Received December 17, 2021, accepted January 4, 2022, date of publication January 7, 2022, date of current version January 14, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3141493

High-Throughput Multi-Frame Decoding of QC-LDPC Codes With Modified Rejection-Based Minimum Finding

ALIREZA HASANI[1,2], **LUKASZ LOPACINSKI**[1], **GORAN PANIC**[1], **AND ROLF KRAEMER**[1,2], **(Life Member, IEEE)**
[1]IHP—Leibniz-Institut für Innovative Mikroelektronik, 15236 Frankfurt (Oder), Germany
[2]Department of Electrical and Computer Engineering, Brandenburg University of Technology, Cottbus–Senftenberg, 03046 Cottbus, Germany

Corresponding author: Alireza Hasani (hasani@ihp-microelectrinics.com)

This work was supported by the German Research Foundation (DFG), project no. 442607813 (An ultra-low-power, modular PSSS-FEC-ARQ combined processor for 100 Gbps wireless communication in the THz-band).

ABSTRACT The key computation in the min-sum decoding algorithm of a Low-Density Parity-Check (LDPC) code is finding the first two minima and also the location of the first minimum among a set of messages passed from Variable Nodes (VNs) to Check Nodes (CNs) in a Tanner graph. In this paper, we propose a modified rejection-based scheme for this task which is able to find the one-hot sequence of the minimum location instead of its index. We show that this modification effectively reduces the complexity of min-sum decoding algorithm. Additionally, we reveal a pipelining potential in such a rejection-based architecture which facilitates the multi-frame decoding of LDPC codes and therefore results in an improvement in decoding throughput with bearable hardware overhead. Synthesis and floorplanning in an industrial 28 nm CMOS technology show improved results in terms of throughput, power, and chip area.

INDEX TERMS Decoding complexity, decoding throughput, LDPC code, min-sum decoding, one-hot sequence.

I. INTRODUCTION

Low-Density Parity-Check (LDPC) codes are one of the selected forward-error correction candidates for next-generation wireless communication systems like 5G and IEEE 802.11ax. However, despite the promising performance, the decoding complexity of LDPC codes is still a barrier toward high-throughput applications of these codes. In order to address this downside of LDPC codes a great deal of effort has been expended on different aspects of them like construction, encoding, and decoding.

The major decoding method for LDPC codes is the iterative Belief Propagation (BP) algorithm run over the Tanner graph representation of these codes. BP algorithm in its primary form is conducted as a sum-product algorithm [1] in which a series of messages are successively exchanged between the nodes of the Tanner graph. Min-sum decoding algorithm [2], [3] is a simplified method compared with the sum-product scheme for BP decoding of LDPC codes replacing complex hyperbolic tangent functions with simpler minimum finding computations. More specifically, in min-sum decoding algorithm the computation of messages passed from CNs to VNs in a Tanner graph during decoding is targeted for simplification, and a CN finds the minimum value among the messages receiving from its neighbor VNs instead of computing complicated hyperbolic tangent functions.

After a more detailed introduction of min-sum decoding in section II it becomes evident that the essential computation of this decoding scheme is finding the first and second minima together with the index (or location) of the first minimum among some binary values. This computation, although simpler than the original computations in sum-product algorithm, should still be performed with the most simplest approach in order to avoid overall decoding complexity. To this goal, different approaches have been proposed specifically for application to min-sum decoding which [4] gives an overview of them.

In this regard, authors in [5] proposed a hardware-friendly rejection-based technique which is able to accomplish this task faster than the former techniques, only with insubstantial

The associate editor coordinating the review of this manuscript and approving it for publication was Faissal El Bouanani.

increase in circuit area. Additionally, we proposed in [4] a modification to this method which suggests to use one-hot sequence representation for the location of minimum instead of its binary index, claiming that this modification has implementation benefits. More specifically, the rejection-based scheme of [5] finds the index of the first minimum, while in our modified approach the one-hot sequence of the first minimum is found. In this paper, we first re-investigate this modification and show how the rejection-based technique can be modified to output one-hot sequence of the minimum. Besides, an important pipelining capacity of this method is revealed which is able to boost decoding throughput while maintaining the overall hardware overhead acceptable and not increasing the consumed energy per bit. The proposed idea of multi-frame decoding with the modified rejection-based scheme is synthesized and also floorplanned in an industrial 28 nm CMOS technology. Results from this synthesis reveal a considerable improvement in throughput, power consumption, and also chip area of the min-sum decoder.

The organization of the paper is as follows. Section II provides the necessary preliminaries, specifically about min-sum decoding algorithm and also rejection-based method for finding the first two minima. In section III, the modified rejection-based scheme is introduced and its pipelining capacity for boosting decoding throughput is revealed in section IV. Section V is devoted to simulation, synthesis, and floorplan results, and final conclusions are made in section VI.

II. PRELIMINARIES

A. QC-LDPC CODES AND LD SCHEDULE

In a nutshell, the codewords of a Quasi-Cyclic LDPC (QC-LDPC) code are spanned by a sparse Row-Column (RC)-constrained [6] matrix composed of only Circulant-Permutation Matrices (CPMs) and zero matrices. A CPM can be assumed as an identity matrix in that all the rows have been shifted cyclically. Fig. 1-(a) shows an example Parity-Check Matrix (PCM) of a QC-LDPC code.

BP is the major decoding algorithm of LDPC codes, performed based on the Tanner graph representation of them. The Tanner graph of an example LDPC code has been shown in Fig. 2 which is composed of two sets of nodes and a number of connections between them. The gray circles, called VNs, represent the code bits or columns of the PCM, while the white circles, called CNs, denote the check sums or rows of the PCM. During the BP algorithm, reliability messages are successively passed between VNs and CNs in an attempt to obtain probabilities expressing whether a given symbol in a received codeword is ‘1’ or ‘0’. The resulting sequence will then have the maximum probability according to the received soft-decision sequence. We assume $Z_{j,l}$ as a Variable-to-Check (VTC) message passed from l^{th} VN to j^{th} CN and $L_{j,l}$ as a Check-to-Variable (CTV) message passed from j^{th} CN to l^{th} VN.

The rules governing the order of the exchange of messages between the nodes of a Tanner graph is referred to as the schedule of the BP algorithm. One basic possibility is the flood-like schedule, in which, in each iteration, firstly all the VNs update their messages and send them to CNs and then CNs update their own messages and send them back to VNs. Flood schedule facilitates a fully parallel decoding architecture, yet at the cost of high interconnect complexity. Layered Decoding (LD) is rather a partially parallel schedule which can propose faster convergence rate, improved coding gain, and reduced hardware overhead over flood schedule with lower decoding complexity [7]. In LD schedule, the rows in the PCM are split in layers, and then the BP algorithm runs over layers in successive order. In other words, each iteration of the algorithm is split into several sub-iterations, during each one reliability messages are exchanged between CNs of that layer and their neighbor VNs. At the end of each sub-iteration the updated reliability messages are handed down to the next layer. Accordingly, only a subset of CNs and VNs participate in each sub-iteration, and layers are processed successively from top to down the PCM.

The complexity of LD schedule is lowered if the layers in the PCM have only single- or zero-weight columns. QC-LDPC codes have inherently such a property, if each row of CPMs is considered as a layer. Moreover, by the shuffling idea proposed in [8], [9] the implementation complexity of LD is further simplified, since it suffices to define only the connections of the first layer of the shuffled PCM for decoding. The shuffling only interleaves the row orders and thus does not degrade the Bit-Error Rate (BER) performance. More details on the principles of shuffling, how it is performed and how it is beneficial are found in [9]. Fig. 1-(b) shows the shuffled version of the example PCM.

B. MIN-SUM DECODING

Min-sum decoding is a simplified variant of BP algorithm, in which the magnitude of a CTV message $|L_{j,l}|$ is approximated by the minimum of the magnitude of all the VTC messages $|Z_{j,l'}|$ arriving in s_j from all its neighbors except v_l. For example, in Fig. 2, CTV messages sent from s_1 to its neighbor VNs v_0, v_1 and v_3 have the magnitude of $|L_{1,0}| = min(|Z_{1,1}|, |Z_{1,3}|)$, $|L_{1,1}| = min(|Z_{1,0}|, |Z_{1,3}|)$ and $|L_{1,3}| = min(|Z_{1,0}|, |Z_{1,1}|)$. This implies that the CTV messages sent from a CN to its neighbor VNs are either the first or second minimum among the VTC messages that CN has just received. For instance, if one assumes that the relation $|Z_{1,0}| < |Z_{1,1}| < |Z_{1,3}|$ holds for the previous example, then it results in $|L_{1,0}| = |Z_{1,1}|$, $|L_{1,1}| = |Z_{1,0}|$ and $|L_{1,3}| = |Z_{1,0}|$. Accordingly, the task of a CN processing unit in a min-sum decoding algorithm reduces to finding the first and second minimum among the incoming VTC messages together with the location of the first minimum, with the latter needed, because the CTV message at that location is simply valuated as the second minimum and all the other CTV messages as the first minimum.

