Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2029560, 10 pages
https://doi.org/10.1155/2021/2029560

Research Article

Reduced-Complexity LDPC Decoding for Next-Generation IoT Networks

Muhammad Asif,[1] **Wali Ullah Khan**,[2] **H. M. Rehan Afzal**,[3] **Jamel Nebhen**,[4] **Inam Ullah**,[5] **Ateeq Ur Rehman**,[6] **and Mohammed K. A. Kaabar**[7,8,9]

[1] *College of Electronics and Information Engineering, Shenzhen University, Shenzhen, Guangdong, 518060, China*
[2] *Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 1855 Luxembourg City, Luxembourg*
[3] *School of Electrical Engineering and Computing, University of Newcastle, NSW, Australia*
[4] *School of Computer Science and Engineering, Prince Sattam Bin Abdulaziz University, Alkharj 11942, Saudi Arabia*
[5] *College of Internet of Things (IoT) Engineering, Hohai University, Changzhou, China*
[6] *Department of Electrical Engineering, Government College University, Lahore 54000, Pakistan*
[7] *Jabalia Camp, United Nations Relief and Works Agency (UNRWA) Palestinian Refugee Camp, Gaza Strip Jabalya, State of Palestine*
[8] *Gofa Camp, Near Gofa Industrial College and German Adebabay, Nifas Silk-Lafto, 26649 Addis Ababa, Ethiopia*
[9] *Institute of Mathematical Sciences, Faculty of Science, University of Malaya, Kuala Lumpur 50603, Malaysia*

Correspondence should be addressed to Mohammed K. A. Kaabar; mohammed.kaabar@wsu.edu

Received 15 April 2021; Revised 19 May 2021; Accepted 24 August 2021; Published 22 September 2021

Academic Editor: Simone Morosi

Low-density parity-check (LDPC) codes have become the focal choice for next-generation Internet of things (IoT) networks. This correspondence proposes an efficient decoding algorithm, dual min-sum (DMS), to estimate the first two minima from a set of variable nodes for check-node update (CNU) operation of min-sum (MS) LDPC decoder. The proposed architecture entirely eliminates the large-sized multiplexing system of sorting-based architecture which results in a prominent decrement in hardware complexity and critical delay. Specifically, the DMS architecture eliminates a large number of comparators and multiplexors while keeping the critical delay equal to the most delay-efficient tree-based architecture. Based on experimental results, if the number of inputs is equal to 64, the proposed architecture saves 69%, 68%, and 52% area over the sorting-based, the tree-based, and the low-complexity tree-based architectures, respectively. Furthermore, the simulation results show that the proposed approach provides an excellent error-correction performance in terms of bit error rate (BER) and block error rate (BLER) over an additive white Gaussian noise (AWGN) channel.

1. Introduction

Internet of things (IoT) will be one of the major trends in next-generation wireless networks for connecting billions of devices to the Internet [1–4]. These communication devices will provide a high data rate with low transmission delay and energy consumption [5–8]. In this regard, low-density parity-check (LDPC) codes [9–15] are one of the most promising candidates in the list of error-control codes and adopted as a primary choice for next-generation IoT networks [16–19]. Compared to other error-correction codes, like Bose-Chaudhuri-Hocquenghem (BCH) codes, Reed Solomon (RS) codes, and turbo codes, LDPC codes have many advantages, e.g., very low error floor, high-speed encoder and decoder, and more varieties in code construction [20–23]. Therefore, LDPC codes have become the focal choice for many communication standards, such as 10-Gigabit Ethernet (802.3an) [24] and Wi-Fi (802.11n/ac/ad) [25–27].

To obtain an optimal performance, LDPC codes are usually decoded with an iterative process between the two decoding phases, i.e., check-node update and variable-node update. Among various decoding algorithms, sum-product (SP) [28] algorithm provides a tremendous decoding performance close to Shannon capacity. However, it suffers from large complexity because of logarithmic and multiplicative functions involved in CNU operation. For hardware implementation of decoder, an area-efficient approximation of SP called min-sum (MS) [29] algorithm was proposed which provides implementation advantages over SP algorithm by computing two minimum values from a set of messages arriving at check nodes. But it suffers from performance degradation. The normalized min-sum (NMS) and offset min-sum (OMS) [30], modified versions of MS, significantly improve the performance of MS by introducing additional normalization and offset factors, respectively.

In hardware implementation of MS decoder, each iteration involves two operations, i.e., CNU and variable-node update (VNU). For CNU, a minimum-value unit (mvu), also called minimum finder, is required to estimate the first two minima $(\mathrm{Min}_1, \mathrm{Min}_2)$ and index of the first minimum value. For large block-length LDPC codes required in high data rate applications, a huge number of minimum-value units are needed to estimate the first two minima and index of Min_1 which significantly increases the complexity of CNU operation. Existing methods require circuitry with high complexity in terms of comparators, multiplexors, latency, and area time. Thus, a low cost algorithm is greatly desired to reduce the complexity of CNU operation of MS decoder.

Recently, some attempts have been utilized to estimate the first two minima from a set of messages arriving at check node. In [31], a single minimum min-sum (smMS) algorithm was proposed which only computes the absolute minimum value and the second minimum value is computed by adding a corrective constant in the first minimum. The smMS provides a significant reduction in hardware complexity of CNU processor, but it suffers from performance degradation. Wang et al. proposed a modification factor min-sum (mfMS) algorithm in [32]; the mfMS algorithm improves the performance of smMS by introducing a modification factor in absolute minimum value. Zhang et al. used the mfMS approach to design a flexible LDPC decoder for multigigabit per second applications [33]. A variable-weight min-sum (vwMS) algorithm was proposed by Angarita et al. in [29] by introducing a variable iteration-based correction factor; the performance of vwMS is better than smMS and mfMS. A simplified variable-weight min-sum (svwMS) is also proposed in [29] which requires low computational cost to determine if more than one input message shares the same first minimum value. In [29, 31–33], the absolute minimum value is calculated first, and then, the second minimum is estimated by applying a modification or correction factor to absolute minimum value. Researches have also investigated various problems on the other related topics of communications [34–44].

