VLSI Implementation of a Low-Complexity LLL Lattice Reduction Algorithm for MIMO Detection

L. Bruderer*, C. Studer‡, M. Wenk*, D. Seethaler‡, and A. Burg*

*Integrated Systems Laboratory
ETH Zurich, 8092 Zurich, Switzerland
e-mail: {bruderer,mawenk,apburg}@iis.ee.ethz.ch

‡Communication Technology Laboratory
ETH Zurich, 8092 Zurich, Switzerland
e-mail: {studerc,seethal}@nari.ee.ethz.ch

***Abstract*—Lattice-reduction (LR)-aided successive interference cancellation (SIC) is able to achieve close-to optimum error-rate performance for data detection in multiple-input multiple-output (MIMO) wireless communication systems. In this work, we propose a hardware-efficient VLSI architecture of the Lenstra-Lenstra-Lovász (LLL) LR algorithm for SIC-based data detection. For this purpose, we introduce various algorithmic modifications that enable an efficient hardware implementation. Comparisons with existing FPGA implementations show that our design outperforms state-of-the-art LR implementations in terms of hardware-efficiency and throughput. We finally provide reference ASIC implementation results for 130 nm CMOS technology.**

I. Introduction

Multiple-input multiple-output (MIMO) technology enables high spectral efficiency by using multiple antennas at both sides of the wireless link and by transmitting multiple data streams concurrently in the same frequency band. The task of the MIMO detector is to separate the spatially multiplexed data streams at the receiver. Maximum likelihood (ML) detection provides optimum error-rate performance, but the associated computational complexity is high, in general, and hardware efficient VLSI implementation of ML detection is challenging [1]. To reduce the complexity associated with MIMO detection, linear detection or successive interference cancellation (SIC) can be employed. The complexity reduction associated with such low-complexity detection schemes comes, however, at the cost of a significantly degraded error-rate performance.

Lattice reduction (LR) techniques were proposed to reduce the performance gap between low-complexity MIMO detection schemes and ML detection [2] and [3]. The basic idea is to perform sub-optimum detection based on lattice-reduced channel matrices. This approach shifts most of the computational complexity to the preprocessing stage, which needs to be performed only when the channel state changes. Unfortunately, most communication standards require this preprocessing step to be performed under tight latency constraints, which requires high-speed LR implementations. So far, hardware-implementation aspects of LR have only been addressed in [4] and [5] for MIMO detection and in [6] for MIMO precoding.

Contributions: In this paper, we introduce a low-complexity LR algorithm for SIC-based MIMO detection that is based on the Lenstra-Lenstra-Lovász (LLL) algorithm [7] and employs the Siegel criterion [8] and [9]. In an attempt to reduce the computational complexity, we relax the size reduction condition [10]. We further employ early termination (ET) of the algorithm based on the actual execution time, in order to guarantee a minimum throughput. To reduce the performance loss in the presence of ET, we reverse the processing order of the elements in the LR algorithm. Finally, we describe a corresponding hardware-efficient VLSI architecture which, compared to state-of-the-art FPGA implementations [4] and [5], achieves at least a fivefold throughput increase with only a slightly higher hardware complexity. We also provide reference implementation results in 130 nm CMOS technology.

Notation: Matrices are set in boldface capital letters, vectors in boldface lowercase letters. For an $\mathrm{N} \times \mathrm{M}$ matrix $\mathbf{A}$, $\mathbf{a}_\mathrm{j}$ denotes its jth column vector and $A_{j,i}$ is the entry in the jth row and ith column of this matrix. $\mathbf{I}_N$ is the $N \times N$ identity matrix. The superscript ^{H} stands for the conjugate transpose. The set of Gaussian integers is $\mathbb{CZ}$. $\Re\{x\}$ and $\Im\{x\}$ extract the real and imaginary part of $x \in \mathbb{C}$ and rounding to the next Gaussian integer is denoted by $\lceil \cdot \rfloor$.

A. MIMO System Model

Consider a MIMO system with M_T transmit and $M_R \geq M_T$ receive antennas. The M_T-dimensional transmit vector is $\mathbf{x} \in \mathcal{X}^{M_T}$ and $\mathcal{X} \subset \mathbb{CZ}$ corresponds to the underlying scalar complex constellation chosen from a quadrature amplitude modulation (QAM) alphabet. The associated complex baseband input-output relation is given by

$$\mathbf{r} = \mathbf{G}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{G}$ stands for the $M_R \times M_T$ complex-valued channel matrix, $\mathbf{r}$ is the M_R-dimensional receive-vector, and $\mathbf{n}$ is an M_R-dimensional noise vector with i.i.d. circularly symmetric complex Gaussian distributed entries.

