VLSI Implementation of MIMO Detection Using the Sphere Decoding Algorithm

Andreas Burg, *Member, IEEE,* Moritz Borgmann, *Student Member, IEEE,* Markus Wenk, Martin Zellweger, Wolfgang Fichtner, *Fellow, IEEE,* and Helmut Bölcskei, *Senior Member, IEEE*

***Abstract*— Multiple-input multiple-output (MIMO) techniques are a key enabling technology for high-rate wireless communications. This paper discusses two ASIC implementations of MIMO sphere decoders. The first ASIC attains maximum-likelihood performance with an average throughput of 73 Mbps at a signal-to-noise ratio (SNR) of 20 dB; the second ASIC shows only a negligible bit error rate degradation and achieves a throughput of 170 Mbps at the same SNR. The three key contributing factors to high throughput and low complexity are: Depth-first tree traversal with radius reduction, implemented in a *one-node-per-cycle* architecture, the use of the ℓ^∞- instead of ℓ^2-norm, and finally the efficient implementation of the enumeration approach recently proposed in [1]. The resulting ASICs currently rank among the fastest reported MIMO detector implementations.**

***Index Terms*— Detection, maximum likelihood (ML), multiple-input multiple-output (MIMO), spatial multiplexing, sphere decoding, very large scale integration (VLSI), wireless communications.**

I. Introduction

THE success of wireless communications has mainly been associated with a continuous increase in system capacity and quality of service. As bandwidth is a scarce resource, this trend can only be continued by using new technologies that provide higher spectral efficiency and improved link reliability. Multiple-input multiple-output (MIMO) communication systems [2], which use multiple antennas at both sides of the wireless link, have recently emerged as a key enabling technology for meeting these requirements. MIMO techniques have been proposed as extensions to current wireless communication standards such as IEEE 802.11 and HSDPA and are part of emerging standards such as IEEE 802.16.

The performance improvements resulting from MIMO wireless technology come at the cost of increased computational complexity in the receiver (and often the transmitter as well). The design of low complexity receivers is, therefore, one of the key problems in MIMO wireless system design. The largest potential for complexity reduction of highest-performance VLSI circuits for signal processing is in the joint optimization of both the algorithms and the register transfer level architecture with the circuit level trade-offs in mind. The actual circuit synthesis can be left to automatic tools. Such a combined approach is critical to achieve practicable solutions for next-generation wireless communication systems.

Before describing the main contributions of this paper, we introduce the system model for a narrowband MIMO link and briefly discuss basic trade-offs in MIMO receiver algorithms.

A. Narrowband MIMO System Model

Our equivalent complex-valued discrete-time baseband system model is as follows: In a MIMO system with M_T transmit and M_R receive antennas, the M_R-dimensional received signal vector is given by

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

where $\mathbf{H}$ denotes the $M_R \times M_T$ channel matrix, $\mathbf{s} = \begin{bmatrix} s_1 & s_2 & \dots & s_{M_T} \end{bmatrix}^T$ is the M_T-dimensional transmit signal vector, and $\mathbf{n}$ stands for the M_R-dimensional additive i.i.d. circularly symmetric complex Gaussian noise vector. The entries of $\mathbf{s}$ are chosen independently from a complex constellation $\mathcal{O}$ with Q bits per symbol, i.e., $|\mathcal{O}| = 2^Q$. The set of all possible transmitted vector symbols is denoted by $\mathcal{O}^{M_T}$. The corresponding uncoded transmission rate is $R = M_T Q$ bits per channel use (bpcu). We furthermore assume $M_R \geq M_T$ throughout the paper. For our numerical simulations, the entries of $\mathbf{H}$ are modeled as i.i.d. Rayleigh fading.

B. MIMO Detection

We assume that the receiver has acquired knowledge of the channel $\mathbf{H}$ (e.g., through a preceding training phase). Algorithms to separate the parallel data streams corresponding to the M_T transmit antennas can be divided into four categories:

1) *Linear detection methods* invert the channel matrix using a zero-forcing (ZF) or minimum mean squared error (MMSE) criterion. The received vectors are then multiplied by the channel inverse, possibly followed by slicing. The drawback is, in general, a rather poor bit error rate (BER) performance.
2) *Ordered successive interference cancellation* decoders such as the vertical Bell Laboratories layered space-time (V-BLAST) algorithm show slightly better performance, but suffer from error propagation and are still suboptimal.
3) *Maximum likelihood (ML) detection*, which solves

$$\hat{\mathbf{s}} = \underset{\mathbf{s} \in \mathcal{O}^{M_T}}{\arg\min} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 \tag{2}$$

is the optimum detection method and minimizes the BER. A straightforward approach to solve (2) is an exhaustive

Manuscript received November 13, 2004; revised January 22, 2005. This paper was presented in part at ESSCIRC, Leuven, Belgium, September 2004, and in part at the IEE 3G Mobile Conference, London, U.K., October 2004.

The authors are with the Swiss Federal Institute of Technology (ETH) Zurich, 8092 Zurich, Switzerland (e-mail: apburg@iis.ee.ethz.ch, moriborg@nari.ee.ethz.ch, fw@iis.ee.ethz.ch, boelcskei@nari.ee.ethz.ch).

Digital Object Identifier 10.1109/JSSC.2005.847505

search. Unfortunately, the corresponding computational complexity grows exponentially with the transmission rate R, since the detector needs to examine 2^R hypotheses for each received vector. While the implementation of exhaustive-search ML has been shown to be feasible in the low rate regime $R \leq 8\,\mathrm{bpcu}$ [3], complexity quickly becomes unmanageable as the rate increases [4], [5]. For example, in a 4×4 system (i.e., $M_T = M_R = 4$) with 16-QAM modulation (corresponding to $R = 16\,\mathrm{bpcu}$), 65 536 candidate vector symbols have to be considered for each received vector.

4) *Sphere decoding* (SD) solves the ML detection problem [6]–[8]. While the algorithm has a nondeterministic instantaneous throughput, its average complexity was shown to be polynomial in the rate [9] for moderate rates, but still exponential in the limit of high rates [10]. However, these asymptotic results do not properly reflect the true implementation complexity of the algorithm, which for most practical cases is still significantly lower than an exhaustive search. The algorithm is thus widely considered the most promising approach towards the realization of ML detection in high-rate MIMO systems. Ever since its introduction in [6] and its application to wireless communications in [8], reduction of the computational complexity of the algorithm has received significant attention [11], [12], [8], [13]. However, most modifications of the algorithm proposed in the literature so far have been suggested with digital signal processor (DSP) implementations in mind. Little attention has been paid to the efficient VLSI implementation of the SD algorithm and the associated performance trade-offs. To the best of our knowledge, the only references dealing with suitable hardware architectures and actual ASIC implementations of SD are [14]–[17], [5].

C. Contributions

The main contributions of this paper are summarized as follows:

- We present two ASIC implementations of *depth-first* sphere decoding, which, to the best of our knowledge, are the first reported ASICs implementing the algorithm and are currently ranked among the fastest reported MIMO detectors.
- We introduce a *one-node-per-cycle* hardware architecture for the efficient implementation of SD with depth-first tree traversal.
- A modified sphere criterion is proposed based on the ℓ^∞-norm instead of the squared ℓ^2-norm, which reduces complexity on the algorithmic *and* on the circuit level at only a small SNR penalty.
- We show that for the purpose of VLSI implementation, the widely used real-valued decomposition is ill-suited. Instead, we describe two methods to operate directly on the complex constellations using an exhaustive search or a low-complexity refinement of a scheme proposed in [1].

D. Outline

The next section briefly reviews the state of the art in low-complexity SD and points out the critical implementation aspects. Subsequently, in Section III, an efficient generic high-level VLSI architecture is described for SD implementation based on the *depth-first* strategy. In Section IV, we show how a suitable modification of the decoding metric can result in significant throughput improvements at the cost of a negligible SNR penalty. In Sections V and VI, we describe in detail two ASIC implementations. The two ASICs are compared and the results are put into perspective with different other reported MIMO detector implementations in Section VII.

II. Sphere Decoding Algorithm

In this section, we briefly review the basics of SD, and we outline what we consider the corresponding state of the art. Our description summarizes the original algorithm [6], introduced by Pohst, and its subsequent extensions and improvements [11], [8], [12], [13]. We distinguish four key concepts, which we describe in the following.

