computers

MDPI

Article

An Efficient QC-LDPC Decoder Architecture for 5G-NR Wireless Communication Standards Targeting FPGA

Bilal Mejmaa [1,*], Malika Alami Marktani [2], Ismail Akharraz [3] and Abdelaziz Ahaitouf [1]

[1] Department of Physics, University Sidi Mohamed Ben Abdellah (USMBA), Fez 30000, Morocco
[2] Department of Intelligent Systems, Georesources & Renewable Energies, University Sidi Mohamed Ben Adellah (USMBA), Fez 30000, Morocco
[3] Department of Informatics and Mathematical Engineering, University Ibn Zohr (UIZ), Agadir 80000, Morocco
* Correspondence: bilal.mejmaa@usmba.ac.ma

Abstract: This novel research introduces a game-changing architecture design for Quasi-Cyclic Low-Density Parity-Check (QC-LDPC) decoders in Fifth-Generation New-Radio (5G-NR) wireless communications, specifically designed to meet precise specifications and leveraging the layered Min-Sum (MS) algorithm. Our innovative approach presents a fully parallel architecture that is precisely engineered to cater to the demanding high-throughput requirements of enhanced Mobile Broadband (eMBB) applications. To ensure smooth computation in the MS algorithm, we use the Sub-Optimal Low-Latency (SOLL) technique to optimize the critical check node process. Thus, our design has the potential to greatly benefit certain Ultra-Reliable Low-Latency Communications (URLLC) scenarios. We conducted precise Bit Error Rate (BER) performance analysis on our LDPC decoder using a Hardware Description Language (HDL) Co-Simulation (MATLAB/Simulink/ModelSim) for two codeword rates (2/3 and 1/3), simulating the challenging Additive White Gaussian Noise (AWGN) channel environment.

Keywords: wireless communication; 5G-NR standards; LDPC codes; min-sum algorithm; HDL Verifier; co-design

check for updates

Citation: Mejmaa, B.; Marktani, M.A.; Akharraz, I.; Ahaitouf, A. An Efficient QC-LDPC Decoder Architecture for 5G-NR Wireless Communication Standards Targeting FPGA. *Computers* **2024**, *13*, 195. https://doi.org/10.3390/computers13080195

Academic Editor: Hamed Taherdoost

Received: 9 July 2024
Revised: 10 August 2024
Accepted: 12 August 2024
Published: 14 August 2024

1. Introduction

Clear communication has always been hindered by the problem of unreliable or noisy channels, ever since telecommunication systems and information theory were developed. With the prevalence of cellular communication and the transmission of sensitive data, the problem has seen a significant rise [1].

Richard Hamming invented the first Error Correction Code (ECC) in 1950 [2] to control and correct errors in data transmission over noisy channels. The encoding component of the ECC's error control procedure includes the addition of redundant information to address potential errors across the message. The decoding process automatically detects and corrects errors once the data are received.

In the context of 5G, significant advancements have been made in the error correction process, as depicted (see Figure 1). This process involves CRC (Cyclic Redundancy Check) verification, where a short, fixed-size check-sum is initially computed based on the information packet. On the other hand, the CRC check recalculates this checksum and compares it to the one received to ensure data integrity. Block segmentation serves the purpose of splitting a longer data stream into smaller segments. Conversely, block concatenation involves combining smaller data blocks into a continuous stream. The rate matching/adaptation technique is employed to synchronize the data rates across various components of the system, ensuring efficient data transfer without overloading certain parts of the system. Bit interleaving involves merging bits from multiple source data streams into a unified stream, following a particular pattern.

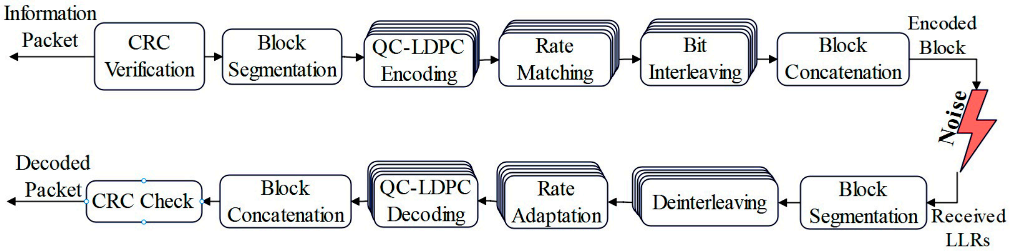

Figure 1. 5G-NR data transmission blocks.

The BER and Signal-to-Noise Ratio (SNR) metrics are employed to characterize the decoding capabilities [3,4]. The SNR is defined as the signal power divided by the noise power, whereas the BER is the number of errors divided by the total number of transmitted bits.

The main contribution of coding theory is to achieve lower SNRs for the same BER [5,6]. If an uncoded system has a BER of 10^{-4} at an SNR level of 4 dB, the corresponding coded system should have the same BER at a lower SNR (3.5 dB). In this case, the coding gain is 0.5 dB, which will decrease the signal power for the same BER. After being long ignored, LDPC codes have become one of the major codes for contemporary standards, as in the second-generation satellite digital video broadcast (DVB-S.2) and 5G-NR wireless communication [7,8]. They were invented by Robert Gallagher at the beginning of the 1960s [9]. Recently, they have been adopted for use in many significant communication systems, such as the Consultative Committee for Space Data Systems (CCSDS) [10].