B. LR-Aided Low-Complexity MIMO Detection

The task of the MIMO detector is to recover $\mathbf{x}$ from $\mathbf{r}$, based on knowledge of the channel matrix $\mathbf{G}$.

Relaxation and Lattice Reduction: In order to access tools from lattice theory for MIMO detection, the condition $\mathbf{x} \in \mathcal{X}^{M_T}$ on the symbol vector is initially relaxed to $\underline{\mathbf{x}} \in (\mathbb{CZ})^{M_T}$. The purpose of relaxation to $(\mathbb{CZ})^{M_T}$ is to be able to interpret the received vector $\mathbf{r}$ as a point of a lattice $\mathbf{G}\underline{\mathbf{x}}$ that is translated away from the lattice by the noise vector $\mathbf{n}$.

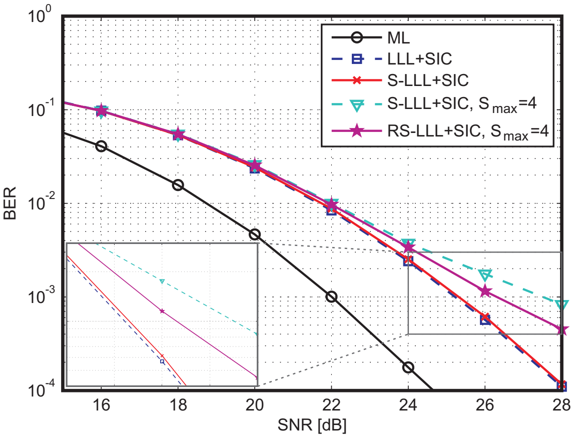

Fig. 1. Uncoded bit error-rate (BER) for LR-aided SIC in an i.i.d. Rayleigh fading $M_T = M_R = 4$ MIMO system with 16-QAM. Remapping to the finite lattice $\mathcal{X}$ is done by quantization [11]. $\delta = 0.75$ has been used for the LLL algorithm, while for the Siegel LLL-variants (S-LLL and RS-LLL) $\epsilon = 0.5$. ML performance is shown as a reference.

The goal of LR is to find a suitable $M_T \times M_T$ unimodular matrix $\mathbf{T}$, i.e., $|\det(\mathbf{T})| = 1$ with $T_{m,n} \in \mathbb{CZ}\ \forall m, n$, such that $\mathbf{B} = \mathbf{GT}$ generates a "more orthogonal" lattice than $\mathbf{G}$. The improved lattice then facilitates the search for the lattice point that is closest to $\mathbf{r}$. Conventional LR-aided algorithms for MIMO detection are based on the LLL algorithm [7]. In this paper, we consider the case where the algorithm is applied to the sorted QR-decomposed channel matrix $\mathbf{GP} = \mathbf{QR}$; $\mathbf{P}$ is a permutation matrix, $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_{M_T}$, and $\mathbf{R}$ is upper triangular. From the LLL algorithm, the QR-decomposition of the reduced lattice generator matrix $\mathbf{B} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$ can be readily obtained.

Successive Interference Cancellation: LR-aided data detection is carried out on a modified version of the input-output relation (1) $\mathbf{r} = \mathbf{B}\underline{\mathbf{z}} + \mathbf{n}$ where $\underline{\mathbf{z}} \in (\mathbb{CZ})^{M_T}$, followed by the computation of $\hat{\underline{\mathbf{x}}} = \mathbf{T}\hat{\underline{\mathbf{z}}}$ and by remapping $\hat{\underline{\mathbf{x}}}$ to the finite lattice $\mathcal{X}^{M_T}$ if $\mathbf{T}\hat{\underline{\mathbf{z}}} \notin \mathcal{X}^{M_T}$. Throughout the paper we consider LR-aided SIC, which was shown to outperform linear detection in terms of error-rate performance [11]. LR-aided SIC essentially solves $\tilde{\mathbf{Q}}^H\mathbf{r} = \tilde{\mathbf{R}}\hat{\underline{\mathbf{z}}}$ for $\hat{\underline{\mathbf{z}}}$ through back-substitution, where in each step, the intermediate result $\hat{\underline{z}}_i$ is quantized to the nearest Gaussian integer before proceeding to the next level.

