

Further Improvements in Decoding Performance for 5G LDPC Codes Based on Modified Check-Node Unit

Bich Ngoc TRAN-THI^{1,2,3}, Thien Truong NGUYEN-LY^{1,2}, Trang HOANG^{1,2}

¹ Dept. of Electronics, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

² Vietnam National University, Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

³ Faculty of Electrical and Electronics Engineering, Vietnam Aviation Academy, Vietnam

hoangtrang@hcmut.edu.vn

Submitted August 10, 2022 / Accepted March 20, 2023 / Online first May 5, 2023

Abstract. *One of the most important units of Low-Density Parity-Check (LDPC) decoders is the Check-Node Unit. Its main task is to find the first two minimum values among incoming variable-to-check messages and return check-to-variable messages. This block significantly affects the decoding performance, as well as the hardware implementation complexity. In this paper, we first propose a modification to the check-node update rule by introducing two optimal offset factors applied to the check-to-variable messages. Then, we present the Check-Node Unit hardware architecture which performs the proposed algorithm. The main objective of this work aims to improve further the decoding performance for 5th Generation (5G) LDPC codes. The simulation results show that the proposed algorithm achieves essential improvements in terms of error correction performance. More precisely, the error-floor does not appear within Bit-Error-Rate (BER) of 10^{-8} , while the decoding gain increases up to 0.21 dB compared to the baseline Normalized Min-Sum, as well as several state-of-the-art LDPC-based Min-Sum decoders.*

Keywords

Bit error rate, CNU architecture, LDPC codes, low computational complexity, Min-Sum algorithm, Normalized Min-Sum

1. Introduction

Nowadays, the development of science and technology, as well as the explosion of Industry 4.0 such as autonomous systems, Internet of Things (IoTs), machine learning, big data, cloud computing, high-speed data, large storage systems, etc., are more and more powerful. The demand for massive data rates in wired and wireless communication systems requires a very high processing speed of the baseband signal. This is a big challenge for the Error Correction Code (ECC) mechanism. The ECC is known as a technique that is used for controlling errors in data over

unreliable or noisy communication channels [1]. In 1948, Claude E. Shannon showed that it was possible to transmit error-free information over a noisy channel if the data transmission rate was below or equal to the channel capacity limit [2]. Since then, many studies have found new transmission techniques with the aim of getting closer and closer to this limitation. Two well-known ECC mechanisms are Automatic Repeat Request (ARQ) and Forward Error Correction (FEC). In which FEC is a mechanism capable of self-correction without having to retransmit packets like ARQ. However, a major disadvantage of the FEC is that it requires many computing tasks and consumes a large number of hardware resources. In this context, Low-Density Parity-Check (LDPC) code is considered as a potential candidate to solve these issues of FEC. The authors in [3] showed that it was possible for LDPC codes to achieve the decoding performance at only 0.0045 dB from the Shannon limit.

LDPC codes were introduced by Gallager in 1962 [4]. Because of the computational effort in implementing encoders and decoders at that time, LDPC codes were mostly ignored for a long time. In 1996, they were rediscovered by Mackay [5]. LDPC codes are a class of FEC codes defined by sparse parity-check matrices or represented by a bipartite graph [6]. Due to their excellent error correction performance such as high coding gain, low error-floor, low cost, and high throughput capabilities, they have found extensive applications in modern communication systems. For instance, LDPC codes have increasingly adapted in various applications such as in storage devices [7], wired and wireless communication standards [8–10], IEEE 802.11 [11], the Second-Generation Satellite Digital Video Broadcast (DVB-S2) [9], and Advanced Television System Committee (ATSC) [12]. In recent years, many studies have focused on one subclass of LDPC codes known as Quasi-Cyclic LDPC (QC-LDPC) codes [13]. These QC-LDPC codes exhibit significant advantages over other types of LDPC codes such as low complexity [14], [15], friendly hardware implementation [16], [17], and excellent iterative error correction performance [18]. Moreover, they can support a flexible code rate, excellent error correction

performance, fast decoding convergence, and a lower error-floor over a noisy channel [19]. QC-LDPC codes are relatively flexible and can be constructed with multiple code rates, numerous information block lengths, and many different sizes of the submatrix. Structured QC-LDPC codes have been proposed to use simple and regular connections between Variable-Node Units (VNUs) and Check-Node Units (CNU), while still maintaining comparable error correction performance to random codes. Thanks to their outstanding advantages, they have found wide application in modern mobile, storage systems and wireless communication systems [20], [21]. Most recently, QC-LDPC codes were accepted by the 3rd Generation Partnership Project (3GPP) as the channel coding scheme for the enhanced Mobile Broadband (eMBB) data channel of 5G communication [22–24].

Besides the LDPC encoder designs, the area of research on LDPC decoders has also attracted much interest. Generally, LDPC decoding methods can be classified into two categories such as hard decision (i.e., requires only one 1-bit message per graph node) and soft decision (i.e., exchanges multi-bit extrinsic messages along the graph edges). Although the hard decision decoding is very easy to implement and consumes less computational complexity, its error correction performance is not appreciable [25], [26]. The soft decision decoding suffers from high computation load and complex hardware implementation, but its error correction performance can approach the Shannon limit [3]. The common class of decoding algorithms used to decode LDPC codes with the soft decision is Message Passing (MP) iterative algorithm. In MP iterative decoding, messages are passed between Variable-Nodes (VNs) and Check-Nodes (CNs) along the edges of the bipartite Tanner graph [6]. This iterative process continuously proceeds until the correct codeword is successfully found or until the maximum number of iterations has been reached. Depending on the update rules that are used to compute check and variable-node messages, there are several message-passing decoding algorithms, such as Belief-Propagation (BP), Min-Sum (MS), and MS-based algorithms [27–35]. Although MS and MS-based algorithms have lower error correction capacity than BP, they are simpler and more suitable for hardware implementations.

From the perspective of the Min-Sum decoder hardware implementations, Check-Node Unit plays a very important role. It provides the check-to-variable messages by finding the first two minimum values among the variable-to-check messages and also the index of the first minimum. This block significantly affects the decoding performance, the hardware complexity, as well as the maximum operating frequency. A lot of methods in the state-of-the-art aimed to improve the decoding gain, as well as to reduce the hardware complexity or trade-off between them by modifying the CNU architecture. For instance, in [30], a single minimum Min-Sum (smMS) algorithm was proposed, in which only the first minimum value was computed, while the second one was estimated by adding a weight

