A Hardware-Efficient VLSI Architecture for Hybrid Sphere-MCMC Detection

Fang-Li Yuan, Chia-Hsiang Yang†, Dejan Marković
Department of Electrical Engineering, University of California, Los Angeles, USA
†Department of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan

Abstract—This paper presents a hybrid soft-output MIMO detector that searches reliable soft-information in both deterministic and probabilistic ways. The fixed-complexity sphere detector (FSD) is first applied to provide near maximum-likelihood (ML) solutions. The solutions are next used to initialize the Markov Chain Monte Carlo (MCMC) detector that uses parallel Gibbs samplers (GSs) for remaining candidate enumeration. A low-complexity VLSI architecture is proposed to demonstrate the feasibility of hardware realization for high-throughput applications. Simulation results indicate that the hybrid detector has a low-complexity variant of breadth-first detectors, to coarsely estimate the log-likelihood ratio (LLR) of each transmitted bit. It also serves as an initiator for GS, the simplest yet powerful MCMC algorithm, for subsequent fine searches. The required number of GS iterations in HSMCD is relaxed compared to traditional MCMC detector, thereby improving the overall processing speed. The problem of poor performance under high-SNR is also eliminated because the lower bound of BER performance is already guaranteed by FSD. Correspondingly, the resource overhead due to multiple channel partitions in a traditional soft-output FSD is reduced with the help of the fine-search capability of GS. Apart from algorithmic discussions, related VLSI architecture of HSMCD is given to demonstrate the feasibility of low-complexity and high-throughput realizations, which hasn’t been thoroughly investigated in prior work.

The remainder of this paper is organized as follows: the MIMO system model along with FSD and MCMC algorithms are reviewed in Section II. Next, the proposed HSMCD architecture and its algorithm/circuit-level simplification are presented in Section III. Simulation results and comparisons are given in Section IV. Section V concludes the paper.

I. INTRODUCTION

MIMO technology, featuring its high spectral efficiency and space-time-coding capability, has been widely used in today’s wireless communication systems to improve the link throughput and the transmission robustness. In order to resolve the exponential-complexity problem with optimal ML detection of MIMO signals, several algorithms such as depth-first [1]-[2], breadth-first [3]-[7], and probabilistic [8][9] detectors have been proposed. Depth-first sphere detectors (SDs) are able to achieve ML performance by allowing sufficient runtime and applying radius-shrinking techniques. The breadth-first SDs, on the other hand, slightly degrade the performance in exchange for better processing speed via pipelined and parallel architectures. Probabilistic approaches (e.g. MCMC) are another options. Potential candidates are found based on conditional probabilities rather than deterministic constraints, such as the number of survivors in breadth-first or the value of search radius in depth-first SDs. The probabilistic method has been reported to achieve linear (or at most polynomial) complexity in terms of the number of visited nodes but with degraded performance in the high-SNR regime [8].

With tighter constraints on processing cycles and higher demands for detection quality required from future systems, hybrid architectures that combine the advantages of multiple stand-alone detection algorithms are drawing increasing attention. The multi-core SD proposed in [10] mixes depth-first and breadth-first search methods to speed up data processing while maintaining flexibility to examine more solutions within the search radius. The hybrid QRD-MCMC detector [11], on the other hand, uses QR decomposition based M-algorithm (QRD-M, a kind of breadth-first SD) to initialize MCMC, solving its high-SNR problem.

In this work, a soft-output hybrid sphere-MCMC detector (HSMCD) is proposed. This method uses FSD, which is a low-complexity variant of breadth-first detectors, to coarsely estimate the log-likelihood ratio (LLR) of each transmitted bit. It also serves as an initiator for GS, the simplest yet powerful MCMC algorithm, for subsequent fine searches. The required number of GS iterations in HSMCD is relaxed compared to traditional MCMC detector, thereby improving the overall processing speed. The problem of poor performance under high-SNR is also eliminated because the lower bound of BER performance is already guaranteed by FSD. Correspondingly, the resource overhead due to multiple channel partitions in a traditional soft-output FSD is reduced with the help of the fine-search capability of GS. Apart from algorithmic discussions, related VLSI architecture of HSMCD is given to demonstrate the feasibility of low-complexity and high-throughput realizations, which hasn’t been thoroughly investigated in prior work.

