Configurable High-Throughput Decoder Architecture for Quasi-Cyclic LDPC Codes

C. Studer*, N. Preyss, C. Roth, and A. Burg*

*Integrated Systems Laboratory
ETH Zurich, 8092 Zurich, Switzerland
email: {studer, apburg}@iis.ee.ethz.ch

***Abstract*— We describe a fully reconfigurable low-density parity check (LDPC) decoder for quasi-cyclic (QC) codes. The proposed hardware architecture is able to decode virtually any QC-LDPC code that fits into the allocated memories while achieving high decoding throughput. Our VLSI implementation has been optimized for the IEEE 802.11n standard and achieves a throughput of 780 Mbit/s with a core area of 3.39 mm^2 in 0.18 µm CMOS technology.**

I. Introduction

The quality-of-service and throughput requirements of modern wireless communication systems require high-performance error-correction schemes. Low-density parity check (LDPC) codes [1] are promising candidates for next-generation wireless communication standards due to the excellent error-correction capabilities. In particular, quasi-cyclic (QC) LDPC codes [2], [3] offer high decoding throughput at low implementation complexity, e.g., [4]–[6], and have been considered in different wireless standards, e.g., DVB-S2 [7], IEEE 802.16 [8], and IEEE 802.11n [9]. The growing use of QC-LDPC codes asks for a single configurable decoder architecture that can easily be scaled (at compile time) to different performance and complexity requirements.

Contributions: In this paper, we describe a configurable high-throughput QC-LDPC decoder architecture based on layered message passing [10]. We present a novel technique to trade memory consumption for (error-rate) performance at design-time of the decoder and propose a parallel decoder architecture that employs flexible cyclic shifters in combination with a configurable sequencer, which enables to decode virtually any QC-LDPC code that fits into the allocated memories. This feature renders the design suitable for multi-standard radio applications, but the specific implementation reported in this paper has been optimized for IEEE 802.11n [9]. The ASIC was fabricated in 0.18 µm CMOS technology. We provide power-efficiency measurement results and compare our ASIC implementation with existing QC-LDPC decoders.

Outline: The remainder of this paper is organized as follows. Section II describes QC-LDPC codes and the employed LDPC decoding algorithm. The configurable VLSI architecture is described in Section III, corresponding implementation results are given in Section IV, and we conclude in Section V.

II. Quasi-Cyclic LDPC Codes and Decoding

LDPC [1] codes are linear block codes satisfying

$$\mathbf{H}\mathbf{x} = \mathbf{0}_{M\times 1} \tag{1}$$

in GF(2), where $\mathbf{x}$ is the N-dimensional binary-valued data vector, $\mathbf{0}_{M\times 1}$ denotes an M-dimensional all-zero vector, and $\mathbf{H}$ is a sparse binary-valued $M \times N$ parity check matrix. QC-LDPC codes are described by a $M_p \times N_p$ LDPC matrix prototype $\mathbf{H}_p$. The parity check matrix in (1) is constructed from $\mathbf{H}_p$ by replacing each entry by a $Z \times Z$ cyclic-shift matrix $\mathbf{P}^c$ (i.e., $M = M_p Z$ and $N = N_p Z$), where c is equal to the corresponding entries in $\mathbf{H}_p$. The cyclic-shift matrices are defined as $\mathbf{P}^c = \prod_{i=1}^{c} \mathbf{P}^1$ (for $c > 0$) with

$$\left[\mathbf{P}^1\right]_{i,j} = \begin{cases} 1, & (i \bmod Z) + 1 = j \\ 0, & \text{otherwise} \end{cases}$$

$\mathbf{P}^0 = \mathbf{I}_Z$, and $\mathbf{P}^- = \mathbf{0}_{Z\times Z}$. Note that all entries of $\mathbf{H}_p$ satisfy $[\mathbf{H}_p]_{m,n} < Z$ $(\forall m, n)$. For example, the rate-$5/6$, $Z = 81$ QC-LDPC matrix prototype of the IEEE 802.11n standard [9] is defined as

$$\mathbf{H}_p = \begin{bmatrix} 13 & 48 & 80 & 66 & 4 & 74 & 7 & 30 & 76 & 52 & 37 & 60 & - & 49 & 73 & 31 & 74 & 73 & 23 & - & 1 & 0 & - & - \\ 69 & 63 & 74 & 56 & 64 & 77 & 57 & 65 & 6 & 16 & 51 & - & 64 & - & 8 & 9 & 48 & 62 & 54 & 27 & - & 0 & 0 & - \\ 51 & 15 & 0 & 80 & 24 & 25 & 42 & 54 & 44 & 71 & 71 & 9 & 67 & 35 & - & 58 & - & 29 & - & 53 & 0 & - & 0 & 0 \\ 16 & 29 & 36 & 41 & 44 & 56 & 59 & 37 & 50 & 24 & - & 65 & 4 & 65 & 52 & - & 4 & - & 73 & 52 & 1 & - & - & 0 \end{bmatrix}.$$

