A 15.8 pJ/bit/iter Quasi-Cyclic LDPC Decoder for IEEE 802.11n in 90 nm CMOS

C. Roth*, P. Meinerzhagen*, C. Studer†, and A. Burg*

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

†Communication Technology Laboratory, ETH Zurich, 8092 Zurich, Switzerland
e-mail: studerc@nari.ee.ethz.ch

***Abstract*—We present a low-power quasi-cyclic (QC) low density parity check (LDPC) decoder that meets the throughput requirements of the highest-rate (600 Mbps) modes of the IEEE 802.11n WLAN standard. The design is based on the layered offset-min-sum algorithm and is runtime-programmable to process different code matrices (including all rates and block lengths specified by IEEE 802.11n). The register-transfer-level implementation has been optimized for best energy efficiency. The corresponding 90 nm CMOS ASIC has a core area of 1.77 mm^2 and achieves a maximum throughput of 680 Mbps at 346 MHz clock frequency and 10 decoding iterations. The measured energy efficiency is 15.8 pJ/bit/iteration at a nominal operating voltage of 1.0 V.**

I. Introduction

The high throughput and quality-of-service requirements of today's wireless communication systems demand highly reliable communication links. Considering the hostile environments these systems operate in, employing high-performance error-correction coding is inevitable to meet these stringent performance requirements. In this context, low density parity check (LDPC) [1] codes are attractive due to their excellent error-correction capabilities. Especially for wireless communication systems, quasi-cyclic (QC) LDPC codes have attracted significant attention since their regular parity check matrices facilitate hardware implementation and allow to easily adjust block length and code rate. QC-LDPC codes are employed in, e.g., IEEE 802.16e, DVB-S2, and IEEE 802.11n [2]. For all of these standards, energy efficiency is a very important design objective. For IEEE 802.11n in particular, this objective must be met at a high throughput (up to 600 Mbps) and for 12 different code configurations.

This paper presents a low-power QC-LDPC decoder ASIC for IEEE 802.11n based on the layered offset-min-sum (OMS) algorithm. Layered decoding [3] converges approximately twice as fast as classical message passing with a flooding schedule and the OMS algorithm [4] provides significant complexity reduction compared to the sum-product algorithm with only negligible performance loss. The described register-transfer-level architecture reduces active power consumption by minimizing the cost (in terms of power) and the number of memory access cycles and by reducing switching activity in the circuit. Furthermore, a low-complexity early-termination procedure avoids redundant decoding iterations with only very little additional hardware. Finally, the optimization for high throughput allows to further reduce power consumption through adaptive voltage scaling for devices that do not support the highest-rate modes of the standard or when rate-adaptation chooses a modulation-coding scheme with a lower-rate throughput.

II. QC-LDPC Decoder Architecture

An LDPC code is defined by a sparse $M \times N$ parity check matrix $\mathbf{H}$. Each row in that matrix describes a parity check equation and the columns are associated with the received code bits. The nth bit participates in the mth parity check if $[\mathbf{H}]_{m,n} = 1$. In a graphical representation (Tanner Graph) of the code, M check nodes represent the parity check equations and N variable nodes represent the code bits. A variable node connects to a certain check node if the associated bit participates in the parity check associated with that specific check node. LDPC codes are decoded iteratively by performing message passing along the edges of the graph. We denote the messages going from variable nodes to check nodes as Q-messages and the messages exchanged in the other direction as R-messages. An L-value is associated with each variable node representing the reliability-information about the corresponding code bit in form of an estimate of the a posteriori log-likelihood ratio (LLR). These estimates are initially received from the baseband receiver and improve with each decoding iteration.

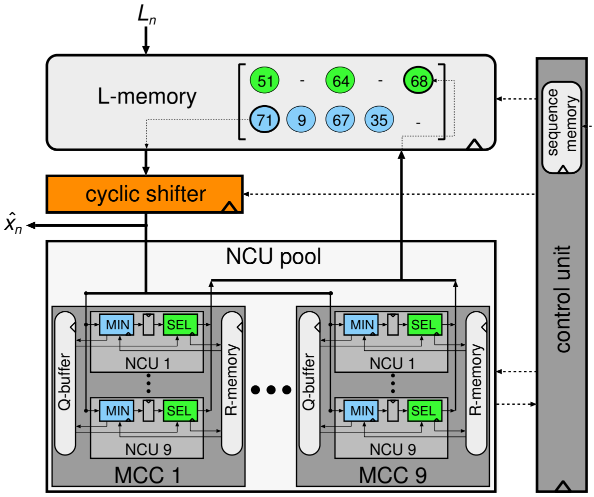

Fig. 1. High-level QC-LDPC decoder architecture.