The remainder of this paper is organized as follows: the MIMO system model along with FSD and MCMC algorithms are reviewed in Section II. Next, the proposed HSMCD architecture and its algorithm/circuit-level simplification are presented in Section III. Simulation results and comparisons are given in Section IV. Section V concludes the paper.

II. MIMO DETECTION ALGORITHMS

A. System Model

Consider a multi-antenna system with M transmit and N receive antennas, in which the N × M matrix H describes the MIMO channel, the M × 1 vector s denotes the transmit symbol, and the N × 1 vector n represents the additive zero-mean i.i.d. complex Gaussian noise with variance σ². For each channel use, the bit sequence b = [b₁₁b₁₂...b₁,Qb₂₁b₂₂...b₂,Q...bM,Q]T is mapped to s via the predefined mapping function Φ(·) before transmission, where the bit element bᵢ,j ∈ {0,1}, and Q is the number of bits mapped to each symbol element. Specifically, the transmit symbol can be written as

$$s = [s₁s₂...s_M]^T = \Phi(b),$$  

where $$s_i = \Phi([b_i,1...b_i,Q]^T) \in \Omega$$ is selected from the constellation space Ω. The size of Ω equals 2Q, implying the total number of possible s equals 2MQ.

The N × 1 receive vector y is represented by

$$y = Hs + n,$$  

or equivalently,

$$\tilde{y} = Q²y = Rs + \tilde{n},$$
if $\mathbf{H}$ is decomposed through QR factorization, $\mathbf{Q}$ and $\mathbf{R}$ are the unitary and the upper-triangular matrices, respectively. Based on (3), the ML estimate of $\mathbf{s}$ and $\mathbf{b}$ can be written as

$$\hat{s}_{\text{ML}} = \arg \min_{\mathbf{s} \in \Omega^M} \| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2, \hat{b}_{\text{ML}} = \Phi^{-1}(\hat{s}_{\text{ML}}). \quad (4)$$

The ML criterion is optimal for providing hard information but should be modified to accommodate systems that use soft-in soft-out forward-error correctors (FECs), such as the LDPC decoder. An ideal soft-output MIMO detector should compute the bit-wise LLR value for those types of FECs:

$$L(b_{i,j}) = \log \left( \frac{\sum_{s=0}^{M} \exp \left( -\sigma^2 \| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2 \right) \delta_{b_{i,j}=0}}{\sum_{s=0}^{M} \exp \left( -\sigma^2 \| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2 \right) \delta_{b_{i,j}=1}} \right). \quad \text{ } (5)$$

(5) can also be approximated by the max-log approach suggested in [1]:

$$L(b_{i,j}) \approx \log \left( \frac{\max_{s=0}^{M} \exp \left( -\sigma^2 \| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2 \right) \delta_{b_{i,j}=0}}{\max_{s=0}^{M} \exp \left( -\sigma^2 \| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2 \right) \delta_{b_{i,j}=1}} \right) = \sigma^{-2} \left( \min_{s=0}^{M} \| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2 \right) \delta_{b_{i,j}=0} - \left( \min_{s=0}^{M} \| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2 \right) \delta_{b_{i,j}=1}. \quad \text{ } (6)$$

By comparing (4) and (6), it is clear that the original ML solution only equals one of the two hypotheses. Therefore, the motivation of this work is to reliably evaluate both of the hypotheses and provide the soft information for the FEC.

### B. Fixed-Complexity Sphere Decoder (FSD)

The graphical representation of $\| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \|^2$ is shown in Fig. 1(a). Since the $\mathbf{R}$ matrix is upper-triangular, the decision of a symbol element $s_i$ can only be affected by the results from $s_{i+1}$ to $s_M$. The partial Euclidean distance (PED) of $s_i$ is defined as

$$\text{PED}_i = \text{PED}_{i+1} + |b_i - r_{ii}\hat{s}_i|^2$$

$$b_i = \tilde{y}_i - \sum_{j=i+1}^{M} r_{ij}\hat{s}_j \quad \text{ } (7)$$

with the initial condition $\text{PED}_{M+1}$ set to zero. The ML solution can be found by searching the symbol vector $\mathbf{s}$ with minimum $\text{PED}_0$ or, equivalently, the “shortest” path in a tree topology (Fig. 1(b)). Each node at the $i$-th layer represents a symbol element $s_i$, and each complete path corresponds to a possible solution. The search process goes from the last ($M$-th) to the first layer, during which the selection of a node is highly dependent on the previously visited nodes along the sub-path. There exist a variety of strategies for choosing $s_i$ based on its PED and some distance constraints, forming the depth-first and the breadth-first SD families.