A. Decoding of LDPC Codes

The parity check in (1) can be represented as a bipartite graph consisting of N variable nodes and M check nodes. Variable nodes are associated with transmitted bits x_n $(n = 1, 2, \ldots, N)$. The mth parity check node is connected to the nth variable node if $[\mathbf{H}]_{m,n} = 1$. LDPC decoding can be represented as message passing (MP) on the bipartite graph [11]. The standard MP schedule consists of two phases and is also known as flooding MP [10]. In the first phase, all messages from the variable to the check nodes (denoted by Q-messages) are computed with the aid of log-likelihood ratios (LLRs)

$$L_n \triangleq \log\left(\frac{\mathrm{P}[x_n = 0|\mathbf{y}]}{\mathrm{P}[x_n = 1|\mathbf{y}]}\right), \quad n = 1, 2, \ldots, N \tag{2}$$

where $\mathbf{y}$ denotes the channel observation. In the second phase, all messages from the check to the variable nodes (denoted by R-messages) are computed. Both phases are repeated until (1) is satisfied (i.e., the algorithm converges to a valid code-word)

or a maximum number of iterations I has been reached. Note that flooding MP is frequently used in the literature, e.g., [12], [13], but has three major disadvantages: i) both phases require a different set of arithmetic operations, ii) both message types (Q and R) as well as the input LLRs (2) need to be stored, and iii) the convergence behavior is relatively slow [10].

B. Layered Offset Min-Sum LDPC Decoding

The decoding algorithm used in this paper avoids the drawbacks of flooding MP and is based on layered LDPC decoding [10], [14] with offset min-sum (OMS) MP [15]. Furthermore, the implemented decoding algorithm exploits the value-reuse properties of OMS as described in [5], [6] and employs a novel technique called message clipping (MC), which allows to trade memory requirements for (error-rate) performance.

The algorithm described in the following is summarized in Alg. 1. Note that layered QC-LDPC decoding only requires memory for two message types [10], [14], denoted by the Z-dimensional vectors $\mathbf{q}_n^i$ (i.e., the Q-values at the variable nodes) and the R-messages from check to variable nodes $\mathbf{r}_{m,n}^i$. All vector operations in Alg. 1 are performed element-wise, except for the cyclic shifts (i.e., all matrix-vector multiplications).

Initialization: On lines 1-2 of Alg. 1, the $\mathbf{q}_n^0$ ($\forall n$) values are initialized by the LLRs (2) and all messages $\mathbf{r}_{m,n}^0$ ($\forall m, n$) are set to zero. Note that no separate storage for the input LLRs is needed for layered LDPC decoding. The algorithm performs a maximum number of I iterations (lines 3-19) and operates row-wise (for each row m of $\mathbf{H}_p$). Message computation is then performed iteratively for the columns $n \in \mathcal{N}_m$ of $\mathbf{H}_p$ for which $\left[\mathbf{H}_p\right]_{m,n} \neq '-'$ (see line 6 and 12 Alg. 1), i.e.,

$$\mathcal{N}_m = \left\{ n \in \mathbb{Z} \mid \left[\mathbf{H}_p\right]_{m,n} \neq '-' \right\}. \tag{3}$$

Message computation of layered MP can be divided into two phases: the first is called MIN phase (lines 6-11) and the second is called SEL phase (lines 12-17).

MIN Phase: The $\mathbf{r}_{m,n}^{i-1}$-message is subtracted from a cyclically-shifted version (by $c = \left[\mathbf{H}_p\right]_{m,n}$) of $\mathbf{q}_n^{i-1}$ and stored in a temporary vector $\mathbf{t}_n$ (cf. line 6). OMS [15] and its value-reuse properties [5], [6] only require to compute the minimum $\mathbf{m}_1$ and the second minimum $\mathbf{m}_2$ for each entry of $\mathbf{t}_n$ ($\forall n$). Both minima are being computed iteratively as shown on lines 8 and 9, where the minimum on line 8 returns the minimum of $\mathbf{m}_1$ or $\mathbf{t}_n$, a temporary vector $\mathbf{x}$ which contains the values not corresponding to the minimum (i.e., the second-lowest values), and an index vector where $[\mathbf{v}]_k$ is set to the current index n if a new minimum (i.e., corresponding to $[\mathbf{t}_n]_k$) has been found. The vector $\mathbf{s}$ contains Z signs, which are used in the SEL phase.

SEL Phase: In this phase, the new Q and R messages are computed iteratively for $n = 1, 2, \ldots, N_b$. To this end, the temporary vector $\mathbf{t}_n = \mathbf{P}^c \mathbf{q}_n^{k-1} - \mathbf{r}_{m,n}^{k-1}$ is computed first (line 14). Then, the sel-function (line 14) compares each of the Z indices $[\mathbf{v}]_k$ (for $k = 1, 2, \ldots, Z$) with n and yields the corresponding entry of $[\mathbf{m}_1]_k$ if $\mathbf{v}_k = n$ and $[\mathbf{m}_2]_k$ otherwise. The new $\mathbf{r}_{m,n}^i$-vector is computed by using OMS MP [15] (cf. line 15) with $\underline{\beta} = \beta \cdot \mathbf{I}_{Z\times 1}$, where only one offset value β is used for all Z entries. The new $\mathbf{q}_{m,n}^i$ vector is computed on line 17 and cyclically shifted using the inverse rotation $c' = Z - c$, such that $\mathbf{P}^c \mathbf{P}^{c'} = \mathbf{I}_{Z\times Z}$.

