



An energy efficient weakly programmable MIMO detector architecture

C. Gimmler-Dumont and N. Wehn

University of Kaiserslautern, Germany

Correspondence to: C. Gimmler-Dumont (gimmler@eit.uni-kl.de)

Abstract. Energy efficient processing is mandatory in today's mobile devices. For the upcoming multiple-antenna systems, algorithmic flexibility enables the dynamic reaction to changing channel conditions. We show that most of the tree search based MIMO detection algorithms are based on the same algorithmic kernels and present a weakly-programmable architecture based on these observations. In this way, the detection algorithm can be chosen and parameterized during runtime according to the current channel conditions and QoS requirements leading to a highly energy efficient implementation. The architecture has been implemented and synthesized on a 65 nm technology, resulting in an area of 0.26 mm^2 and a power consumption of only 15 mW.

1 Introduction

Multiple-antenna or MIMO systems have the potential to increase the data rate of wireless communication systems and, thus, they have been adopted by recent communication standards like WiMax, WiFi, or LTE. The best communications performance is achieved for an iterative processing between MIMO detector and channel decoder. As for channel codes, the encoding on the transmitter side is simple but the detection complexity of the MIMO signals at the receiver increases exponentially with the number of antennas.

There exist a wide range of sub-optimal algorithms which allow trade-offs of communications performance versus implementation complexity and energy consumption. Linear detection methods, e.g. successive interference cancellation (SIC) introduced by Foschini et al. (1999), have the lowest complexity but also the lowest communications performance. Optimal soft-input soft-output sphere detection is the most demanding detection algorithm but it offers the best communications performance and the ability for iterative de-

tection and decoding (Vikalo et al. (2004)). A fixed target error rate can be achieved with different detection algorithms which cover a signal-to-noise ratio (SNR) range of more than 10 dB.

In current standards, like LTE, the quality of service (QoS) is dynamically adjusted at runtime, e.g. higher data throughput rates are specified for higher SNR. This is due to the fact that for higher SNR, the required communications performance can be achieved by lower complexity algorithms, which enables higher throughput. Energy consumption is one of the most critical implementation metrics for mobile receivers. Thus, complexity adaptive processing is key for enabling energy efficient architectures (Yu et al. (2012)). Therefore, flexible energy efficient MIMO detection architectures are mandatory to fulfill the requirements of modern communication standards like LTE.

Visionary implementation styles are based on libraries of flexible algorithmic kernels, e.g. the Nucleus concept proposed by Ramakrishnan et al. (2009). These kernels offer core functionalities which can be reused in different algorithms or scenarios. However, defining the best set of kernels is one of the major challenges of this approach. Extracting the common parts of different algorithms allows a high degree of shared and highly optimized components and, thus, a flexible implementation with a low overhead. Weakly programmable architectures based on such kernels can lead to highly energy efficient hardware implementations.

However, in the literature, mainly highly optimized architectures are presented which perform exactly one algorithm, e.g. Borlenghi et al. (2011); Studer et al. (2011). There exist only a few processor architectures which are able to perform different algorithms. In Jafri et al. (2009), the processor architecture is based on very small-grained operations, i.e. complex number operations. In Chen et al. (2012), a processor architecture based on matrix operations is presented which performs maximum ratio combining, linear

MMSE detection, and MMSE-SIC detection. However, none of the processor architectures supports tree search algorithms which offer the best communications performance. Thus, they only cover a limited range of the possible channel scenarios.

Almost all detection algorithms can be mapped to a search in a tree structure. We will show that all these algorithms can be constructed by only five coarse-grained algorithmic kernels: an enumeration unit which decides which child node is visited next, a computation unit for the interference reduction, a computation unit for the Euclidean distance, administration of nodes, and administration of computed metrics. We present an architecture based on these kernels which is able to perform most of the existing algorithms. The complexity of these algorithms covers all classes from linear up to almost exponential complexity, e.g. linear SIC, fixed effort detection (Wong et al. (2002); Barbero and Thompson (2008)), or sphere detection. As all algorithms are performed with the same algorithmic kernels, the overhead for flexibility is negligible.