constant to the first minimum value. This approach provided a significant reduction in CNU hardware complexity. However, this smMS algorithm suffered from high error-floor. By using the same method, a variable weight single minimum Min-Sum (vwsmsMS) algorithm was presented in [31]. In this algorithm, the weight parameter was computed for each iteration or the range of iterations during the decoding process. However, it required an optimization of the weight parameter that depends on the iteration number, the value of the correction factor in the previous iterations, and the Signal-to-Noise Ratio (SNR) values. In [32], the Simplified 2-Dimensional Scaled (S2DS) algorithm was proposed. This algorithm achieved good error correction performance with low hardware complexity. However, it was applied only for short length codes and suffered the early error-floor at high SNR values. The authors in [33] proposed the Second Minimum Approximation MSA (SMA-MSA) in which the first minimum and pseudo-second minimum were found in an approximation way instead of finding the exact two minimum values. As a result, the error correction performance of high-rate LDPC codes in the error-floor region was improved. However, its BER performance deteriorated as the rate decreased. To offer further performance improvement of LDPC decoders, the authors in [34] introduced an Improved Adapted Min-Sum (IAMS) algorithm. In this algorithm, a new CN-update function and the column degree adaptation method aimed to reduce the error-probability of degree-1 VNs. A limitation of this IAMS decoding was that the error-floor occurred around Frame-Error-Rate (FER) of 10^{-5} . The authors in [35] presented an Improved OMS (IOMS) algorithm by using a multiplication factor to modify the check node updating. Although, the IOMS showed a gain of 0.1 dB with respect to the conventional OMS decoder, the major disadvantage of this algorithm is that the BER still suffered from error-floor at BER of 10^{-6} .

Inspired by the state-of-the-art studies above, we recognized that error correction, as well as hardware implementation, could still be optimized further. In this work, firstly, we introduce a modification for the check-node update rule of the MS algorithm for 5G LDPC codes that aims to provide further improvements in decoding performance without error-floor within BER of 10^{-8} . The basic principle of this approach is to apply the correction factors to check-to-variable messages in order to compensate for the overestimation of these messages (known as the main reason for the degradation in the error correction performance and the error-floor phenomenon). To save hardware resources when executing on FPGAs, the correction factors are proposed in the form of the power of 2 or the sum of the power of 2. In this case, they can be implemented by using only the addition and the shift operations. Finally, we propose the CNU hardware architecture and implement it on FPGA to compare the hardware complexity between the proposed CNU architecture and the others.

The rest of this paper is organized as follows. Section 2 gives the short concepts of 5G LDPC codes. Sec-

tion 3 reminds the basis of Min-Sum, as well as the proposed modification decoding algorithm. The proposed Check-Node processing unit architecture is discussed in Sec. 4. Implementation results are presented in Sec. 5. Finally, Section 6 concludes the paper.

2. 5G QC-LDPC Codes

As mentioned before, recently, QC-LDPC codes have been adopted as the channel coding scheme for the enhanced Mobile Broadband (eMBB) in 5G by 3GPP [23]. Compared to Turbo codes, which have already used in 4th Generation Long Term Evolution (4G LTE) [36–38], LDPC codes possess the outstanding advantages as follows:

- Better area throughput efficiency and significantly higher achievable maximum throughput.
- Reduced decoding computational complexity and improved decoding latency (especially when operating at high code rates) due to a higher degree of parallelization.
- Excellent error correction performance.
- Suitable for large codeword lengths and high code rates.

QC-LDPC codes are a special structure of LDPC codes to facilitate implementation in practical applications. They are defined by a base matrix \mathbf{B} of size $L \times C$, with integer entries $b_{i,j} \geq -1$ where $i = \{1, \dots, L\}$ and $j = \{1, \dots, C\}$ as depicted in Fig. 1.

The parity-check matrix \mathbf{H} of a QC-LDPC code can be obtained by its base matrix \mathbf{B} and expansion factor (or the lifting size) Z . To construct the parity check matrix \mathbf{H} , each entry of matrix \mathbf{B} is replaced by a square matrix of size $Z \times Z$ determined by the following rules: entries $b_{i,j} = -1$ are replaced by the all-zero matrix; entries $b_{i,j} \geq 0$ are replaced by a circulant permutation matrix, obtained by right-shifting the identity matrix by $b_{i,j}$ positions. It follows that the parity-check matrix \mathbf{H} is of size $M \times N$, with $M = Z \times L$ rows and $N = Z \times C$ columns.

The QC-LDPC coding scheme in 5G can be described by using two base graphs (i.e., BG1 and BG2) and fifty-one expansion factors Z . These base graphs BG1 and BG2 have a similar structure. The BG1 is targeted for larger information block lengths ($500 \leq K \leq 8448$) and higher code rates ($1/3 \leq R \leq 8/9$), while BG2 is targeted for smaller information block lengths ($40 \leq K \leq 2560$) and lower

$$\mathbf{B} = \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,C} \\ b_{2,1} & b_{2,2} & \dots & b_{2,C} \\ \vdots & \vdots & \ddots & \vdots \\ b_{L,1} & b_{L,2} & \dots & b_{L,C} \end{pmatrix}$$

Fig. 1. Base matrix of QC-LDPC codes.

| Set index j | Expansion factor Z |
|---------------|-------------------------------------|
| 0 | 2, 3, 5, 7, 9, 11, 13, 15 |
| 1 | 4, 6, 10, 14, 18, 22, 26, 30 |
| 2 | 8, 12, 20, 28, 36, 44, 52, 60 |
| 3 | 16, 24, 40, 56, 72, 88, 104, 120 |
| 4 | 32, 48, 80, 112, 144, 176, 208, 240 |
| 5 | 64, 96, 160, 224, 288, 352 |
| 6 | 128, 192, 320 |
| 7 | 256, 384 |

Tab. 1. Expansion factors Z of 5G QC-LDPC codes.

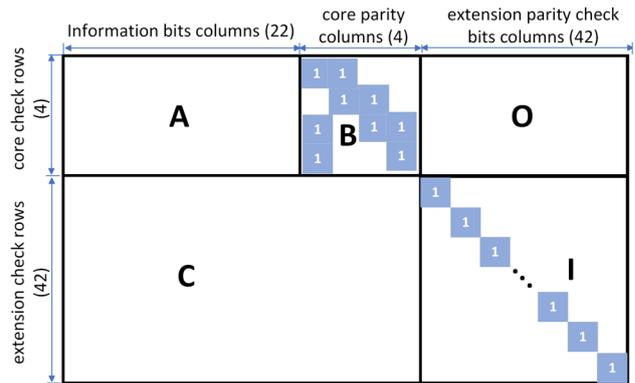


Fig. 2. The structure of the base matrix BG1 proposed for 5G [22] (a 1 in the matrix indicates the existence of a base edge).

code rates ($1/5 \leq R \leq 2/3$). The expansion factor Z is defined by $Z = a \times 2^j$ where the parameter $a = \{2, 3, 5, 7, 9, 11, 13, 15\}$, and $0 \leq j \leq 7$. For the 5G standard, the expansion factor Z has a wide range of values from 2 to 384 as shown in Tab. 1 [22]. Since there is a lot of expansion factors Z , thus, they can support for various information block lengths, K , and code rates, R .

In this paper, we work on the BG1 of size 46×68 , which is illustrated in Fig. 2.