C. Message Clipping

Since layered LDPC decoding performs updates of the $\mathbf{q}_{m,n}^i$-vector using the results of the previous iteration (cf. line 16), the dynamic range of the Q-messages is, in general, very large. In order to reduce the amount of bits required to store all $\mathbf{q}_{m,n}^i$-vectors ($\forall n, m$), the maximum magnitude of the Q-values can be clipped. Unfortunately, this straightforward approach yields poor error-rate performance. In order to combat this problem, we propose a novel approach referred to as message clipping (MC) in the following. Instead of clipping the Q-values, the R-messages are clipped (on line 15) according to

$$\mathrm{clip}(\mathbf{r}, \mathbf{t}) = \max\left\{ \min\{\mathbf{r}, Q_{\max} - \mathbf{t}\}, -Q_{\max} - \mathbf{t} \right\}$$

which ensures that that $\left|[\mathbf{q}_n^i]_k\right| \leq Q_{\max}$ ($\forall n, k$) and $Q_{\max}$ denotes the MC parameter. This approach offers a trade-off between memory consumption (to store the Q-messages) and error-rate performance (corresponding simulation results are shown in Section III-E).

III. VLSI Architecture

An overview of the configurable QC-LDPC decoder architecture is shown in Fig. 1. The architecture consists of two memories, i.e., one for the Q-values and one for the R-messages, a cyclic shifter (CS), and a pool of node computation units (NCUs). To enable reconfigurability for different LDPC matrix prototypes, the architecture contains a configurable control unit and a sequence (SEQ) memory.

Algorithm 1 QC L-OMS LDPC decoder with MC

1: $\mathbf{r}_{m,n}^0 \leftarrow \mathbf{0}_{Z\times 1}$, $n = 1, 2, \ldots, N_b$, $m = 1, 2, \ldots, M_b$
2: $\mathbf{q}_n^0 \leftarrow \left[L_{(n-1)Z+1} \; L_{(n-1)Z+2} \; \cdots \; L_{nZ} \right]^T$, $\forall n$
3: **for** $i = 1, 2, \ldots, I$ **do**
4: **for** $m = 1, 2, \ldots, M_b$ **do**
5: $\mathbf{m}_1 \leftarrow \infty \cdot \mathbf{I}_{Z\times 1}$, $\mathbf{m}_2 \leftarrow \infty \cdot \mathbf{I}_{Z\times 1}$, $\mathbf{s} \leftarrow \mathbf{I}_{Z\times 1}$
6: **for** $n \in \mathcal{N}_m$ **do**
7: $c = \left[\mathbf{H}_p\right]_{m,n}$, $\mathbf{t}_n \leftarrow \mathbf{P}^c \mathbf{q}_n^{i-1} - \mathbf{r}_{m,n}^{i-1}$
8: $[\mathbf{m}_1, \mathbf{x}, \mathbf{v}] \leftarrow \min\left\{\mathbf{m}_1, |\mathbf{t}_n|\right\}$
9: $\mathbf{m}_2 \leftarrow \min\{\mathbf{m}_2, \mathbf{x}\}$
10: $\mathbf{s} \leftarrow \mathbf{s} \cdot \mathrm{sign}(\mathbf{t})$
11: **end for**
12: **for** $n \in \mathcal{N}_m$ **do**
13: $c = \left[\mathbf{H}_p\right]_{m,n}$, $\mathbf{t}_n \leftarrow \mathbf{P}^c \mathbf{q}_n^{i-1} - \mathbf{r}_{m,n}^{i-1}$
14: $\mathbf{m} \leftarrow \mathrm{sel}(\mathbf{m}_1, \mathbf{m}_2, \mathbf{v}, n)$
15: $\mathbf{r}_{m,n}^i \leftarrow \mathrm{clip}\big(\mathbf{s} \cdot \mathrm{sign}\big(\mathbf{t}_n\big) \cdot \max\left\{\mathbf{m} - \underline{\beta}, 0\right\}, \mathbf{t}_n\big)$
16: $\mathbf{q}_n^i \leftarrow \mathbf{P}^{c'}\big(\mathbf{t}_n + \mathbf{r}_{m,n}^i\big)$, $c' = Z - \left[\mathbf{H}_p\right]_{m,n}$
17: **end for**
18: **end for**
19: **end for**