Besides the single minimum-based algorithms, some efforts have been made to propose architectures which compute the two minima from a set of messages for CNU operation [45–49]. A sorting-based architecture was proposed by Xie et al. in [46] for finding two minima, but it suffers from large critical delay. Chen-Long et al. proposed a tree-based architecture in [47] which requires some additional complexity but provides critical delay less than that of sorting-based architecture. A low-complexity tree-based architecture [48] was proposed by Lee et al. which reduces some hardware complexity of tree-based structure while keeping the critical delay between those of the sorting-based and tree-based architectures. This manuscript presents an efficient approach, known as dual min-sum (DMS) architecture, for finding the first two minima $(\mathrm{Min}_1$ and $\mathrm{Min}_2)$ from a set of variable nodes participating in CNU operation. Compared to existing sorting-based and tree-based architectures, the proposed scheme efficiently eliminates a large number of comparators and multiplexors while keeping the critical delay almost equal to the tree-based architecture. Based on experimental results, if the number of inputs is equal to 64, the proposed architecture saves 69%, 68%, and 52% area over the sorting-based, tree-based, and low-complexity tree-based architectures, respectively. Furthermore, the simulation results show that the proposed approach outperforms its counterparts by providing an excellent error-correction performance close to NMS algorithm over an additive white Gaussian noise (AWGN) channel.

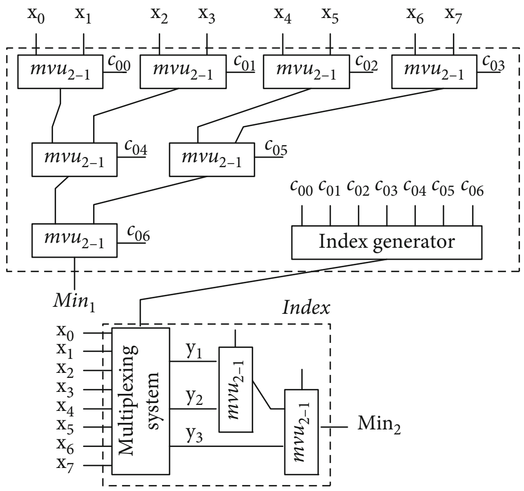

FIGURE 1: Sorting-based search module architecture.

The remainder of this correspondence is arranged as follows. In Section 2, the basic concepts about LDPC codes and min-sum decoding are given. A detailed review of the state-of-the-art architectures for finding the first two minima is given in Section 3. Section 4 presents a proposed architecture to find the first two minima for CNU operation of min-sum LDPC decoder. The performance analysis and hardware implementation of the proposed architecture are given in Section 5, and the conclusion of this correspondence is presented in Section 6.

2. Min-Sum LDPC Decoding

An (N, K) LDPC code can be described by the null space of a $M \times N$ sparse parity-check matrix H, where M denotes to

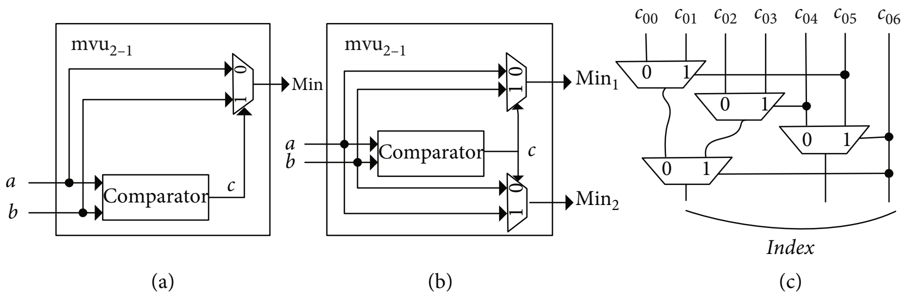

FIGURE 2: Detailed structure of three component units. (a) mvu$_{2-1}$; (b) mvu$_{2-2}$; (c) index generator.

the number of parity-checks and N denotes to the block length of code. It can also be specified by a bipartite graph or Tanner graph having M check nodes and N variable nodes. The check nodes $\{c_1, \cdots, c_M\}$ specify the rows of H and variable nodes $\{v_1, \cdots, c_N\}$ specify the columns of H. The degree of check node (d_c) is equal to the number of nonzero entries in a row of H, and the degree of variable node (d_v) is equal to the number of nonzero entries in a column of H.

Let $N_m = \{n : H_{mn} = 1\}$ denote the set of variable nodes involve in check node c_m and $M_n = \{m : H_{mn} = 1\}$ denote the set of check nodes connected to variable node v_n. Also, let $N_{m\backslash n}$ represent the set N_m with excluding the variable node n and set $M_{n\backslash m}$ represents exclusion of check node m from the set M_n. The log-likelihood ratio (LLR) for a random variable can be defined as $\ln((1-\gamma)/\gamma)$, where γ represents the probability of transmitted bit being equal to zero. In addition, let $\varphi^{(j)}_{n\longrightarrow m}$ denote the LLR message for bit n, sent from variable node v_n to check node c_m in the jth iteration. Similarly, $\psi^{(j)}_{m\longrightarrow n}$ denotes the LLR message for bit n, sent from check node c_m to variable node v_n in the jth iteration. Finally, $w = [w_1, w_2, \cdots, w_N]$ and $r = [r_1, r_2, \cdots, r_N]$ denote the transmitted and the received codewords, respectively. Also, let us assume that $\ell = [\ell_1, \ell_2, \cdots, \ell_N]$ denote the intrinsic reliability provided by the channel. The MS decoding consist of the following steps:

(1) Initialize $j = 1, \cdots, J_{\max}$, where $J_{\max}$ represents the maximum number of iterations

(2) Initialize $\psi^{(j=0)}_{m\longrightarrow n} = 0$, $\forall m\varepsilon\{1, \cdots, M\}$, $\forall n\varepsilon N_m$

(3) VNU function: $\forall n\varepsilon\{1, \cdots, N\}$, $\forall m\varepsilon M_n$

$$\varphi^{(j)}_{n\longrightarrow m} = \ell_n + \sum_{m'\varepsilon M_{n\backslash m}} \psi^{(j-1)}_{m'\longrightarrow n}. \tag{1}$$

