# A 2.78 mm$^2$ 65 nm CMOS Gigabit MIMO Iterative Detection and Decoding Receiver

Filippo Borlenghi*, Ernst Martin Witte*, Gerd Ascheid*, Heinrich Meyr*†, Andreas Burg‡
*Institute for Communication Technologies and Embedded Systems, RWTH Aachen University, 52056 Aachen, Germany
†Visiting Professor at the Integrated Systems Laboratory 1, EPFL, 1015 Lausanne, Switzerland
‡Telecommunications Circuits Laboratory, EPFL, 1015 Lausanne, Switzerland
email: {borlenghi,witte,ascheid,meyr}@ice.rwth-aachen.de, andreas.burg@epfl.ch

***Abstract*—Iterative detection and decoding (IDD), combined with spatial-multiplexing multiple-input multiple-output (MIMO) transmission, is a key technique to improve spectral efficiency in wireless communications. In this paper we present the—to the best of our knowledge—first complete silicon implementation of a MIMO IDD receiver. MIMO detection is performed by a multi-core sphere decoder supporting up to 4×4 as antenna configuration and 64-QAM modulation. A flexible low-density parity check decoder is used for forward error correction. The 65 nm CMOS ASIC has a core area of 2.78 mm$^2$. Its maximum throughput exceeds 1 Gbit/s, at less than 1 nJ/bit. The MIMO IDD ASIC enables more than 2 dB performance gains with respect to non-iterative receivers.**

## I. Introduction

State-of-the-art wireless communication standards employ multiple-input multiple-output (MIMO) technology with bit-interleaved coded modulation (BICM) supporting high modulation orders, advanced forward error-correcting (FEC) coding, and rate adaptation. Receivers with close-to-optimum performance reduce the signal-to-noise ratio (SNR) at which a given data rate is reliably supported, thus maximizing the operating range. Iterative detection and decoding (IDD) [1] enables near-capacity operation and provides a performance advantage of more than 2 dB over non-iterative receivers. As shown in Fig. 1, in an IDD system detector and decoder exchange soft information[1]. Both components repeatedly compute bit-wise posterior L-values $\lambda^{\mathrm{p}}$ based on prior L-values $\lambda^{\mathrm{a}}$ provided by the other component and then forward new extrinsic L-values $\lambda^{\mathrm{e}} = \lambda^{\mathrm{p}} - \lambda^{\mathrm{a}}$. Unfortunately, IDD entails considerable complexity, especially in the context of MIMO. While recent papers describe building blocks for IDD in MIMO systems [2], [3], no complete MIMO IDD receiver has been reported so far. Hence, the corresponding hardware architecture and efficiency (in terms of area and energy) are still unknown.

*Contributions:* In this work, we present the first complete MIMO IDD receiver suitable for emerging communication standards such as IEEE 802.11n and WiMAX. For soft-in soft-out (SISO) MIMO detection, the SISO sphere-decoder (SD) implementation in [3] is used since it offers max-log maximum a posteriori (MAP) optimality with full exploitation of MIMO spatial diversity. Complexity can be reduced at run time to take advantage of favourable channel conditions or relaxed error rate requirements. Channel decoding is performed by a decoder for quasi-cyclic (QC) LDPC codes [4], which have excellent error-correction capabilities and are included in various communication standards. As shown in Fig. 2, the scalable architecture achieves communication performance gains up to 2.5 dB at $I = 4$ iterations over a non-iterative receiver ($I = 1$) at low SNR, for a target block error rate (BLER) of 1 %. At high SNR the throughput exceeds 1 Gbit/s with an energy well below 1 nJ/bit, almost equally distributed between detector and decoder.

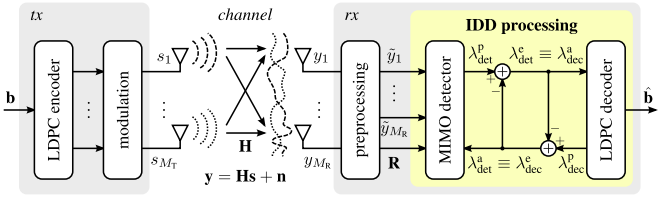


Fig. 1. MIMO IDD system model.

