VLSI IMPLEMENTATION OF PIPELINED SPHERE DECODING WITH EARLY TERMINATION

A. Burg, M. Wenk, and W. Fichtner

Integrated Systems Laboratory
ETH Zurich, 8092 Zurich, Switzerland
email: {apburg, mawenk, fw}@iis.ee.ethz.ch
web: www.iis.ee.ethz.ch

ABSTRACT

The sphere decoding algorithm allows to implement the detection stage in multiple-input multiple-output communication systems with maximum likelihood error rate performance, while the average computational complexity of the algorithms remains far below an exhaustive search. This paper addresses two important problems associated with the practical implementation of sphere decoding: the mitigation of the error rate performance caused by constraining the maximum instantaneous decoding effort and the introduction of pipelining into recursive one-node-per-cycle VLSI architectures for depth-first sphere decoding. The result of this work is a sphere decoder implementation for a 4×4 system with 16-QAM modulation in a 0.13 μm technology that achieves a guaranteed minimum throughput of 761 Mbps.

1. INTRODUCTION

Multiple-input multiple-output (MIMO) systems employing *spatial multiplexing* [1, 2] constitute the basis for many upcoming wireless communication standards such as IEEE 802.11n and IEEE 802.16e. Unfortunately, corresponding receivers are associated with a considerable hardware complexity. In particular, implementations of MIMO detectors with maximum likelihood (ML) bit error rate (BER) performance pose an important research challenge since the complexity of brute-force exhaustive search algorithms grows exponentially in the transmission rate.

The Schnorr-Euchner sphere decoding (SESD) algorithm with radius reduction [3, 4] allows to solve the ML detection problem with a complexity that is - at least on average - far below an exhaustive search. While efficient VLSI implementations of this scheme have been described in [5], it has also been recognized that the sphere decoding algorithm suffers from two main obstacles to its application in high-throughput MIMO systems:

1. The most severe problem in practical implementations is that the complexity of finding the ML solution is variable and in the worst case still corresponds to an exhaustive search. Hence, the decoding effort must be constrained which degrades the BER performance [6].
2. The second problem is that recursive algorithms (i.e., the SESD) are not immediately amenable to pipelining [7], an architectural transformation which allows to increase the throughput of digital integrated circuits with a usually proportionally lower increase in silicon area.

In this paper both issues will be addressed.

Contributions

The first contribution describes a new algorithm which, in coded systems, allows to partially mitigate the performance degradation caused by early termination of the sphere decoding algorithm. The second contribution shows how the one-node-per-cycle VLSI architecture described in [5] can be pipelined to achieve a considerably higher throughput. Finally, implementation results for a pipelined SESD in a 0.13 μm technology are presented, providing reference for the true silicon complexity of the algorithm.

Outline

The remainder of this section introduces the system model and briefly summarizes the sphere decoding algorithm under consideration in this paper. Section 2 describes our novel approach to mitigate the performance loss associated with early termination. Section 3 is concerned with the application of pipelining to high-throughput VLSI architectures for SESD and Section 4 summarizes our implementation results.

1.1 System Model

For the subsequent description of algorithms, consider a flat-fading MIMO system with M_T transmit and M_R receive antennas. The transmitter operates in spatial multiplexing mode which implies that each entry of the M_T-dimensional transmitted vector $\mathbf{s}$ is chosen independently from a set of complex-valued constellation points $\mathcal{O}$ so that $\mathbf{s} \in \mathcal{O}^{M_T}$. The input-output relation describing the M_R-dimensional received vector $\mathbf{y}$ is given by

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

where $\mathbf{H}$ denotes the $M_T \times M_R$-dimensional channel matrix and where the entries of the noise vector $\mathbf{n}$ are i.i.d. complex Gaussian distributed with variance σ^2 per complex dimension. The signal to noise ratio (SNR) is defined as $\mathrm{SNR} = \mathcal{E}\{\|\mathbf{s}\|^2\}/\sigma^2$, where $\mathcal{E}\{\cdot\}$ denotes the expectation. The ML criterion for estimating $\mathbf{s}$ from $\mathbf{y}$ using knowledge of $\mathbf{H}$ is given by

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

Since $|\mathcal{O}^{M_T}|$ grows exponentially in the transmission rate, solving (2) with an exhaustive search is prohibitively complex for rates greater than 8 bits per channel use [8].