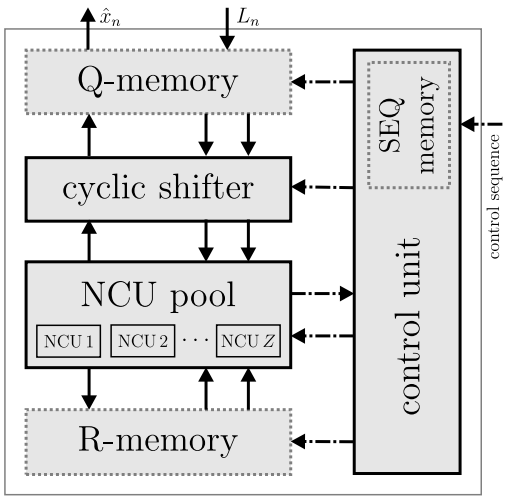

Fig. 1. Configurable QC-LDPC decoder architecture overview.

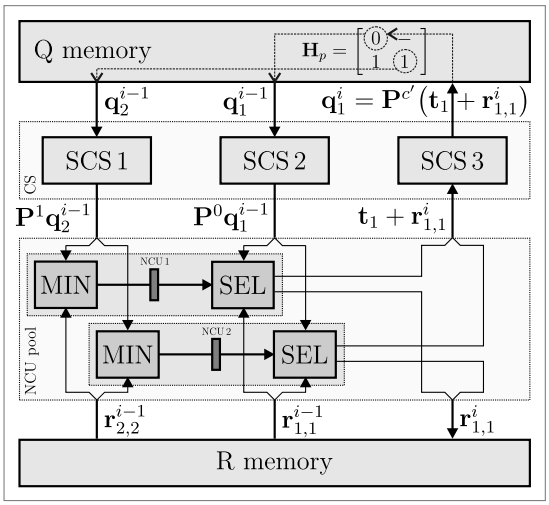

Fig. 2. Architectural details and decoding schedule of the QC-LDPC decoder for $Z = 2$ and $N_b = M_b = 2$.

In order to maximize the throughput, the decoder operates on Z elements in a parallel manner (using Z NCUs). The architecture computes one entry of $\mathbf{H}_p$ per clock cycle, so that at most N_b clock cycles are required per row of the LDPC matrix prototype. The throughput is further increased by parallel computation of the MIN and SEL phases (line 6-12 and 12-17, respectively) of Alg. 1. This parallel computation requires that the MIN and SEL phases do not access the same data at the same time. To avoid memory access conflicts, the MIN phase is performed on row $m+1$, whereas the SEL phase operates on row m. Furthermore, the SEL phase never accesses the nth column before it will be used in the MIN phase, to maintain convergence of layered LDPC decoding. These memory access constraints are enforced by the control unit (see Section III-D).

A. Node Computation Units

To enable parallel computation of the SEL and MIN phases, each of the Z NCUs is divided into a SEL unit and MIN unit by an intermediate pipeline stage as shown in Fig. 2.

MIN Unit: The kth MIN unit ($k = 1, 2, \ldots, Z$) iteratively computes $[\mathbf{m}_1]_k$, $[\mathbf{m}_2]_k$, $[\mathbf{v}]_k$, and $[\mathbf{s}]_k$ for all $n \in \mathcal{N}_{m+1}$ on row $m+1$ (cf. lines 5-11 of Alg. 1) by using the inputs $\left[\mathbf{r}_{m+1,n}^{i-1}\right]_k$ and $\left[\mathbf{P}^c\mathbf{q}_n^{i-1}\right]_k$. When the MIN phase of row $m+1$ has finished, both minima (i.e., $\mathbf{m}_1$ and $\mathbf{m}_2$), the index vector $\mathbf{v}$, and the sign-vector $\mathbf{s}$ are passed from the MIN to the SEL units.

SEL Unit: The SEL units iteratively compute the new clipped output messages $\mathbf{r}_{m,n}^i$ and $\mathbf{t} + \mathbf{r}_{m,n}^i$ as shown on lines 12-17 of Alg. 1 using $\mathbf{m}_1$, $\mathbf{m}_2$, $\mathbf{v}$, and $\mathbf{s}$ from the MIN unit and $\mathbf{r}_{m+1,n}^{i-1}$ as well as $\mathbf{P}^c\mathbf{q}_n^{i-1}$ from the R- and Q-memory, respectively.

Partial Parity Check: During computation of the new Q-values and R-messages, each SEL unit performs a partial parity check (PPC) of (1), i.e., the SEL units check whether $\mathbf{h}_{m'}^T\hat{\mathbf{x}} = 0$, where $\mathbf{h}_{m'}^T$ stands for the m'th row (for $m' = 1, 2, \ldots, M$) of $\mathbf{H}$ in (1) and $\hat{\mathbf{x}}$ corresponds to the binary-valued estimates obtained from the sign-bits of $\mathbf{q}_n^i$ (for $n \in \mathcal{N}_m$). After processing all m' rows of $\mathbf{H}$, the decoding procedure is terminated prematurely if all PPCs are satisfied. It is important to note that combining all PPCs does, in general, not correspond to the parity check in (1), since the evaluation of the PPCs are done sequentially and some estimates (since they result from the $\mathbf{q}_n^i$-values) might change during computation of all rows. We emphasize, however, that the proposed approach *never* terminates the decoder if (1) is *not* satisfied, but situations occur, where (1) is satisfied but the combination of all PPCs is not. Hence, the presented approach is slightly sub-optimal (in terms of number of decoding iterations) but the average number of iterations is still reduced significantly for medium to high SNRs, which leads to an improvement in terms of energy-efficiency in these SNR regimes (corresponding measurement results are provided in Section IV-B).

