A Novel Reduced-Complexity Soft-Input Soft-Output MMSE MIMO Detector: Algorithm and Efficient VLSI Architecture

Dominik Auras, Rainer Leupers, Gerd H. Ascheid
Institute for Communication Technologies and Embedded Systems
RWTH Aachen University, 52056 Aachen, Germany
email: auras@ice.rwth-aachen.de

***Abstract*—A novel reduced-complexity soft-input soft-output minimum mean square error detection algorithm for MIMO systems together with an area-throughput efficient VLSI architecture is described. A detailed comparison to related work is presented. The proposed VLSI architecture of the novel algorithm represents – to the best of our knowledge – the most area-throughput efficient SISO MIMO detector ASIC reported so far, being 2.3x more efficient than its best competitor. It achieves a throughput of up to 923 Mbit/s and occupies down to half of the competitor's area while sustaining the IEEE 802.11n standard's peak data rate.**

I. Introduction

Bit-interleaved coded modulation with iterative decoding (BICM-ID) promises impressive communication performance gains [2]. To adopt this technique in current and future multi-antenna (MIMO) systems with high data rates, e.g. for the IEEE 802.11n standard [3], soft-input soft-output (SISO) minimum mean square error (MMSE) MIMO detection is an economically reasonable near-future option for VLSI implementation. Those detectors have to sustain a high throughput of at least $\Theta_c = 720$ Mbit/s in order to support the fastest 802.11n mode, which defines a peak information data rate of $\Theta_b = 600$ Mbit/s for a code rate of $r = 5/6$ and four spatial streams. The MMSE-PIC ASIC [4] is the currently most area-throughput efficient representative of this field that meets these requirements. However, it supports the fastest 802.11n mode only without iterative MIMO decoding. As a consequence, the communication performance gains can only be achieved for lower throughputs, or we have to replicate the ASIC, which entails a high area complexity. In the latter case any improvement on the replicated detector has a strengthened impact on the system complexity. Eventually, we found that an improvement factor of more than 2x is possible.

Contribution: In this work, we propose a novel reduced-complexity variant of SISO MMSE MIMO detection along with an area-throughput efficient VLSI architecture. We base our algorithm on the MMSE detector derived in [1] from the expectation propagation (EP) framework.

This work has been supported by the Ultra High-Speed Mobile Information and Communication Research Centre, RWTH Aachen University.

Outline: Section II introduces this work's context. Section III summarizes related work. In Section IV, we propose our novel IC-LMMSE algorithm and a suitable VLSI architecture. Section V summarizes the analysis of our work. The implementation results are given in Section VI. The last section concludes this paper and provides an outlook.

II. System Model

Fig. 1 depicts the considered spatial multiplexing $N_t \times N_r$ MIMO system with BICM-ID. A message $\boldsymbol{b} \in \{0,1\}^{N_b}$ is encoded with rate $r = N_b/N_c$ and interleaved, yielding the code word $\boldsymbol{c} \in \{0,1\}^{N_c}$. Let $\mathcal{X} \subset \mathbb{C}$ be a modulation alphabet with $K = \log_2 |\mathcal{X}|$ bits per symbol. The code word is partitioned into N_s subvectors $\boldsymbol{c}_n \in \{0,1\}^{KN_t}$. They are subsequently mapped to symbol vectors $\boldsymbol{x}_n \in \mathcal{X}^{N_t}$ that are transmitted independently. Assuming a frequency-flat fading channel characterized by $\boldsymbol{H}_n \in \mathbb{C}^{N_r \times N_t}$ and perfect synchronization in time and frequency, the received symbol vector at time n is $\boldsymbol{y}_n = \boldsymbol{H}_n \boldsymbol{x}_n + \boldsymbol{w}_n$ where $\boldsymbol{w}_n \in \mathbb{C}^{N_r}$ is a white Gaussian noise process with $\mathbb{E}[\boldsymbol{w}_n \boldsymbol{w}_n^H] = N_0 \boldsymbol{I}_{N_r}$.

Perfect channel knowledge at the receiver is assumed. Using iterative MIMO decoding, detector and channel decoder exchange extrinsic soft information $\boldsymbol{\lambda}^e = \boldsymbol{\lambda}^p - \boldsymbol{\lambda}^a$ in terms of log-likelihood ratios (LLRs), where $\boldsymbol{\lambda}^p$ are the detector's posterior LLRs and $\boldsymbol{\lambda}^a$ are the prior LLRs fed back from the decoder.

A. Exact MIMO Detector for BICM-ID Receiver

A BICM-ID receiver requires a SISO channel decoder and detector. The exact detector computes the posterior LLRs

$$\begin{aligned}\lambda^p_{n,t,k} &= \log \frac{p(c_{n,t,k}=0|\boldsymbol{y}_n)}{p(c_{n,t,k}=1|\boldsymbol{y}_n)} \\ &= \log \frac{\sum_{\boldsymbol{c}_n : c_{n,t,k}=0} p(\boldsymbol{y}_n|\boldsymbol{c}_n)p(\boldsymbol{c}_n)}{\sum_{\boldsymbol{c}_n : c_{n,t,k}=1} p(\boldsymbol{y}_n|\boldsymbol{c}_n)p(\boldsymbol{c}_n)}\end{aligned} \tag{1}$$

where we sum over all $\boldsymbol{c}_n$ that have the bit $c_{n,t,k}$ set to one or zero respectively. The prior distribution $p(\boldsymbol{c}_n)$ is computed from the decoder feedback $\boldsymbol{\lambda}^a$ as

$$p(\boldsymbol{c}_n) = \prod_{t=1}^{N_t} \prod_{k=1}^{K} \frac{\exp(-c_{n,t,k}\lambda^a_{n,t,k})}{1+\exp(-\lambda^a_{n,t,k})}. \tag{2}$$

In the remainder, we drop the time index n for convenience.