The min-sum algorithm, a foundational technique in iterative decoding for LDPC codes, was initially derived as a simplified approximation of the Belief Propagation (BP) algorithm. Its efficiency and relatively low computational complexity have made it a popular choice for practical implementations. Subsequent variants such as Offset Min-Sum (OMS) [11], Approximate Min-Sum (AMS) [12], and Scaling Min-Sum (SMS) [13] have been introduced to enhance performance by incorporating additional approximations.

The OMS algorithm introduces an offset value to the message passing process, aiming to improve error floor performance. AMS simplifies the min-sum computations by using approximations, potentially reducing computational complexity. Finally, SMS applies scaling factors to the messages to enhance decoding performance.

The authors in [11] proposed a fully parallel architecture built on the first base graph (BG1). The internal modules of this architecture are equipped with multiplexers that have been designed based on the columns of the base graph. This has led to 19 parallel 68:1 multiplexers, which may require additional resources as they are constructed based on columns instead of rows. Additionally, the circular shifting blocks in this architecture receive addresses from a ROM memory, which can limit its design flexibility based on the characteristics of the base graph. The research outlined in reference [12] strives to improve the MS algorithm for decoding fixed-point 5G-LDPC codes. While the “improved adapted MS (IAMS)” algorithm showcased promising results in the study, the conducted decoding simulation (utilizing BG1 with R = 2/3 and Z = 104) suffered from alarming latency issues, requiring an excessive number of 15 iterations to achieve satisfactory decoding.

In their research paper [14], the authors introduced a sequential check node design to estimate the second smallest value in the MS algorithm by incorporating a variable weight parameter to the first minimum value. This design approach reduces hardware complexity by eliminating the need to calculate the second minimum value. However, the decoding process using this check node structure still poses challenges in terms of flexibility and latency, requiring up to 20 iterations. Furthermore, attaining the ideal level of parallelism becomes challenging due to the partially parallel architecture implemented in both the [12,14] designs.

To the best of our knowledge, despite extensive research on LDPC decoding, no prior investigations have succeeded in developing a single module for 5G-LDPC decoding that seamlessly combine reduced complexity, decreased latency, and superior performance, thus highlighting the novelty and significance of our study. This article aims to revolutionize 5G-

LDPC decoding by introducing a ground-breaking decoder tailored explicitly for extended codeword lengths, emphasizing flexibility and reduced latency as its core objectives. Due to the 5G base graphs' smaller number of rows compared to the columns, the read and write networks are built using Mp:1 and 1:Mp multiplexers (based on rows) to alleviate the interconnection blocks. The shifting amounts specified as base graph entries can be utilized as inputs of the decoder thanks to the flexible shifting block design proposed in our architecture. Moreover, our architecture empowers the model to seamlessly adapt to diverse structures of the first base graph (BG1), liberating the decoder from the confines of a singular application and rendering it invaluable across a multitude of assorted applications. An optimal low-latency method [15] is included in the control node processing block with very satisfactory results in terms of decoder latency. We have modified this approach and applied it to our 5G-NR decoder for the first time. Following the simulation of this architecture, our decoder achieves a remarkable feat by accomplishing the error correction process in a remarkably low number of iterations—a mere six. The discussed performance is substantiated through extensive simulations conducted across a range of iterations and code rates, affirming the exceptional efficiency and effectiveness of our design.

The rest of the article is structured as follows. Preliminaries for LDPC codes in the 5G-NR standards are presented in Section 3. Next, a digital representation of the internal blocks of the proposed architecture is given in Section 4. We discuss the results and related analyses in Sections 5 and 6. Finally, the article concludes in Section 7.

2. Materials and Methods

Initially, the structure was constructed in Simulink using fundamental blocks to create a basic model consisting of 280 bits as a codeword. Subsequently, this design was transcribed into hardware description language (VHDL) and tested using ModelSim SE 10.6 software, focusing on a single frame. The development of this architecture paved the way for enhanced versions, as elaborated in the results section. A co-simulation process was undertaken to evaluate the decoder's performance-based on signal-to-noise ratio (SNR) and bit error rate (BER). The co-simulation process utilizes two software components: MATLAB/Simulink r2022b handles all the blocks of the communication system, incorporating the AWGN model for flexibility and future modifications, whereas the HDL simulator (such as Vivado, ModelSim, or Quartus) describes the decoding process.

3. LDPC Codes in 5G-NR

LDPC codes have been adopted by 5G standards because of their remarkable error correction performance for high codeword lengths and built-in parallelism for hardware implementation [16]. They specify a sparse M × N Parity-Check Matrix (PCM) H that contains a much smaller number of 1s compared to 0 s [17]. For Hi, j represents the i-th column and j-th row of H, with $N - M = K$ information bits and M parity bits, where the code rate is denoted as $R = K/N$. Furthermore, N and M represent the number of variable nodes (VNs) and check nodes (CNs), respectively. The j-th column represents the j-th VN (vj), and the i-th row represents the i-th CN (ci) [18].

Considering C = [x1, x2, . . ., xN], where xi ϵ [0, 1], C is a codeword if and only if

$$\mathrm{H} \cdot \mathrm{C}^{\mathrm{T}} = 1 \quad (1)$$

The weight (dci, dvi) is defined as the number of 1s within the rows and columns, respectively. The LDPC codes are classified into two categories; a regular code is when both dci and dvi are constant, otherwise, it is characterized as an irregular code [19].