B. Cyclic Shifter

Cyclic shifts according to the entries of $\mathbf{H}_p$ are required when the MIN and SEL units read data from the Q-memory and when the SEL unit writes new data to the Q-memory (see Fig. 2). These cyclic shifts are performed in the CS unit, which consists of three subset cyclic shifters (SCSs). Since $Z \leq Z_{\max}$ (where $Z_{\max}$ denotes the maximum sub-block size supported by the architecture) can be different for each LDPC matrix prototype (i.e., depending on the code-block size) a flexible unit is required that is able to perform an arbitrary cyclic shift $0 \leq c < Z$. Several architectures that perform cyclic shifts of a subset of $Z_{\max}$ have been described in the literature. A corresponding survey can be found in [16]. However, most of the proposed architectures lead to either irregular structures, result in large circuit and interconnection area, or are only suited for a small number of different Z-values.

In order to attain reconfigurability of the QC-LDPC decoder, we designed a simple and efficient subset cyclic shifter (SCS) that is able to perform shifts for *any* subset of $Z_{\max}$ by an arbitrary shift amount $0 \leq c < Z$. The SCS architecture is depicted in Fig. 3. Each SCS consists of a shifter control

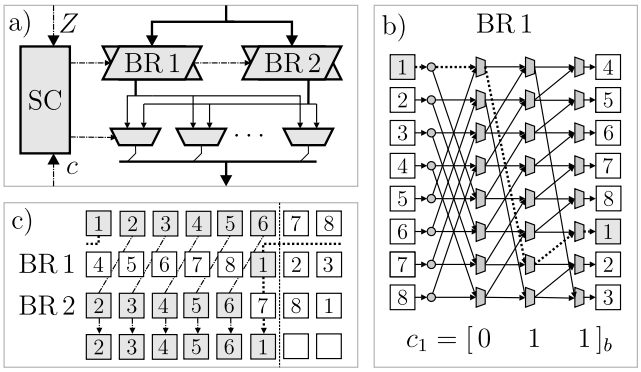

Fig. 3. Subset cyclic shifter architecture a) overview, b) barrel rotator (BR), and c) operation principle for $Z_{\max} = 8$, $Z = 6$, and $c = 1$.

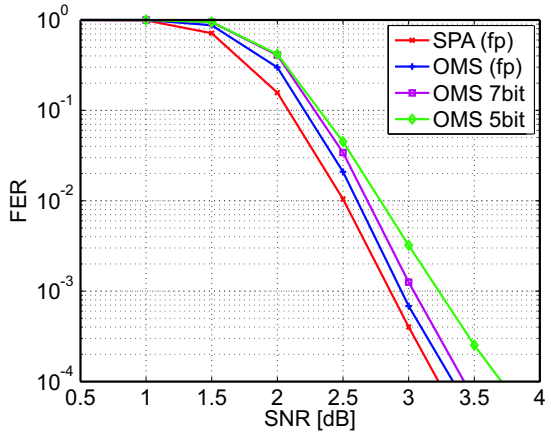

Fig. 4. Frame error rate (FER) comparison using the $R = 1/2$, $Z = 81$ code [9] in an AWGN channel.

(SC) unit, two barrel rotators (denoted by BR 1 and BR 2), and an output multiplexer stage. The first barrel rotator (BR 1) performs a cyclic left-shift by $c_1 = Z_{\max} - Z + c$, while BR 2 left-shifts the input by $c_2 = c$. The shift amounts c_1 and c_2 are computed in the SC unit. The output multiplexer stage selects the outputs $1, 2, \ldots, Z - c$ from BR 2 and the outputs $Z - c + 1, Z - c + 2, \ldots, Z$ from BR 1.

The key advantages of the proposed SCS are i) the low circuit area and ii) its fast operation. The total number of 2-to-1 multiplexers required in each SCS corresponds to

$$B_q\big(2Z_{\max}\lceil\log_2(Z_{\max})\rceil + Z_{\max}\big)$$

where B_q denotes the number of bits required for each entry of $\mathbf{q}$. Furthermore, each SCS only consists of $\lceil\log_2(Z_{\max})\rceil + 1$ multiplexer stages, which, thanks to its regular structure, enables pipelining.