II. Reversed Siegel LLL Algorithm

The Lovász criterion used in the original LLL algorithm [7] can be replaced with a variant proposed by Siegel [8]; this criterion exhibits, in general, lower computational complexity and entails virtually no loss in terms of error-rate performance, which can be observed in Fig. 1. LLL LR based on Siegel's criterion is referred to as the Siegel LLL (S-LLL) algorithm in the remainder of the paper and has initially been described in [9]. In the following paragraphs, we describe novel methods to further reduce the complexity of the S-LLL algorithm, which enables low-complexity implementation.

A. S-LLL Without Size Reduction

The original S-LLL algorithm performs a so-called full size-reduction step at the end of the algorithm [9], [5]. This step ensures that the size-reduction condition [7]

$$|\tilde{R}_{k,k}| > 2\max\{|\Re\{\tilde{R}_{k,i}\}|, |\Im\{\tilde{R}_{k,i}\}|\}, \quad i = k+1, \ldots, M_T$$

is fulfilled. It can be shown that omitting the full size-reduction does not affect the performance of SIC-based detection schemes . Hence, by avoiding the size reduction procedure, the computational complexity of the S-LLL algorithm can be reduced by about $M_T(M_T-1)/2$ operations (mainly corresponding to multiplications and additions).

Algorithm 1 Reverse Siegel LLL (RS-LLL) Algorithm

```
Init: $\tilde{\mathbf{Q}} \leftarrow \mathbf{Q}$, $\tilde{\mathbf{R}} \leftarrow \mathbf{R}$, $\mathbf{T} \leftarrow \mathbf{I}_{M_T}$, $k \leftarrow M_T$, $S \leftarrow 0$
while $(k \geq 2)$ and $(S < S_{\max})$ do
    if $\epsilon\tilde{R}^2_{k-1,k-1} \geq \tilde{R}^2_{k,k}$ then
        $S = S + 1$
        $\mu = \lceil \tilde{R}_{k-1,k}/\tilde{R}_{k-1,k-1} \rfloor$
        if $\mu \neq 0$ then
            $\tilde{\mathbf{r}}_k \leftarrow \tilde{\mathbf{r}}_k - \mu\tilde{\mathbf{r}}_{k-1}$
            $\mathbf{t}_k \leftarrow \mathbf{t}_k - \mu\mathbf{t}_{k-1}$
        end if
        Exchange $\tilde{\mathbf{r}}_k$ with $\tilde{\mathbf{r}}_{k-1}$ and $\mathbf{t}_k$ with $\mathbf{t}_{k-1}$
        Apply Givens rotation to $\tilde{\mathbf{R}}$ and to $\tilde{\mathbf{Q}}$, such that
        $\tilde{R}_{k,k-1}$ becomes zero
        $k \leftarrow \min(k+1, M_T)$
    else
        $k \leftarrow k - 1$
    end if
end while
```

B. Early Termination

In [12], it was shown that the worst-case (iteration) complexity of the S-LLL is unbounded. Hence, it is of paramount importance to include an ET mechanism into the LR algorithm, in order to meet stringent latency requirements and to guarantee a minimum throughput. Each iteration of the S-LLL algorithm possibly alters the lattice basis $\tilde{\mathbf{R}}$, $\tilde{\mathbf{Q}}$, and $\mathbf{T}$ (corresponding to a column swap), which involves a number of costly computations. Since the number of column swaps is approximately proportional to the time required for processing a matrix on a given architecture (because the computational overhead caused by iterations that do not affect the lattice basis is negligible), our solution is to terminate the algorithm after performing $S_{\max}$ column swaps. This ET scheme is in contrast to previous work, where the algorithm is terminated after a given number of iterations (see, e.g., [13]).