C. REJECTION-BASED FIRST TWO MINIMA FINDING

Several works have been thus far devoted to the efficient methods for finding the first two minima and the index of

$$\boldsymbol{H}_{qc}=\left[\begin{array}{ccc|ccc|ccc|ccc}
0&0&0&0&0&1&0&1&0&1&0&0\\
0&0&0&1&0&0&0&0&1&0&1&0\\
0&0&0&0&1&0&1&0&0&0&0&1\\
\hline
0&0&1&0&0&0&1&0&0&0&1&0\\
1&0&0&0&0&0&0&1&0&0&0&1\\
0&1&0&0&0&0&0&0&1&1&0&0\\
\hline
0&1&0&1&0&0&0&0&0&0&0&1\\
0&0&1&0&1&0&0&0&0&1&0&0\\
1&0&0&0&0&1&0&0&0&0&1&0\\
\hline
1&0&0&0&1&0&0&0&1&0&0&0\\
0&1&0&0&0&1&1&0&0&0&0&0\\
0&0&1&1&0&0&0&1&0&0&0&0
\end{array}\right]$$

(a)

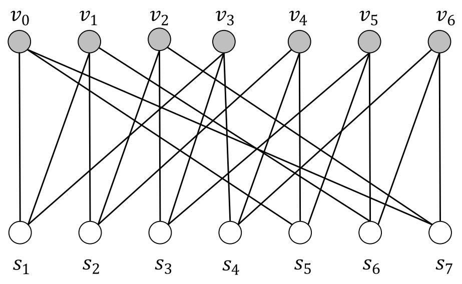
$$\rightarrow \boldsymbol{H}_{qc}^{sh}=\begin{bmatrix}\boldsymbol{H}_{qc}^{sh,(1)}\\ \\ \boldsymbol{H}_{qc}^{sh,(2)}\\ \\ \boldsymbol{H}_{qc}^{sh,(3)}\end{bmatrix}=\left[\begin{array}{ccc|ccc|ccc|ccc}
0&0&0&0&0&1&0&1&0&1&0&0\\
0&0&1&0&0&0&1&0&0&0&1&0\\
0&1&0&1&0&0&0&0&0&0&0&1\\
1&0&0&0&1&0&0&0&1&0&0&0\\
\hline
0&0&0&1&0&0&0&0&1&0&1&0\\
1&0&0&0&0&0&0&1&0&0&0&1\\
0&0&1&0&1&0&0&0&0&1&0&0\\
0&1&0&0&0&1&1&0&0&0&0&0\\
\hline
0&0&0&0&1&0&1&0&0&0&0&1\\
0&1&0&0&0&0&0&0&1&1&0&0\\
1&0&0&0&0&1&0&0&0&0&1&0\\
0&0&1&1&0&0&0&1&0&0&0&0
\end{array}\right]$$

(b)

FIGURE 1. **(a) PCM of a (12,4)-QC-LDPC code; (b) its shuffled version.**

$$H=\begin{bmatrix}
1&1&0&1&0&0&0\\
0&1&1&0&1&0&0\\
0&0&1&1&0&1&0\\
0&0&0&1&1&0&1\\
1&0&0&0&1&1&0\\
0&1&0&0&0&1&1\\
1&0&1&0&0&0&1
\end{bmatrix}$$

(a)



(b)

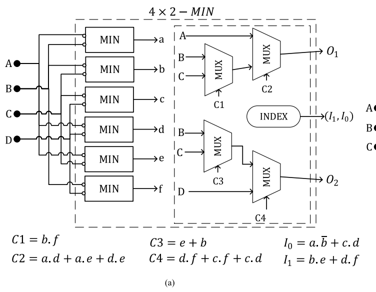
FIGURE 2. **(a) The PCM of a (7,3)-LDPC code; (b) Its Tanner graph.**

the minimum exclusively for min-sum LDPC decoders [5], [10]–[18]. The sorting-based approach in [10] is a radix-2 tree structure which makes $\rho - 2 + \lceil \log \rho \rceil$ comparisons between the ρ binary numbers in a tree-like structure in order to find the first two minima min_1 and min_2. The other architecture proposed in [10] constructs a 2^k-input minimum-value generator using two 2^{k-1}-input minimum-value generators in a tree form. In [11], a two-input module is implemented with one adder and one multiplexer and a three-input module with three adders, five multiplexers and several simple logical gates (refer to Figures 13-a and 13-b of [11]). Based on these two modules bigger multi-input designs can be realized in hierarchical form. Authors in [12] modify the sorting-based approach of [10] to perform better in area and speed. However, their modification only touches how min_2 is found and the remaining procedures i.e. determining min_1 and its index remains intact. [13] further generalizes the approach of [10] aiming at using mixed radix architectures that improve the architecture latency. The modified tree structure of [14] requires less number of comparisons to find min_1 and min_2 with respect to [10] and [16] achieves the same goal by reusing intermediate comparison results calculated for min_1 for collecting the candidates of min_2. [17] presents an algorithm called exMin that reduces the required hardware by using an estimation of the second minimum as min_2.

In [15] and [18] a different strategy is adopted wherein the finding of the first two minima is based on scanning of each input data from the Most Significant Bit (MSB) to the Least Significant Bit (LSB). This strategy is referred to as bit-serial architecture and assumed as an alternative to the tree structure.

Among them, tree-based architectures have been of greater importance, whose latest variant is the rejection-based method proposed by the authors in [5]. Let $x_0, \ldots, x_{\rho-1}$ be the ρ binary fixed-point numbers, each K bits long, whose first two minima min_1 and min_2 are to be found. In this method, two 3×2-MIN and 4×2-MIN modules, shown in Fig. 3 serve as the fundamental building blocks for constructing larger modules with arbitrary number of inputs. In these modules in the first step all the two combinations of the $\rho = 3$ or 4 numbers are compared in pair by the MIN units, thus yielding the flags a, b, c in 3×2-MIN or a, b, c, d, e, f in 4×2-MIN modules. These flags are then used to form the select bits of the multiplexers, according to the relations stated at the bottom of the two modules. In these relations, the signs ".", "+" and "bar" represent logical operations of AND, OR and complement respectively. Note further that each MIN unit in these modules outputs "1" if its upper input is bigger than its lower input and "0" otherwise.

Building larger modules is viable by proper configuration of the two fundamental 3×2-MIN and 4×2-MIN modules. Fig. 4 shows such configurations for the five-, six-, seven- and eight-input modules. It should be noted that the outputs O_1 and O_2 in figures 3 and 4 are not sorted, meaning that it may be $min_1 = O_1$ and $min_2 = O_2$ or vice versa. However, the index output specifies the location of min_1. The index output in this method is in the form of binary representation of the location of min_1. For example, if $\rho = 4$, the index output will



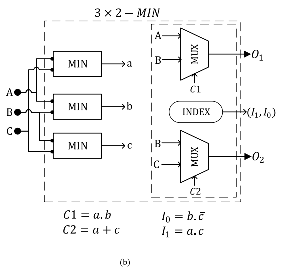


FIGURE 3. Fundamental sorting modules in rejection-based technique (a) Four-input two-output module; (b) Three-input two-output module [5].

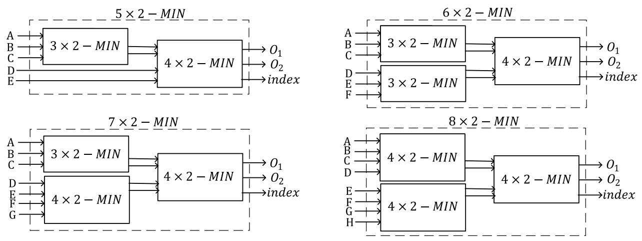


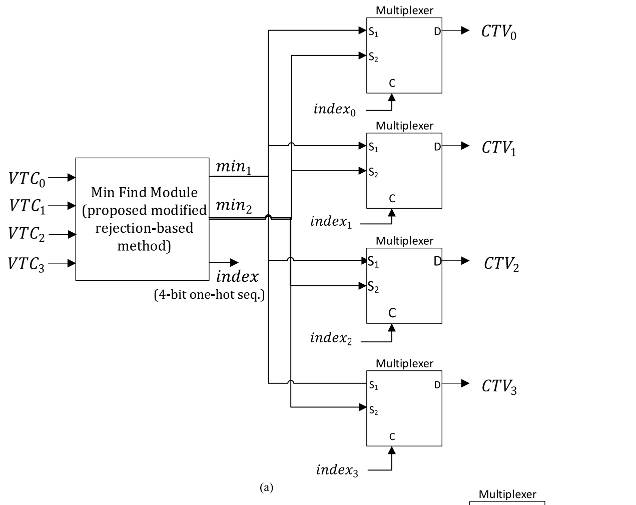
FIGURE 4. Configuration of fundamental modules to build larger comparator modules based on rejection scheme for $\rho = 5, 6, 7$ and 8.

be either "00", "01", "10" or "11" in order to specify the index of the minimum.

