Performance Tradeoffs in the VLSI Implementation of the Sphere Decoding Algorithm

A. Burg*, M. Borgmann†, C. Simon†, M. Wenk*, M. Zellweger*, and W. Fichtner*
*Integrated Systems Laboratory
Swiss Federal Institute of Technology (ETHZ)
Zurich, Switzerland
Email: apburg@iis.ee.ethz.ch

†Communication Technology Laboratory
Swiss Federal Institute of Technology (ETHZ)
Zurich, Switzerland
Email: moriborg@nari.ee.ethz.ch

***Abstract*— Sphere decoding (SD) allows to solve high-dimensional MIMO maximum likelihood detection problems with significantly lower complexity than other methods. The SD algorithm has, however, mostly only been analyzed with DSP implementations in mind. We show that VLSI implementations call for new performance metrics, analyze the resulting implementation tradeoffs for the decoding of complex signal constellations, and we develop design guidelines and a generic architecture. When using the ℓ^∞-norm for the sphere constraint instead of the ℓ^2-norm, significant reductions in circuit complexity and improvements in tree pruning efficiency are possible at a minimum performance penalty. As a proof of concept, a high performance ASIC implementation is presented.**

I. Introduction

Many problems in mobile communications can be described with a simple linear multiple-input multiple-output (MIMO) model. Examples include multiantenna systems or multiuser detection in CDMA. The corresponding complex-valued input-output relation is

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

where $\mathbf{H}$ is an $N \times M$ effective channel matrix and $\mathbf{y}$ is the N-dimensional received vector, disturbed by the complex additive white Gaussian noise vector $\mathbf{n}$. The symbols $\mathbf{s} \in \mathcal{O}^M$ are composed of values independently chosen from a complex constellation $\mathcal{O}$. Real constellations can be considered as a special case. When $\mathbf{H}$ is known at the receiver, the maximum likelihood (ML) detector is given by

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathcal{O}^M} \|\mathbf{H}\mathbf{s} - \mathbf{y}\|^2. \tag{2}$$

In fading MIMO channels, ML detection exploits Nth order diversity, which is not achieved by linear and successive cancellation receivers. Hence, ML detection is attractive in the high SNR regime. Unfortunately, the complexity of an exhaustive search implementation of (2) is exponential in the transmission rate. For the case where $\mathcal{O}^M$ is a (real) integer lattice $\mathcal{L}^M$, sphere decoding (SD) has been proposed by Pohst [1] as an alternative approach, which has recently been introduced into communications. The algorithm achieves ML performance with an expected complexity that is only *polynomial* in the rate [2]. Numerous optimizations have been proposed to reduce the implementation complexity of the original SD algorithm on general purpose processors and digital signal processors (DSPs) [3]. However, the VLSI implementation of the algorithm has only received limited attention so far.

Contributions: In this paper, we describe implementation tradeoffs for high-throughput sphere decoding with complex modulation schemes in VLSI and introduce an efficient architecture. We also present slightly suboptimal numerical simplifications that significantly reduce the circuit complexity and increase the efficiency of the algorithm, while preserving diversity order in fading MIMO channels.

Outline: In Section II, we review the state of the art in sphere decoding and emphasize key concepts. Section III introduces an efficient high-level VLSI architecture for high-throughput SD implementations. In Section IV, implementation options for the realization of a complex SD are considered and compared. In Section V, we propose the *square-root sphere criterion* and its suboptimal variations that lead to significantly reduced circuit complexity and higher throughput. Section VI finally describes the implementation of a high performance 4×4 16-QAM sphere decoding chip.

Notation: $\|\mathbf{a}\|_p$ denotes the ℓ^p-norm of the vector $\mathbf{a}$. When the subscript is omitted, we tacitly mean the ℓ^2-norm (the Euclidean norm). $\mathcal{E}\{\cdot\}$ stands for the expectation operator.