This BG1 consists of five submatrices \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{O} , and \mathbf{I} , where the submatrices \mathbf{A} and \mathbf{B} are called as the core (or the kernel), the \mathbf{C} , \mathbf{O} and \mathbf{I} are the extensions (submatrix \mathbf{O} is an all-zero matrix, and \mathbf{I} is the identity matrix) [22]. It can also be observed that the columns of the BG1 consist of three parts: the information bits part (the first twenty-two columns), the core parity check bits part (the next four columns), and the extension parity check bits part (the last forty-two columns). The rows of the BG1 are divided into two parts: the core check part (the first four rows) and the extended check part (the last forty-two rows). Furthermore, in order to adapt to different information bit lengths and code rates, shortening and puncturing methods are used for the base matrix of 5G LDPC code. Puncturing is applied to both information and parity bits, while shortening is designed just for the information bits (by zero padding) [22], [39].

It is also noticeable that the 5G LDPC codes are dramatically irregular, for both Variable-Node (VN) and Check-Node (CN) sides. For instance, in the base matrix

| Check-node degree (d_c) | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 19 |
|-----------------------------|---|---|----|---|---|---|---|----|----|
| Number of rows in the BG1 | 1 | 5 | 18 | 8 | 5 | 2 | 2 | 1 | 4 |

Tab. 2. Check-node degree statistics of the BG1. (The check-node degree is defined as the number of adjacent edges connected to that check-node.)

| Variable-node degree (d_v) | 1 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 28 | 30 |
|--------------------------------|----|---|---|---|---|---|---|----|----|----|----|----|----|
| Number of columns in the BG1 | 42 | 1 | 1 | 2 | 4 | 3 | 1 | 4 | 3 | 4 | 1 | 1 | 1 |

Tab. 3. Variable-node degree statistics of the BG1. (The variable-node degree is defined as the number of adjacent edges connected to that variable-node.)

BG1, the check-node degree (d_c) and the variable-node degree (d_v) vary largely from 3 to 19 and from 1 to 30, respectively [40]. More specially, there are only 4 rows that exist the highest check-node degree of 19 in the total of 46 rows [20]. The CN degree of the BG1 5G LDPC codes varying over a wide range would make it more difficult in hardware implementation. There are forty-two columns (among sixty-eight columns) of degree-1 VNs. They occupy almost 62% of VNs. These VNs (or coded bits) significantly affect the decoding performance since they are weakly protected and more error-prone. The check-node and variable-node degrees of the BG1 5G LDPC codes are shown in Tabs. 2 and 3, respectively.

3. LDPC Code's Decoding Algorithms

3.1 The Min-Sum Decoding Algorithm

This section presents the brief concepts of LDPC decoding algorithms. LDPC code is a linear block code defined by an $M \times N$ sparse parity-check matrix denoted by \mathbf{H} , where M rows correspond to M check-nodes (or parity-check bits) and N columns correspond to N variable-nodes (or coded bits). The word *sparse* or *low-density* means that the parity-check matrix contains only a few 1's in comparison to the amount of 0's. Let K denote the size of the information bit. K is defined by $K = N - M$. The coded bit n ($1 \leq n \leq N$) is checked by the parity check equation m ($1 \leq m \leq M$) if the entry $H(m,n) = 1$ ($n = 1, \dots, N$; $m = 1, \dots, M$) which are called the neighbors. The neighbor set of the VNs connected to the CN m is defined as $H(m)$ and that of the CNs connected to the VN n as $H(n)$. The number of neighbors of a VN (CN) is called its degree, denoted by d_v (d_c) i.e., $d_v = |H(n)|$ and $d_c = |H(m)|$.

Let's consider a codeword $\mathbf{c} = (c_1, c_2, \dots, c_N)$ is modulated by Binary Phase-Shift Keying (BPSK) and transmitted over the real communication Additive White Gaussian Noise (AWGN) channel. The received information can be

described by $\mathbf{y} = (y_1, y_2, \dots, y_N)$ with $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where \mathbf{x} is the transmitted codeword; $\mathbf{z} = (z_1, z_2, \dots, z_N)$ is an independent Gaussian random variable with zero mean and noise variance $\sigma^2 = N_0/2$ (N_0 is the single-sided noise power density). For the sake of simplicity, the following notations concerning the iterative LDPC decoding will be used throughout this paper.

- γ_n : a priori information of the decoder concerning variable-node n
- $\tilde{\gamma}_n$: a posteriori information (AP) provided by the decoder, concerning variable node n
- $\alpha_{m,n}$: variable-to-check message, i.e., the message sent from variable-node n to check-node m
- $\beta_{m,n}$: check-to-variable message, i.e., the message sent from check-node m to variable-node n

The Message-Passing (MP) iterative algorithm of Min-Sum (MS) decoding is described as follows:

At the initial step, the priori information γ_n is computed for each VN n , and the variable-to-check messages $\alpha_{m,n}$ are initialized accordingly.

$$\gamma_n = \log \frac{\Pr(x_n = 0 | y_n)}{\Pr(x_n = 1 | y_n)}, \quad (1)$$

$$\alpha_{m,n} = \gamma_n. \quad (2)$$

In each iteration:

At CNU processing: each check node processes d_c incoming variable-to-check messages. The outputs of the CNU provide the updated check-to-variable messages, $\beta_{m,n}$, given by:

$$\beta_{m,n} = \prod_{n' \in H(m) \setminus n} \text{sign}(\alpha_{m,n'}) \cdot \left(\min_{n' \in H(m) \setminus n} |\alpha_{m,n'}| \right). \quad (3)$$

Equation (3) can be expressed in another way as:

$$\beta_{m,n} = A \cdot \begin{cases} \min 2 & \text{if } |\alpha_{m,n}| = \min 1 \\ \min 1 & \text{otherwise} \end{cases} \quad (4)$$

where

$$A = \prod_{n' \in H(m) \setminus n} \text{sign}(\alpha_{m,n'})$$

and $\min 1$ and $\min 2$ are the first and second minimum among all the magnitudes of incoming variable-to-check node messages.

At VNU processing: variable-to-check messages (the outputs of VNU) are updated as follows:

$$\alpha_{m,n} = \gamma_n + \sum_{m' \in H(n) \setminus m} \beta_{m',n}. \quad (5)$$

Finally, according to the input γ_n value and the current value of $\beta_{m,n}$ messages, the a posteriori information ($\tilde{\gamma}_n$) is updated as:

$$\tilde{\gamma}_n = \gamma_n + \sum_{m \in H(n)} \beta_{m,n}. \quad (6)$$

After each iteration, the decoder computes a hard-decision vector $\hat{\mathbf{x}}$ (i.e., estimated codeword) and the corresponding syndrome $\mathbf{s} = \mathbf{H}\hat{\mathbf{x}}^T$ where

$$\hat{x}_n = \frac{1 - \text{sign}(\tilde{y}_n)}{2}. \quad (7)$$

The decoder stops when either a codeword has been found (i.e., $\mathbf{s} = \mathbf{0}$) or the maximum number of iterations has been reached.