QC-LDPC codes are defined by a parity check matrix that can be described by an $M_p \times N_p$ matrix prototype $\mathbf{H}_p$ where each entry $[\mathbf{H}_p]_{i,j}$ of $\mathbf{H}_p$ is either associated with a $Z \times Z$ identity matrix that is cyclically shifted by $[\mathbf{H}_p]_{i,j}$ or with an all-zero matrix if $[\mathbf{H}_p]_{i,j} =$ '–'. Each row of $\mathbf{H}_p$ thus corresponds to a layer of Z parity check equations and each column is associated with Z consecutive code bits (L-values). There are 12 matrix prototypes specified by the IEEE 802.11n standard [2] corresponding to three different block lengths $Z \in \{27, 52, 81\}$ and four different code rates $r \in \{1/2, 2/3, 3/4, 5/6\}$. All matrix prototypes have the same number of columns ($N_p = 24$) but varying numbers of rows according to the code rate. The layered message passing algorithm implemented by the architecture in Fig. 1 updates for each layer all L-values involved in the layer concurrently. To this end, Z node computation units (NCUs) sequentially perform the OMS algorithm for each parity check equation in the current layer. The QC structure of the parity check matrix ensures that there is no data exchange

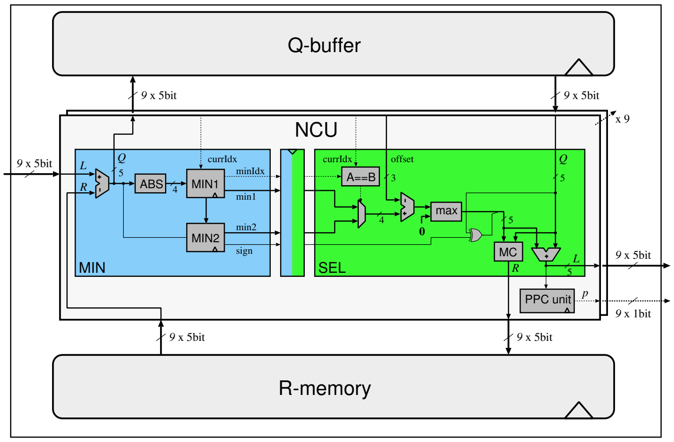

Fig. 2. Architecture of the NCU embedded in a macro computation cell.

among the individual NCUs since the parity checks in the same layer never involve the same code bits. Inside an NCU, new R-messages are iteratively computed based on all previous L-values and R-messages associated with the current parity check equation. The updated R-messages and old L-values are then combined to form the updated L-values, thereby improving the LLR estimates of the corresponding code bits.

The decoding algorithm starts by initializing the L-memory with the LLRs of the code bits based on the channel outputs from the baseband receiver. Then, each layer (row) of $\mathbf{H}_p$ is processed sequentially. This process is repeated until a predefined number of iterations has been reached or until the implemented early-termination procedure stops the decoding. For each layer i, the N_p groups of Z code bits (L-values), corresponding to the columns in $\mathbf{H}_p$, for which $[\mathbf{H}_p]_{i,j} \neq$ '–' are processed sequentially. To this end, Z L-values of successive code bits are fetched from the L-memory in each cycle and shifted by the cyclic shifter (CS) unit in order to provide each NCU with the proper L-value as defined by the corresponding entries of $\mathbf{H}_p$. The logic is controlled by a control sequence kept in a programmable sequence memory. This feature allows to reconfigure the decoder to any QC-LDPC code that fits into the allocated hardware resources.

A. Node Computation Units

The implementation of the NCU is detailed in Fig. 2. To support the largest block length specified by IEEE 802.11n, there are $Z_{\max} = 81$ NCUs. Depending on the code configuration, only $Z < Z_{\max}$ units are active. The computation in the NCU happens in two phases. In the first phase, the MIN unit iteratively computes the minimum and second minimum of the corresponding Q-messages based on the previous L-values and R-messages. At the end of the layer, the MIN unit forwards the minima to the SEL unit which updates both the R-messages and L-values in the second phase. To enable high throughput, the operation of the MIN and SEL units are pipelined and interleaved, i.e., the two units process subsequent layers concurrently. The resulting data dependencies must be considered by choosing the order in which the columns are processed appropriately to avoid the need to stall the pipeline.