1.2 The Sphere Decoding Algorithm

Sphere decoding aims at avoiding an exhaustive search. To this end, the algorithm starts from the QR decomposition of $\mathbf{H} = \mathbf{Q}\mathbf{R}$ where $\mathbf{Q}$ is unitary and $\mathbf{R}$ is upper triangular and considers $\hat{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$. With this unitary transformation of the received vector the solution of (2) corresponds to

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s} \in \mathcal{O}^{M_T}} \{d(\mathbf{s})\} \quad \text{with} \quad d(\mathbf{s}) = \|\hat{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2, \tag{3}$$

where the distance $d(\mathbf{s}) = d_1(\mathbf{s})$ can be computed recursively according to

$$d_i\left(\mathbf{s}^{(i)}\right) = d_{i+1}\left(\mathbf{s}^{(i+1)}\right) + |b_{i+1} - R_{ii}s_i|^2 \tag{4}$$

$$\text{with} \quad b_{i+1} = \hat{y}_i - \sum_{j=i+1}^{M_T} R_{ij}s_j \tag{5}$$

after initializing $d_{M_T+1}(\mathbf{s}) = 0$. Since the partial Euclidean distances (PEDs) $d_i(\mathbf{s}^{(i)})$ depend only on $\mathbf{s}^{(i)} = [s_i \quad \ldots \quad \mathbf{s}_{M_T}]^T$ they can be associated with nodes in a tree. Finding the ML solution then corresponds to traversing this tree to identify the leaf with the smallest PED. The basic idea that leads to a complexity reduction compared to an exhaustive search is to restrict the search to only those $\mathbf{s} \in \mathcal{O}^{M_T}$ for which $\mathbf{Rs}$ lies within a hypersphere of radius r around $\hat{\mathbf{y}}$. To this end, the SESD traverses the tree depth-first and prunes all nodes from the tree for which $d_i(\mathbf{s}^{(i)}) > r^2$. The children of a node are thereby examined in ascending order of their PEDs and the radius is updated according to $r^2 \leftarrow d(\mathbf{s})$ whenever a leaf is found for which $d(\mathbf{s}) < r^2$.

For a one-node-per-cycle VLSI architecture for SESD the decoding effort to identify the ML solution is largely determined by the number of visited nodes [5] which corresponds to the number of forward and backward iterations in the tree (the latter can also span multiple levels of the tree). Unfortunately, this effort is variable (depending on the transmitted vector symbol, the noise-, and channel realizations) and can, for individual symbols, by far exceed the average decoding effort, in the worst case requiring an exhaustive search. *Early termination* (ET) solves this problem by imposing a constraint $D_{\max}$ on the number of visited nodes. When this limit is reached, the decoder stops and returns the best solution it has found so far[1]. However, for symbols affected by ET the output of the decoder does not necessarily correspond to the ML solution which severely degrades the BER performance [6]. For the uncoded case it has been shown in [6] that replacing the per-symbol runtime constraint with a block runtime-constraint can alleviate this problem by allocating the available processing resources (i.e., time) to those symbols requiring a higher decoding effort. In this paper, we pursue a completely different approach to mitigate the performance loss caused by ET. This technique can also be combined with the block runtime-constraint proposed in [6].