In contrast to the existing approaches, we present the first weakly programmable MIMO detector architecture which offers just the necessary flexibility. In this way, the detection algorithm can be chosen and parameterized during runtime according to the current channel conditions and QoS requirements leading to a highly energy efficient implementation. The architecture has been implemented and synthesized on a 65 nm technology, resulting in an area of 0.26 mm², a power consumption of 15 mW and throughputs between 35 Mbit/s up to 720 Mbit/s.

In the following sections, we will introduce the employed system model and the basics of tree search based MIMO detection. In Sect. 4, we will derive the algorithmic kernels which are shared by most MIMO detection algorithms. Implementation results of the flexible detection architecture and the trade-off between throughput and communications performance resulting from this flexibility are presented in Sect. 5. Sect. 6 concludes the paper.

2 System Model

In this paper, we focus on a bit interleaved coded modulation (BICM) scheme like shown in Fig. 1. The source generates a random information word \mathbf{u} of length K_c which is encoded by the channel encoder. The interleaved codeword X^N is mapped directly to complex symbols s chosen from a 2^Q -ary QAM modulation scheme. M_T symbols are combined in one transmission vector s_t . The whole modulated sequence is represented by

$$S^T = (s_1, s_2, \dots, s_t, \dots, s_T) \quad (1)$$

T time slots are needed to transmit all symbols of one codeword. The transmission of vector s_t in time step t is modeled

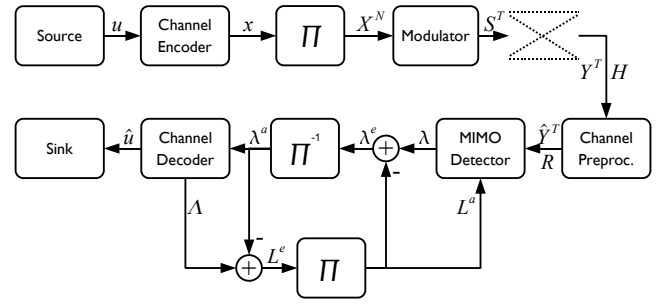


Fig. 1. MIMO-BICM system with feedback loop between MIMO Detector and Channel Decoder

by

$$\mathbf{y}_t = \mathbf{H}_t \cdot \mathbf{s}_t + \mathbf{n}_t \quad (2)$$

with \mathbf{H}_t the channel matrix of dimension $M_T \times M_R$ and \mathbf{n}_t the noise vector of dimension M_R whose entries are zero-mean and unit variance Gaussian variables. The elements of \mathbf{H}_t are modeled as independent, complex, zero-mean, Gaussian random variables. Real and imaginary part are independent variables each with variance $\sigma^2 = N_0/2$. All received vectors \mathbf{y}_t are gathered in the matrix \mathbf{Y}^T

$$\mathbf{Y}^T = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_t, \dots, \mathbf{y}_T) \quad (3)$$

Before the decoding starts, the channel preprocessing applies a QR decomposition on \mathbf{Y}^T and \mathbf{H}_t . This results in the transformed received vectors $\hat{\mathbf{Y}}^T$ and updated channel matrices \mathbf{R}_t . The decoding process iterates over the MIMO detector and the channel decoder, which exchange probability information about the codeword. The soft-in-soft-out MIMO detector determines the likelihood of the bits for each received vector \mathbf{y}_t using the a-priori information \mathbf{L}^a from the channel decoder. Only the extrinsic information $\lambda^e = \lambda - \mathbf{L}^a$ is passed on to the channel decoder.

The channel decoder processes the whole codeword at a time. It uses the interleaved a-priori information λ^a from the MIMO detector for the calculation of the estimated information bit sequence $\hat{\mathbf{u}}$ and the a-posteriori logarithmic likelihood ratios (LLRs) $\mathbf{\Lambda}$ of the codeword. The extrinsic information $\mathbf{L}^e = \mathbf{\Lambda} - \lambda^a$ is returned to the MIMO detector thus closing the iterative loop.