During each iteration, the sum-product algorithm (SPA) [20] computes the extrinsic information of each node using the data received from their connected nodes.

$$l_n = \log(P_r(x_i = 0/r_i)/P_r(x_i = 1/r_i)) = 2Y_n/\sigma^2 \quad (2)$$

The a priori data (Y_n) refer to the received information (r_i) that represents the decoder input. The a posteriori (l_n) is the updated information, based on belief propagation according to the probabilities of the Log-Likelihood Ratio (LLRs) [11,21]. Next, the CN units compute the new soft information based on the following equation:

$$f(x) = \log \tanh(|x|/2) \tag{3}$$

The VN units update the LLRs coming from the corresponding CNs as follows:

$$VN(Y_i) = Y_{i,1} + Y_{i,2} + \ldots + Y_{i,n} \tag{4}$$

Then, each VN sends the computed messages again to the corresponding CNs for the next iteration.

The min-sum algorithm (MSA) simplifies the complexity of the sum-product [22,23] by approximating the SPA function (Equation (3)) within the CN process into the minimum function:

$$CN(Y_i) = \Pi(\mathrm{sgn}(Y_i) \cdot \min(|Y_i|)) \tag{5}$$

There have been numerous updates made to the MS algorithm, including Offset MS (OMS) and Adapted MS (AMS). The enhanced MS algorithms diminish the overestimation of message magnitudes [11,12].

The LDPC codes in conformity with 5G standards recognize a significant breakthrough relating to decoding performances and the length of transmitted data [24,25]. They constitute a family of QC-LDPC codes, which can be described by the use of protograph codes [26,27].

A protograph code is defined by its matrix Hp = Mp × Np and its expansion factor Z [28–30]. The values of the lifting size (Z) fall within an integer range ($2 \leq Z \leq 384$). The entries (Hp(i,j)) of the matrix Hp would be from -1 to $Z - 1$. Each entry of the matrix is expanded as zeros Z × Z binary submatrix for (Hp(i,j)) = -1, and the identity Z × Z matrix is shifted by (Hp(i,j)) for positive values [31,32].

Following the expansion of Hp, the LDPC matrix becomes H = M (Mp × Z) × N (Np × Z) [33,34]. The 5G standards indicate two base graphs:

BG1 = 46 × 68, used in scenarios requiring high throughput [24,35].

BG2 = 42 × 52, which is used for applications with lower throughput [12,35].

The structures of the two base matrices are presented (see Figure 2).

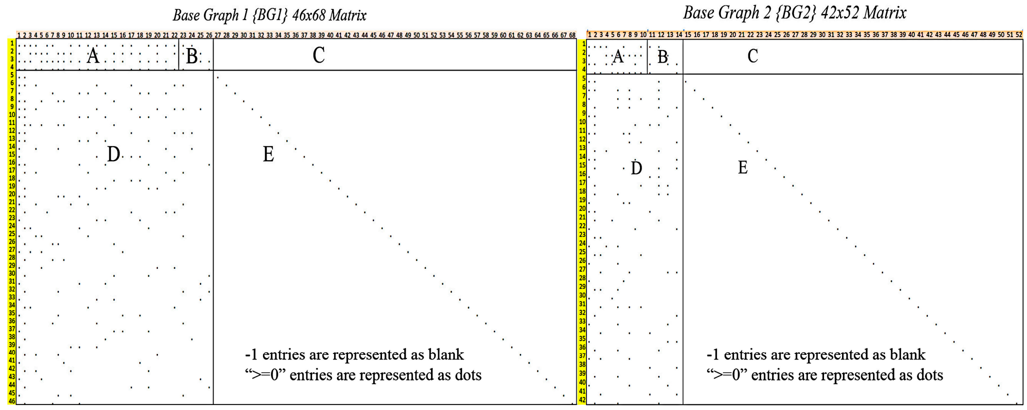
Base Graph 1 {BG1} 46x68 Matrix
A
B
C
D
E
-1 entries are represented as blank
">=0" entries are represented as dots
Base Graph 2 {BG2} 42x52 Matrix
A
B
C
D
E
-1 entries are represented as blank
">=0" entries are represented as dots

Figure 2. Structure of 5G NR base graphs; -1 entries are represented as blank, and "Hp(i,j) $\geq$ 0" are represented as dots.

For both base graphs, the block "A" corresponds to the information (message) bits, while "B" has the double diagonal structure that assigns the encoding process, and "C" is an all-zeros matrix. The extension matrices "D" (message bits) and "E" (parity bits) support incremental redundancy requests [36,37].

Figure 3 provides a comparative analysis of BG1 and BG2, clarifying their operational characteristics within communication systems. By contrasting these base graphs, key performance indicators such as achievable encoding rates (R) and supported message bit sizes (K) can be determined, aiding in the selection of optimal parameters for specific coding applications.