## II. System Architecture

The core of the MIMO IDD receiver comprises two processing elements (PEs), the MIMO detector and the channel decoder, which exchange L-values through a shared L-memory (Fig. 3). The two PEs operate on different granularities: detection is performed symbol-wise by demapping each $2^Q$-QAM modulated received vector $\tilde{\mathbf{y}}$ to $M_{\mathrm{T}}Q$ soft bits $\{\lambda^{\mathrm{e}}\}$; decoding operates on an entire codeblock (CB) of $N_{\mathrm{CB}}$ bits. MIMO detection and channel decoding take turns in processing each CB, resulting in an inefficient (50%) utilization of

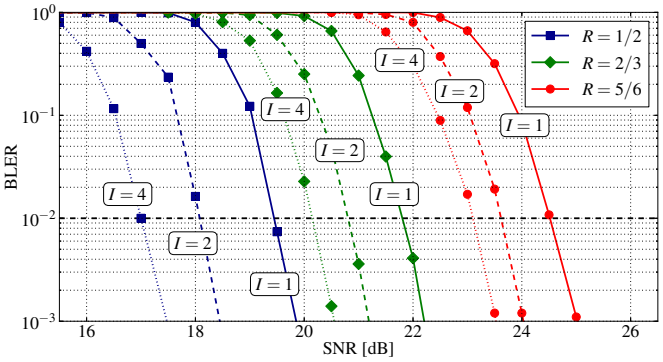


Fig. 2. Performance for 4×4 64-QAM with 802.11n LDPC codes (block length 1944) in an i.i.d. Rayleigh-fading channel (assumed perfectly known).

[1]Preprocessing for $M_{\mathrm{T}}$ transmit and $M_{\mathrm{R}}$ receive antennas includes sorted QR decomposition to compute the upper-triangular matrix $\mathbf{R} \in \mathbb{C}^{M_{\mathrm{T}} \times M_{\mathrm{T}}}$, with $\mathbf{H} = \mathbf{QR}$, $\mathbf{Q} \in \mathbb{C}^{M_{\mathrm{R}} \times M_{\mathrm{T}}}$ and $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$, and the vector $\tilde{\mathbf{y}} = \mathbf{Q}^H\mathbf{y}$.

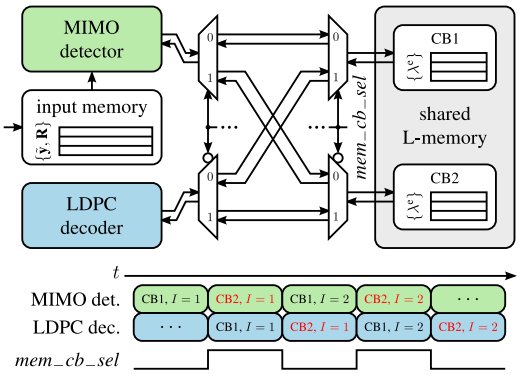


Fig. 3. System architecture with interleaved schedule.

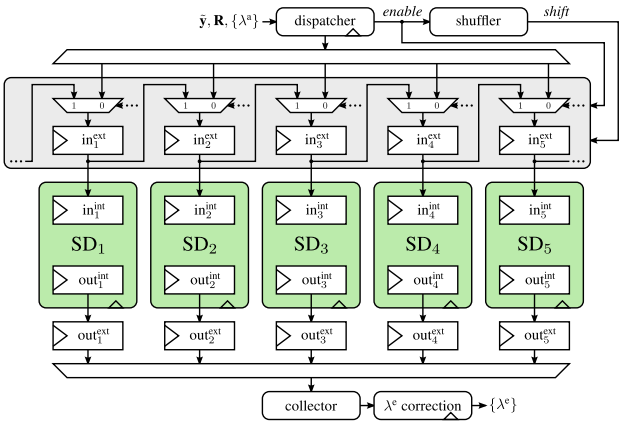


Fig. 4. Multi-core SD-based MIMO detector.

the PEs when only a single CB is considered. In this work, this limitation is overcome by always processing two CBs, stored in different L-memory blocks (CB1 and CB2), in an interleaved fashion, as shown in Fig. 3. After each iteration, the access to CB1 and CB2 is swapped transparently by switching the multiplexers between the PEs and the L-memory ports.

### A. Multi-Core MIMO Detector

The use of depth-first SISO sphere decoding presents several architectural challenges, not only in implementing the algorithm itself [3], [5], but also for system integration, mostly due to the variable run-time of SD and its high computational complexity at low SNR. To sustain a sufficient throughput multiple SD instances can be deployed in a scalable multi-core architecture (Fig. 4). Our reference implementation includes five SD cores, which can be deactivated selectively by clock gating as needed.

