves exactly the MLD, and, the curve associated with ‘MLD’ is

**Algorithm 3** The List Administration for Line 18 of Algorithm 2

```

input :  $\beta^{\text{MAP}}, \mathbf{z}^{\text{MAP}}, \overline{\beta}_\ell^{\text{MAP}}, \ell \in \mathcal{I}^I \cup \mathcal{I}^Q,$ 
          $\lambda^{\text{MAP}}, \mathbf{x}^{\text{MAP}},$  and  $\overline{\lambda}_\ell^{\text{MAP}}, \forall \ell$ 
output:  $\lambda^{\text{MAP}}, \mathbf{x}^{\text{MAP}},$  and  $\overline{\lambda}_\ell^{\text{MAP}}, \ell \in \mathcal{I}^A \cup \mathcal{I}^I \cup \mathcal{I}^Q$ 

1 if  $\beta^{\text{MAP}} < \lambda^{\text{MAP}}$  then
2   for each  $\ell \in \mathcal{I}^A$  do
3     if  $z_\ell^{\text{MAP}} \neq x_\ell^{\text{MAP}}$  then
4        $\overline{\lambda}_\ell^{\text{MAP}} = \lambda^{\text{MAP}}$ 
5     end
6   end
7   for each  $\ell \in \mathcal{I}^I \cup \mathcal{I}^Q$  do
8     if  $z_\ell^{\text{MAP}} \neq x_\ell^{\text{MAP}}$  then
9       if  $\overline{\beta}_\ell^{\text{MAP}} < \lambda^{\text{MAP}}$  then
10         $\overline{\lambda}_\ell^{\text{MAP}} = \overline{\beta}_\ell^{\text{MAP}}$ 
11       else
12         $\overline{\lambda}_\ell^{\text{MAP}} = \lambda^{\text{MAP}}$ 
13       end
14     else
15       if  $\overline{\beta}_\ell^{\text{MAP}} < \overline{\lambda}_\ell^{\text{MAP}}$  then
16         $\overline{\lambda}_\ell^{\text{MAP}} = \overline{\beta}_\ell^{\text{MAP}}$ 
17       end
18     end
19   end
20    $\lambda^{\text{MAP}} = \beta^{\text{MAP}}, \mathbf{x}^{\text{MAP}} = \mathbf{z}^{\text{MAP}}$ 
21 else
22   for each  $\ell \in \mathcal{I}^A$  do
23     if  $z_\ell^{\text{MAP}} \neq x_\ell^{\text{MAP}}$  and  $\beta^{\text{MAP}} < \overline{\lambda}_\ell^{\text{MAP}}$  then
24        $\overline{\lambda}_\ell^{\text{MAP}} = \beta^{\text{MAP}}$ 
25     end
26   end
27   for each  $\ell \in \mathcal{I}^I \cup \mathcal{I}^Q$  do
28     if  $z_\ell^{\text{MAP}} \neq x_\ell^{\text{MAP}}$  then
29       if  $\overline{\beta}_\ell^{\text{MAP}} < \overline{\lambda}_\ell^{\text{MAP}}$  then
30         $\overline{\lambda}_\ell^{\text{MAP}} = \overline{\beta}_\ell^{\text{MAP}}$ 
31       end
32     else
33       if  $\overline{\beta}_\ell^{\text{MAP}} < \overline{\lambda}_\ell^{\text{MAP}}$  then
34         $\overline{\lambda}_\ell^{\text{MAP}} = \overline{\beta}_\ell^{\text{MAP}}$ 
35       end
36     end
37   end
38 end

```

therefore associated with the algorithm in [4]. Besides, the algorithm in [37] performs close to the MLD when the channel is *i.i.d.* Rayleigh fading, i.e., the channel in the simulation Scenario #1. But, as reported by [4], the algorithm in [37] performs poorly in comparison with the MLD when the channel is correlated. The compared SO detectors do not consider the *a priori* bit LLR in their algorithms; and, only the results after  $I = 1$  iteration of detection and decoding are illustrated. Note that for the purpose of complexity reduction, the QBSD

TABLE III  
SCENARIOS FOR COMPUTER SIMULATIONS

Scenario	Antenna Configuration ( $N_t, N_r$ )	Symbol Modulation	Code Rate	Information Bits in a Frame	Channel Correlation
Scenario #1	(8,4)	64-QAM	1/2	1,350	0
Scenario #2	(8,4)	64-QAM	2/3	1,800	0
Scenario #3	(8,4)	64-QAM	1/2	1,350	0.4
Scenario #4	(8,4)	16-QAM	1/2	1,050	0

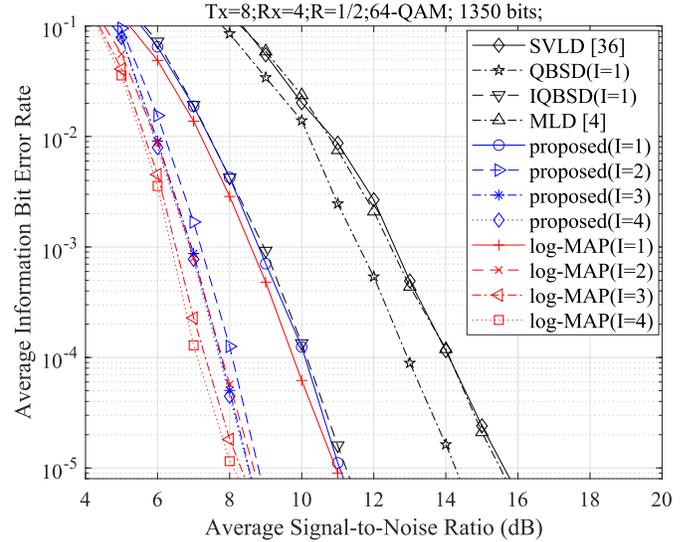


Fig. 3. Information BER curves associated with several detectors for simulation Scenario #1 in Table III.

visits very few lattice points of  $(k, s)$ . When the worst case happens, the set of visited lattice points may contains no  $\mathbf{x}$  that has  $x_\ell = +1$  or  $-1$  in its  $\ell$ -th bit. Under such condition, the QBSD is said to suffer from the bit vacancy problem [18] and performs poorly as shown by Fig. 3. Please refer to the sets  $\mathcal{X}_\ell^{+1}$  and  $\mathcal{X}_\ell^{-1}$  in (7) for comparison. The IQBSD overcomes the bit vacancy problem by visiting bit flipped lattice points, in addition to the original lattice points visited by QBSD; and, the IQBSD performs nearly the same as our proposed detector.

