A 757 Mb/s 1.5 mm^2 90 nm CMOS Soft-Input Soft-Output MIMO Detector for IEEE 802.11n

C. Studer*, S. Fateh‡, and D. Seethaler*

ETH Zurich, 8092 Zurich, Switzerland

e-mail: *{studerc,seethal}@nari.ee.ethz.ch, ‡fateh@iis.ee.ethz.ch

***Abstract*—Multiple-input multiple-output (MIMO) wireless technology is the key to meet the demands for data rate, quality-of-service, and bandwidth-efficiency of modern wireless communication systems. MIMO technology is therefore adopted in many recent communication standards, such as IEEE 802.11n. Here, the MIMO detector has a strong impact on the overall system performance. In fact, the full potential of MIMO communication systems can only be achieved by means of iterative MIMO decoding using *soft-input soft-output* (SISO) data detection. In this paper, we present—to the best of our knowledge—the first VLSI implementation of a SISO detector for iterative MIMO decoding. The presented ASIC supports SISO detection for four spatial streams and enables more than 6 dB signal-to-noise-ratio improvement over state-of-the-art MIMO detectors. The 1.5 mm^2 ASIC is fabricated in 90 nm CMOS and achieves 757 Mb/s, which exceeds the 600 Mb/s IEEE 802.11n peak data-rate.**

I. Introduction

Modern wireless communication systems, such as the IEEE 802.11n WLAN standard [1], are based on multiple-input multiple-output (MIMO) technology, which meets the demand for reliable, high-speed, and bandwidth-efficient data transmission. In these systems, MIMO detection, i.e., the separation of the spatially-multiplexed data streams, and channel decoding are among the main challenges in computational complexity and corresponding efficient implementations are the key to facilitate high-performance and low-cost user equipment.

ASIC implementations of state-of-the-art high-performance MIMO detection using sphere-decoding (SD) [2], [3] are unable to achieve the 600 Mb/s peak data-rate of IEEE 802.11n, which is due to SD's prohibitive worst-case complexity. Recent ASIC implementations of suboptimum MIMO detection, e.g., the k-Best detector [4] or soft-output minimum mean-square error (MMSE) detection [5], exceed the 600 Mb/s peak data-rate, but at the cost of inferior error-rate performance, which eventually degrades the system throughput, coverage, and range. All these techniques rely on a single channel-decoding step without iteratively exchanging information with the MIMO detector. However, as it was shown in [6], the full potential of MIMO wireless communication systems can only be achieved through *iterative MIMO decoding*.

At the heart of an iterative MIMO decoder is a soft-input soft-output (SISO) MIMO detector (referred to as "SISO detector"), which iteratively exchanges reliability information of the coded bits with a SISO channel decoder. A SISO detector exhibits, in general, very high computational complexity (see, e.g., [6]), which necessitates the design of low-complexity algorithms and corresponding dedicated ASIC implementations.

Contributions: In this paper, we present—to the best of our knowledge—the first ASIC implementation of a SISO detection algorithm for iterative MIMO decoding. To this end, we develop a reduced-complexity variant of the MMSE parallel interference cancellation (PIC) algorithm proposed in [7] and design a VLSI architecture consisting of eight parallel processing units (PUs) to achieve the peak data-rate of IEEE 802.11n. We provide measurement results of the 90 nm CMOS ASIC and finally demonstrate that substantial performance gains can be achieved compared to state-of-the-art (non-iterative) MIMO-detector implementations.

Notation: Matrices are set in boldface capital letters, vectors in boldface lowercase letters. The superscript ^{H} stands for conjugate transpose and $\mathbf{I}_M$ is the $M \times M$ identity matrix. $\mathrm{P}[\cdot]$ denotes probability. Expectation and variance are referred to as $\mathbb{E}[\cdot]$ and $\mathrm{Var}[\cdot]$, respectively.