III. MODIFIED REJECTION-BASED SCHEME

Besides the binary representation, there is an alternative way of specifying the location of min_1, referred to as the one-hot sequence. A one-hot sequence is a ρ-bit binary sequence, in that all the bits except one are '0'. The only '1' in the sequence resides in the same location as the index of the min_1. For example, for $\rho = 4$, the one-hot sequence may be "1000", "0100", "0010" or "0001", in each case the location of '1' specifies the index of the min_1.

We claim that the one-hot sequence is the superior way of locating min_1 instead of its binary representation, since it simplifies the overall min-sum decoding algorithm. To argue that, the schematic diagram of the relevant part of the min-sum decoding scheme is sketched in Fig. 5. This diagram illustrates how CTV messages are computed from the received VTC messages at a CN with degree 4. Among the four received VTC messages, min_1, min_2 and location of min_1 must be determined. Then, the value of the CTV message at the location of first minimum takes the value of min_2 and all the other CTV messages take the value of min_1. In Fig. 5-a the *index* output is the location of min_1 specified as the one-hot sequence based on the modified rejection-based technique described shortly after. As shown, as many multiplexers as the weight of the processing CN, in this example 4, are needed, and the bits of *index* output, namely $index_i$ are used directly as the select bits of the multiplexers. If a select bit is '0', S_1 which is min_1 is selected, and otherwise S_2 which is min_2. Hence, only one of the multiplexers outputs min_2 and all the others output min_1. In contrast, in Fig. 5-b *index* is



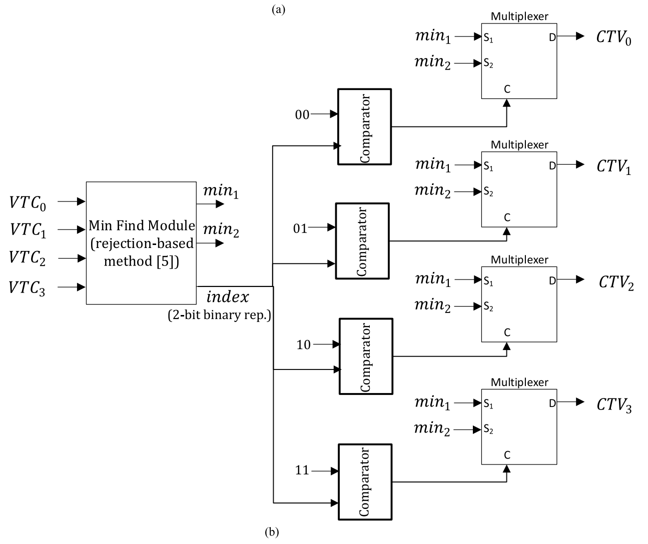


FIGURE 5. Computation of CTV messages in a min-sum decoding algorithm, when the index of min_1 is specified by (a) one-hot sequence according to the proposed modified rejection-based technique, and (b) binary representation according to rejection-based technique of [5].

the binary representation of min_1 location provided by the rejection-based technique of [5]. As a result, a set of additional comparators which have been boldfaced in the figure are required to produce the select bits of the multiplexers. A comparator outputs '1' only if its two inputs equal. The first input of the comparators are respectively the different possibilities of the *index*, in this example "00", "01", "10", and "11". Accordingly, only one of the comparators will have a '1' in output, leading to min_2 to appear at the output of the multiplexer connected to that comparator. The other comparators will have '0' in output, leading to min_1 to appear at the output of their multiplexers. Based on this argument, one can claim that one-hot sequence is the preferred way for representation of the location of min_1 in min-sum decoding method, since the need for extra comparators is eliminated.

The two 3×2-MIN and 4×2-MIN circuits in Fig. 3 can be modified to output the one-hot sequence of the location of min_1 instead of its binary index. Fig. 6 illustrates the modified circuits in them the outputs $I0$, $I1$, $I2$, and $I3$ are the bits of

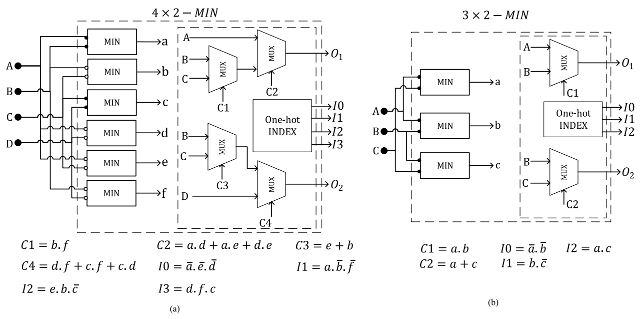


FIGURE 6. Fundamental sorting units in modified rejection-based technique (a) Four-input two-output module; (b) Three-input two-output module.

TABLE 1. Truth table for the 3 × 2-MIN module of the modified rejection-based method.

Status	a	b	c	$C1$	$C2$	$I0$	$I1$	$I2$
$A<B<C$	0	0	0	0	0	1	0	0
$A<C<B$	0	0	1	0	1	1	0	0
$B<A<C$	0	1	0	0	0	0	1	0
$B<C<A$	1	1	0	1	1	0	1	0
$C<A<B$	1	0	1	0	1	0	0	1
$C<B<A$	1	1	1	1	1	0	0	1

the one-hot sequence. These bits are computed based on the flag bits output by MIN units. The corresponding relations for deriving these bits are stated at the bottom of this figure, and they are developed based on the truth tables 1 and 2, respectively for 3 × 2-MIN and 4 × 2-MIN modified circuits. These relations have been expressed in an optimized form which is straightforward to do so with using basic methods for optimized implementation of logic functions like Karnaugh map [19] or one of other advanced methods.

In our modified rejection-based scheme only the fundamental 3 × 2-MIN and 4 × 2-MIN circuits are modified to output the location of min_1 as a one-hot sequence. However, building larger modules with arbitrary number of inputs follows the same procedure as exemplified for the original scheme in Fig. 4.

IV. MULTI-FRAME DECODER ARCHITECTURE

In this section, we reveal a potential for improving decoding throughput of LDPC codes. The flowchart of the decoding procedure is shown in Fig. 7. Decoding begins with fetching a new sequence whose corresponding codeword is sought. This sequence is initially checked if it satisfies the parity-check equations. If yes, it means that it is already an error-free codeword and a new sequence can be taken in for decoding. Otherwise, it enters the decoding loop until it is decoded successfully or the preset number of iterations represented by J_{max} is reached and decoding fails. Each round of the decoding loop is in fact one sub-iteration performed on a layer of a PCM.

TABLE 2. Truth table for the 4 × 2-MIN module of the modified rejection-based method.

Status	a	b	c	d	e	f	$C1$	$C2$	$C3$	$C4$	$I0$	$I1$	$I2$	$I3$
$A<B<C<D$	0	0	0	0	0	0	0	0	0	0	1	0	0	0
$A<B<D<C$	0	0	1	0	0	0	0	0	0	0	1	0	0	0
$A<C<B<D$	0	1	0	0	0	0	0	0	1	0	1	0	0	0
$A<C<D<B$	0	1	0	0	0	1	1	0	1	0	1	0	0	0
$A<D<B<C$	0	0	1	0	0	1	0	0	0	1	1	0	0	0
$A<D<C<B$	0	1	1	0	0	1	1	0	1	1	1	0	0	0
$B<A<C<D$	1	0	0	0	0	0	0	0	0	0	0	1	0	0
$B<A<D<C$	1	0	1	0	0	0	0	0	0	0	0	1	0	0
$B<C<A<D$	1	0	0	0	1	0	0	1	1	0	0	1	0	0
$B<C<D<A$	1	0	0	1	1	0	0	1	1	0	0	1	0	0
$B<D<A<C$	1	0	1	1	0	0	0	1	0	1	0	1	0	0
$B<D<C<A$	1	0	1	1	1	0	0	1	1	1	0	1	0	0
$C<A<B<D$	0	1	0	0	1	0	0	0	1	0	0	0	1	0
$C<A<D<B$	0	1	0	0	1	1	1	0	1	0	0	0	1	0
$C<B<A<D$	1	1	0	0	1	0	0	1	1	0	0	0	1	0
$C<B<D<A$	1	1	0	1	1	0	0	1	1	0	0	0	1	0
$C<D<A<B$	0	1	0	1	1	1	1	1	1	1	0	0	1	0
$C<D<B<A$	1	1	0	1	1	1	1	1	1	1	0	0	1	0
$D<A<B<C$	0	0	1	1	0	1	0	0	0	1	0	0	0	1
$D<A<C<B$	0	1	1	1	0	1	1	0	1	1	0	0	0	1
$D<B<A<C$	1	0	1	1	0	1	0	1	0	1	0	0	0	1
$D<B<C<A$	1	0	1	1	1	1	0	1	1	1	0	0	0	1
$D<C<A<B$	0	1	1	1	1	1	1	1	1	1	0	0	0	1
$D<C<B<A$	1	1	1	1	1	1	1	1	1	1	0	0	0	1