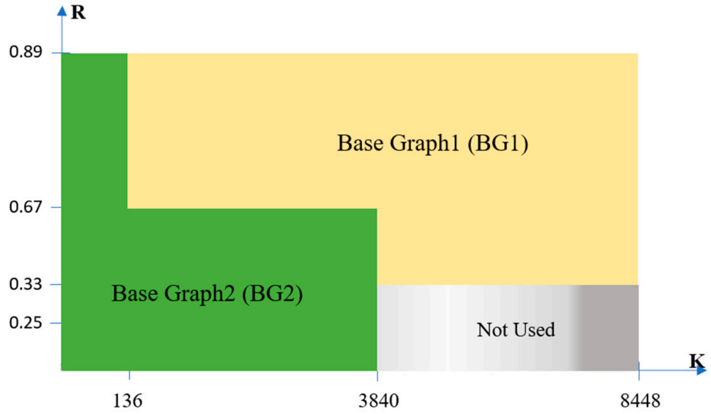

Figure 3. Structure of both base graphs according to the rate (R) and the number of message bits (K).

4. Proposed 5G-NR Decoder Architecture

Within this section, we introduce our innovative decoder micro-architecture that redefines the landscape of the IMT-2020 (5G-NR) wireless standards. Our design not only offers unparalleled flexibility, reduced latency, and low hardware requirements, but also achieves an unprecedented level of throughput, surpassing all prior endeavors in the field. The proposed design exhibits unparalleled versatility, seamlessly accommodating a wide spectrum of 5G applications ranging from immersing VR (virtual reality) experiences and cutting-edge AR (augmented reality) technologies to the integration of smart homes and cities, as well as the automation needs of industrial sectors.

4.1. Comprehensive Decoder Architecture

The decoder is based on the first base graph (BG1). As previously mentioned, the entries of the base matrix vary within the range $-1 \leq$ Hp(i,j) ≤ 383 [38–40].

The first 22 columns of BG1 entries correspond to the message bits and the following 46 correspond to parity bits. The rate matching determines how many parity bits should be added to the message bits to accomplish the decoding process by applications and their needed performance. Using the maximum expansion factor, and after the expansion of the matrix, this architecture affords the maximum information length in 5G-NR 8448-bits (22 × Z) [41,42]. A simplified schematic of the architecture is presented (see Figure 4). To provide the decoder with the channel's received LLRs during initialization in the first iteration, the control unit's In-Adr output sends a binary "1" to the "2:1Mux" (see Figure 4). After the first iteration, In-Adr provides a binary "0" to the "2:1Mux" for the subsequent decoding iterations of the processed frame, until the correction is carried out or a maximum number of iterations is achieved. The same process is repeated for the next frames coming from the channel. The second output of the control unit is the Lyr-Ctrl, which indicates the current processing layer for all the blocks.

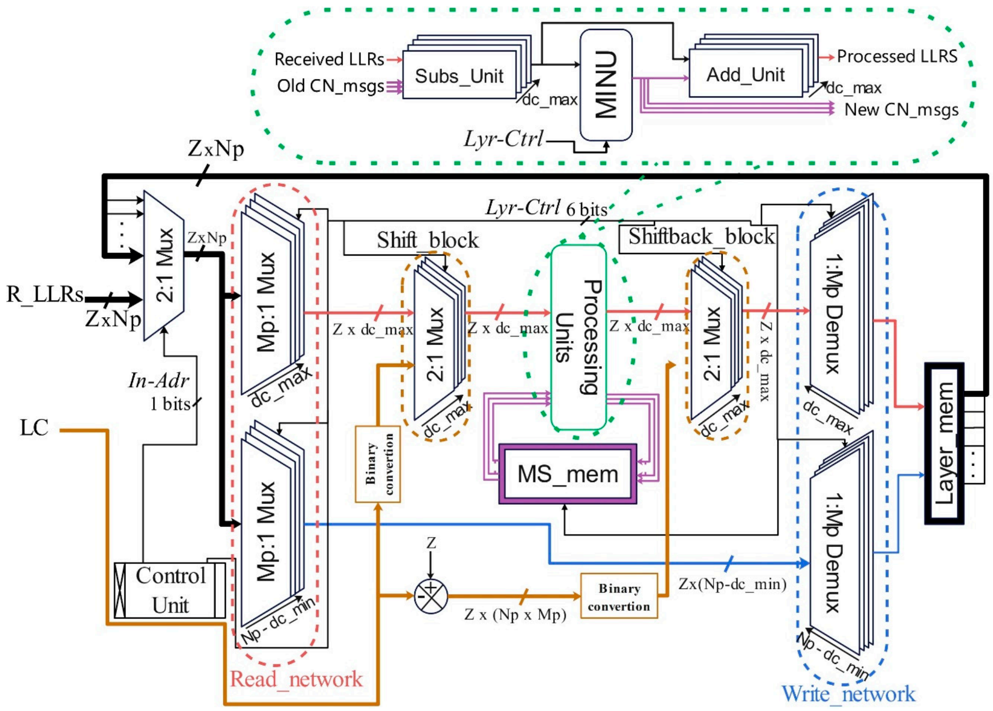

Figure 4. Global architecture of the proposed decoder. Data flow is color-coded for clarity: bold black for the main data path, blue for raw data, pink for processed data, orange for layer controls (shifting data), and violet for the processed data memory.

After being supplied into the decoder, the received LLRs are separated using a read-network into treated and raw data corresponding to non −1s and −1s entries of the base graph, respectively, based on the current processes row (layer). The row weight dc deviates between dcmin = 3 and dcmax = 19; therefore, the treated data are passed using 19 “Mp:1Mux” during each layer process. The raw data are simultaneously sent by several (Np-3) “Mp:1Mux” directly to the write-network so that they can be concatenated with the processed data after the layer decoding is complete.