More BER and FER comparisons of our proposed and log-MAP detectors for different simulation scenarios are given in Figures 4 and 5. For each iteration of the SISO detection in the IDD, our proposed detector performs inferiorly to the log-MAP detector by less than 0.3 dB at BER level  $10^{-5}$  or FER level  $10^{-2}$ . Be aware that our proposed detector produce exactly the same  $L_D(x_\ell)$  and  $L_E(x_\ell)$  as the max-log-MAP detector. This fact has also been verified through the simulations. Accordingly, we may say that our proposed max-log-MAP detector performs inferiorly to the log-MAP detector by less than 0.3 dB.

#### D. Complexity Analysis for Our Algorithms

The computational complexity of our Algorithms 1 and 2 are studied. We assume that the channel is in slow fading. Under such scenario, the preprocessing in lines 2-4 of Algorithm 1 is conducted once for the transmission of one data frame. The processing of multiple payload vectors obviously is

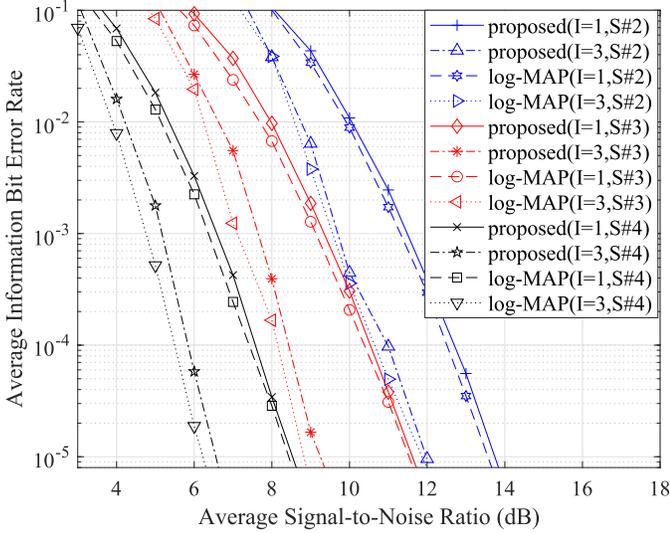


Fig. 4. Information BER comparisons of the log-MAP detector and our proposed Algorithm 2 for the Scenarios #2, #3, and #4 in Table III.

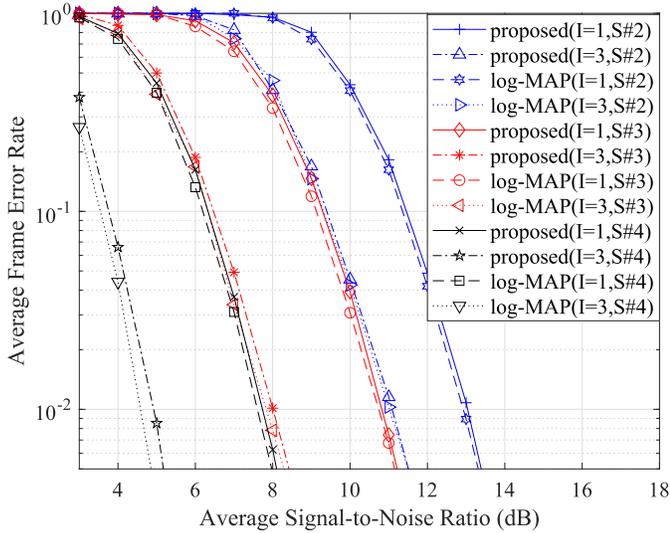


Fig. 5. Information FER comparisons of the log-MAP detector and our proposed Algorithm 2 for the Scenarios #2, #3, and #4 in Table III.

more complex than the channel preprocessing. Therefore, only the number of real-valued multiplications (RMULs) required to process each payload vector  $\mathbf{u}$  in one iteration of the SO/SISO detection scheme is counted.

For Algorithm 1, the product  $\mathbf{q}_k^H \mathbf{u}$  associated with each  $k$  in line 7 requires  $N_r$  complex-valued multiplications or  $4N_r$  RMULs. Because  $\Re\{s\}$  is a lattice point, the  $r_k \Re\{s\}$  can be obtained by applying shift operations to  $r_k$ ; thus, computation of  $r_k \Re\{s\}$  does not involve RMULs. The computation of  $\rho(k, s)$  for each  $(k, s)$  in line 10 requires 4 RMULs. The computation of  $\sum_{x_i} x_i L_A(x_i)$  does not involve RMULs, because each  $x_i \in \{+1, -1\}$ . The list administration in lines 11-29 involves comparisons and value assignments; and, no RMUL operation is required. Thus, the Algorithm 1 requires  $4N_t N_r + 4N_t M$  RMULs.

TABLE IV

THE NUMBERS OF RMULS REQUIRED BY THE SO/SISO DETECTION ALGORITHMS IN ONE ITERATION

Scenario ( $N_t, N_r, M$ )	Proposed Algorithm 1	Proposed Algorithm 2	SORBD [20]	QBSD [21]	IQBSD [21]	max-log-MAP [21]
(8, 4, 64)	2,176	272	1,433	289	529	12,297
(8, 4, 16)	640	208	663	287	447	3,079
(4, 2, 4)	96	56	140	96	136	196
(16, 4, 64)	4,352	544	2,858	570	1,050	24,586

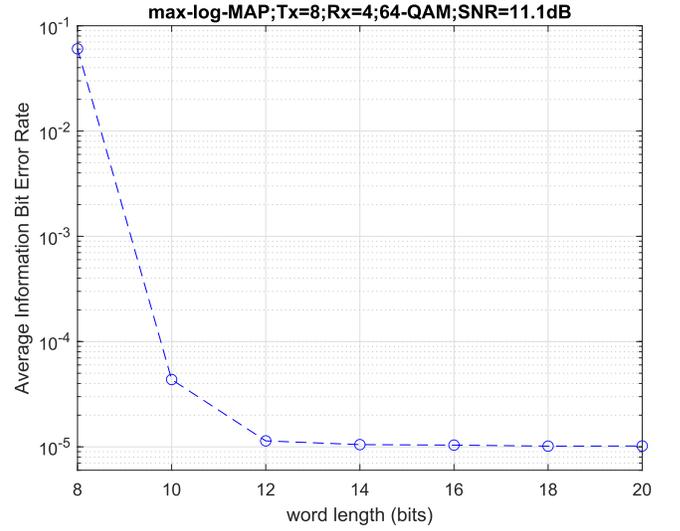


Fig. 6. Fixed-point simulation results for our proposed Algorithm 2 after one detection and decoding iteration.

