LUND UNIVERSITY

VLSI Implementation of a Soft-Output Signal Detector for Multi-Mode Adaptive MIMO Systems

Liu, Liang; Löfgren, Johan; Nilsson, Peter; Öwall, Viktor

[Link to publication](#)

Total number of authors:
4

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

VLSI Implementation of a Soft-Output Signal Detector for Multi-Mode Adaptive MIMO Systems

Liang Liu, *Member, IEEE,* Johan Löfgren, *Student Member, IEEE,* Peter Nilsson, *Senior Member, IEEE,* and Viktor Öwall , *Member, IEEE*

Abstract—This paper presents a multi-mode soft-output multiple-input multiple-output (MIMO) signal detector that is efficient in hardware cost and energy consumption. The detector is capable of dealing with spatial-multiplexing (SM), space-division-multiple-access (SDMA), and spatial-diversity (SD) signals of $4{\times}4$ antenna and 64-QAM modulation. Implementation-friendly algorithms, which reuse most of the mathematical operations in these three MIMO modes, are proposed to provide accurate soft detection information, i.e., log-likelihood ratio (LLR), with much reduced complexity. A unified reconfigurable VLSI architecture has been developed to eliminate the implementation of multiple detector modules. In addition, several block level technologies, such as parallel metric update and fast bit-flipping, are adopted to enable a more efficient design. To evaluate the proposed techniques, we implemented the triple-mode MIMO detector in a 65-nm CMOS technology. The core area is 0.25 mm^2 with 83.7 K gates. The maximum detecting throughput is 1 Gb/s at 167-MHz clock frequency and 1.2-V supply, which archives the data rate envisioned by the emerging long-term evolution advanced (LTE-A) standard. Under frequency-selective channels, the detector consumes 59.3 pJ, 10.5 pJ, and 169.6 pJ energy per bit detection in SM, SD, and SDMA modes, respectively.

Index Terms—Multiple-input multiple-output (MIMO), signal detector, soft-output, spatial-multiplexing (SM), spatial-diversity (SD), space-division-multiple-access (SDMA), very-large scale integration (VLSI).

I. INTRODUCTION

TO meet the growing demands for better user experience, the International Telecommunication Union (ITU) has released its requirements for next-generation wireless networks, where much higher spectral efficiency, higher coverage, and lower latencies are expected [1]. It has been a broad agreement that enhanced multiple-input multiple-output (MIMO) technologies play an essential role in emerging wireless standards, e.g., IEEE 802.16m (WiMAX Profile 2.0) [2] and 3GPP Long Term Evolution Advanced (3GPP LTE-A) (Release 10) [3], to achieve or exceed the International Mobile Telecommunications Advanced (IMT-A) target.

Cellular systems experience highly dynamic channel conditions, where the signal-to-noise ratio (SNR) and fading properties vary within huge ranges. To guarantee the quality of service (QoS) for users with a specified error rate and data throughput, it is necessary that the system is equipped with multiple MIMO technologies, which are dynamically adapted to the fluctuating channels. This is because single-mode

MIMO schemes have shown their limitations in satisfying such requirements. For example, the widely-used spatial multiplexing (SM) technique [4] suffers from huge performance loss when the spatial channel becomes highly correlated [5]. Currently, extensive discussions are ongoing about the multi-mode adaptive MIMO schemes in 3GPP-LTE and WiMAX [6]. MIMO transmission techniques to be switched include SM [7], space-division-multiple-access (SDMA) [5], [8], and spatial diversity (SD) [9]. For such an adaptive system, multiple signal detectors are needed at the receiver side with each one corresponding to the respective mode. A straightforward implementation strategy will incur considerable silicon area overhead and be immensely inefficient since most of the modules would remain in an idle state for a large part of the time. As a consequence, an efficient implementation is expected to integrate multiple MIMO detectors into a single module, which can be reconfigured for the respective mode at run-time. Moreover, in real-life wireless systems, signal detectors are usually attached with channel decoders to provide robustness against noise and fading. Therefore, a detector should be capable of not only providing the binary estimation of each bit but also its reliability measurement, e.g., log-likelihood ratio (LLR) [7], to achieve further performance enhancement. Finally, the chip area and power consumption should be low enough to be adopted in practical systems, especially for hand-held devices where the high performance and flexibility need to be combined with energy efficiency. To the best of our knowledge, VLSI implementation of such a reconfigurable multi-mode soft-output MIMO detector remains missing in open literatures.

In an attempt to fill this gap, this paper proposes a soft-output signal detector that supports 64-QAM modulated SM/SDMA/SD triple-mode signals for up to $4{\times}4$ MIMO transmission. Furthermore, it achieves near maximum *a posteriori* probability (MAP) detection performance and provides gigabit-per-second throughput. The unification of multi-mode processing is mainly realized by algorithm-level exploitation, where the algorithms for each mode consist of similar mathematical operations to enable substantial hardware reuse. First, we develop soft-output detection algorithms for SM and SDMA modes based on an efficient extension and modification of the hard-output fixed-complexity sphere decoder (FSD) [10]. More specifically, we introduce a symbol-level bit-flipping scheme, which generates accurate LLR values with marginal hardware increment. Additionally, a polygon-shaped constraint technique is adopted to facilitate the reduction of unnecessary node extensions in the tree search procedure.

L. Liu, J. Löfgren, P. Nilsson, and V. Öwall are with Department of Electrical and Information Technology, Lund University, Lund, Sweden (email: {Liang.Liu, Johan.Lofgren, Peter.Nilsson, Viktor.Owall}@eit.lth.se).

For SD signal detecting, e.g., Alamouti space-frequency block codes (SFBC) [11], [12], we propose a low complexity MAP algorithm owning a unified detection procedure that is independent of antenna number. It allows for parallel detection of the real and imaginary parts of each transmitted symbol with the help of QR decomposition to the orthogonal real-valued channel matrix. Taking advantage of these implementation-oriented algorithms, a unified VLSI architecture is subsequently developed, capable of being reconfigured to support different MIMO modes at run-time. To further improve the implementation efficiency, e.g., reduce the detection latency, we introduce a parallel metric update strategy, which processes multiple candidate vectors simultaneously for soft-value computation and a fast bit-flipping scheme to select the bit-flipped symbol with simple boundary-check operations. To validate the effectiveness of foregoing design solutions, we designed the proposed triple-mode soft-output signal detector using *Synopsys* tools with a 65-nm CMOS standard cell library. Occupying only 0.25 mm^2 core area (83.7K equivalent gate count), the detector achieves 1 Gb/s throughput in SM and SD modes with 4×4 64-QAM configuration, representing a 44% saving to state-of-the-art in terms of hardware efficiency. The throughput for detecting SDMA signal is 250 Mb/s. Working at frequency-selective channels, e.g., the extended vehicular A (EVA) channel specified in LTE standard [13], the detector consumes 59.3mW power in SM mode, resulting in a 59.3 pJ/bit energy consumption. The energy needed to detect a bit in SD and SDMA modes is 10.5 pJ and 169.6 pJ, respectively.

The remainder of this paper is organized as follows: Section II briefly introduces the system model and soft-output MIMO signal detection. Section III describes and evaluates the proposed detection algorithms. Section IV shows the VLSI architecture and module circuit design. The implementation results and performance comparison are presented in Section V, and conclusions are drawn in Section VI.

II. BACKGROUND

A. System Model

As illustrated in Fig. 1, we consider a downlink switching SM/SDMA/SD MIMO system with one base station (BS) and K user equipments (UEs). Both the BS and UEs are equipped with N transmit and receive antennas. The received $N \times 1$ complex signal vector at the kth UE is given by

$$\tilde{r}_k = \tilde{H}_k^c \sum_{k=1}^{K} \tilde{P}_k \tilde{s}_k + \tilde{n}_k, \quad (1)$$

where $\tilde{s}_k = [\tilde{s}_k^{(0)}, \dots, \tilde{s}_k^{(L-1)}]^T$ is the L-layer transmitted vector for user k, in which each component is taken independently from a set of Grey-labeled M-QAM constellation points. Each symbol vector $\tilde{s}_k$ is associated with a bit-level vector b_k (i.e., $\tilde{s}_k = \mathcal{MAP}(b_k)$), which is obtained by error correction coding (ECC) to the original binary source. In (1), $\tilde{n}_k$ is the vector of independent Gaussian noise samples with mean zero and variance $N_0/2$, $\tilde{H}_k^c$ is the $N \times N$ complex channel matrix between the BS and the kth UE, and $\tilde{P}_k$ is the $N \times L$ pre-coding matrix which is selected from a pre-defined code-book and is assumed to be known to both BS and
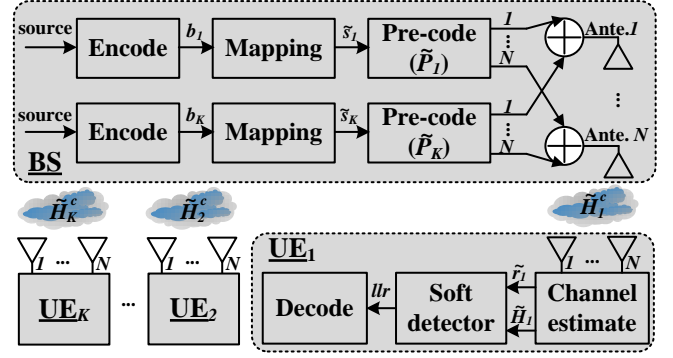

Fig. 1. LTE downlink multi-mode MIMO transmission.

UE [14]. The switch between different MIMO transmissions is realized by changing the matrix $\tilde{P}_k$. Throughout this paper, we set $\tilde{P}_k$ to be an $N \times N$ identity matrix (I_N) in SM mode. While in SD mode, $\tilde{P}_k$ is an Alamouti coding matrix [12]. For SDMA system, $\tilde{P}_k$ is a unitary pre-coding matrix such that $\tilde{P}_k^H \tilde{P}_k = 1$ and $\tilde{P}_k^H \tilde{P}_{l,l \neq k} = 0$, where $(\cdot)^H$ means Hermitian transposition. Moreover, we assume point-to-point transmission in SM and SD modes, i.e., $K = 1$ and $L = N$. Finally, L is set to 1 in SDMA mode, because the number of layer per UE is limited to one in LTE [14].