Following the read network, the (19 × Z) LLRs are handled by the dcmax shifting blocks. The construction of the shifting blocks is based on a new particular method that involves assembling a (‘Z’ by ‘b’) matrix of “2:1Mux” with $Z \leq 2b$. The shifting blocks obtain the base graph entries through the input LC after they are converted to binary addresses using the binary conversion block. The matrix [Z × b] “2:1Mux” accomplishes the shifting of every 384 LLR sequence by addressing the “2:1Mux” with the converted LC entries. Similar steps are taken for the shift-back blocks, while the multiplexers contained therein are managed by the minus circuit that subtracts the base graph’s elements from the expansion factor (Z−Hp(i,j)).

Before the shift-back blocks and immediately after the shifting blocks, the data proceed along their path to the processing units. The recorded check node output values of the same layer from the previous iterations are received from the MS-mem and subtracted from the current data values (received LLRs) via a tree of dcmax subs-units, each of which has Z subtractors. It must be noted that the data do not change throughout the subs-unit blocks during the first iteration since there is no data stored in memory. The internal architecture of the MINU block is discussed in the following subsection where the subs-unit blocks outputs take place to find the two minimums. According to MSA, the block yields the first min instead of all the other values and then replaces the first min value with the second min. Each block of the dcmax add-unit accommodates Z adders, so it calculates the summation of MINU and subs-unit outputs to determine the processed LLRs. The write-network collects both the processed and raw data along with two different demultiplexer trees

(1:Mp Demux), based on the locations of non-1 and −1 entries in the base graph, to store in memory for future decoding layers.

The decoder architecture requires only two principal memories. First, the MS-mem block stores the MINU outputs at the end of each decoding layer to be used for the subunit subtraction. It consists of several Mp memory subblocks. Every subblock corresponds to a layer in the base matrix. Therefore, each subblock is designed according to the weight of the convenient row. In the base graph, the first four layers' weights are dc = 19, so they require 19 sub-memories each, while the fifth layer weighs 3, so it requires 3 sub-memories. Furthermore, each sub-memory contains Z memory locations. Then, the Lay-Mem block comprises Np memory banks of Z units to store the updated data after each layer process.

4.2. MINU Processing Unit

The MINU processing unit receives Z sequences of dcmax values. Thus, it covers Z check node (CN) units, where each one processes 19 values to find the two minimums.

The CN unit structure is presented (see Figure 5). Each of the 384 CNs within the MINU is constructed from the SOLL unit and an output selector that organizes the two minimums in their positions according to the index of Min1. Every CN unit input is regulated by a "2:1Mux", which supplies a maximum stored value to complete its 19 inputs for the layers of a weight smaller than dcmax.

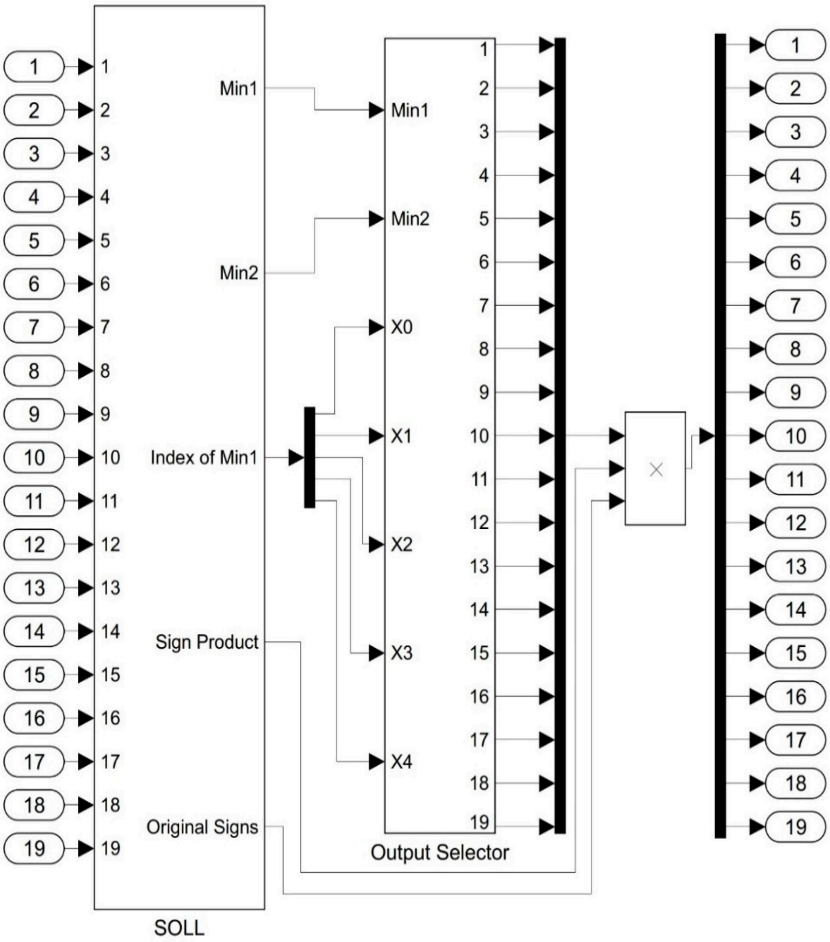

Figure 5. Check node unit structure and its modeling on Simulink.