A. Sphere Constraint

The main idea in SD is to reduce the number of candidate vector symbols to be considered in the search that solves (2), *without* accidentally excluding the ML solution. This goal is achieved by constraining the search to only those points $\mathbf{Hs}$ that lie inside a hypersphere with radius r around the received point $\mathbf{y}$. The corresponding inequality is referred to as the *sphere constraint* (SC):

$$d(\mathbf{s}) < r^2 \quad \text{with} \quad d(\mathbf{s}) = \|\mathbf{y} - \mathbf{Hs}\|^2. \tag{3}$$

B. Tree Pruning

Only imposing the SC (3) does not lead to complexity reductions as the challenge has merely been shifted from finding the closest point to identifying points that lie inside the sphere. Hence, complexity is only reduced if the SC can be checked other than again exhaustively searching through all possible vector symbols $\mathbf{s} \in \mathcal{O}^{M_T}$. Two key elements allow for such a computationally efficient solution:

1) Computing Partial Euclidean Distances: We start by noting that the channel matrix $\mathbf{H}$ in (3) can be triangularized using a QR decomposition according to $\mathbf{H} = \mathbf{QR}$, where the $M_R \times M_T$ matrix $\mathbf{Q}$ has orthonormal columns (i.e., $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_{M_T}$), and the $M_T \times M_T$ matrix $\mathbf{R}$ is upper triangular. It can easily be shown [13] that

$$d(\mathbf{s}) = c + \|\hat{\mathbf{y}} - \mathbf{Rs}\|^2 \quad \text{with} \quad \hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y} = \mathbf{Rs}^{\mathrm{ZF}} \tag{4}$$

where $\mathbf{s}^{\mathrm{ZF}}$ is the zero-forcing (or unconstrained ML) solution[1] $\mathbf{s}^{\mathrm{ZF}} = \mathbf{H}^\dagger\mathbf{y}$. The constant c is independent of the vector symbol $\mathbf{s}$ and can hence be ignored in the metric computation. In the following, for simplicity of exposition, we set $c = 0$.

If we build a tree such that the leaves at the bottom correspond to all possible vector symbols $\mathbf{s}$ and the possible values of the entry s_{M_T} define its top level, we can uniquely describe each

[1] $\mathbf{H}^\dagger$ denotes the pseudoinverse of $\mathbf{H}$.

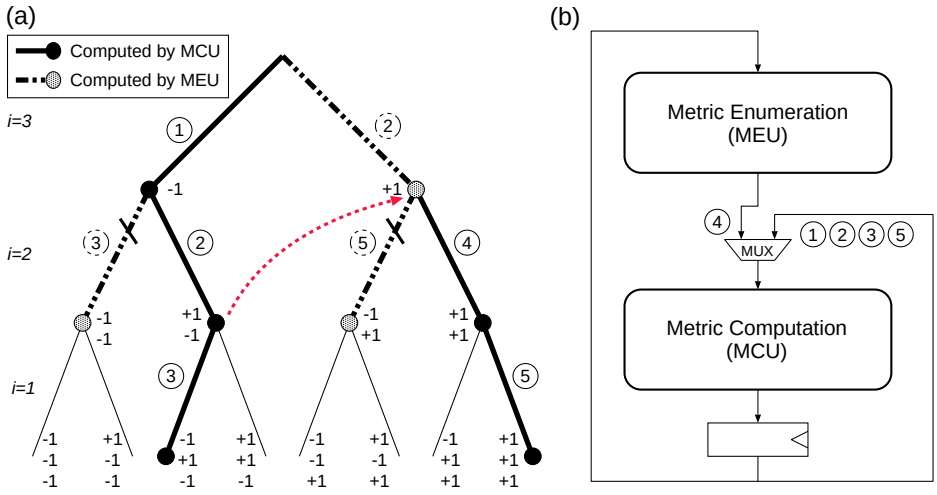

Fig. 1. One-node-per-cycle depth-first tree traversal example (a) and high-level architecture (b). The numbers on the branches indicate the cycles in which the corresponding children are processed by the MCU or by the MEU.

node at level i ($i = 1, 2, \ldots, M_T$) by the *partial vector symbols* $\mathbf{s}^{(i)} = \begin{bmatrix} s_i & s_{i+1} & \ldots & s_{M_T} \end{bmatrix}^T$, as illustrated in Fig. 1(a) for a 3×3 system with BPSK modulation. Now, we can recursively compute the (squared) distance $d(\mathbf{s})$ by traversing down the tree and effectively evaluating $d(\mathbf{s})$ in (4) in a row-by-row fashion: We start at level $i = M_T$ and set $T_{M_T+1}\big(\mathbf{s}^{(M_T+1)}\big) = 0$. The *partial (squared) Euclidean distances* (PEDs) $T_i\big(\mathbf{s}^{(i)}\big)$ are then given by

$$T_i\big(\mathbf{s}^{(i)}\big) = T_{i+1}\big(\mathbf{s}^{(i+1)}\big) + |e_i\big(\mathbf{s}^{(i)}\big)|^2 \tag{5}$$

with $i = M_T, M_T - 1, \ldots, 1$, where the *distance increments* $|e_i\big(\mathbf{s}^{(i)}\big)|^2$ can be obtained as

$$|e_i\big(\mathbf{s}^{(i)}\big)|^2 = \left| \hat{y}_i - \sum_{j=i}^{M_T} R_{ij} s_j \right|^2. \tag{6}$$

We can make the influence of s_i more explicit by writing

$$|e_i\big(\mathbf{s}^{(i)}\big)|^2 = \left| b_{i+1}\big(\mathbf{s}^{(i+1)}\big) - R_{ii} s_i \right|^2 \quad \text{with} \tag{7}$$

$$b_{i+1}\big(\mathbf{s}^{(i+1)}\big) = \hat{y}_i - \sum_{j=i+1}^{M_T} R_{ij} s_j. \tag{8}$$

Finally, $d(\mathbf{s})$ is the PED of the corresponding leaf: $d(\mathbf{s}) = T_1(\mathbf{s})$. Since the distance increments $|e_i\big(\mathbf{s}^{(i)}\big)|^2$ are nonnegative, it follows immediately that whenever the PED of a node violates the (partial) SC given by

$$T_i\big(\mathbf{s}^{(i)}\big) < r^2 \tag{9}$$

the PEDs of all its children will also violate the SC. Consequently, the tree can be pruned above this node. This approach effectively reduces the number of vector symbols (i.e., leaves of the tree) to be checked.

2) Tree Traversal and Radius Reduction: When the tree traversal is finished, the leaf with the lowest $T_1(\mathbf{s})$ corresponds to the ML solution. The traversal can be performed *breadth-first* or *depth-first*. In both cases, the number of nodes reached and hence the decoding complexity depend critically on the choice of the radius r. The K-best algorithm [14], [15] approximates a breadth-first search by keeping only (up to) K nodes with the smallest PEDs at each level. The advantage of the K-best algorithm over a full (depth-first or breadth-first) search is its uniform data path and a throughput that is independent of the channel realization and the SNR. However, the K-best algorithm does not necessarily yield the ML solution.

In a *depth-first* implementation, the complexity and dependence of the throughput on the *initial radius* can be reduced by shrinking the radius r whenever a leaf is reached. This procedure does not compromise the optimality of the algorithm, yet it decreases the number of visited nodes compared to a constant radius procedure. As an added advantage of the depth-first approach with radius reduction, the initial radius may be set to infinity, alleviating the problem of initial radius choice. However, in contrast to the K-best algorithm, a depth-first traversal does not yield a deterministic throughput. In this paper, we consider *only depth-first* tree traversal with infinite initial radius.

C. Admissible Sets

The *admissible set* of children $\mathbf{s}^{(i)}$ of a particular parent $\mathbf{s}^{(i+1)}$ in the tree is simply defined by the constellation points s_i for which the PED satisfies $T_i\big(\mathbf{s}^{(i)}\big) < r^2$. In the case of real-valued constellations, one can determine the boundaries of an *admissible interval* using (6) in conjunction with (5) and the partial SC (9). All *admissible children* are then contained within these boundaries. Unfortunately, in the practically more relevant case of complex-valued constellations, admissible intervals cannot be specified. A solution for QAM constellations that is frequently found in the literature is to decompose the M_T-dimensional complex signal model in (1) into a $2M_T$-dimensional real-valued problem according to[2]

$$\begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} = \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \begin{bmatrix} \Re\{\mathbf{s}\} \\ \Im\{\mathbf{s}\} \end{bmatrix} + \begin{bmatrix} \Re\{\mathbf{n}\} \\ \Im\{\mathbf{n}\} \end{bmatrix}. \tag{10}$$

This approach results in a tree that is twice as deep as the original tree (corresponding to the complex-valued formulation) with a

[2] $\Re\{\mathbf{x}\}$ and $\Im\{\mathbf{x}\}$ denote the real and imaginary parts, respectively, of $\mathbf{x}$.

smaller number of children per node. The number of leaves remains unchanged. However, we will argue later that performing SD directly on the complex constellation is more efficient in VLSI implementations.