The double-buffered input of each SD unit is serviced by a *dispatcher* that exploits the processing time to preload the next received vector to be detected. At high SNR, the SD cores approach their minimum run-time of only $M_\text{T}+2$ cycles. Hence, to avoid idle times, the dispatcher and the input memory are designed to provide a complete data set for detecting a new received vector in each cycle. The input memory is split into multiple banks to achieve the required bandwidth. For each received vector requested by the dispatcher, an address generation unit computes the addresses for the different banks based on the vector index and based on the parameters $M_\text{T}$, $Q$, and $N_\text{CB}$. The data is then aggregated in a single packet and forwarded to the SD input buffers. A new read operation is initiated by the dispatcher whenever at least one input buffer is available. Unfortunately, the last vectors of a CB are occasionally buffered in front of a busy core while at least one other core is available. The resulting delay can be avoided by connecting the input buffers in a ring and shifting queued data from busy cores to idle cores (*shuffler* unit).

At the detector output, a *collector* forwards the results to the shared L-memory. To avoid stalls of SD cores, the collector acts as soon as an SD output buffer contains valid data, transferring a complete $\lambda^\text{e}$ vector per cycle. Since the SD run-time may vary for each vector, the output must be written back out-of-order based on the received vector index to avoid costly reordering operations. The SD run-time is controlled by *soft* (e.g., $\lambda^\text{e}$ clipping) and *hard* (e.g., a maximum number of cycles per vector or per CB) constraints [6], enforced by the dispatcher. Different scheduling policies are supported, such as maximum-first [6], ensuring at least successive interference cancellation (SIC) detection (corresponding to the minimum run-time) for all received vectors, and fair-share scheduling, with equal maximum run-time for all vectors. A post-processing $\lambda^\text{e}$ *correction* step improves performance in the presence of run-time constraints [6] by applying a precomputed correction function, stored in a programmable look-up table, to the L-values.

### B. Channel Decoder

QC-LDPC codes are used in many standards such as IEEE 802.11n and WiMAX because they combine good error-correction capabilities with a hardware-friendly, regular parity check matrix structure, that can be described by an $M_p \times N_p$ prototype matrix $\mathbf{H}_p$. Non-zero elements of $\mathbf{H}_p$ correspond to a cyclically-shifted $Z \times Z$ identity matrix. IEEE 802.11n for example defines different $\mathbf{H}_p$ (with $N_p = 24$ and variable $M_p$) corresponding to different subblock sizes $Z \in \{27, 54, 81\}$ ($Z_\text{MAX} = 81$) and code rates $R \in \{1/2, 2/3, 3/4, 5/6\}$.

The decoder used in this work [4] is run-time programmable and can decode any QC-LDPC code that fits into the available hardware resources. The corresponding architecture (Fig. 5) processes one $\mathbf{H}_p$ element per cycle. To this end, $Z$ L-values are read in parallel and are cyclically shifted according to the corresponding $\mathbf{H}_p$ entry. $Z$ parallel *node computation units* (NCUs) execute the layered offset-min-sum (OMS) algorithm

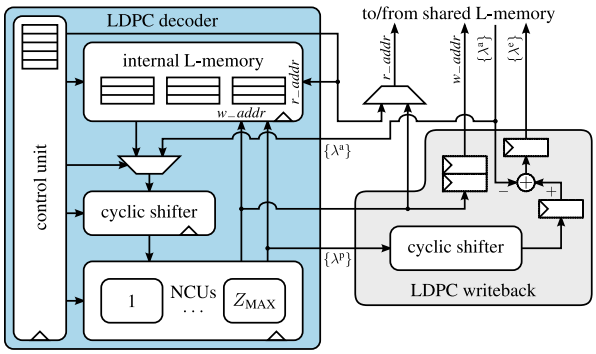


Fig. 5. LDPC decoder and writeback unit architecture.

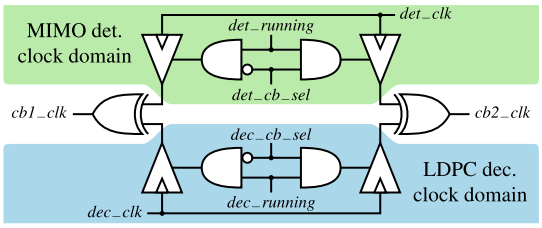


Fig. 6. Switching scheme for the shared L-memory clock.

to update the L-values. The internal storage subsystem employs standard-cell based memories [7] to achieve the required bandwidth and to reduce power consumption by fine-grained clock-gating. The *internal L-memory* is partitioned into three banks, each with $N_p = 24$ words and a word width of 27 L-values (each 5 bit-wide), selectively activated based on $Z$. In the last LDPC iteration, a *writeback* unit reads and aligns the $\{\lambda^{\mathrm{p}}\}$ computed by the decoder and the corresponding $\{\lambda^{\mathrm{a}}\}$ stored in the shared L-memory, computes the new $\{\lambda^{\mathrm{e}}\}$ and writes them back to the shared L-memory.