(4) CNU function: $\forall m\varepsilon\{1, \cdots, M\}$, $\forall n\varepsilon N_m$

$$\psi^{(j)}_{m\longrightarrow n} = \min_{n'\varepsilon N_{m\backslash n}} \left|\varphi^{(j)}_{n'\longrightarrow m}\right| \prod_{n'\varepsilon N_{m\backslash n}} \text{sign}\left(\varphi^{(j)}_{n'\longrightarrow m}\right). \tag{2}$$

(5) Hard decision: applying a hard decision to compute the transmitted sequence $\widehat{W} = (\widehat{w}_1, \widehat{w}_2, \cdots, \widehat{w}_N)$ as

$$\begin{aligned} \mu_n &= \ell_n + \sum_{m'\varepsilon M_n} \psi^{(j)}_{m'\longrightarrow n}, \\ \widehat{w}_n &= \begin{cases} 0, & \text{if } \mu_n \geq 0, \\ 1, & \text{otherwise.} \end{cases} \end{aligned} \tag{3}$$

If $\widehat{W}H^T = 0$ or the maximum number of iteration $J_{\max}$ is reached, move to Step 6; otherwise, set $j = j + 1$ and go back to Step 3.

(6) Output: declare the estimated sequence $W\wedge^{(j)}$ as the decoder output

As compared to conventional SP and NMS algorithms, although the performance of MS algorithm is lower, it requires much simpler hardware circuitry for CNU operation performed in check-node update processor. In practical implementation of MS decoder, instead of finding the minimum value in (2), two minimum values are computed from the set of messages arriving at check node and a suitable one is selected depending upon the information received at the check node. Thus, the MS decoder reduces the hardware complexity and provides implementation advantages in terms of area and delay. In the next section, we introduce some existing architectures to find the first two minima for CNU operation of MS decoder.

3. Related Architectures

Generally, the hardware circuit used to find the first two minima from a set of messages arriving at check node is known as search module (SM). Let, for a given set of mw-bit messages received at check node, $X = \{x_0, x_1, \cdots, x_{m-1}\}$; SM generates three outputs: (1) the first minimum value of set $\{X\}$, (2) the second minimum value of $\{X\}$, and (3) the index of the first minimum value. For hardware realization, two 2-input units, mvu$_{2-1}$ and mvu$_{2-2}$, are used as the fundamental units of a search module. mvu$_{2-1}$, as shown in Figure 1(a), consists of one comparator and one w-bit 2-to-1 multiplexor and it returns the smaller value from two inputs. mvu$_{2-2}$ consists of one comparator and two w-bit 2-to-1 multiplexors, and it returns both smaller and larger values,

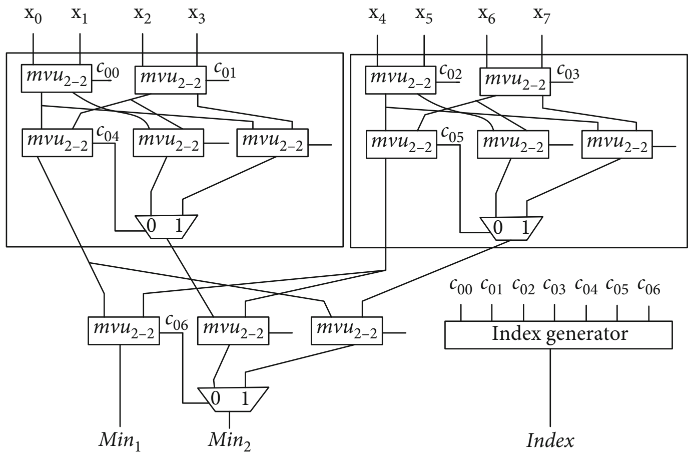

Figure 3: Tree-based search module architecture.

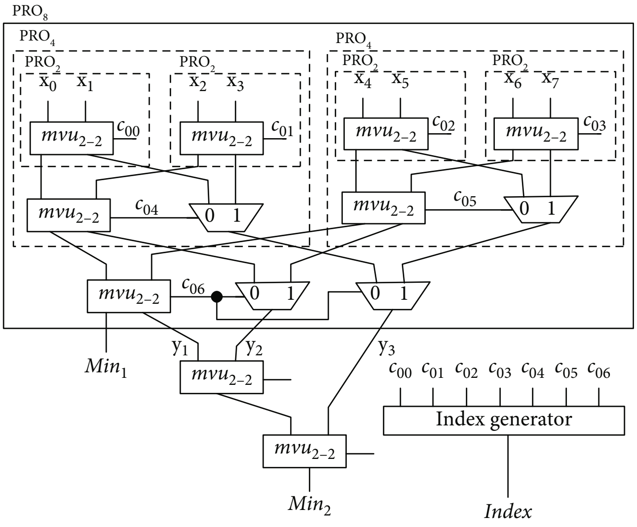

Figure 4: Low-complexity tree-based search module architecture.

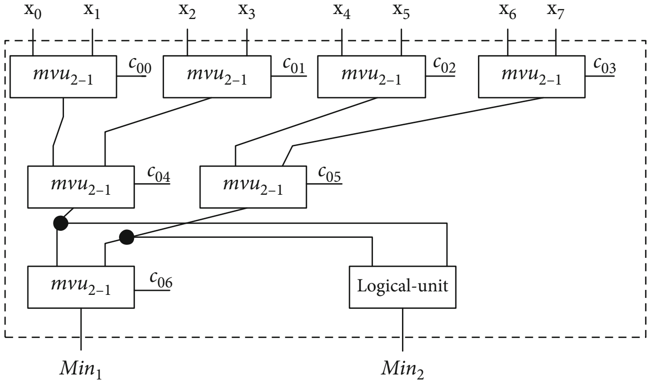

Figure 5: The proposed search module for DMS architecture.