3 MIMO Detection

A received symbol vector \mathbf{y}_t can be seen as a weighted superposition of the entries of \mathbf{s}_t , disturbed by Gaussian noise. The task of the MIMO detector is the equalization and separation of the originally sent sequence of symbols \mathbf{s}_t . It works on one received vector \mathbf{y}_t at a time. For all detection related explanations, the time indexes of \mathbf{y} , \mathbf{H} , etc. are dropped for ease of notation. Even if not mentioned specifically for each

equation, the vectors \mathbf{s} and \mathbf{x} are always the complex representation and the bit representation, respectively, of the same symbol vector. $x_{q,m}$ denotes the q th bit of the m th symbol in \mathbf{s} .

There exist a wide range of suboptimal algorithms for MIMO detection, which allow trade-offs of communications performance versus implementation complexity and energy consumption. In this paper, we restrict ourselves to the most common detection algorithms which are based on a tree search. It can be shown that the likeliness of having sent a certain vector \mathbf{s} is related to the metric $d(\mathbf{s})$ (Vikalov et al. (2004)).

$$d(\mathbf{s}) = \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 - N_0 \sum_{\forall q,m} \ln P(x_{q,m}) \quad (4)$$

Smaller values of $d(\mathbf{s})$ relate to a higher probability of \mathbf{s} having been sent. The metrics $d(\mathbf{s})$ are used to compute LLR values for the channel decoder.

$$\lambda(x_{q,m}) \approx \min_{\forall \mathbf{s} | x_{q,m}=+1} d(\mathbf{s}) - \min_{\forall \mathbf{s} | x_{q,m}=-1} d(\mathbf{s}) \quad (5)$$

For the optimum calculation of Eq. (5), several minimum searches on $d(\mathbf{s})$ over all vectors \mathbf{s} have to be performed. However, calculating all possible $d(\mathbf{s})$ quickly becomes infeasible for a larger number of receive- and transmit antennas and/or higher order modulations as the complexity grows with 2^{QM} . In order to simplify the metric calculations as well as the minimum search, the calculation of Eq. (4) is mapped on a tree structure. Therefore, the channel matrix \mathbf{H} is decomposed into a unitary matrix \mathbf{Q} and an upper-triangular matrix \mathbf{R} . The Euclidean distance is rewritten as

$$\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 = \|\hat{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 \quad (6)$$

with $\hat{\mathbf{y}} = \mathbf{Q}^H \mathbf{y}$. Equation Eq. (4) is replaced by the equivalent metric

$$d(\mathbf{s}) = \|\hat{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 - N_0 \sum_{\forall q,m} \ln P(x_{q,m}) \quad (7)$$

The triangular structure of \mathbf{R} allows the recursive calculation of $d(\mathbf{s})$

$$d_m = d_{m+1} + \gamma_m(s^{(m)}) \quad (8)$$

with the starting point $d_{M+1} = 0$ and $d(\mathbf{s}) = d_1$. The metric update $\gamma_m(s^{(m)})$ depends on the partial symbol vector $s^{(m)} = (s_m, s_{m+1}, \dots, s_M)$.

$$\gamma_m(s^{(m)}) = |\hat{y}_m - \sum_{j=m}^M R_{m,j} s_j|^2 - N_0 \sum_{q=1}^Q \ln P(x_{q,m}) \quad (9)$$

This equation can be further simplified by introducing the interference-reduced symbol \hat{y}'_m , which is the same for all children of a node.

$$\hat{y}'_m = \hat{y}_m - \sum_{j=m+1}^M R_{m,j} s_j \quad (10)$$

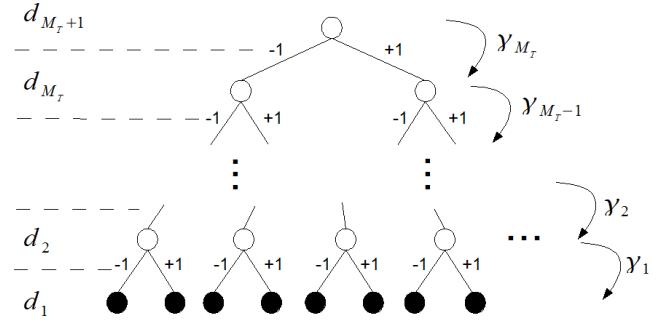


Fig. 2. Detection problem represented by a tree for the modulation alphabet $\{-1, +1\}$ (BPSK) and M_T antennas

$$\gamma_m(s^{(m)}) = |\hat{y}'_m - R_{m,m} s_m|^2 - N_0 \sum_{q=1}^Q \ln P(x_{q,m}) \quad (11)$$

The recursive calculation of Eq. (8) can be represented by a tree with $M + 1$ levels, as shown for the modulation alphabet $\{-1, +1\}$ in Fig. 2. The root node corresponds to d_{M+1} and each leaf node corresponds to the metric $d(\mathbf{s})$ of one possible vector \mathbf{s} . Each level corresponds to the detection of one symbol s_m . Branches are labeled with an element of the modulation alphabet. When advancing from a parent node to a child node, the child node's metric d_m is calculated from the metric of its parent d_{m+1} and the branch metric γ_m .