On the other hand, for Algorithm 2, the product  $\mathbf{q}_k^H \mathbf{u}$  for each  $k$  in line 5 requires  $4N_r$  RMULs. The computation of either  $\beta^{I, \text{MAP}}$  or  $\beta^{Q, \text{MAP}}$  associated with each  $k$  in line 6 requires  $\sqrt{M}$  RMULs, because there are  $\sqrt{M}$  possible values for either  $\Re\{s\}$  or  $\Im\{s\}$ . The computation of  $\rho^A(k, s)$  and then  $\beta^{\text{MAP}}$  in line 7 requires additional 2 RMULs. The computation of  $\beta_e^{I, \text{MAP}}$  does not involve RMULs, because the  $\rho^I(k, s), \forall s$ , are generated for computing  $\beta^{I, \text{MAP}}$ . In totality, the Algorithm 2 requires  $4N_t N_r + 2N_t \sqrt{M} + 2N_t$  RMULs.

Complexity comparisons of our proposed algorithms with some other algorithms are given in Table IV. The number of RMULs required by the QBSD or IQBSD is dominated by  $6N_t N_r$  [21]. The number of RMULs required by the max-log-MAP detector is claimed to be dominated by  $N_t N_r M$  [21]. However, the number of RMULs required by our Algorithm 2 is dominated by only  $4N_t N_r$ , because we apply (10) to obtain the scalar  $y_k, \forall k$ , and the following computations are all relevant to  $y_k, \forall k$ . Accordingly, compared with the QBSD and IQBSD, our proposed Algorithm 2 not only performs with lower BER but also requires fewer RMULs.

The complexity comparison between our two algorithms can be discussed in an alternative way. Our Algorithm 1 visits  $N_t M$  lattice points, while Algorithm 2 visits  $N_t (2\sqrt{M} + 1)$  lattice points. For large  $M$ , Algorithm 2 requires roughly  $2/\sqrt{M}$  complexity that of Algorithm 1. One last comment is that our Algorithm 2 does not perform division, exponential, and

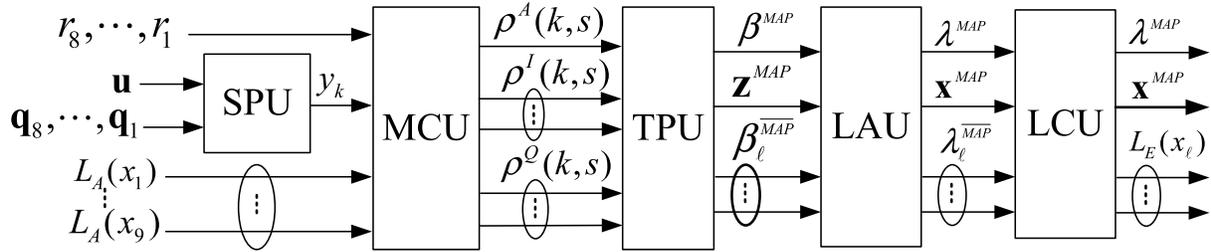


Fig. 7. Our proposed hardware architecture for SISO detection of SM signals for the antenna configuration  $(N_t, N_r) = (8, 4)$ .

squared root operations. This property makes our Algorithm 2 feasible for hardware implementation.

Additionally, for the first iteration, Algorithm 2 can ignore  $L_A(x_\ell), \forall \ell$ , because  $L_A(x_\ell) = 0$ . Fast enumeration and bit flipping [33] can be applied to compute  $\beta^{I,MAP}$ ,  $\beta^{Q,MAP}$ ,  $\beta_\ell^{I,MAP}$ , and  $\beta_\ell^{Q,MAP}$ . The total RMULs required by this fast SO Algorithm 2 is reduced to  $4N_t N_r + 2N_t \log_2 M + 2N_t$  RMULs.

#### IV. THE PROPOSED ARCHITECTURES AND IMPLEMENTATION RESULTS

Following the computational steps in Algorithm 2, we propose our hardware architectures to SISO detect SM signals for the Scenario #1 in Table III, i.e., the scenario  $(N_t, N_r, M) = (8, 4, 64)$ . The floating-point simulations in Fig. 3 reveal that after one detection and decoding iteration, our Algorithm 2 leads to decoded information BER  $10^{-5}$  at SNR 11.1 dB. The corresponding fixed-point simulations over different word lengths were conducted, the results of which are illustrated in Fig. 6. Although not provided here, similar floating- and fixed-point simulation results reveal that after three detection and decoding iterations, our Algorithm 2 leads to information BER  $10^{-5}$  at SNR 8.1 dB. Thus, by fixing the word length of each real number to be 12 bits, we design a SISO detector leading to information BER  $10^{-5}$  after one/three detection and decoding iterations when SNR is no greater than 11.1/8.1 dB.

##### A. Proposed Hardware Architectures

The hardware architecture of our SISO detector is illustrated in Fig. 7. This architecture is responsible for the payload processing of each vector  $\mathbf{u}$ , i.e., the operations after line 4 of Algorithm 2. The channel preprocessing, indicated by the operations in line 2 of Algorithm 2, can be accomplished by the triangulation block depicted in [4, Fig. 4]. With input  $\mathbf{h}_k$  followed by  $\mathbf{I}$ , the triangulation block produces  $\mathbf{q}_k$  and  $r_k$ , respectively. These values are saved in a memory unit and are retrieved when the detector starts to process each payload vector  $\mathbf{u}$ . The architecture in Fig. 7 contains five major units, i.e., signal projection unit (SPU), metric computation unit (MCU), tree pruning unit (TPU), list administration unit (LAU), and LLR computation unit (LCU).