II. State of the Art in Sphere Decoding

Under the term sphere decoding we subsume the original SD algorithm [1] and all the variations and extensions proposed later [4]–[6]. The algorithm consists of four key concepts that need to be clearly differentiated:

A. Sphere Constraint

The main idea is to reduce the number of points that need to be considered in the search for the ML solution. The list of candidates is constrained to only those points $\mathbf{H}\mathbf{s}$ that lie inside a sphere with a given radius C around the received point $\mathbf{y}$:

$$d(\mathbf{s}) \le C^2 \quad \text{with} \quad d(\mathbf{s}) \triangleq \|\mathbf{H}\mathbf{s} - \mathbf{y}\|^2. \tag{3}$$

We refer to equation (3) as the *sphere constraint* (SC).

B. Tree Pruning

Efficient checking of the SC only becomes feasible after transforming (2) into an equivalent problem with triangular channel matrix. Assuming that $N \ge M$, this goal can, for example, be achieved using the QR decomposition of the channel $\mathbf{H} = \mathbf{Q}\mathbf{R}$ with the $M \times M$ upper triangular matrix $\mathbf{R}$ and the $N \times M$ unitary matrix $\mathbf{Q}$. In this case, the SC (3) becomes

$$\hat{d}(\mathbf{s}) \le \hat{C}^2 \quad \text{with} \quad \hat{d}(\mathbf{s}) \triangleq \|\mathbf{R}\mathbf{s} - \hat{\mathbf{y}}\|^2$$

where the M-dimensional $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$, and where in the case $N > M$ the radius C needs to be adjusted to yield the modified radius $\hat{C}$. As a result of the triangular structure of $\mathbf{R}$, the distance $\hat{d}(\mathbf{s})$ can now be computed recursively as follows:

$$T_i(\mathbf{s}) \;\leftarrow\; T_{i+1}(\mathbf{s}) + \left(\hat{y}_i - \sum_{j=i+1}^{M} R_{ij}s_j - R_{ii}s_i\right)^2 \qquad (4)$$

with $T_{M+1}(\mathbf{s}) \leftarrow 0$ for all $\mathbf{s} \in \mathcal{O}^M$. Finally, $\hat{d}(\mathbf{s}) = T_1(\mathbf{s})$. We term $T_i(\mathbf{s})$ the *partial Euclidean distance* (PED) of a symbol $\mathbf{s}$ at level i. All possible symbols $\mathbf{s} \in \mathcal{O}^M$ can now be considered by tree traversal with the root at $i = M + 1$. Since the PED increases monotonically from level to level, a branch together with all its children can be pruned whenever its PED exceeds $\hat{C}^2$. The remaining branches belong to an *admissible set* of constellation points that need to be followed further. Ideally this set should constitute only a small subset of the constellation $\mathcal{O}$. The goal of the SD algorithm is to prune large parts of the tree, such that the complexity of the search for the ML solution is greatly reduced.

One of the main issues with the original Pohst algorithm is the choice of the sphere radius $\hat{C}$. If chosen too small, no solution is found — however, too many nodes need to be considered for a radius chosen too large. If we set $\hat{C}^2 = \hat{d}(\mathbf{s})$ whenever an admissible point $\mathbf{s}$ is found [6], the sphere radius decreases throughout the algorithm, and as a result fewer points need to be considered. This *radius updating* can be performed in a smart fashion, such that a restart of the algorithm with each radius update can be avoided [3].

C. Admissible Intervals

When the underlying constellation $\mathcal{O}$ is *real*, e.g., if it constitutes a real lattice, we can easily verify that at a given level i, any constellation point s_i that lies *between* two admissible points is also admissible. As a consequence, the admissible set is actually an *interval*. Checking whether s_i is in the admissible set therefore only amounts to comparing s_i to the boundaries of that interval.

The latter point often leads to the impression that sphere decoding concepts are only applicable to real lattices. However, it is crucial to realize that only the first two ideas described above are actually prerequisites for the tremendous complexity savings of the SD algorithm over a brute force search. In fact, sphere decoding is *applicable to arbitrary sets of constellation points*, in particular to *complex lattices* [7], [8]. In this case, the PED computation becomes

$$T_i(\mathbf{s}) \;\leftarrow\; T_{i+1}(\mathbf{s}) + \left|\hat{y}_i - \sum_{j=i}^{M} R_{ij}s_j\right|^2 \qquad (5)$$

but membership in the admissible set cannot simply be determined using the bounds of an admissible interval. We shall explore solutions to this problem in Section IV.