The complex system can be transformed to its real-valued representation $r_k = H[s_1, \dots, s_K]^T + n_k$, where

$$
\begin{aligned}
r_k &= [\Re(\tilde{r}_{k,1}), \Im(\tilde{r}_{k,1}), \dots, \Re(\tilde{r}_{k,N}), \Im(\tilde{r}_{k,N})]^T \\
s_k &= [\Re(\tilde{s}_{k,1}), \Im(\tilde{s}_{k,1}), \dots, \Re(\tilde{s}_{k,L}), \Im(\tilde{s}_{k,L})] \quad (2) \\
n_k &= [\Re(\tilde{n}_{k,1}), \Im(\tilde{n}_{k,1}), \dots, \Re(\tilde{n}_{k,N}), \Im(\tilde{n}_{k,N})]^T,
\end{aligned}
$$

and

$$
H = \begin{bmatrix}
\Re(\tilde{H}_{1,1}) & -\Im(\tilde{H}_{1,1}) & \dots & \Re(\tilde{H}_{1,N}) & -\Im(\tilde{H}_{1,N}) \\
\Im(\tilde{H}_{1,1}) & \Re(\tilde{H}_{1,1}) & \dots & \Im(\tilde{H}_{1,N}) & \Re(\tilde{H}_{1,N}) \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\Re(\tilde{H}_{N,1}) & -\Im(\tilde{H}_{N,1}) & \dots & \Re(\tilde{H}_{N,N}) & -\Im(\tilde{H}_{N,N}) \\
\Im(\tilde{H}_{N,1}) & \Re(\tilde{H}_{N,1}) & \dots & \Im(\tilde{H}_{N,N}) & \Re(\tilde{H}_{N,N})
\end{bmatrix}. \quad (3)
$$

In (2) and (3), $\tilde{H} = \tilde{H}_k^c[\tilde{P}_1, \dots, \tilde{P}_K]$ is the equivalent channel, $[\cdot]^T$ means vector transposition, $\Re(\cdot)$ and $\Im(\cdot)$ represent the real and imaginary parts of a complex number, respectively.

B. Soft-Output MIMO Signal Detection

Hard-output signal detectors tries to recover the original vector s_k, given r_k and H. While the objective of soft-output detector is to provide reliability information by computing the LLRs for each bit of b_k, e.g., for the lth bit, we have

$$L(b_{k,l} \mid r_k) = ln\frac{P(b_{k,l} = 1 \mid r_k)}{P(b_{k,l} = 0 \mid r_k)} = L_E(b_{k,l} \mid r_k) + L_A(b_{k,l}). \quad (4)$$

In (4), $L_A(b_{k,l})$ is the *a priori* probability and $L_E(b_{k,l} \mid r_k)$ is the extrinsic information. For simplification, we will omit the user index k in the following. According to [7], $L_E(b_l \mid r)$ can be rewritten as

$$L_E(b_l \mid r) = ln\frac{\sum_{b \in \chi_l^1} P(r \mid b_l) \exp(1/2 b_{[l]}^T L_{A[l]})}{\sum_{b \in \chi_l^0} P(r \mid b_l) \exp(1/2 b_{[l]}^T L_{A[l]})}, \quad (5)$$

where χ_l^1 and χ_l^0 are the sets of bit-level vectors having the lth bit equal to 1 and 0, respectively, $\boldsymbol{b}_{[l]}$ denotes the sub-vector of $\boldsymbol{b}$ with the lth bit b_l being omitted, $\boldsymbol{L}_{A[l]}$ is the sub-vector of the *a priori* information vector $\boldsymbol{L}_A = [L_A(b_1), L_A(b_2), \ldots, L_A(b_{N \log_2 M})]^T$ omitting $L_A(b_l)$. The computation of (5) is usually simplified with max-log approximation, yielding the maximum *a posteriori* probability (MAP) algorithm as

$$L(b_l \mid \boldsymbol{r}) \approx \min_{\boldsymbol{b} \in \chi_l^0} \frac{1}{N_0} |\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s}|^2 - \min_{\boldsymbol{b} \in \chi_l^1} \frac{1}{N_0} |\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s}|^2. \quad (6)$$

Note that the *a priori* information is not considered in (6), meaning that we do not take into account the turbo receiver scheme where the inner detector and the outer decoder exchange extrinsic information iteratively [7].

From a hardware design perspective, tree-search algorithms [15]–[21] are promising alternatives to the direct implementation of (6) due to their effectiveness in confining the detection procedure within a much smaller search space. A tree-search algorithm formulates the detection as a $2N$-depth $\sqrt{M}$-ary tree search problem by rewriting the Euclidean distance as $|\boldsymbol{y} - \boldsymbol{R}\boldsymbol{s}|^2$, where $\boldsymbol{R}$ is an upper triangular matrix obtained by $\boldsymbol{H} = \boldsymbol{Q}\boldsymbol{R}$, $\boldsymbol{y} = \boldsymbol{Q}^H \boldsymbol{r}$, and $\boldsymbol{Q}$ is a unitary matrix. Starting from the top ($2N^{th}$) layer, the calculation of the Euclidean distance T is carried out in a recursive way as

$$T_i = T_{i+1} + inc_i,$$
$$inc_i = |y_i - \sum_{j=i+1}^{2N} R_{ij}s_j - R_{ii}s_i|^2 = |y_i' - R_{ii}s_i|^2 \quad (7)$$

where T_i is the partial Euclidean distance (PED) at the ith layer. The soft-output tree-search algorithm generates a list $\mathcal{L}$ of candidate vectors by going through the tree and finds the two elements of (6) within the list, i.e.,

$$L(b_l \mid \boldsymbol{r}) \approx \min_{\boldsymbol{b} \in \mathcal{L} \cap \chi_l^0} \frac{1}{N_0} |\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s}|^2 - \min_{\boldsymbol{b} \in \mathcal{L} \cap \chi_l^1} \frac{1}{N_0} |\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s}|^2. \quad (8)$$

In tree-search detection, $\mathcal{L} \cap \chi_l^{1/0}$ can be empty. Under such circumstance, a constant value is usually adopted to demonstrate that b_l equals to 1 or 0 with a large probability [16]. Specified in this paper, the breadth-first fixed-complexity sphere decoder (FSD) [10], [19] will be explored, because of its low computational complexity, completely regular and feedforward-only dataflow, and near-optimal performance.

III. Soft-Output Detection Algorithms

In this section we will develop detection algorithms for SM, SDMA, and SD MIMO modes, respectively. These proposed algorithms feature low computational complexity and demonstrate similar mathematical operations that can be conveniently integrated into a single VLSI architecture. In detail, we focus on the modification of FSD for SM and SDMA modes. For SD mode, we propose an extensively simplified MAP algorithm by leveraging the orthogonality of Alamouti signals and the matrix-decomposition operations.

A. Low-Complexity LLR Generation Based on FSD

FSD divides the real-valued search tree into two unique parts using a parameter D. A full-search is performed in the first D layers, exhaustively expanding all $\sqrt{M}$ branches per node, while in the remaining ($2N$-D) layers, a single-search is adopted, expanding only one best branch per node. It has been analyzed in [22] that FSD achieves close-to-ML performance if $(D+1)^2 \geq 2N$. For example, $D = 2$ allows the FSD to present an asymptotical ML performance for MIMO system with $N = 4$. However, FSD is more efficient in finding the ML solution in a hard-output scenario instead of generating a list of vectors around the ML result, resulting in poor performance from a soft-output perspective [19]. In this section, we extend the original FSD to provide accurate soft values while maintaining its low computational complexity. With this purpose, we utilize a symbol-level bit-flipping scheme for performance improvement and a polygon-shaped constraint technique to reduce unnecessary node extensions.

1) LLR Accuracy Improvement by Modified Bit-Flipping: Compared to the hard-output ML detection, i.e.,

$$\boldsymbol{s^{ML}} = \arg \min_{\boldsymbol{s} \in 2N^{\sqrt{M}}} \frac{1}{N_0} |\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s}|^2, \quad (9)$$

the soft-output detection consists of two minima search procedures, as demonstrated in (6). One of them is obtained by (9), which is then referred as $T^{ML} = \frac{1}{N_0} |\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s^{ML}}|^2$. The other can then be formulated as

$$T_l^{\overline{ML}} = \min_{\boldsymbol{b} \in \chi_l^{\overline{ML}}} \frac{1}{N_0} |\boldsymbol{r} - \boldsymbol{H}\boldsymbol{s_l^{\overline{ML}}}|^2, \quad (10)$$

in which $\chi_l^{\overline{ML}}$ is the binary complement to the lth bit in the ML bit vector $\boldsymbol{b}^{ML}$. Basically, there are two major reasons that FSD tends to generate poor-quality LLRs. One is the occurrence of vacant bits in the candidate list $\mathcal{L}$ corresponding to $\chi_l^{\overline{ML}}$ (i.e., $\mathcal{L} \cap \chi_l^{\overline{ML}} = \varnothing$), existing in most tree search algorithms. Even for those existing bits, FSD cannot ensure the minimization of (10). This is because unlike K-Best detection, where strict sorting is performed at every layer, FSD simply extends all nodes at the first D layers while only one at the remaining. Such a tree travel scheme does not guarantee the inclusion of best vectors (i.e., vectors with smallest Euclidean distances) in the candidate list.