D. Optimum Ordering

With radius reduction, it is desirable to find candidate solutions that lie close to the ML solution as early as possible in order to shrink the sphere as fast as possible and hence expedite the tree pruning. A scheme proposed by Schnorr and Euchner [12] and modified for the finite lattice case in [13] traverses the members of the admissible sets in ascending order of their PEDs. In the case of real-valued lattice constellations, given a starting point and an initial direction, this ordering is predefined. The decoder starts with the center of the admissible interval and proceeds to the boundaries in a zig-zag fashion. As shown in [13], there is no need to explicitly compute the boundaries; instead, due to the Schnorr–Euchner (SE) ordering, it is sufficient to terminate once the SC is violated. In the case of complex-valued constellations, SE ordering is still possible even without the real-valued decomposition (10). However, depending on the constellation, no obvious predefined order may exist. Hence explicit sorting of the admissible children by their PEDs may be required, incurring a high implementation complexity.

III. One-Node-Per-Cycle Architecture

The VLSI architecture of the two SD ASICs described in Sections V and VI implements depth-first tree traversal. Hardware utilization is maximized when the decoder visits *a new node in each cycle* and when *no node is ever visited twice*. This property can be achieved with an isomorphic VLSI architecture that consists of two main entities:

1) The *metric computation unit* (MCU) is responsible for the forward recursion of the tree traversal. Given the PED $T_{i+1}\big(\mathbf{s}^{(i+1)}\big)$ of a parent node, it finds the starting point for the SE enumeration among the children together with the corresponding PED $T_i\big(\mathbf{s}^{(i)}\big)$. If the SC (9) is met, tree traversal proceeds to the next level ($i \leftarrow i-1$). If none of the children meets the SC, a dead end is declared. When a leaf is reached, the radius r is updated.
2) The *metric enumeration unit* (MEU) operates in parallel to the MCU and handles the backward recursion. To this end, it follows the MCU on its path with one cycle delay and chooses a *preferred child* for each node between the root and the node whose children are currently examined by the MCU. The choice is made according to the SE enumeration, and membership in the list of *preferred children* is conditioned on compliance with the SC. Once the MCU reaches a leaf or a dead end, the MEU can immediately select the next node to be visited (according to the *depth-first* paradigm) from the list of preferred children and provide the corresponding PED to the MCU in the next cycle. Decoding terminates when the list of preferred children is empty.

A simplified block diagram of the architecture is shown in Fig. 1(b). The decoding procedure is illustrated by means of an example in Fig. 1: The circled numbers represent the cycles in which the corresponding nodes of the tree are examined by the MEU and the MCU and also indicate the setting of the multiplexer [cf. Fig. 1(b)] in the respective cycles.

A. Performance Metric

The average throughput of the *one-node-per-cycle* architecture in bits per second is given by

$$\Phi = \frac{M_T Q}{\mathcal{E}\{D\} t_{\mathrm{CLK}}} \tag{11}$$

where $\mathcal{E}\{D\}$ is the expected number of visited nodes per vector symbol, and t_{CLK} is the length of the critical path of the circuit. $\mathcal{E}\{D\}$ can be evaluated and optimized on the algorithmic level, while t_{CLK} is governed by the circuit implementation.

B. Implications of Employing a Real-Valued Decomposition

In light of the architecture described above, the commonly used real-valued decomposition (10) needs to be reconsidered. The approach doubles the depth of the tree, compared to the tree corresponding to the complex-valued constellation. Hence, the expected number of nodes to be visited $\mathcal{E}\{D\}$ nearly doubles [17]. At the same time, the processing of a single node becomes simpler. However, in order to be able to compensate for the increase in $\mathcal{E}\{D\}$, it would be necessary to shorten the critical path to half of its original length. On the register transfer level, the main difference between the real-valued case and the complex-valued case lies in the higher degree of parallelism for the latter. Therefore, a reduction of t_{CLK} can hardly be achieved by using the real-valued decomposition; the only seizable advantage would be an area reduction. We can, therefore, conclude that operating directly on the complex-valued constellation is key to achieving high throughput.

IV. Simplified Norm Algorithm

The simplified norm algorithm was first introduced to SD in [16] and can be used to reduce complexity on both the circuit and the algorithmic level, at the cost of only a minor performance degradation. Setting[3] $T_i = X_i^2$ in the following, the main idea is to rewrite the partial SC (9) as

$$X_i = \sqrt{X_{i+1}^2 + |e_i|^2} < r \tag{12}$$

and to approximate the ℓ^2-norm by a different norm $X_i \approx f\big(|X_{i+1}|, |e_i|\big)$ such as the ℓ^1-norm or the ℓ^∞-norm, respectively, according to

$$X_i \approx |X_{i+1}| + |e_i| \tag{13}$$

or

$$X_i \approx \max\big(|X_{i+1}|, |e_i|\big). \tag{14}$$

In the complex-valued case, corresponding approximations for computing $|e_i|$ from $\Re\{e_i\}$ and $\Im\{e_i\}$ are given by

$$|e_i| \approx |\Re\{e_i\}| + |\Im\{e_i\}| \tag{15}$$

or

$$|e_i| \approx \max\big(|\Re\{e_i\}|, |\Im\{e_i\}|\big). \tag{16}$$

[3]For simplicity, we write $T_i = T_i\big(\mathbf{s}^{(i)}\big)$, $e_i = e_i\big(\mathbf{s}^{(i)}\big)$, and $b_{i+1} = b_{i+1}\big(\mathbf{s}^{(i+1)}\big)$ from now on.

A. Impact on Circuit Complexity

We shall next assess the implications of the PED approximations (13)–(16) on the circuit complexity. Let us start by considering the circuit in the top left corner of Fig. 2 along with the corresponding area/delay trade-off curves. The curves result from a combination of either (13) with (15) or (14) with (16). The reference circuit implements (5), i.e., it uses the squared ℓ^2-norm. In order to accurately capture the properties of the final circuit, the PED computation is followed by a comparator, which checks compliance with the SC. Design space exploration is performed by synthesizing the test circuits with different delay constraints that vary from the minimum achievable delay (with today's state-of-the-art synthesis tools) to the maximum delay obtained with area-only optimization.

Both the ℓ^1- and the ℓ^∞-norm entail shorter delay and significantly less area compared to the squared ℓ^2-norm. We shall, however, see later (cf. Section V-A) that in some cases, the area advantage may vanish in the context of the overall architecture. In comparison, the two approximations (13) with (15) and (14) with (16) span a similar area/delay trade-off curve and are thus almost equivalent from an implementation point of view. Differences exist only in the minimum delay or minimum area limits. In the latter case, the ℓ^∞-norm is slightly smaller but slower because the max function leads to an interrupted carry propagation profile. However, we shall next show that on the algorithmic level, there are significant performance differences between the ℓ^1-norm and the ℓ^∞-norm.

B. Impact on Tree Pruning

In order to fully assess the impact of the above described norm approximations on throughput, we shall next study the influence of the reduced-complexity norms on the average number of visited nodes $\mathcal{E}\{D\}$. For 4×4 and 6×6 systems with 16-QAM modulation, Fig. 3 shows the average number of visited nodes as a function of SNR for the different norms under investigation. Remarkably, the use of the ℓ^∞-norm approximation reduces the average number of visited nodes significantly, while the ℓ^1-norm has just the opposite effect. In order to obtain an intuitive understanding of this behavior, assume that the SD has arrived at the leaf that corresponds to the ML solution $\mathbf{s}^{\mathrm{ML}}$, a fact of which it is not aware until all other leaves have been pruned from the tree. The *residual radius* (after radius reduction) is given by the respective norm of the noise vector $\tilde{\mathbf{n}} = \mathbf{y} - \mathbf{H}\mathbf{s}^{\mathrm{ML}}$. Since $\|\tilde{\mathbf{n}}\|_\infty \leq \|\tilde{\mathbf{n}}\|_2 \leq \|\tilde{\mathbf{n}}\|_1$, it follows that the residual radius will depend on the detector type. A smaller residual radius tends to remove more nodes already at the higher levels of the tree, so that the pruning of the remaining nodes is expedited. Consequently, fewer nodes need to be visited before the tree has been pruned completely and the search can terminate. The result is the observed complexity reduction.

Fig. 3 also shows that the relative savings due to the ℓ^∞-norm grow with increasing M_T. While a general systematic analytical treatment of the impact of the norm on tree pruning seems difficult, the asymptotic scaling behavior (in M_T) of the complexity savings can be explained as follows. Consider the expected residual radius corresponding to the squared ℓ^2-norm and the ℓ^∞-norm as a function of M_T. In the case of the ℓ^2-norm, we find

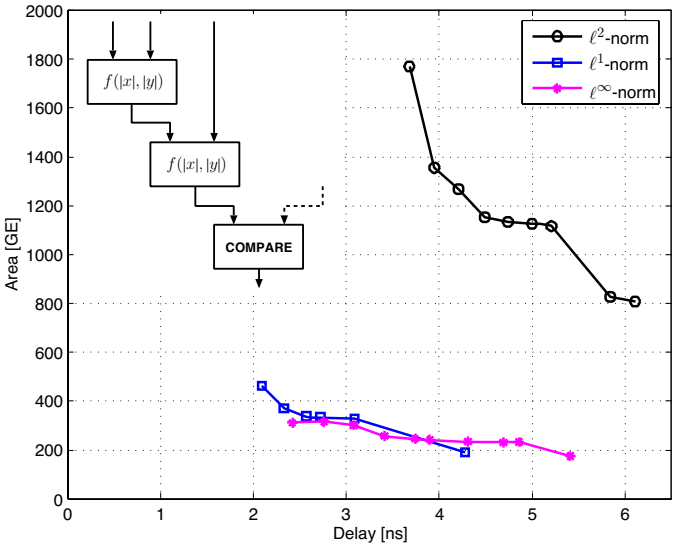

Fig. 2. Design space exploration for the computation of T_{i+1} and X_{i+1} using the squared ℓ^2-norm and the ℓ^1-norm and ℓ^∞-norm, respectively, followed by a comparator.