II. MIMO System and Algorithm Description

We consider a coded MIMO system with M_T transmit and $M_\mathrm{R} \geq M_\mathrm{T}$ receive antennas (see Fig. 1) employing spatial multiplexing as specified in IEEE 802.11n [1]. The information bits $\mathbf{b}$ are encoded (e.g., using a convolutional code) and the coded bit-stream $\mathbf{x}$ is mapped to a sequence of transmit vectors $\mathbf{s} \in \mathcal{O}^{M_\mathrm{T}}$, where $\mathcal{O}$ corresponds to the scalar complex constellation of size 2^Q. Each transmit vector $\mathbf{s}$ is associated with $M_\mathrm{T}Q$ binary values $x_{i,b} \in \{0,1\}$, $i = 1,\ldots,M_\mathrm{T}$, $b = 1,\ldots,Q$, corresponding to the bth bit of the ith entry (i.e., spatial stream) of $\mathbf{s}$. The baseband input-output relation of the wireless MIMO channel is given by $\mathbf{y} = \mathbf{Hs} + \mathbf{n}$, where $\mathbf{H}$ stands for the $M_\mathrm{R} \times M_\mathrm{T}$ complex-valued channel matrix, $\mathbf{y}$ is the M_R-dimensional received vector, and $\mathbf{n}$ is M_R-dimensional i.i.d. zero-mean complex Gaussian distributed with variance N_0 per entry.

A. Principle of Iterative MIMO Decoding

Iterative MIMO decoding applies the key ideas of turbo-decoding [8] to data detection in MIMO systems. Here, reliability information of the coded bits—in terms of log-likelihood ratios (LLRs)—is iteratively exchanged between the SISO detector and the SISO channel decoder (see Fig. 1) to successively improve the error-rate performance. In each iteration, the SISO detector computes the LLRs [6]

$$L^\mathrm{D}_{i,b} = \log\left(\frac{\mathrm{P}[x_{i,b} = 1 \,|\, \mathbf{y}]}{\mathrm{P}[x_{i,b} = 0 \,|\, \mathbf{y}]}\right) \qquad (1)$$

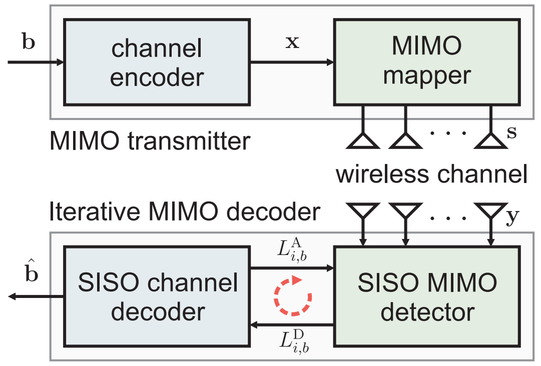


Figure 1. MIMO communication system using iterative MIMO decoding.

for each coded bit $x_{i,b}$, based on the received vector $\mathbf{y}$, the channel matrix $\mathbf{H}$, and the a-priori LLRs $L^{\mathrm{A}}_{i,b}$, $\forall i, b$. The LLRs $L^{\mathrm{D}}_{i,b}$ are then delivered to the SISO channel decoder, which computes *new* a-priori LLRs $L^{\mathrm{A}}_{i,b}$, $\forall i, b$, that are used by the SISO detector in the next iteration. After a given number of iterations (denoted by I), the SISO channel decoder computes final estimates $\hat{\mathbf{b}}$ for the information bits.

B. Reduced-Complexity SISO MMSE-PIC Algorithm

Even for a small number of spatial streams (say $M_{\mathrm{T}} > 2$), exact computation of the LLRs in (1) exhibits prohibitive complexity. Therefore, a complexity-reduced variant of the SISO MMSE-PIC algorithm in [7] is considered in the following. Our algorithm performs SISO detection in *six steps* and is summarized below (refer to [9] for more details):

1) Gram matrix and matched-filter: To reduce the amount of recurrent (and hence, redundant) operations, compute the Gram matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ and the matched-filter output according to $\mathbf{y}^{\mathrm{MF}} = \mathbf{H}^H\mathbf{y}$.