MIN unit: The MIN unit first computes the current Q-message on-the-fly as the difference from the L-value L coming from the CS unit and the R-message R retrieved from the R-memory. From the absolute values of the Q-messages, the MIN1 and MIN2 sub-units compute and store the minimum, the associated column index *minIdx* in $\mathbf{H}_p$ as well as the second minimum and the product of the sign-bits of the Q-messages. In addition, the Q-message is written into the

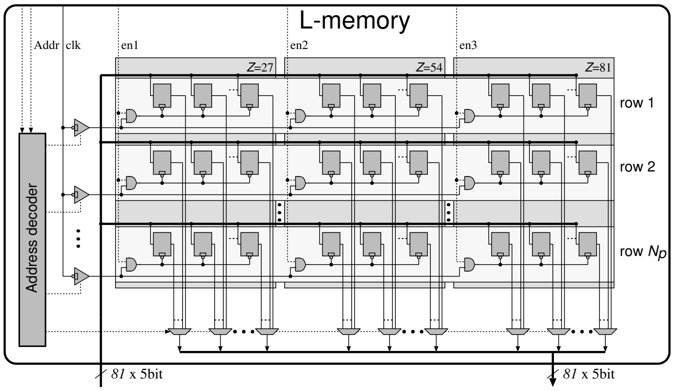

Fig. 3. L-memory architecture.

Q-buffer. At the end of the current layer, the results are latched into the pipeline register connecting the MIN unit to the SEL unit.

SEL unit: The circuit selects the second minimum provided by the MIN unit if the current column index *currIdx* is equal to *minIdx* and the minimum otherwise. In the OMS algorithm, an offset is then subtracted from the selected operand. The magnitude of the new R-message is obtained by a simple *max*-function and its sign is the result of the Q-message (retrieved from the Q-buffer) sign multiplied by the sign forwarded by the MIN unit. The new L-value is computed by adding the updated R-message to the Q-message. In order to limit the dynamic range of the L-values, we employ message clipping (MC) [5] to the R-messages. This optimization enables to reduce the word length of the L-values and Q-messages from 7 bit to 5 bit at a slight loss in error-correction performance of the decoder. In addition to the energy savings resulting from the reduced-size storage for L-values and Q-messages, this approach yields a 16 % and 19 % lower power consumption in the CS unit and NCU, respectively, according to post-layout power simulations compared to an otherwise identical implementation without MC. Overall, MC leads to a power reduction of 21 %. Furthermore, the messages and L-values are represented using sign-magnitude number format and all arithmetic operations are performed in this format. Power simulations show that this number format yields a total power reduction of 9.5 % compared to the case where 2's complement is employed.

PPC unit: The partial parity check (PPC) unit evaluates if the current parity check equation is satisfied. If this is the case, the output of the PPC unit, denoted as p in Fig. 2, is set to logic-0 at the end of the current layer. Combining the check bits from all active NCUs by a simple OR-operation results in a per-layer PPC bit that is logic-0 only if all parity check equations in the layer are satisfied. If a certain number of consecutive per-layer PPCs are correct, the decoding process is stopped. This novel early-termination approach is configurable and improves the energy efficiency of the decoder in the high-SNR regime (cf. Sec. III).

B. Cyclic Shifter

The CS unit corresponds to the area-efficient subset cyclic shifter described in [5]. Cyclic shifts are required when the MIN unit reads data from the Q-memory to route the Z individual L-values to the associated NCUs.