3.2 The Proposed Decoding Algorithm

Although Min-Sum decoding consumes fewer hardware resources than Belief-Propagation (known as the best decoding algorithm), error correction performance is not good enough because of the overestimation issue of check-node messages. To compensate for this overestimation, many approaches have been proposed in the literature, which aim to improve error correction capacity [27–35]. However, the limitation of these algorithms is the existing error-floor when the SNR is high. In this work, we propose an algorithm that aims to get better error correction performance without error-floor within BER of 10^{-8} with unmarked increasing hardware complexity. The main idea is to modify check-to-variable messages (i.e., the outputs of CNU) at CNU processing. Let $\min 1$, $\min 2$ be the original minimum values as given by (4), and $\min 1'$, $\min 2'$ be the modified values of $\min 1$ and $\min 2$, respectively.

For the proposed algorithm, the CNU processing step is modified by:

$$\beta_{m,n} = A \cdot \begin{cases} \min 2' & \text{if } |\alpha_{m,n}| = \min 1 \\ \min 1' & \text{otherwise} \end{cases} \quad (8)$$

where $A = \prod_{n \in H(m) \setminus n} \text{sign}(\alpha_{m,n})$, $\min 1' = \max\{\min 1 - \delta, 0\}$,

$$\min 2' = \begin{cases} \max\{\min 2 - \delta, 0\} & \min 1 \geq \min 2 - \tau \\ \max\{\min 2 - (\tau + \delta), 0\} & \text{otherwise} \end{cases} \quad (9)$$

$0 < \delta$, $\tau \leq 1$ are the offset factors that are “finely tuned” by simulation. These values depend on the codes and target BER. From the hardware perspective, they should be selected as the power of 2 or the sum of powers of 2, which can be implemented by using only adders and shift registers.

The optimal factors δ and τ are obtained by simulations for the given target BER of 10^{-8} . The optimization procedure for the offset factors δ and τ is described in detail as follows. First, δ is fixed to 0.5 and τ is optimized such that the target BER is met. Next, we fix this optimum τ (just found in the previous step), and perform the same optimization procedure for δ . As mentioned before, to save hardware implementation resources, the values of these parameters are limited to hardware friendly values. Finally, the optimal offset values δ and τ for variant 5G LDPC codes are listed in Tab. 4.

| LDPC codes (N, M) | Code rate (R) | Offset factors | |
|--------------------------|-------------------|----------------|--------|
| | | δ | τ |
| (7424,3200) | 3/5 | 0.5 | 0.375 |
| (8832,4608) | 1/2 | 0.25 | 0.75 |
| (6720,2496) | 2/3 | 0.25 | 0.875 |

Tab. 4. The proposed optimal offset factors for some 5G LDPC.

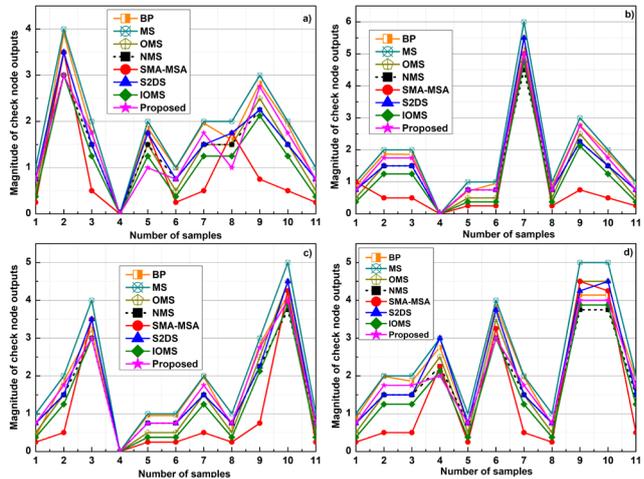


Fig. 3. Comparison of the Check-Node unit outputs among LDPC decoding algorithms.

In this work, we propose a method to adjust check-to-variable messages by using two offset factors instead of one (such as in OMS algorithm). The main reason comes from the fact that through simulations, we realized that by using two offset factors, the gap of check-to-variable messages between the BP and our proposed algorithm is very small. Sometimes, it is even better than the existing MS-based decoding algorithms. By using 2 offset factors, we can fine-tune the check-to-variable messages. This makes check-to-variable messages easier to get closer to the BP algorithm. To demonstrate our proposal, we have compared the check-to-variable messages among our proposal and some various existing MS-based decoding algorithms (e.g. OMS, NMS, SMA-MSA, S2DS, IOMS) including the MS and BP algorithms as shown in Fig. 3.

For the sake of simplicity, we calculated for 4 check-to-variable messages, which correspond to Figs. 3a, 3b, 3c and 3d, respectively. We tested 11 samples. From the simulation results, we can see that the check-to-variable messages of the proposed algorithm is very close to the BP algorithm. This means our algorithm could provide further improvements in decoding performance than other algorithms.

4. The Proposed Check-Node Unit Architecture

The Check-Node Unit (CNU) is known as one of the most complicated components of the LDPC decoder. The main task of CNU is to find the first and second minimum

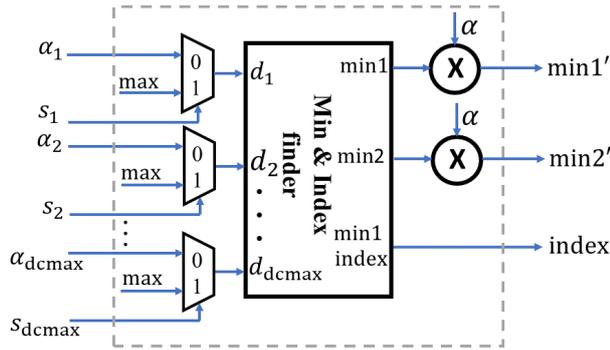


Fig. 4. The baseline CNU architecture for the NMS decoding algorithm (note that the sign part is not shown).

values among the variable-to-check messages (i.e., incoming CNU inputs) and provide the check-to-variable messages to the VNU. This block has a significant impact on the decoding performance, as well as the hardware complexity. In this section, first, we present the design for baseline CNU architecture, and then, we focus on hardware modification for the proposed CNU.

Figure 4 illustrates the baseline CNU architecture of the Normalized Min-Sum (NMS). For the sake of simplicity, only the first two minimum values (denoted by $\min1$ and $\min2$, respectively) among the variable-to-check messages and the index of the first minimum value (denoted by index) are presented, while the signs of the output messages can be simply computed by XORing the adequate signs of input messages. The NMS is a modified version of the Min-Sum decoder that relies on the use of a normalization (or scaling) factor $0 < \alpha < 1$ within the CNU processing step to compensate for the overestimation of check-to-variable messages. The NMS is known as one of the decoding algorithms that are suitable for hardware implementation. For the purpose of simplification, we shall also assume that all the check-nodes have the same degree, which will be denoted by d_{cmax} . In case the check-node is irregular some extra control logics (i.e., s_i) are required in order to “inactivate” the inputs of the “Min & Index finder” block. The “inactivate” in this scenario means the last ($d_{\text{cmax}} - d_c$), for check-nodes of degree $d_c < d_{\text{cmax}}$, will be set to the maximum value. At the inputs, multiplexers are used to select the input data according to either the “real” variable-to-check message (α_i) or the maximum value (max).