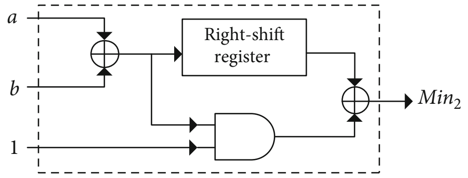

Figure 6: Detailed structure of logical unit for DMS architecture.

as depicted in Figure 1(b). Also, assume m inputs of SM be a power of 2, i.e., $m = 2^k$. If m is not a power of 2, then such SM can be obtained by pruning some leaf nodes of the balanced SM having 2^k inputs as described in previous literatures [45–47]. Next, we present some state-of-the-art architectures to find the first two minima and index of the first minimum value.

The sorting-based SM architecture for eight inputs is depicted in Figure 2. The overall process of sorting-based SM is partitioned into two steps: (1) Min_1 is computed with the binary search tree and (2) an index-controlled multiplexing system is used to compute Min_2. In Figure 1(c), the index of Min_1 can be estimated from comparison results. A set of candidates, $Y = \{y_1, y_2, y_3\}$, is computed by the multiplexing system which employs three 8-to-1 multiplexors to estimate the value of Min_2. Once the set Y is in hand, two mvu_{2-1} are required to compute Min_2. Consequently, the sorting-based SM requires nine 2-to-1 multiplexors, nine comparators, and three 8-to-1 multiplexors for processing eight inputs. But it causes the long critical delay due to serially connected multiplexing system.

The sorting-based architecture is not feasible for high-speed applications because it induces a large critical delay due to serially connected multiplexing system. A tree-based architecture, as depicted in Figure 3, was proposed in [47] for high-speed realization. In tree-based SM, Min_1 and Mi n_2 have almost the same processing time due to the hierarchical tree architecture. Compared to sorting-based SM, it requires more comparators and multiplexors for finding Min_2. Three mvu_{2-1} and one 2-to-1 multiplexor are additionally required for combining two subtrees. But the serially connected multiplexing system is completely removed which reduces the critical delay.

```
Input: a set $X$ of $m$ positive values.
for $j = 1: m$ do
    Step 1
    Partition set $X$ into pairs of values and find the minimum value of each pair. Continue partitioning, and find $\mathrm{Min}_1$ from the last pair of values.
    Step 2
    Input the last pair of values in Step 1 to logical unit, and estimate $\mathrm{Min}_{\sim 2}$.
end for
Output: $X_{\min}=\{\mathrm{Min}_1, \mathrm{Min}_{\sim 2}\}$
```

Algorithm 1: DMS algorithm.

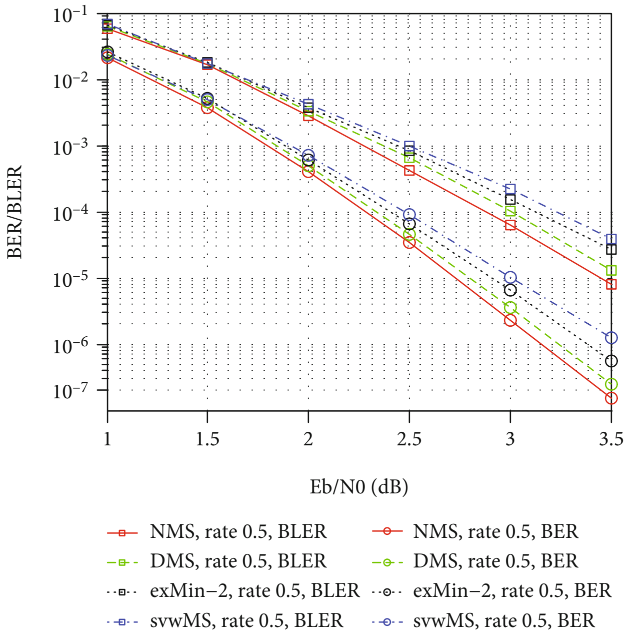

Figure 7: Error-correction performance of the proposed approach and its competitors for the IEEE802.16e (2304, 1152) LDPC code.

The tree-based architecture provides implementation advantages over sorting-based architecture in terms of critical delay, but it is not cost-effective for large block-length LDPC codes. Thus, it has higher hardware complexity that arises from large number of comparators and multiplexors. A low-complexity tree-based architecture was proposed in [48] which reduces the number of comparators while keeping the critical delay between those of the sorting-based and tree-based architectures. A low-complexity tree-based SM, referred to as $\mathrm{SM}_{\mathrm{pro}}$, for eight inputs is depicted in Figure 4 where a PRO_8 unit provides a candidate set, $Y = \{y_1, y_2, y_3\}$, for finding Min_2. A tree structure composed of two mvu_{2-1} is required to find Min_2 from candidate set Y. $\mathrm{SM}_{\mathrm{pro}}$ requires nine comparators and twenty 2-to-1 multiplexors to process eight inputs. Therefore, the existing sorting-based and tree-based search modules are not cost-effective for large block-length LDPC codes. Hence, a low-cost SM architecture is greatly needed for hardware implementation of MS-LDPC decoder. Next, we present SM, known as DMS architecture, which reduces the hardware complexity of MS decoder for large block-length LDPC codes.

4. Proposed Architecture

The complexity of comparators and multiplexors is considerable for hardware realization of the MS-LDPC decoder. A DMS-based SM is presented which reduces a large number of comparators and multiplexors while keeping the critical delay almost equal to the tree-based architecture. The proposed SM is conceptually similar to sorting-based SM. But the serially connected multiplexing system for finding Min_2 is completely removed which reduces the hardware complexity and critical delay. The proposed DMS-based SM estimates the $\mathrm{Min}_{\sim 2}$ value using a logical unit, as depicted in Figure 5. The complexity and delay of logical unit are much less than those of the serially connected multiplexing system. The hardware complexity of both the proposed and sorting-based architectures is the same to find Min_1. But the DMS-based SM estimates the $\mathrm{Min}_{\sim 2}$ using a logical unit which reduces the hardware complexity.