C. Reverse Siegel-LLL (RS-LLL)

It is well-known that the error-rate performance of SIC is dominated by the weakest stream, i.e., the one corresponding to the lower-most diagonal element $\tilde{R}_{M_T,M_T}$ [11]. The S-LLL (as well as the LLL) algorithm starts by processing the top-left element $\tilde{R}_{1,1}$ and then progresses in an iterative fashion toward the bottom-right of $\tilde{\mathbf{R}}$. In the presence of tight run-time constraints, it is, however, possible that the lower-most element $\tilde{R}_{M_T,M_T}$ cannot be processed. We therefore propose a reverse procedure—refered to as reverse S-LLL (RS-LLL)

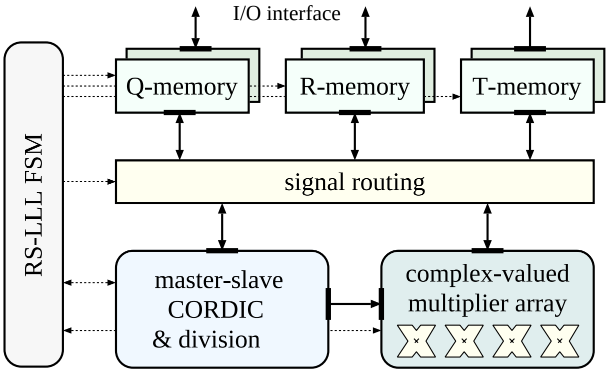

Fig. 2. Overview of the proposed RS-LLL architecture.

algorithm in the sequel—which begins at $\tilde{R}_{M_T,M_T}$ and proceeds iteratively in reverse direction toward $\tilde{R}_{1,1}$. This strategy substantially improves the error-rate performance compared to the common "forward" procedure in the presence of ET (e.g., given $S_{\max} = 4$, the SNR loss at 10^{-3} BER is 1.2 dB lower, see Fig. 1), as the performance-dominating streams are processed first. In addition, it can be shown that a basis which is reduced by the RS-LLL algorithm without ET, meets the Siegel criterion as well, i.e., delivers the same LR "quality" as the S-LLL algorithm.

The RS-LLL algorithm implemented in this paper (excluding size-reduction and including ET) is summarized in Alg. 1. The parameter $\epsilon \in [0.25, 0.5]$ (line 3) influences the performance and the complexity of the algorithm. Fruthermore, the error-rate performance of all discussed LR algorithms can be improved further by operating on a regularized version of the channel matrix [11]. For the sake of simplicity of exposition, however, regularization is not considered in the remainder of the paper, since it would only affect the input of the LR algorithms and is supported by the architecture presented next.

III. Architecture

The high-level architecture proposed for the FPGA and ASIC implementation of the RS-LLL algorithm is shown in Fig. 2. The complex-valued data-path of the design consists of three memories and two arithmetic units which are comprised of an extended CORDIC circuit and of an array of four complex-valued multipliers. Memories and arithmetic units are connected through a dedicated routing network and the operation of the data-path components is controlled by a reactive finite state machine (FSM). The memories are double-buffered to support concurrent data transfer to and from the RS-LLL unit during operation and were realized using flip-flop arrays instead of using memory macro-cells. This implementation is necessary to enable the irregular and parallel access that is required to fully utilize the arithmetic units. To save area, the R-memory exploits the fact that $\tilde{\mathbf{R}}$ is upper triangular and its diagonal elements are real-valued.

In the following, we step through the RS-LLL algorithm described in Alg. 1 and explain how the operations are mapped to the data-path elements.

A. Computation of the Siegel Criterion

Each iteration starts by computing and checking the Siegel criterion (line 3 of Alg. 1), which can be mapped to a single complex-valued multiplication operation. Since a total of four complex-valued multipliers is available, three independent evaluations of the Siegel criterion can be performed in one clock cycle. Hence, no time is spent for proceeding to another diagonal element, if the criterion is not met for a particular k.

B. Size Reduction

When the Siegel criterion is met, two columns need to be exchanged and a size reduction must be performed (lines 5-9 of Alg. 1). This size reduction starts with the computation of the coefficient μ, which is obtained through an integer-rounded division operation. In [14], a corresponding architecture based on the Newton-Raphson method is presented. We found, however, that the result of the division μ can be limited to a small dynamic range, as the entries of $\mathbf{T}$ also need to be limited to enable an efficient fixed-point implementation. As a consequence of limiting the dynamic range of $\mathbf{T}$ such that the error-rate performance loss remains negligible, the coefficient μ can also be constrained to low precision. To take advantage of this observation, the division (line 5 of Alg. 1) is computed with a non-restoring divider [15] which can easily be built into the extended CORDIC circuit that is already present in the data-path. The rounding operation is realized with a simple look-up table, which saves the more complex classical rounding operation.