As mentioned in Sec. 2, the 5G LDPC codes are extremely irregular. For instance, for the BG1 of 5G LDPC codes have check-node degree (d_c) vary in a long range from 3 to 19. It can be observed that in the worst case, d_{cmax} equals to 19, it means 19 messages are sent from variable-node to check-node at the same time. The structure of the “Min & Index finder” for finding the two first minimum values among 19 incoming messages and the index of the first minimum value is described in more detail in Figs. 5 and 6. It is constructed based on the Tree Structure (TS) architecture proposed in [41], [42]. This provides a low-cost and high-speed approach for hardware implementation [39].

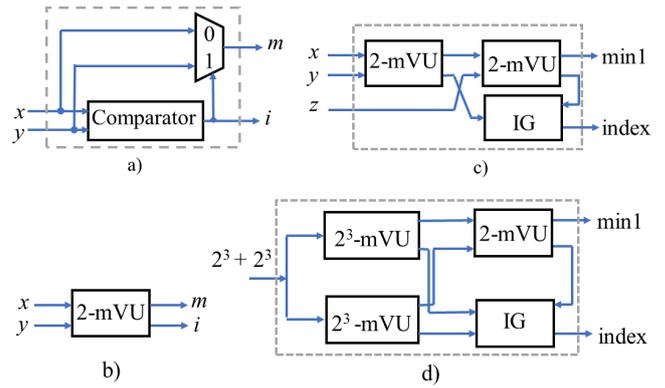


Fig. 5. The architecture of the 16 inputs-minimum value unit (mVU) using the TS approach [41]: a) The architecture of the detail 2-mVU. b) The 2-mVU. c) The 3-input mVU. d) The 16-input mVU.

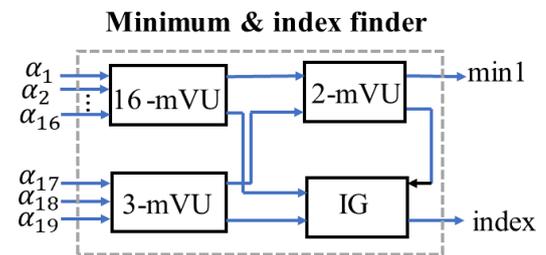


Fig. 6. The architecture of the minimum and index finder of the CNU.

To implement the CNU architecture with 19 inputs (i.e., $d_{\text{cmax}} = 19$), we can decompose them into the sum of 16 and 3 inputs as shown in Fig. 6. Thus, the result of d_{cmax} -mVU block is realized by combining corresponding blocks similarly to the technique used in [41]. The architectures of 3-mVU and 16-mVU are illustrated in Figs. 5c and 5d, respectively. In general, the 2^k -mVU is constructed from the basic minimum Value Unit (mVU) 2-mVU. The 2-mVU includes one comparator and one multiplexer as shown in Fig. 5a. The outputs m and i of the 2-mVU block are defined as follows:

$$m = \begin{cases} x & \text{if } x \leq y \\ y & \text{otherwise} \end{cases}, \quad (10)$$

$$i = \begin{cases} 0 & \text{if } x \leq y \\ 1 & \text{otherwise} \end{cases}. \quad (11)$$

Finally, the index of the first minimum value among d_{cmax} variable-to-check messages is determined by the output of the Index Generator (IG) block (i.e., index). This block is implemented in the same approach in [20].

In this work, the baseline CNU is designed such that it consumes the least of hardware resources. The main idea is to reuse the hardware for finding the first and the second minimum values. Firstly, the first minimum value and its index are determined. Then, we use the same processing unit for finding the second value by “inactivating” (i.e., set to max value) the input message of the “Min & Index finder” at the position of the first minimum value. To improve

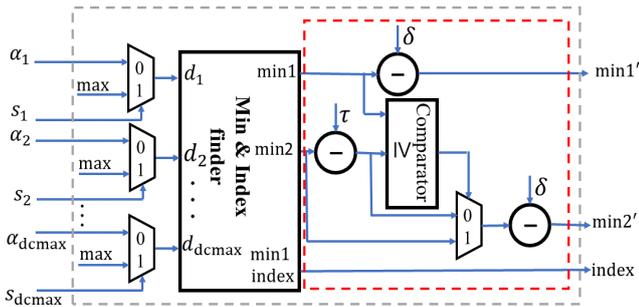


Fig. 7. The proposed CNU architecture for 5G LDPC decoder.

the throughput, the CNU is executed in only two consecutive clock cycles. In the first clock cycle, it executes the first minimum (min1) and its index (index). In the next clock cycle, it finds the second minimum value (min2) by re-utilizing the same hardware architecture. It is worth noting that during the second clock cycle, the CNU block does not impose any penalty on the operating clock frequency.

Figure 7 presents the CNU architecture which performs the proposed algorithm in Sec. 3.2. This architecture is also based on the baseline CNU (as given in Fig. 4) with some adjustments for the value of min1 and min2. The main modifications are indicated inside the red dashed rectangle. As mentioned above, the BG1 of 5G LDPC codes is irregular on both Variable-Node and Check-Node degrees. Additionally, the VNs with low degrees tend to be more prone to error [22]. In particular, on the degree-1 VNs of the extension part of the 5G LDPC codes, the error probability is significantly higher than in other VN degrees. This affects the error correction performance and causes the error-floor. Moreover, the approximation used in the MS and MS-based decoding causes an overestimation of check-node messages, which leads to a degradation in the error-rate performance of the decoder. In state-of-the-art research, the error correction performance can be remarkably improved by modifying the exchange messages (i.e., $\alpha_{m,n}$ and $\beta_{m,n}$). In the proposed CNU architecture, we manipulated the check-to-variable message ($\beta_{m,n}$) by applying the offset factors δ and τ to the min1 and min2 values (at the outputs of “Min & Index finder” block). To reduce the hardware resources when implementing the offset factors with a fix-point number, the offset factors are represented in the form of the power of 2 or the sum of the powers of 2. This means that these coefficients can be implemented through addition and shift operations easily. For instance, in case $\delta = 0.25$ and $\tau = 0.875$, they can be represented by 2^{-2} and the sum of $2^{-1} + 2^{-2} + 2^{-3}$, respectively.

To keep the hardware complexity comparison on an equal basis, we implemented all CNU architectures on the same platform. The proposed CNU and the others are synthesized and implemented using the Kintex device (xc7k70tfbv676-1) in the Xilinx tool Vivado 2019.2. The number of bits used for the representation of exchanged messages ($\alpha_{m,n}$, $\beta_{m,n}$) is 4. The maximum check-node degree is 19 (i.e., $d_{\text{cmax}} = 19$). Table 5 shows the hardware resources on FPGA (post place and route) required to im-

| Decoder | NMS (Baseline) | S2DS [32] | SMA-MSA [33] | IOMS [35] | This work* |
|-----------------|---|-----------|--------------|-----------|------------|
| Device | Xilinx Kintex-7 (xc7k70tfbv676-1) Vivado 2019.2 | | | | |
| LUTs | 182 | 193 | 197 | 185 | 197 |
| FFs | 8 | 11 | 11 | 8 | 12 |
| Max freq. (MHz) | 229 | 245 | 241 | 252 | 230 |

Tab. 5. CNU hardware resources for various MS-based algorithms ($d_{\text{cmax}} = 19$) (The hardware resources are reported for offset factors $\delta = 0.5$ and $\tau = 0.375$).

plement one CNU block for the five decoders under investigation.