2. EARLY TERMINATED SESD WITH SOFT OUTPUT

The algorithm described in the following is motivated by the observation that decisions on vector symbols (and thus also on the associated bits) affected by ET are on average less reliable compared to other symbols for which the SESD was able to complete the search for the ML solution within the allocated runtime limit. The basic idea is to supplement the binary decisions of the SESD constrained by ET (ET-SESD) with reliability information derived from the termination status of the decoder. The resulting log-likelihood ratios (LLR) can then be forwarded to a subsequent soft-input channel decoder (as illustrated in Fig. 1) which may use this additional information to more reliably recover the actual data.

2.1 Computing Approximate Log-Likelihood Ratios from Bit Error Probabilities

Before considering the specific problem of deriving LLRs as a function of the termination status of the ET-SESD we shall briefly introduce a new approach to compute approximate LLRs from a limited set of side information. To this end, let $b_m^{(i)}$ denote the ith bit transmitted from the mth antenna[2]. The LLR of an ideal soft-output detector is given by the ratio of the probabilities that a zero or a one has been transmitted conditioned on the received vector, the channel, and the SNR.

$$L\left(b_m^{(i)}\right) = \log\left(\frac{P(b_m^{(i)} = 0|\mathbf{y}, \mathbf{H}, \mathrm{SNR})}{P(b_m^{(i)} = 1|\mathbf{y}, \mathbf{H}, \mathrm{SNR})}\right) \tag{6}$$

Now consider a scenario where only the output $\hat{b}_m^{(i)}$ of a hard-decision MIMO detector and some arbitrary side-information on the average reliability of $\hat{b}_m^{(i)}$ is available to compute soft-information. Under these circumstances the best possible estimate of the LLR of $b_m^{(i)}$ is given by

$$\tilde{L}\left(b_m^{(i)}\right) = \log\left(\frac{P(b_m^{(i)} = 0|\hat{b}_m^{(i)}, \mathcal{T})}{P(b_m^{(i)} = 1|\hat{b}_m^{(i)}, \mathcal{T})}\right), \tag{7}$$

where the set $\mathcal{T}$ comprises all available side information. Assuming a symmetric error probability for $\hat{b}_m^{(i)}$ conditioned on $\mathcal{T}$ so that

$$\begin{aligned} P(b_m^{(i)} \neq 1|\hat{b}_m^{(i)} = 1, \mathcal{T}) =& P(b_m^{(i)} \neq 0|\hat{b}_m^{(i)} = 0, \mathcal{T}) \\ =& P(b_m^{(i)} \neq \hat{b}_m^{(i)}|\mathcal{T}) \end{aligned} \tag{8}$$

one can write (7) as a function of the bit error probability of the corresponding hard-output detector, conditioned on $\mathcal{T}$ according to

$$\tilde{L}(b_m^{(i)}) = \begin{cases} W_m^{(i)}(\mathcal{T}), & \hat{b}_m^{(i)} = 0 \\ -W_m^{(i)}(\mathcal{T}), & \hat{b}_m^{(i)} = 1 \end{cases} \quad \text{with} \tag{9}$$

$$W_m^{(i)}(\mathcal{T}) = \log\left(\frac{1 - P(b_m^{(i)} \neq \hat{b}_m^{(i)}|\mathcal{T})}{P(b_m^{(i)} \neq \hat{b}_m^{(i)}|\mathcal{T})}\right), \tag{10}$$

where we have substituted $P(b_m^{(i)} = \hat{b}_m^{(i)}|\mathcal{T}) = 1 - P(b_m^{(i)} \neq \hat{b}_m^{(i)}|\mathcal{T})$.