To tackle these two issues with reasonable complexity overhead, we suggest a modified bit-flipping scheme by replacing the whole vector re-calculation [23] with a per symbol re-calculation scheme. Its basic idea is described as follows: when calculating the LLR $L(b_{i,l})$ for the lth bit in the ith scalar symbol s_i, the strategy is to first find the locally best symbol with the lth bit value different to $\boldsymbol{b}_i^{ML}$, i.e.,

$$s_{i,l}^{BF} = \arg \min_{b_{i,l} \neq b_{i,l}^{ML}} |y_i' - R_{i,i}s_i|^2, \quad (11)$$

and then compute the bit-flipped LLR by

$$L^{BF}(b_{i,l} \mid y_i') = |y_i' - R_{i,i}s_i^{ML}|^2 - |y_i' - R_{i,i}s_{i,l}^{BF}|^2$$
$$= inc_i^{ML} - inc_{i,l}^{BF}. \quad (12)$$

In (11) and (12), y_i' is the received symbol at the ith layer with the interference from previously detected signals being
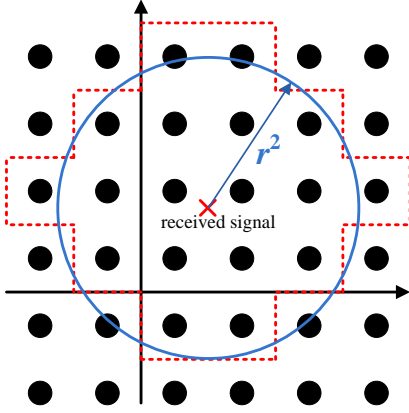
Fig. 2. Polygon-shaped constraint with $\boldsymbol{L}_{2N-1} = [5, 5, 3, 3, 1, 1, 0, 0]$.

canceled and $\boldsymbol{b}_i^{ML}$ is the bit-level vector corresponding to s_i^{ML}, which is denoted as the ith scalar symbol of the ML vector. It should be pointed out that although the ML result is obtained by minimizing $|\boldsymbol{y} - \boldsymbol{Rs}|^2$, it does not promise a locally best result, i.e., $inc_{i,l}^{ML}$ is not necessarily smaller than $inc_{i,l}^{BF}$. Therefore, the sign of $L^{BF}(b_{i,l} \mid y_i')$ should be adjusted to positive or negative according to the corresponding bit value of $b_{i,l}^{ML}$. It should also be mentioned that inc_i^{ML} in (12) has already been calculated during tree search, which can thus be reused in bit-flipping for hardware saving.

So far, we may have two possible LLRs for each bit, which are acquired by FSD tree search ($L_{b_l}^{FSD}$) and the bit-flipping scheme presented above ($L_{b_l}^{BF}$). The final result is selected according to the magnitude of these two candidates

$$L(b_l) = \begin{cases} L_{b_l}^{BF}, & \text{if } |L_{b_l}^{BF}| \leq |L_{b_l}^{FSD}| \text{ or } \mathcal{L} \cap \chi_l^{\overline{ML}} = \varnothing \\ L_{b_l}^{FSD}, & \text{otherwise.} \end{cases}$$
(13)

The selection criteria in (13) finds the minimum of $|L_{b_l}^{BF}|$ and $|L_{b_l}^{FSD}|$, which is efficient in relieving the problem of getting the pseudo-minimum of (10), leading to a more accurate approximation of the MAP result.

2) Complexity Reduction with Polygon-Shaped Constraint: According to the analysis in Sec.III-A1, the exhaustive expansion at upper layers of the FSD tree introduces a lot of computational waste by including vectors with large Euclidean distances in the candidate list. To reduce such unnecessary visits to some nodes, we adopt the imbalanced-expansion technique proposed in [24] to find the list of vectors closer to the ML result. This technique is briefly repeated here for convenience of presentation.

The concept of imbalanced-expansion is to approximate the circular-shaped constraint in a sphere decoder [17] with a polygon-shaped constraint, as illustrated in Fig. 2. The polygon constraint is realized by introducing an extension number limitation L_i^m, by which only the L_i^m best nodes are extended from the mth father node at the ith ($i > 2N - D$) layer. The detailed explanation can be found in [24], where a smaller L_i^m is set for the node with larger PED to expand more/fewer branches from more/less reliable nodes. Moreover, considering that the FSD performs full extension only at the first two real-

valued tree-search layers for a 4×4 system, L_i^m is applied only for the $(2N-1)^{th}$ layers, i.e., $i = 2N-1$ (the constraint to the top layer is accomplished by setting the corresponding L_{2N-1}^m to 0).

Compared to the radius constraint, the polygon-shaped constraint is more efficient from a hardware implementation perspective. Firstly, with a radius constraint r^2, a node dissatisfying the constraint will not be pruned until its PED is completely calculated and compared with r^2. On the other side, the polygon-shaped constraint with extension number limitation early prunes less reliable paths before PED calculations by using the well-known zigzag enumeration technology [25]. Moreover, the number of nodes to be extended is fixed with a given constraint $\boldsymbol{L}_{2N-1} = [L_{2N-1}^1, \cdots, L_{2N-1}^{\sqrt{M}}]$, which is not the case for radius-constrained algorithms where the node extension number is variable depending on the channel and the noise. Therefore, the polygon-constraint algorithm has a very regular data flow and the corresponding control circuitry can be significantly simplified. Finally, the proposed scheme is convenient in tuning the complexity-performance tradeoff by setting $L^{total} = \sum \boldsymbol{L}_{2N-1}$ to a smaller/larger number.

3) Application to SDMA Mode: The aforementioned FSD detection is originally developed for SM signal. In the following, the algorithm is modified to be adopted for the SDMA mode. Detecting downlink SDMA signals is unique in that only signals dedicated to the kth user (i.e., $\tilde{s}_k$) are reserved, while the signals intended for other users (i.e., $\tilde{s}_{l,l \neq k}$) are discarded after detection [26]. To take full utilization of this feature, we add a layer-reordering step to the pre-processing stage of FSD such that the desired signal $\tilde{s}_k$ is moved to the top layer of the FSD search tree where multiple candidates are extended. The reordering is accomplished by a permutation matrix $\boldsymbol{W_k}$, which moves the kth column of the channel matrix $\tilde{\boldsymbol{H}}_k$ to the last position, i.e., $\boldsymbol{W_k} = [\boldsymbol{w_1}, ...\boldsymbol{w_{k-1}}, \boldsymbol{w_{k+1}}, ..., \boldsymbol{w_K}, \boldsymbol{w_k}]$, where $\boldsymbol{w_i}$ denotes an $N \times 1$ vector whose i^{th} element is one, and zeros elsewhere. Therefore, the system model can be rewritten as

$$\begin{aligned} \tilde{\boldsymbol{r}}_k &= \tilde{\boldsymbol{H}}_k \boldsymbol{W}_k \tilde{\boldsymbol{s}}_k^P + \tilde{\boldsymbol{n}}_k \\ &= \tilde{\boldsymbol{H}}_k^P \tilde{\boldsymbol{s}}_k^P + \tilde{\boldsymbol{n}}_k, \end{aligned}$$
(14)

where $\tilde{\boldsymbol{s}}_k^P = [\tilde{s}_1, ...\tilde{s}_{k-1}, \tilde{s}_{k+1}, ..., \tilde{s}_K, \tilde{s}_k]^T$ is the transmit vector with $\tilde{s}_k$ being moved to the last position. Taking the reordered channel matrix $\tilde{\boldsymbol{H}}_k^P$ as an input, the imbalanced-FSD tree search in Section III-A2 is then conducted to get a list of candidate vectors, based on which the LLRs corresponding to $\tilde{s}_k$ are computed.

Since multiple candidates of $\tilde{s}_k$ are extended at upper layers, it can be expected in the final list that $\tilde{s}_k$ has a good diversity in its bit values. Moreover, the single-extension at the remaining layers attaches only the best node of $\tilde{s}_{l,l \neq k}$ to the candidates of $\tilde{s}_k$. Hence, it is highly likely that the final list contains the actual minimum of (10) for the bits corresponding to $\tilde{s}_k$. In view of the above analysis, the candidate list obtained by the layer-reordered FSD tree search is good enough to generate high-quality soft values for $\tilde{s}_k$. Thereby, we can turn off the bit-flipping operation in SDMA mode in order to reduce power consumption.

B. Linear-Complexity MAP Detection for SD Mode

Both the IEEE 802.16m and 3GPP-LTE standards adopt the frequency domain version of Alamouti code as the transmission diversity scheme, where pairs of adjacent sub-carriers are coded together to keep the orthogonality in fast-changing channels. In addition, in order to provide robustness against antenna correlations, the application of SFBC in both standards is limited to two physical antennas [6]. For example, in LTE the signals with four transmit antennas are coded as

$$
\begin{bmatrix}
\tilde{s}_1^1 & \tilde{s}_2^1 & \tilde{s}_3^1 & \tilde{s}_4^1 \\
\tilde{s}_1^2 & \tilde{s}_2^2 & \tilde{s}_3^2 & \tilde{s}_4^2 \\
\tilde{s}_1^3 & \tilde{s}_2^3 & \tilde{s}_3^3 & \tilde{s}_4^3 \\
\tilde{s}_1^4 & \tilde{s}_2^4 & \tilde{s}_3^4 & \tilde{s}_4^4
\end{bmatrix}
=
\begin{bmatrix}
\tilde{x}_1 & \tilde{x}_2 & 0 & 0 \\
0 & 0 & \tilde{x}_3 & \tilde{x}_4 \\
-\tilde{x}_2^* & \tilde{x}_1^* & 0 & 0 \\
0 & 0 & -\tilde{x}_4^* & \tilde{x}_3^*
\end{bmatrix}, \quad (15)
$$

where $\tilde{s}_i^j$ is the complex signal transmitted from the jth antenna at the ith subcarrier. By leveraging these properties, we demonstrate a linear-complexity MAP detection based on QR decomposition to the orthogonal real-valued channel matrix. In addition, the introduced detection unifies the procedures for both 2×2 and 4×4 MIMO. That is, the receiver can use identical processing to detect Alamouti signals, which is independent of the antenna number N.