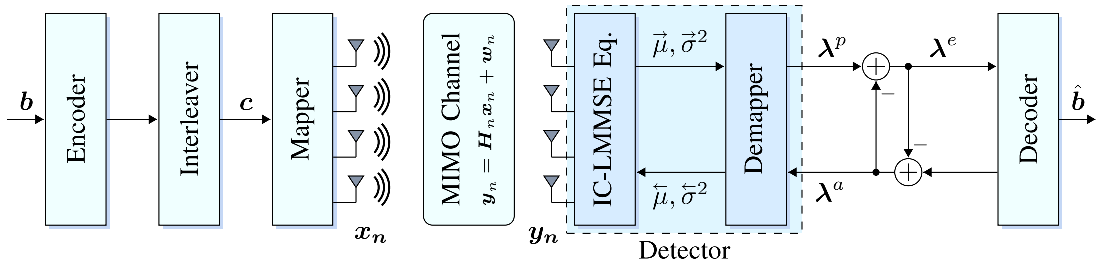

Fig. 1. MIMO BICM-ID System Model with Linear Detector

B. Linear MIMO Detection

The exact detector suffers from an exponential complexity. Linear MIMO detection trades off algorithmic performance for substantially reduced implementation complexity. The underlying idea is to separate the superimposed spatial streams and subsequently perform streamwise demapping. The separation usually encompasses an interference cancellation followed by an equalizer. We have exemplarily included an equalizer and demapper node inside the detector in Fig. 1. They exchange information in terms of Gaussian distributions describing the current belief on the data symbols, parameterized by the means $\overleftarrow{\mu}_t$, $\overrightarrow{\mu}_t$ and variances $\overleftarrow{\sigma}_t^2$, $\overrightarrow{\sigma}_t^2$.

First, the demapper node computes $\overleftarrow{\mu}_t = \mathbb{E}[x_t]$ and $\overleftarrow{\sigma}_t^2 = \mathbb{E}[x_t^2] - \overleftarrow{\mu}_t^2$ from the decoder feedback $\boldsymbol{\lambda}^a$. The equalizer uses these parameters and the channel observations $\boldsymbol{H}, \boldsymbol{y}, N_0$ to produce $(\overrightarrow{\mu}_t, \overrightarrow{\sigma}_t^2)$. Finally, the demapper computes the LLRs $\boldsymbol{\lambda}^p$ from the equalizer message and sends them to the decoder.

III. Related Work

In the following, we introduce two reduced-complexity linear SISO detection algorithms for MIMO systems together with an overview of their respective VLSI implementations: the MMSE-PIC algorithm that is described in [4], and our proposed reduced-complexity IC-LMMSE algorithm which bases on the MIMO detector presented in [1]. We selected the MMSE-PIC due to its similarity to our work and because it is the up-to-now most area-throughput efficient SISO MIMO detector ASIC. We outline the algorithmic and architectural differences in detail in Section V-B. The descriptions are kept somewhat similar to ease the comparison.

A. Reduced-Complexity MMSE-PIC Algorithm

We summarize the algorithm detailed in [4] using our notation. The algorithm assumes a gray-mapped constellation as e.g. defined in the IEEE 802.11n standard. The detection is performed in six steps.

1) Gram matrix & matched filter: Compute the Gram matrix $\boldsymbol{R} = \boldsymbol{H}^H\boldsymbol{H}$ and the matched filter output $\boldsymbol{y}^{\text{mf}} = \boldsymbol{H}^H\boldsymbol{y}$.

2) Symbol statistics: Map the distribution of the code bits $\boldsymbol{\lambda}^a$ to streamwise Gaussian distributions $\mathcal{CN}(\overleftarrow{\mu}_t, \overleftarrow{\sigma}_t^2)$. The detector computes $\overleftarrow{\mu}_t = \mathbb{E}[x_t]$ and $\overleftarrow{\sigma}_t^2 = \mathbb{E}[x_t^2] - \overleftarrow{\mu}_t^2$ for every spatial stream $t = 1 \dots N_t$ according to the method in [5]. To this end, it converts the LLRs to bit probabilities using

$$p_{t,k} = \frac{1}{2}\left(1 + \tanh\left(\frac{\lambda_{t,k}^a}{2}\right)\right). \tag{3}$$

The lower subtraction in Fig. 1 in the path from decoder to detector is omitted, thus we use posterior LLRs.

3) Parallel interference cancellation: Compute

$$\boldsymbol{y}_t^{\text{mf}} = \boldsymbol{y}^{\text{mf}} - \sum_{j \neq t} \boldsymbol{r}_j \overleftarrow{\mu}_j \tag{4}$$

for $t = 1 \dots N_t$ where $\boldsymbol{r}_j$ denotes the j-th column of $\boldsymbol{R}$.

4) MMSE filter matrix: Compute the matrix

$$\widetilde{\boldsymbol{C}}_{\boldsymbol{y}} = \boldsymbol{R}\boldsymbol{C}_{\boldsymbol{x}} + N_0\boldsymbol{I}_{N_t} \tag{5}$$

with $\boldsymbol{C}_{\boldsymbol{x}} = \operatorname{diag}(\overleftarrow{\sigma}_1^2, \dots, \overleftarrow{\sigma}_{N_t}^2)$ then decompose $\widetilde{\boldsymbol{C}}_{\boldsymbol{y}} = \boldsymbol{L}\boldsymbol{U}$ into lower and upper triangular matrices $\boldsymbol{L} = [l_{ij}] \in \mathbb{C}^{N_t \times N_t}$ and $\boldsymbol{U} = [u_{ij}] \in \mathbb{C}^{N_t \times N_t}$ respectively, where $\boldsymbol{L}$ has a unity diagonal $l_{ii} = 1$. Subsequently, invert the triangular matrices using forward and back substitution to obtain

$$\widetilde{\boldsymbol{G}}^H = \boldsymbol{U}^{-1}\boldsymbol{L}^{-1} \tag{6}$$

by solving $\boldsymbol{L}\boldsymbol{U}\widetilde{\boldsymbol{G}}^H = \boldsymbol{I}_{N_t}$ for $\widetilde{\boldsymbol{G}}^H$. The detector reuses the reciprocals u_{ii}^{-1} computed for the decomposition in the back substitution step. We use $\widetilde{\boldsymbol{C}}_{\boldsymbol{y}}$ and $\widetilde{\boldsymbol{G}}$ here to distinguish from the different $\boldsymbol{C}_{\boldsymbol{y}}$ and $\boldsymbol{G}$ used in our proposed algorithm.