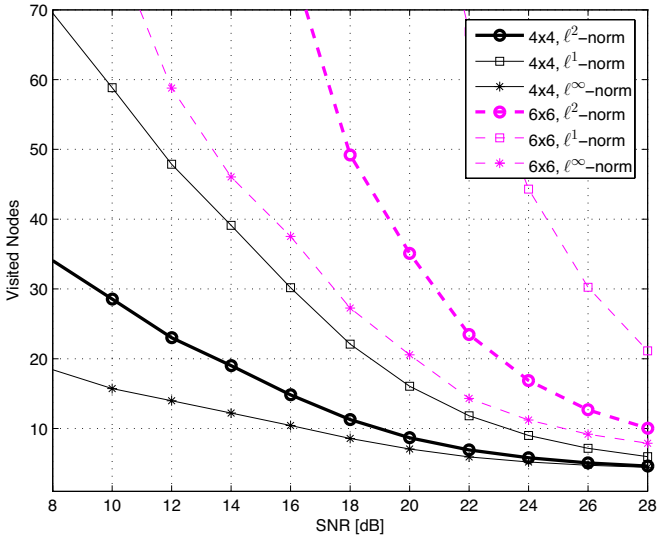

Fig. 3. Average number of visited nodes for the (squared) ℓ^2-norm and the ℓ^1- and ℓ^∞-norms, averaged over 250 000 channel realizations, for 4×4 and 6×6 systems with 16-QAM modulation.

that $\mathcal{E}\{\|\tilde{\mathbf{n}}\|_2\} \propto \sqrt{M_T}$, while for the ℓ^∞-norm $\mathcal{E}\{\|\tilde{\mathbf{n}}\|_\infty\} \propto \sqrt{\log M_T}$ [18], assuming $M_T = M_R$ for simplicity. Clearly, the radius and thus the number of visited nodes in the ℓ^2-norm case grows significantly faster with M_T than in the ℓ^∞-norm case.

C. Impact on BER Performance

Approximating the ℓ^2-norm by the ℓ^1-norm or the ℓ^∞-norm results in a modified SD algorithm that no longer implements an ML detector. The impact on BER of using the ℓ^1-norm or the ℓ^∞-norm instead of the (squared) ℓ^2-norm is quantified in Fig. 4 for a 4×4 system with 16-QAM modulation. It can be shown analytically that both approximations preserve the diversity order (i.e., the slope of the BER curve at high SNR) of the ML detector, a fact that is also visible in Fig. 4. Moreover, we note

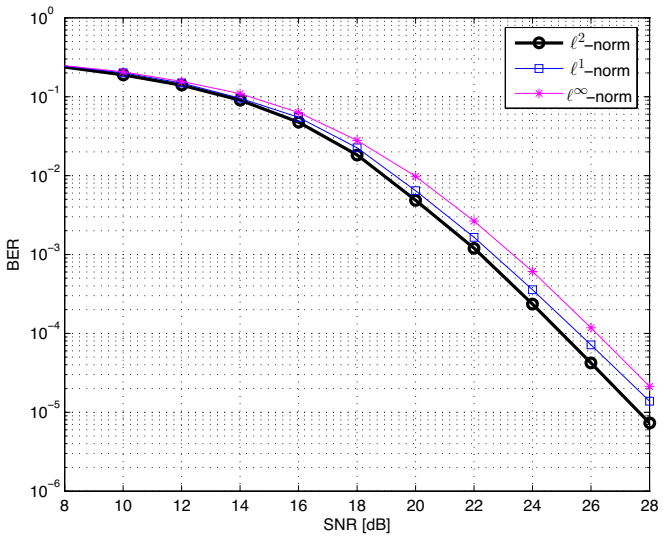

Fig. 4. BER performance of the ℓ^2-, ℓ^1-, and ℓ^∞-norm decoders for a 4×4 system with 16-QAM modulation. Using the computationally significantly more attractive ℓ^∞-norm results in a performance degradation of 1.4 dB at high SNR.

that the use of the ℓ^1-norm and the ℓ^∞-norm approximations infer only a 0.4 dB and a 1.4 dB high-SNR penalty, respectively.

V. First ASIC Implementation

The ASIC implementation described in this section follows the *one-node-per-cycle* architecture introduced in Section III. The vector $\hat{\mathbf{y}}$ is computed using the zero-forcing solution, i.e., $\hat{\mathbf{y}} = \mathbf{R}\mathbf{s}^{\mathrm{ZF}}$. Implementations based on the squared ℓ^2-norm and the ℓ^∞-norm approximation will be discussed, but our final implementation uses no approximations and thus achieves optimum BER performance. A block diagram of the corresponding circuit is shown in Fig. 5 and is described in more detail in the following.

A. MCU Sphere ALU: Exhaustive Search Enumeration

As already noted in Section III-B, the real-valued decomposition (10) leads to a significant throughput degradation. It is, therefore, mandatory to operate on the complex-valued constellation directly. The Sphere ALU of the first ASIC implementation (ASIC-I) exhaustively computes the PEDs of all children of a given node and checks compliance with the SC to determine the *admissible set*. The economic implementation of this scheme depends on the ability to efficiently compute the distance increments (6) for all $s_i \in \mathcal{O}$. It is obvious from (8) that b_{i+1} only depends on $\mathbf{s}^{(i+1)}$ and is thus common to all the children of the node under consideration at level $i+1$. So the distance increments can be obtained with minimum effort by precomputing b_{i+1} and simply evaluating (7) for all $s_i \in \mathcal{O}$.

In the following, we describe two alternative approaches to an efficient implementation of the PED computation. For the optimization of a VLSI implementation, the complexity of the different types of operations must be taken into account. Full complex-valued multiplications with two variable operands and squaring operations have high circuit complexity, while multiplications with constellation points have negligible circuit complexity (comparable to adders).

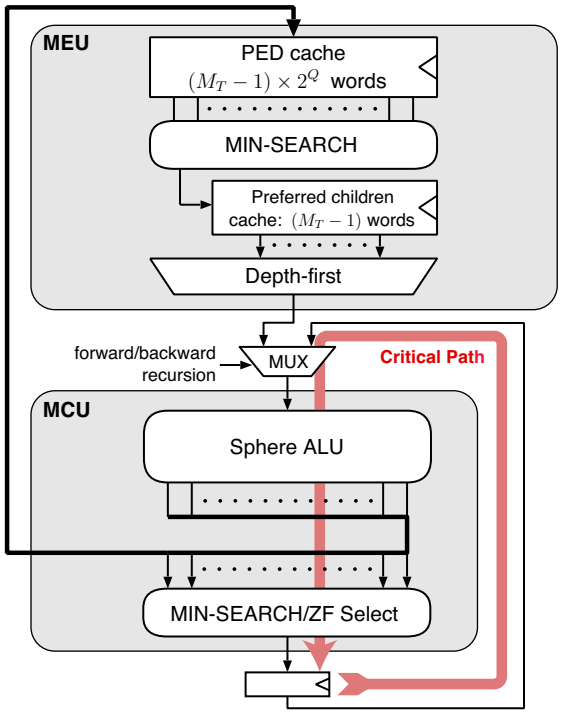

Fig. 5. Architecture of the *exhaustive search* SD ASIC. Strict SE enumeration would require a full minimum search (MIN-SEARCH) unit in the MCU, but our actual implementation always selects the child according to the sliced ZF solution to shorten the critical path.

1) Application of the ℓ^∞-Norm Approximation: The complexity of the PED computation is dominated by the squaring operations. Approximating the ℓ^2-norms according to (14) and (16) alleviates this problem at the cost of a slight performance degradation. The schematic of the corresponding ALU for 16-QAM modulation is shown in Fig. 6(a). With the ℓ^∞-norm approximation, costly full complex-valued multiplications only appear in the computation of b_{i+1}. The complexity of the remaining part of the circuit is mostly determined by the $2{\cdot}2^Q$ instantiations of the ℓ^2-norm approximations (14) and (16).