2) Soft-symbols and variances: Compute soft-symbols for each spatial stream $i = 1, \ldots, M_{\mathrm{T}}$, according to

$$\hat{s}_i = \mathbb{E}[s_i] = \sum_{a\in\mathcal{O}} \mathrm{P}[x_{i,b} = [a]_b]\, a \tag{2}$$

where $[a]_b$ corresponds to the bth bit associated with the constellation point $a \in \mathcal{O}$. The soft-estimates in (2) are computed on the basis of the a-priori LLRs $L^{\mathrm{A}}_{i,b}$ (provided by the SISO channel decoder) according to $\mathrm{P}[x_{i,b} = x] = \frac{1}{2}\big(1 + (2x-1)\tanh\big(\frac{1}{2}L^{\mathrm{A}}_{i,b}\big)\big)$. In the first iteration, no a-priori information is available, which implies $L^{\mathrm{A}}_{i,b} = 0$, $\forall i, b$. The variances $E_i = \mathrm{Var}[s_i]$ of the soft-symbols are computed analogously to (2).

3) Parallel interference cancellation (PIC): Next, the SISO detector performs PIC according to

$$\hat{\mathbf{y}}^{\mathrm{MF}}_i = \mathbf{y}^{\mathrm{MF}} - \sum_{j\neq i}\mathbf{g}_j\hat{s}_j = \mathbf{g}_i s_i + \underbrace{\mathbf{n} + \sum_{j\neq i}\mathbf{g}_j e_j}_{\text{noise}-\text{plus}-\text{interference}} \tag{3}$$

for each stream i, where $\mathbf{g}_i$ stands for the ith column of $\mathbf{G}$ and $e_j = s_j - \hat{s}_j$. The SISO MMSE-PIC algorithm now performs approximate detection based on (3). To this end, the single-stream system in (3) is considered as *independent* from the other spatial streams $j \neq i$ and the errors e_j are assumed as zero-mean Gaussian with variances E_j.

4) Matrix inversion for MMSE filtering: For each spatial stream in (3), an MMSE-filter operation is performed to suppress the noise-plus-interference term. The original algorithm [7] requires M_{T} matrix inversions for the computation of all M_{T} MMSE filter vectors, which inhibits the efficient implementation in hardware. Hence, we deploy a low-complexity method that yields *the same* LLRs (see [9] for the proof) and only requires *one* matrix inversion of the same size for the simultaneous computation of *all* filter vectors. To this end, we compute the inverse $\mathbf{A}^{-1} = (\mathbf{G}\boldsymbol{\Lambda} + N_0\mathbf{I}_{M_{\mathrm{T}}})^{-1}$, where $\boldsymbol{\Lambda}$ is an $M_{\mathrm{T}} \times M_{\mathrm{T}}$ diagonal matrix with $\Lambda_{i,i} = E_i$, $\forall i$ and the rows of $\mathbf{A}^{-1}$ correspond to the M_{T} filter vectors.

5) MMSE filtering: Compute the MMSE filter outputs according to $z_i = \mu_i^{-1}\mathbf{a}_i^H\hat{\mathbf{y}}^{\mathrm{MF}}_i$, $\forall i$, where $\mathbf{a}_i^H$ is the ith row of the matrix $\mathbf{A}^{-1}$ and $\mu_i = \mathbf{a}_i^H\mathbf{g}_i$.

6) LLR computation: The SISO MMSE-PIC algorithm finally approximates the LLRs in (1) according to

$$L^{\mathrm{D}}_{i,b} \approx \rho_i\Big(\min_{a\in\mathcal{Z}_b^{(0)}}|z_i - a|^2 - \min_{a\in\mathcal{Z}_b^{(1)}}|z_i - a|^2\Big) \tag{4}$$

with $\rho_i = \frac{\mu_i}{1-E_i\mu_i}$ being the ith-stream post-equalization signal-to-noise-plus-interference-ratio and $\mathcal{Z}_b^{(0)}$ and $\mathcal{Z}_b^{(1)}$ refer to the subsets of $\mathcal{O}$, where the bth bit is 0 and 1, respectively.