C. Storage

In QC-LDPC decoder implementations, a large fraction of the total dissipated power is consumed by the memories required to store the R-messages and L-values and to buffer the Q-messages. For example, in the circuit described in [5] storage macro cells consume 68% of the total power. Hence, the optimization of the storage itself and of the memory access rate is an important aspect of the low-power strategy. In order to minimize power per read/write access, we propose the memory architecture shown in Fig. 3. Based on the results presented in [6], standard cell based latches are employed as storage cells. To perform a write access, the address decoder produces a one-hot encoded signal that enables one specific clock gate to pulse only the clock of the row selected by the write address. In case of the L-memory, additional AND-gates allow to disable entire blocks of unused columns to improve energy efficiency when $Z < Z_{\max}$. At the memory output, a multiplexer based read logic is implemented. Power simulations show that this approach is more power- and area-efficient than a read logic based on tri-state buffers [6]. Due to the parallel operation of MIN and SEL units there are conditions where updated L-values can be fed back directly into the CS unit, thereby bypassing the L-memory and reducing the write access rate to the storage array. This reduction of write accesses is more efficient with denser matrix prototypes, which arise for higher-rate codes. For example, for the rate-$1/2$, $Z = 81$ code [2], our sequence generation algorithm yields a bypassing rate of 55 %, whereas for the rate-5/6, $Z = 81$ code, 90 % of the L-memory write accesses can be bypassed. The latter case reduces power consumption in the L-memory by 75 % and the overall power by 7.5 % compared to an otherwise identical reference implementation without this bypassing scheme.

As shown in Fig. 1, a constant number of NCUs is grouped into several macro computation cells (MCCs) in our design. The NCUs combined into an MCC share a common R-memory and common Q-buffer. The main motivation behind this partitioning is to enable a trade-off between data locality and circuit overhead as explained in the following. In one extreme case, the number of MCCs can be chosen equal to the number of NCUs, effectively allocating a small separate R-memory and Q-buffer for each NCU. This approach reflects the local data processing of the algorithm, but also involves a considerable circuit overhead due to replicated address decoders and clock gates. On the other extreme, sharing a single R-memory and Q-buffer among *all* NCUs minimizes circuit overhead but also further separates storage from the corresponding data processing. From a layout perspective, the good data locality implied by a large number of MCCs enables a more local structure, which significantly facilitates placement, routing, and clock tree generation for the storage arrays, which turns out to be associated with considerable overhead for routing and buffer insertion for large arrays. In order to quantify this, several layouts were created for varying numbers of MCCs. Our results show that the case without any partitioning (i.e., only a single R-memory and Q-buffer) yields a highly inefficient layout in terms of area and speed. For more fine-grained partitionings, the designs become more efficient until the circuit overhead starts to inflate the silicon area. Grouping 9 NCUs in an MCC as shown in Fig. 1 proves to yield a good trade-off.

III. ASIC Implementation Results

The described ASIC (see Fig. 4) has been implemented in 90 nm CMOS technology. The measured key figures of the circuit are summarized in Tbl. I. The core occupies a silicon area of 1.77 mm^2 with an active cell area of 398 kGE. Measurements at a supply voltage of 1.0 V show that a clock frequency of 346 MHz can be achieved, which translates into a maximum throughput of 680 Mbps (information bits) at 10 decoding iterations with the rate-5/6, $Z = 81$ code [2]. The error-correction performance in terms of frame error rate (FER) in an AWGN channel is shown in Fig. 5 for 5 and 10 iterations. The performance of the decoder is compared with the one of the ideal layered floating-point sum-product algorithm (SPA) at 30 iterations.

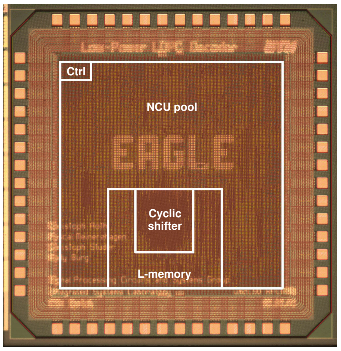

Fig. 4. Microphotograph of the fabricated ASIC.

TABLE I
Comparison of QC-LDPC decoder implementations

Publications	[7]	[8]	[9]	[5]	This work
Technology [nm]	180	90	130	180	90
V_{dd} [V]	1.8	1.0	1.2	1.8	1.0
Basis of results	post-layout estimations		ASIC measurements		
$Z_{\max}$	96	96	64	81	81
Core area [mm^2]	3.39	3.5	2.46	3.39	1.77
Max. throughput^a in [Mbps]	57 (113^c)	1667	115 (166^c)	390 (780^c)	679
Hardware eff.^{a} in [µm^2/Mbps]	59.8 (7.5^c)	2.1	21.5 (7.7^c)	8.7 (1.1^c)	2.6
Energy eff.^{b} in [pJ/bit/iter]	243 (37.5^c)	34.2	63.2 (30.4^c)	800 (124^c)	15.8

[a] at 10 iterations, $r = 5/6$.
[b] measured at nominal supply voltage.
[c] Technology scaling to 90 nm, $V_{dd} = 1.0$ V: $t_{pd} \sim 1/s$, $A \sim 1/s^2$, $P \sim 1/s \cdot (V'_{dd}/V_{dd})^2$.