It can be seen that the proposed CNU architecture consumes the hardware resources in terms of LUTs and FFs almost the same as S2DS and SMA-MSA ones. It requires a slight increase in hardware resources up to 7.61% of LUTs, but quite high FFs (i.e., 33.33%) compared to the IOMS and baseline NMS CNU architectures. The required number of FFs of the proposed CNU architecture increases compared to the baseline because two offset factors (i.e., δ and τ) are applied in the proposed CNU architecture instead of just one normalization factor (i.e., α) like the baseline NMS. Thus, more registers are required to perform the shift, addition and subtraction operations. Regarding the maximum operating frequency of CNU blocks, it can be seen that the maximum operating frequency of CNU blocks changes minimally (less than 10% compared to the baseline). The proposed CNU architecture provides a maximum frequency of 230 MHz. As mentioned before, by using the same hardware resources to find min1 and min2 values, the operating frequency is not markedly changed.

5. Simulation Results

Finally, to verify the error correction performance of this work, we conducted Monte Carlo simulations for various 5G LDPC codes with the BG1 and the expansion factor Z of 192. The simulation results were obtained using MATLAB R2019a software. Three code rates of 1/2, 2/3, 3/5 and codeword lengths of 8832, 6720, and 7424 are considered. The codeword is modulated in Binary Phase-Shift Keying (BPSK) and transmitted over an Additive White Gaussian Noise (AWGN) channel. To maintain the consistency with hardware design perspective, in the software, we used the same design parameters. More precisely, the priori information (γ_n) and exchanged messages ($\alpha_{m,n}$, $\beta_{m,n}$) are represented by 4-bit fixed point number, while the posteriori information ($\tilde{\gamma}_n$) is quantized by 6-bit. For comparison purposes, we have also included the BER performance curves of the baseline NMS (the normalization factor $\alpha = 0.75$) [29], the S2DS (the scaling factor of 0.75) [32], the SMA-MSA (the optimal factors $\alpha_2 = 0.25$ and $\gamma = 0.75$) [33], the IOMS (the offset factors $\gamma = 0.875$ and $\eta = 0.5$) [35] and the proposed algorithm (the codeword lengths and their optimal offset factors are given in Tab. 4.

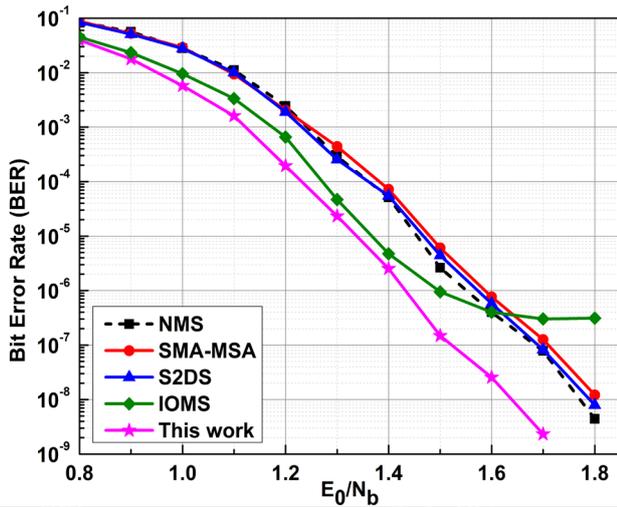


Fig. 8. BER performance of various LDPC decoders for the (8832, 4608), code rate 1/2.

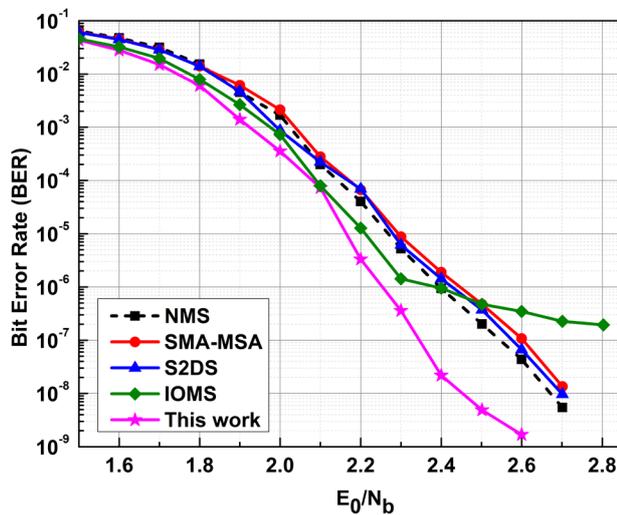


Fig. 9. BER performance of various LDPC decoders for the (6720, 2496), code rate 2/3.

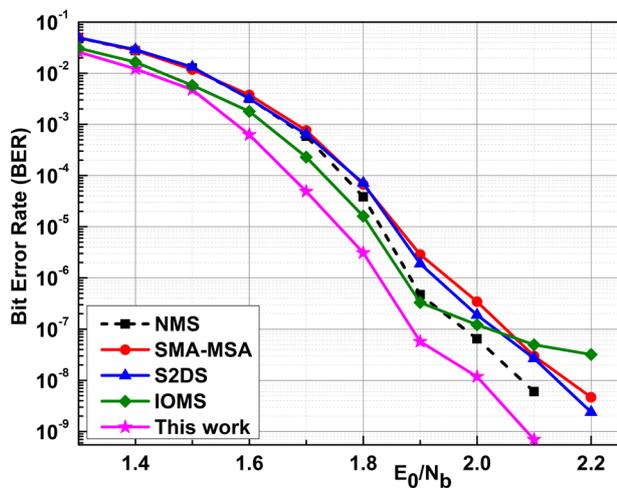


Fig. 10. BER performance of various LDPC decoders for the (7424, 3200), code rate 3/5.

The maximum number of decoding iterations is set to 20. The BER performance curves of five decoders are shown in Figs. 8, 9 and 10.

From the simulation results, it can be seen that the IOMS suffers error-floor earlier than the others. The error-floor begins to appear at a BER of 10^{-6} . Both algorithms S2DS and SMA-MSA have almost the same and very close error correction capacity to the NMS.

It is worth noting that the proposed algorithm is a potential candidate for applications requiring increased decoding performance. It not only achieves the best error correction performance but also does not occur error-floor within BER of 10^{-8} . For instance, at BER of 10^{-8} , the proposed algorithm shows a gain up to 0.21 dB, 0.25 dB and 0.26 dB with respect to the baseline NMS, S2DS and SMA-MS decoders, respectively.