C. Memories for Q and R

Since the Q-memory and the R-memory require two write operations and one read operation per clock cycle for all Z messages, the memories have been designed such that one (read or write) address corresponds to $Z_{\max}$ messages. To increase the memory bandwidth without (significantly) increasing the area, we used a double-clocking strategy, i.e., both memories operate at twice the clock frequency of the remaining logic, i.e., $f_{\text{mem}} = 2f_{\text{clk}}$. Since the timing characteristics of the memories available in our process technology are sufficiently fast, this double-clocking approach offers a two-fold increase in terms of memory bandwidth without leading to a significant area overhead.

D. Control Unit

The control unit consists of the SEQ memory and a simple controller, which generates all control signals for the decoder architecture. During the initialization phase of the LDPC decoder, the control unit can be configured with the OMS scaling parameter β, the sub-block size Z, and the maximum number of iterations I. Additionally, a control sequence needs to be loaded into the sequence (SEQ) memory prior to decoding. This control sequence contains all necessary information on $\mathbf{H}_p$ in combination with the decoding schedule (i.e., memory addresses, cyclic shifts etc.). During operation, the control words are being translated to corresponding signals and addresses. The key advantages of the proposed approach is that the decoder remains fully configurable, i.e., all possible QC-LDPC matrices that fit into the allocated memories can be processed, and the complexity of the control unit is significantly reduced.

The process of transforming $\mathbf{H}_p$ into a control sequence is done off-line. Each sequence consists of L control words using B_s bit. Note that only the information for one iteration is stored in the control sequence. The number of clock cycles required for one iteration corresponds to L, which determines the throughput of the LDPC decoder for a particular $\mathbf{H}_p$. Obtaining a short sequence is therefore a major goal of the sequence construction. The length of a sequence is determined by the number of entries in the LDPC matrix prototype $\mathbf{H}_p$ that are not equal to '-' and by the constraint that columns used by the SEL unit operating on row m and the MIN unit operating on row $m + 1$ must always be accessed first by the MIN unit (cf. Section III). The decoder is able to process the columns of $\mathbf{H}_p$ in an arbitrary ordering (as it is done in [6]), which can be exploited (during sequence construction) to reduce L.

E. Optimization for IEEE 802.11n

So far, the configurable QC-LDPC decoder architecture has been described independent of a particular standard. The steps described in the following have been performed to optimize the architecture for IEEE 802.11n [9], which is used for setting the design parameters for the ASIC implementation shown in Section IV.

Architectural Optimizations: The decoder has been designed to support at least all QC-LDPC matrix prototypes of IEEE 802.11n [9] while achieving a decoding throughput higher than 600 Mbit/s. To this end, we instantiated $Z_{\max} = 81$ NCUs, since the standard only requires $Z \in \{27, 54, 81\}$. To reduce the power consumption, we divided the NCU pool into three blocks consisting of 27 NCUs. Each of the three NCU blocks can be switched off using clock gating, which is used to reduce the power consumption for the modes $Z \in \{27, 54\}$. Corresponding power measurement results are shown in Section IV-B.

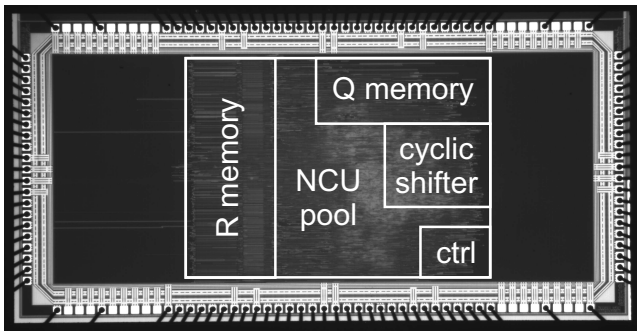

Fig. 5. Die photo of the QC-LDPC decoder in 0.18 µm CMOS technology. Note that there is unused circuit area on both sides of the ASIC.

Numeric Precision Optimization: Since IEEE 802.11n supports many different modulation and coding schemes, the numeric precision requirements have been evaluated using frame error rate (FER) simulations in an AWGN channel. Fig. 4 compares the FER of layered message passing using the floating-point (fp) sum-product algorithm (SPA) with layered OMS ($\beta = 0.15$). Note that OMS only loses approximately 0.2 dB to the SPA. The impact of MC is shown by using 7 bits and 5 bits. Note that both OMS simulations include early termination based on the PPCs described in Section III-A. MC to 5 bits leads to a 0.45 dB SNR loss compared to MC using 7 bits (at 10^{-3} FER) but reduces the amount of bits required in the Q-memory by 28%. Thus, we decided to limit the Q-values to $B_q = 5$ bit. The R-messages require $B_r = 5$ bit and the input LLRs have been quantized to 5 bit according to [15]. The curve associated with OMS 5 bit in Fig. 4 corresponds to the FER performance of the final implementation.

Q- and R-Memory Design: IEEE 802.11n requires at most $N_{b,\max} Z_{\max} = 24 \cdot 81$ Q-values and the R-memory requires at most $88 \cdot 81$ R-messages, where 88 corresponds to the maximum number of sub-blocks not equal to '-' in IEEE 802.11n [9]. Therefore, we instantiated eight 64×102 single-port SRAMs (total 52'224 bit) for the R-memory and three 32×135 two-port SRAMs (total 12'960 bit) for the Q-memory. The number of bits per sequence word is $B_s = 42$ bit and memory for a maximum of 128 sequence words has been allocated.