D. Schnorr-Euchner Enumeration

Without radius updating, the order in which nodes are visited is irrelevant for the pruning of the tree. However, radius updating leads to the greatest complexity reduction if symbols with smaller distance are visited first. Also, in order to find admissible symbols as fast as possible, depth-first traversal of the tree is mandatory. With the *Schnorr-Euchner* (SE) enumeration [5], on each level nodes with the smallest PED are followed first, leading to a more rapid shrinkage of the sphere radius. Hence the tree is pruned more efficiently. As an additional advantage of the SE enumeration, the initial sphere radius can be set to infinity. In that case, the first admissible point found is always the so called *Babai point* or zero-forcing decision-feedback point.

Summarizing, combined radius updating with SE enumeration is highly recommended, whenever applicable. We shall investigate how to perform the enumeration in practical implementations in Section IV.

III. Considerations for VLSI Implementation

Dedicated VLSI architectures differ from implementations on DSPs through their potential for massively parallel processing and the availability of customized operations and operation sequences that can be executed in a single cycle. The potential of an algorithm to exploit these properties is crucial to guarantee an efficient high-throughput implementation.

A. General Architecture

The VLSI architecture of a high-throughput SD application specific integrated circuit (ASIC) should be designed to ensure that the decoder examines the branches of a *new node in each cycle* and that *no node in the tree is ever visited twice*. This paradigm guarantees maximum throughput efficiency. It is achieved by partitioning the decoder into two parallel units:

1) The *metric computation unit (MCU)* starts from $T_{i+1}(\mathbf{s})$ (i.e., the metric of the branch that leads to the current node) and finds the starting point for the SE enumeration along with the PED $T_i(\mathbf{s})$ of the corresponding branch. When the bottom of the tree is reached, the MCU stores the symbol corresponding to the current path and updates the radius. In this case, or if the admissible set is empty, a new node branching off the current path further up in the tree is visited next. If no more valid branches are found, the decoder stops.
2) The *metric enumeration unit (MEU)* operates solely on nodes that have already been visited by the MCU. It carries out the SE enumeration to find the branch with the smallest PED among those that have not been visited yet. It keeps a list of these admissible children for all nodes on the current path. When the MCU reaches a leaf or a dead end, the MEU can decide immediately where the search should be continued in the next cycle. Tree traversal is performed depth-first.

The described procedure is exemplified in Fig. 1 for $M = 3$ and BPSK modulation. On the left, the branches that are examined by the MCU on the way down are marked. The MEU follows along the same nodes with one cycle delay and computes the PED of the branch that would follow next in the SE enumeration. After cycle 3, the MCU has found the first complete candidate symbol. In the meantime, the MEU has determined that only the branch to node B still meets the updated SC and has already computed its PED. Therefore, the MCU can proceed immediately to check the branch leading to

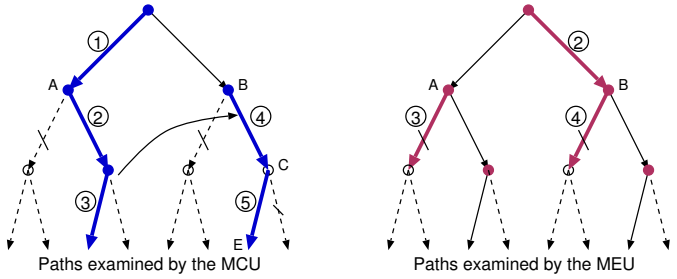

Fig. 1. Tree traversal example with $M = 3$ and BPSK. The order and cycle number in which the MCU and MEU examine the branches is shown

node C in cycle 4 and finally to the leaf E in cycle 5. Note that no cycles are wasted to slowly climb up the tree step-by-step.