2) Squared ℓ^2-Norm Implementation: For the ℓ^2-norm, a reduction of the ALU area can be achieved by further resource sharing and a reduction of the number of costly operations as follows: We start by noting that (7) can be rewritten as

$$|e_i| = |b_{i+1}|^2 - 2\Re\{(R_{ii}b_{i+1})s_i^*\} + |R_{ii}|^2|s_i|^2. \qquad (17)$$

Only the computation of $|b_{i+1}|^2$, $R_{ii}b_{i+1}$, and $|R_{ii}|^2$ requires full complex-valued multiplications and squaring operations. However, these quantities need to be computed only once per parent node. Writing out the subsequent multiplications of $|R_{ii}|^2$ with $|s_i|^2$ and $R_{ii}b_{i+1}$ with s_i^*, it can be observed that the same products appear multiple times, since the components of s_i are restricted to only a few possible values. These products are computed only once using optimized constant-coefficient multipliers, and are then added appropriately in the SUM blocks to finally arrive at (7) for all $s_i \in \mathcal{O}$. The block diagram of the corresponding Sphere ALU for 16-QAM modulation is shown in Fig. 6(b). Clearly, only the inexpensive SUM operations have to be replicated to compute the PEDs of all 2^Q children. Without the optimization motivated by (17), the Sphere ALU for a squared ℓ^2-norm SD would resemble the architecture in

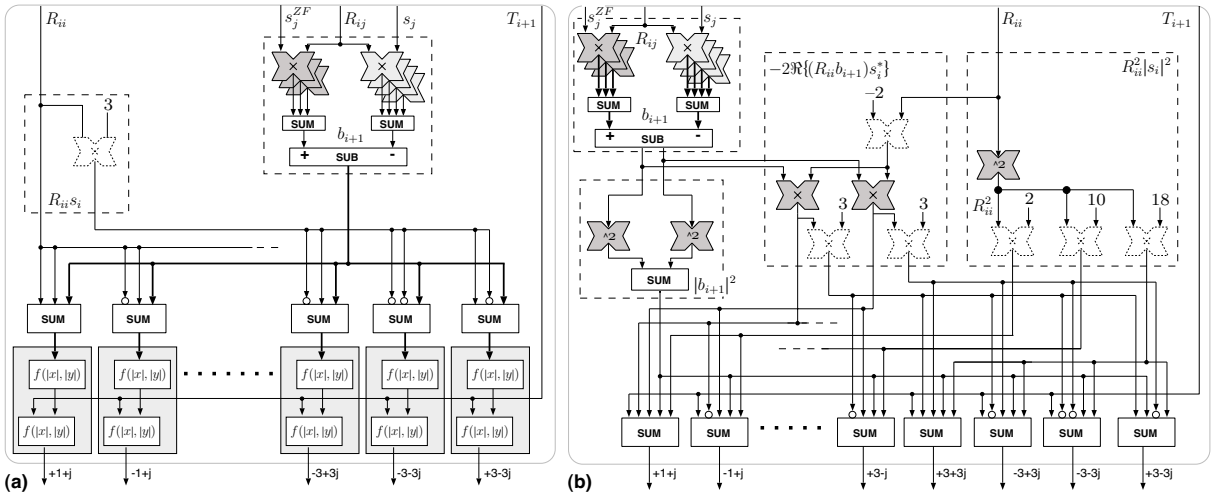

Fig. 6. Sphere ALUs for an *exhaustive search* SD ASIC for a 4 × 4 MIMO system with 16-QAM modulation. (a) ℓ^1- or ℓ^∞-norm approximation of (4)–(8). (b) Squared ℓ^2-norm implementation of (4)–(8) according to (17). The 16-QAM constellation is scaled such that $s_i \in \{\pm 1 \pm 1j, \pm 1 \pm 3j, \pm 3 \pm 1j, \pm 3 \pm 3j\}$ and $\hat{\mathbf{y}} = \mathbf{R}\mathbf{s}^{\mathrm{ZF}}$ is used. Costly operations are highlighted in dark gray, while less complex operations are shown in lighter shades.

Fig. 6(a). Each of the 2^Q norm computation units would then be highly complex, containing two full squaring operations.

B. MCU MIN-SEARCH: Critical Path Reduction with Modified SE Scheme

Strict SE ordering dictates that the preferred child to be visited should always be the one with the smallest PED among those that have not been visited yet. In the MCU, this requirement reduces to finding the minimum among all the possible T_i at the output of the Sphere ALU, as none of the children has been visited yet. Unfortunately, this exhaustive search for a minimum (or *MIN-SEARCH*) over 2^Q candidates (cf. Fig. 5) is a rather slow operation that lies on the critical path of the circuit. It either requires Q levels of slow compare/select logic or a large number of parallel comparators, which are extremely costly in terms of area. As the MIN-SEARCH is part of the forward tree traversal recursion, pipelining is not possible.

In order to resolve this bottleneck, we determine the first child to be examined *a priori* by choosing the sliced ZF solution, independently of its PED. This modification replaces the MIN-SEARCH in the MCU with a simple multiplexer (ZF Select) so that the critical path is shortened and a higher clock rate can be achieved. As opposed to a random choice for the first child node to be visited, the ZF solution still leads to a rapid shrinkage of the radius, so that the efficiency of the tree pruning process is almost unaffected. However, strict SE ordering is resumed for the remaining children that are visited later in the backward passes, for which the MIN-SEARCH is performed "in the background" by the MEU.

We note further that the modified algorithm not only compromises the SE enumeration, but also no longer adheres strictly to the *one-node-per-cycle* paradigm. An extra cycle is invested whenever the decoder reaches a leaf or when the a priori chosen child does not meet the SC. In both cases, the decoder pauses on the current node to await the decision on the next child to be visited, made by the MIN-SEARCH in the MEU in the next cycle. But, since only few of the visited nodes are leaves, and the probability of the chosen child not meeting the SC is rather small (especially for higher-order constellations), the reduction of the critical path outweighs the few additional cycles. Compared to strict SE ordering, the number of visited nodes increases by approximately 10% for moderate SNRs, while synthesis results indicate a critical path reduction by more than 20%; moreover, area is slightly reduced.

C. MEU Implementation

The MEU performs *exhaustive search* enumeration. During the forward pass, the Sphere ALU in the MCU has already computed the PEDs of all children of the nodes on the path between the root and the current node. The corresponding PEDs are stored in the *PED cache*, which is $M_T - 1$ entries deep and 2^Q words wide. An additional flag marks the nodes that have already been visited. The MIN-SEARCH unit in the MEU constantly monitors the cache line that contains the siblings of the node whose children are currently examined in the MCU. SE enumeration is performed by selecting the sibling with the smallest PED (that has not been visited yet and meets the SC) to enter the *preferred children* cache. When the MCU reaches a leaf or a dead end in the preceding cycle, it is provided with a new parent node from this preferred children cache. The selection follows the *depth-first* paradigm. Since the PED of the selected node is already available, the MCU can continue with the children of the selected node without delay.

D. Implementation

The actual ASIC implementation, whose chip micrograph is shown in Fig. 7, is based on the squared ℓ^2-norm Sphere ALU. The throughput of the ASIC, as a function of the SNR, is also depicted in Fig. 7. The initial radius is set to infinity, and the search is always continued until all leaves are pruned from the tree. The main characteristics of the ASIC realization can be found in Table I.

E. Discussion

The key to the efficient implementation described in this section is extensive resource sharing according to (7) and (17).

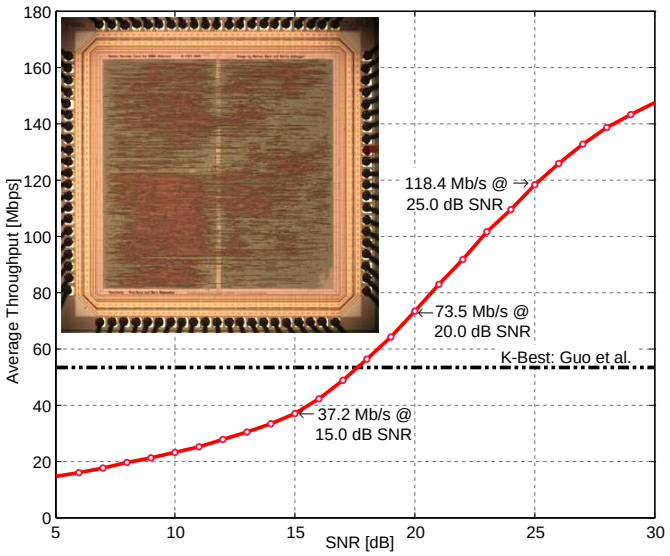

Fig. 7. Average throughput and chip micrograph of the *exhaustive search* SD ASIC-I. The reference results for the K-best decoder are taken from the conclusions section of [15].