5) Filtering: Compute the bias terms $\tilde{\mu}_t = \tilde{\boldsymbol{g}}_t^H\boldsymbol{r}_t$ for $t = 1 \dots N_t$ where $\tilde{\boldsymbol{g}}_t^H$ denotes the t-th row of $\widetilde{\boldsymbol{G}}^H$. Using the reciprocals $\tilde{\mu}_t^{-1} = 1/\tilde{\mu}_t$, update the symbol estimate

$$\overrightarrow{\mu}_t = \tilde{\mu}_t^{-1}\tilde{\boldsymbol{g}}_t^H\boldsymbol{y}_t^{\text{mf}} \tag{7}$$

and compute the per-stream SNR

$$\rho_t = \tilde{\mu}_t/(1 - \overleftarrow{\sigma}_t^2\tilde{\mu}_t) = 1/\overrightarrow{\sigma}_t^2. \tag{8}$$

6) Demapping: Compute the extrinsic LLRs

$$\lambda_{t,k}^e = \rho_t\left(\min_{x \in \mathcal{X}_k^0} |x - \overrightarrow{\mu}_t|^2 - \min_{x \in \mathcal{X}_k^1} |x - \overrightarrow{\mu}_t|^2\right) \tag{9}$$

where $\mathcal{X}_k^0$ and $\mathcal{X}_k^1$ are subsets of $\mathcal{X}$ with the k-th bit set to zero or one respectively. The difference of the min-terms is computed with piece-wise linear functions [6]. Note that the prior LLRs are not used at all in this step.

In summary, the algorithm maps the posterior decoder LLRs to Gaussian distributions, performs per-stream interference cancellation, per-stream equalization and finally per-stream max-log demapping without priors.

TABLE I
MMSE-PIC DATAPATH CONFIGURATION

Processing Unit	Add.	Mult.	Shift	LUT	Recip.	Mem [kBit]	Area [kGE]
$\boldsymbol{R}$ and $\boldsymbol{y}^{\text{mf}}$	16	16	–	–	–	2.09	50.4
$\overleftarrow{\mu}_t$, $\overleftarrow{\sigma}_t^2$ and $\widetilde{\boldsymbol{C}}_{\boldsymbol{y}}$	8	8	7	3	–	0.44	34.4
PIC part 1 & 2	8	4	–	–	–	3.36	38.4
LUD & $\boldsymbol{L}^{-1}$	6	10	2	1	1	1.41	68.1
Back Subst.	11	10	6	–	–	1.49	70.2
Filter & SNR	12	12	4	1	1	2.17	112.4
LLR comp.	3	3	9	–	–	0.58	10.3
Total	64	63	28	5	2	14.52	384.2

B. MMSE-PIC ASIC

The ASIC implementation [4] of the above algorithm is organized into a coarse grained pipeline of six stages with eight processing units (PUs). The pipeline is shifted every 18 clock cycles. The data is forwarded to the next pipeline stage in one clock cycle termed as "exchange cycle". Every PU consists of several pipelined arithmetic units (AUs), a register-based data memory, an interconnection network and a local state machine. The AU set contains (complex-valued) adders, multipliers, arithmetic shifters, lookup tables and a custom Newton-Raphson based reciprocal unit. The total processing latency equals 108 clock cycles. The PU pipeline has a global control unit. Tbl. I summarizes the data path configuration alongside with the area distribution of the ASIC.

A great deal of the architecture is dedicated to improve the numerical stability, suggested by the total of 28 arithmetic shifters. The general strategy is to delay rescaling as much as possible, in order to benefit from normalized operands. For example, the reciprocal unit first normalizes its input to the range $[0.5, 1)$, thus the result $1/x$ is in $(1, 2]$. Subsequent operations involving $1/x$, e.g. as multiplicand, may benefit from the confined dynamic range, e.g. through smaller word lengths. This can be seen as a locally limited floating point arithmetic. Similarly, the ASIC shifts $\widetilde{\boldsymbol{C}}_{\boldsymbol{y}}$ column-wise such that the diagonal entries are within $[0.5, 1)$. Only when computing the new symbol estimates and the per-stream SNRs, rescaling is performed.

The ASIC was fabricated in 90nm CMOS technology. It achieves a maximum clock frequency of 568 MHz and can sustain a throughput of $\Theta_c = 757$ Mbit/s. Thus it supports the fastest 802.11n mode with margin. In total, the ASIC[1] occupies an area of 384.2 kGE.

IV. PROPOSED ALGORITHM & VLSI ARCHITECTURE

This section describes the essence of our proposed algorithm and its corresponding VLSI implementation. It bases on the MIMO detector[2] derived in [1].

[1]We omit the area of the input/output interface of the chip itself and assume that the VLSI architecture would be integrated into a system on chip.

[2]We do not perform iterations between demapper and equalizer nodes.

A. Reduced-Complexity IC-LMMSE MIMO Detection

We assume that a gray-mapped constellation is used, which holds true for the considered IEEE 802.11n standard [3]. Our algorithm encompasses six steps in total.

1) Symbol statistics: Map the LLRs $\boldsymbol{\lambda}^a$ to $\overleftarrow{\mu}_t$ and $\overleftarrow{\sigma}_t^2$. We convert the decoder's posterior LLRs, i.e. the lower subtraction in Fig. 1 in the path from decoder to detector is omitted, to bit probabilities according to

$$p_{t,k} = \text{logistic}(\lambda_{t,k}^a) = \frac{1}{1 + \exp(-\lambda_{t,k}^a)}. \tag{10}$$

Then we compute the terms $\mathbb{E}[x_t]$ and $\mathbb{E}\left[x_t^2\right]$ as in [5]. To improve the numerical stability of the next steps, we limit the symbol variances to a lower bound $\overleftarrow{\sigma}_{\text{min}}^2$.

2) Covariance matrix: Compute the covariance matrix

$$\boldsymbol{C}_{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{C}_{\boldsymbol{x}}\boldsymbol{H}^H + N_0\boldsymbol{I}_{N_r} \tag{11}$$

with $\boldsymbol{C}_{\boldsymbol{x}} = \text{diag}(\overleftarrow{\sigma}_1^2, \ldots, \overleftarrow{\sigma}_{N_t}^2)$, whereby we limit the noise density to a lower bound $N_{0,\text{min}}$. Note that this matrix is hermitian positive definite (HPD), which means that $\boldsymbol{x}^H\boldsymbol{C}_{\boldsymbol{y}}\boldsymbol{x} > 0 \,\forall\, \boldsymbol{x} \neq 0$ and $\boldsymbol{C}_{\boldsymbol{y}}^H = \boldsymbol{C}_{\boldsymbol{y}}$ holds.