The computation of μ is followed by the update of $\tilde{\mathbf{r}}_k$ and of $\mathbf{t}_k$. For $\tilde{R}_{k-1,k}$, the update is obtained from the CORDIC as a side-product of the preceding division operation. For the remaining elements of $\tilde{\mathbf{r}}_k$, the update is carried out on the complex-valued multipliers. For the updates of $\mathbf{t}_k$, it is not efficient to utilize the complex-valued multipliers, because of the small widths of the involved operands. Hence, a dedicated low-precision multiply-and-accumulate logic was introduced directly in the T-memory.

C. Givens Rotations

After the size reduction and the column exchange, the upper-triangular structure of $\tilde{\mathbf{R}}$ must be restored (line 11 of Alg. 1). Application of a complex-valued Givens rotation null the matrix element $\tilde{R}_{k,k-1}$ and update the corresponding elements of $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{Q}}$, such that $\mathbf{GT} = \tilde{\mathbf{Q}}\tilde{\mathbf{R}}$. We employ a master-slave CORDIC architecture as proposed in [16], [17] to perform these tasks. A complex-valued (master) CORDIC in vectoring mode performs the nulling, while a second (slave) CORDIC concurrently computes the corresponding phasor. With these phasors, the updates of the remaining entries of $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{Q}}$ can then be performed more efficiently on the complex-valued multipliers. The number of such multiplier units that yield optimum resource utilization and area efficiency is determined by the ratio of vectoring to rotation operations and by the number of required micro-rotations [17]. Accordingly, for the complex-valued RS-LLL algorithm supporting $M_T = M_R = 4$, four complex-valued multipliers and a master-slave CORDIC (that computes three micro-rotations per clock cycle and performs nine micro-rotations in total) were selected.

TABLE I
FPGA IMPLEMENTATION RESULTS AND COMPARISON

	[5]	This work		[4]
LR algorithm	S-LLL	RS-LLL		LLL
FPGA type	Virtex-II Pro		Virtex-IV	
Speed grade	n.a.	-5	-12	n.a.
Max. clock frequency [MHz]	100	45	79	140
Slices	7 349	4 379	4 805	3 617
Slice FFs	9 051	1 503	1 492	n.a.
Slice LUTs	10 254	8 171	8 206	n.a.
Block RAMs	69	0	0	n.a.
Multipliers	24	16	18	10
Avg. cycles per matrix	420	14		130
Avg. throughput [MMat/s]	0.2	3.2	5.6	1.1

TABLE II
IMPLEMENTATION RESULTS IN 130 NM (1P/8M) CMOS TECHNOLOGY

	This work
Max. clock frequency	333 MHz
Cell area[a]	107 kGE
Core area	0.925 mm^2
Avg. cycles per matrix	14
Avg. throughput	23.8 MMat/s

[a] One gate equivalent (GE) corresponds to a two-input drive-2 NAND gate.

IV. IMPLEMENTATION RESULTS

The RS-LLL architecture described above was implemented for a $M_T = M_R = 4$ MIMO system on a Virtex-II Pro VP70 FPGA, a Virtex-IV VLX160 FPGA, and in 130 nm CMOS technology. In the following we compare our design to existing FPGA implementations and present reference VLSI implementation results.

A. FPGA Implementation and Comparison

Implementation results for the suggested RS-LLL algorithm are summarized in Tbl. I together with previously reported designs [4], [5]. To enable a fair comparison, figures are reported with single-buffered memories and a high runtime limit (i.e., $S_{\max} = 20$).

The design reported in [5] implements the S-LLL algorithm and achieves significantly lower throughput (at comparable circuit complexity) than our implementation. The FPGA design in [4] implements the complex-valued LLL algorithm. Our RS-LLL implementation achieves a fivefold improvement in terms of throughput at the cost of only slightly more FPGA resources. We therefore conclude that the presented RS-LLL implementation outperforms state-of-the-art LR FPGA implementations in terms of hardware-efficiency and throughput.

B. ASIC Implementation

The RS-LLL architecture has been implemented on a 130 nm CMOS process. Corresponding implementation results are given in Tbl. II. Our implementation delivers 23.8 M matrices per second while requiring 107 kGE. Since no ASIC implementation has been described so far in the open literature, we are unable to perform comparison with existing designs.