Combining *exhaustive search* enumeration with the ℓ^1- or ℓ^∞-norm allows for resource sharing according to (7) only, since a decomposition similar to (17) is not possible for the ℓ^1- or ℓ^∞-norm. As a result, repeated instantiations of the norm approximation circuits are required as shown in Fig. 6(a). The circuit complexity advantage of ℓ^1- or ℓ^∞-norm decoding thus vanishes, and the chip area increases compared to the squared ℓ^2-norm (at least for higher-order constellations). Finally, the PED cache with 2^Q entries requires significant chip area and makes the design less scalable to higher-order constellations.

VI. Improved ASIC Implementation

The second ASIC implementation described in this paper also adopts the one-node-per-cycle architecture. However, it computes $\hat{\mathbf{y}}$ using $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$ and employs the ℓ^∞-norm approximation as well as a scheme for direct SE enumeration (described below) in systems with QAM modulation. The block diagram of the circuit is shown in Fig. 8.

A. Direct SE Enumeration for PSK-Like Constellations

In [1], Hochwald and ten Brink proposed a scheme that allows to compute boundaries of *admissible intervals* for complex-valued constellations having the constellation points arranged on concentric circles (e.g., PSK, 16-QAM). However, the original proposal requires the computation of trigonometric functions and other costly operations (cf. [1, eq. (25)]). In the following, we propose a slight modification of the ideas in [1], which results in a low complexity VLSI implementation.

1) Direct Enumeration for PSK Modulation: We shall first describe the basic idea for PSK modulation. In general, the *preferred child* $s_i^{(0)}$ in the forward pass (i.e., the starting point for the SE enumeration) is given by the constellation point minimizing the PED increment $|b_{i+1} - R_{ii}s_i|^2$. As all constellation points lie on a circle around the origin and R_{ii} can be chosen to

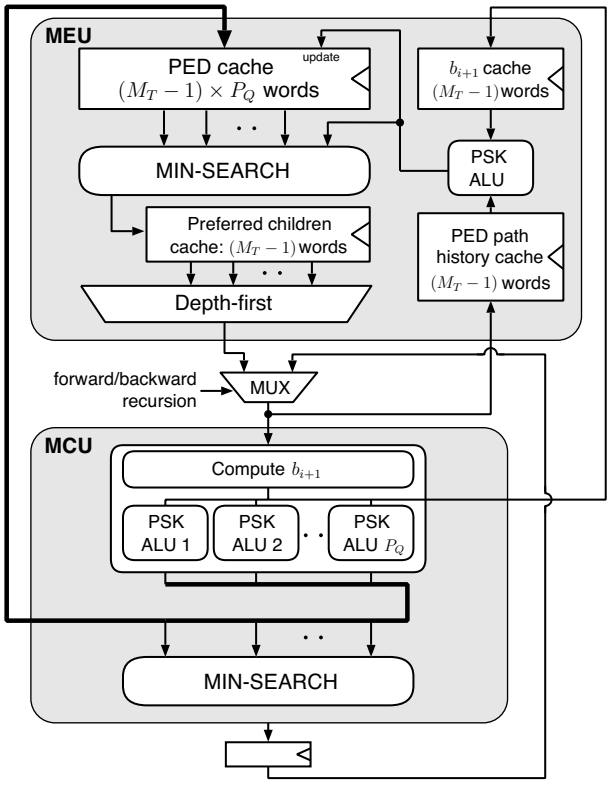

Fig. 8. Block diagram of an SD ASIC with direct QAM enumeration.

be positive real without loss of generality, one can easily show that the preferred child can also be obtained from

$$s_i^{(0)} = \arg\min_{s_i\in\mathcal{O}} \big|\mathrm{arc}(b_{i+1}) - \mathrm{arc}(s_i)\big| \tag{18}$$

where $\mathrm{arc}(\cdot)$ denotes the phase of a complex number. Hence, the starting point $s_i^{(0)}$ can be computed based on the phases of b_{i+1} and the s_i only. SE Enumeration for PSK modulation now amounts to proceeding from $s_i^{(0)}$ in a zig-zag fashion along the unit circle. The procedure is illustrated in Fig. 9(a) for 16-PSK modulation. The direction of the initial step can be found considering the phase of b_{i+1} and the two neighbors of $s_i^{(0)}$. Once the PED of a constellation point violates the SC, the admissible interval is exceeded and enumeration terminates. If already the PED of the closest constellation point exceeds r^2, the admissible interval is empty and a dead end is declared.

2) Direct Enumeration for QAM Modulation: A hybrid approach between an *exhaustive search* and *direct PSK enumeration* allows to extend the proposed scheme to QAM modulation. We start by grouping the constellation points into P_Q subsets, according to their distance from the origin. For QPSK, 16-QAM, and 64-QAM, $P_Q = 1$, 3, and 9 subsets, respectively, must be formed. The following *direct QAM enumeration procedure* can be used to construct a list of constellation points in SE ordering:

1) *Within each subset*, the *preferred child* is determined based on a minimization of $\big|\mathrm{arc}(b_{i+1}) - \mathrm{arc}(s_i)\big|$, and subsequently the corresponding PED is computed.
2) The PEDs of the P_Q preferred children are compared, and the point with the smallest PED *across the subsets* is chosen.

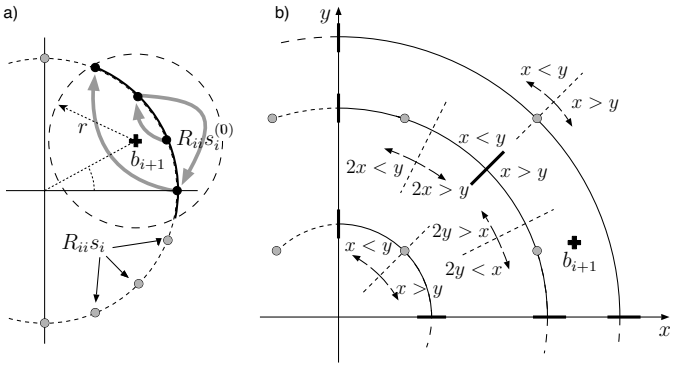

Fig. 9. (a) Direct PSK enumeration scheme (b) and its application to 16-QAM without trigonometric functions. The inequalities represent the decision boundaries for determining the starting point and the initial direction of the enumeration.

3) Before the next point in the SE ordering can be obtained, the constellation point selected in step 2) is replaced by the next candidate in the corresponding subset according to the *direct PSK enumeration*, and the corresponding PED is computed. The algorithm proceeds with step 2) until all subsets are empty.

We note that the initialization step 1) requires the computation of P_Q PEDs. In the subsequent iterations, only a single PED per pass has to be computed.

B. MCU Sphere ALU Implementation

PSK Enumeration: Recalling that the task of the MCU is to find the starting point of the enumeration, we can conclude that the MCU implements steps 1) and 2) of the *first pass* in the above described enumeration procedure. The MCU employs P_Q PSK ALUs. Each of them solves (18) for one PSK subset to find the closest constellation point and the initial enumeration direction, which can both be identified through the introduction of suitable decision boundaries instead of using trigonometric functions. These boundaries can be specified in terms of relations between the real and imaginary parts of b_{i+1}, which can be checked efficiently with very little hardware effort. An example for 16-QAM modulation is shown in Fig. 9(b), where the bold lines mark the decision boundaries for finding the closest point $s_i^{(0)}$, and the dashed lines are the boundaries that determine the initial direction of the direct PSK enumeration. Our implementation also exploits the symmetry of the constellation to reduce the problem to the first quadrant, thus requiring the examination of the absolute values of the real and imaginary parts of b_{i+1} only. This adjustment infers extra cost resulting from the need to map the solution in the first quadrant back into the actual quadrant. However, this extra cost is more than compensated for by the reduced number of decision boundaries.

Metric Computation: After the closest point in each subset has been determined, the associated PEDs are computed. As opposed to ASIC-I, where 2^Q candidate symbols are considered in parallel, (7) needs to be evaluated for only $P_Q \ll 2^Q$ candidate symbols; therefore, it is not worthwhile to pursue resource sharing.

ℓ^∞-Norm Approximation: The application of the ℓ^∞-norm approximation is straightforward, as it only changes the computation of the PEDs in the PSK ALUs. However, as opposed to the *exhaustive search* architecture in ASIC-I, fewer (only $2P_Q$) instantiations of the ℓ^2-norm approximation are needed and an overall area advantage is achieved by the ℓ^∞-norm approximation.

C. MCU MIN-SEARCH Implementation

The starting point of the enumeration is finally found as the minimum across the preferred children of the different PSK subsets. As opposed to the *exhaustive search*, used in the ASIC-I implementation, the number of candidates to be compared in the MIN-SEARCH is significantly reduced. Fortunately, this also reduces the delay of the MIN-SEARCH and its contribution to the critical path so that, unlike in ASIC-I, there is no need to deviate from strict SE ordering.

We conclude by noting that the combination of phase-based direct PSK enumeration and the ℓ^∞-norm approximation results in a "hybrid" overall "norm" that is neither ℓ^2nor ℓ^∞. Correspondingly, the exact solution and search time will in general deviate (slightly) from a strict *exhaustive search* based ℓ^∞-norm implementation.