3) Interference cancellation: Compute the single vector

$$\boldsymbol{y}^{\text{ic}} = \boldsymbol{y} - \sum_{t=1}^{N_t} \boldsymbol{h}_t \overleftarrow{\mu}_t \tag{12}$$

where $\boldsymbol{h}_t$ denotes the t-th column of $\boldsymbol{H}$.

4) MMSE filter matrix: Decompose $\boldsymbol{C}_{\boldsymbol{y}}$ according to

$$\boldsymbol{C}_{\boldsymbol{y}} = \boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^H \tag{13}$$

where $\boldsymbol{L} = [l_{ij}] \in \mathbb{C}^{N_r \times N_r}$ is a lower triangular matrix $l_{ij} = 0 \,\forall\, i < j$ with unity diagonal $l_{ii} = 1$. The diagonal matrix $\boldsymbol{D} = \text{diag}(d_t) \in \mathbb{R}^{N_r \times N_r}$ has real-valued positive entries $d_t > 0$. We limit the elements d_t to a lower bound d_{min} for numerical reasons. Then, find $\boldsymbol{G} = \boldsymbol{C}_{\boldsymbol{y}}^{-1}\boldsymbol{H}$ by solving

$$\boldsymbol{L}\boldsymbol{D}\boldsymbol{L}^H\boldsymbol{G} = \boldsymbol{H} \tag{14}$$

for $\boldsymbol{G}$ using forward and back substitution to compute

$$\begin{aligned} \boldsymbol{L}\boldsymbol{G}' &= \boldsymbol{H} \\ \boldsymbol{L}^H\boldsymbol{G} &= \boldsymbol{D}^{-1}\boldsymbol{G}' \end{aligned} \tag{15}$$

where the inverse $\boldsymbol{D}^{-1}$ is a byproduct of the decomposition that can be reused in the back substitution.

5) Filtering: Compute the bias terms $\tilde{\mu}_t = \boldsymbol{g}_t^H\boldsymbol{h}_t$ where $\boldsymbol{g}_t$ is the t-th column of $\boldsymbol{G}$, limit the bias to a minimum $\tilde{\mu}_{\text{min}}$, and compute the reciprocal $\tilde{\mu}_t^{-1} = 1/\tilde{\mu}_t$. Then produce updated symbol statistics according to

$$\begin{aligned} \vec{\mu}_t &= \overleftarrow{\mu}_t + \tilde{\mu}_t^{-1}\boldsymbol{g}_t^H\boldsymbol{y}^{\text{ic}} \\ \vec{\sigma}_t^2 &= \tilde{\mu}_t^{-1} - \overleftarrow{\sigma}_t^2 \end{aligned} \tag{16}$$

for every spatial stream. We apply another minimum $\vec{\sigma}_{\text{min}}^2$ to the variances and subsequently invert them to obtain the per-stream SNRs $\rho_t = 1/\vec{\sigma}_t^2$.

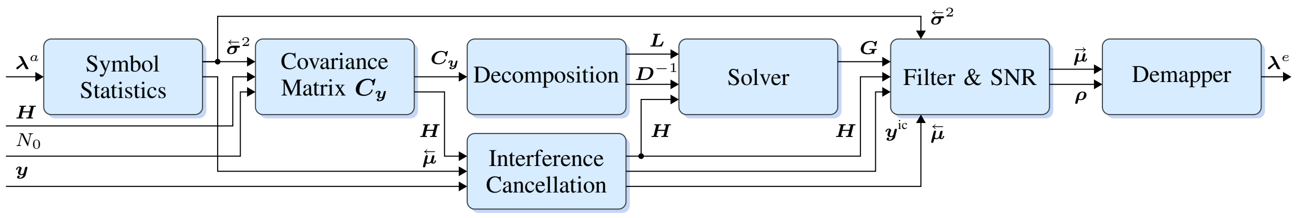

Fig. 2. Architecture Overview of the Reduced-Complexity IC-LMMSE MIMO Detector

6) Demapping: Using the max-log approximation and omitting the priors, demap the previously computed symbol statistics according to

$$\lambda_{t,k}^e = \rho_t \left(\min_{x \in \mathcal{X}_k^0} |x - \vec{\mu}_t|^2 - \min_{x \in \mathcal{X}_k^1} |x - \vec{\mu}_t|^2 \right) \quad (17)$$

where $\mathcal{X}_k^0$ and $\mathcal{X}_k^1$ are subsets of $\mathcal{X}$ with the k-th bit set to zero or one respectively. The difference of the min-terms is computed with piece-wise linear functions [6].

In summary, the decoder's posterior LLRs are mapped to streamwise Gaussian distributions, one interference cancellation step is performed, followed by per-stream equalization and streamwise demapping without considering prior LLRs. Note that the algorithmic differences of the two are highlighted in Section V-B.

B. IC-LMMSE ASIC

Our architecture computes $\boldsymbol{\lambda}^e$ from $\boldsymbol{H}$, N_0, $\boldsymbol{y}$ and $\boldsymbol{\lambda}^a$. As depicted in Fig. 2, it is organized into a pipeline of six stages that take 18 clock cycles per pipeline cycle, similar to [4]. It encompasses in total seven processing units (PUs). For $N_t = 4$ antennas, 64-QAM ($K = 6$), $r = 5/6$ and a reasonable clock frequency of $f_{clk} = 540\,\text{MHz}$, we achieve a data rate of $\Theta_b = \frac{rN_tK}{18} f_{clk} = 600\,\text{Mbit/s}$, which meets the maximum throughput requirement of the IEEE 802.11n standard.

We use one clock cycle to exchange data between the stages. Where required, forwarding buffers are in place, e.g. for $\boldsymbol{H}$. All data is kept in registers. We manually allocated the arithmetic units and devised operation schedules for each PU. The design shortens the critical path where possible, e.g. it uses pipelined arithmetic units. Tbl. II summarizes the data path configuration.