Without loss in generality, we take the detection of $\tilde{x}_1$ and $\tilde{x}_2$ as an example to illustrate the algorithm. For decoding Alamouti signals, we process the signals in a per antenna manner instead of the per subcarrier vector detection in SM and SDMA modes. With the assumption that the channel is frequency flat, at least over two sub-carriers, the received signals at the jth antenna is formulated as

$$
\begin{bmatrix}
\tilde{r}_1^j \\
(\tilde{r}_2^j)^*
\end{bmatrix}
=
\begin{bmatrix}
\tilde{H}_{j,1} & -\tilde{H}_{j,3} \\
\tilde{H}_{j,3}^* & \tilde{H}_{j,1}^*
\end{bmatrix}
\begin{bmatrix}
\tilde{x}_1 \\
\tilde{x}_2^*
\end{bmatrix}, \quad (16)
$$

where $\tilde{r}_n^j$ $(n = 1, 2)$ is the signal received at the jth antenna of subcarrier n and $\tilde{H}_{j,i}$ $(i = 1, 3)$ is the channel between the ith transmit antenna and j^{th} receive antenna. To simplify the expression, the noise is ignored in the derivation. By adopting the orthogonal real-value decomposition shown in (2) and (3), the complex Alamouti system model in (16) is rewritten as $\boldsymbol{r}^j = \boldsymbol{H_j x}$, where

$$
\begin{aligned}
\boldsymbol{r^j} &= [\Re(\tilde{r}_1^j), \Im(\tilde{r}_1^j), \Re(\tilde{r}_2^j), -\Im(\tilde{r}_2^j)]^T, \\
\boldsymbol{x} &= [\Re(\tilde{x}_1), \Im(\tilde{x}_1), \Re(\tilde{x}_2), -\Im(\tilde{x}_2)]^T,
\end{aligned} \quad (17)
$$

and

$$
\boldsymbol{H_j} =
\begin{bmatrix}
\Re(\tilde{H}_{j,1}) & -\Im(\tilde{H}_{j,1}) & -\Re(\tilde{H}_{j,3}) & \Im(\tilde{H}_{j,3}) \\
\Im(\tilde{H}_{j,1}) & \Re(\tilde{H}_{j,1}) & -\Im(\tilde{H}_{j,3}) & -\Re(\tilde{H}_{j,3}) \\
\Re(\tilde{H}_{j,3}) & \Im(\tilde{H}_{j,3}) & \Re(\tilde{H}_{j,1}) & \Im(\tilde{H}_{j,1}) \\
-\Im(\tilde{H}_{j,3}) & \Re(\tilde{H}_{j,3}) & -\Im(\tilde{H}_{j,1}) & \Re(\tilde{H}_{j,1})
\end{bmatrix}. \quad (18)
$$

The real-valued matrix $\boldsymbol{H_j}$ owns two very important properties: 1) all columns of $\boldsymbol{H_j}$ are orthogonal to each other and 2) $\boldsymbol{H_j}$ has the same element, i.e., $\Re(\tilde{H}_{j,1})$, at its diagonal positions. With these two features, the QR-decomposition of $\boldsymbol{H_j}$ (i.e., $\boldsymbol{H_j} = \boldsymbol{Q_j R_j}$) results in a diagonal matrix $\boldsymbol{R_j} = d_j \boldsymbol{I}_4$. Multiplied by $\boldsymbol{Q}_j^H$, the system model for the jth antenna becomes $\boldsymbol{y}^j = \boldsymbol{R_j x}$, where $\boldsymbol{y}^j = \boldsymbol{Q}_j^H \boldsymbol{r}^j$. Utilizing the diagonal matrix $\boldsymbol{R_j}$, a low-complexity maximal-ratio combining (MRC) of the signals received at each antenna

is conducted to produce the final system model as $\boldsymbol{y} = \boldsymbol{Rx}$, in which

$$
\begin{aligned}
\boldsymbol{y} &= [y_1, y_2, y_3, y_4]^T = \sum_{j=1}^N \boldsymbol{y}^j, \\
\boldsymbol{R} &= \sum_{j=1}^N \boldsymbol{R}_j = \sum_{j=1}^N d_j \boldsymbol{I}_4 = d\boldsymbol{I}_4.
\end{aligned} \quad (19)
$$

The soft values are obtained by performing the MAP detection for each real-valued component in $\boldsymbol{y}$ independently as

$$
L(b_{i,l} \mid y_i) = \min_{\boldsymbol{b}_i \in \chi_l^0} \frac{1}{N_0} |y_i - dx_i|^2 - \min_{\boldsymbol{b}_i \in \chi_l^1} \frac{1}{N_0} |y_i - dx_i|^2. \quad (20)
$$

Compared to the original MAP detection in (6), the complexity of (20) is significantly lower, which is linear to the real-valued constellation size (i.e., $\sqrt{M}$) by decoding the real and imaginary parts of $\tilde{x}_1$ and $\tilde{x}_2$ separately. It is worth to reemphasize that the presented algorithm is conducted using almost the same operations for different antenna configurations with a minor difference in the parameter N of (19).

C. Performance Simulation

The performance of the presented detection algorithms is evaluated using simulations. The simulation setup is based on a simplified LTE downlink system with 64-QAM modulation and 4×4 antenna configuration. The system bandwidth is 5 MHz and the number of sub-carriers is 512, of which 300 sub-carriers are used for the information transmission. For multi-path fading propagation, we apply the extended vehicular A (EVA) channel model described in [13]. There are 9 paths in the channel with a largest delay of 2510 ns. In addition, we assume that the receiver has perfect channel knowledge. The error correcting code is the rate $R_c = 1/3$ parallel concatenated turbo code specified in the LTE standard [14]. The generator polynomials $g_0(D) = 1 + D^2 + D^3$ and $g_1(D) = 1 + D + D^3$ are employed together with an internal interleaver of size 6144. The number of turbo decoder iterations is set to 6. Since the proposed algorithm for SD mode is theoretically a MAP detection, its performance will not be presented in this section.

1) Effects of $\boldsymbol{L}_{2N-1}$ Distribution: Fig. 3 shows the simulated bit-error-rate (BER) of the proposed SM detection algorithm with a same L^{total} (e.g., $L^{total} = \sum \boldsymbol{L}_{2N-1} = 16$) but different $\boldsymbol{L}_{2N-1}$ distributions. The best detection performance is attained when $\boldsymbol{L}_{2N-1} = [5, 4, 3, 2, 2, 0, 0, 0]$. Associated with the corresponding constraint shapes plotted in Fig. 4, we observe that the early-pruned FSD algorithm (with bit-flipping scheme) performs better when the pruning parameter $\boldsymbol{L}_{2N-1}$ leads to a constraint that better approximates the circular-shaped admissible region.

2) Performance with Different L^{total}: Fig. 5 shows the BER of the early-pruned FSD with different L^{total} values. It can be seen that the complexity-performance tradeoffs can be tuned by adjusting L^{total} and a better performance is obtained with a larger L^{total}. We also compared the proposed algorithm with other fixed-complexity soft-output detections, e.g., MMSE [27], K-Best ($K = 64$) [16], and list FSD (LFSD)
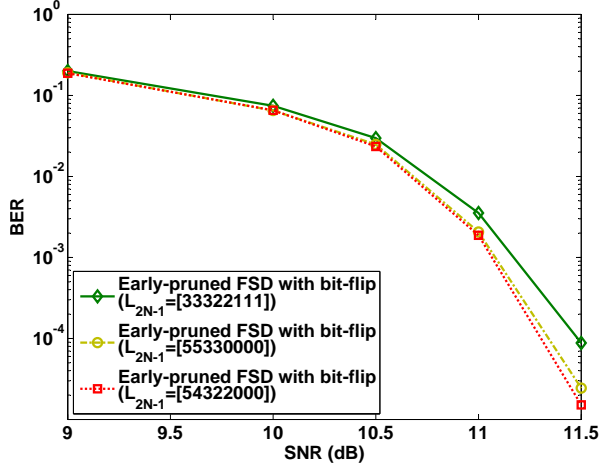
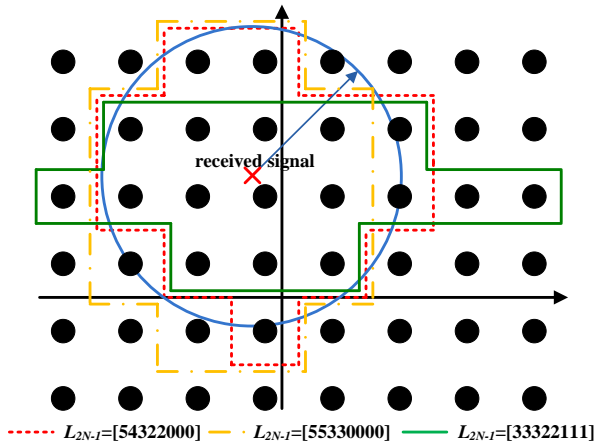Fig. 3. Simulated BER performance in spatial-multiplexing transmission with $L^{total} = 16$ but different $\boldsymbol{L}_{2N-1}$ distributions.

Fig. 5. Simulated BER performance in spatial-multiplexing transmission with different L^{total}. (CORRECTION! There is an error in the IEEE on-line version, where the non-EVA channel results were plotted for the List-FSD and MMSE by mistake.)

Fig. 4. The constraint shapes of different $\boldsymbol{L}_{2N-1}$ distributions.

Fig. 6. Simulated BER performance in space-division-multiple-access transmission with the demonstration of how the bit-flip affects different layers.

[19]. LLR values of ± 8 are adopted for non-existing bits in these algorithms. Our algorithm offers better performance than other fixed-complexity tree-search detections with a much smaller candidate list size (e.g., $L^{total} = 16$ in our algorithm comparing to $K = 64$ in K-Best detection and N_L in LFSD). This is mainly achieved by the bit-flipping technique, which not only provides full bit-value diversity for LLR computation but also improves its reliability by outputting the final LLR with the minima of L^{BF} and L^{FSD}.