Based on this tree search, many different MIMO detection algorithms exist which approximate Eq. (5). The main differences between the algorithms can be described by how they traverse the tree, e.g., breadth-first, depth-first, or metric-first, and how branches of the tree are pruned. In general, these algorithms exhibit different communications performances and implementation complexities. All tree search based algorithms require the same channel preprocessing which performs the QR decomposition of the channel matrix \mathbf{H} . This preprocessing only has to be done when the channel is changing and not for every transmission vector. For these reasons, we concentrate only on the MIMO detection itself. QR decomposition implementations can be found in Nazar et al. (2010) for example.

4 Algorithmic kernels

Complexity adaptive processing enables energy efficient implementations. For different channel conditions, a variety of MIMO detection algorithms exist which trade-off communications performance and energy consumption. An energy efficient architecture, thus, needs the flexibility to support several detection algorithms. A weakly programmable architecture which is based on algorithmic kernels needed by all algorithms may offer this flexibility with a negligible overhead. In this section, we will derive the algorithmic kernels

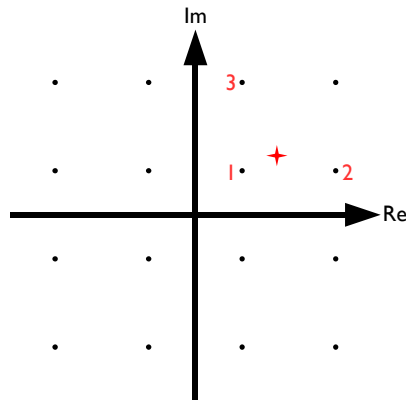


Fig. 3. SISO detection problem

of tree search based MIMO detection which are common to most detection algorithms.

We will start with some observations on the detection in single-input single-output (SISO) systems. The SISO transmission can be modeled by Eq. (2) and the detection is then performed according to Eq. (5) for $M_T = M_R = 1$. We will now depict the different steps involved in the SISO detection. Thereby, we will differentiate between maximum likelihood (ML) or hard-output and maximum a-posteriori (MAP) or soft-output detection. Figure 3 illustrates the SISO detection problem. A complex value y (represented by a red star) has been received. The ML solution is simply the modulation symbol which is closest to y (indicated by the number 1). The algorithm consists therefore of a single step: determining the closest modulation symbol. For the MAP solution, the closest symbols for each bit being 0 or 1 have to be found according to Eq. (5). This requires the enumeration of the best symbols starting with the closest one (indicated by the numbers 1, 2, 3), followed by the calculation of the Euclidean distance between y and those symbols. Each distance is used to update the minima for the calculation of Eq. (5).

The detection of MIMO signals cannot be mapped on a diagram as shown in Fig. 3 as the signal space is multi-dimensional and not rectangular. Therefore, we used the QR decomposition of the channel matrix \mathbf{H} to map the detection problem on a tree search. The Euclidean distance is then computed as follows

$$\left\| \begin{pmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_M \end{pmatrix} - \begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,M} \\ 0 & r_{2,2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{M,M} \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_M \end{pmatrix} \right\|^2 \quad (12)$$

The last row of Eq. (12) equals the SISO detection problem. After finding the most likely symbol for the last row, the result can be used to remove the interference of s_M from all other layers. The row $M - 1$ is then interference free and can be treated accordingly. The repetition of these steps results

in the successive interference cancellation (SIC) algorithm which solves the MIMO detection problem by splitting it into several SISO detection problems and by adding an interference cancellation step.