Processing steps in the decoding loop are as follows:

1) Initialization/Update VTC messages: For a new sequence for decoding at its first iteration and first sub-iteration A Posteriori Probability (APP) values $Y_l,\ 0 \leq l < n$ are initialized with y_l, the soft-decision sequence at the output of the channel, and CTV messages are initialized with 0, i.e., $L_{j,l} = 0,\ \ 1 \leq j \leq E, 0 \leq l < n$. Here, n is the code length and E is the number of rows in each layer of the PCM. VTC messages are initialized at each iteration with APP values, i.e., $Z_{j,l} = Y_l,\ 1 \leq j \leq E, 0 \leq l < n$.
2) Update CTV messages: CTV messages are updated as

$$L_{j,l} = \underbrace{\prod_{l' \in \mathcal{B}(\boldsymbol{h}_j)\backslash l} sgn(Z_{j,l'})}_{I} \times \underbrace{\min_{l' \in \mathcal{B}(\boldsymbol{h}_j)\backslash l} |Z_{j,l'}|}_{II}, \tag{1}$$

where

$$\mathcal{B}(\boldsymbol{h}_j) = \{l : h_{j,l} = 1, 0 \leq l < n\}$$

denotes the set of VNs neighbor to the CN $\boldsymbol{h}_j$ in layer 1.
3) Update APP values: APP values are updated according to:

$$Y_l = y_l + L_{j,l},\ j \in \mathcal{A}_l,\ \ 0 \leq l < n. \tag{2}$$

with

$$\mathcal{A}_l = \{j : h_{j,l} = 1, 1 \leq j \leq E\}$$

representing the set of CNs in layer 1 connected to v_l.

Fig. 8 shows the implemented decoding architecture for the example (12,4)-QC-LDPC code whose PCM was shown in

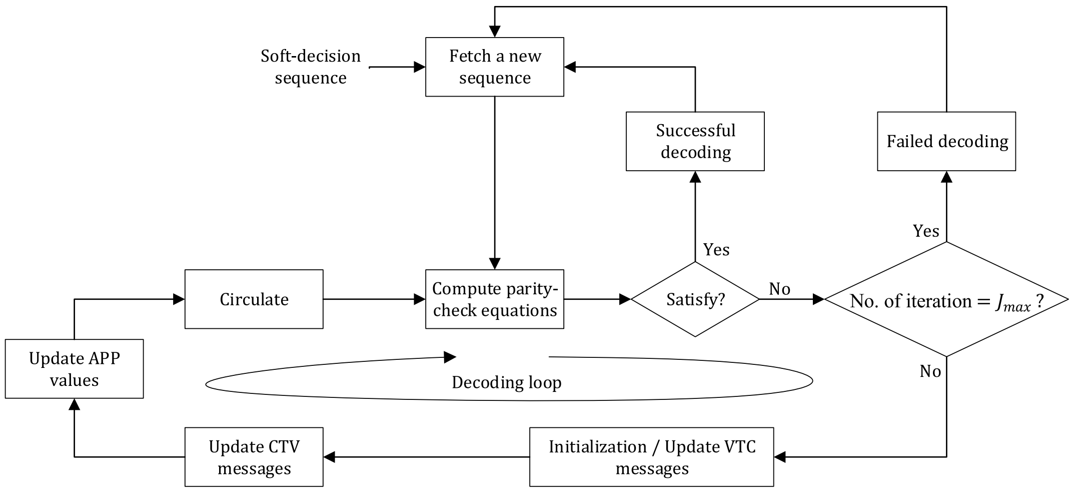


FIGURE 7. **Flowchart of the LD algorithm [20].**

Fig. 1. Besides, a block of "update CTV messages" in this figure is shown in details in Fig. 9. Such a block is comprised of two main parts each specified by a dashed contour. The upper one is exactly a circuit similar to Fig. 5-(a) responsible for calculating the magnitude of the CTV messages according to part II of (1). VTC messages input to this circuit are first converted from 2's complement format to sign-magnitude format. The lower circuit in Fig. 9 is responsible for calculating the sign of CTV messages according to part I of (1). The inputs of this circuit are the sign, i.e., the MSB of the VTC messages. At the last stage, the sign of a CTV message is combined with its magnitude, yielding the corresponding CTV message in 2's complement format.

Each processing step in the decoding loop can be implemented in one clock cycle, except for the step of updating CTV messages, which is equivalent to finding the first two minima and also the location of the first minimum among a multiple of VTC messages. The rejection-based approach outlined in section II-C or its modified version proposed in section III do not carry out their task in one step. Therefore, they need more than one clock cycle to perform their task if reserving one clock cycle for each step.

By examining the structure of 3 × 2-MIN and 4 × 2-MIN modules in Fig. 3 or 6 it is deduced that they need two clock cycles for their operation. In the first clock cycle the pairwise comparisons of the inputs A, B, C and D are carried out and the flags a, b, c, d, e and f are formed. Then, in the second clock cycle the outputs of the module are generated. When building modular circuits with 5-8 inputs, as exemplified by Fig. 4, two levels of fundamental units are needed. This results in the latency of 2 × 2 clock cycles. Likewise, when building larger modular circuits with 9-16 inputs, the number of levels of fundamental modules will be 3 and hence they need 3 × 2 clock cycles. By deduction, we can state that the number of levels of fundamental modules needed for building a configuration with ρ inputs is $\lceil \log_2 \rho \rceil - 1$. This yields the latency of $2(\lceil \log_2 \rho \rceil - 1) + 1$. The one extra clock cycle is required to perform the final sorting between the two outputs of the module in order to determine which one is min_1 and which one is min_2.

The multi-level structure of the (modified) rejection-based scheme can be used smartly to increase decoding throughput without considerable hardware overhead. In this multi-level structure when one layer is given input, its previous layers are already done and they are in idle mode. Consider e.g. the configuration in Fig. 10, i.e., a 16 × 2-MIN module requiring 7 clock cycles to finish its work. During the first two clock cycles the four 4 × 2-MIN modules at level-1 are active. During the clock cycles 3 and 4 these two modules stand by and the two 4 × 2-MIN modules at level-2 start operating. The last three clock cycles belong then to the single 4 × 2-MIN module at level-3. Apparently, for this architecture, the flow of inputs could become non-stop and the module can take new inputs every clock cycles.

The functional (also known as Register-Transfer Level (RTL)) simulation of the 16-input modified rejection-based module of Fig. 10 conducted with Modelsim^{TM} is illustrated in Fig. 11. The figure shows that the module needs initially 7 clock cycles to output the result of the first inputs. But afterward, it is given input every clock cycles, and it gives output every clock cycles as well. In this figure, "clk" is the clock signal and the 16 inputs are denoted by "d1" to "d16". The signals "O1" and "O2" are min_1 and min_2 respectively and "min_index" is the location of min_1 as a one-hot sequence.

Based on this fact, the decoder architecture in Fig. 8 can accept several sequences for decoding consecutively. The downside of this multi-frame processing idea is the additional overhead. When several sequences are taken in for decoding, it is not known which one will be converged to a valid codeword and leave the decoder sooner than the others. Therefore, the decoded sequences will be most likely out of the order with which they have entered the decoder and a