The SOLL unit performs parallel computations of the first and second minimums (Min1 and Min2), as well as the index of the first minimum, while also indicating the original sign of each LLR and its product. Moreover, the process involves only (2 × dcmax-3) comparison blocks. With BG1 and the maximum codeword length, the SOLL approach requires only 35 comparison blocks for every CN unit. This method involves conducting two types of comparisons. Initially, the MMB evaluates and sorts two inputs, assigning the lesser and greater values to the corresponding 'S' and 'B' outputs, respectively. Subsequently, MB determines the minimum value among the 'B' outputs from the MMB blocks, with the MB having a single output 'S' which represents the second minimum output of the SOLL unit after the comparison is complete.

After the MMB blocks have completed their comparison operation, the SOLL unit assigns the smallest value between the 'S' outputs of MMB as the first minimum output. Additionally, each MMB block indicates the position of the smaller values, with the 'L' output being assigned binary '0' if the left input is smaller and '1' if the right input is smaller. A tree of "2:1Mux" is then used to generate an address, with the 'L' outputs of the MMB blocks being used to address the tree and identify the min1-indexed position (see Figure 6).

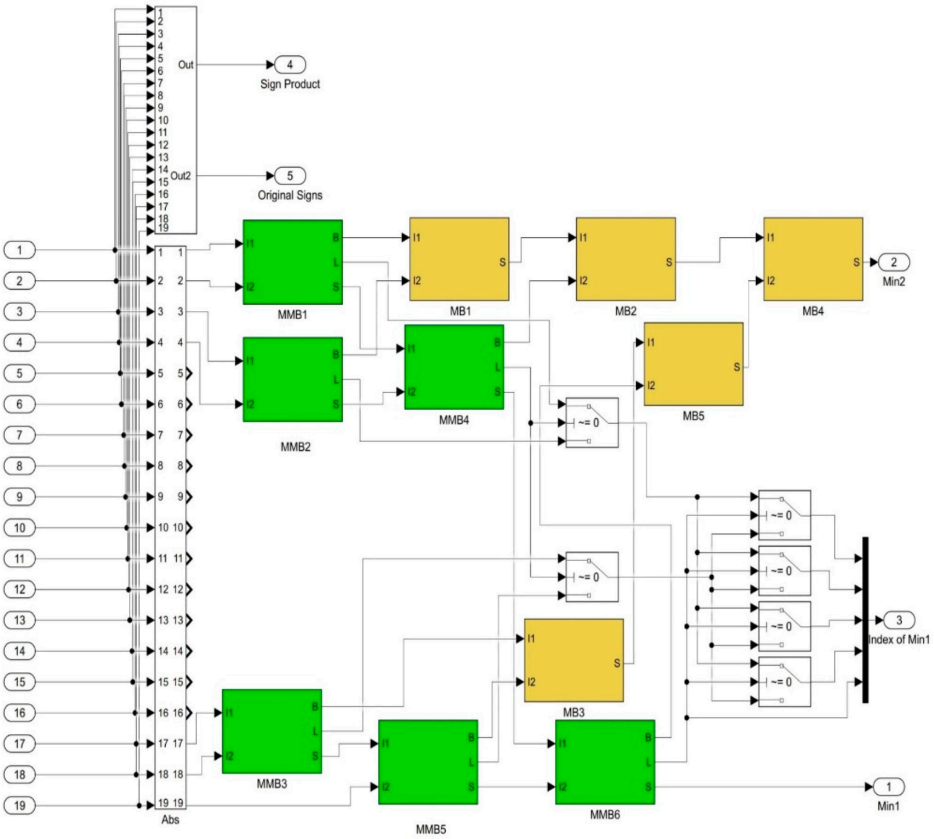

Figure 6. Simulink redesign of the SOLL unit for 5G-NR usage.

5. Results

The simulation results unveil a remarkable enhancement in the decoder's ability to tackle 5G-NR applications, surpassing the performance of all previous studies [11–14,39,43] in multiple dimensions. Not only do our results showcase a superior performance in terms of the number of iterations and BER (bit error rate) analysis, but they also outshine earlier studies regarding handling extended codeword lengths.

Our decoder's baseline architecture (Z = 8, R = 2/3) was modeled in Simulink, converted to VHDL, and then updated to the highest codeword length in 5G-NR (Z = 384, R = 1/3) with the intent of FPGA implementation. Table 1 presents the simulation outcomes of the enhanced architecture (Z = 384) in comparison to prior works, showcasing two codeword rates of 2/3 and 1/3. The rate 1/3 (model 1) exhibits the best performance compared to 2/3 (model 2) because it offers all BG1 entries and uses the maximum number of parity bits. However, the rate of 2/3 provides fewer parity bits, reducing hardware complexity for certain applications such as live streaming.

Table 1. Comparison of the proposed 5G-NR LDPC decoder with the state-of-the-art.