The SIC algorithm traverses the tree only once from top to bottom, thus, it has a low complexity. However, for a better communications performance, it is necessary to approximate Eq. (5) over many MIMO vectors s and not as a SISO detection problem. Real tree search algorithms like the depth-first sphere search or the K-best algorithm traverse the tree and compute the distances of several complete vectors. In addition to the detection steps introduced before, a node administration unit is required which stores the nodes which have been reached by the search but are not completed yet.

The main tree search based detection algorithms can thus be constructed by only five *coarse-grained algorithmic kernels*:

- The *enumeration unit EN* determines the visiting order for the children of one particular node.
- The *interference reduction unit IR* computes the interference-reduced symbols according to Eq. (10).
- The *metric computation unit MC* computes the recursive metric according to Eq. (11).
- The *node administration unit NA* stores all intermediate nodes with their results and chooses the node which is visited next in the tree.
- The *minima administration unit MA* stores the results of leaf nodes and updates the minima for the calculation of Eq. (5).

Table 1 is summarizing again which kernels are used by which algorithm. We chose soft-input soft-output sphere detection as tree search algorithm (Hochwald and ten Brink (2003)). In the computation of Eq. (5) it considers all symbol vectors s , which lie inside a sphere of radius r around the received vector y , i.e., for which $d(s) < r$. Whenever a partial metric d_i exceeds the sphere radius, the corresponding part of the tree is excluded from the search. The number of processed nodes in the tree is dynamic and depends on the current channel realization. When a large radius is chosen, sphere detection offers near-optimal communications performance at the cost of a low throughput. The throughput can be increased by reducing the radius, which, however, will lead to a degradation of the communications performance.

5 Results

Based on these observations, we designed a MIMO detector architecture consisting of five algorithmic kernels. The configuration of these kernels allows the processing of different algorithms with negligible overhead. The design was implemented on a 65 nm low power bulk CMOS library from ST

Table 1. Algorithmic kernels which allow the construction of most tree search based detection algorithms.

Algorithm / Kernels	EN	IR	MC	NA	MA	Throughput
SISO hard-output	X					1200 Mbit s ⁻¹
SISO soft-output	X		X		X	400 Mbit s ⁻¹
MIMO SIC hard-output	X	X				480 Mbit s ⁻¹
MIMO SIC soft-output	X	X	X		X	390 Mbit s ⁻¹
MIMO sphere soft-output	X	X	X	X	X	40–120 Mbit s ⁻¹

Table 2. Implementation Results: Area of the kernels after synthesis and the area of the whole detector after Place & Route.

Kernel	Area
EN	0.012 mm ²
MC	0.038 mm ²
IR	0.022 mm ²
MA	0.042 mm ²
NA	0.050 mm ²
Sum of the kernels	0.164 mm ²
After P&R, incl. I/O and control	0.254 mm ²

Microelectronics. We considered the following PVT parameters: Worst Case (WC, 1.1 V, 125°C), Nominal Case (NOM, 1.2 V, 25 °C) and Best Case (BC, 1.3 V, -40 °C). Synthesis was performed with Synopsis Design Compiler in topographical mode, Placement & Routing (P&R) with Synopsys IC Compiler. Synthesis as well as P&R were performed with the Worst Case PVT settings of the 65nm library. The final design runs with a clock frequency of 300 MHz after P&R. The implementation results for the different parts of the detector are summarized in Table 2. Throughput results for the different algorithms are shown in Table 1. The power consumption after P&R is 14.4 mW.

Two examples for the configuration are illustrated in Fig. 4: MMSE-SIC detection and soft-input soft-output sphere detection. The scheduling of the operations for the SIC algorithm is always fixed. Therefore, we discuss only the configuration of the sphere search in Fig. 4. Especially for iterative receivers, the soft-input soft-output sphere detector offers the best communications performance. During runtime, throughput can be traded off against communications performance by adjusting the sphere radius. However, due to the nature of the depth-first search, the throughput is dynamic and varies with channel conditions and over the outer iterations. The configuration in Fig. 4 utilizes all five algorithmic kernels. All operations can be performed in parallel if data is available. The *enumeration* unit determines a sequence of the best children of a node. The *metric calculation* computes the recursive metric for each node. The enumeration is stopped when a maximum number is reached or when one of the child nodes violates the radius constraint. Inter-