The energy efficiency measurements are summarized in Tbl. II. All measurements were performed at $\mathrm{T} = 300$ K using typical stimuli in an AWGN scenario. The impact of the early-termination algorithm is shown at different SNRs for all rate-5/6 codes [2] for a maximum number of 10 iterations and 600 Mbps throughput. With increasing SNRs, the incorporated early-termination algorithm is more likely to stop decoding after a few iterations, which improves energy efficiency significantly. At 4 dB SNR, the energy consumption per bit is reduced by 58 % without any loss in terms of FER compared to the case where early-termination is disabled. Furthermore, for small block lengths (e.g., $Z = 27$), the inactive MCCs and unused columns in the L-memory as well as unused parts of pipeline registers can be turned off, which reduces power consumption considerably.

Fig. 6 shows detailed throughput and energy efficiency measurement results for the rate-5/6, $Z = 81$ code at 4 dB SNR when taking voltage scaling into account. For a fixed number of iterations, reducing the supply voltage enables to trade circuit speed for power. For example, the presented ASIC can be operated at 0.9 V supply voltage while still complying with the 600 Mbps mode of the standard at 10 decoding iterations, which results in roughly 20 % lower energy consumption per bit compared to operating the ASIC at its nominal supply voltage. Power simulations showed that scaling the voltage of a fast implementation is more energy-efficient than only tailoring the design to the required speed. Another way to improve energy efficiency is to trade decoding performance in terms of FER for lower energy per bit by reducing the number of decoding iterations.

TABLE II

Energy efficiency measurements for rate-$5/6$ codes at 10 decoding iterations, $V_{dd} = 1.0\,\mathrm{V}$, $f = 305\,\mathrm{MHz}$

Z	SNR	w/o early-termination			w/ early-termination[a]	
		mW	pJ/bit	%	pJ/bit	%
27	1	51.3	24.5	100	24.5	100
	4	53.6	25.6	100	10.8	42
	7	51.2	24.5	100	5.4	22
54	1	82.6	18.6	100	18.6	100
	4	88.1	19.9	100	8.4	42
	7	82.6	18.6	100	4.1	22
81	1	111.6	15.6	100	15.6	100
	4	114.5	16	100	6.7	42
	7	107	14.9	100	3.3	22

[a] Results are obtained by scaling the measurement results by the expected number of iterations.

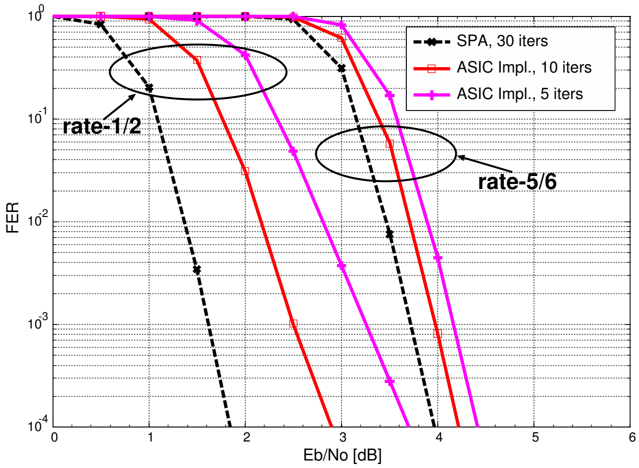

Fig. 5. Decoder frame-error rate (FER) performance comparison for $Z = 81$ codes [2] in an AWGN channel.

For example, decreasing the number of iterations from 10 to 5 at 600 Mbps target throughput enables a reduction of more than 50% in terms of energy per bit while the loss in SNR is only 0.2 dB as shown in Fig. 5.

A comparison to other QC-LDPC decoders is provided in Tbl. I. To account for differences in process technology we scale the results to 90 nm and 1.0 V supply voltage. Note that some of the referenced designs are optimized for the IEEE 802.16e standard instead of IEEE 802.11n considered in our implementation. Our decoder exhibits a 2.4 times and 1.9 times better energy efficiency than the decoders presented in [7] and [9], respectively, and our circuit is more hardware-efficient when taking technology scaling into account. Comparing our design to the highly parallel radix-4 architecture described in [8] shows that the more parallel computation approach can improve hardware efficiency but yields a lower energy efficiency. Compared to our previous work [5], which served as a reference design for the present implementation, we were able to improve energy efficiency by a factor of 7.8 at the cost of a loss in hardware efficiency. As opposed to the described decoder implementation, the reference design in [5] strives for reduced silicon area by relying on higher clock frequencies and by reducing the number of storage elements, which are implemented using area-efficient macro cells.