6. Conclusion

In this paper, we proposed a modification to the Check-Node processing update rule and its architecture that aims to provide more improvements in decoding performance for 5G LDPC codes. To do this, first, two optimal offset factors for the particular codes were suggested by simulations at the target BER of 10^{-8} . Then, the CNU architecture was proposed and implemented on FPGA for comparison purposes. Although hardware resources increased, error correction capacity improved significantly. The simulations showed that the proposed algorithm did not exist error-floor within BER of 10^{-8} and provided the decoding gain up to 0.21 dB compared to the baseline NMS, as well as several state-of-the-art LDPC decoders.

Acknowledgments

We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

References

- [1] HAMMING, R. W. Error detecting and error correcting codes. *The Bell System Technical Journal*, 1950, vol. 29, no. 2, p. 147–160. DOI: 10.1002/j.1538-7305.1950.tb00463.x
- [2] SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 1948, vol. 27, no. 3, p. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x
- [3] CHUNG, S. Y., FORNEY, G. D., RICHARDSON, T. J., et al. On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit. *IEEE Communications Letters*, 2001, vol. 5, no. 2, p. 58–60. DOI: 10.1109/4234.905935

- [4] GALLAGER, R. Low-density parity-check codes. *IRE Transactions on Information Theory*, 1962, vol. 8, no 1, p. 21–28. DOI: 10.1109/TIT.1962.1057683
- [5] MACKAY, D. J., NEAL, R. M. Near Shannon limit performance of low density parity check codes. *Electronics Letters*, 1997, vol. 33, no. 6, p. 457–458. DOI: 10.1049/el:19961141
- [6] TANNER, R. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 1981, vol. 27, no. 5, p. 533 to 547. DOI: 10.1109/TIT.1981.1056404
- [7] SUN, H., ZHAO, W., LV, M., et al. Exploiting intracell bit-error characteristics to improve min-sum LDPC decoding for MLC NAND flash-based storage in the mobile device. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 2016, vol. 24, no. 8, p. 2654–2664. DOI: 10.1109/TVLSI.2016.2535224
- [8] TSATSARAGKOS, I., PALIOURAS, V. A reconfigurable LDPC decoder optimized for 802.11 n/ac applications. *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, 2017, vol. 26, no. 1, p. 182–195. DOI: 10.1109/TVLSI.2017.2752086
- [9] KIM, S. M., PARK, C. S., HWANG, S. Y. A novel partially parallel architecture for high-throughput LDPC decoder for DVB-S2. *IEEE Transactions on Consumer Electronics*, 2010, vol. 56, no. 2, p. 820–825. DOI: 10.1109/TCE.2010.5506007
- [10] ANDRADE, J., FALCAO, G., SILVA, V. Flexible design of wide-pipeline-based WiMAX QC-LDPC decoder architectures on FPGAs using high-level synthesis. *Electronics Letters*, 2014, vol. 50, no. 11, p. 839–840. DOI: 10.1049/el.2013.3411
- [11] BALATSOUKAS-STIMMING, A., PREYSS, N., CEVRERO, A., et al. A parallelized layered QC-LDPC decoder for IEEE 802.11 ad. In *2013 IEEE 11th International New Circuits and Systems Conference (NEWCAS)*. Paris (France), 2013, p. 1–4. DOI: 10.1109/NEWCAS.2013.6573590
- [12] MYUNG, S., PARK, S. I., KIM, K. J., et al. Offset and normalized min-sum algorithms for ATSC 3.0 LDPC decoder. *IEEE Transactions on Broadcasting*, 2017, vol. 63, no. 4, p. 734–739. DOI: 10.1109/TBC.2017.2686011
- [13] FOSSORIER, M. P. C. Quasicyclic low-density parity-check codes from circulant permutation matrices. *IEEE Transactions on Information Theory*, 2004, vol. 50, no. 8, p. 1788–1793. DOI: 10.1109/TIT.2004.831841
- [14] LI, J., LIU, K., LIN, S., et al. Decoding of quasi-cyclic LDPC codes with section-wise cyclic structure. In *Proceedings of the IEEE Information Theory and Applications Workshop (ITA'14)*. San Diego (CA, USA), 2014, p. 1–10. DOI: 10.1109/ITA.2014.6804221
- [15] CAI, F., ZHANG, X., DECLERCQ, D., et al. Finite alphabet iterative decoders for LDPC codes: Optimization, architecture and analysis. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2014, vol. 61, no. 5, p. 1366–1375. DOI: 10.1109/TCSI.2014.2309896
- [16] LI, Z., CHEN, L., ZENG, L., et al. Efficient encoding of quasi-cyclic low-density parity-check codes. *IEEE Transactions on Communications*, 2006, vol. 54, no. 1, p. 71–81. DOI: 10.1109/TCOMM.2005.861667
- [17] LIU, H., HUANG, Q., DENG, G., et al. Quasi-cyclic representation and vector representation of RS-LDPC Codes. *IEEE Transactions on Communications*, 2015, vol. 63, no. 4, p. 1033 to 1042. DOI: 10.1109/TCOMM.2015.2399395
- [18] JIANG, N., PENG, K., SONG, J., et al. High-throughput QC-LDPC decoders. *IEEE Transactions on Broadcasting*, 2009, vol. 55, no. 2, p. 251–259. DOI: 10.1109/TBC.2008.2012359
- [19] CHANG, D., YU, F., XIAO, Z., et al. FPGA verification of a single QC-LDPC code for 100 Gb/s optical systems without error floor down to BER of 10^{-15} . In *Optical Fiber Communication Conference (p. OTuN2)*, Optical Society of America. Los Angeles (USA), 2011. DOI: 10.1364/OFC.2011.OTuN2
- [20] THI BAO NGUYEN, T., NGUYEN TAN, T., LEE, H. Low-complexity high-throughput QC-LDPC decoder for 5G new radio wireless communication. *Electronics*, 2021, vol. 10, no. 4, p. 1–18. DOI: 10.3390/electronics10040516
- [21] MA, L., CHOU, H. F., SHAM, C. W. A novel data packing technique for QC-LDPC decoder architecture applied to NAND flash controller. In *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*. Osaka (Japan), 2019, p. 897–898. DOI: 10.1109/GCCE46687.2019.9015393
- [22] RICHARDSON, T., KUDEKAR, S. Design of low-density parity-check codes for 5G new radio. *IEEE Communications Magazine*, 2018, vol. 56, no. 3, p. 28–34. DOI: 10.1109/MCOM.2018.1700839
- [23] ETSI. *5G; NR; Multiplexing and Channel Coding (Release 15), document 3GPP TS 38.212, V15.2.0*, 2018. [Online] Cited 2022-07-31. Available at: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.02_00_60/ts_138212v150200p.pdf
- [24] MAUNDER, R. G. *The 5G Channel Code Contenders. ACCELERCOMM White Paper*, 2016, p. 1–13.
- [25] LUBY, M. G., MITZENMACHER, M., SHOKROLLAHI, M. A., et al. Improved low-density parity-check codes using irregular graphs. *IEEE Transactions on Information Theory*, 2001, vol. 47, no. 2, p. 585–598. DOI: 10.1109/18.910576
- [26] JOSE, R., PE, A. Analysis of hard decision and soft decision decoding algorithms of LDPC codes in AWGN. In *2015 IEEE International Advance Computing Conference (IACC)*. Bangalore (India), 2015, p. 430–435. DOI: 10.1109/IADCC.2015.7154744
- [27] RICHARDSON, T. J., URBANKE, R. L. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 2001, vol. 47, no. 2, p. 599 to 618. DOI: 10.1109/18.910577
- [28] FOSSORIER, M. P. C., MIHALJEVIC, M., IMAI, H. Reduced complexity iterative decoding of low-density parity-check codes based on belief propagation. *IEEE Transactions on Communications*, 1999, vol. 47, no. 5, p. 673–680. DOI: 10.1109/26.768759
- [29] CHEN, J., DHOLAKIA, A., ELEFThERIOU, E., et al. Reduced-complexity decoding of LDPC codes. *IEEE Transactions on Communications*, 2005, vol. 53, no. 8, p. 1288–1299. DOI: 10.1109/TCOMM.2005.852852
- [30] DARABIHA, A., CARUSONE, A. C., KSCHISCHANG, F. R. A bit-serial approximate min-sum LDPC decoder and FPGA implementation. In *2006 IEEE International Symposium on Circuits and Systems*. Kos (Greece), 2006, p. 149–152. DOI: 10.1109/ISCAS.2006.1692544
- [31] ANGARITA, F., VALLS, J., ALMENAR, V., et al. Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2014, vol. 61, no. 7, p. 2150–2158. DOI: 10.1109/TCSI.2014.2304660
- [32] CHO, K., LEE, W. H., CHUNG, K. S. Simplified 2-dimensional scaled min-sum algorithm for LDPC decoder. *Journal of Electrical Engineering & Technology*, 2017, vol. 12, no. 3, p. 1262–1270. DOI: 10.5370/JEET.2017.12.3.1262
- [33] CATALÀ-PÉREZ, J. M., LACRUZ, J. O., GARCIA-HERRERO, F., et al. Second minimum approximation for min-sum decoders suitable for high-rate LDPC codes. *Circuits, Systems, and Signal Processing*, 2019, vol. 38, no. 11, p. 5068–5080. DOI: 10.1007/s00034-019-01107-z