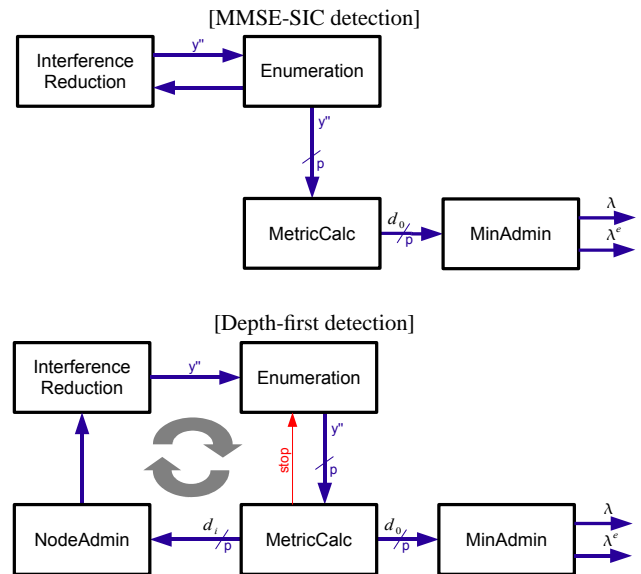


Fig. 4. The presented MIMO detector architecture consists of five algorithmic kernels, which can be configured for different detection algorithms.

mediate nodes which fulfill the radius constraint are stored together with their metrics in the *node administration* until their processing is continued. Leaf nodes are stored in the *minima administration*. Whenever possible, the *interference reduction* unit processes a node from the *node administration* and passes the result to the *enumeration* unit. This recursive loop continues until all nodes within the sphere have been computed. In contrast to other depth-first sphere decoders (e.g. Burg et al. (2005), Witte et al. (2010)) which employ a one-node-per-cycle architecture, the presented architecture computes two nodes per cycle. This is a novel approach, which doubles the throughput compared to state-of-the-art implementations.

In order to show the throughput flexibility (and thus the energy efficiency) of the implemented design, we determined the maximum achievable throughput for different channel conditions. Therefore, we simulated a MIMO-BICM system including a 64-state convolutional code as channel code, using 4 × 4 antennas with 16-QAM modulation and a codeword length of 2304 bits. We tried to reach a frame error rate

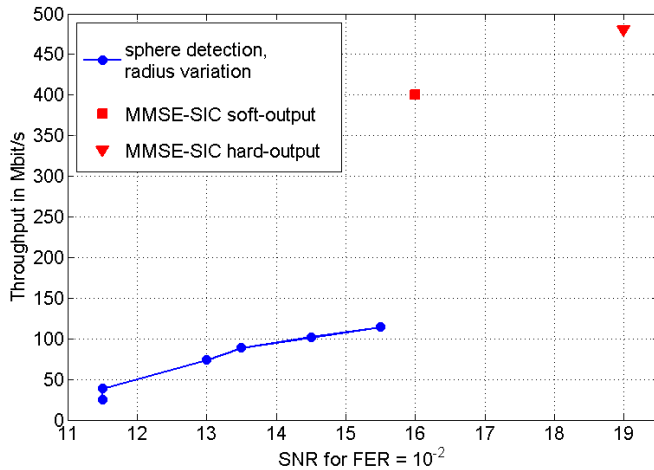


Fig. 5. Throughput vs. QoS flexibility of the presented detector architecture

(FER) of 10^{-2} for varying channel conditions, i.e. different signal-to-noise ratios (SNR). The results are shown in Fig. 5. The presented architecture covers a wide range of SNR values (almost 10 dB) and is thus able to react to different channel environments.

6 Conclusions

In this paper, we presented the first weakly-programmable MIMO detector architecture which offers just the necessary flexibility. We have demonstrated that the common tree search based MIMO detection algorithms are all constructed by the same algorithmic kernels allowing the design of a weakly-programmable architecture. In this way, the detection algorithm can be chosen and parameterized at run-time according to the current channel conditions and QoS requirements. This approach leads to a highly energy-efficient implementation.

Acknowledgements. This work was partly supported by the German Research Foundation (DFG) as part of the priority program “Dependable Embedded Systems” (SPP 1500).