3) Performance of SDMA Detection: Fig. 6 shows the BER of the SDMA detection. To demonstrate the effectiveness of the layer-reordering scheme described in Sec. III-A3, we compared the performance when the signals are detected at the top layer and other layers (both with and without bit-flipping). Due to the multi-node extension, the performance degradation is minor without bit-flipping when signals are detected at the top layer. On the other hand, huge performance loss is presented for the detection at the remaining layers. Therefore, the layer-reordering strategy provides us the opportunity to turn off the bit-flipping operation in SDMA mode to save power at the price of a small performance degradation.
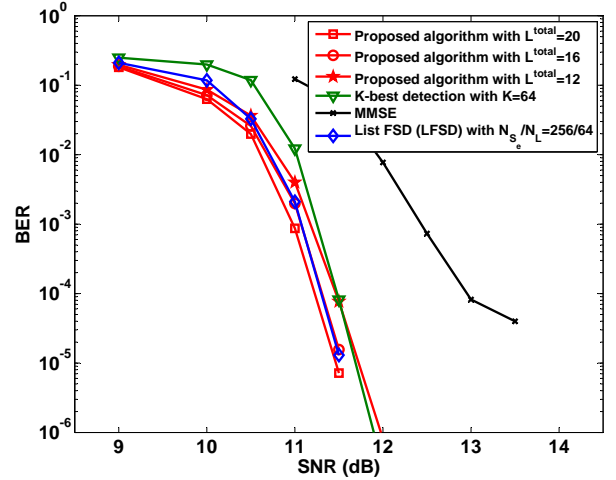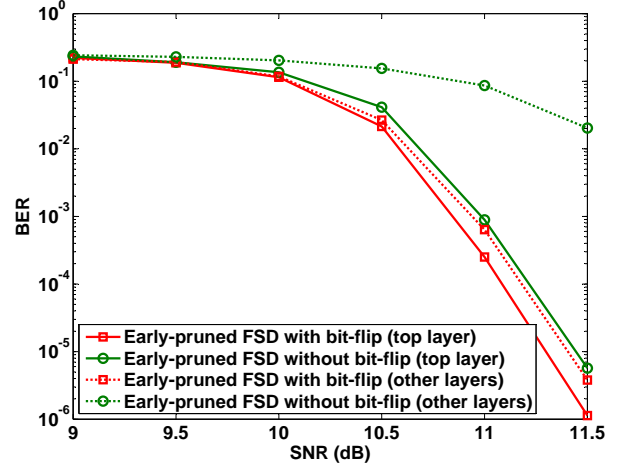
IV. VLSI ARCHITECTURE FOR MULTI-MODE MIMO DETECTION

In this section, we describe the unified VLSI architecture for implementing the soft-output MIMO detection algorithms. As shown in Fig. 7, the detector consists of three major parts to support the detection of SM, SDMA, and SD signals of up to 64-QAM modulation with 4×4 MIMO configuration. First, a tree search block (TSB) conducts the real-valued tree search using the early-pruned FSD algorithm and generates a list of candidate vectors $\mathcal{L}$. Then, a list LLR calculation block (LLCB) calculates soft values based on these vectors and their corresponding Euclidean distances (i.e., the computation of (8)). Finally, a bit-flip block (BFB) computes LLRs with the symbol-level bit-flipping method and selects the final soft output according to (13).

A. Tree Search Block (TSB)

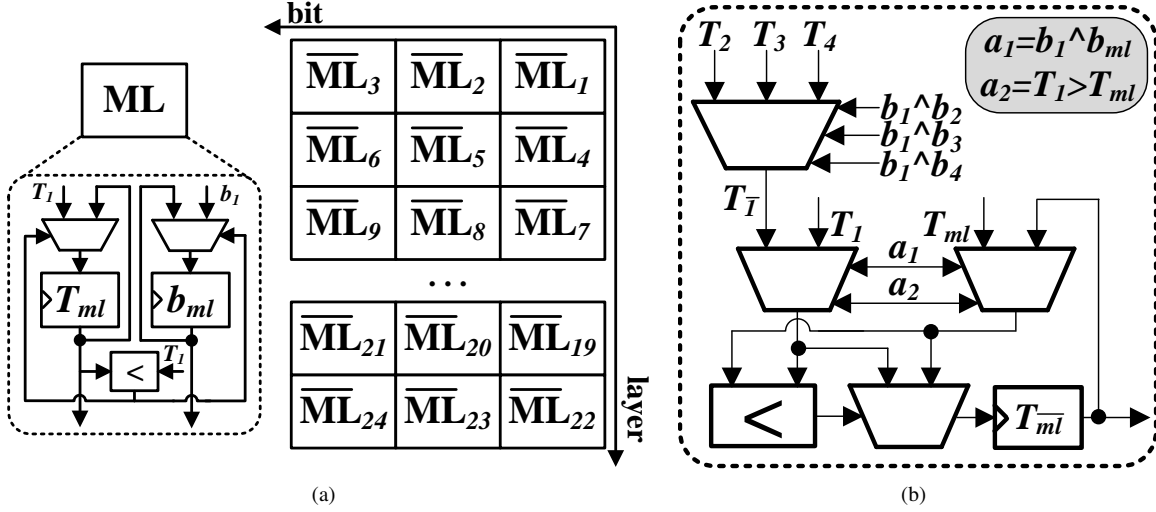The TSB is basically a hard-output MIMO detector [24]. It has four stages of process elements (PEs), corresponding to

Fig. 8. Metric update unit (a) the overall structure, including one ML component and 24 $\overline{ML}$ components (b) the detailed circuit for each $\overline{ML}$ component
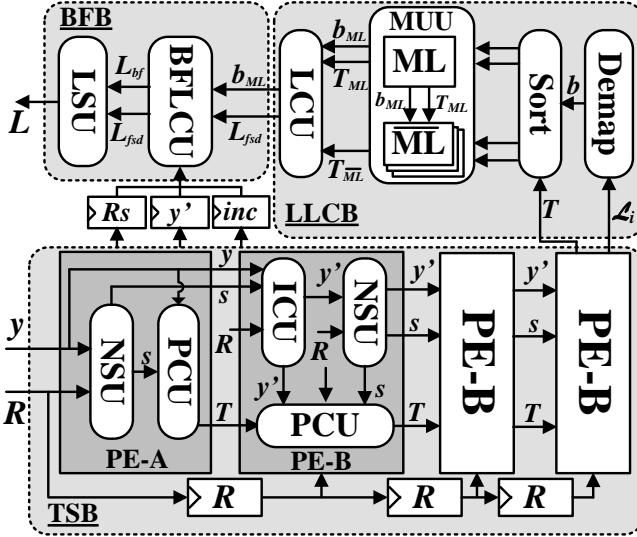

Fig. 7. Proposed VLSI architecture for the multi-mode MIMO detector.

the eight layers of the real-valued search tree in the case of 4×4 MIMO configuration. Each stage computes two adjacent tree layers independently and concurrently. This is due to the fact that $R_{i,i+1} = 0$ ($i = 1,3,\ldots,2N-1$) when QR-decomposition is performed to the orthogonal real-valued representation of the channel matrix [24], [28], [29]. There are three function blocks involved in each PE: 1) an interference cancelation unit (ICU), which suppresses inter-antenna interference introduced by previously detected signals (i.e., the computation of y' in (7)), 2) a node selection unit (NSU) that selects the to-be-detected nodes using the real-value zigzag enumeration method [25], and 3) a PED calculation unit (PCU), which computes the accumulated partial Euclidean distance according to (7). In TSB, PEs are grouped into two types: a) PE-A conducts the imbalanced-extension at the 8th and 7th layer with ICU excluded and b) PE-B performs single-node extension for the remaining layers. In our design, node computations at each stage are carried out in a partial-parallel

time-multiplexing fashion, i.e., the PE processes four nodes simultaneously per iteration and completes the computation of all L^{total} nodes in $C = \frac{1}{4}L^{total}$ cycles. This architecture supports different L^{total} values by adjusting the iteration time C. For example, when a larger L^{total} is required for better performance, the number of iteration cycles is increased to include more candidate vectors in $\mathcal{L}$ at the price of throughput reduction. In addition, the intermediate results (e.g. y' and inc_i) of TSB are deposited into registers to be reused in the bit-flipping operation, leading to further hardware savings. It should be reemphasized that TSB takes $\frac{1}{4}L^{total}$ cycles to generate the candidate list $\mathcal{L}$ by outputting a size-four candidate vector list $\mathcal{L}_i$ per clock cycle.

B. List LLR Calculation Block (LLCB)

Accepting the candidate vector list ($\mathcal{L}$) and their Euclidean distances (T) as inputs, the LLCB first translates the symbol vector to its corresponding bit vector (**b**) with the *Demap* unit, which are then sorted according to the ascending order of T. Given these sorted results, the metric update unit (MUU) finds s^{ML} and T^{ML} in (9) and the corresponding $T^{\overline{ML}}$ in (10) for each bit of the transmitted $N\log_2(M)$-length binary vector. As illustrated in Fig. 8(a), the MUU is basically a metric table, where each cell maintains the best path metric (i.e., Euclidean distance) among the candidates that may have the value 1 or 0 on the specified positions. More specifically, the metric table in our design is divided into two groups: one ML component and $N\log_2(M)$ $\overline{ML}$ components. The ML cell contains the ML bit vector (i.e., $\boldsymbol{b}^{ML}$) and its Euclidean distance (i.e., T^{ML}). The lth $\overline{ML}$ cell stores the best path metric ($T_l^{\overline{ML}}$) with its corresponding binary vector having the lth bit value different to $\boldsymbol{b}^{ML}$, e.g., $\boldsymbol{b}_l^{\overline{ML}}$ satisfying (10).