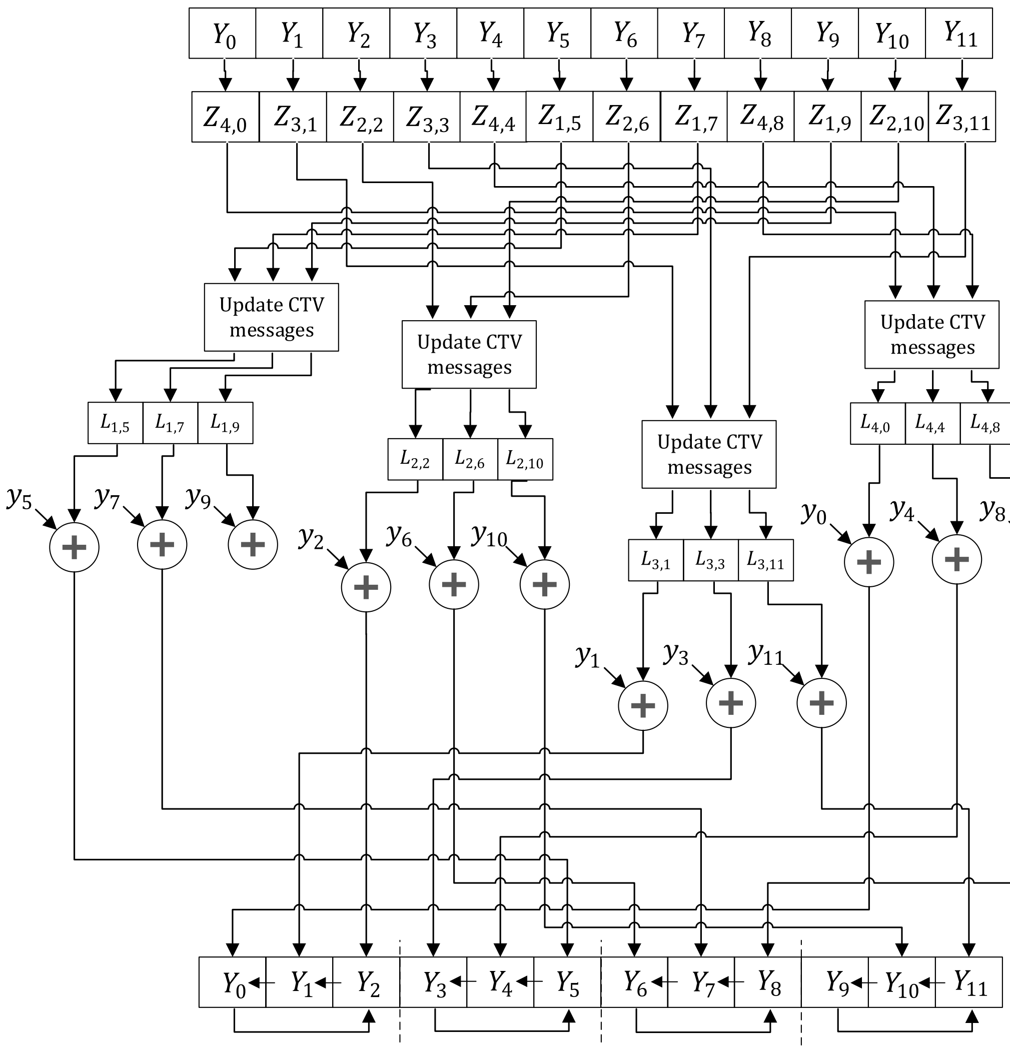

Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7 Y_8 Y_9 Y_{10} Y_{11}
$Z_{4,0}$ $Z_{3,1}$ $Z_{2,2}$ $Z_{3,3}$ $Z_{4,4}$ $Z_{1,5}$ $Z_{2,6}$ $Z_{1,7}$ $Z_{4,8}$ $Z_{1,9}$ $Z_{2,10}$ $Z_{3,11}$
Update CTV messages
$L_{1,5}$ $L_{1,7}$ $L_{1,9}$
$L_{2,2}$ $L_{2,6}$ $L_{2,10}$
$L_{3,1}$ $L_{3,3}$ $L_{3,11}$
$L_{4,0}$ $L_{4,4}$ $L_{4,8}$
y_5 y_7 y_9 y_2 y_6 y_{10} y_1 y_3 y_{11} y_0 y_4 y_8

FIGURE 8. Decoding architecture within the decoding loop.

need for numbering the sequences and tracking their order is inevitable. On one side, this introduces additional hardware overhead to accomplish the numbering and tracking of the sequences. On the other side, it imposes a restriction on the maximum number of sequences which can be decoded at the same time. The latter is due to the memory size which is needed to order decoded codewords. We refer to this parameter as the multiplicity factor which is a positive integer and determines the maximum number of sequences that can be decoded simultaneously. When multiplicity factor is one, there is not any cohesiveness in the decoder and a sequence enters decoder only if the previous sequence has left it. For multiplicity factor bigger than one the decoding throughput is expected to increase however with the price of hardware overhead.

On the other hand, the flip-flops storing APP, CTV and VTC values must be replicated, each for storing values specific to one particular sequence. Fig. 12 shows the effect of this change in the corresponding sections of Fig. 8. The number of replications equals the parameter of multiplicity factor in the design.

V. PERFORMANCE ANALYSIS AND COMPARISON RESULTS

The proposed LDPC decoder has been synthesized and floor-planned in a 28 nm CMOS process with 1.0 V supply. In order to analyze the decoder in terms of chip area, maximum allowed clock frequency, power consumption, and achieved throughput the experimental setup shown in Fig. 13 has been utilized. The entire steps of generating messages, encoding, Binary Phase-Shift Keying (BPSK) modulation, sending over an Additive White Gaussian Noise (AWGN) channel and then demodulation were carried out with MATLAB. The resulting soft-decision sequences are quantized uniformly and converted to fixed-point values in 2's complement format and then saved in a testbench file. This testbench file is utilized during the step of netlist simulation for estimating decoding throughput. Synopsys Design Compiler was used

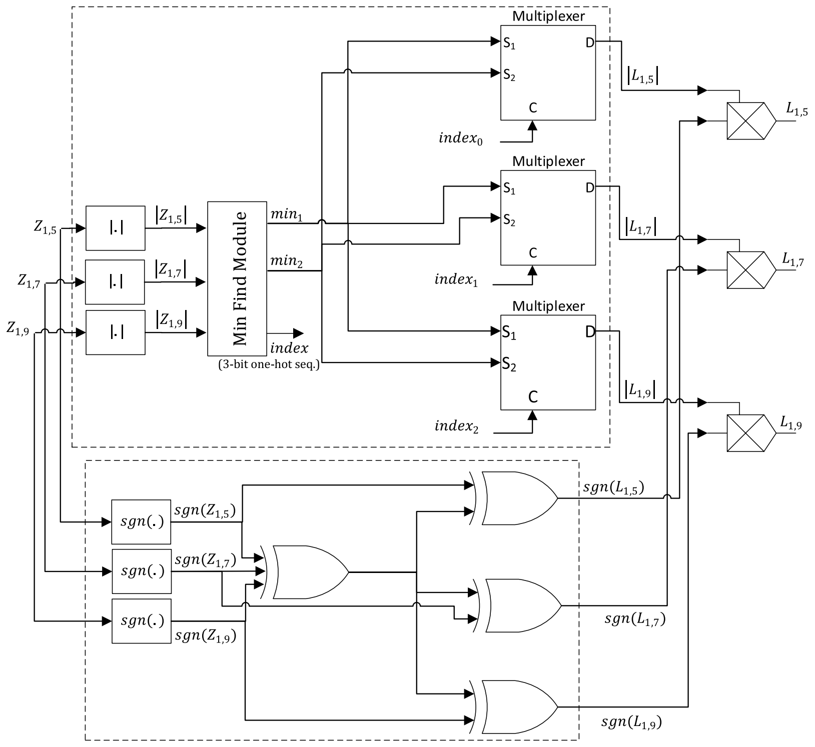


FIGURE 9. Internal structure of a block of "update CTV messages" in Fig. 8.

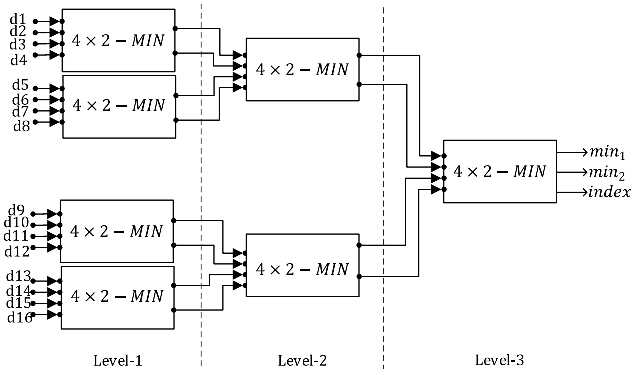


FIGURE 10. (Modified) Rejection-based 16 × 2-MIN module.

TABLE 3. List of parameters with corresponding values used during simulation and synthesis.

Parameter	Value
Synthesis technology	28 nm CMOS
Modulation	BPSK
Channel	AWGN
LDPC codes	IEEE 802.11 $n = 648$ $R = 1/2, 2/3, 3/4, 5/6$
J_{max}	5
Nos. of frames (MATLAB)	1e6
Multiplicity factor	1, 4, 8
Quantization	(6,2), uniform

to generate gate-level netlist of our design. Post-synthesis simulations was carried out with Cadence NCSim-Simulator and static-timing analysis as well as power analysis of the netlist was conducted with Synopsys Prime Time. For the ease of reference, all the used parameters with their preset values are presented in Table 3

Our experiments and simulations have been conducted with four QC-LDPC codes (four code-rates) from IEEE 802.11 standard, although any QC-LDPC code from other communication standards like 5G and WiMax can also be examined. Synthesis results are found in Table 4. The synthesis has been conducted with multiplicity factor of 1, 4, and 8, for each code. This parameter, as discussed in section IV, indicates the maximum number of frames that can be decoded simultaneously.