B. Complexity and Performance Metric

For the architecture described above, we introduce suitable metrics that characterize throughput and implementation complexity:

1) Performance: The recursive tree-pruning scheme applied in the SD algorithm suggests that the described processing of a single node in one step (i.e., in one cycle) is the maximum degree of parallelism that can be achieved while fully preserving its complexity advantage over an exhaustive search. Processing multiple subsequent nodes in parallel would instead gradually lead back to the implementation of an exhaustive search decoder [9]. Although this approach opens up further tradeoffs between chip area and throughput, we do not pursue it here. In a *one node per cycle* architecture, the overall performance of the decoder is governed by two criteria:

1) the *average number of visited nodes* $\mathcal{E}\{K\}$ before the ML solution is found, which determines the number of cycles per symbol
2) the *cycle time* t_{CLK}, which is given by the critical path through the longest chain of consecutive operations in one cycle.

The first criterion characterizes the efficiency of the tree pruning and can be considered purely on an algorithmic level. The second criterion is concerned with the efficiency of the hardware implementation. The overall throughput is given by $\Phi = M \log_2|\mathcal{O}|/(\mathcal{E}\{K\}\, t_{\mathrm{CLK}})$. Note that, as opposed to optimizations that target DSPs, the pure number of operations (FLOPs) is of little significance.

2) Circuit Complexity: The circuit complexity is measured by the area required for the integration of all processing elements and the memory. Just as the delay it varies significantly depending on the type of operation and on the associated word-width. While tradeoffs between area and delay are possible, we strive for maximum throughput in this paper.

IV. Schnorr-Euchner Enumeration in Arbitrary Sets and Complex Lattices

The most critical part in the design of a SD is finding the admissible set, and the implementation of the SE enumeration. In the complex case, there is no admissible interval. Three approaches to identify the admissible set have been proposed:

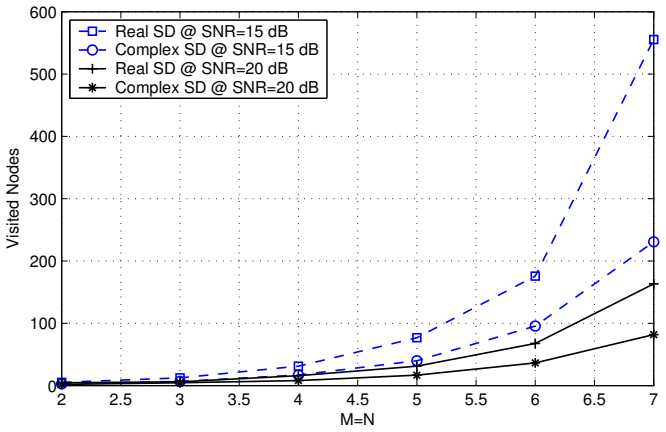

Fig. 2. Number of visited nodes vs. number of complex dimensions using real lattice decomposition and directly on the complex 16-QAM constellations

1) Real Decomposition: In the case of rectangular (complex) Q-QAM constellations, a common procedure is to decompose the M-dimensional complex problem into an equivalent real-valued problem according to

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

The result is a real lattice $\mathcal{L}^{2M}$ of dimension $2M$ with $\sqrt{Q}$ constellation points per dimension. The decoder can now explicitly compute the center point of an admissible interval, from which it proceeds with the enumeration in a zig-zag fashion [5] until a constellation point is outside the admissible interval. This condition can be checked based on explicit computation of the boundaries or by simply checking the SC [3].

However, traversing the resulting, deeper tree with fewer branches per node reduces the potential for parallel processing compared to a more shallow tree with more branches per node. Since the number of visited nodes is the main performance metric in VLSI implementations, the decomposition of the complex into a real lattice entails a significant performance degradation (cf. Fig. 2). Moreover, on a circuit level, no advantage can be taken from the orthogonality of the real and complex part and the symmetries in the complex constellations. Additionally, the computation of the center point of the admissible interval for SE enumeration is slow. Consequently, decomposition into a real lattice is not advisable for a high-throughput VLSI implementation.

2) Exhaustive Search: To directly determine the admissible set, an exhaustive search over the full constellation $\mathcal{O}$ can be performed. Explicit sorting of the PEDs is subsequently used to realize the SE enumeration. As opposed to the first solution, this method allows for arbitrary complex constellations and does not increase the depth of the search tree.