The SPU is comprised of 4 complex-valued multipliers and is for computing the inner product  $y_k = \mathbf{q}_k^H \mathbf{u}$ , indicated by line 5 of Algorithm 2. Each complex-valued multiplier is comprised of 3 real-valued multipliers and 5 adders [38].

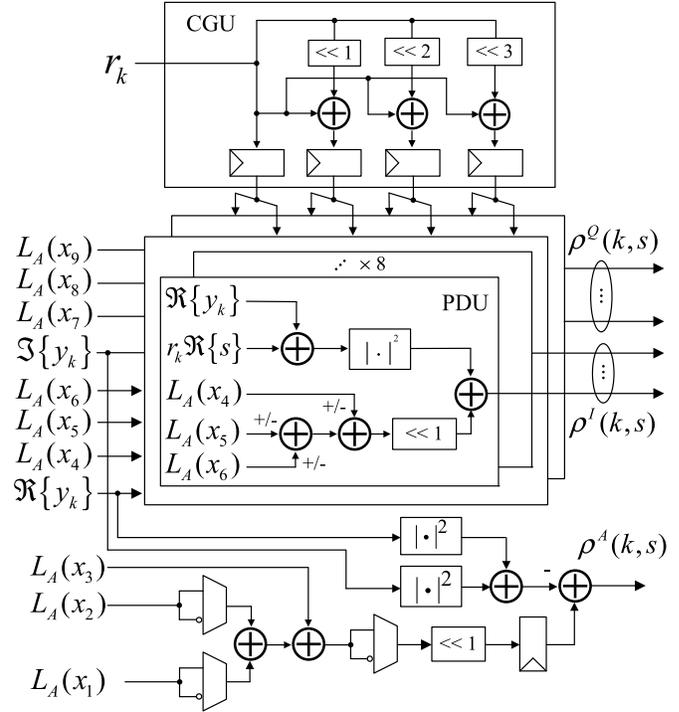


Fig. 8. The MCU in Fig. 7.

The input  $\mathbf{u}$  stays unchanged for  $N_t = 8$  clock cycles, but  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_8$  are sent sequentially to SPU. The  $y_k, \forall k$ , are delivered sequentially at the output of SPU.

The MCU, depicted in Fig. 8, is for computing  $\rho^A(k, s)$ ,  $\rho^I(k, s)$ , and  $\rho^Q(k, s)$  in (18), with fixed  $k$  and  $\forall s \in \mathcal{S}$ . The lattice points  $r_k \Re\{s\}$  and  $r_k \Im\{s\}, \forall s$ , are the same. They are generated by one candidate generation unit (CGU). There are 8 partial Euclidean distance units (PDUs) for producing  $\rho^I(k, s), \forall s$ , and 8 PDUs for producing  $\rho^Q(k, s), \forall s$ . The parallel outputs of MCU include one value of  $\rho^A(k, s)$ , 8 values of  $\rho^I(k, s)$ , and 8 values of  $\rho^Q(k, s)$ .

The TPU, depicted in Fig. 9, is for computing the partial MAP information  $\mathcal{L}^{\text{part}}$  in (29). As revealed by (20), the  $\beta^{I,MAP}$  and  $\mathbf{z}^{I,MAP}$  are computed from  $\rho^I(k, s), \forall s$ . Then, as revealed by (23),  $\beta_\ell^{I,MAP}, \ell \in \mathcal{I}^I$ , are computed from those  $\rho^I(k, s), \forall s$ , that satisfy  $x_\ell = \frac{\rho^I(k, s)}{z_\ell^{I,MAP}}$ . Each  $z_\ell^{I,MAP}$  or equivalent  $\frac{\rho^I(k, s)}{z_\ell^{I,MAP}}$  is used to select appropriate values from  $\rho^I(k, s), \forall s$ , according to the mapping information in Fig. 2. For the similar design,  $\beta_\ell^{Q,MAP}, \mathbf{z}_\ell^{Q,MAP}$ , and  $\beta_\ell^{Q,MAP}, \forall \ell$ , are

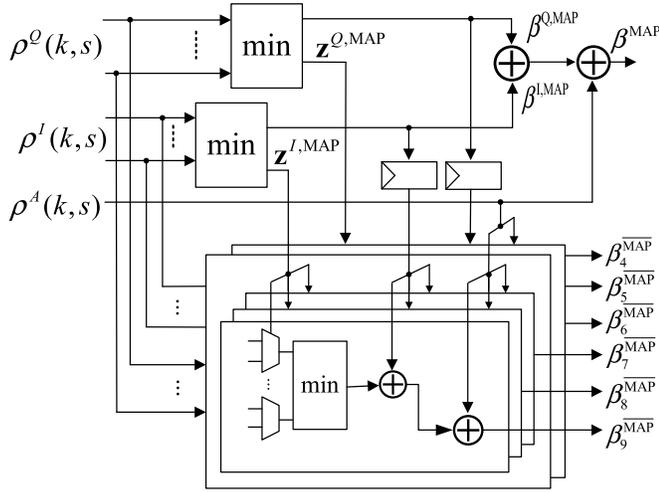


Fig. 9. The TPU in Fig. 7.

produced. Next, according to line 7 of Algorithm 2,  $\beta^{\text{MAP}}$  and  $\mathbf{z}^{\text{MAP}}$  are computed. According to lines 9 and 12 of Algorithm 2, the  $\beta_\ell^{\text{MAP}}, \ell \in \mathcal{I}^I \cup \mathcal{I}^Q$  are produced at the outputs of TPU.

Finally, according to Algorithm 3, the LAU in Fig. 10 is comprised of comparators, multiplexers, and registers for generating the most recent  $\lambda^{\text{MAP}}, \mathbf{x}^{\text{MAP}}$ , and  $\lambda_\ell^{\text{MAP}}, \forall \ell$ . The circuits to update  $\lambda_\ell^{\text{MAP}}, \forall \ell \in \mathcal{I}^A$ , are different from the circuits to update  $\lambda_\ell^{\text{MAP}}, \forall \ell \in \mathcal{I}^I \cup \mathcal{I}^Q$ . They are designed separately in Fig. 10.