	This Study							
	Model 1	**Model 2**	**[11]**	**[12]**	**[13]**	**[14]**	**[39]**	**[43]**
Expansion factor	384	384	384	104	32	384	384	-
Rate	1/3	2/3	1/3	2/3	4/5	1/2	1/3	1/2
SNR (dB)	1.7 dB	2.9 dB	-	-	4 dB	-	2.4 dB	-
Codeword length	26,112	13,440	26,112	3432	1280	16,128	26,112	12
Information length	8448	8448	8448	2288	-	8448	8448	-
Number of iterations	4	6	10	15	20	20	15	10
Decoding algorithm	MS	MS	OMS	AMS	SMS	MS	MS	MS
Matrix size	46 × 68	13 × 35	46 × 68	13 × 35		22 × 44		-
Decoding parallelism	Fully	Fully	Fully	Partially		Partially		Partially
Standard	5G-NR	5G-NR	5G-NR	5G-NR		5G-NR		4G-LTE

6. Discussion

It should also be noted that the maximum number of iterations is improved over previous designs in 5G-NR and 4G-LTE. Model 1 can be compared to the design in [11] because it has the same decoder parameters. In particular, our design significantly reduces latency with six iterations of improvements and up to a 0.2 dB (for BER = 2.10^{-3}) SNR increase (see Figure 7a). The rate 2/3 was not simulated for the maximum expansion factor in previous studies, and therefore we are providing an estimated comparison with references [12,14]. In [12], a rate of 2/3 was used with an expansion factor of Z = 104, while in [14], they used a rate of 1/2 which was expected to perform better than 2/3. Despite this, our design achieved an SNR performance gain of up to 0.2 dB compared to [14] and 0.6 dB compared to [12] for BER = 2.10^{-2} (see Figure 7b). Furthermore, we have achieved a remarkable reduction in the number of iterations, bringing it down to 6, as opposed to 15 and 20 iterations found in references [12,14], respectively (see Table 1). It is noteworthy that our study has simulated the maximum attainable codeword length for a rate of 2/3.

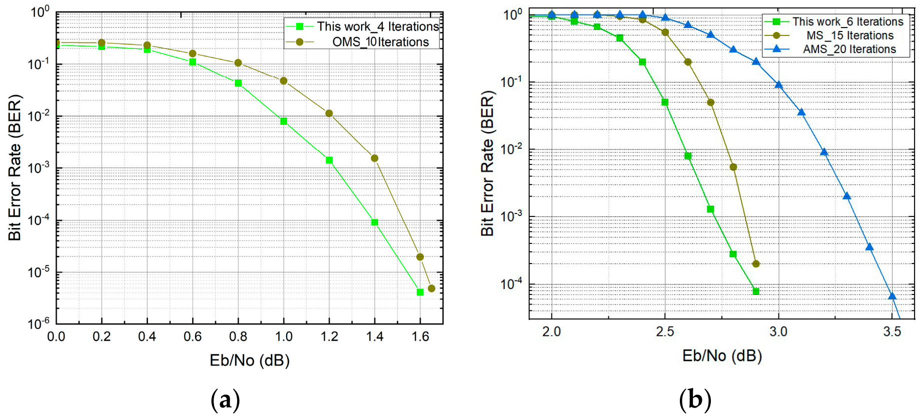

Figure 7. Comparison of the proposed decoder's SNR performance with other designs, (**a**) [11] based on the rate 1/3 and (**b**) (AMS [12], MS [14]) based on the rate 2/3.

The previously discussed decoder simulation results were obtained using the MS algorithm. The AMS and OMS algorithms are known to outperform the MS algorithm because they tend to overestimate LLR values. However, this issue was not a significant challenge in our work, and due to the small number of iterations, the correction process was achieved before the overestimation became a concern.

7. Conclusions

This paper presents an innovative architecture for a layered QC-LDPC decoder that conforms to the 5G-NR standards. Our proposed design enables the processing of the maximum information length in 5G-NR. The decoder features a low latency, flexible model design, and reduced complexity due to the read/write networks, check node units, and shifting block designs. This research has also demonstrated the efficiency of the modified SOLL method for 5G-NR, highlighting its convenience and demonstrating positive outcomes. This shows significant improvements in SNR analysis and the number of iterations when compared to previous works. Moreover, the design supports a wide range of expansion factors and base matrices and can be easily modified to accommodate different code rates. As a next step, we plan to implement the decoder on FPGA; the synthesis process has already commenced. The promising results thus far reinforce our design's potential for practical IMT-2020 applications.

Author Contributions: I.A., methodology, formal analysis, validation, and supervision; M.A.M., visualization, resources, and formal analysis; A.A., methodology, formal analysis, validation, supervision, and data curation; B.M., software, conceptualization, formal analysis, data curation, and writing—original draft preparation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study were obtained at the Engineer Sciences Laboratory from the Polydisciplinary Faculty of Taza, University Sidi Mohamed Ben Abdellah (USMBA), Fez. They are available upon reasonable request. Data sharing is subject to ethical and privacy considerations. Do not hesitate to contact Bilal MEJMAA at bilal.mejmaa@usmba.ac.ma for inquiries regarding data access and availability.