2.2 A Pragmatic Application to SESD with Early Termination

For the ET-SESD, the relevant side information $\mathcal{T}$ is comprised of the SNR, the runtime limit $D_{\max}$, and of a binary flag T which indicates whether the decoding process had to be terminated prematurely ($T = 1$) or not ($T = 0$)

$$\mathcal{T} : \{\mathrm{SNR}, D_{\max}, T\}. \tag{11}$$

The conditional (uncoded) error probabilities required for the computation of $W_m^{(i)}$ can be easily obtained by computer simulations using a fast-fading (temporally white) narrow-band channel. For $T = 0$ (no early termination) $P(b_m^{(i)} \neq \hat{b}_m^{(i)}|\mathcal{T})$ simply corresponds to the BER performance of the SESD without runtime constraint. For $T = 1$ only bits affected by ET after $D_{\max}$ visited nodes should ideally be taken into account to obtain the corresponding BER. However, the average error probability (including those bits, not affected by ET) of a SESD with ET after $D_{\max}$ visited nodes is a reasonable approximation to $P(b_m^{(i)} \neq \hat{b}_m^{(i)}|\mathcal{T})$ since the error performance is clearly dominated by those

[1] Note that if the initial radius is set to $r = \infty$, the SESD always finds the nulling and canceling solution after M_T visited nodes.

[2] In the following, we assume $P(b_m^{(i)} = 0) = P(b_m^{(i)} = 1) = 1/2$, where $P(\cdot)$ denotes the probability of an event.

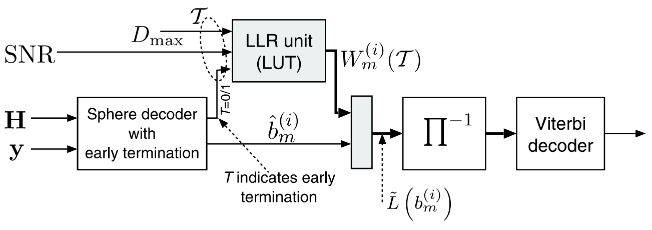

Figure 1: Block diagram of SESD with ET and soft-output.

symbols affected by the runtime constraint. Once the conditional error probabilities are known, the reliability estimates $W_m^{(i)}(\mathcal{T})$ can be computed and stored in a small look-up table (LUT). In the present implementation, no distinction is thereby made between the bits encoded within the same vector symbol, considering only their average reliability so that $W_m^{(i)}(\mathcal{T}) = W(\mathcal{T})$.

During decoding, the LUT is then indexed by $D_{\max}$, by the quantized SNR and by the early termination indicator $\mathcal{T}$ as illustrated by the block diagram in Fig. 1. The LUT output $W(\mathcal{T})$ is then combined with the tentative decision of the SESD according to (9) and the resulting LLR estimate is passed on to the channel decoder via a deinterleaver (Π^{-1}).

2.3 BER Simulation Results

For evaluating the BER performance improvement achieved by the described algorithm consider a coded MIMO-OFDM system with $M_R = M_T = 4$ and 16-QAM modulation. The FFT-length is 64 and the cyclic prefix has a length of 16 samples. Forward error correction coding is performed with a rate 1/2 convolutional code with constraint length $K = 7$ specified by the polynomial [133o,171o]. The length of a code block is defined by the number of bits in a single MIMO-OFDM symbol and the bits are interleaved randomly across tones and antennas. At the receiver, perfect channel knowledge is assumed and a (soft-input) Viterbi decoder with a traceback length of 55 is employed for decoding of the convolutional code. For the subsequently presented simulations, the channels used for the generation of the entries of the LLR LUTs were chosen to exhibit the same spatial correlation properties as the channels applied in the OFDM system under consideration.

2.3.1 Comparison of Algorithms

Fig. 2 shows the BER performance of the ET-SESD with $D_{\max} = 7$ and $D_{\max} = 10$ visited nodes, with and without reliability information. The frequency selective channel model used for the simulations corresponds to the model "C" defined by the IEEE 802.11n task group [9] where we have set an antenna spacing of one wavelength. Clearly, the use of approximate reliability information leads to a considerable BER performance improvement compared to the case where only hard-decisions are forwarded to the channel decoder. From Fig. 2 it can also be observed that the gain from reliability information increases for better BER performance requirements and decreases as $D_{\max}$ increases.