The DMS-based SM for eight inputs is depicted in Figure 5, where seven comparators and seven 2-to-1 multiplexors are required to find Min_1. The logical unit, as depicted in Figure 6, requires two adders, one right-shift register, and one AND gate for estimating Min_2. The first step of DMS approach is to replace the CNU function in (2) with

$$\psi^{(j)}_{m\longrightarrow n} = \min\nolimits_{n'\varepsilon N_m} \left|\psi^{(j)}_{n'\longrightarrow m}\right| \prod_{n'\varepsilon N_m} \operatorname{sign}\left(\psi^{(j)}_{n'\longrightarrow m}\right). \qquad (4)$$

In other words, the sign and output magnitudes are estimated from all N_m variable nodes arriving at check node c_m. The next step is to find the first two minimum values for CNU operation. Let $\lambda^{(j)}_{\min}$ and $\lambda^{(j)}_{\mathrm{sub}}$ denote Min_1 and Min_2, respectively. The magnitude of check-node output is computed as

$$\left|\psi^{(j)}_{m\longrightarrow n}\right| = \begin{cases} \lambda_{\mathrm{sub}} = \left\lceil \dfrac{1}{2}(a+b) \right\rceil, & \text{if } \varphi^{(j)}_{n\longrightarrow m} = \lambda_{\min}, \\ \lambda_{\min}, & \text{otherwise}, \end{cases} \qquad (5)$$

where a and b denote the variable nodes participating in the last mvu_{2-1} of DMS architecture. Thus, the DMS

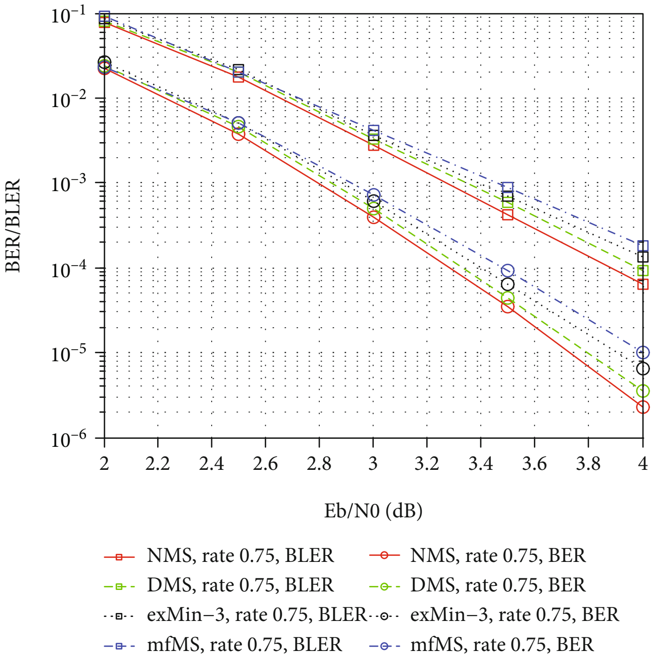

Figure 8: Error-correction performance of the proposed approach and its competitors for the IEEE802.16e (2304, 576) standard LDPC code.

architecture reduces the hardware complexity for CNU operation of the MS-LDPC decoder.

As an illustrative example, assume a set X of eight input values, $X = \{2, 8, 1, 6, 5, 3, 7, 4\}$. Based on Step 1 of the DMS algorithm, set X as partitioned into pair of values as $R = \{\{2, 8\}, \{1, 6\}, \{5, 3\}, \{7, 4\}\}$. Finding the minimum value of each pair, a subset is obtained as $X_1 = \{2, 1, 3, 4\}$. Again, partitioned subset X_1 into a pair of values as $R_1 = \{\{2, 1\}, \{3, 4\}\}$. Finding the minimum value of each pair, we obtain the last pair of values as $\{1, 3\}$ which returns the first minimum value as $\text{Min}_1 = \{1\}$. According to Step 2 of the DMS algorithm, the last pair of values, $\{1, 3\}$, is passed to the logical unit for finding $\text{Min}_{\sim 2}$. Based on (5), $\text{Min}_{\sim 2}$ can be estimated as $\lceil (1/2)(1 + 3) \rceil = 2$. Afterward, the DMS algorithm returns the output as $X_{\min} = \{1, 2\}$. It is important to mention that the DMS algorithm returns Min_1 which is always the first minimum value of set X, but it returns $\text{Min}_{\sim 2}$ which is the estimated second minimum value among the values of X; it may or may not be the exact second minimum value. Consequently, the DMS algorithm provides an efficient architecture which is more cost-effective for large block-length LDPC codes.

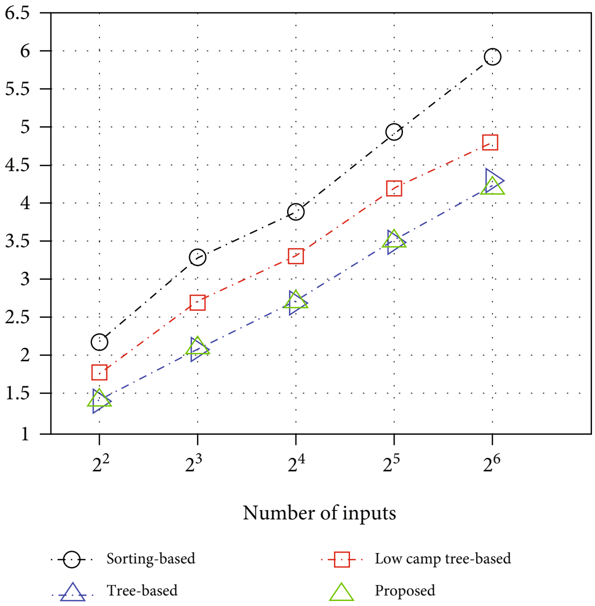

Figure 9: Critical delay of the proposed architecture and its counterparts for different numbers of inputs.

5. Experimental Results

5.1. Performance Analysis. In this section, the error-correction performance of the proposed DMS approach in terms of bit error rate (BER) and block error rate (BLER) is compared with its counterparts under the same conditions. The standard IEEE802.16e LDPC codes with code rates 0.5 and 0.75 having a block length of 2304 are used for evaluating the performance of the proposed and some other existing algorithms. The performance of the proposed approach is compared with the NMS, mfMs, svwMS, and exMin-n [49] algorithms with maximum number of decoding iterations equal to 50. Binary phase-shift keying (BPSK) transmission is assumed over an AWGN channel. Figures 7 and 8 depict the performance analysis for the (2304, 1152) and (2304, 576) IEEE802.16e LDPC codes.