References

Barbero, L. G. and Thompson, J. S.: Fixing the Complexity of the Sphere Decoder for MIMO Detection, *IEEE T. Wirel. Commun.*, 7, 2131–2142, 2008.

- Borlenghi, F., Witte, E. M., Ascheid, G., Meyr, H., and Burg, A.: A 772Mbit/s 8.81bit/nJ 90nm CMOS soft-input soft-output sphere decoder, in: *Proc. IEEE Asian Solid State Circuits Conf. (A-SSCC)*, 297–300, 2011.
- Burg, A., Borgmann, M., Wenk, M., Zellweger, M., Fichtner, W., and Bolcskei, H.: VLSI implementation of MIMO detection using the sphere decoding algorithm, *IEEE J. Solid-St. Circ.*, 40, 1566–1577, doi:10.1109/JSSC.2005.847505, 2005.
- Chen, X., Minwegen, A., Hassan, Y., Kammler, D., Li, S., Kempf, T., Chattopadhyay, A., and Ascheid, G.: FLEXDET: Flexible, Efficient Multi-Mode MIMO Detection Using Reconfigurable ASIP, in: *Field-Programmable Custom Computing Machines (FCCM)*, 2012 IEEE 20th Annual International Symposium on, 69–76, doi:10.1109/FCCM.2012.22, 2012.
- Foschini, G. J., Golden, G. D., Valenzuela, R. A., and Wolniansky, P. W.: Simplified processing for high spectral efficiency wireless communication employing multi-element arrays, *IEEE J. Sel. Area. Comm.*, 17, 1841–1852, 1999.
- Hochwald, B. and ten Brink, S.: Achieving Near-Capacity on a Multiple-Antenna Channel, *Communications, IEEE Transactions on*, 51, 389–399, doi:10.1109/TCOMM.2003.809789, 2003.
- Jafri, A. R., Karakolah, D., Baghdadi, A., and Jezequel, M.: ASIP-based flexible MMSE-IC Linear Equalizer for MIMO turbo-equalization applications, in: *Proc. DATE '09. Design, Automation & Test in Europe Conf. & Exhibition*, 1620–1625, 2009.
- Nazar, G. L., Gimmler, C., and Wehn, N.: Implementation comparisons of the QR decomposition for MIMO detection, in: *Proceedings of the 23rd symposium on Integrated circuits and system design (SBCCI '10)*, 210–214, ACM, doi:10.1145/1854153.1854204, 2010.
- Ramakrishnan, V., Witte, E., Kempf, T., Kammler, D., Ascheid, G., Leupers, R., Meyr, H., Adrat, M., and Antweiler, M.: Efficient and portable SDR waveform development: The Nucleus concept, in: *Military Communications Conference, 2009. MILCOM 2009, IEEE*, 1–7, doi:10.1109/MILCOM.2009.5379897, 2009.
- Studer, C., Fateh, S., and Seethaler, D.: ASIC Implementation of Soft-Input Soft-Output MIMO Detection Using MMSE Parallel Interference Cancellation, *IEEE J. Solid-St. Circ.*, 46, 1754–1765, doi:10.1109/JSSC.2011.2144470, 2011.
- Vikalov, H., Hassibi, B., and Kailath, T.: Iterative Decoding for MIMO Channels Via Modified Sphere Decoding, *Wireless Communications, IEEE Transactions on*, 3, 2299–2311, 2004.
- Witte, E. M., Borlenghi, F., Ascheid, G., Leupers, R., and Meyr, H.: A Scalable VLSI Architecture for Soft-Input Soft-Output Single Tree-Search Sphere Decoding, *IEEE T. Circuits II*, 57, 706–710, doi:10.1109/TCSII.2010.2056014, 2010.
- Wong, K., Tsui, C., Cheng, R., and Mow, W.: A VLSI architecture of a K-best lattice decoding algorithm for MIMO channels, in: *Proc. IEEE Int. Symp. Circuits and Systems ISCAS 2002, Vol. 3*, doi:10.1109/ISCAS.2002.1010213, 2002.
- Yu, Z., van Berkel, C., and Li, H.: A complexity adaptive channel estimator for low power, in: *Design, Automation Test in Europe Conference Exhibition (DATE)*, 2012, 1531–1536, 2012.