Figure 2(a) illustrates the search scheme of FSD. During the tree-search process, all of the symbols at the $M$-th layer are enumerated. For the rest of the layers, only the symbol with the smallest incremental distance, $|\|b_i - r_{ii}\hat{s}_i\|^2 - b_{i,j}|$, is chosen. Instead of exhaustively searching the entire vector space of size $|\Omega|^M$, only $|\Omega|$ candidates are examined by each process to approach (4) or (6), thus reducing the order of complexity from exponential to polynomial [12].

The hard-output FSD was first proposed in [4], with a soft-output version reported in [5]. The main difference between these two approaches lies in their channel partition strategies. Fig. 2(b) shows the basic architecture of FSD. In this figure, the hard FSD reorders the channel matrix such that the column with the smallest norm (assume $\mathbf{h}_M$ without loss of generality) is put in the last place. This guarantees the signal with the largest noise amplification can be fully examined in the last layer. In contrast, soft FSD adopts $M$ partitions to allow symbol elements from each transmit antenna a chance to be completely scanned. All of the hypotheses in (6) will be evaluated at least once by the (at most) $M \times |\Omega|$ candidates enumerated in soft FSD. This, however, causes resource overhead due to parallel realizations of FSDs. In a $4 \times 4$ 16-QAM system, the soft FSD requires a $4 \times 4$ silicon area compared to the hard FSD, though it is still 20% smaller than the typical breadth-first detector with similar performance [13].

### C. Markov Chain Monte Carlo (MCMC) Detector

Shown in Algorithm 1 is the MCMC algorithm that uses the Gibbs sampler (GS) to enumerate a list of the most likely transmit vectors. It summarizes the results in [8] and assumes that no information exchange between MIMO detector and FEC is performed in this work. The GS starts by randomly initializing a bit sequence $\mathbf{b}^0_{0,1}$. It then iteratively updates one bit at a time based on the transient conditional probability.
Initialization: \( b_{k+1}^i = [ b_{k,1}^i \cdots b_{M,M}^i ] \)

1: for \( k = 1, \ldots, K \) do
2:    for \( i = 1, \ldots, M \) do
3:        for \( j = 1, \ldots, Q \) do
4:            Set \( b_{i, j}^k = 0 \) with probability \( P_{i, j}^k = \left[ 1 + e^{-\varepsilon \cdot s_{i, j}^k} \right]^{-1} \),
            where \( \varepsilon = \| \tilde{y} - R_s y_{i, j}^k \| / \| \tilde{y} - R_s y_{i, j, 0} \| \), and
            \( s_{i, j, 0/1} = \Phi \left( \left[ b_{1,1}^k \cdots b_{i-1,1}^k 0/1 b_{i+1,1}^k \cdots b_{M,M}^i \right]^T \right) \).
5:        Calculate \( d_{i, j}^k = \| \tilde{y} - R_s b_{i, j+1}^k \| \),
            where \( b_{i, j+1}^{k-1} = [ b_{i, 1}^k \cdots b_{i-1, 1}^k b_{i, j}^k b_{i+1, 1}^k \cdots b_{M,M}^i ]^T \).
6:        Use \( d_{i, j}^k \) and \( b_{i, j+1}^{k-1} \) to evaluate \( L(b_{i, j}) \).
7:    end for
8: end for
9: end for

**Algorithm 1:** MCMC detection using Gibbs sampling

Fig. 3. Illustration of MCMC detection in a 4×4 BPSK system. In each iteration, the bit of interest is set to 0 with probability formulated by GS. A complete GS iteration is done when all bits are updated once.

\( P_{i, j}^k \) (step 4 of the algorithm). A complete Gibbs sample is generated when the last bit is updated. After that, the iteration goes back to the first bit and continues the process over again, as shown in Fig. 3. Since the whole process can be viewed as a Markov chain, the Gibbs samples are expected to gradually converge to the optimal solution or a steady state. As suggested in [8], running multiple GSs in parallel significantly improves the performance. The reason is intuitive: since each GS starts at different states of the Markov chain, the probability of getting promising hypotheses in (6) is higher compared to the scheme that uses a single GS. Though MCMC is claimed to approach ML solution with low complexity, it has poor performance in the high-SNR regime. The sequential iteration nature of GS also hinders its applicability to high-throughput hardware realizations. Solutions to these problems will be shown in the next section.