### C. Shared L-Memory Architecture

The detector and the decoder exchange data through two shared L-memory blocks (CB1 and CB2). Since both are accessed either by the detector or by the decoder exclusively, each of them has only one read and one write port (Fig. 3). The internal structure has to cope with the different access patterns of the PEs without hindering the throughput. While the decoder transfers vectors of $Z$ L-values, the detector operates on $M_{\mathrm{T}}Q$-wide $\lambda^{\mathrm{e}}$ vectors. The shared L-memory is designed to satisfy the maximum bandwidth, required by the decoder. Both CB1 and CB2 are structured in three banks with $N_p = 24$ words of 27 L-values (each 5 bit-wide). Their access ports match the internal L-memory of the decoder, which simply redirects to the external memory the first read and the last write access to each word (Fig. 5).

Since there is no integer relation between $Z$ and $M_{\mathrm{T}}Q$ and since these parameters are run-time configurable, detector accesses require an *alignment* unit to cyclically shift the $\lambda^{\mathrm{e}}$ vector and align it within the memory word. Moreover, detector accesses are frequently split across two memory words, even within the same bank: for instance, for $Z = 27$, $M_{\mathrm{T}} = 4$ and $Q = 6$, received vector 2 corresponds to L-values 25 to 27 in the first word and 1 to 21 in the second word of the first bank. Single-cycle access is enabled for such cases by a custom address decoder integrated into the employed latch-based standard-cell memories. At a small address decoding and alignment overhead, this approach effectively avoids multi-cycle accesses and stalls in the PEs which would affect the system throughput significantly.

To achieve the maximum possible throughput, the detector and the decoder can operate at different asynchronous clock frequencies. While control signals are synchronized by 3-stage synchronizers at the clock domain boundary, each of the two shared L-memory blocks is either synchronized with the detector or the decoder. The switching is realized by selecting one of the two clocks at the input of CB1 and CB2 as shown in Fig. 6. To prevent glitches, a control unit ensures that the CB select signals *det_cb_sel* and *dec_cb_sel* are complementary and only toggle when both PEs are done processing (i.e., both signals *det_running* and *dec_running* are low).

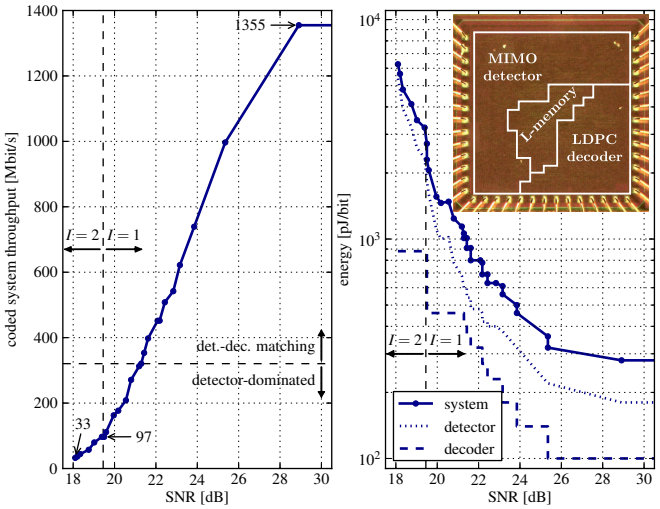


Fig. 7. Average system throughput and energy over SNR for a target BLER of 1 % (4×4 64-QAM, $N_{\mathrm{CB}} = 1944$, $R = 1/2$) and chip micrograph.

## III. Implementation Results

The proposed IDD architecture has been fabricated in a 65 nm low-power technology. The ASIC (Fig. 7) occupies a total core area of 2.78 mm$^2$, corresponding to 1.58 MGE (one gate equivalent GE corresponds to a 2-input drive-1 NAND gate). The MIMO detector accounts for 55 % of the area (872 kGE), with each SD core ranging between 140 and 145 kGE. The other main detector units are the input memory (70 kGE), the collector and $\lambda^{\mathrm{e}}$ correction unit (23 kGE) and the alignment unit (23 kGE). The LDPC decoder, with the write-back unit, and the shared L-memory occupy 28 % (447 kGE) and 13 % (210 kGE) of the total area, respectively. The maximum clock frequencies have been measured independently for the two PEs. At nominal supply voltage $V_{\mathrm{dd}} = 1.2$ V, the detector achieves 135 MHz[2] and the decoder 299 MHz.