All syntheses have been conducted with (6,2) quantization bits, meaning that floating-point values are converted to 6-bit fixed-point values in which 2 bits are dedicated to the fractional part and the remaining 4 bits to integer part. This specific choice is based on MATLAB simulation results depicted in Fig. 14 showing that the BER performance degradation with (6,2)-bit fixed-point values compared with the case of floating-point values is negligible. In simple words, allocating more bits during quantization increases the precision, resulting in an improved performance closer to the performance of an ideal LDPC decoder with floating-point values. However, this increases the hardware area and complexity. The E_b/N_0 values in Table 4 for which the BER is 1e-7 are also derived based on the simulations in Fig. 14. The loss in gain at BER of 1e-7 when converting from floating-point to

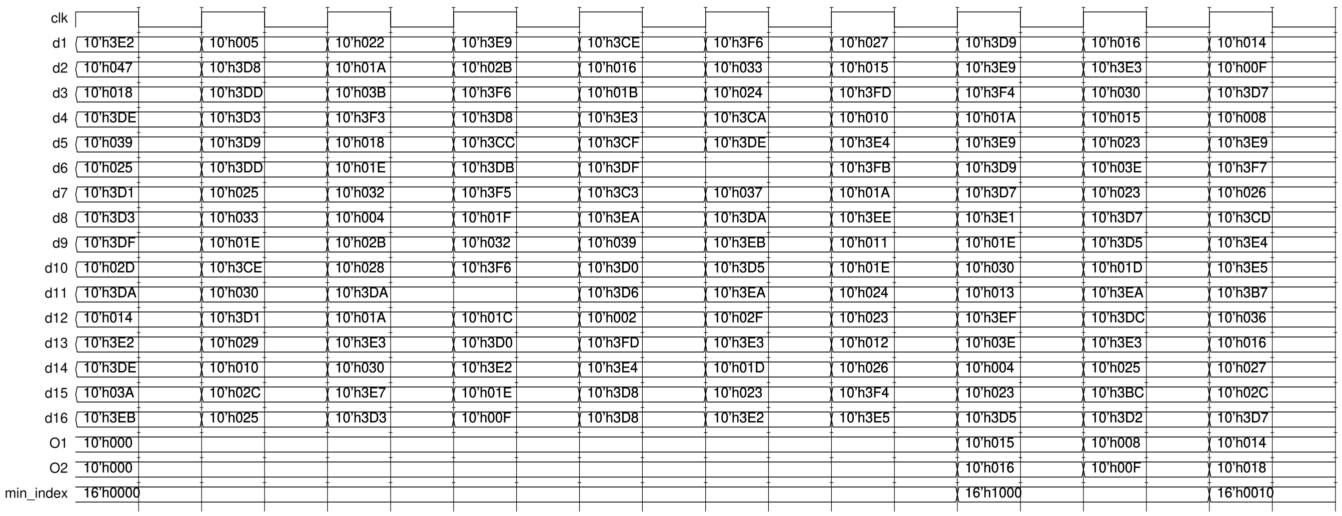


FIGURE 11. RTL simulation of the function of a modified rejection-based 16 × 2-MIN module.

TABLE 4. Implementation results for the proposed high-throughput decoder with multi-frame processing capability in 28 nm technology with $J_{max} = 5$, and $(6, 2)$ quantization.

Code	Eb/N0 [dB] @BER=1e-7	Multiplicity Factor	Area $[mm^2]$	f_{clk} $[GHz]$	Throughput $[Gb/s]$	Energy efficiency $[pJ/b]$
IEEE 802.11 (648,324)	2.95	1	0.038	3.33	3.68	35.54
		4	0.089	2.5	9.5	36.04
		8	0.149	2.5	17.13	35.7
IEEE 802.11 (648,432)	3.84	1	0.041	3.33	4.52	31.01
		4	0.092	3.33	15.88	31.02
		8	0.155	3.33	28.4	36.5
IEEE 802.11 (648,486)	4.86	1	0.040	3.33	6.49	19.61
		4	0.092	3.33	21.71	19.49
		8	0.158	3.33	39.17	18.34
IEEE 802.11 (648,540)	5.5	1	0.043	3.33	7.87	18.49
		4	0.094	3.33	24.74	18.52
		8	0.160	3.33	43.29	15.65

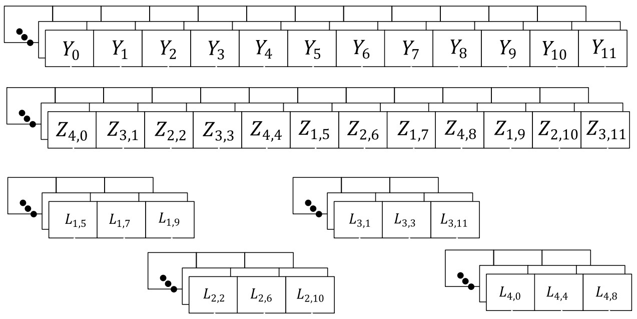

Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7 Y_8 Y_9 Y_{10} Y_{11}
$Z_{4,0}$ $Z_{3,1}$ $Z_{2,2}$ $Z_{3,3}$ $Z_{4,4}$ $Z_{1,5}$ $Z_{2,6}$ $Z_{1,7}$ $Z_{4,8}$ $Z_{1,9}$ $Z_{2,10}$ $Z_{3,11}$
$L_{1,5}$ $L_{1,7}$ $L_{1,9}$
$L_{3,1}$ $L_{3,3}$ $L_{3,11}$
$L_{2,2}$ $L_{2,6}$ $L_{2,10}$
$L_{4,0}$ $L_{4,4}$ $L_{4,8}$

FIGURE 12. Replication of flip-flops storing APP, CTV and VTC values for realization of multi-frame decoding.

fixed-point is nearly 0.1 dB. The average number of iterations needed by the decoder to reach a specific BER performance is also important to consider when examining the effect of quantization. The average number of iterations related to the BER curves of Fig. 14 are shown in Fig. 15, also showing that the increase in number of iterations is negligible when using quantized messages during decoding. It is also worth mentioning that the MATLAB model for acquiring BER curves with fixed-point messages is an exact translation of the implemented decoding architecture shown in Fig. 8 and 12.

As the multiplicity factor rises from 1 to 4 and then 8, throughput also increases largely, while the energy efficiency remains nearly unaltered. Besides, the increase in chip area is with a smaller rate than the increase rate of throughput. To show this evidently Fig. 16 depicts the bar graph of the increase rate of both throughput and chip area for the four examples. In these graphs multiplicity factor of 1 is the baseline for comparison with which the increase rate of throughput and chip area for multiplicity factor of 4 and 8 are compared. The graphs approve that the idea of multi-frame decoding effectively improves the overall decoding throughput with a bearable hardware overhead.

The clock frequency values in Table 4 (and also Table 5) are the maximum values by which the design can work, and they are estimated based on the parameter of Worst Negative Slack (WNS) reported by the synthesis tool. The clock frequency may be increased so long as WNS is positive. When WNS becomes negative, it indicates that the clock frequency is too

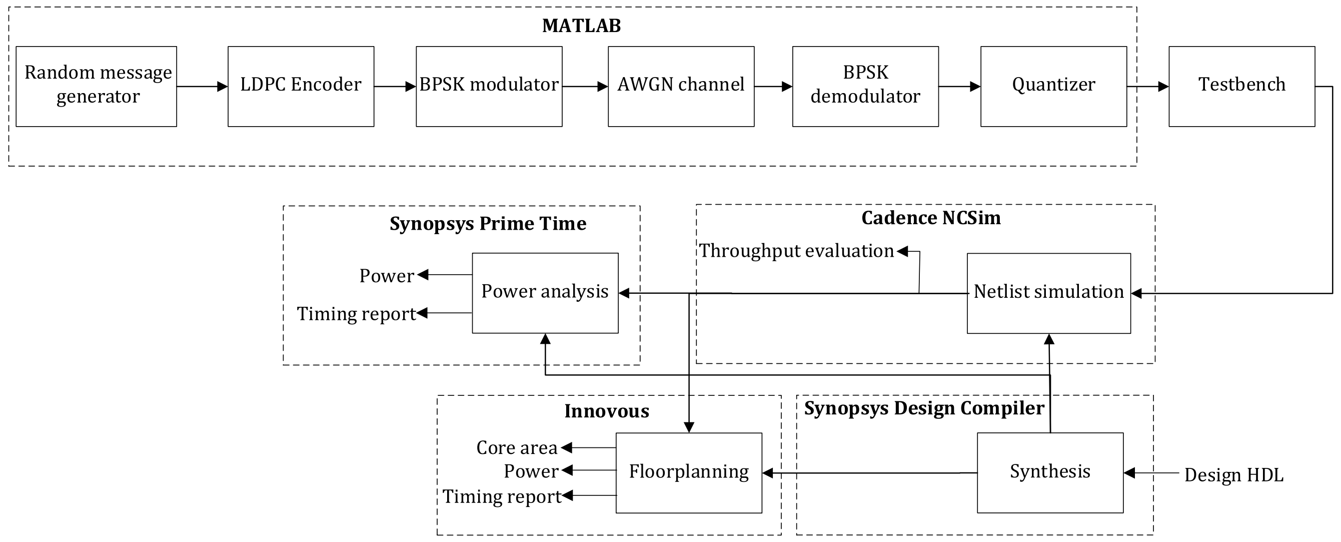


FIGURE 13. Experimental setup for synthesis and floorplanning of the proposed decoder.