In most reported soft-output MIMO detectors [15]–[17], the metric table processes only one candidate symbol per clock cycle, which destroys the parallelism of the detector architecture and decreases the detection throughput. Making use of the fact that the incoming candidate vectors are strictly sorted, we design a parallel metric update unit, capable of
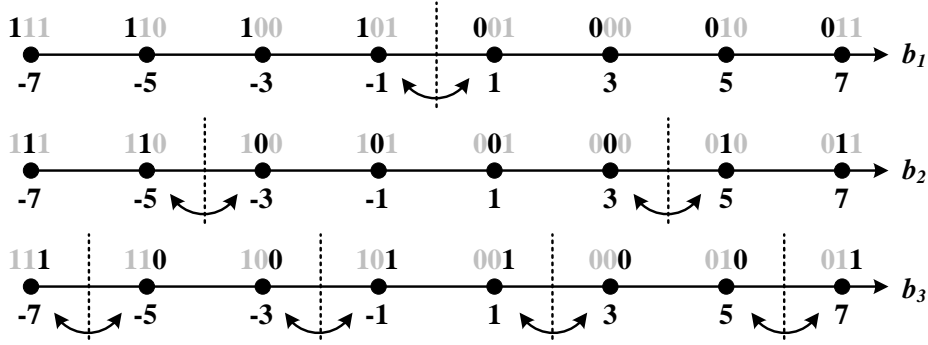
Fig. 9. Examples of Gray-coded 64-QAM points (real-valued) and the flipping schemes of different bits.

handling multiple vectors simultaneously using simple and implementation-friendly operations. Specified in our detector, the metric table processes four vectors per clock cycle, compatible with the parallel strategy adopted in the TSB. The proposed parallel metric update algorithm is initialed with $T^{ML} = T^{\overline{ML}} = \infty$. Whenever the sorted candidate vectors (i.e., $[\boldsymbol{b}^1, \boldsymbol{b}^2, \boldsymbol{b}^3, \boldsymbol{b}^4]$ with $T^1 \leq T^2 \leq T^3 \leq T^4$) are available, the ML table compares T^{ML} with the best metric T^1 and refreshes the corresponding registers with $\min(T^{ML}, T^1)$. The ML bit vector $\boldsymbol{b}^{ML}$ is updated accordingly. The structure of the lth $\overline{ML}$ metric table is shown in Fig. 8(b). The first step of $\overline{ML}$ metric update is to find among $[T^2, T^3, T^4]$ the best metric (denoted by $T^{\bar{1}}$) whose binary vector has the lth bit value different to b_l^1. In case that $\boldsymbol{b}^1$, $\boldsymbol{b}^2$, $\boldsymbol{b}^3$, and $\boldsymbol{b}^4$ have the same value on the lth bit, $T^{\bar{1}}$ is set to ∞. The remaining procedure for updating $\overline{ML}$ table is divided into following three cases by using the properties that T^{ML} is always smaller than $T^{\overline{ML}}$ and T^1 is no larger than $T^{\bar{1}}$:

1) If $\boldsymbol{b}^{ML}$ agrees with $\boldsymbol{b}^1$ on the lth bit value, i.e., $b_l^{ML} {}^\wedge b_l^1 = 0$, $T^{\overline{ML}}$ is updated with $\min(T^{\overline{ML}}, T^{\bar{1}})$;
2) If $\boldsymbol{b}^{ML}$ disagrees with $\boldsymbol{b}^1$ on the lth bit value, i.e., $b_l^{ML} {}^\wedge b_l^1 = 1$ and T^{ML} has been updated with T^1, i.e., $T^{ML} > T^1$, $T^{\overline{ML}}$ is replaced by $\min(T^{ML}, T^{\bar{1}})$;
3) If $b_l^{ML} {}^\wedge b_l^1 = 1$ but T^{ML} remains unchanged in the ML metric update process, i.e., $T^{ML} \leq T^1$, $T^{\overline{ML}}$ is updated with $\min(T^{\overline{ML}}, T^1)$.

The throughput can be effectively enhanced by the proposed parallel metric update scheme. More importantly, the operations involved in this metric update unit contain only exclusive *OR*, comparison, and multiplex selection, which are suitable for low-cost hardware implementation. The MUU obtains the final T^{ML}, $\boldsymbol{b}^{ML}$, and $T^{\overline{ML}}$ within $\frac{1}{4}L^{total}$ clock cycles and sends them to the LLR calculation unit (LCU) to finish the LLR computation.

C. Bit-Flip Block (BFB)

The BFB includes a bit-flipping LLR calculation unit (BFLCU) and an LLR selection unit (LSU). For the lth bit of the ML vector $\boldsymbol{b}^{ML}$, BFLCU executes the symbol-level bit-flipping scheme according to (12). The objective is to find the best bit-flipped symbol, denoted by $s_l^{\overline{ML}}$, which owns the smallest Euclidean distance among the symbols having the lth bit value opposite to $\boldsymbol{b}^{ML}$. Instead of calculating the Euclidean
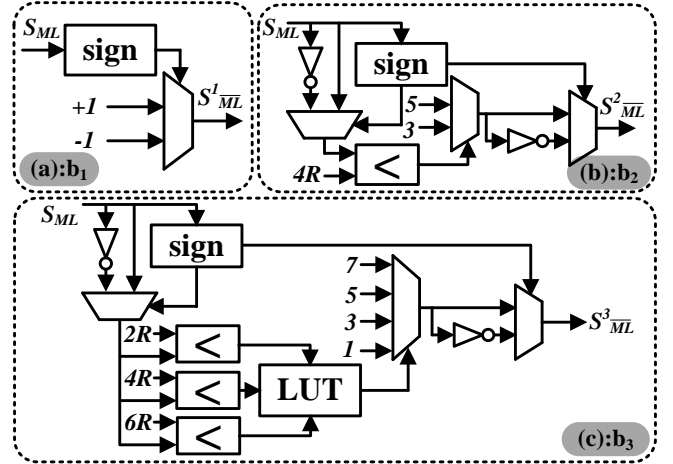

Fig. 10. Circuit diagram for bit-flipped symbol selection with (a) for the first bit b_1, (b) for the second bit b_2, and (c) for the third bit b_3.

distances of all $M/2$ possible bit-flipped symbols and finding the minimum with extensive comparison, we propose to observe the location of s^{ML} in the constellation plane and then select $s_l^{\overline{ML}}$ with simple boundary check. Here, we take the gray-coded modulation scheme adopted in LTE system [30] as an example, where a specific bit changes only I or Q branch of the modulated signal. More specifically, among the log_2^M bits, the $log_2^M/2$ odd bits determine the position of the modulated signal on the I-axis, and the even bits determine the position on the Q-axis. Hence, the bit-flipping can be conducted for the real and imaginary signals separately, compatible with the real-value decomposed system in this paper. The real-valued 64-QAM constellation (or equivalently, the 8-PAM modulation) adopted in 3GPP-LET is shown Fig. 9, where the boundaries (dashed lines) for flipping the bit b_1, b_2, and b_3 are illustrated. Exploiting the above regularity, the selection of $s_l^{\overline{ML}}$ can be accomplished conveniently with boundary and sign-bit check. For example, the best bit-flipped symbol corresponding to the first bit (i.e., b_1) is obtained by only checking the sign-bit of s^{ML} and setting $s_1^{\overline{ML}}$ to $-sign(s^{ML}) \times 1$. In Fig. 10, the detailed circuit diagrams of the bit-flipped symbol selection for b_1, b_2, and b_3, are demonstrated respectively.

LSU receives two possible LLR values, which respectively are obtained by the LLCB (L^{FSD}) and the bit-flipping LLR calculation unit (L^{BF}) and selects the final LLR based on the
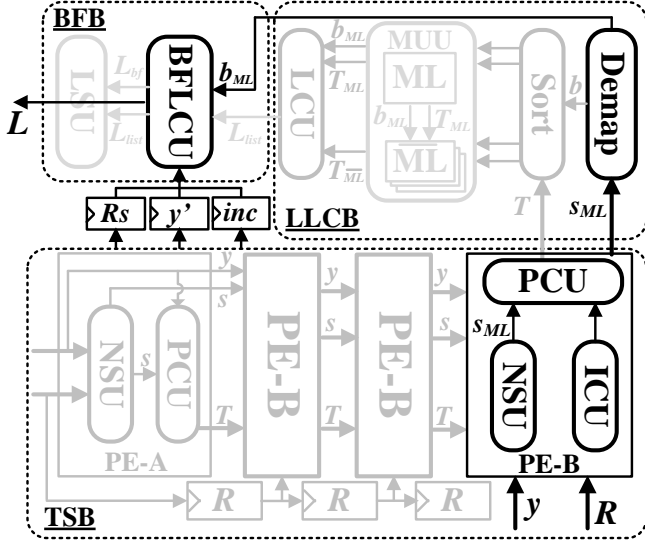
Fig. 11. Date path when the detector is configured to spatial-diversity mode.

Fig. 12. Layout of the detector core.

TABLE I
AREA PARTITION BY FUNCTION BLOCKS

Function Blocks	Gate Count	Percentage
Tree-Search Block	46.4 KG	55.4%
List LLR Calculation Block	11.7 KG	14%
Bit-Flip Block	12.6 KG	15.1%
Other Logic	13 KG	15.5%

criteria shown in (13).

D. Data-Path to Other Detection Modes

In this section, we present how the proposed architecture is reconfigured to realize the detection of SD and SDMA signals. The operations for SDMA mode are similar to SM with the minor difference that the LLRs are only calculated for the symbol dedicated to the kth user (i.e., $\tilde{s}_k$). Therefore, we simply by-pass parts of the units in the LLCB for SDMA signal detection, such as the $\overline{ML}$ cells $\overline{ML}_7 - \overline{ML}_{24}$ in Fig. 8(b). The tree-search block is completely reused in these two modes. Moreover, as mentioned in Section III-A3, the computation of LLR for SDMA signals is based only on the candidate vectors generated by the tree search process. Therefore, the BFB is also turned off, using clock gating, in SDMA mode to save power.