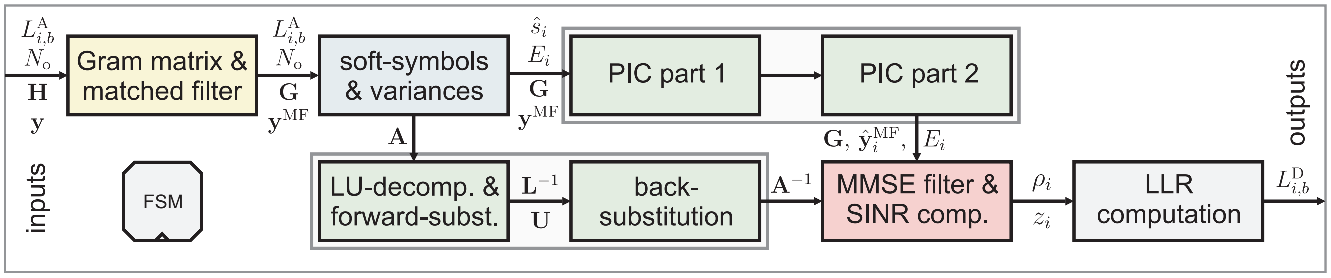
III. VLSI Architecture

In order to efficiently compute the reduced-complexity SISO MMSE-PIC algorithm in hardware, we propose an architecture consisting of eight processing units (PUs) each having similar structure. The high-level VLSI architecture of the PU-partitioning, along with the corresponding six processing steps (as described in Sec. II-B), is depicted in Fig. 2. We optimized each PU independently, which led itself to a high-throughput and area-efficient VLSI architecture, while requiring low development and verification time.

The proposed architecture processes six receive vectors concurrently and in a pipelined manner. Each PU performs the assigned tasks in $T_{\mathrm{s}} = 18$ clock cycles, which was chosen to arrive at a low silicon complexity while achieving the 600 Mb/s peak data rate of IEEE 802.11n. The results of a PU are passed to the subsequent PUs (or to the output of the detector) every 18th cycle, which is referred to as the "exchange-cycle" in the following. This systolic-like processing scheme leads to an overall latency of 108 clock cycles and achieves a constant throughput of $\frac{M_{\mathrm{T}}Q}{T_{\mathrm{s}}}f_{\mathrm{clk}}$ bit/s scaling linearly in the clock frequency f_{clk}. Consequently, in this architecture, the throughput is maximized by minimizing the lengths of the critical paths in all PUs.

A. Processing Unit (PU) Architecture

The architectural principle underlying each PU is depicted on the left-hand side of Fig. 3. Each PU contains a finite state machine (FSM) controlling the data memory, an interconnection network, and a task-specific set of arithmetic units (AUs). The data memories are formed by arrays of flip-flops in order to meet the high memory-bandwidth required by the parallel AU-instances and to enable irregular access to multiple data words. The total set of AUs corresponds to adders, multipliers,



$L_{i,b}^{\mathrm{A}}$ N_o $\mathbf{H}$ $\mathbf{y}$
Gram matrix & matched filter
$L_{i,b}^{\mathrm{A}}$ N_o $\mathbf{G}$ $\mathbf{y}^{\mathrm{MF}}$
soft-symbols & variances
$\hat{s}_i$ E_i $\mathbf{G}$ $\mathbf{y}^{\mathrm{MF}}$
PIC part 1
PIC part 2
$\mathbf{A}$
LU-decomp. & forward-subst.
$\mathbf{L}^{-1}$ $\mathbf{U}$
back-substitution
$\mathbf{A}^{-1}$
$\mathbf{G}$, $\hat{\mathbf{y}}_i^{\mathrm{MF}}$, E_i
MMSE filter & SINR comp.
ρ_i z_i
LLR computation
$L_{i,b}^{\mathrm{D}}$
inputs
outputs
FSM

Figure 2. Proposed high-level VLSI architecture of the SISO MMSE-PIC detector.

multiply-accumulate (MAC) units, arithmetic shifters (mainly used to improve numerical precision), comparators, look-up tables (required to approximate the probabilities $\mathrm{P}[x_{i,b} = [a]_b]$), and reciprocal units. The set of AUs required by a specific PU is determined such that all required operations are completed in exactly $T_\mathrm{s} = 18$ clock cycles.