D. MEU Implementation

The MEU executes steps 2) and 3) in the QAM enumeration procedure described above. For every subset, it keeps track of the preferred children of each node between the current node and the root of the tree. As opposed to the *exhaustive search* decoder, this only requires a PED cache with P_Q entries per line (level), as opposed to 2^Q entries. However, every time a child has been visited by a forward or backward iteration, the corresponding entry in the cache needs to be updated. Consequently, the MEU contains an additional PSK ALU, which is, however, much simpler than the PSK ALUs in the MCU, as no decision boundaries need to be checked. The next constellation point is simply obtained by direct PSK enumeration [cf. Fig. 9(a)]. Most of the complexity in evaluating (7) is in computing b_{i+1}. However, as this term has already been computed in the MCU, it is kept in a small cache in the MEU.

A *PED path history cache* is finally needed to store the PEDs of the parent nodes along the current path from the root, which are needed by the PSK ALU in order to compute (5). When all the children in the admissible interval of a PSK subset have been visited, the corresponding entry in the PED cache is marked as invalid. Exactly like in ASIC-I, a MIN-SEARCH constantly determines the preferred child across subsets and places it into the *preferred children* cache, from which the next node is chosen by the depth-first multiplexer in the case of a leaf or a dead end.

E. Chip Realization

This second ASIC reported in this paper, for a 4×4 system with 16-QAM modulation, is based on the ℓ^∞-norm approximation and the enumeration scheme described in Section VI-A. Fig. 10 shows the expected average throughput together with the layout of the chip. The throughput was computed based on the average number of cycles per received vector (obtained by computer simulations) and the maximum clock frequency (estimated with postlayout static timing analysis). The characteristics of the design can also be found in Table I.

TABLE I
COMPARISON OF ASIC IMPLEMENTATIONS FOR MIMO DETECTION

						This work	
Reference	[19]	[20]	[3]	[14]	[15]	ASIC-I [16]	ASIC-II [17]
Antennas	4×4						
Modulation	QPSK	QPSK	QPSK	16-QAM	16-QAM	16-QAM	16-QAM
Detector	V-BLAST	exh. search ML-APP	exh. search	K-best sphere	K-best sphere	depth-first sphere	depth-first sphere
BER performance	suboptimum	ML		close to ML		ML	close to ML
Technology	0.35 μm	0.18 μm	0.25 μm	0.35 μm	0.35 μm	0.25 μm	0.25 μm
Core Area [GE]	190K	140K	40K	52K +preproc.	91K +preproc.	117K +preproc.	50K +preproc.
Max. clock	80 MHz	122 MHz	100 MHz	100 MHz	100 MHz	51 MHz	71 MHz
Throughput	128 Mbps	28.8 Mbps	50 Mbps	10 Mbps	52 Mbps	73 Mbps @ SNR = 20 dB (360 mW)	169 Mbps @ SNR = 20 dB

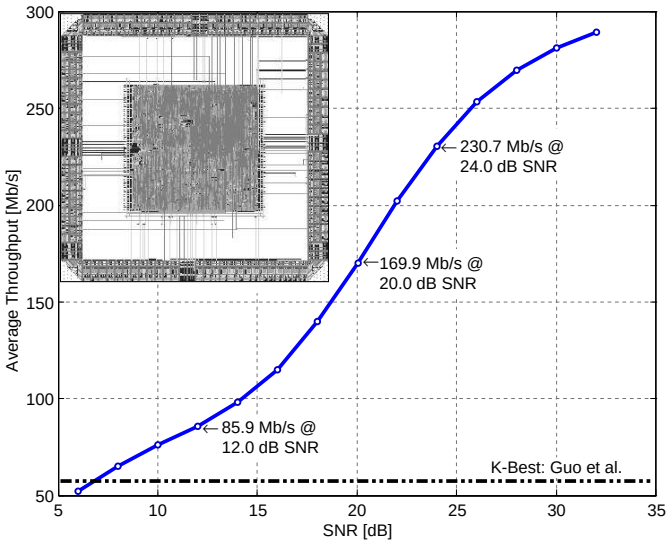

Fig. 10. Average throughput and layout of the improved SD ASIC-II with direct QAM enumeration. The reference results for the K-best decoder are taken from the conclusions section of [15].

F. Discussion

The implementation described in this section requires the parallel computation of a much smaller number of PEDs (compared to ASIC-I) and hence benefits significantly from the ℓ^∞-norm approximation, as resource sharing in the squared ℓ^2-norm case is less efficient. The length of the overall critical path is reduced by roughly 25% compared to the same architecture based on the squared ℓ^2-norm. Also, a significant reduction in the average number of visited nodes obtained through the use of the ℓ^∞-norm contributes significantly to the high throughput of the chip. The corresponding performance loss (due to the use of a suboptimal "norm") is a 1.4 dB SNR degradation (cf. Fig. 4). The complexity of the MCU and the memory (cache) requirements in the MEU scale only with P_Q. Since $P_Q \ll 2^Q$ for higher-order modulation, the architecture is particularly well suited for large constellations.

VII. COMPARISON

We shall next provide an overview and comparison of the most relevant reported ASIC implementations of MIMO detection algorithms, which are summarized in Table I.

A. Comparison of Depth-First Architectures

We start by comparing the two implementations of the *depth-first* SD presented in this paper. Both SD ASICs are based on the same *one-node-per-cycle* isomorphic architecture and operate directly on the complex-valued constellations. The implementations mainly differ in the preprocessing strategy and in the realization of the SE enumeration. The improved ASIC implementation (ASIC-II, described in Section VI) yields twice the throughput of the first one (ASIC-I, described in Section V) at half the chip area.

Throughput: The throughput gains of ASIC-II over ASIC-I can mostly be attributed to the higher clock rate (40% increase) and to the reduction of the average number of visited nodes due to the use of the ℓ^∞-norm (50% increase). The strict adherence to the *one-node-per-cycle* paradigm in the second architecture also contributes to the increased throughput. In particular at high SNR, the few additional cycles needed by ASIC-I make a significant difference (up to 25% in a 4×4 system). At low and medium SNRs the influence of the additional cycles is only marginal ($< 10\%$). Also, the strict SE enumeration in the second architecture leads to a slightly more rapid shrinkage of the sphere, which yields an additional marginal throughput increase ($< 10\%$).

Bit Error Rate: From a BER performance perspective, ASIC-I finds the ML solution (provided that the initial radius is set to infinity and search time is not constrained). The second ASIC suffers from a slight performance loss due to the use of the ℓ^∞-norm (cf. Fig. 4), but still achieves full diversity gain.

Scaling in Number of Antennas and Constellation Size: The circuit area of both architectures grows only slowly with the number of transmit antennas M_T. However, the throughput will drop with increasing M_T [4]. In terms of the impact of the constellation size, the *exhaustive search* strategy suffers significantly from the exponential growth of the number of constel-

lation points in Q, while the area corresponding to the direct QAM enumeration increases less rapidly with the constellation size.

B. Depth-First vs. K-Best

The choice of the depth-first tree traversal paradigm yields an architecture that is radically different from the K-best approach in [14] and [15]. Depth-first tree traversal is implemented in a sequential, nonpipelined isomorphic architecture, whereas the K-best algorithm is based on a parallel, heavily pipelined hardware structure with significant time sharing to reduce chip area. The main advantage of the K-best approach is guaranteed constant throughput. However, restriction to the K best candidates in general entails a slight BER performance loss [14], [15]. Interestingly, it turns out that the use of *radius reduction* increases the *average* throughput of depth-first tree traversal so much that it matches or exceeds the *constant* throughput of the K-best implementation in most cases (cf. Fig. 10). However, the instantaneous throughput achieved by depth-first traversal architectures may drop severely (down to 260 kbps for ASIC-II) under worst-case channel conditions. Imposing a constraint on the maximum number of steps can alleviate this problem. The impact of such a modification on BER performance has not been investigated systematically.

C. Comparison to Other MIMO Detectors

Apart from SD, implementations of *exhaustive-search ML* and a *V-BLAST* ASIC have been reported in the literature [3], [20], [19]. Exhaustive-search ML is clearly the architecture of choice for low rate systems ($R \leq 8\,\mathrm{bpcu}$). For higher rates, the computational complexity of exhaustive search ML decoding becomes prohibitive. The V-BLAST algorithm is mostly of interest for higher-order modulation, as the decoding complexity is almost independent of the constellation size. However, the BER performance of the algorithm is rather poor due to its inability to exploit the full diversity available in the channel. For a 4×4 MIMO system using 16-QAM modulation or higher, *sphere decoding* should be the method of choice, as its implementation complexity is low and the BER performance is very close to ML.