According to the description in Section III-B, the procedures involved with the SD signal detection can be decomposed into two parts: the maximal-ratio combining (MRC) shown in (19) and the LLR calculation shown in (20). Thanks to the diagonal property of matrix R in (19), the MRC in our algorithm has been substantially simplified to contain only real additions. As a consequence, we use the interference cancelation unit (ICU) in the last stage of PE, mainly composed of accumulations, to conduct the MRC operation. The main task of calculating (20) is to find two minimum Euclidean distances with the corresponding bit vectors having the lth value equal to 1 and 0, respectively. Due to the diagonal property of the equivalent channel matrix R in (19), this minima-search procedure is conducted for each real-valued scalar symbol independently, which is then equivalent to the symbol-level bit-flipping operation in the SM signal detection algorithm, i.e., (12). As a result, instead of searching all $\sqrt{M}$ real-valued constellation points, the LLR computation in SD mode can be divided into two steps. The first step is to get the ML scalar symbol (s^{ML}) using the node selection unit (NSU) in the last stage of type-B process element (PE-B), which performs single-node extension
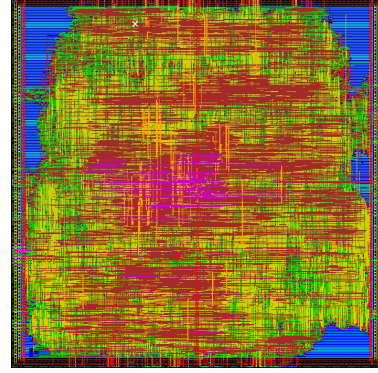
by finding the best scalar symbol of each farther node. The second step is to find another minimum symbol ($s_l^{\overline{ML}}$) with its lth bit value different to s^{ML} and compute the deviation of their Euclidean distances. This job can be accomplished by the bit-flipping LLR calculation unit (BFLCU) in the BFB. Based on the above analysis, the detailed data-path when the detector is configured to SD mode is given in Fig. 11 (the black parts are active data-paths, while the grey parts are disabled with clock-gating technique). It is worthwhile to mention that the proposed architecture is capable of detecting SD signals in a fully parallel fashion, leading to an enormous throughput enhancement. This is a consequence of the fact that PE-B in the detector simultaneously deals with four nodes per clock cycle and the number of symbols to be processed in our SD detection has also been fixed to four (e.g., the parameter i in (20) is within the range of $[1, 2, 3, 4]$).

The proposed architecture is also flexible in supportting different modulations and antenna configurations (MACs). The multi-stage PE structure and the candidate sharing scheme [24] in TSB are used for supporting multiple antenna numbers and constellation sizes, respectively. Additionally, LLCB can also be configured to process different MACs by activating different $\overline{ML}$ cells. For example, only $\overline{ML}$ cells $\overline{ML}_1 - \overline{ML}_{12}$ are used when the detector is configured to 2×2 64-QAM mode. Finally, we use sub-sets of the bit-flipping circuitry for different modulations, i.e., [(a),(b),(c)] in Fig. 10 for 64-QAM, [(a),(b)] for 16-QAM, and only (a) for QPSK. Accordingly, the checking boundaries is changed in different modulations, e.g., $s_2^{\overline{ML}}$ is selected from $sign(s^{ML}) \times [1, 3]$ for 16-QAM, by comparing $|s^{ML}|$ with $2R$.

V. IMPLEMENTATION RESULTS AND COMPARISON

The designed triple-mode soft-output MIMO detector is modeled in Verilog Hardware Description Language (Verilog-HDL), synthesized using Synopsys Design Compiler with a

TABLE III
COMPARISON OF THIS WORK TO PREVIOUS 4×4 SOFT-OUTPUT MIMO DETECTORS

	JSAC' 06 [15]	TVLSI' 07 [16]	TVLSI' 11 [21]	JSSC' 12 [31]	ISCAS' 10 [20]	This Work
MIMO Modes	SM	SM	SM	SM	SM	SM/SD/SDMA
Antenna Size	4×4	4×4	4×4	4×4	4×4	4×4
Modulation	16-QAM	64-QAM	64-QAM	64-QAM	64-QAM	64-QAM
Algorithm	Soft-output K-best	Soft-output K-best	Soft-output best-first	SISO MMSE-PIC	Soft-output K-best	Early-pruned FSD with bit-flip
Process Technology	0.18 μm	0.13 μm	65 nm	90 nm	65 nm	65 nm
Max. Clock Rate	200 MHz	270 MHz	333 MHz	568 MHz	833 MHz	167 MHz
Throughput	106.6 Mb/s	8.57 Mb/s	83.3 Mb/s	757 Mb/s	2 Gb/s	1 Gb/s
Core Area	0.56 mm^2	2.38 mm^2	N/A ^{2}	1.5 mm^2	0.57 mm^2	0.25 mm^2
Gate Count	97 kG^a	280 kG^a	64 kG^a	410 kG^b/160 kG^a	298 kG^a	83.7 kG^a
Hardware Efficiency	0.91 kG/(Mb/s)^a	32.67 kG/(Mb/s)^a	0.77 kG/(Mb/s)^a	0.21 kG/(Mb/s)^a	0.15 kG/(Mb/s)^a	0.084 kG/(Mb/s)^a
Power Consumption	N/A	94 mW @ 1.2 V	11.5 mW @ 1.0 V	189.1 mW @ 1.2 V	280 mW @ 1.3 V	59.3 mW @ 1.2 V (SM mode)
Normalized Power Consumption	N/A	47 mW	16.6 mW	136.6 mW	238.6 mW	59.3 mW
Normalized Energy Consumption	N/A	5484 pJ/bit	199.2 pJ/bit	180.4 pJ/bit	119.3 pJ/bit	59.3 pJ/bit

^{a} the gate count without the pre-processing operations to channel decomposition.
^{b} the gate count including the pre-processing operations to channel decomposition.

TABLE II
DETECTION THROUGHPUT AND POWER/ENERGY CONSUMPTION OF DIFFERENT MIMO MODES

MIMO Mode	Throughput	Power Consumption	Energy Consumption
SM	1 Gb/s	59.3 mW	59.3 pJ/bit
SDMA	250 Mb/s	42.4 mW	169.6 pJ/bit
SD	1 Gb/s	10.5 mW	10.5 pJ/bit

65-nm CMOS standard digital cell library, and routed using Cadence SoC Encounter/Silicon Ensemble. We use 12 bits for the PEDs and 8 bits for the output LLRs, which have been decided by fixed-point simulations. Fig. 12 shows the layout of the detector core, which occupies 0.25mm^2 area (at 68% cell density) and takes 83.7K equivalent gates. Here an equivalent gate is counted as the size of a two-input NAND gate. The pre-processing to the channel matrix (e.g., real-value decomposition and QR decomposition) is not included in this implementation. To enable a more comprehensive understanding of the hardware resource distribution, the total core area is partitioned by function blocks, as shown in Table I. We can see that the extra area required to support soft-output generation (i.e., list LLR calculation block and bit-flip block) is approximately an additional 50% to that required by the hard-output detector (i.e., tree-search block).

At normal 1.2-V core power supply, the detector can operate at a clock rate of 167 MHz for all the three MIMO modes. The maximum allowable clock rate is obtained by post-layout simulation with delay information back annotated in SDF format. With clock frequency f_c, the throughput of the detector is formulated as

$$\text{Throughput} = f_c \times R_{SFC} \times \frac{\log_2^M \times N}{C}, \quad (21)$$

where M is the constellation size, N is the antenna number, R_{SFC} is the coding rate of the space-frequency code, and C is the number of clock cycles needed for calculating the all nodes. The parameter R_{SFC}=1 for SM transmission and $R_{SFC} = 1/N$ in SDMA mode, since only one layer of signal is transmitted for each user. As a result, the detection throughput in SDMA mode is lower than that in SM mode. The pair-wise Alamouti space-frequency code is adopted in SD mode, in which R_{SFC} is set to $1/2$. In the post-out simulation, L^{total} is set to 16. Therefore, C equals to 4 in SM/SDMA detection and 1 in SD detection. Table II summarizes peak throughput when the detector is configured to different MIMO modes. To investigate power efficiency, power simulations were conducted for the post-layout design annotated with switching activities. Operating at 167MHz, the detector consumes 59.3-mW core power with 1.2-V supply voltage and 25°C temperature when configured to SM mode. The corresponding energy consumption per bit detection is 59.3pJ/bit. When configured to SD and SDMA mode, the detector uses a clock-gating technology for those shielded function blocks to achieve low power. The power and energy consumption in these modes is also tabulated in Table II. The power consumption is much lower in SD mode by closing large parts of the function blocks, as demonstrated in Fig. 11. On the other hand, the SDMA signal detector reuse most of the blocks except for BFB and parts of LLCB, and thus consumes slightly less power than the SM detector. Thereby, the detector consumes the highest energy when configured to SDMA mode due to the low detection throughput.

Table III lists the overall performance of our detector and several recently reported 4×4 soft-output detectors. The proposed detector is the only one that supports the signal detection of spatial-multiplexing (SM), spatial-diversity (SD), and space-division-multiple-access (SDMA) MIMO modes.