IV. Implementation Results

Fig. 5 shows a die photo of the fabricated QC-LDPC decoder ASIC in 0.18 µm (1P/6M) CMOS technology. The ASIC has been characterized using speed and power measurements. The key figures of the QC-LDPC decoder implementation are summarized in Tbl. IV. The implementation requires 3.39 mm^2 core area and achieves 208 MHz and 416 MHz for the slow logic clock and for the fast memory clock, respectively. Tbl. I shows a detailed area breakdown of the QC-LDPC decoder. It can clearly be observed that most of the circuit area is occupied by the Q- and R-memories. The NCU Pool and the cyclic shifter require approximately a third of the decoder's area and the circuit complexity of the configurable control unit is low.

TABLE I
Detailed area breakdown

Unit	mm^2	kGE[a]	%
Control unit	0.10	10.3	3.0
NCU pool	0.59	60.8	17.4
Q memory	0.83	85.6	24.5
R memory	1.16	119.6	34.2
Cyclic shifter	0.45	46.4	13.3
Miscellaneous[b]	0.26	26.8	7.6
Total	3.39	349.5	100

[a] One gate equivalent (GE) corresponds to the area of a two-input drive-one NAND gate of size 9.7 µm^2.
[b] Denotes remaining logic, i.e., pipeline registers, logic required for the input/output interface, etc.

TABLE II
Sequence length L and throughput Θ

Z	Rate	L	Θ [Mbit/s]
27	1/2	94	134
	2/3	88	190
	3/4	89	212
	5/6	95	222
54	1/2	88	286
	2/3	90	373
	3/4	90	419
	5/6	89	471
81	1/2	91	415
	2/3	89	565
	3/4	86	656
	5/6	80	780

A. Sequence Lengths and Throughput

Tbl. II shows the sequence lengths for all possible codes in IEEE 802.11n [9] as well as the resulting throughput for $f_{\text{clk}} = 208$ MHz using a maximum of $I = 5$ iterations. The throughput of the decoder can be computed as

$$\Theta = \frac{Z \cdot N_b \cdot R}{L \cdot I + 32} f_{\text{clk}}$$

where 32 additional clock cycles are required to load and flush the pipeline. The maximum achievable throughput for the $R = 5/6$, $Z = 81$ code corresponds to 780 Mbit/s and exceeds the throughput requirements of IEEE 802.11n.

B. Energy-Efficiency Measurements

Tbl. III shows energy-per-data-bit measurement results of the implemented ASIC. The impact of early-termination using PPCs, as well as the impact of clock gating is shown at different SNRs for different $R = 1/2$ codes [9]. For higher SNRs, the energy-efficiency improves significantly, since it becomes more likely that the PPCs are satisfied after a few iterations. For small block sizes (i.e., $Z = 27$), the power consumption can be dramatically reduced if clock gating is used, since part of the NCUs can be disabled during decoding of the smaller block-sizes (i.e., for $Z = \{27, 54\}$). Note that for $Z = 27$, clock-gating combined with the PPCs improves the energy-efficiency up to 63% at high SNRs, whereas in the low-SNR regime, still up to 48% improvement can be achieved.

TABLE III

ENERGY-EFFICIENCY MEASUREMENTS FOR RATE-$1/2$ CODES

Z	SNR	no power save		clk–gate		clk–gate & PPC	
		nJ/bit	%	nJ/bit	%	nJ/bit	%
27	1	11.5	100	6.0	52	6.0	52
	3	11.9	100	6.2	52	5.6	47
	7	11.6	100	6.1	53	3.1	27
54	1	5.6	100	4.4	79	4.4	79
	3	5.8	100	4.5	79	4.2	73
	7	5.6	100	4.4	79	2.2	40
81	1	3.8	100	3.8	100	3.8	100
	3	4.0	100	4.0	100	3.7	92
	7	3.9	100	3.9	100	1.9	49

TABLE IV

COMPARISON OF LDPC DECODERS FOR IEEE 802.11N

	This work	[6]	[4][a]	[17][b]
Technology [nm]	180	130	130	65
Core area [mm^2]	3.39	1.85	1.9	0.74
Memory area [mm^2]	1.99	1.04	–	0.26
Cell area [kGE]	144.3[c]	99.9	195	217
Max. clock freq. [MHz]	208	500	400	240
Max. throughput [Mbps]	780	1618	1000[d]	410

[a] Corresponding to the pipelined version [4].

[b] Corresponding to the $K = 4$, $N_{\mathrm{c2v}} = 5$, and $N_{\mathrm{SO}} = 8$ decoder [17].

[c] The memory area has been excluded (cf. Tbl. I).

[d] The maximum throughput is given for 2.2 dB SNR.