The very last LCU in Fig. 7 is for computing  $L_D(x_\ell)$  and  $L_E(x_\ell), \forall \ell$ , as indicated by line 21 of Algorithm 2. Besides, when  $L_A(x_\ell) = 0, \forall \ell$ , the  $\mathbf{x}^{\text{MAP}}$  in (14) is reduced to the MLD solution [4, eq. (2)], i.e.,

$$\mathbf{x}^{\text{MLD}} = \arg \min_{\mathbf{x}} \|\mathbf{u} - \mathbf{h}_k s\|^2. \quad (30)$$

For the first iteration of the SISO detection in IDD, the LCU outputs the MLD solution  $\mathbf{x}^{\text{MLD}} = \mathbf{x}^{\text{MAP}}$ . Thus, our architecture is able to provide both the MLD and max-log-MAP detection outputs.

The payload processing of each  $\mathbf{u}$  for our architecture in Fig. 7 is scheduled in Fig. 11. Our architecture produces sequentially  $y_k, \rho^A(k,s), \rho^I(k,s), \rho^Q(k,s), \beta^{\text{MAP}}, \beta_\ell^{\text{MAP}}, \lambda_\ell^{\text{MAP}}$ , and  $\lambda^{\text{MAP}}$  for  $k = 1, 2, \dots, 8$ . The pipeline structure in our architecture allows the final  $\mathbf{x}^{\text{MAP}}, \lambda^{\text{MAP}}$ , and  $\lambda_\ell^{\text{MAP}}, \forall \ell$ , to be available at the 12-th clock cycle. The  $L_E(x_\ell), \forall \ell$ , are delivered at the 13-th clock cycle.

Modification to the architecture in Fig. 7 may lead to an hardware detector with higher throughput rate. For example, the architecture in Fig. 12 is roughly comprised of two parallel architectures in Fig. 7. The top LAU updates and produces the SISO information set  $\mathcal{L}^{\text{MAP}}$  for  $k = 1, 2, 3$ , and 4 at clock cycles 5-8. The bottom LAU outputs  $\mathcal{L}^{\text{MAP}}$  for  $k = 5, 6, 7$ , and 8 at clock cycles 5-8. The two  $\mathcal{L}^{\text{MAP}}$ 's obtained at clock cycle 8 can be processed by a list administration unit, i.e., the LAU1, that is similar to the LAU to produce a new  $\mathcal{L}^{\text{MAP}}$  at the 9-th clock cycle, because the update of  $\mathcal{L}^{\text{MAP}}$  using  $\mathcal{L}^{\text{temp}}$  and old  $\mathcal{L}^{\text{MAP}}$  is similar to the update of  $\mathcal{L}^{\text{MAP}}$

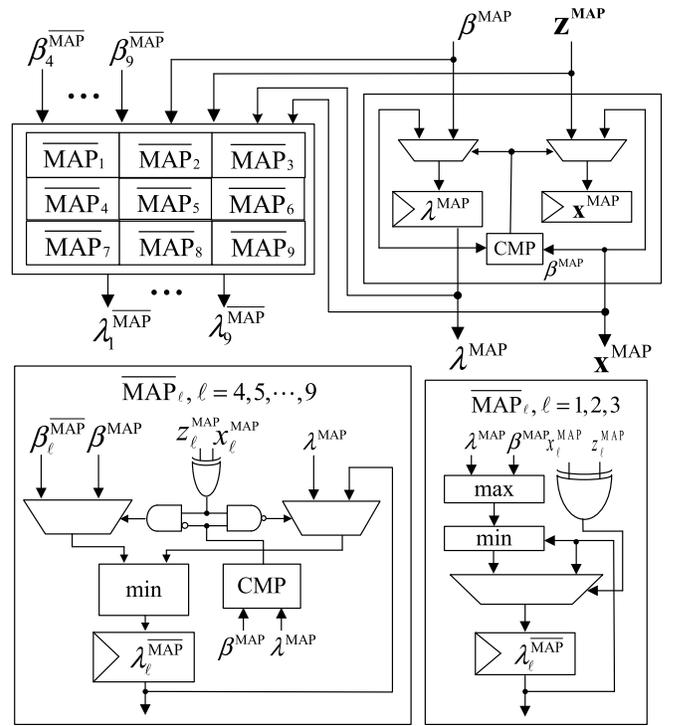


Fig. 10. The LAU in Fig. 7.

Clock	1	2	3	4	5	6	7	8	9	10	11	12	13
SPU Output	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$	$y_8$					
MCU Output		$\rho^I$	$\rho^I$	$\rho^I$	$\rho^I$	$\rho^I$	$\rho^I$	$\rho^I$					
TPU Output			$\beta^{\text{MAP}}$	$\beta^{\text{MAP}}$	$\beta^{\text{MAP}}$	$\beta^{\text{MAP}}$	$\beta^{\text{MAP}}$	$\beta^{\text{MAP}}$	$\beta^{\text{MAP}}$	$\beta^{\text{MAP}}$			
LAU Output					$\lambda_\ell^{\text{MAP}}$								
LCU Output													$L_E(x_\ell)$

 Fig. 11. Timing chart for payload processing of the hardware architecture in Fig. 7 to process each payload vector  $\mathbf{u}$ .

using the two old  $\mathcal{L}^{\text{MAP}}$ 's. Compared with the one in Fig. 7, the architecture in Fig. 12 is able to process each  $\mathbf{u}$  every 4 clock cycles at the cost of roughly two times of hardware expense.

Based on the highly modularized architecture in Fig. 7, we can design max-log-MAP detector for detecting SM coded signals with arbitrary  $N_t$  transmit antennas,  $N_r$  receive antennas, and  $M$ -QAM symbols.