V. CONCLUSION

We studied the S-LLL algorithm for LR-aided SIC in MIMO systems, regarding the suitability for VLSI implementation. To this end, the algorithm's complexity has been reduced and a hardware-efficient architecture has been described. Corresponding implementation results on FPGAs demonstrate that reference architectures proposed in [4], [5] are outperformed in terms of throughput by at least a factor of five, while requiring only slightly more FPGA resources. Reference VLSI implementation results in 130 nm CMOS technology are shown and we conclude that SIC-based MIMO detection aided by the RS-LLL algorithm proposed in this paper seems to be a promising technique for sub-optimal and low-complexity MIMO detection.

ACKNOWLEDGMENTS

This work was supported in part by the Swiss National Science Foundation project No. PP002-119052 and by the STREP project No. IST-026905 (MASCOT) within the sixth framework programme of the European Commission. The authors would like to thank F. Gürkaynak, N. Felber, and W. Fichtner for their support in designing the RS-LLL ASIC.

REFERENCES

[1] H. Bölcskei, D. Gesbert, C. Papadias, and A. J. van der Veen, Eds., *Space-Time Wireless Systems: From Array Processing to MIMO Communications*. Cambridge Univ. Press, 2006.

[2] H. Yao and G. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. IEEE GLOBECOM*, vol. 1, Nov. 2002, pp. 424–428.

[3] C. Windpassinger and R. F. H. Fischer, "Low-complexity near-maximum-likelihood detection and precoding for MIMO systems using lattice reduction," in *Proc. IEEE ITW*, Mar. 2003, pp. 345–348.

[4] B. Gestner, W. Zhang, X. Ma, and D. V. Anderson, "VLSI implementation of a lattice reduction algorithm for low-complexity equalization," in *Proc. IEEE ICCSC*, May 2008, pp. 643–647.

[5] L. Barbero, D. Milliner, T. Ratnarajah, J. Barry, and C. Cowan, "Rapid prototyping of Clarkson's lattice reduction for MIMO detection," in *Proc. IEEE ICC*, Jun. 2009.

[6] A. Burg, D. Seethaler, and G. Matz, "VLSI implementation of a lattice-reduction algorithm for multi-antenna broadcast precoding," in *Proc. IEEE ISCAS*, May 2007, pp. 673–676.

[7] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, pp. 515–534, 1982.

[8] C. L. Siegel, *Lectures on the Geometry of Numbers*. Berlin, Germany: Springer-Verlag, 1989.

[9] I. V. L. Clarkson, "Approximation of linear forms by lattice points with applications to signal processing," Ph.D. dissertation, The Australian National University, 1997.

[10] C. Ling and N. Howgrave-Graham, "Effective LLL reduction for lattice decoding," in *Proc. IEEE ISIT*, Jun. 2007, pp. 196–200.

[11] D. Wübben and D. Seethaler, "On the performance of lattice reduction schemes for MIMO data detection," in *Proc. 41th Asilomar Conference on Signals, Systems and Computers*, Nov. 2007, pp. 1534–1538.

[12] J. Jaldén, D. Seethaler, and G. Matz, "Worst- and average-case complexity of LLL lattice reduction in MIMO wireless systems," in *Proc. IEEE ICASSP*, Apr. 2008, pp. 2685–2688.

[13] H. Vetter, V. Ponnampalam, M. Sandell, and P. A. Hoeher, "Fixed complexity LLL algorithm," *IEEE Trans. Signal Process.*, vol. 57, pp. 1634–1637, Apr. 2009.

[14] B. Gestner and D. V. Anderson, "Single Newton-Raphson iteration for integer-rounded divider for lattice reduction algorithms," in *Proc. 51st MWSCAS*, Aug. 2008, pp. 966–969.

[15] B. Parhami, *Computer arithmetic*. Oxford University Press, 2000.

[16] P. Luethi, A. Burg, S. Haene, D. Perels, N. Felber, and W. Fichtner, "VLSI implementation of a high-speed iterative sorted MMSE QR decomposition," in *Proc. IEEE ISCAS*, May 2007, pp. 1421–1424.

[17] C. Senning, C. Studer, P. Luethi, and W. Fichtner, "Hardware-efficient steering matrix computation architecture for MIMO communication systems," in *Proc. IEEE ISCAS*, May 2008, pp. 304–307.