Figure 7 compares the error-correction performance of the proposed DMS algorithm with NMS, svwMS, and

Table 1: Comparisons of hardware complexity for different search modules.

Architecture		Sorting based [46]	Tree based [47]	Low-complexity tree based [48]	Proposed
Components	Comparators	$2^k + k - 2$	$2^{k+1} - 3$	$2^k + k - 2$	$2^k - 1$
	Multiplexors (2-to-1)	$2^k + k - 2$	$3.2^k - 4$	$3.2^k - 4$	$2^k - 1$
	Multiplexors (2^k-to-1)	k	0	0	0
	(Adders, shift registers)	0	0	0	(2, 1)
Critical delay		$(k + \lceil \log_2 k \rceil)\tau_c + \tau_{Mk} + (k + \lceil \log_2 k \rceil)\tau_{M2}$	$k\tau_c + (2k - 1)\tau_{M2}$	$(k + \lceil \log_2 k \rceil)\tau_c + (k + 1 \lceil \log_2 k \rceil)\tau_{M2}$	$k * \tau_c + \tau_{Lu} + k * \tau_{M2}$

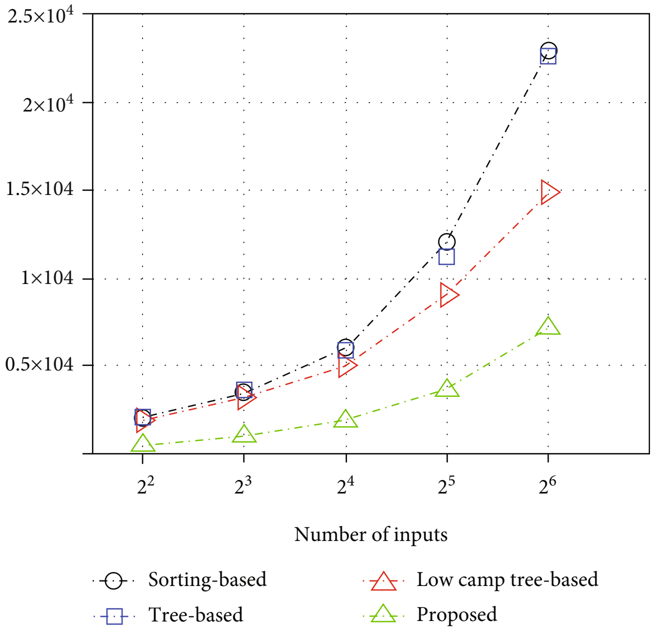

Figure 10: Area complexity of the proposed architecture and its counterparts for different numbers of inputs.

exMin-n, for $n = 2$. Numerical results show that the DMS algorithm provides an excellent error performance close to the NMS algorithm with code rate 0.5 and code length of 2304 for IEEE802.16e standard LDPC code. At a BER of 1 0^{-6}, the DMS algorithm performs very close to NMS with a degradation of 0.09 dB. On the other hand, the exMin-2 and svwMS algorithms perform with a degradation of 0.20 dB and 0.30 dB, respectively.

Similarly, the error-correction performance of the DMS algorithm is also compared with NMS, mfMS, and exMin-n, for $n = 3$, for IEEE802.16e standard LDPC code with code rate 0.75 and a code length of 2304. Figure 9 reveals that the DMS algorithm performs close to the NMS algorithm with a degradation of 0.06 dB at BER of 10^{-5}. But the exMin-3 and mfMS algorithms provide a performance loss of 0.22 dB and 0.26 dB, respectively. As a result, the proposed DMS algorithm outperforms its counterparts under the same conditions by providing an error-correction performance very close to the NMS algorithm.

5.2. Complexity and Speed Performance. As compared to the state-of-the-art architectures [46–48], the proposed DMS architecture reduces the computational complexity for CNU operation of the MS-LDPC decoder. According to Table 1, a comparison of the hardware complexity and critical delay of DMS architecture with sorting- and tree-based architectures is shown, where τ_c, τ_{M2}, τ_{Mk}, and τ_{Lu} denote the delay of comparator, multiplexor (2-to-1), multiplexor (2^k-to-1), and logical unit, respectively. The sorting-based [46] and low-complexity tree-based [48] architectures require $2^k + k - 2$ comparators, and the tree-based [47] architecture requires $2^{k+1} - 3$ comparators to find the first two minima. As the DMS architecture completely removes the multiplexing system inevitable for sorting-based SM, it requires $2^k - 1$ comparators for finding two minima. The sorting-based SM requires $2^k + k - 2$ 2-to-1 and $k2^k$-to-1 multiplexors, where the tree- and low-complexity tree-based architectures require $3.2^k - 4$ comparators to find the first two minima. But the DMS architecture requires $2^k - 1$ multiplexors for finding Min$_1$ and Min$_{\sim 2}$. Also, the DMS architecture additionally requires two adders, one right-shift register, and one AND gate for the implementation of logical unit, but it keeps the critical delay almost equal to that of the tree-based architecture. Consequently, if the number of input values is equal to 16, for example, the DMS architecture eliminates 16.66% comparators compared with the sorting-based and low-complexity tree-based architectures and 48.27% comparators compared with the tree-based architecture. Also, the proposed architecture requires 65.90% less multiplexors compared with the tree-based and low-complexity tree-based architectures.

For fair comparison, four types of architectures are implemented in 6-bit CMOS standard cell library process: the sorting-based [46], tree-based [47], low-complexity tree-based [48], and proposed DMS architectures. Figure 9 depicts the critical delay for four architectures against different numbers of inputs. To the best of our knowledge, the tree-based [47] architecture is assumed to be the best architecture in literature for high-speed realization. Figure 9 shows that the critical delay of the DMS architecture is almost the same as that of the tree-based [47] architecture.