- For arbitrary  $N_t$ , the architecture in Fig. 7 can be used to search through each possible antenna index per clock cycle. The number of processing cycles required by the architecture becomes  $N_t$ . We can use more functional units as in Fig. 12 to obtain an architecture to process each payload vector with fewer processing cycles. Our design of the two architectures in Figs. 7 and 12 roughly

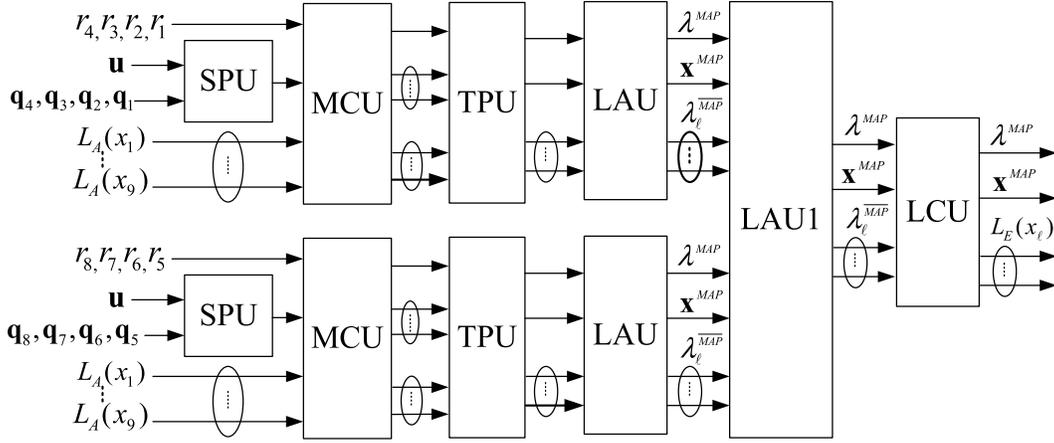


Fig. 12. An architecture that is different from the one in Fig. 7 but provides higher data throughput rate.

implies that

$$\begin{aligned} & (\text{processing cycles}) \\ & \times (\text{cost of the max-log-MAP detector}) \\ & \propto (\text{number of transmit antennas}). \end{aligned} \quad (31)$$

The cost of max-log-MAP detector is traded for processing cycles (or throughput rate) when the number of transmit antennas is fixed.

- For arbitrary  $N_r$  receive antennas, we only design the number of complex-valued multipliers in SPU equal to  $N_r$ ; and, the remaining architecture in Fig. 7 does not need to be changed.
- For arbitrary  $M$ -QAM symbols, the number of PDUs in the MCU should be equal to  $2\sqrt{M}$ . The TPU should be tailored to provide  $\beta_\ell^{\text{MAP}}$ ,  $\ell \in \mathcal{I}^I \cup \mathcal{I}^Q$ . Additionally, the LAU needs to be slightly tailored to include  $\log_2 M$  cells of  $\overline{\text{MAP}}_\ell$ 's to store  $\lambda_\ell^{\text{MAP}}$ ,  $\ell \in \mathcal{I}^I \cup \mathcal{I}^Q$ .

### B. Implementation Results

After logic synthesis and automatic placement and routing (APR), gate-level simulations are performed with test patterns to generate net/register switching activity files (in SAIF or TCF format). The synthesis tool is then used to read these files and to report the power consumption of the designed hardware circuits based on the timing/power models defined in the standard cell library (in.lib format). The implementation results of our architecture in Fig. 7 are given in Fig. 13 and Table V. The SPU requires 13.2K gates, because it includes twelve 12-bit multipliers. The MCU requires 19.3K gates, because it includes parallel processing elements to produce 8 metric values of  $\rho^I(k, s)$ 's, 8 metric values of  $\rho^Q(k, s)$ 's, and one  $\rho^A(k, s)$ . Other functional units are basically comprised of registers, multiplexers, and comparators. No divider is used. Note that the use of 3 real multipliers and 5 adders [38], rather than 4 real multipliers and 2 adders, for one complex multiplier leads to few gates and lower power consumption in our circuits at the cost of slightly larger latency. Nonetheless, the implementation results reveal that the critical path of our design occurs in the TPU. This fact indicates that we achieve

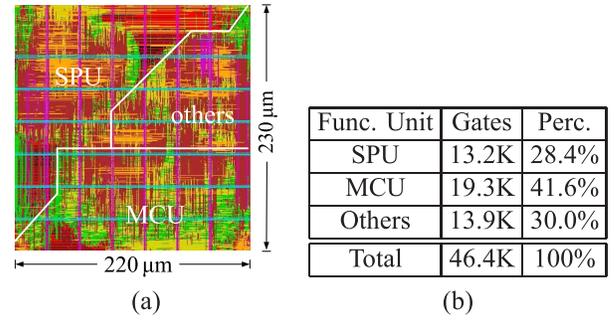


Fig. 13. (a) Chip layout and (b) gate counts of the functional units for the architecture in Fig. 7.

fewer gates and less power consumption without sacrificing circuit latency. Clearly, our proposed architecture is very hardware-efficient. The results in Table V also reveal that our architecture in Fig. 12 requires roughly two times of gates that required by the architecture in Fig. 7.

To our knowledge, our hardware architectures are the first ones to SISO detect the coded SM signals. From Table V, our architectures provide lower throughput rate than the two hard-output detectors in [4] and [37]. However, as revealed by Fig. 3, the nearly or exactly MLD detector followed by Viterbi decoder may lead to decoder BER roughly equal to only  $7.5\text{E-}3$  at SNR 11.1 dB. Recall from the first paragraph of Section IV that the our SO/SISO detector leads to decoder BER  $10^{-5}$  after one/three detection and decoding iterations when the SNR is no larger than 11.1/8.1 dB. It might be unfair to compare the hard-output and soft-output hardware architectures that function differently. Nevertheless, we define

$$\text{BER Weighted Gate Count Efficiency} = \frac{\text{Gate Count}}{\text{Throughput Rate}} \times \frac{\text{Decoder BER}}{10^{-5}}. \quad (32)$$

The figure of merit in (32) considers not only the hardware efficiency but also the detection quality. Due to the much lower decoder BER, our proposed architectures have better BER weighted gate count efficiency than the two hard-output detectors in [4] and [37].