**III. Hybrid Sphere-MCMC Detector (HSMCD)**

As stated in the previous section, soft FSD is area inefficient due to multiple partitions. On the other hand, MCMC is vulnerable in the high-SNR regime and should be further optimized to raise its throughput. The hybrid sphere-MCMC detector (HSMCD) is proposed to tackle these issues. Figure 4 shows the architecture of HSMCD, which consists of a single FSD and \( P \) parallel GSs for the evaluation of (6). Instead of using only the minimum-distance candidate of QRD-M(9) SD (QRD-M with \( M \) equals nine) to initialize one of the parallel GSs as shown in [11], the proposed detector makes all of the GSs initialized with the minimum \( P \) possible solutions found by FSD. The idea arises from several reasons. First, the FSD is able to achieve comparable performance to the QRD-M(9) SD with a \( 7 \times \) lower complexity in terms of the number of visited nodes, thus qualifying as a good initiator for GS (Fig. 5). Second, the initialization period before GS converges can be shortened with the help of FSD, thereby converging to the steady state faster. Third, for each symbol vector \( \hat{s} \) found by FSD, its distance to \( \tilde{y} \) and its metric \( \| \tilde{y} - R_s \hat{s} \| \) can be directly adopted by GSs to calculate \( P_{i, j}^k \) in an iterative way, as demonstrated later. Since the hardware design of FSD is similar to existing breadth-first SDs, the following subsection will concentrate on the complexity reduction of GS.
A. Complexity Reduction of Gibbs Sampler

Figure 6(a) shows the direct-mapped GS. In each iteration, a total of $4M^2 + 4M$ real multiplications are required to get $\varepsilon$, making it the most computationally intensive block. In order to eliminate the redundancy introduced by the squared $L^2$-norm and the metric calculations for $s_{i,j,0}$ and $s_{i,j,1}$, it is necessary to reformulate $\varepsilon$. The simplification starts by noting that either $s_{i,j,0}$ or $s_{i,j,1}$ will equal the mapping result of $b_{i,j}^{k-1}$, regardless of value of $\varepsilon$. Using this property, $\varepsilon$ can be rewritten as

$$\varepsilon = \begin{cases} 2 \left( \tilde{y} - R s_{i,j,0}^k \right) \cdot (R \Delta) + \left\| R \Delta \right\|^2, & s_{i,j,0} = 1 \Phi \left( b_{i,j}^{k-1} \right) \\ 2 \left( \tilde{y} - R s_{i,j,1}^k \right) \cdot (R \Delta) - \left\| R \Delta \right\|^2, & s_{i,j,1} = 1 \Phi \left( b_{i,j}^{k-1} \right) \end{cases}$$

(8)

where $\Delta = s_{i,j,0}^k - s_{i,j,1}^k$. The operator $\cdot$ computes the summation of element-wise inner-product, namely $a \cdot b = \sum_{i=1}^{M} \Re (a_i) \Re (b_i) + \Im (a_i) \Im (b_i)$. Since $s_{i,j,0}$ and $s_{i,j,1}$ are mapped from bit sequences where only a single bit is different, it can be expected that $s_{i,j,0}$ and $s_{i,j,1}$ only differ in the $i$-th symbol. For this reason, $\Delta$ can be simplified as $\Delta = [0...0\Delta_i(0...0)^T]$, where

$$\Delta_i = s_{i,j,0} - s_{i,j,1} = \Phi \left( b_{i,j}^{k-1} \right) \cdot (R \Delta) + \left\| R \Delta \right\|^2, \quad (10a)$$

and

$$\varepsilon = \begin{cases} 2 \left( \tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right) \right) \cdot (\Delta_i \cdot r_i) + \left\| \Delta_i \right\|^2 \left\| r_i \right\|^2, & b_{i,j}^{k-1} = 0 \\ 2 \left( \tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right) \right) \cdot (\Delta_i \cdot r_i) - \left\| \Delta_i \right\|^2 \left\| r_i \right\|^2, & b_{i,j}^{k-1} = 1 \end{cases}$$

(11)

where $r_i$ denotes the $i$-th column of the $R$ matrix. Since $\Delta_i$ is derived from the mapping results of two finite symbols, it also belongs to a finite set. Accordingly, $\Delta_i$ and $\left\| \Delta_i \right\|^2$ can be generated using simple logic, resulting in the modified architecture shown in Fig. 6(b). However, the complexity reduction is marginal because $d_{i,j}^k$, one of the required components for LLR evaluation, cannot be explicitly acquired using (11). The metric and distance update schemes are therefore proposed to solve this problem.