Despite the superiority of this reconfigurable multi-mode property, our detector demonstrates the best hardware efficiency, which is calculated by normalizing the number of equivalent gate count to the detection throughput. The cost reduction is mainly provided by the algorithm-architecture co-design method, which reuses the same building blocks in different MIMO modes. Besides, the symbol-level bit-flipping strategy is capable of generating high-quality LLRs with very small hardware overhead. Moreover, it allows for extensive branch pruning in the tree search, which reduces the cost significantly by saving a huge amount of hardware-consuming Euclidean distance calculations. For example, as demonstrated in Fig. 5, our tree-search algorithm expands only 16 branches to provide a similar detection performance to the K-Best algorithm in [16] where $64\sqrt{M}$ branches are calculated at each layer. Although our detector does not demonstrate the highest throughput among the reported implementations, it is one of the very few detectors that achieve gigabit-per-second throughput to meet the requirement of next-generation cellular system. To ensure a fair enough comparison, the power consumption in Table III is normalized to the 65-nm technology and 1.2-V supply voltage and is formulated as

$$\text{Power}_{norm} = \text{Power} \times \left(\frac{1.2V}{\text{Voltage}}\right)^2 \times \frac{65nm}{\text{Technology}}. \quad (22)$$

VI. CONCLUSION

This paper for the first time presents the VLSI implementation of a triple-mode soft-output MIMO detector that supports the detection of spatial-multiplexing, space-division-multiple-access, and spatial-diversity signals. This design applies algorithm-level modifications to a fixed-complexity sphere decoder, such as early-prune technology with polygon-shaped constraint and symbol-level bit-flipping, to improve the soft information accuracy. In addition, an antenna-number-independent MAP algorithm is shown to be very effective in detecting Alamouti signals. In terms of VLSI implementation, a unified architecture is developed to be reconfigured to support different MIMO modes, leading to extensive hardware saving. Moreover, several circuit-level techniques are adopted in the design of function blocks to further improve the implementation efficiency. Post-layout simulation results have shown that the proposed detector reaches gigabit-per-second detection throughput and outperforms the other reported works in terms of multi-mode support capability, hardware efficiency (44% saving in normalized gate count) as well as energy efficiency (50% reduction in energy consumption).

REFERENCES

[1] 3GPP Technical Report 36.913: Requirements for Further Advancements for Evolved Universal Terrestrial Radio Access (E-UTRA) (LTE-Advanced) [Online]. Available: http://www.3gpp.org/FTP/ Specs/html-info/36913.htm.

[2] IEEE 802.16m System Requirements [Online]. Available: http://ieee802.org/16/tgm/docs/80216m-07_002r4.pdf.

[3] Overview of 3GPP Release 10 V0.0.8 (2010-09) [Online]. Available: http://www.3gpp.org/ftp/Information/WORK_PLAN/Description_Releases/Rel-10_description_20100924.zip.

[4] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R.A. Valenzuela, "V-BLAST: an architecture for realizing very high data rates over the rich-scattering wireless channel," *International Symposium on Signals, Systems, and Electronics (ISSSE)*, pp. 295-300, Sept. 1998.

[5] B. C. Kian, A. Doufexi, and S. Armour, "On the performance of SU-MIMO and MU-MIMO in 3GPP LTE downlink," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1482-1486, Sept. 2009.

[6] Q. Li, *et al.*, "MIMO techniques in WiMAX and LTE: a feature overview," *IEEE Communications Magazine*, vol. 48, no. 5, pp. 86-92, May. 2010.

[7] B. M. Hochwald and S. Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. on Commun.*, vol. 51, no. 3, pp. 389-399, May. 2003.

[8] G. Bauch and G. Dietl, "Multi-user MIMO for achieving IMT-Advanced requirements," *International Conference on Telecommunications (ICT)*, pp. 1-7, Nov. 2008.

[9] C. Spiegel, *et al.*, "MIMO schemes in UTRA LTE, a comparison," *IEEE Vehicular Technology Conference (VTC Spring)*, pp. 2228-2232, May. 2008.

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

[11] V. Tarokh, H. Jafarkhani, and A. Calderbank, "Space-time block coding for wireless communications: performance results," *IEEE J. on Selected Areas in Communications*, vol. 17, pp. 451-460, Mar. 1999.

[12] S. Alamouti, "A Simple Transmit Diversity Technique for Wireless Communications," *IEEE J. on Selected Areas in Communications*, vol. 16, pp. 1451-1458, Oct. 1998.

[13] 3GPP Technical Specification 36.101 V10.1.1: User Equipment (UE) radio transmission and reception (Release 10)[Online]. Available: http://www.3gpp.org/ftp/Specs/archive/36_series/36.101/36101-a11.zip.

[14] 3GPP Technical Specification 36.213 V9.1.0: Physical layer procedures (Release 9) [Online]. Available: http://www.3gpp.org/ftp/Specs/2010-03/Rel-9/36_series/36213-910.zip.

[15] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Selected Areas in Communication*, vol. 4, no. 3, pp. 491-503, Mar. 2006.

[16] S. Chen, T. Zhang, and Y. Xin, "Relaxed K-Best MIMO signal detector design and VLSI implementation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 3, pp. 328-337, Mar. 2007.

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

[18] Chun-Hao Liao, To-Ping Wang, and Tzi-Dar Chiueh, "A 74.8 mW soft-output detector IC for 8×8 spatial-multiplexing MIMO communications ," *IEEE J. Solid-State Circuits*, vol. 45, no. 2, pp. 411-421, Feb. 2010.

[19] L.G. Barbero and J.S. Thompson, "Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-MIMO systems," *IEEE Trans. on Vehicular Technology*, vol. 57, no. 5, pp. 2804-2814, Sep. 2008.

[20] D. Patel, *et al.*, "VLSI implementation of a WiMAX/LTE compliant low-complexity high-throughput soft-output K-Best MIMO detector," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 593-596, May. 2010.

[21] C.-A. Shen, *et al.*, "A best-first soft/hard decision tree searching MIMO decoder for a 4 64-QAM system," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 99, 2011.

[22] J. Jalden, L. G. Barbero, B. Ottersten, and J. S. Thompson, "The error probability of the fixed-complexity sphere decoder," *IEEE Trans. on Signal Processing*, vol. 57, no. 7, pp. 2711-2720 , Jul. 2009.

[23] V. Ponnampalam, D. McNamara, A. Lillie, and M. Sandell, "On generating soft outputs for lattice-reduction-aided MIMO detection," *IEEE International Conference on on Communications (ICC)*, pp. 4144-4149, May. 2007.

[24] L. Liu, J. Löfgren and P. Nilsson, "Area-efficient configurable high-throughput signal detector supporting multiple MIMO modes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 9, pp. 2085-2096, Sept. 2012.

[25] M. Wenk, *et al.*, "K-best MIMO detection VLSI architectures achieving up to 424 Mbps," *Proc. of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1151-1154, May. 2006.

[26] S. Dragan, L. Angel, and B. P. Constantinos, "Design and experimental validation of MIMO multiuser detection for downlink packet data," *EURASIP J. on Applied Signal Processing*, vol. 11, pp. 1769-1777, 2005.

[27] P. Fertl, J. Jalden, and G. Matz, "Capacity-based performance comparison of MIMO-BICM demodulators," *IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 166-170, Jul. 2008.

12

[28] L. Azzam and E. Ayanoglu, "Reduced complexity sphere decoding for square QAM via a new lattice representation," *in Proc. IEEE Global Telecommun. Conf.*, pp. 4242-4246, Nov. 2007.

[29] J. Lofgren and P. Nilsson, "On MIMO k-best sphere detector architecture complexity reductions," *International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1-9, Dec. 2008.

[30] 3GPP Technical Specification 36.211 V9.1.0: Physical Channels and Modulation (Release 9) [Online]. Available: http://www.3gpp.org/ftp/Specs/2010-03/Rel-9/36_series/36211-910.zip.

[31] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754-1765, 2011.

Viktor Öwall (M'90) received the M.Sc. and Ph.D. degrees in electrical engineering from Lund University, Lund, Sweden, in 1988 and 1994, respectively. During 1995 to 1996, he joined the Electrical Engineering Department, the University of California at Los Angeles as a Postdoc where he mainly worked in the field of multimedia simulations. Since 1996, he has been with the Department of Electrical and Information Technology, Lund University, Lund, Sweden. He is currently full Professor at the same department and since 2009 the Head of Department. He is the Director of the VINNOVA Industrial Excellence Center in System Design on Silicon (SoS). His main research interest is in the field of digital hardware implementation, especially algorithms and architectures for wireless communication, image processing and biomedical applications. Research projects include combining theoretical research with hardware implementation aspects in the areas of wireless communication, video processing, and digital holography.

Dr. Öwall was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-II: ANALOG AND DIGITAL SIGNAL PROCESSING from 2000-2002 and of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS from 2007-2009. He was Guest Editor for the Special issue on ISCAS 2010 of the IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS.

Liang Liu (S'10-M'11) received his B.S. degree in 2005 and Ph.D. degree in 2010 from Fudan University, Shanghai, P.R.China. From Jan. 2010 to Apr. 2010, he was with Electrical, Computer and Systems Engineering Department, Rensselaer Polytechnic Institute (RPI, New York, USA) as a visiting scholar. He is currently a post-doc researcher with the Electrical and Information Technology (EIT) Department, Lund University, Lund, Sweden. His research interest is in the field of digital circuits design for wireless communication system.

Johan Löfgren (S'06) received his M.S. degree in 2002 from Chalmers University of Technology (Gothenburg, Sweden). Thereafter he started working with mobile phone development for Ericsson and Sony Ericsson. In 2006 he joined the Electrical and Information Technology Department at Lund University (Lund, Sweden) as a Ph.D. candidate. Johan's research interests are in the implementation issues in digital communication. He is currently involved in projects focusing on channel estimation and symbol detection algorithms and implementations.

Peter Nilsson (S'90-M'96-SM'12) received the M.S. degree in electrical engineering, the Licentiate of Engineering degree in 1992, and the Ph.D. degree in engineering in 1996 from Lund Institute of Technology, Lund University, Lund, Sweden. After receiving the Ph.D. degree, he was an Assistant Professor in the Department of Applied Electronics (now Department of Electrical and Information Technology), Lund University. In November 1997, he became an Associate Professor at the same department and in December 2003, the degree "Docent" was awarded. In July 2008, he became a Full Professor. Peter was the National Program Manager for Socware Research & Education, a System-on-Chip program for research and Master's educations, 2000-05. Peter has authored or co-authored 80 peer-reviewed journal and conference papers. He was also an Associate Editor for IEEE Transactions on Circuits and Systems I, 2004-05, and is a member of the VLSI Systems and Applications Technical Committee in IEEE Circuits and Systems Society. Peter's main interest is in the field of implementation of digital circuits.