Fig. 7 shows the average coded throughput and energy consumption over SNR of the complete IDD system for a configuration with 4×4 64-QAM, $N_{\mathrm{CB}} = 1944$ and $R = 1/2$. The run-time constraints of SD, $I$ and the number of LDPC inner iterations $I_{\mathrm{LDPC}}$ are adjusted to achieve a target BLER of 1 % at the highest system throughput, which increases roughly linearly with the SNR. For $I = 2$ the detector average run-time per iteration slightly increases with respect to $I = 1$ due to the lower SNR; moreover, the system throughput scales with $1/I$, resulting in different slopes for $I = 2$ and $I = 1$. Up to 21 dB the detector is slower than the decoder (with $I_{\mathrm{LDPC}} = 10$) and hence determines the throughput. In this regime voltage scaling could be exploited to reduce the throughput gap, increasing the detector $V_{\mathrm{dd}}$ for a higher throughput (up to 24 % at $V_{\mathrm{dd}} = 1.4$ V) and reducing the decoder $V_{\mathrm{dd}}$ to save energy (up to 30 % at $V_{\mathrm{dd}} = 1.0$ V).

[2]Due to area constraints, the IO pads were placed only on three sides of the chip, leading to an IR drop on the remaining side and a 20 % degradation of the detector frequency; with only one core active at a time, the IR drop decreases and the maximum frequency matches post-layout results (169 MHz).

TABLE I
MIMO DETECTOR COMPARISON

| | | This work | [2] | [8] | [9] |
|---|---|---|---|---|---|
| Number of antennas | | $\leq 4 \times 4$ | $\leq 4 \times 4$ | $\leq 4 \times 4$ | $4 \times 4$ |
| Modulation order | | $\leq 64$ | $\leq 64$ | $\leq 64$ | 64 |
| Iterative MIMO decoding | | yes | yes | no | no |
| CMOS tech. [nm] / $V_{dd}$ [V] | | 65/1.2 | 90/1.2 | 65/1.2 | 130/1.3 |
| Area [kGE] | | 872[a] | 410 | 215[a] | 114[a] |
| Uncoded throughput [Mbit/s] | SISO, 2 its. | 66 | 378[b] | - | - |
| | soft-out | 194 | 757[b] | 296[b] | - |
| | hard-out | 1251 | 757 | 807[c] | 655[b] |
| | SIC | 2710 | 757 | 2000 | 655 |
| Area efficiency [Mbit/s/kGE] | SISO, 2 its. | 0.08 | 0.92[b] | - | - |
| | soft-out | 0.22 | 1.85[b] | 1.38[b] | - |
| | hard-out | 1.43 | 1.85 | 3.75[c] | 5.75[b] |
| | SIC | 3.11 | 1.85 | 9.30 | 5.75 |
| Energy [pJ/bit] | SISO, 2 its. | 2690 | 500[b] | - | - |
| | soft-out | 920 | 250[b] | 128[b] | - |
| | hard-out | 180 | 250 | 47[c] | 200[b] |
| | SIC | 90 | 250 | 19 | 200 |

[a] Required QRD not included because not executed at symbol rate.
[b] Suboptimal performance.
[c] This operating point [8] is assumed to be close to hard-out ML performance in absence of more specific simulation data.

Above 21 dB the detector is fast enough to match the decoder throughput, which is adjusted by decreasing $I_{\text{LDPC}}$ as the SNR increases. In this operational range, the energy consumption of the two components is similar with a slight prevalence of the detector, which consumes 50 % to 65 % of the total energy.

A comparison with literature is difficult since typically the focus is either on a single PE or on the complete baseband with suboptimal receivers. Tab. I compares our SISO detector with other detector implementations. Four cases are considered: max-log-MAP optimal performance with $I = 1$ (soft-out) and $I = 2$ (SISO, 2 its.), corresponding to the highest detection effort; hard-out maximum-likelihood (ML) and SIC detection, with worse performance, but also much lower complexity. Our implementation is the only one to achieve max-log-MAP optimal performance and with support for IDD, with the corresponding area and energy costs. The detector in [2] closes the performance gap to SISO sphere decoding (1.5 dB for $I = 1$ and close to 1 dB for $I = 2$, with the same setup used for Fig. 2 and $R = 1/2$ at a BLER of 1 %), however, only under certain conditions and after several iterations [3]. Furthermore, the SD run-time constraints can be configured to perform hard-out ML or SIC detection. In such scenarios, the energy efficiency of our detector is in the range of the implementations in [8] and [9], which do not have to cope with the complexity of IDD and show a gap of 1 dB or more from the respective optimal performance (max-log-MAP with $I = 1$ for [8] and ML for [9]).