IV. Conclusion

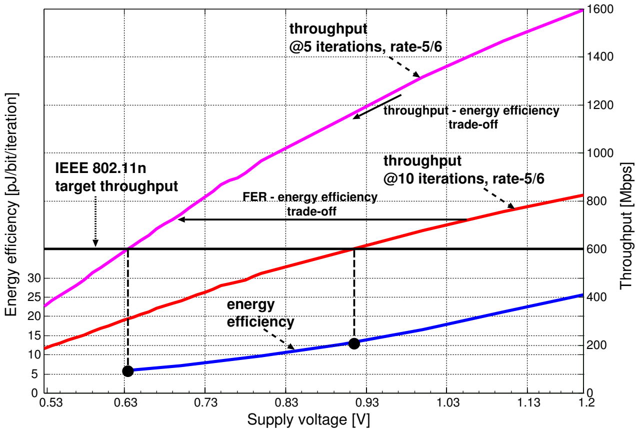

Fig. 6. Throughput and energy efficiency measurement results for the rate-5/6, $Z = 81$ code at 4 dB SNR.

We have described a low-power LDPC decoder ASIC that is fully compliant with the IEEE 802.11n standard and meets the throughput requirements of the highest-rate (600 Mbps) mode. The circuit achieves — to the best of our knowledge — the best energy efficiency across comparable designs reported in the open literature. In order to achieve this result, we have applied a variety of optimizations from the algorithm to the circuit level. In particular, we carefully partition the design to maintain data locality and pay attention to reducing the amount of storage and the number of memory access cycles. Furthermore, we employ fine-grained clock gating and prefer sign-magnitude over 2's complement number representation to reduce switching activity. Storage is implemented in a distributed fashion using latches. On the algorithm level, the use of early-termination further improves energy efficiency. The proposed algorithm uses per-layer partial parity checks to implement early-termination with minimal hardware overhead.

Acknowledgment

The authors would like to thank Prof. H. Kaeslin, F. Gürkaynak, B. Muheim, and Dr. N. Felber for their support during ASIC design and testing. The presented work was kindly supported by the Hasler Foundation and the Swiss National Science Foundation.

References

[1] R. Gallager, "Low density parity check codes," *Trans. of the IRE Prof. Group on Inf. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[2] *IEEE Unapproved Draft Std P802.11n/D11.0*, Jun. 2009.

[3] E. Sharon, S. Litsyn, and J. Goldberger, "An efficient message-passing schedule for LDPC decoding," in *Proc. 23rd IEEE Convention of Electrical and Electronics Engineers in Israel*, Sep. 2004, pp. 223–226.

[4] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X.-Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Trans. on Comm.*, vol. 53, no. 7, pp. 1288–1299, Aug. 2005.

[5] C. Studer, N. Preyss, C. Roth, and A. Burg, "Configurable high-throughput decoder architecture for quasi-cyclic LDPC codes," in *Proc. 42nd Asilomar Conf. on Signals, Systems and Computers*, Oct. 2008, pp. 1137–1142.

[6] P. Meinerzhagen, C. Roth, and A. Burg, "Towards generic low-power area-efficient standard cell based memory architectures," in *Proc. 53rd IEEE Int'l. Midwest Symp. on Circuits and Systems*, Aug. 2010, pp. 129–132.

[7] T.-C. Kuo and A. Willson, "A flexible decoder IC for WiMAX QC-LDPC codes," in *Proc. IEEE Custom Integrated Circuits Conf.*, Sep. 2008, pp. 527–530.

[8] Y. Sun and J. R. Cavallaro, "A low-power 1-Gbps reconfigurable LDPC decoder design for multiple 4G wireless standards," in *Proc. IEEE Int'l. SOC Conf.*, Sep. 2008, pp. 367–370.

[9] X.-Y. Shih, C.-Z. Zhan, and A.-Y. Wu, "A real-time programmable LDPC decoder chip for arbitrary QC-LDPC parity check matrices," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2009, pp. 369–372.