To minimize the length of the critical path, fixed-point arithmetic is used and the AU-internal word-lengths are optimized with the aid of simulations. Further reduction of the length of the critical path is obtained by inserting a pipeline-register at the input of each AU, which is then re-timed with the aid of the synthesis tool. The feed-through capability allows a parallel transfer of all the data-memory contents from one PU to the subsequent PU(s) in the exchange-cycle. In this cycle, some AUs also provide computation results, which are directly passed to the corresponding next PU. To reduce dynamic power consumption in the case that no data-frame needs to be processed, the clock of each PU can be gated individually.

B. Matrix Inversion Using the LU-Decomposition

The computation of $\mathbf{A}^{-1}$ in Step 4 of the SISO MMSE-PIC algorithm (see Section II-B) dominates the computational complexity of the algorithm. In order to perform matrix-inversion at high throughput and with sufficiently high arithmetic precision, we propose the use of a LU-decomposition (LUD) based inversion procedure. In contrast to other methods (such as, e.g., QR-based matrix inversion), we observed that it is economic and exhibits good numerical stability. As shown in Fig. 2, the required inversion computations are performed in two separate PUs, where the first PU computes the LU-decomposition (LUD) $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L}$ and $\mathbf{U}$ are lower- and upper-triangular matrices, respectively, and the forward-substitution procedure to solve $\mathbf{L}\mathbf{v}_i = \mathbf{e}_i$ for $\mathbf{v}_i$, $i = 1, \ldots, M_\mathrm{T}$, where $\mathbf{e}_i$ denotes the ith unit vector. The second PU associated to the LUD-based matrix inversion step computes the back-substitution $\mathbf{U}\mathbf{x}_i = \mathbf{v}_i$ for $\mathbf{x}_i$, $i = 1, \ldots, M_\mathrm{T}$, which finally yields the desired inverse according to $\mathbf{A}^{-1} = [\,\mathbf{x}_1 \cdots \mathbf{x}_{M_\mathrm{T}}\,]$.

C. Newton-Raphson-Based Reciprocal Unit

At various steps of the algorithm (such as for the matrix inversion in Step 4 and the computation of μ_i and ρ_i in Step 5 and Step 6, respectively) reciprocal values (i.e., $1/x$) have to be computed. We identified these reciprocal computations as critical in terms of the maximum achievable clock-frequency as well as in terms of the required arithmetic precision.

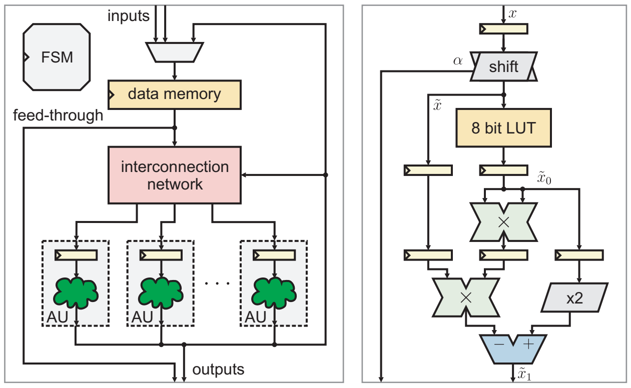


Figure 3. Left: PU architecture overview. Right: Register transfer-level architecture of the pipelined Newton-Raphson-based reciprocal unit.