Tab. II compares different LDPC decoders and shows the high efficiency, especially in terms of area, achieved in this work with respect to state-of-the-art designs. By adjusting $I_{\text{LDPC}}$, the decoder also provides a mean to trade off performance and energy efficiency. Therefore, the IDD receiver combining the SD detector and the LDPC decoder is essentially *energy proportional*, since the design spends only the energy necessary to achieve the required performance in a given scenario.

TABLE II
LDPC DECODER COMPARISON

| | This work | [8] | [10] | [11] |
|---|---|---|---|---|
| Max. block length | 1944 | not spec. | 2304 | 2304 |
| CMOS tech. [nm] / $V_{dd}$ [V] | 65/1.2 | 65/1.2 | 65/1.2 | 130/1.2 |
| Area [$mm^2$] | 0.78 | 3.60 | 3.36 | 3.03 |
| Coded throughput [Mbit/s] | 586[a] | 235[b] | 880[a] | 728[a] |
| Area eff. [Mbit/s/$mm^2$] | 751 | 65 | 262 | 240 |
| Energy eff. [pJ/bit/iteration] | 21 | 156 | 13 | 47 |

[a] Maximum block length, code rate 5/6 and 10 iterations.
[b] Block length 768 bit, code rate 3/4 and 10 iterations.

## IV. CONCLUSIONS

We have shown the first complete architecture and silicon implementation of MIMO IDD, capable of extending the operating range of a wireless communication system towards channel capacity. Beside demonstrating the feasibility of IDD in a practical system, the energy-proportional ASIC achieves high throughput and energy efficiency in the operating range typically covered by non-iterative and suboptimal receivers.

## ACKNOWLEDGEMENTS

The authors thank C. Roth for the decoder design and the Microelectronics Design Center (ETH Zürich) for the support in the chip testing. This work has been supported by the Ultra High-Speed Mobile Information and Communication (UMIC) Research Centre, RWTH Aachen University.

## REFERENCES

[1] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding using soft feedback," *IET Electron. Lett.*, vol. 34, no. 10, pp. 942–943, May 1998.
[2] 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, Jul. 2011.
[3] F. Borlenghi *et al.*, "A 772 Mbit/s 8.81 bit/nJ 90 nm CMOS soft-input soft-output sphere decoder," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2011, pp. 297–300.
[4] C. Roth *et al.*, "A 15.8 pJ/bit/iter quasi-cyclic LDPC decoder for IEEE 802.11n in 90 nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2010, pp. 1–4.
[5] E. M. Witte *et al.*, "A scalable VLSI architecture for soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Circuits Syst. II*, vol. 57, no. 9, pp. 706–710, Sep. 2010.
[6] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4827–4842, Oct. 2010.
[7] P. Meinerzhagen, C. Roth, and A. Burg, "Towards generic low-power area-efficient standard cell based memory architectures," in *Proc. IEEE Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2010, pp. 129–132.
[8] M. Winter *et al.*, "A 335 Mbit/s 3.9 $mm^2$ 65 nm CMOS flexible MIMO detection-decoding engine achieving 4G wireless data rates," in *Dig. Tech. Papers, IEEE ISSCC*, Feb. 2012, pp. 216–218.
[9] M. Shabany and P. G. Gulak, "A 0.13 µm CMOS 655 Mbit/s 4×4 64-QAM k-best MIMO detector," in *Dig. Tech. Papers, IEEE ISSCC*, Feb. 2009, pp. 256–257a.
[10] X. Peng *et al.*, "A 115 mW 1 Gbit/s QC-LDPC decoder ASIC for WiMAX in 65 nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2011, pp. 317–320.
[11] B. Xiang *et al.*, "An 847-955 Mbit/s 342-397 mW dual-path fully-overlapped QC-LDPC decoder for WiMAX system in 0.13 µm CMOS," *IEEE J. Solid-State Circuits*, vol. 46, no. 6, pp. 1416–1432, Jun. 2011.