The most area-efficient architecture was proposed by Lee et al. in [48]. Figure 10 shows that when k is equal to 6, the proposed architecture saves 69%, 68%, and 52% area over the sorting-based, tree-based, and low-complexity tree-based architectures, respectively. Consequently, the proposed architecture is proved to be the most area-efficient architecture for high-speed realization. Consequently, the

DMS architecture reduces the hardware complexity of the MS-LDPC decoder for CNU operation.

6. Conclusion

An efficient approach has been proposed to find the first two minima for CNU operation of the MS-LDPC decoder. The proposed architecture is conceptually similar to the sorting-based architecture, but it completely removes the large-sized multiplexing system which results in a prominent reduction in hardware complexity and critical delay. The proposed architecture estimates the second minimum value by utilizing a logical unit circuit having complexity and delay less than those of the multiplexing system. Based on the experimental results, the proposed architecture provides a critical delay almost the same as that of the tree-based architecture. More specifically, the proposed SM eliminates a large number of comparators and multiplexors for CNU operation of the MS-LDPC decoder. Therefore, the DMS architecture saves 69%, 68%, and 52% area over the sorting-based, tree-based, and low-complexity tree-based architectures, respectively. Furthermore, simulation results show that the proposed approach outperforms its competitors in terms of bit error rate (BER) and block error rate (BLER) by providing an excellent error-correction performance over an AWGN channel.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] W. U. Khan, J. Liu, F. Jameel, V. Sharma, R. Jantti, and Z. Han, "Spectral efficiency optimization for next generation NOMA-enabled IoT networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15284–15297, 2020.

[2] X. Li, Q. Wang, M. Liu et al., "Cooperative wireless-powered NOMA relaying for B5G IoT networks with hardware impairments and channel estimation errors," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5453–5467, 2021.

[3] W. U. Khan, F. Jameel, M. A. Jamshed, H. Pervaiz, S. Khan, and J. Liu, "Efficient power allocation for NOMA-enabled IoT networks in 6G era," *Physical Communication*, vol. 39, 2020.

[4] A. Junejo, M. K. A. Kaabar, and S. Mohamed, "Future Robust Networks: Current Scenario and Beyond for 6G," *IMCC Journal of Science*, vol. 1, pp. 67–81, 2021.

[5] X. Li, M. Zhao, Y. Liu, L. Li, Z. Ding, and A. Nallanathan, "Secrecy analysis of ambient backscatter NOMA systems under I/Q imbalance," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12286–12290, 2020.

[6] W. U. Khan, F. Jameel, N. Kumar, R. Jantti, and M. Guizani, "Backscatter-enabled efficient V2X communication with non-orthogonal multiple access," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1724–1735, 2021.

[7] X. Li, Y. Zheng, W. U. Khan et al., "Physical layer security of cognitive ambient backscatter communications for green Internet-of-things," *IEEE Transactions on Green Communications and Networking pp.*, vol. 5, no. 3, pp. 1066–1076, 2021.

[8] W. U. Khan, X. Li, M. Zeng, and O. A. Dobre, "Backscatter-enabled NOMA for future 6G systems: a new optimization framework under imperfect SIC," *IEEE Communications Letters*, vol. 25, no. 5, pp. 1669–1672, 2021.

[9] H. H. Attar, A. A. A. Solyman, M. R. Khosravi, L. Qi, M. Alhihi, and P. Tavallali, "Bit and packet error rate evaluations for half-cycle stage cooperation on 6G wireless networks," *Physical Communication*, vol. 44, p. 101249, 2020.

[10] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, USA, 1963.

[11] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 33, no. 6, pp. 457-458, 1997.

[12] M. Asif, W. Zhou, M. Ajmal, and N. A. Khan, "A construction of high performance quasicyclic LDPC codes: a combinatoric design approach," *Wireless Communications and Mobile Computing*, vol. 2019, 10 pages, 2019.

[13] M. Asif, W. Zhou, Q. Yu, X. Li, and N. A. Khan, "A deterministic construction for jointly designed quasicyclic LDPC coded-relay cooperation," *Wireless Communications and Mobile Computing*, vol. 2019, 12 pages, 2019.

[14] M. Asif, W. Zhou, Q. Yu, S. Adnan, and M. S. Iqbal, "Jointly designed quasi-cyclic LDPC-coded cooperation with diversity combining at receiver," *International Journal of Distributed Sensor Networks*, vol. 16, no. 7, 2020.

[15] M. Asif, W. Zhou, J. S. Ally, and N. A. Khan, "An algebraic construction of quasi-cyclic LDPC codes based on the conjugates of primitive elements over finite fields," in *IEEE International Conference on Communication Technology (ICCT)*, pp. 115–119, 2018.

[16] W. U. Khan, N. Imtiaz, and I. Ullah, "Joint optimization ofNOMA-enabled backscatter communications for beyond5G IoTnetworks," *Internet Technology Letters*, vol. 4, no. 2, 2021.

[17] F. Jameel, "Time slot management in backscatter systems for large-scale IoT networks," in *Wireless-Powered Backscatter Communications for Internet of Things*, pp. 51–65, Springer, Cham, 2021.

[18] W. U. Khan, "NOMA-enabled wireless powered backscatter communications for secure and green IoT networks," in *Wireless-Powered Backscatter Communications for Internet of Things*, pp. 103–131, Springer, Cham, 2021.

[19] F. Jameel, "Multi-tone carrier backscatter communications for massive IoT networks," in *Wireless-Powered Backscatter Communications for Internet of Things*, pp. 39–50, Springer, Cham, 2021.

[20] B. Oudjani, H. Tebbikh, and N. Doghmane, "Modification of extrinsic information for parallel concatenated Gallager/convolutional code to improve performance/complexity trade-offs," *AEU-International Journal of Electronics and Communications*, vol. 83, no. 1, pp. 484–491, 2018.

[21] E. M. Ar-Reyouchi, Y. Chatei, K. Ghoumid, M. Hammouti, and B. Hajji, "Efficient coding techniques algorithm for cluster-heads communication in wireless sensor networks," *AEU-International Journal of Electronics and Communications*, vol. 82, no. 12, pp. 294–304, 2017.