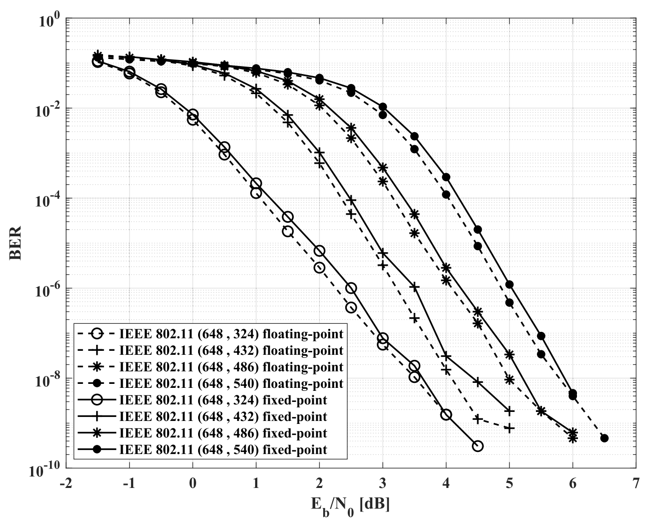


FIGURE 14. Effect of quantization on BER performance of IEEE 802.11 codes with LD.

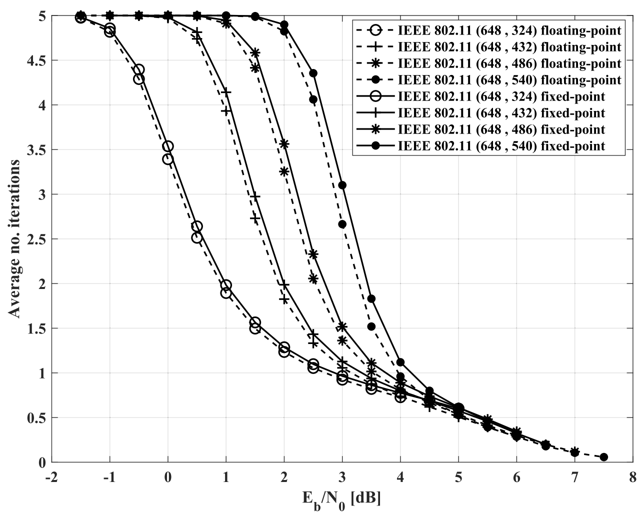


FIGURE 15. Average number of iterations for BER performance of Fig. 14.

fast for data to travel through data paths of the chip within one clock cycle. Therefore f_{clk} must be lowered. Among the four LDPC codes of study, the PCM becomes larger as the code-rate decreases. In summary, decoding complexity is expected to rise as code-rate decreases and multiplicity factor increases. This serves as the reason for smaller clock frequency possible with half-rate code when multiplicity factor is 4 or 8.

The further step of floorplanning was performed only for two codes of minimum and maximum rate, i.e., 1/2 and 5/6. The critical path delay in the two cases are respectively 0.4 and 0.3 ns, yielding the maximum clock frequency of 2.5 and 3.33 GHz. The verified netlist is imported in Cadence-Innovus tool for backend-design process with eight-metal layers. Standard cells, power grids and stripes are placed on the core area. Afterward, clock-tree synthesis and timing optimization are performed. Further steps are power routing, signal routing and insertion of leaf cells, and finally is the post-route static-timing analysis and optimization. Chip layout of the implemented LDPC decoder for the (648,324) code is also shown in Fig. 17.

In order to have a comparison to the state-of-the-art Table 5 presents our results, together with the results of some previous works of the field. The present work has absolute superiority over the state-of-the-art in regard to chip area and clock frequency. The clock frequency the proposed design can work with is at least 3 times the value of other works. The core area in our design is smaller than all other designs, even the design of [21] which has implemented a partially parallel architecture. This is a clear indication that the proposed architecture is of lower complexity, despite the fact that we

TABLE 5. Application-Specific Integrated Circuit (ASIC) results for the proposed architecture and other works.

	This work[§]		[22][§]	[23][*]	[21][§]	[24][§]	[25][§]	[26][§]	[27][§]	[5][§]
Process technology	28 nm		28 nm FD-SOI	28 nm CMOS	28 nm CMOS	28 nm CMOS	28 nm CMOS	65 nm CMOS	65 nm CMOS	90 nm CMOS
LDPC code	(648,324)	(648,540)	(2048 , 1723)	(30000,26786)	(672,336)	IEEE 802.11ad	IEEE 802.11ad	(672 , 546)	(2048 , 1723)	IEEE 802.11
Algorithm	Min-sum		Finite-alphabet	Adaptive degeneration	Min-sun	-	Time-distributed min-sum	Min-sum	Split-row	Min-sum
J_{max}	5		5	49	4	15	10	9	11	10
Quanitization bits	(6,2)		3	5	-	5	5	4	5	5
E_b/N_0 [dB]	2.95 (@ BER = 1e-7)	5.5 (@ BER = 1e-7)	4.95 (@ BER = 1e-7)	10.1 (@ BER = 1e-15)	4 (@ BER = 1e-6)	4.4-5 (@ BER = 1e-6)	4.6-5.4 (@ BER = 1e-6)	-	4.55 (@ BER = 1e-7)	3-4.5 (@ BER = 1e-7)
Architecture	Partial-parallel		Full-parallel	-	Partial-parallel	Parallel	Parallel	Full-parallel	Full-parallel	Partial-parallel
Core area $[mm^2]$	0.22	0.23	16.2	3.73	0.78	0.63	1.99	12.9	4.84	5.2
f_{max} [MHz]	2500	3300	862	373	470	260	202	257	195	336
Throughput [Gbps]	17.13	43.29	588	200	18.4	1.5-12	6.78-16.95	160.8	36.3	1.7-5.13
Power [mW]	890	1012	13350	301	166	6.2-180	104-399	5360	1359	451.3
Energy efficiency [pJ/b]	35.7	15.65	22.70	1.51	9.02	4.13-15	15.34-23.54	33.33	37.13	87.97
Normalized throughput $[Gb/s/mm^2]$	77.86	188	36.3	53.62	23.58	2.38-19.05	3.41-8.52	12.47	7.5	0.33-0.99

[§] Post-layout results; [*] Synthesis results

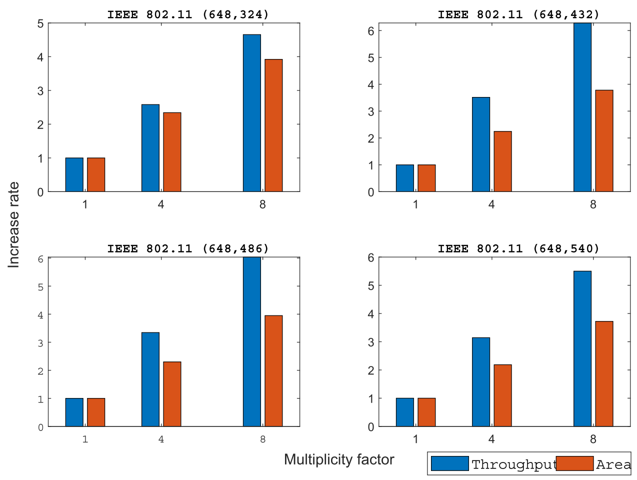


FIGURE 16. Increase rate of throughput and chip area with increasing multiplicity factor for four IEEE 802.11 codes.

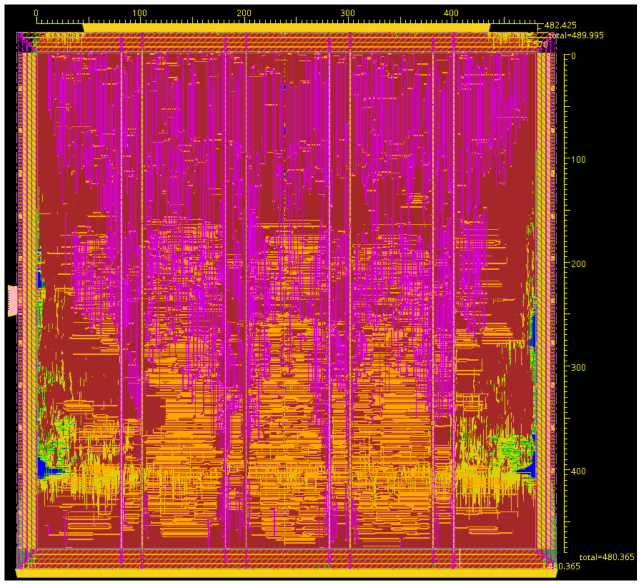


FIGURE 17. Chip layout of the proposed multi-frame LDPC decoder with modified rejection-based method for the (648,324) code.

have considered a bigger quantization precision compared to others. In order to have a more fairer comparison in regard to the decoding throughput the last row of the table holds normalized throughput in each case, in which the achieved throughput has been normalized with respect to chip area. These results show a considerable improvement in the normalized throughput as a result of our multi-frame decoding idea. In terms of power and energy efficiency our design is however insubstantially inferior. This is likely because of the circuitry needed for tracking frame numbers, storing APP, CTV and VTC messages for multiple frames, and also circulating the APP values during the decoding.