C. Comparison

In Tbl. IV,the presented implementation is compared with dedicated QC-LDPC decoders for IEEE 802.11n, i.e., the designs of Gunnam *et al.* [6], Sun *et al.* [4], and Rovini *et al.* [17]. Due to differences in process technologies, fair area, speed, and power comparisons are difficult to state. The circuit complexity and memory area of all four designs is comparable when considering technology scaling. Note that the implementation described in this paper achieves the lowest clock frequency, which is mainly due to the process technology and the detrimental input/output timing of the off-chip drivers. Nevertheless, our implementation achieves a throughput of up to 780 Mbps, which is sufficient to fulfill the requirements of the standard. Additionally, the achieved throughput is comparable to that of [4] and even higher than that of [17]. Therefore, we can conclude that configurability of our architecture does not come at a (significant) performance penalty.

V. CONCLUSION

We presented a fully reconfigurable LDPC decoder architecture for quasi-cyclic codes based on layered offset-min-sum message passing. Thanks to the subset cyclic shifter and the reconfigurable control unit, the proposed architecture is able to decode virtually any QC-LDPC code that fits into the allocated memories. The memory requirements have been significantly reduced by clipping of the decoder-internal messages, which has shown to enable a trade-off between error rate performance and circuit area at design-time. The QC-LDPC decoder has been optimized for IEEE 802.11n and fabricated in 0.18 µm CMOS technology. Measurement results and comparison with dedicated LDPC decoders for IEEE 802.11n demonstrate that the flexibility of our architecture not necessarily leads to a significant performance penalty in terms of area, throughput, and energy-efficiency.

ACKNOWLEDGMENT

The authors want to thank M. Braendli for his support during the back-end design. Furthermore, we would like to thank Prof. W. Fichtner, Dr. N. Felber, and Dr. H. Kaeslin for their support during design and testing of the QC-LDPC decoder ASIC.

REFERENCES

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

[2] M. P. C. Fossorier, "Quasi-cyclic low-density parity-check codes from circulant permutation matrices," *IEEE Trans. on Inf. Theory*, vol. 50, no. 8, pp. 1788–1793, Aug. 2004.

[3] H. Zhong and T. Zhang, "Block-LDPC: a practical LDPC coding system design approach," *IEEE Trans. on Circuits and Systems I*, vol. 52, no. 4, pp. 766–775, Apr. 2005.

[4] Y. Sun, M. Karkooti, and J. R. Cavallaro, "High throughput, parallel, scalable LDPC encoder/decoder architecture for OFDM systems," in *Proc. IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, Oct. 2006, pp. 39–42.

[5] K. K. Gunnam, G. S. Choi, W. Wang, E. Kim, and M. B. Yeary, "Decoding of quasi-cyclic LDPC codes using an on-the-fly computation," in *Proc. Fortieth Asilomar Conference on Signals, Systems and Computers*, Oct. 2006, pp. 1192–1199.

[6] K. K. Gunnam, G. S. Choi, W. Wang, and M. B. Yeary, "Multi-rate layered decoder architecture for block ldpc codes of the ieee 802.11n wireless standard," in *Proc. IEEE International Symposium on Circuits and Systems*, May 2007, pp. 1645–1648.

[7] *Digital Video Broadcasting (DVB) User guidelines for the second generation system for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2), ETSI TR 102 376*, Feb. 2005.

[8] *IEEE P802.16e, Part 16, "Air Interface for Fixed and Mobile Broadband Wireless Access Systems," Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands, and Corrigendum 1*, Feb. 2006.

[9] *IEEE P802.11n/D5.02, Part 11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Enhancements for Higher Throughput"*, July 2008.

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

[11] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. on Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[12] A. J. Blanksby and C. J. Howland, "A 690-mW 1-Gb/s 1024-b, rate-1/2 low-density parity-check code decoder," vol. 37, no. 3, pp. 404–412, Mar. 2002.

[13] X.-Y. Shih, C.-Z. Zhan, C.-H. Lin, and A.-Y. Wu, "An 8.29mm2 52mW multi-mode LDPC decoder design for mobile WiMAX system in 0.13um CMOS process," vol. 43, no. 3, pp. 672–683, Mar. 2008.

[14] S. Sharon, J. Litsyn, and J. Goldberger, "Efficient serial message-passing schedulers for LDPC decoding," *IEEE Trans. on Inf. Theory*, vol. 53, no. 11, pp. 4076–4091, Nov. 2007.

[15] 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.

[16] C.-H. Liu, C.-C. Lin, H.-C. Chang, C.-Y. Lee, and Y. Hsua, "Multi-mode message passing switch networks applied for QC-LDPC decoder," in *IEEE International Symposium on Circuits and Systems*, May 2008, pp. 752–755.

[17] M. Rovini, G. Gentile, F. Rossi, and L. Fanucci, "A scalable decoder architecture for IEEE 802.11n LDPC codes," in *Proc. IEEE GLOBECOM*, Nov. 2007, pp. 3270–3274.