Assume that bit $b_{i,j}^{k-1}$ equals 0 before it becomes $b_{i,j}^{k-1}$, without loss of generality. By definition, this corresponds to the metric $M_0 = \tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right)$. It also implies $M_1 = \tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right) - \Delta = \tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right) + \Delta_i r_i$, according to (9). If the bit remains the same after the iteration, i.e. $b_{i,j}^{k} = b_{i,j}^{k-1}$, then $M_0$ doesn’t need any change. Otherwise, $M_1$ should replace $M_0$ to become the new metric. Summarizing all the possible cases yields the following metric update equation:

$$\tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right) = \begin{cases} \tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right) + \Delta_i r_i, & b_{i,j}^{k-1} = 0 \text{ and } b_{i,j}^{k} = 1 \\ \tilde{y} - R \Phi \left( b_{i,j}^{k-1} \right) - \Delta_i r_i, & b_{i,j}^{k-1} = 1 \text{ and } b_{i,j}^{k} = 0 \end{cases}$$

(12)

The distance update can be derived similarly:

$$d_{i,j}^k = \begin{cases} d_{i,j}^{k-1} + \varepsilon, & b_{i,j}^{k-1} = 0 \text{ and } b_{i,j}^{k} = 1 \\ d_{i,j}^{k-1} - \varepsilon, & b_{i,j}^{k-1} = 1 \text{ and } b_{i,j}^{k} = 0 \end{cases}$$

(13)

Realizing (11)-(13) gives the architecture shown in Fig. 6(c). No extra cost is added since the bit sequence, metric, and distance are already initialized by the FSD block.

The evaluation of $P_{i,j}$ can be done after $\varepsilon$ is ready. To implement this nonlinear function, either a look-up table (LUT) or a (bi-)linear approximation scheme can be used. A digital randomizer is then used to decide $b_{i,j}$ based on the value of $P_{i,j}$.

The number of real multiplications is counted as a first-order complexity estimate of each architecture. In each iteration, $4M^2 + 4M$ multiplications are required in the direct-mapped GS, while the second and the third version costs $2M^2 + 3M + 1$ and $M + 1$, respectively. A $4M$-times reduction is therefore achieved from the arithmetic complexity perspective, which translates to a $16 \times$ saving when a $4 \times 4$ system is considered.
The critical path is also greatly reduced since the real-time mapping, metric, and distance calculations are all removed from the iteration loop.

The realization of (10) needs more attention. In order to efficiently implement (10), the following two properties of $\Delta_i$ are exploited: 1) Both $\Delta_i$ and $\|\Delta_i\|^2$ are finite and can be computed using simple logic; 2) Though $\Delta_i$ is a complex number by definition, it only contains either the real or the imaginary part in practice. In other words, these two parts cannot appear in $\Delta_i$ at the same time. Fig. 7(a) shows the calculation of $\Delta_i$ in the 16-QAM case. Although not shown due to the limited space available, derivations for other symbol-mapping options such as QPSK and 64-QAM can be constructed similarly. By observing (9) and the constellation plot in Fig. 7(a), it is seen that $\Delta_i$ can only be equal to 2 (if $\Re(s_{i,j,0}) = 1$ or 6 if $\Re(s_{i,j,0}) = 3$) when the GS iteration goes to $b_{i,3}$ (see highlighted dots on the right). Because the value of $\Re(s_{i,j,0})$ depends solely on $b_{i,2}$, it is concluded that

$$\Delta_i = \begin{cases} 6, & b_{i,2} = 1 \\ 2, & b_{i,2} = 0 \end{cases}$$

when $b_{i,3}$ is to be updated. Likewise, $\Delta_i$ can only equal $+2$ or $-2$ according to the value of $b_{i,3}$, when $b_{i,2}$ is examined. The cases for $b_{i,1}$ and $b_{i,0}$ are similar, except that $\Delta_i$ becomes an imaginary number. These results are summarized on the right of Fig. 7(a) in a C-language style.