TABLE II
IC-LMMSE ASIC DATAPATH CONFIGURATION

Processing Unit	Add.	Mult.	Shift	LUT	Recip.	Mem [bit]
Symbol statistics	9	2	–	2	–	263
Covariance matrix	8	12	–	–	–	718
Decomposition	12	3	2	1	1	787
Solver	16	16	–	–	–	1791
IC step	4	4	–	–	–	585
Filter & SNR	9	9	2	1	1	2005
Demapper	14	2	–	–	–	399
Forwarding	–	–	–	–	–	1227
Total	72	48	4	4	2	7775

The algorithm is numerically stable. It allows an implementation that exploits the data symmetries due to the HPD property of $\boldsymbol{C_y}$, which reduces the number of real-valued operations and storage requirements. The several lower bounds applied effectively limit the dynamic range of the resulting matrix inverse.

Note that in Section V-B we provide a comparison of our overall architectural design to the MMSE-PIC architecture.

In the remainder, we shortly describe the architectural details that are not implicitly given by the algorithm specification. Since most stage implementations are straight-forward and similar, we depict only the Filter & SNR unit, which appeared to be the most interesting one for us.

1) Symbol statistics: Two parallel identical data paths, one per complex dimension, implement the method in [5] to compute $\overleftarrow{\mu}_t$ and $\overleftarrow{\sigma}_t^2$. We use a lookup table of 32 10-bit entries for $\text{logistic}(\lambda)$, $\lambda \in [0, 8)$. For λ outside $[0, 8)$, we use the symmetry around 0 and the fact that $\text{logistic}(\lambda) \approx 1$ for $\lambda \geq 8$. The PU is runtime configurable to support 4-, 16- and 64-QAM.

2) Covariance matrix: We use a weighted inner product unit, implementing $\text{Re}/\text{Im}\{\boldsymbol{h}_t^H \boldsymbol{C_x} \boldsymbol{h}_t\}$, with three pipeline stages to compute one complex dimension of $\boldsymbol{H}^H \boldsymbol{C_x} \boldsymbol{H}$ per cycle. An adder for the noise density N_0 finalizes the diagonal. In total only 16 values, i.e. the upper triangular part, have to be computed due to $\boldsymbol{C_y} = \boldsymbol{C_y}^H$.

3) Decomposition: The data path of the decomposition PU comprises three multipliers, a few adders and multiplexers, and a reciprocal unit with three pipeline stages. This unit computes $1/x$ as follows. We normalize the input x to the range $[1, 2)$ by scaling it with 2^α, $\alpha \in \mathbb{Z}$. Note that x is always positive here. The scaled input is represented with only 13 bits. An initial approximation $\tilde{x}_0 \approx 1/x$ is found using a lookup table of 32 6-bit entries for $1/x - 0.5$. One Newton-Raphson

Algorithm 1: LDL Decomposition

```
  Require: $\boldsymbol{A} = \boldsymbol{C_y} \in \mathbb{C}^{N_r \times N_r}$ is HPD
1 for $j = 1 \dots N_r$ do
2     $d_j = \text{Re}\{a_{jj} - \sum_{k=1}^{j-1} l_{jk} t_{jk}^*\}$
3     $d_j^{-1} = 1/d_j$
4     for $i = j+1 \dots N_r$ do
5         $t_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik} t_{jk}^*$
6         $l_{ij} = d_j^{-1} t_{ij}$
7     end
8 end
```

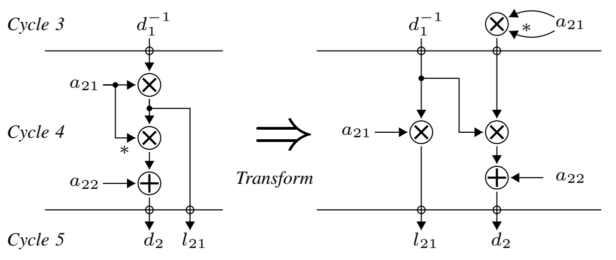

Fig. 3. Schedule Transformation

iteration $\tilde{x}_1 = 2\tilde{x}_0 - x\tilde{x}_0^2$ increases the result's precision to about 12 bits [7]. We rescale the reciprocal before storing it.

The LDL decomposition scheduled on this data path is given as Alg. 1. The complex products are computed in a 3-product form, i.e. we trade off additions for multiplications using $(a+jb)(c+jd) = ac-bd+j((a+b)(c+d)-ac-bd)$. Furthermore, we have transformed the schedule slightly to fit it into 17 clock cycles, keeping the exchange cycle free of operations. The computation of the four d_i (for $M = 4$) and the four $1/d_i$, which constitutes the critical path of the algorithm, is modified to avoid two chained multiplications in one clock cycle. Since the four reciprocals require twelve cycles to finish, we have five cycles to compute d_i. We reserve one cycle for d_2 and one for d_3, and three for d_4. The computation of $d_2 = \mathrm{Re}\{a_{22} - l_{21}t_{21}^*\}$ with $t_{21} = a_{21}$ requires two chained multiplications in one cycle, namely $d_1^{-1}a_{21}$ and $l_{21}a_{21}^*$, see the left part of Fig. 3. By reformulating the last product according to

$$\begin{aligned}\mathrm{Re}\{l_{21}a_{21}^*\} &= \mathrm{Re}\{(d_1^{-1}a_{21})a_{21}^*\}\\ &= d_1^{-1}\mathrm{Re}\{a_{21}a_{21}^*\}\end{aligned} \tag{18}$$

we obtain the transformed schedule on the right side of Fig. 3. This adds a new multiplication, but avoids to compute two dependent in one cycle. We transformed a total of four products in the presented way.

4) Solver: The solver PU comprises four complex multiply-accumulate (CMAC) units with two pipeline stages. It exploits the stream-level parallelism by computing all four columns of $\boldsymbol{G}$ in parallel. Per column, the CMAC operations are done sequentially. We insert three pipeline stall cycles due to the data dependencies. The input registers for $\boldsymbol{L}$ and $\boldsymbol{D}^{-1}$ are shared among the four CMAC units. For the operation $\boldsymbol{D}^{-1}\boldsymbol{G}$, the adders of the CMAC are bypassed. We initialize the accumulator registers with $\boldsymbol{H}$.

5) Interference cancellation: This PU uses a single CMAC unit with two pipeline stages to compute $\boldsymbol{y}^{\text{ic}} = \boldsymbol{y} - \boldsymbol{H}\overleftarrow{\boldsymbol{\mu}}$ in 17 cycles. The accumulator register is initialized with $\boldsymbol{y}$.