TABLE V  
COMPARISONS OF THE HARDWARE ARCHITECTURES

	[37]	[4]	Proposed Architecture #1 (Fig. 7)	Proposed Architecture #2 (Fig. 12)
Detection Scheme	Modified SVLD	MLD	max-log MAP	max-log MAP
Detection Output	Hard Output	Hard Output	SISO	SISO
Antennas (Tx × Rx)	8 × 4	8 × 4	8 × 4	8 × 4
Modulation (-QAM)	64	64	64	64
Channel Preprocessing	not required	included	not included	not included
Technology (nm)	180	90	40	40
Gate Count (KGE)	87.4	70.5	46.4	94.2
Frequency (MHz)	286	384.6	400	400
Processing Cycles	3	8	8	4
Power Consumption (mW)	121.3	-	46	94.8
<sup>1</sup> Throughput Rate (Mbps)	3,861	973.52	450	900
Decoder BER at SNR 11.1 dB	$7.5 \times 10^{-3}$	$7.5 \times 10^{-3}$	$10^{-5}$	$10^{-5}$
BER Weighted Gate Count Efficiency (KGE/Mbps)	16.9775	54.3132	0.1031	0.1047

$$^1\text{Throughput Rate} = \frac{\text{Technology}}{40 \text{ nm}} \times \text{Frequency} \times \frac{\text{bits per SM mapping}}{\text{Processing Cycles}}$$

## V. CONCLUSION

We study the low-complexity implementation of the max-log-MAP detector for SISO detection of coded SM signals in the IDD receiver. Several tricks have been utilized to help us to arrive at the low-complexity Algorithm 2. First, by applying (10) to the original  $L_D(x_\ell)$  in (7), we obtain a new expression for the  $L_D(x_\ell)$  in (11). This leads the Algorithm 2 to process each scalar  $y_k$ , rather than  $N_r \times 1$  vector  $\mathbf{u}$  or  $\mathbf{h}_k$ . Second, we regard the max-log-MAP detector as a tree search scheme. By expressing the metric  $\rho(k, s)$  in (18) in terms of information associated with the antenna index  $k$ , the real-part of symbol  $s$ , and imaginary-part of symbol  $s$ , Algorithm 2 can prune unnecessary lattice points during its tree search. Only  $2\sqrt{M} + 1$  lattice points, rather than  $M$  lattice points, are visited during the  $k$ -th tree traversal. Third, motivated by the STS algorithm, our Algorithm 2 computes each  $\mathcal{L}^{\text{part}}$  in each iteration. The most recent MAP information  $\mathcal{L}^{\text{MAP}}$  is iteratively updated by each  $\mathcal{L}^{\text{part}}$  through the list administration. Such operation allows our Algorithm 2 to retain the max-log-MAP solution in the end. Surprisingly, our Algorithm 2 requires fewer RMULs than sub-optimal soft-output detectors in the general scenarios. The low-complexity Algorithm 2 then leads us to design low-complexity hardware architectures in Figs. 7 and 12 for SISO detection of coded SM signals. Finally, the highly modularized design of our hardware architecture allows us to design max-log-MAP detectors for coded SM signals with arbitrary  $N_t$ ,  $N_r$ , or  $M$  values. For

our design of the hardware architecture of the max-log-MAP detector, the relationship among the number of transmit antennas, processing cycles, and hardware cost is roughly given in (31).

## ACKNOWLEDGMENT

The authors greatly appreciate the Taiwan Semiconductor Research Institute (TSRI) for providing technical support over the duration of this work.

## REFERENCES

- [1] M. Wen *et al.*, "A survey on spatial modulation in emerging wireless systems: Research progresses and applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 9, pp. 1949–1972, Sep. 2019.
- [2] R. Y. Mesleh, H. Haas, S. Sinanovic, C. W. Ahn, and S. Yun, "Spatial modulation," *IEEE Trans. Veh. Technol.*, vol. 57, no. 4, pp. 2228–2241, Jul. 2008.
- [3] J. Jeganathan, A. Ghrayeb, and L. Szczecinski, "Spatial modulation: Optimal detection and performance analysis," *IEEE Commun. Lett.*, vol. 12, no. 8, pp. 545–547, Aug. 2008.
- [4] T.-H. Liu, Y.-Z. Ye, C.-K. Huang, C.-E. Chen, Y.-T. Hwang, and Y.-S. Chu, "A low-complexity maximum likelihood detector for the spatially modulated signals: Algorithm and hardware implementation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 66, no. 11, pp. 1820–1824, Nov. 2019.
- [5] P. Yang, M. Di Renzo, and Y. Xiao, "Design guidelines for spatial modulation," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 6–26, 1st Quart., 2014.
- [6] C. Xu, S. Sugiura, S. X. Ng, P. Zhang, L. Wang, and L. Hanzo, "Two decades of MIMO design tradeoffs and reduced-complexity MIMO detection in near-capacity systems," *IEEE Access*, vol. 5, pp. 18564–18632, 2017.
- [7] R. Mesleh, M. D. Renzo, H. Haas, and P. M. Grant, "Trellis coded spatial modulation," *IEEE Trans. Wireless Commun.*, vol. 9, no. 7, pp. 2349–2361, Jul. 2010.
- [8] E. Basar, U. Aygolu, E. Panayirci, and H. V. Poor, "New trellis code design for spatial modulation," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2670–2680, Aug. 2011.
- [9] M. Koca and H. Sari, "Bit-interleaved coded spatial modulation," in *Proc. IEEE 23rd Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sydney, NSW, Australia, Sep. 2012, pp. 1949–1954.
- [10] C. Vladeanu, "Turbo trellis-coded spatial modulation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Anaheim, CA, USA, Dec. 2012, pp. 4024–4029.
- [11] Z.-Y. Wang, P.-Y. Tsai, Y.-H. Hwang, and I.-W. Lai, "A double K-best viterbi-sphere decoder for trellis-coded generalized spatial modulation with multiple code rates," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Brighton, U.K., May 2019, pp. 1547–1551.
- [12] C. Li, Y. Cheng, Y. Zhang, and Y. Huang, "Low-complexity soft-output detectors for LDPC coded spatial modulation systems," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2015, pp. 1–6.
- [13] D. Feng, H. Xu, J. Zheng, and B. Bai, "Nonbinary LDPC-coded spatial modulation," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2786–2799, Apr. 2018.
- [14] L. Wang, C. Liang, Z. Yang, and X. Ma, "Two-layer coded spatial modulation with block Markov superposition transmission," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 643–653, Feb. 2016.
- [15] L. Liu, "Energy-efficient soft-input soft-output signal detector for iterative MIMO receivers," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 61, no. 8, pp. 2422–2432, Aug. 2014.
- [16] C. Studer and H. Bölskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4827–4842, Oct. 2010.
- [17] E. M. Witte, F. Borlenghi, G. Ascheid, R. Leupers, and H. Meyr, "A scalable VLSI architecture for soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 9, pp. 706–710, Sep. 2010.
- [18] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
- [19] J. Chen, J. Hu, and G. E. Sobelman, "Stochastic iterative MIMO detection system: Algorithm and hardware design," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 4, pp. 1205–1214, Apr. 2015.