[22] J. Liu and R. C. De-Lamare, "Rate-compatible LDPC codes with short block lengths based on puncturing and extension techniques," *AEU-International Journal of Electronics and Communications*, vol. 69, no. 11, pp. 1582–1589, 2015.

[23] A. Kadi, S. Najah, and M. Mrabti, "An exponential factor appearance probability belief propagation algorithm for regular and irregular LDPC codes," *AEU-International Journal of Electronics and Communications*, vol. 69, no. 6, pp. 933–936, 2015.

[24] Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolic, "An efficient 10GBASE-T ethernet LDPC decoder design with low error floors," *IEEE Journal of Solid-State Circuits*, vol. 45, no. 4, pp. 843–855, 2010.

[25] J. Jin and C. Y. Tsui, "An energy efficient layered decoding architecture for LDPC decoder," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 8, pp. 1185–1195, 2010.

[26] T. H. Tran, Y. Nagao, H. Ochi, and M. Kurosaki, "ASIC design of 7.7 Gbps multi-mode LDPC decoder for IEEE 802.11ac," in *Int. Symp., Commun. and Inf. Technologies (ISCIT)*, pp. 259–263, Incheon, 2014.

[27] S. Ajaz and H. Lee, "Multi-Gb/s multi-mode LDPC decoder architecture for IEEE 802.11ad standard," in *Asia Pacific Conf., Circuits and Systems (APCCAS)*, pp. 153–156, Ishigaki, Japan, 2014.

[28] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, 2005.

[29] F. Angarita, J. Valls, V. Almenar, and V. Torres, "Reduced-complexity min-sum algorithm for decoding LDPC codes with low error-floor," *IEEE transactions on circuits and systems I*, vol. 61, no. 7, pp. 2150–2158, 2014.

[30] J. Chen and M. Fossorier, "Density evolution for two improved BP-based decoding algorithms of LDPC codes," *IEEE Communications Letters*, vol. 6, no. 5, pp. 208–210, 2002.

[31] A. Darabiha, A. Carusone, and F. Kschischang, "A bit-serial approximate min-sum LDPC decoder and FPGA implementation," in *in Proc. IEEE Int. Symp. Circuits Syst*, pp. 149–152, 2006.

[32] Q. Wang, K. Shimizu, T. Ikenaga, and S. Goto, "A power-saved 1Gbps irregular LDPC decoder based on simplified min-sum algorithm," in *in Proc. Int. Symp. VLSI Des., Autom. Test*, pp. 1–4, 2007.

[33] C. Zhang, Z. Wang, J. Sha, L. Li, and J. Lin, "Flexible LDPC decoder design for multigigabit-per-second applications," *IEEE transactions on circuits and systems I*, vol. 57, no. 1, pp. 116–124, 2010.

[34] W. U. Khan, Z. Yu, S. Yu, G. A. S. Sidhu, and J. Liu, "Efficient power allocation in downlink multi-cell multi-user NOMA networks," *IET Communications*, vol. 13, no. 4, pp. 396–402, 2018.

[35] W. U. Khan, F. Jameel, T. Ristaniemi, S. Khan, G. A. S. Sidhu, and J. Liu, "Joint spectral and energy efficiency optimization for downlink NOMA networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 645–656, 2019.

[36] F. Jameel, "Towards intelligent IoT networks: reinforcement learning for reliable backscatter communications," in *2019 IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, 2019.

[37] W. U. Khan, F. Jameel, G. A. S. Sidhu, M. Ahmed, X. Li, and R. Jantti, "Multiobjective optimization of uplink NOMA-enabled vehicle-to-infrastructure communication," *IEEE Access*, vol. 8, pp. 84467–84478, 2020.

[38] F. Jameel, S. Zeb, W. U. Khan, S. A. Hassan, Z. Chang, and J. Liu, "NOMA-enabled backscatter communications: toward battery-free IoT networks," *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 95–101, 2020.

[39] W. U. Khan, "Secure backscatter communications in multi-cell NOMA networks: enabling link security for massive IoT networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 2020.

[40] F. Jameel, "Efficient power-splitting and resource allocation for cellular V2X communications," in *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[41] F. Jameel, "Reinforcement learning for scalable and reliable power allocation in SDN-based backscatter heterogeneous network," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 2020.

[42] M. Nawaz, W. U. Khan, Z. Ali, A. Ihsan, O. Waqar, and G. A. S. Sidhu, "Resource optimization framework for physical layer security of dual-hop multi-carrier decode and forward relay networks," *IEEE Open Journal of Antennas and Propagation*, vol. 2, pp. 634–645, 2021.

[43] S. Yu, W. U. Khan, X. Zhang, and J. Liu, "Optimal power allocation for NOMA-enabled D2D communication with imperfect SIC decoding," *Physical Communication*, vol. 46, p. 101296, 2021.

[44] K. Bakht, F. Jameel, Z. Ali et al., "Power allocation and user assignment scheme for beyond 5G heterogeneous networks," *Wireless Communications and Mobile Computing*, vol. 2019, 11 pages, 2019.

[45] X. Qian, C. Zhixiang, P. Xiao, and G. Satoshi, "A sorting-based architecture of finding the first two minimum values for LDPC decoding," in *IEEE, Int. Coll. on Signal Processing and its Applicattions*, pp. 95–98, 2001.

[46] Q. Xie, Z. Chen, X. Peng, and S. Goto, "A sorting based architecture of finding the first two minimum values for LDPC decoding," in *in Proc. IEEE CSPA*, pp. 95–98, 2011.

[47] W. Chen-Long, S. Ming-Der, and L. Shin-Yo, "Algorithms of finding the first two minimum values and their hardware implementation," *IEEE transactions on circuits and systems I*, vol. 55, no. 11, pp. 3430–3437, 2008.

[48] Y. Lee, B. Kim, J. Jun, and P. In-Cheol, "Low-complexity tree architecture for finding the first two minima," *IEEE Transactions on Circuits and Systems II, Express Briefs*, vol. 62, no. 1, pp. 61–64, 2015.

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