Therefore, we designed a custom reciprocal unit delivering one reciprocal value per clock cycle (shown on the right-hand side of Fig. 3). First, to improve numerical precision, the input value x is shifted according to $\tilde{x} = 2^\alpha x$ (with $\alpha \in \mathbb{Z}$) such that the MSB of $\tilde{x}$ becomes non-zero (the scaling 2^α is accounted for in later stages using arithmetic shifters). Next, based on an initial guess $\tilde{x}_0$ of $1/\tilde{x}$ obtained from an 8 bit look-up table (LUT), a *single* Newton-Raphson iteration according to $\tilde{x}_1 \leftarrow 2\tilde{x}_0 - \tilde{x}\tilde{x}_0^2$ is performed. The resulting unit provides $\tilde{x}_1 \approx 1/\tilde{x}$ with 15 bit precision (excluding the initial shift), which was shown to be sufficient to attain a very small implementation loss (see Fig. 5). The insertion of two pipeline stages in the reciprocal unit finally moved the critical path to a 24 bit×28 bit multiplier of the back-substitution PU.

IV. Performance and Implementation Results

The final design performs SISO detection of four spatial streams and supports BPSK, QPSK, 16-QAM, and 64-QAM. The ASIC was fabricated in 90 nm (1P/9M) CMOS technology. Fig. 5 shows the chip micrograph (due to library constraints, no signal routing was used on the 9th metal layer).

A. Performance of Iterative MIMO Decoding

Fig. 5 demonstrates the SNR-performance advantages of iterative MIMO decoding using the SISO MMSE-PIC detector (based on the proposed SISO MMSE-PIC ASIC and based on the corresponding ideal floating-point algorithm according to [7]) over non-iterative (i.e, $I = 1$) state-of-the-art MIMO detection schemes based on hard-output SD [2], [3], k-Best detection [4], and soft-output MMSE detection [5]. We note that for $I = 1$, SISO MMSE-PIC detection coincides with soft-output MMSE detection. One can observe that for the

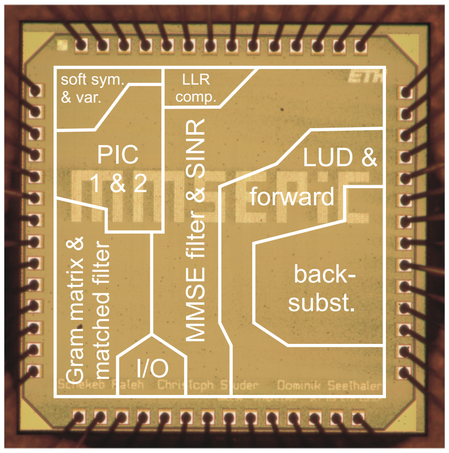


Figure 4. SISO MMSE-PIC chip micrograph with highlighted PUs (I/O refers to logic required for the input/output interface of the chip).

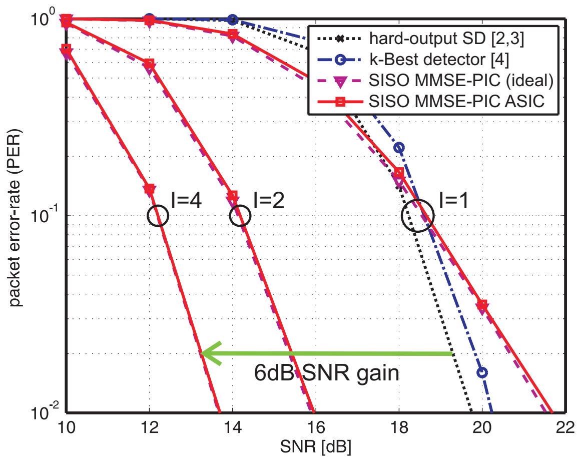


Figure 5. Packet error-rate (PER) comparison of various MIMO detection algorithms in a typical 40 MHz IEEE 802.11n scenario (MCS 27) with 4-spatial streams, 16-QAM, rate-1/2 convolutional code, 864 information bits per packet, and using a TGn type C channel model. The arrow indicates the SNR-performance gain through iterative MIMO decoding using the SISO MMSE-PIC algorithm with four iterations over (non-iterative) hard-output SD.

non-iterative case, hard-output SD slightly outperforms all other detection algorithms at 1% PER. However, two or four iterations with the SISO MMSE-PIC detector yield 3.9 dB and 6 dB SNR improvement, respectively, over the corresponding non-iterative algorithms. Finally, one can observe that the implementation loss of the proposed SISO MMSE-PIC ASIC compared to that of the ideal algorithm is less than 0.2 dB.