- [20] Q. Tang, Y. Xiao, P. Yang, Q. Yu, and S. Li, "A new low-complexity near-ML detection algorithm for spatial modulation," *IEEE Wireless Commun. Lett.*, vol. 2, no. 1, pp. 90–93, Feb. 2013.
- [21] C. Li, J. Wang, Y. Cheng, and Y. Huang, "Low-complexity soft-decision aided detectors for coded spatial modulation MIMO systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–11, Dec. 2016.
- [22] G. Xiang, G. Li, Y. Xu, Y. Ou, J. Wang, and H. Zhu, "A low-complexity soft output detection algorithm for spatial modulation systems," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2017, pp. 33–37.
- [23] Z. Zhang, C. Gong, H. Li, Y. Dong, X. Wang, and X. Dai, "Soft-input soft-output detection via expectation propagation for massive spatial modulation MIMO systems," *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1173–1177, Apr. 2021.
- [24] S. Sugiura, C. Xu, S. X. Ng, and L. Hanzo, "Reduced-complexity iterative-detection-aided generalized space-time shift keying," *IEEE Trans. Veh. Technol.*, vol. 61, no. 8, pp. 3656–3664, Oct. 2012.
- [25] B. Zheng, X. Wang, M. Wen, and F. Chen, "Soft demodulation algorithms for generalized spatial modulation using deterministic sequential Monte Carlo," *IEEE Trans. Wireless Commun.*, vol. 65, no. 6, pp. 3953–3967, Jun. 2017.
- [26] B. Zheng, M. Wen, F. Chen, N. Huang, F. Ji, and H. Yu, "The K-best sphere decoding for soft detection of generalized spatial modulation," *IEEE Trans. Commun.*, vol. 65, no. 11, pp. 4803–4816, Nov. 2017.
- [27] K.-C. Lai, H.-Y. Su, D.-G. Peng, and S.-Z. He, "Detection algorithm for single-carrier spatial modulation signals in frequency-selective fading channels," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13341–13356, Nov. 2020.
- [28] L. Xiao *et al.*, "An improved soft-input soft-output detector for generalized spatial modulation," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 30–34, Jan. 2016.
- [29] M. Shabany, R. Doostnejad, M. Mahdavi, and P. G. Gulak, "A 38 pj/b optimal soft-MIMO detector," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 9, pp. 1062–1066, Sep. 2017.
- [30] M. Mahdavi, O. Edfors, V. Owall, and L. Liu, "Angular-domain massive MIMO detection: Algorithm, implementation, and design tradeoffs," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 6, pp. 1948–1961, Jun. 2020.
- [31] T.-H. Liu, S.-L. Wang, Y.-J. Lin, Y.-T. Hwang, C.-E. Chen, and Y.-S. Chu, "Fixed-complexity tree search schemes for detecting generalized spatially modulated signals: Algorithms and hardware architectures," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 68, no. 2, pp. 904–917, Feb. 2021.
- [32] C. Studer, A. Burg, and H. Bölcskei, "Soft-output sphere decoding: Algorithms and VLSI implementation," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 2, pp. 290–300, Feb. 2008.
- [33] L. Liu, J. Lofgren, P. Nilsson, and V. Öwall, "VLSI implementation of a soft-output signal detector for multimode adaptive multiple-input multiple-output systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 21, no. 12, pp. 2262–2273, Dec. 2013.
- [34] S. Yang and L. Hanzo, "Fifty years of MIMO detection: The road to large-scale MIMOs," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 1941–1988, 4th Quart., 2015.
- [35] J. Proakis and M. Salehi, *Digital Communications*, 5th ed. New York, NY, USA: McGraw-Hill, 2008.
- [36] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.
- [37] G. H. Lee and T. H. Kim, "Implementation of a near-optimal detector for spatial modulation MIMO systems," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 63, no. 10, pp. 954–958, Oct. 2016.
- [38] E. E. Swartzlander and H. H. Saleh, "Floating-point implementation of complex multiplication," in *Proc. Conf. Rec. 43rd Asilomar Conf. Signals, Syst. Comput.*, Nov. 2009, pp. 926–929.



**Tsung-Hsien Liu** (Member, IEEE) received the B.S. degree in electrical engineering from the National Taiwan University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1998. In August 1999, he joined the Department of Communications Engineering, National Chung Cheng University, Chiayi, Taiwan, as a Faculty Member, where he is currently a Professor. Since August 2020, he has been the Chairperson with the Department of Communications Engineering, National Chung Cheng University. His research interests include applying signal processing techniques to wireless communications, with emphasis on developing fast implementation algorithms, designing baseband hardware architectures, and analyzing statistical performance for the multiple antenna baseband receivers. He served as the Chairperson for the Tainan Chapter, IEEE Communication Society, from 1998 to 1999.



**Ting-Xu Jiang** received the B.S. and M.S. degrees in electrical engineering from the National Chung Cheng University, Chiayi, Taiwan, in 2019 and 2021, respectively. Since 2021, he has been an IC Design Engineer at MediaTek Inc., Hsinchu, Taiwan. His research interests include VLSI design and digital signal processing for the communication baseband receivers.



**Ching-Che Chung** (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electronics engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 1997 and 2003, respectively. From 2004 to 2008, he was a Post-Doctoral Researcher at the National Chiao-Tung University, working in system-on-chip design methodologies and high-speed interface circuit design. In August 2008, he joined the Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan, where he is currently a Professor. His research interests include wireless and wireline communication systems, low-power and system-on-a-chip (SoC) design technology, mixed-signal IC design and sensor circuits design, artificial intelligence (AI) chip design, all-digital phase-locked loop, all-digital delay-locked loop, and its applications.



**Yuan-Sun Chu** received the B.S. degree in electrical engineering from Feng Chia University, Taichung, Taiwan, in 1978, and the M.S. and Ph.D. degrees in electrical engineering from Katholieke Universiteit Leuven, Leuven, Belgium, in 1986 and 1991, respectively. He is currently a Professor with the Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan. His research interests include computer architecture, communication IC design, and low-power IC design.