6) Filter & SNR: The PU, depicted in Fig. 4, uses amongst others an 8-operand inner product unit, implementing $\mathrm{Re}/\mathrm{Im}\{\boldsymbol{g}_t^H\boldsymbol{x}\}$, with three pipeline stages. The inner product unit computes one complex dimension per cycle. A Newton-Raphson based reciprocal unit, similar to the one used in the decomposition stage, with three pipeline stages computes $\tilde{\mu}_t^{-1}$ and $(\vec{\sigma}_t^2)^{-1}$. Due to lower precision requirements, the normalized unsigned input has a word length of only 9 bit. We use a 16-entries lookup table with 5 bits per entry for

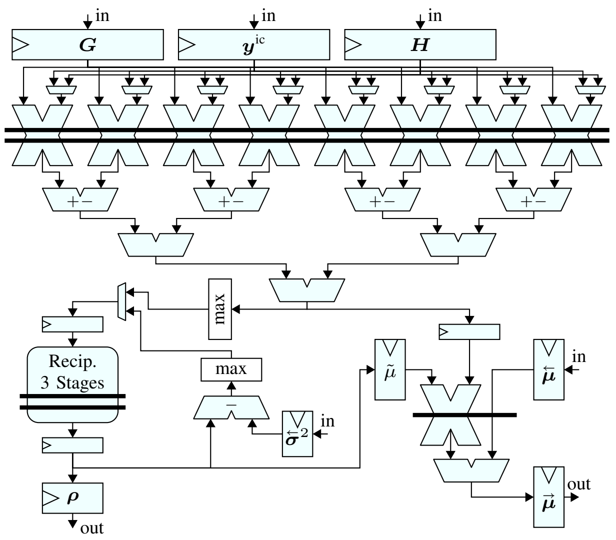

Fig. 4. Filter & SNR Stage

the initial approximation of $1/x$. Furthermore, a real-valued multiply-accumulate unit with two pipeline stages computes one complex dimension of $\overleftarrow{\mu}_t + \tilde{\mu}_t^{-1}\boldsymbol{g}_t^H\boldsymbol{y}^{\text{ic}}$ per cycle.

7) Demapper: The demapper PU implements the piece-wise linear functions described in [6]. It has two parallel identical data paths, each responsible for the bits associated with the real and imaginary symbol parts respectively. The LLRs scaled by ρ_t with one multiplier per data path are truncated and saturated to 9 bits. At runtime, we can configure the PU for 4-, 16- and 64-QAM.

V. Analysis

A. Conditions and Assumptions

A 40 MHz 802.11n-like scenario similar to [4] is considered assuming a 4×4 MIMO system with gray-mapped 4-/16-/64-QAM modulation, max-log demapping, a spatially uncorrelated Rayleigh channel and perfect channel knowledge. We use a rate-1/2 tail-biting convolutional code with polynomials $[133, 171]_8$, and a max-log BCJR decoder. A frame consists of 864 information bits. The average signal-to-noise ratio (SNR) per receive antenna is defined as $\text{SNR} = \mathbb{E}[\|\boldsymbol{Hx}\|^2]/(N_rN_0)$. We determined the required word lengths to obtain an SNR loss of ≤ 0.1dB compared to the floating-point model at a frame error rate of 10%. All simulations have been performed with 10^5 frames.

B. Comparison to MMSE-PIC

In this subsection, we highlight differences in the algorithm and architecture design between our work and its closest related work, the MMSE-PIC [4]. The most prominent difference is the filter matrix, which is defined as

$$\boldsymbol{G}^H = \left(\boldsymbol{H}^H\boldsymbol{H}\boldsymbol{C}_{\boldsymbol{x}} + N_0\boldsymbol{I}_{N_t}\right)^{-1}\boldsymbol{H}^H \tag{19}$$

for the MMSE-PIC algorithm while we use

$$\boldsymbol{G} = \left(\boldsymbol{H}\boldsymbol{C}_{\boldsymbol{x}}\boldsymbol{H}^H + N_0\boldsymbol{I}_{N_r}\right)^{-1}\boldsymbol{H}. \tag{20}$$

This results in further differences. The MMSE-PIC splits the filter matrix according to

$$\boldsymbol{G}^H = \widetilde{\boldsymbol{G}}^H \boldsymbol{H}^H \tag{21}$$

and because $\boldsymbol{G}^H \boldsymbol{y} = \widetilde{\boldsymbol{G}}^H \boldsymbol{H}^H \boldsymbol{y}$, it computes

$$\boldsymbol{y}^{\mathrm{mf}} = \boldsymbol{H}^H \boldsymbol{y} \tag{22}$$

and the gram matrix $\boldsymbol{R} = \boldsymbol{H}^H \boldsymbol{H}$, then discards $\boldsymbol{H}$. Subsequently only $\boldsymbol{y}^{\mathrm{mf}}$ and $\boldsymbol{R}$ are used, e.g. in the IC step (4). Based on the LU decomposition

$$\widetilde{\boldsymbol{C}}_{\boldsymbol{y}} = \boldsymbol{R}\boldsymbol{C}_{\boldsymbol{x}} + N_0 \boldsymbol{I}_{N_t} = \boldsymbol{L}\boldsymbol{U} \tag{23}$$

it inverts $\widetilde{\boldsymbol{C}}_{\boldsymbol{y}}$ to obtain

$$\widetilde{\boldsymbol{G}}^H = \boldsymbol{U}^{-1}\boldsymbol{L}^{-1}. \tag{24}$$

The forward and back substitution steps that are used to solve for $\widetilde{\boldsymbol{G}}^H$ in $\boldsymbol{L}\boldsymbol{U}\widetilde{\boldsymbol{G}}^H = \boldsymbol{I}$ are simple due to the identity matrix on the right hand side.

On the other hand, the IC-LMMSE algorithm's choice for $\boldsymbol{G}$ entails that the constituent matrix $\boldsymbol{C}_{\boldsymbol{y}}$ is hermitian, i.e. $\boldsymbol{C}_{\boldsymbol{y}}^H = \boldsymbol{C}_{\boldsymbol{y}}$. This allows to use the LDL decomposition, which has good numerical stability and lower computational complexity than the LU decomposition. However, scaling in presence of the LDL decomposition is more complex, as $\boldsymbol{C}_{\boldsymbol{y}}^H = \boldsymbol{C}_{\boldsymbol{y}}$ needs to be kept. The MMSE-PIC can scale $\widetilde{\boldsymbol{C}}_{\boldsymbol{y}}$ column-wise, since the LUD does not depend on the symmetry. But as we found, the IC-LMMSE algorithm performs well without scaling in the considered scenarios.