The realization of (10) is depicted in Fig. 7(b) based on the above discussion. Because only a few possible values are applied to the multiplications with $r_i$ and $|r_i|^2$, the datapath control scheme is adopted. In each GS iteration, appropriate control signals are produced to configure the multiplexers. The notation “$\ll i$” means to shift the input left by $i$ bit(s), which is equivalent to multiplying it by $2^i$. The possible values of $\Delta_i$, i.e. $\{6, 2, -2\}$, are rewritten as $\{2(2^2 - 1), 2(2^1 - 1), 2(0 - 1)\}$ so that a unified architecture can be employed to handle all of the cases. Putting these simplification techniques altogether yields the multiplier-free circuit shown, where the highlighted part is used to perform multiplications with real or imaginary numbers. If $\Delta_i$ is real, the incoming data (r_re and r_im) will go straight into the next stage; otherwise they will be swapped and negated before moving forward.

### B. Flexible Design

The regularity of the proposed HSMCD architecture is the key factor that makes it flexible to support various symbol mapping types from QPSK to 64-QAM, and antenna array sizes from $2 \times 2$ to $4 \times 4$. The number of parallel GSs is also configurable based on channel condition. More parallelism is required in the high-SNR regime because GS can easily get stuck before reaching the optimum steady state. When less parallelism is adopted, the power-gating technique is used to lower the processing energy. The detector is also compatible to hard-output generation. In this case, all of the GSs are turned off, leaving only the hard FSD enabled.

### IV. Simulation Results

In this section, the error performance of HSMCD is evaluated and compared with other detection algorithms. A $4 \times 4$ MIMO system encoded by (1944,972) rate-1/2 irregular LDPC code is considered. The corresponding LDPC decoder with 20 iterations is used at the receiver. A spatially uncorrelated Rayleigh fading channel with perfect channel state information is assumed.
The coded BER curves of various algorithms are given in Fig. 8. Here, HSMCD($P,I$) denotes a HSMCD with $P$ parallel GSs, each taking $I$ iterations. Simulation results show that HSMCD(8,2) already outperforms soft FSD by 1 dB at $10^{-5}$ BER. It also achieves comparable performance to MCMC(24,4). This can be interpreted as a $2 \times$ throughput improvement since the parameter $I$, which dominates the overall processing time, is reduced from four to two. Complexity comparisons between FSD, QRD-M, MCMC, and HSMCD are summarized in Table I and Fig. 9. It is seen that QRD-M(16) performs the best but is 80% more complex than HSMCD(8,2) when transmitting 16-QAM signals. It is also interesting to note that the advantage of HSMCD becomes more appreciable when 64-QAM signals are transmitted. In this case, the complexity of QRD-M(16) and soft FSD are $2.9 \times$ and $2.3 \times$ greater than HSMCD, respectively. This is because the number of real multiplications in HSMCD only scales with $\log_2 ||\Omega||$, while other SD algorithms scale with $||\Omega||$. In summary, the proposed HSMCD(8,2) is able to achieve a $2 \times$ throughput improvement and a $2.3 \times$ complexity reduction compared to stand-alone MCMC and soft-output FSD, respectively.

V. CONCLUSION

A hybrid soft-output MIMO detector aiming for efficient hardware implementation is proposed in this paper. The soft-information is searched efficiently by using one fixed-complexity sphere detector and eight parallel Gibbs samplers in both deterministic and probabilistic domains. The proposed VLSI architecture of the Gibbs sampler has a $16 \times$ lower computational complexity compared to the direct-mapped re-alization by using equivalent mathematical expressions. The hybrid MIMO detector is flexible to support modulations from QPSK to 64-QAM, antenna size from $2 \times 2$ to $4 \times 4$, and compatibility to hard-output generation due to architectural regularity. It achieves better BER performance than individual soft-output FSD and MCMC detectors through joint detection. A $2.3 \times$ hardware complexity reduction and a $2 \times$ throughput improvement are achieved compared to the non-hybrid realization.

ACKNOWLEDGMENT

This work is supported by the Focus Center for Circuit & System Solutions (C2S2), one of six research centers funded under the Focus Center Research Program, a Semiconductor Research Corporation Program.