B. Implementation Results

The proposed SISO MMSE-PIC ASIC has the following key characteristics (see also Tbl. I).[1] Its core area is 1.5 mm^2 (at 86% cell density) and its maximum clock frequency is 568 MHz, which results in a maximum throughput of 757 Mb/s per iteration (for 4-spatial streams and 64-QAM) achieving the 600 Mb/s peak data-rate specified in IEEE 802.11n with margin. The power consumption[2] is 769 mW leading to an energy-efficiency of 1.02 nJ/bit per iteration.

[1] All measurement results (for maximum clock-frequency and power consumption) were carried out on an HP 83 000 F660 VLSI test system.

[2] Measured at max. throughput, 1.2 V core supply, and $T = 300$ K.

Table I
ASIC IMPLEMENTATION RESULTS AND COMPARISON

	This work	Burg *et al.* [5]	Shabany and Gulak [4]
Detection algorithm	SISO MMSE-PIC	soft-output MMSE	hard-output k-Best
Iterative MIMO decoding	yes	no	no
SNR operating point[a] [dB]	13.7	21.7	20.3
CMOS technology [nm]	90	130	130
Preprocessing area [kGE]	410[b]	251	–
Detection area [kGE]		67	114
Max. throughput [Mb/s]	757	1386[c]	950[c]

[a] Corresponding to the minimum SNR required for 1% PER (see Fig. 5.)
[b] One gate equivalent (GE) corresponds to a 2-input drive-1 NAND gate.
[c] Throughput scaled by 1.45 to account for 130 nm CMOS technology.

Tbl. I provides a comparison of the proposed SISO MMSE-PIC ASIC with two state-of-the-art non-iterative MIMO detector implementations [4], [5] that exhibit constant throughput and achieve the 600 Mb/s peak data-rate of the IEEE 802.11n standard.[3] We note that no VLSI implementation of a SISO detection algorithm for iterative MIMO decoding was reported in the open literature. From Tbl. I one can observe that, by enabling the significant SNR-performance gains offered by iterative MIMO decoding, the proposed SISO MMSE-PIC ASIC is only two times less efficient in terms of kGE/Mb/s than the soft-output MMSE detector of [5]. We note that the area result of the k-Best detector in [4] is rather optimistic, as it does not include the necessary preprocessing circuitry.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support from H. Bölcskei, A. Burg, N. Felber, W. Fichtner, F. Gürkaynak, and Q. Huang during the ASIC design and writing of the paper.

REFERENCES

[1] *IEEE Draft Standard; Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications; Amendment 4: Enhancements for Higher Throughput*, P802.11n/D3.0, Sep. 2007.
[2] A. Burg *et al.*, "VLSI implementation of the sphere decoding algorithm," in *Proc. IEEE ESSCIRC*, Sept. 2004, pp. 303–306.
[3] C.-H. Yang and D. Marković, "A 2.89mw 50GOPS 16×16 16-core MIMO sphere decoder in 90nm CMOS," in *Proc. IEEE ESSCIRC*, Sept. 2009, pp. 344–347.
[4] M. Shabany and P. G. Gulak, "A 0.13 µm CMOS, 655 Mb/s 4×4 64-QAM k-best MIMO detector," in *Dig. Techn. Papers, IEEE ISSCC*, Feb. 2009.
[5] A. Burg *et al.*, "A 4-stream 802.11n baseband transceiver in 0.13 µm CMOS," in *Dig. Techn. Papers, Symp. on VLSI Circuits*, Jun. 2009, pp. 282–283.
[6] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE T-COM*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
[7] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE T-COM*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.
[8] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding," in *Proc. of IEEE ICC*, May 1993, pp. 1064–1070.
[9] C. Studer, S. Fateh, and D. Seethaler, "Soft-input soft-output MIMO detection using MMSE parallel interference cancellation: Algorithm and VLSI implementation," *in preparation*.

[3] We note that the hard-output SD implementations of [2], [3] do *not* achieve the 600 Mb/s peak data-rate of the IEEE 802.11n standard [1].