- [34] CUI, H., GHAFARI, F., LE, K., et al. Design of high-performance and area-efficient decoder for 5G LDPC codes. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2020, vol. 68, no. 2, p. 879–891. DOI: 10.1109/TCSI.2020.3038887
- [35] TRAN-THI, B. N., NGUYEN-LY, T. T., HONG, H. N., et al. An improved offset min-sum LDPC decoding algorithm for 5G new radio. In *2021 International Symposium on Electrical and Electronics Engineering (ISEE)*. Ho Chi Minh City (Vietnam), 2021, p. 106–109. DOI: 10.1109/ISEE51682.2021.9418782
- [36] ETSI. *LTE; Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN). 3GPP TS 36.300, V11.6.0, 2013*. [Online] Cited 2022-07-31. Available at: https://www.etsi.org/deliver/etsi_ts/136300_136399/136300/11.06_00/ts_136300v110600p.pdf
- [37] AHN, S. K., KIM, K. J., MYUNG, S., et al. Comparison of low-density parity-check codes in ATSC 3.0 and 5G standards. *IEEE Transactions on Broadcasting*, 2019, vol. 65, no. 3, p. 489–495. DOI: 10.1109/TBC.2018.2874541
- [38] HUI, D., SANDBERG, S., BLANKENSHIP, Y., et al. Channel coding in 5G new radio: A tutorial overview and performance comparison with 4G LTE. *IEEE Vehicular Technology Magazine*, 2018, vol. 13, no. 4, p. 60–69. DOI: 10.1109/MVT.2018.2867640
- [39] LI, H., BAI, B., MU, X., et al. Algebra-assisted construction of quasi-cyclic LDPC codes for 5G new radio. *IEEE Access*, 2018, vol. 6, p. 50229–50244. DOI: 10.1109/ACCESS.2018.2868963
- [40] CUI, H., LE TRUNG, K., GHAFARI, F., et al. An enhanced offset min-sum decoder for 5G LDPC codes. In *2019 25th Asia-Pacific Conference on Communications (APCC)*. Ho Chi Minh City (Vietnam), 2019, p. 490–495. DOI: 10.1109/APCC47188.2019.9026399
- [41] WEY, C. L., SHIEH, M. D., LIN, S. Y. Algorithms of finding the first two minimum values and their hardware implementation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2008, vol. 55, no. 11, p. 3430–3437. DOI: 10.1109/TCSI.2008.924892
- [42] LEE, Y., KIM, B., JUNG, J., et al. Low-complexity tree architecture for finding the first two minima. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2015, vol. 62, no. 1, p. 61–64. DOI: 10.1109/TCSII.2014.2362663

About the Authors ...

Bich Ngoc TRAN-THI was born in Hoa Binh province,

Vietnam. She received her M.S. and B.S. in Physic (Communication System) from Southern Federal University (SFU), Rostov-on-Don, Russian Federation, in 2002 and 2004, respectively. Currently, she is studying doctorate courses in the Department of Electronics, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology. Her research interest includes antennas, wireless communications, Low-Density Parity-Check decoder.

Thien Truong NGUYEN-LY was born in Tien Giang province, Vietnam. He received the B.S. and M.S. degrees in Electronics and Telecommunications Engineering from the Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City, Vietnam, in 2010 and 2012, respectively. He received his Ph.D. degree in Telecommunications Engineering from the Broadband Wireless Systems Laboratory, CEA-LETI, MINATEC Campus, Grenoble, France, and ETIS ENSEA/UCP/CNRS UMR-8051, Cergy-Pontoise, France, in 2018. Currently, he is working as a lecturer at the Faculty of Electrical and Electronics Engineering, HCMUT. His current research interests include error-correction codes, analysis and implementation of Low-Density Parity-Check decoder architectures on FPGA/ASIC platform, speech processing, and embedded system design.

Trang HOANG (corresponding author) was born in Nha Trang city, Vietnam. He received the Bachelor of Engineering, and Master of Science degrees in Electronics-Telecommunication Engineering from Ho Chi Minh City University of Technology in 2002 and 2004, respectively. He received the Ph.D. degree in Microelectronics-MEMS from CEA-LETI and University Joseph Fourier, France, in 2009. From 2009–2010, he did postdoctorate research in Orange Lab-France Telecom. Since 2010, he has been a lecturer, promoted to Associate Professor in 2014, at the Faculty of Electricals–Electronics Engineering, Ho Chi Minh City University of Technology. His field of research interest is in the domain of ASIC/FPGA implementation, speech recognizer, IC architecture, MEMS, fabrication, and security routers.