At first sight, the full search appears to have a very high implementation complexity as the PEDs need to be computed for all candidate constellations. However, (5) can easily be decomposed into

$$T_i(\mathbf{s}) = T_{i+1}(\mathbf{s}) + |b_i(\mathbf{s})|^2 - 2\Re\{b_i^*(\mathbf{s})R_{ii}s_i\} + |R_{ii}|^2 s_i^* s_i$$

TABLE I
APPROXIMATIONS FOR $\sqrt{x^2+y^2}$

ℓ^1-norm	$\lvert x\rvert + \lvert y\rvert$
ℓ^∞-norm	$\max(\lvert x\rvert, \lvert y\rvert)$
hybrid 1	$\frac{3}{8}(\lvert x\rvert + \lvert y\rvert) + \frac{5}{8}\max(\lvert x\rvert, \lvert y\rvert)$
hybrid 2	$\max\left(\max(\lvert x\rvert, \lvert y\rvert), \frac{7}{8}\max(\lvert x\rvert, \lvert y\rvert) + \frac{1}{2}\min(\lvert x\rvert, \lvert y\rvert)\right)$

with $b_i(\mathbf{s}) = \hat{y}_i - \sum_{j=i+1}^{M} R_{ij}s_j$. As a result, most of the costly operations can be shared among all candidate points.

The drawback of this approach is that all PEDs on the path down need to be stored to perform the enumeration. Additionally, the decision for the smallest metric in the MCU involves a search over all constellations, which is slow and leads to a long cycle time.

3) Hybrid Schemes: Depending on the constellation, hybrid approaches between exhaustive search and ordered enumeration may also be possible, as proposed in [7]: Starting from PSK modulation, admissible intervals are defined based on the phase of the constellation points. Subsequently, QAM modulation is described as the union of PSK subsets, within which enumeration is straightforward. SE ordering across subsets is achieved through explicit sorting of the PEDs.

The difficulty in the application of this approach is the computation of the starting points for the PSK enumerations. However, with a specific modulation scheme in mind it can be performed by simple direct comparisons between the real and imaginary parts, and no angles need to be computed, as opposed to [7]. As a result, this approach generally yields the lowest circuit complexity for QAM modulation and requires only a few PEDs to be stored and compared.

V. THE SQUARE ROOT SPHERE CRITERION

The computation of the PED on each level can be decomposed into the computation of an error term $e_i(\mathbf{s})$ and the recursive update of the PED:

$$e_i(\mathbf{s}) \leftarrow \hat{y}_i - \sum_{j=i}^{M} R_{ij}s_j \tag{6}$$

$$T_i(\mathbf{s}) \leftarrow T_{i+1}(\mathbf{s}) + |e_i(\mathbf{s})|^2. \tag{7}$$

The squaring operation in (7) consumes a large chip area and is slow, limiting the performance of the PED computation. Moreover, controlling the dynamic range of the numbers becomes a more severe issue after squaring. In order to eliminate the square, we have proposed to operate on the *square root* $\widetilde{T}_i(\mathbf{s}) = \sqrt{T_i(\mathbf{s})}$ of the PEDs [8], which yields an equivalent detector. By taking the square root of (7), we obtain

$$\widetilde{T}_i(\mathbf{s}) \leftarrow \sqrt{\widetilde{T}_{i+1}(\mathbf{s})^2 + |e_i(\mathbf{s})|^2}. \tag{8}$$

For this type of expression, numerous approximations of the form $\sqrt{x^2+y^2} \approx f(|x|,|y|)$ are available. Four approximations with efficient VLSI implementations are given in Table I.

They all lead to new (suboptimal) detectors, which can be interpreted as minimizing another norm for the *triangularized* problem (instead of the Euclidean or ℓ^2-norm, which corresponds to the ML solution). In particular, the first approximation leads to the minimization of the ℓ^1-norm of $\mathbf{Rs}-\hat{\mathbf{y}}$ [10]:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathcal{O}^M} \sum_{i=1}^{M} |e_i(\mathbf{s})| = \arg\min_{\mathbf{s}\in\mathcal{O}^M} \|\mathbf{Rs}-\hat{\mathbf{y}}\|_1.$$

The second approximation leads to a minimax optimization, or the minimization of the ℓ^∞-norm:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s}\in\mathcal{O}^M} \max_{i\in\{1,2,\ldots,M\}} |e_i(\mathbf{s})| = \arg\min_{\mathbf{s}\in\mathcal{O}^M} \|\mathbf{Rs}-\hat{\mathbf{y}}\|_\infty.$$

The remaining approximations correspond to hybrid norms. Note that only the ℓ^2-norm exhibits the property that the minimization on the original and the triangularized problem yield the same solution.