2.3.2 Impact of the Channel

The question arises to what extend the performance of the ET-SESD and the performance gain from the proposed algorithm depend on the variability of the channel within a single code block. In order to analyze this dependency we shall consider three artificial channels with one, two, and four sample-spaced taps of equal power which are all i.i.d. complex Gaussian (temporally and spatially white). The corresponding simulation results are summarized in Tbl. 1 which

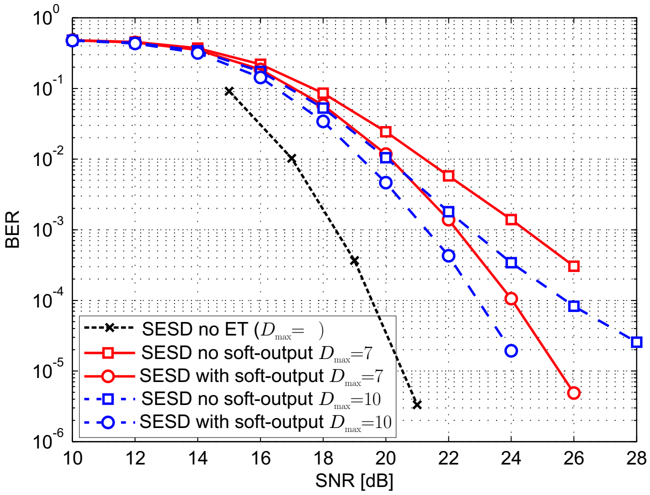

Figure 2: BER performance for a rate 1/2 coded 4×4 system with 16-QAM modulation.

reports the SNR penalty of the ET-SESD (based on the ℓ^∞-norm) after $D_{\max} = 7$ with and without reliability information compared to an ML detector (i.e., SESD *without* ET). It can be seen that with only a single tap (flat-fading) the ET-SESD suffers from a considerable performance penalty similar to the uncoded case [6] since no frequency diversity is available to partially compensate for the lack of spatial diversity due to ET. However, as frequency diversity increases (as for the two and four-tap channel), the performance loss caused by ET reduces quickly. The use of reliability information provides an advantage for all three power-delay profiles under consideration, where the corresponding gain is most pronounced for the two-tap channel showing a 3.2 dB performance improvement at a BER of 10^{-4}.

Table 1: Comparison of ET-SESD ($D_{\max} = 7$) to ML.

Channel	ML detector	SNR gap of ET-SESD	
	BER @ SNR	hard-out	soft-out
1 tap	10^{-3} @ 18.7 dB	12.4 dB	11.3 dB
	10^{-4} @ 20.5 dB	n.a.	n.a.
2 tap	10^{-3} @ 17.1 dB	4.5 dB	3.2 dB
	10^{-4} @ 18.4 dB	6.9 dB	3.7 dB
4 tap	10^{-3} @ 16.6 dB	2.7 dB	2.2 dB
	10^{-4} @ 17.8 dB	4.5 dB	2.8 dB

3. PIPELINED VLSI ARCHITECTURE

We shall now turn our attention to the optimization of the VLSI architecture of the SESD. The goals are either to achieve higher average throughput or to allow for a larger $D_{\max}$ for a given guaranteed minimum throughput requirement, striving for better BER performance. To this end, we start with the one-node-per-cycle architecture proposed in [5] and depicted, in a slightly modified form, in Fig. 3(a). The implementation is comprised of a *metric computation unit* (MCU) which handles the forward iteration through the tree and of a *metric enumeration unit* (MEU) which prepares for the moment when the forward iteration stalls and the decoder needs to proceed with a node closer to the root. The critical path of the corresponding circuit is the first-order feedback loop [c.f. Fig. 3(b)] through the MCU which is ex-