VI. CONCLUSION

A modified rejection-based scheme for finding the first two minima and location of the first minimum in a min-sum decoding algorithm of an LDPC code was proposed. In this modified method the location of the minimum is derived as a one-hot sequence instead of an index, thus leading to simplification of the min-sum decoding. In addition, rejection-based scheme allows for further pipelining of the decoding procedure and thus a multi-frame decoding architecture. This idea can effectively increase decoding throughput without prohibitive hardware overhead and thus is a practical idea. Our synthesis and post-layout simulation results in an industrial 28 nm CMOS technology approves the effectiveness of the multi-frame processing in increasing throughput with reasonable hardware overhead.

ACKNOWLEDGMENT

The authors are thankful to the support of the Brandenburg University of Technology (BTU) Cottbus–Senftenberg, Prof. Dr. Jörg Nolte and Dr. Steffen Büchner.

REFERENCES

[1] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, MA, USA: Cambridge Univ. Press, 2008.

[2] J. Chen and M. P. C. Fossorier, "Near optimum universal belief propagation based decoding of low-density parity check codes," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 406–414, Mar. 2002.

[3] M. P. C. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Trans. Commun.*, vol. 47, no. 5, pp. 673–680, May 1999.

[4] A. Hasani, L. Lopacinski, S. Buchner, J. Nolte, and R. Kraemer, "A modified rejection-based architecture to find the first two minima in min-sum-based LDPC decoders," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Seoul, South Korea, May 2020, pp. 1–6.

[5] S. Kumawat, R. Shrestha, N. Daga, and R. Paily, "High-throughput LDPC-decoder architecture using efficient comparison techniques & dynamic multi-frame processing schedule," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 5, pp. 1421–1430, May 2015.

[6] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.

[7] M. M. Mansour and N. R. Shanbhag, "Memory-efficient turbo decoder architectures for LDPC codes," in *IEEE Workshop Signal Process. Syst.*, San Diego, CA, USA, Oct. 16-18, 2002, pp. 159–164.

[8] A. Hasani, L. Lopacinski, S. Buchner, J. Nolte, and R. Kraemer, "A modified shuffling method to split the critical path delay in layered decoding of QC-LDPC codes," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Istanbul, Turkey, Sep. 2019, pp. 1–6.

[9] A. Hasani, L. Lopacinski, and R. Kraemer, "Reduced-complexity decoding implementation of QC-LDPC codes with modified shuffling," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–14, Dec. 2021.

[10] C.-L. Wey, M.-D. Shieh, and S.-Y. Lin, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3430–3437, Dec. 2008.

[11] X.-Y. Shih, C.-Z. Zhan, C.-H. Lin, and A.-Y. Wu, "An 8.29 mm^2 52 mW multi-mode LDPC decoder design for mobile WiMAX system in 0.13 μm CMOS process," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 672–683, Feb. 2008.

[12] Q. Xie, Z. Chen, X. Peng, and S. Goto, "A sorting-based architecture of finding the first two minimum values for LDPC decoding," in *Proc. IEEE 7th Int. Colloq. Signal Process. Appl.*, Penang, Malaysia, Mar. 2011, pp. 95–98.

[13] L. G. Amaru, M. Martina, and G. Masera, "High speed architectures for finding the first two maximum/minimum values," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 12, pp. 2342–2346, Dec. 2012.

[14] W. Cui, Y. Yang, X. Jiang, and S. Kim, "Modified tree structure approach for finding the first two minimum values," in *Proc. IEEE China Summit Int. Conf. Signal Inf. Process. (ChinaSIP)*, Xi'an, China, Jul. 2014, pp. 563–567.

[15] G. Tzimpragos, C. Kachris, D. Soudris, and I. Tomkos, "A low-latency algorithm and FPGA design for the min-search of LDPC decoders," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops*, Phoenix, AZ, USA, May 2014, pp. 269–274.

[16] Y. Lee, B. Kim, J. Jung, and I. C. Park, "Low-complexity tree architecture for finding the first two minima," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 1, pp. 61–64, Jan. 2014.

[17] I. Tsatsaragkos and V. Paliouras, "Approximate algorithms for identifying minima on min-sum LDPC decoders and their hardware implementation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 8, pp. 766–770, Aug. 2015.

[18] J. H. Lee and M. H. Sunwoo, "Low-complexity first-two-minimum-values generator for bit-serial LDPC decoding," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 63, no. 5, pp. 483–487, May 2016.

[19] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic Design with VHDL*. New York, NY, USA: McGraw-Hill, 2009.

[20] A. Hasani, L. Lopacinski, and R. Kraemer. (2020). *A Modified Shuffling Method to Reduce Decoding Complexity of QC-LDPC Codes*. PREPRINT (Version 1) available at Res. Square. [Online]. Available: https://www.researchsquare.com/article/rs-69956/v1

[21] M. Li, J.-W. Weijers, V. Derudder, I. Vos, M. Rykunov, S. Dupont, P. Debacker, A. Dewilde, Y. Huang, L. Van der Perre, and W. Van Thillo, "An energy efficient 18 Gbps LDPC decoding processor for 802.11ad in 28 nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Xiamen, China, Nov. 2015, pp. 1–5.

[22] R. Ghanaatian, A. Balatsoukas-Stimming, T. C. Müller, M. Meidlinger, G. Matz, A. Teman, and A. Burg, "A 588-Gb/s LDPC decoder based on finite-alphabet message passing," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 2, pp. 329–340, Feb. 2017.

[23] K. Cushon, P. Larsson-Edefors, and P. Andrekson, "Low-power 400-Gbps soft-decision LDPC FEC for optical transport networks," *J. Lightw. Technol.*, vol. 34, no. 18, pp. 4304–4311, Sep. 15, 2016.

[24] M. Weiner, M. Blagojevic, S. Skotnikov, A. Burg, P. Flatresse, and B. Nikolic, "A scalable 1.5-to-6 Gb/s 6.2-to-38.1 mW LDPC decoder for 60 GHz wireless networks in 28 nm UTBB FDSOI," in *IEEE ISSCC Dig. Tech. Papers*, San Francisco, CA, USA, Feb. 2014, pp. 464–465.

[25] M. Milicevic and P. G. Gulak, "A multi-Gb/s frame-interleaved LDPC decoder with path-unrolled message passing in 28-nm CMOS," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 10, pp. 1908–1921, Oct. 2018.

[26] P. Schlafer, N. Wehn, M. Alles, and T. Lehnigk-Emden, "A new dimension of parallelism in ultra high throughput LDPC decoding," in *Proc. SiPS*, Taipei, Taiwan, Oct. 2013, pp. 153–158.

[27] T. Mohsenin, D. N. Truong, and B. M. Baas, "A low-complexity message-passing algorithm for reduced routing congestion in LDPC decoders," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 5, pp. 1048–1061, May 2010.

ALIREZA HASANI received the B.Sc. and M.Sc. degrees in communication systems from the Iran University of Science and Technology (IUST), Tehran, Iran. He then joined the Brandenburg University of Technology (BTU), Cottbus, Germany, to pursue his study as a Ph.D. student. During that time, he conducted research on channel coding for high-throughput communications systems with collaboration of the IHP, Frankfurt (Oder), Germany. His research interests include any kind of industrial-oriented research in the field of communication systems.

LUKASZ LOPACINSKI received the M.Sc. degree in computer science from the West Pomeranian University of Technology, Szczecin, Poland, in 2009, and the Ph.D. degree from the Brandenburg University of Technology (BTU) Cottbus–Senftenberg, Cottbus, Germany, in 2017. From 2007 to 2013, he worked in industrial companies in the field of embedded systems and wireless communications. From 2013 to 2016, he was a Research Assistant with the BTU. Since 2016, he has been working with the IHP Microelectronics, Frankfurt (Oder), Germany.

GORAN PANIC was born in Sarajevo, Bosnia and Herzegovina, in 1976. He received the Diploma degree from the Faculty of Electronic Engineering in Nis, University of Nis, Serbia, in 2001, and the Ph.D. degree in engineering from the Technical University Cottbus–Senftenberg, Germany, in 2014. Following his graduation, he worked as a Research Assistant in the area of microprocessor architectures at the University of Nis. He is currently employed as a Scientist at the System Department, IHP, Frankfurt (Oder), Germany. His research interests include SoC design, and low-power and communication systems.

ROLF KRAEMER (Life Member, IEEE) received the Diploma and Ph.D. degrees in electrical engineering and computer-science from RWTH Aachen University, in 1979 and 1985, respectively. He joined the Philips Research Laboratories, in 1985, where he worked in different positions and responsibilities. In 1998, he became a Professor at the Technical University of Cottbus with the joint appointment of the Department Head of Wireless Systems, IHP, Frankfurt (Oder). In the IHP, he leads the Research Department with approximately 70 researchers in topics of high-speed wireless communications, context-aware middleware, sensor networks as well as embedded processors for encryption, and protocol acceleration.