In the complex case, the same approximations can be used to compute the absolute value of the complex error term $|e_i(\mathbf{s})|$ from its real and imaginary part. The reduction in circuit complexity is significant. Still, the impact on bit error rate (BER) is small. We use the exact absolute value computation in our simulations.

1) Impact on Complexity: Since

$$\|\mathbf{Rs}-\hat{\mathbf{y}}\|_\infty \le \|\mathbf{Rs}-\hat{\mathbf{y}}\|_2 \le \|\mathbf{Rs}-\hat{\mathbf{y}}\|_1$$

the accumulated distance at the bottom of the tree is smallest for the ℓ^∞-norm. After the radius update, it is therefore more probable that the PED for a certain node further up in the tree is larger than the new radius, as compared to the ℓ^2-norm case. Therefore, more branches in the tree will be pruned. The situation is just the opposite for the ℓ^1-norm, therefore tree pruning becomes less effective. The impact on complexity is clearly visible in Fig. 3. It is remarkable that employing the ℓ^∞-norm almost halves the complexity at low SNR.

2) Impact on Bit Error Rate: The approximations of the ℓ^2-norm warp the sphere that is searched when looking for the ML solution. However, we can show analytically that full diversity is preserved regardless of the particular norm employed. This fact can be verified in Fig. 4. Among the approximations considered, the ℓ^∞-norm detector shows the

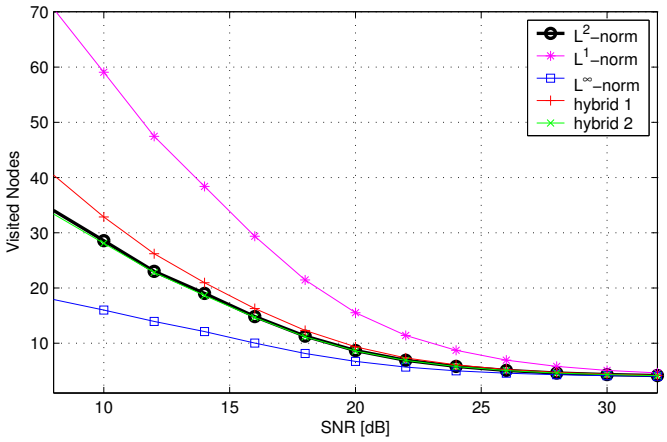

Fig. 3. Number of visited nodes vs. SNR for the square root sphere detector with different approximations. Modulation: 16-QAM, N=M=4

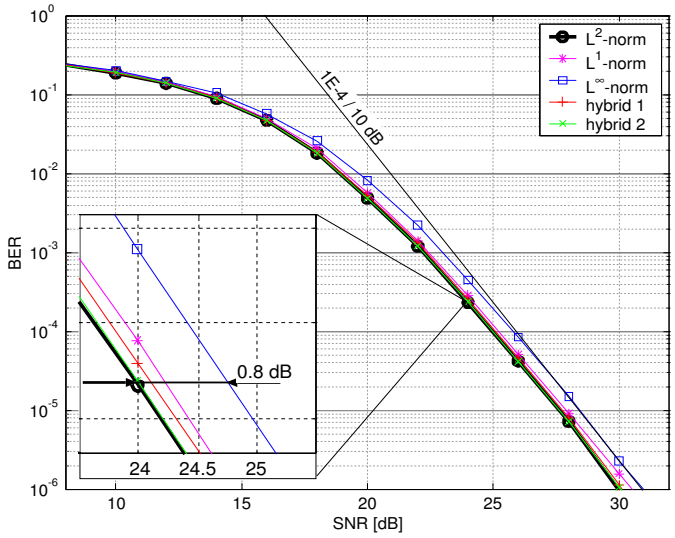

Fig. 4. BER vs. SNR for the square root sphere detector with different approximations. Modulation: 16-QAM, N=M=4