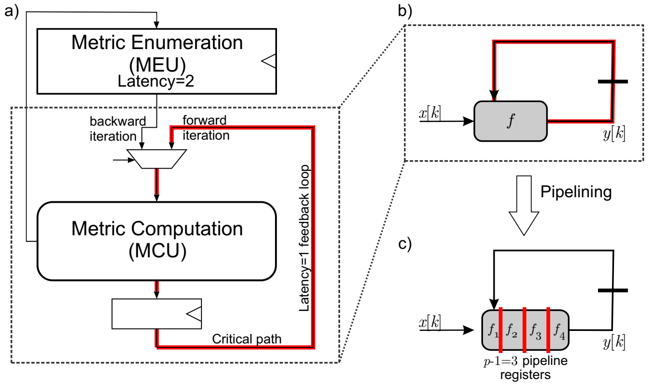

Figure 3: a) One-node-per-cycle VLSI architecture, b) data dependency graph of a first order feedback loop, c) first order feedback loop after insertion of $p - 1 = 3$ pipeline register.

cited when the decoder proceeds in forward direction from a node to one of its children.

To achieve a considerably higher throughput with this architecture, the combinational logic in the MCU must be broken up by inserting pipelining registers. Unfortunately, due to the presence of feedback such a straightforward modification also alters the functionality of the circuit [5].

3.1 Pipelining Recursive Algorithms

Pipeline interleaving [7] is a method that allows to cut the combinational delay in a feedback loop provided that multiple independent data streams can be processed. To illustrate the basic idea, consider the data dependency graph (DDG) in Fig. 3(b) which is described by

$$y[k] = f\left(y\left[k-1\right], x\left[k\right]\right). \tag{12}$$

Inserting $p - 1$ pipeline registers into the corresponding circuit yields the DDG shown in Fig. 3(c). With proper retiming of the registers this architectural transformation reduces the length of the critical path by up to a factor of $1/p$, but the transfer function of the modified circuit is now given by

$$y[k] = f\left(y\left[k-p\right], x\left[k-p+1\right]\right), \tag{13}$$

which is no longer equivalent to (12). However, it is easy to show that for p independent data streams $x_0[t], \ldots, x_{p-1}[t]$ setting $x[tp + n - p + 1] = x_n[t]$ (i.e., $k = tp + n$) yields p independent $y_n[t] = y[tp + n]$ with $n = 0, \ldots, p - 1$ so that

$$y_n[t] = f\left(y_n\left[t-1\right], x_n\left[t\right]\right) \tag{14}$$

as desired. In other words, the pipelined circuit can effectively process p data streams concurrently in an interleaved fashion at a higher clock rate which enables a higher aggregate throughput.

3.2 Pipelined Sphere Decoder Architecture

Since for the purpose of MIMO detection, subsequent received vectors are considered to be independent of each other, pipeline interleaving is applicable to sphere decoding. Fig. 4 illustrates the insertion of $p - 1 = 2$ pipeline stages into the direct-QAM enumeration based one-node-per-cycle VLSI architecture described in [5].

The architectural transformation of the originally purely combinatoric MCU and of the corresponding first-order feedback loop is straightforward. However, the MEU that originally had a latency of $L_{\mathrm{MEU}} = 2$ requires special attention as it contains cache memories which retain data over multiple iterations. To be able to process p data streams in an interleaved fashion, each of these cache memories must be replicated p times as shown in Fig. 4. Because the additional memory can already be used to match the delays of the pipeline registers inserted in the MCU, no additional pipeline stages would be required in the MEU. However, the length of the critical path through the MEU must be kept below or at least on par with the length of the shortened critical path through the pipelined MCU. To this end, up to $(p-1)L_{\mathrm{MEU}}$ pipeline stages can be inserted into the data path of the MEU as needed to adjust the length of the critical path and potential latency differences can be equalized by proper address generation for the replicated caches.