This might also be a result of the regularization using the five lower bounds $\overleftarrow{\sigma}^2_{\min}$, $\vec{\sigma}^2_{\min}$, $N_{0,\min}$, $d_{\min}$ and $\tilde{\mu}_{\min}$. The MMSE-PIC only uses a single lower bound $N_{0,\min}$ for the noise density.

The previously mentioned simplified forward and back substitution steps in case of the MMSE-PIC algorithm also led to a slightly different partitioning into processing units. The MMSE-PIC groups the LU decomposition with the forward substitution, and performs the back substitution in a separate unit. The IC-LMMSE ASIC groups forward and back substitution into the solver stage.

Another difference resides in the IC step. The MMSE-PIC, as the name implies, performs parallel interference cancellation. It computes one $\boldsymbol{y}^{\mathrm{mf}}_t = \boldsymbol{y}^{\mathrm{mf}} - \sum_{j \neq t} \boldsymbol{r}_j \overleftarrow{\mu}_j$ per spatial stream, where the sum over $j \neq t$ excludes the t-th element. The IC-LMMSE computes only a single $\boldsymbol{y}^{\mathrm{ic}} = \boldsymbol{y} - \sum_j \boldsymbol{h}_j \overleftarrow{\mu}_j$ which can be interpreted as a residual channel observation that cannot be explained by the current symbol hypothesis. This leads to different equations for $\vec{\mu}_t$. The current hypothesis $\overleftarrow{\mu}_t$ is already contained in $\boldsymbol{y}^{\mathrm{mf}}_t$, which gives $\vec{\mu}_t = \tilde{\mu}_t^{-1} \tilde{\boldsymbol{g}}_t^H \boldsymbol{y}^{\mathrm{mf}}_t$. In case of the IC-LMMSE algorithm, the term $\tilde{\mu}_t^{-1} \boldsymbol{g}_t^H \boldsymbol{y}^{\mathrm{ic}}$ is the innovation gained from the residual, thus we have $\vec{\mu}_t = \overleftarrow{\mu}_t + \tilde{\mu}_t^{-1} \boldsymbol{g}_t^H \boldsymbol{y}^{\mathrm{mf}}$.

A commonality of the two ASICs are the similar design criteria. Both are designed to sustain the $\Theta_b = 600$ Mbit/s information rate of the IEEE 802.11n standard. They are organized as coarse-grained pipelines. Each pipeline cycle takes 18 clock cycles. Data is exchanged during one dedicated cycle in both architectures. Both use a Newton-Raphson based reciprocal unit design. The mapping and demapping stages are similar, except for the use of the $\mathrm{logistic}$ function in the IC-LMMSE, versus the $\tanh$ function in the MMSE-PIC. Considering the interface between units, the MMSE-PIC uses a global control unit, while our architecture has only local control units and bases on handshaking together with data flow tokens. Both architectures use local state machines per PU to steer the operation schedules.

C. Algorithmic Performance

The fixed-point ASIC implementation[3] achieves an error-rate performance close to the double-precision floating-point model. The frame error rate (FER) over SNR for 64-QAM is plotted in Fig. 5. The SNR loss is measured at an FER of 10%. We found about 0.2 dB loss due to the algorithmic simplifications (max-log, omit priors). The additional loss due to finite word length effects amounts to less than 0.1 dB. We made similar observations for all three supported modulation schemes over zero to three detector-decoder iterations. Note that the zero-th iteration matches soft-output MMSE detection.

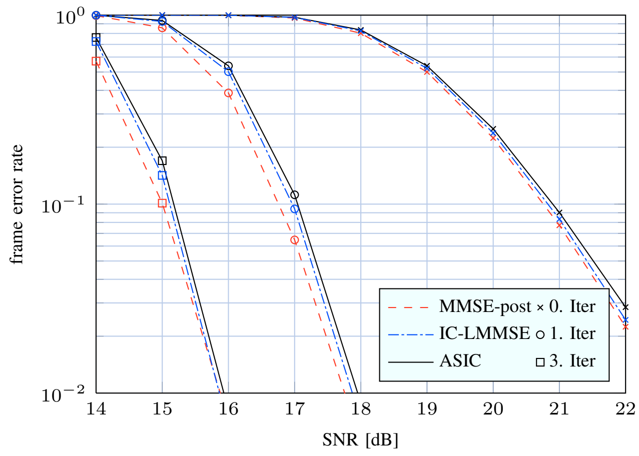

Fig. 5. Frame Error Rate over SNR for 64-QAM. *MMSE-post* is the original MMSE detector with posterior feedback. *IC-LMMSE* is the proposed algorithm with floating-point arithmetic. *ASIC* is the implementation including finite word length effects.

VI. Implementation Results

A. ASIC Implementation

The proposed architecture has been synthesized with Synopsys Design Compiler G-2012.06 in topographical mode using a 90nm standard-performance CMOS library. We consider three different design points at different clock frequencies. Their

[3]The fixed-point formats $[I.F]$, with I bits before and F bits after the point, are as follows, whereupon formats for complex values apply to the real and imaginary part respectively, and u$[I.F]$ denotes an unsigned format. We have: $\boldsymbol{\lambda}^a$ [5.2], $\boldsymbol{\lambda}^e$ [5.4], $\boldsymbol{H}$ [3.8], $\boldsymbol{y}$ [7.4] (same for $\boldsymbol{y}^{\mathrm{ic}}$), $\overleftarrow{\mu}_t$ [4.2], $\overleftarrow{\sigma}^2_t$ u[7.4], $\vec{\mu}_t$ [6.6], $\vec{\sigma}^2_t$ u[8.4], ρ_t u[4.8], $\boldsymbol{G}$ [2.18], $\boldsymbol{C}_{\boldsymbol{y}}$ [11.6], $\boldsymbol{L}$ [4.11], $\boldsymbol{D}^{-1}$ u[0.18], t_{ij} [10.6], scaled d_i u[1.12], $\tilde{\mu}_t$ u[3.12], $\tilde{\mu}_t^{-1}$ u[8.6]. We use these lower bounds: $\overleftarrow{\sigma}^2_{\min} = 2^{-4}$, $\vec{\sigma}^2_{\min} = 2^{-4}$, $d_{\min} = 1$, $\tilde{\mu}_{\min} = 2^{-8}$ and $N_{0,\min} = 1$.