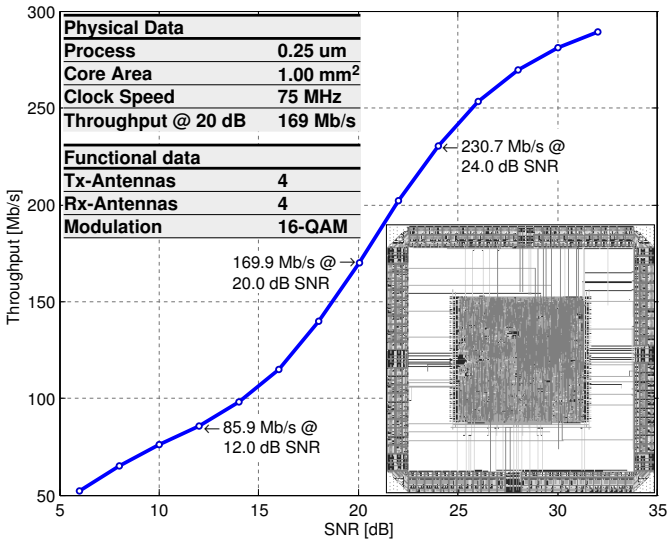

Fig. 5. Layout and summary of technical data of the sphere decoder ASIC

most pronounced constant loss in BER, although the degradation may well be acceptable in many cases.

Concluding, the ℓ^∞-norm approximation represents a very attractive approach as it leads to greatly reduced search complexity as well as reduced chip area at only a minor BER penalty.

VI. ASIC Implementation Results

As a proof of concept, a SD ASIC for a 4×4 system with 16-QAM modulation has been realized in a $0.25\ \mu\mathrm{m}$ technology. It is based on a direct implementation of complex SE enumeration using a decomposition into three nested PSK constellations. For the metric computation, the square root sphere algorithm is used in conjunction with the ℓ^∞-norm. The critical path starts with the computation of $b_i(\mathbf{s})$, followed by the part of the MCU that finds the starting point for the PSK enumeration, and the metric computation. It then continues with the selection of the minimum and into the MEU, adding up to a total delay of $13.5\,\mathrm{ns}$, allowing for a clock of $75\,\mathrm{MHz}$. The active core area of the chip covers only $1\ \mathrm{mm}^2$.

The implementation is about two times smaller and exceeds the throughput of the K-best lattice decoder in [11] by about a factor of three at $\mathrm{SNR} = 20\,\mathrm{dB}$. Compared to a previously presented implementation [8], it only requires a third of the area and achieves a 50% higher clock rate. Also, fewer iterations are required for the decoding due to the complexity reduction by the ℓ^∞-norm approximation and other minor optimizations. The result is a more than doubled throughput at $\mathrm{SNR} = 20\mathrm{dB}$. The technical specifications and the throughput of the ASIC at different SNRs are given in Fig. 5 together with the layout.

VII. Conclusions

We have presented a VLSI architecture for high performance sphere decoding in complex lattices. Implementation techniques to realize Schnorr-Euchner enumeration in complex sets are described and compared. To reduce circuit complexity and cycle time, approximations to the ML criterion are proposed, which preserve diversity order. In particular, an ℓ^∞-norm approximation is shown to additionally improve the tree pruning process in the SD significantly with only a small SNR penalty. The feasibility was shown with an actual ASIC implementation that, to the best of our knowledge, exceeds the throughput of all other presented VLSI realizations.

References

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

[2] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," in *Proc. IEEE ICASSP*, vol. 2, 2002, pp. 1497–1500.

[3] 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.

[4] 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.

[5] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Programming*, vol. 66, pp. 181–191, 1994.

[6] 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.

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

[8] A. Burg, M. Wenk, M. Zellweger, M. Wegmueller, N. Felber, and W. Fichtner, "VLSI implementation of the sphere decoder algorithm," in *Proc. ESSCIRC-2004*, 2004.

[9] 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.

[10] R. van Nee, A. van Zelst, and G. Awater, "Maximum likelihood decoding in a space division multiplexing system," in *Proc. Vehicular Tech. Conf. 2000-Spring*, vol. 1, 2000, pp. 6–10.

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