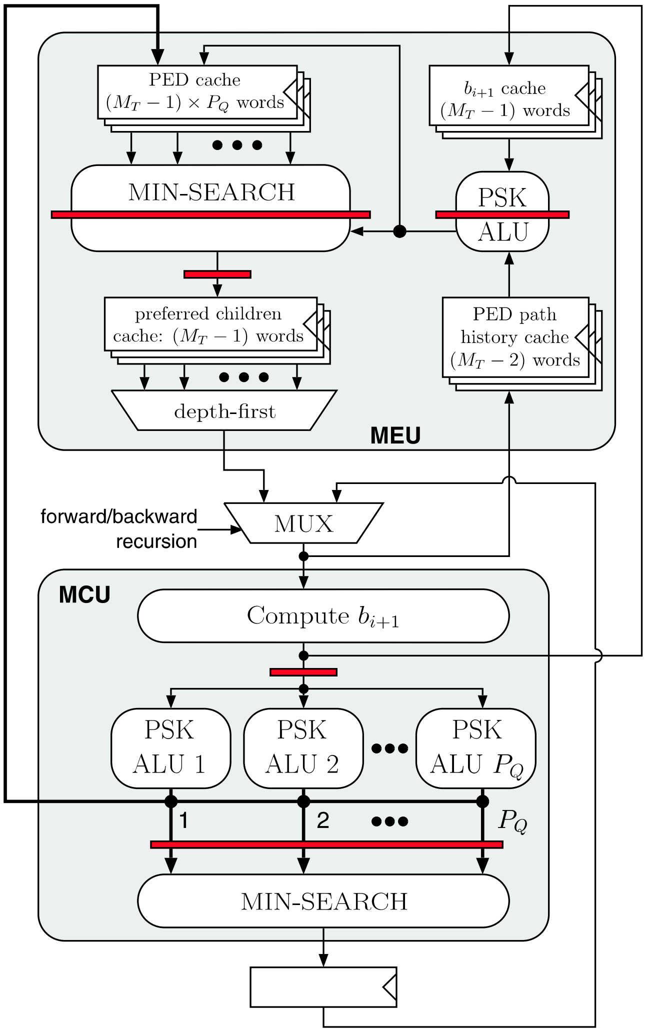

Figure 4: Block diagram of the SESD described in [5] with 3 pipeline stages. For 16-QAM, $P_Q = 3$.

4. VLSI IMPLEMENTATION RESULTS

To assess the true silicon complexity of the pipelined architecture and to properly estimate the achievable throughput the proposed circuit has been implemented in a 0.25μm and in a 0.13μm technology. The corresponding results are summarized in Tbl. 2 together with the implementation results of the original unpipelined architecture described in [5]. The layout of the pipelined SESD core (in 0.13μm technology) and the layout of the corresponding ASIC are depicted in

Fig. 5.

Table 2: VLSI implementation results for sphere decoders.

System conf.	$M_T = M_R = 4$, 16-QAM modulation		
Reference	[5]	This work	This work
Pipelined	NO	YES	YES
Technology	0.25 μm		0.13 μm
Area[3]	50K GE	73K GE	90K GE
Max. clock	71 MHz	180 MHz	333 MHz
Guaranteed minimum throughput with early termination			
$D_{\max} = 7$	162 Mbps	411 Mbps	761 Mbps
$D_{\max} = 10$	113 Mbps	288 Mbps	533 Mbps
Average throughput (no early termination)			
SNR = 22 dB[4]	193 Mbps	488 Mbps	903 Mbps

4.1 Impact of Pipelining

A comparison of the second and the third column in Tbl. 2 shows that the pipelined design is roughly 50% larger but allows for more than twice the clock frequency compared to the original design [5]. While the speedup is below the maximum factor of three due to nonideal placement of the pipeline registers, the overall AT-product is still better compared to the design in [5].

Consideration of the layout in Fig. 5 and of the area of the individual components reveals further insight into what constitutes the major portions of the design: With 55% the MEU consumes by far the most area and has grown significantly after the insertion of pipeline registers. The main reason for this is the need to replicate the caches. The MCU on the other hand only requires few additional registers and consumes only 20% of the silicon real-estate. Besides 3% control overhead, the remaining 16% and 6% of the total area are used by double-buffered memories (built from flip-flops) that store $\mathbf{R}$ and $\hat{\mathbf{y}}$ for the three symbols that are being processed concurrently.