TABLE III
IC-LMMSE ASIC IMPLEMENTATION RESULTS

Processing Unit	Smallest	Efficient	Fastest
Symbol statistics [kGE]	6.4	6.7	7.5
Covariance matrix [kGE]	22.2	23.0	25.9
Decomposition [kGE]	20.2	22.3	28.3
Solver [kGE]	69.3	77.0	105.1
IC step [kGE]	6.0	6.2	6.6
Filter & SNR [kGE]	27.5	28.3	33.8
Demapper [kGE]	6.9	7.0	7.9
Forwarding [kGE]	7.8	7.8	7.8
Total area [kGE]	166.3	178.3	222.9
Clock frequency [MHz]	540	625	692
Throughput Θ_c [Mbit/s]	720	833	923
Efficiency [Mbit/s/kGE]	4.33	4.67	4.14

TABLE IV
COMPARISON TO OTHER SISO DETECTORS

	This work	[4]	[8]	[9]
Number of antennas	4×4	4×4	$\leq 4 \times 4$	$\leq 4 \times 4$
Throughput Θ_c [Mbit/s]	833	757	avg. 34.2[e]	avg. 51.1[d]
Preprocessing area [kGE]	178.3	384.2[a]	–[b]	–[c]
Detection area [kGE]			265	175
Efficiency [Mbit/s/kGE]	4.67	1.97	0.13	0.29

[a] Area for chip IO interface excluded
[b] Area for optional initial detection not included
[c] Area for required QRD not included
[d] We assume 100 visited nodes per symbol vector [4].
[e] We assume $N_s = 64, N_{gs} = 8, N_p = 8$ [8].

per-unit area breakdown is given in Tbl. III for three different design points: the smallest meeting the design constraints, the fastest possible and the most area-throughput efficient one.

At the target frequency of 540 MHz, the ASIC occupies a total area of about 166.3 kGE. With a throughput of $\Theta_c = 720$ Mbit/s, this design point is the smallest variant that supports the fastest 802.11n mode of $\Theta_b = 600$ Mbit/s.

The ASIC's maximum achievable clock frequency is 692 MHz. This variant sustains a $\Theta_c = 923$ Mbit/s, while occupying about 222.9 kGE. This is about 34% larger than the requirements-achieving design.

The highest area-throughput efficiency of ~4.67 Mbit/s/kGE is attained at a clock frequency of 625 MHz. This efficient variant can sustain $\Theta_c = 833$ Mbit/s and occupies only 7.2% more area than the smallest design, namely 178.3 kGE. We believe that this is a reasonable design point, because it leaves some timing margin for the surrounding system parts at the highest throughput.

B. Comparison

We compare our ASIC in terms of area-throughput efficiency to other reported circuits in Tbl. IV. To the best of our knowledge, the IC-LMMSE ASIC is the most efficient SISO MIMO detector reported so far. At its highest efficiency of 4.67 Mbit/s/kGE, it is 2.3x more efficient than its current best competitor, the MMSE-PIC [4]. With up to 923 Mbit/s, it is faster than the MMSE-PIC, and with down to 166.3 kGE, it requires less than half the area of the MMSE-PIC.

It also outperforms the MCMC detector [8] and the sphere decoder [9] in terms of efficiency by an order of magnitude. However, the design goals are of course different. Our ASIC has been designed to sustain a constant throughput at a reasonable complexity while enabling iterative MIMO decoding. The sphere decoder strives for optimal detection performance, but is penalized by its high complexity and strongly varying run-time. The MCMC detector is specially suitable to achieve good communication performance at a moderate throughput with minimal area, as its smallest variant occupies only around 50 kGE. Note that, while the SNR operating points of these detectors are different to our IC-LMMSE detector, for several detector-decoder iterations it has been shown [4] that the gap closes quickly.

VII. CONCLUSION

We present a novel reduced-complexity SISO MMSE MIMO detection algorithm and propose a corresponding VLSI architecture. Its implementation is the – to the best of our knowledge – most area-throughput efficient SISO MIMO detector ASIC reported so far. It outperforms the best competitor, the MMSE-PIC [4], by 2.3x in terms of efficiency, offers a higher throughput and occupies significantly less area, at identical algorithmic performance.

Our ASIC meets the IEEE 802.11n standard's peak data rate requirement of 600 Mbit/s. As such, it is a suitable candidate to bring the impressive communication performance gains of BICM-ID to current wireless networks.

Right now, we are planning to extend our ASIC to support iterative detection, i.e. iterations between equalizer and demapper, as described in the original algorithm paper [1].

REFERENCES

[1] M. Senst and G. Ascheid, "How the framework of expectation propagation yields an iterative IC-LMMSE MIMO receiver," in *Proc. IEEE GLOBECOM*, 2011, pp. 1 – 6.
[2] B. Hochwald and S. Ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, 2003.
[3] "IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 5: Enhancements for higher throughput," *IEEE Std 802.11n-2009*, pp. 1–565, 2009.
[4] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754 – 1765, 2011.
[5] A. Tomasoni *et al.*, "A low complexity turbo MMSE receiver for W-LAN MIMO systems," in *Proc. IEEE ICC*, 2006, pp. 4119–4124.
[6] I. Collings, M. Butler, and M. McKay, "Low complexity receiver design for MIMO bit-interleaved coded modulation," in *IEEE 8th ISSSTA*, 2004, pp. 12–16.
[7] M. Schulte, J. Omar, and E. Swartzlander Jr, "Optimal initial approximations for the Newton-Raphson division algorithm," *Computing*, vol. 53, no. 3-4, pp. 233–242, 1994.
[8] U. Deidersen, D. Auras, and G. Ascheid, "A parallel VLSI architecture for markov chain monte carlo based MIMO detection," in *Proc. ACM GLSVLSI*, 2013, pp. 167–172.
[9] E. M. Witte, "Efficiency and flexibility trade-offs for soft-input soft-output sphere-decoding architectures," Ph.D. dissertation, RWTH Aachen University, 2013.