We note that the ASIC implementations of SD in Table I do not include the channel matrix preprocessing such as matrix inversion, QR decomposition, or Cholesky factorization, the cost of which needs to be added to the complexity figures in Table I. However, the complexity of preprocessing is only critical in wideband MIMO-OFDM systems, where it needs to be performed on a tone-by-tone basis. A solution to this problem has recently been presented in [21], [22].

VIII. Conclusion

Sphere decoding can be implemented in VLSI with comparatively low complexity and high throughput. Despite the nonconstant throughput, depth-first tree traversal with radius reduction appears to be a favorable implementation strategy. In practice, it will be applied in combination with early-termination techniques, which will help to guarantee a minimum throughput and reduce power consumption in the receiver. Optimum search termination strategies are yet to be investigated. Future research will also need to address the issue of obtaining *a posteriori probabilities* from sphere decoding for soft input decoding and iterative processing. Initial architectures for this problem have recently been presented in [5].

References

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

[2] A. Paulraj, R. Nabar, and D. Gore, *Introduction to Space-Time Wireless Communications*. Cambridge Univ. Press, 2003.

[3] A. Burg, N. Felber, and W. Fichtner, "A 50 Mbps 4×4 maximum likelihood decoder for multiple-input multiple-output systems with QPSK modulation," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, vol. 1, 2003, pp. 332–335.

[4] A. Burg and D. Garrett, "VLSI implementation of MIMO detection," in *Space-Time Wireless Systems: From Array Processing to MIMO Communications*, H. Bölcskei, D. Gesbert, C. Papadias, and A. J. van der Veen, Eds. Cambridge, UK: Cambridge Univ. Press, 2005, ch. 27, in press.

[5] D. Garrett, L. Davis, S. ten Brink, B. Hochwald, and G. Knagge, "Silicon complexity for maximum likelihood MIMO detection using spherical decoding," *IEEE J. Solid-State Circuits*, vol. 39, pp. 1544–1552, 2004.

[6] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications," *SIGSAM Bull.*, vol. 15, no. 1, pp. 37–44, Feb. 1981.

[7] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[8] E. Viterbo and J. Boutros, "A universal lattice decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.

[9] B. Hassibi and H. Vikalo, "On sphere decoding algorithm. I. Expected complexity," *IEEE Trans. Signal Processing*, submitted.

[10] J. Jalden and B. Ottersten, "An exponential lower bound on the expected complexity of sphere decoding," in *Proc. IEEE ICASSP*, vol. 4, May 2004, pp. 393–396.

[11] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

[12] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, no. 2, pp. 181–191, Sept. 1994.

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

[14] K. Wong, C. Tsui, R.-K. Cheng, and W. Mow, "A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels," in *Proc. IEEE ISCAS'02*, vol. 3, 2002, pp. 273–276.

[15] Z. Guo and P. Nilsson, "A VLSI architecture of the Schnorr-Euchner decoder for MIMO systems," in *Proc. IEEE CAS Symposium on Emerging Technologies*, June 2004, pp. 65–68.

[16] A. Burg, M. Wenk, M. Zellweger, M. Wegmueller, N. Felber, and W. Fichtner, "VLSI implementation of the sphere decoding algorithm," in *Proc. ESSCIRC-2004*, Leuven, Belgium, Sept. 2004, pp. 303–306.

[17] A. Burg, M. Borgmann, C. Simon, M. Wenk, M. Zellweger, and W. Fichtner, "Performance tradeoffs in the VLSI implementation of the sphere decoding algorithm," in *Proc. IEE 3G Mobile Communication Conf.*, London, U.K., Oct. 2004, pp. 93–97.

[18] H. David and H. Nagaraja, *Order statistics*. New York: Wiley, 2003.

[19] Z. Guo and P. Nilsson, "A VLSI implementation of MIMO detection for future wireless communications," in *Proc. IEEE PIMRC'03*, vol. 3, 2003, pp. 2852–2856.

[20] D. Garrett, G. Woodward, L. Davis, G. Knagge, and C. Nicol, "A 28.8 Mb/s 4×4 MIMO 3G high-speed downlink packet access receiver with normalized least mean square equalization," in *IEEE ISSCC Dig. Tech. Papers*, vol. 1, Feb. 2004, p. 420.

[21] M. Borgmann and H. Bölcskei, "Efficient matrix inversion for linear MIMO-OFDM receivers," in *Proc. 38th IEEE Asilomar Conf. Signals Syst. Comput.*, Pacific Grove, CA, Nov. 2004.

[22] D. Cescato, M. Borgmann, H. Bölcskei, J. Hansen, and A. Burg, "Interpolation-based QR decomposition in MIMO-OFDM systems," in *Proc. IEEE SPAWC'05*, New York, NY, June 2005.

Andreas Burg (S'97–M'05) was born in Germany in September 1975. He received the Diploma degree from the Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland, in 2000, where he is currently working toward the Dr. sc. techn. degree.

He has been a research assistant at the Integrated Systems Laboratory (IIS) of ETH Zurich since 2000. During his doctoral studies, he also worked at the Bell-Labs Wireless Research Laboratory, Holmdel, NJ, for one year. His research interests include the design of VLSI circuits and systems and digital signal processing for wireless communications.

Mr. Burg was the recipient of the "Willi Studer Award" and the ETH Medal for his diploma and his diploma thesis, respectively.

Moritz Borgmann (S'99) was born in Hamburg, Germany. After undergraduate studies at the Technical University of Munich, Munich, Germany, he received the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 2000. He is currently working toward the Dr. sc. techn. degree at the Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland.

He was a visiting researcher with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, in 2001. His research interests include coding and modulation as well as efficient receiver designs for wideband wireless MIMO communications and corresponding performance limits.

Mr. Borgmann was the recipient of a permanent scholarship from the German National Academic Foundation.

Markus Wenk was born in Switzerland in 1980. He is currently working toward the M. S. degree at the Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland, specializing in the field of microelectronics and communications.

Mr. Wenk was the recipient of the "Prix du Jeune Entrepreneur 2004" of the "Section Suisse des Conseillers du Commerce Ext érieur de la France" for his work on sphere decoding at the Integrated Systems Laboratory, ETH Zurich.

Martin Zellweger was born in Switzerland in 1980. He is currently working toward the M. S. degree at the Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland.

Mr. Zellweger was the recipient of the "Prix du Jeune Entrepreneur 2004" of the "Section Suisse des Conseillers du Commerce Ext érieur de la France" for his work on sphere decoding at the Integrated Systems Laboratory, ETH Zurich.

Wolfgang Fichtner (F'90) received the Dipl. Ing. degree in physics and the Ph.D. degree in electrical engineering from the Technical University of Vienna, Austria, in 1974 and 1978, respectively.

From 1975 to 1978, he was an Assistant Professor with the Department of Electrical Engineering, Technical University of Vienna. From 1979 through 1985, he was with AT&T Bell Laboratories, Murray Hill, NJ. Since 1985 he has been Professor and Head of the Integrated Systems Laboratory, Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland. In October 1999, he became Chairman of the Information Technology and Electrical Engineering Department, ETH Zurich. In 1993, he founded ISE Integrated Systems Engineering AG, a company in the field of Technology CAD, which was acquired by Synopsys, Inc. in 2004. His research activities cover physics-based simulation of semiconductor devices and technologies in microelectronics and optoelectronics, physical characterization and electronic measurement in deep-submicrometer and nanotechnologies, as well as design and test of digital integrated circuits. He is the author of three books and the author and coauthor of 360 reviewed journal and conference papers.

Dr. Fichtner is a member of the Swiss National Academy of Engineering. In 2000, he was the recipient of the IEEE Andrew S. Grove Award for his contributions to Technology CAD. Since 2001, he has been a corresponding member of the Austrian Academy of Sciences.

Helmut Bölcskei (M'98–SM'02) was born in Austria on May 29, 1970. He received the Dr. techn. degree in electrical engineering from Vienna University of Technology, Vienna, Austria, in 1997.

In 1998 he was with Vienna University of Technology. From 1999 to 2001, he was a Postdoctoral Researcher with the Information Systems Laboratory, Department of Electrical Engineering, Stanford University, Stanford, CA. During that period he was also a consultant for Iospan Wireless, Inc., San Jose, CA. From 2001 to 2002, he was an Assistant Professor of Electrical Engineering with the University of Illinois at Urbana-Champaign. Since February 2002, he has been an Assistant Professor of communication theory at the Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland. He was a Visiting Researcher with Philips Research Laboratories, Eindhoven, The Netherlands, ENST Paris, France, and the Heinrich Hertz Institute, Berlin, Germany. His research interests include communication and information theory with special emphasis on wireless communications and signal processing.

Dr. Bölcskei serves as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the *EURASIP Journal on Applied Signal Processing*, and is on the editorial board of *Foundations and Trends in Networking*. He was the recipient of the 2001 IEEE Signal Processing Society Young Author Best Paper Award and was an Erwin Schrödinger Fellow (1999–2001) of the Austrian National Science Foundation (FWF).