A potential problem arising from the concurrent processing of multiple symbols with SESD is that due to the variable runtime symbols may overtake preceding symbols in the decoding process. However, in practice, the required reorder buffers can be hidden in the subsequent interleavers and the use of ET reduces the decoding-delay spread to only a few symbols.

4.2 Impact of ET-Based Soft-Outputs

The area overhead associated with providing soft-outputs to partially mitigate the impact of early termination as described in Section 2 corresponds to the area required for the implementation of the LUT in Fig. 1. Synthesis results show that even an extensive table with a total of 460 8-bit entries (e.g., 10 for the SNR × 23 for $D_{\max}$ × 2 for the termination status) requires only 1K GE which is less than 2% of the overall core area. In terms of timing, it is found that the critical path of the LUT (which could also be pipelined if needed) lies far below the critical path of the pipelined SESD and poses therefore no limitation on the achievable throughput.

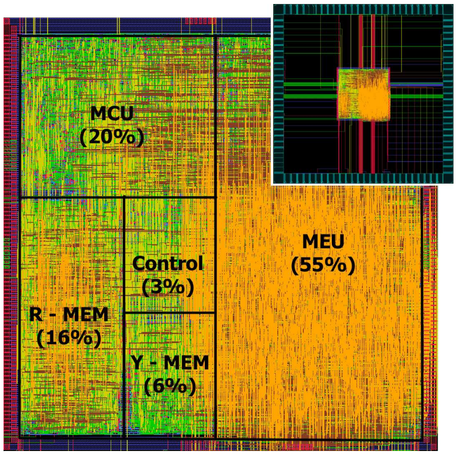

Figure 5: Layout of the 3× pipelined SESD in a 0.13μm 1P/5M technology.

5. CONCLUSIONS

Approximate soft-information derived from expected error probabilities can be used to partially mitigate the performance loss associated with sphere decoding with early termination. The implementation of the proposed algorithm requires only a small look-up table which incurs only a negligible increase in circuit complexity. To achieve very high throughput with low area-delay products, sphere decoder architectures can be pipelined to allow for efficient processing of multiple received vectors in an interleaved fashion.

REFERENCES

[1] G. Foschini and M. Gans, “On limits of wireless communications in a fading environment when using multiple antennas,” *Wireless Personal Communications*, vol. 6, no. 3, pp. 311–334, 1998.

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

[3] C. P. Schnorr and M. Euchner, “Lattice basis reduction: Improving practical lattice basis reduction and solving subset sum problems,” *Math. Programming*, vol. 66, pp. 181–199, 1994.

[4] M. O. Damen, H. El Gamal, and G. Caire, “On maximum-likelihood detection and the search for the closest lattice point,” *IEEE Transactions on Information Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[5] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, “VLSI implementation of MIMO detection using the sphere decoder algorithm,” *IEEE Journal of Solid-State Circuits*, 2005.

[6] A. Burg, M. Borgmann, M. Wenk, C. Studer, and H. Bölcskei, “Advanced receiver algorithms for mimo wireless communications,” in *Proc. ACM Design Automation and Test in Europe Conf.*, Mar. 2006.

[7] H. Kaeslin, “Lecture notes on VLSI I,” 2004, IIS/D-ITET, ETH-Zurich.

[8] 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. on Electronics, Circuits, and Systems*, vol. 1, Dec. 2003, pp. 332–335.

[9] *IEEE 802.11 TGn Channel Models*, May 2004, IEEE 802.11-03/940r4.

[3] Cell area (excluding routing overhead) is specified in gate equivalents (GE), where one GE corresponds to the cell area of a 2-input drive-one NAND gate in the respective technology.

[4] For an SNR of 22 dB the ℓ^{∞}-norm SESD described in [5] visits on average 5.9 nodes per received vector.