Acknowledgments: The author would like to express his deepest gratitude to his research supervisors Ismail Akharraz and Abdelaziz Ahaitouf, for their invaluable guidance, insightful critiques, and unwavering support throughout this research. My sincere thanks also go to Malika Alami Marktani for her expert advice and assistance in the hardware description language phase of my work. A special appreciation is owed to the Laboratory of Engineer Sciences for providing the necessary facilities and resources.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aqil, C.; Akharraz, I.; Ahaitouf, A. A New Reliability Ratio Weighted Bit Flipping Algorithm for Decoding LDPC Codes. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 6698602. [CrossRef]
2. Singh, A.K. Error detection and correction by hamming code. In Proceedings of the International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, India, 22–24 December 2016; IEEE: Piscataway, NJ, USA, 2016.
3. Ly, T.; Truong, N. Efficient Hardware Implementations of LDPC Decoders, Through Exploiting Impreciseness in Message-Passing Decoding Algorithms. Ph.D. Thesis, Université de Cergy Pontoise, Cergy, France, 2017.
4. Li, H.; Guo, J.; Guo, C.; Wang, D. A low-complexity min-sum decoding algorithm for LDPC codes. In Proceedings of the 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; IEEE: Piscataway, NJ, USA, 2017.
5. Andrew, T. Online-course Ldpc and Polar Codes in 5G Standard. 2018. Available online: https://nptel.ac.in/courses/117/106/108106137/ (accessed on 1 June 2019).
6. Aqil, C.; Akharraz, I.; Ahaitouf, A. Reliability Ratio Weighted Bit Flipping–Sum Product Algorithm for Regular LDPC Codes. In *E3S Web of Conferences*; EDP Sciences: Les Ulis, France, 2022; p. 01084.
7. Liu, Y.; Olmos, P.M.; Mitchell, D.G.M. On generalize LDPC codes for 5G ultra reliable communication. In Proceedings of the Information Theory Workshop (ITW), Guangzhou, China, 25–29 November 2018; IEEE: Piscataway, NJ, USA, 2018.
8. Gallager, R. Low-density parity-check codes. *IRE Trans. Inf. Theory* **1962**, *8*, 1962. [CrossRef]
9. Ángel, Á.; Fernández, V.; Matuz, B. An efficient NB-LDPC Decoder Architecture for Space Telecommand Links. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *68*, 1213–1217.
10. Tran-Thi, B.N.; Nguyen-Ly, T.T.; Hoang, T. An FPGA design with high memory efficiency and decoding performance for 5G LDPC decoder. *Electronics* **2023**, *12*, 3667. [CrossRef]

11. Verma, A.; Shrestha, R. A New VLSI Architecture of Next-Generation QC-LDPC Decoder for 5G New-Radio Wireless-Communication Standard. In Proceedings of the International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; IEEE: Piscataway, NJ, USA, 2020.
12. Cui, H.; Ghaffari, F.; Le, K.; Declercq, D.; Lin, J.; Wang, Z. Design of High-Performance and Area-Efficient Decoder for 5G LDPC codes. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *68*, 879–891. [CrossRef]
13. Nadal, J.; Baghdadi, A. Parallel and flexible 5G LDPC decoder architecture targeting FPGA. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2021**, *29*, 1141–1151. [CrossRef]
14. Petrović, V.L.; El Mezeni, D.M. Reduced-Complexity Offset Min-Sum Based Layered Decoding for 5G LDPC Codes. In Proceedings of the 28th Telecommunications Forum (TELFOR), Belgrade, Serbia, 24–25 November 2020; IEEE: Piscataway, NJ, USA, 2020.
15. Verma, A.; Shrestha, R. Low computational-complexity SOMS-algorithm and high-throughput decoder architecture for QC-LDPC codes. *IEEE Trans. Veh. Technol.* **2022**, *72*, 66–80. [CrossRef]
16. Li, Y.; Li, Y.; Ye, N.; Chen, T.; Wang, Z.; Zhang, J. High Throughput Priority-Based Layered QC-LDPC Decoder with Double Update Queues for Mitigating Pipeline Conflicts. *Sensors* **2022**, *22*, 3508. [CrossRef] [PubMed]
17. Selvakumari, R.S. Performance analysis of Min-Sum based LDPC decoder architecture for 5G new radio standards. *Mater. Today Proc.* **2022**, *62*, 4965–4972.
18. Liu, J.C.; Wang, H.C.; Shen, C.A.; Lee, J.W. Low-complexity LDPC decoder for 5G URLLC. In Proceedings of the Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Chengdu, China, 26–30 October 2018; IEEE: Piscataway, NJ, USA, 2018.
19. Benhayoun, M.; Razi, M.; Mansouri, A.; Ahaitouf, A. Embedded Parallel Implementation of LDPC Decoder for Ultra-Reliable Low-Latency Communications. *Appl. Comput. Intell. Soft Comput.* **2023**, *2023*, 5573438. [CrossRef]
20. Wang, B.; Zhu, Y.; Kang, J. Two effective scheduling schemes for layered belief propagation of 5G LDPC codes. *IEEE Commun. Lett.* **2020**, *24*, 1683–1686. [CrossRef]
21. Mosallaei, B. Enhancing Channel Decoding Efficiency in 5G Networks Using Machine Learning-Assisted LDPC Coding. *Int. J. Eng. Appl. Sci.* **2024**, *11*, 25.
22. Tera, S.P.; Alantattil, R.; Paily, R. A Flexible FPGA-Based Stochastic Decoder for 5G LDPC Codes. *Electronics* **2023**, *12*, 4986. [CrossRef]
23. Stark, M.; Wang, L.; Bauch, G.; Wesel, R.D. Decoding rate-compatible 5G-LDPC codes with coarse quantization using the information bottleneck method. *IEEE Open J. Commun. Soc.* **2020**, *1*, 646–660. [CrossRef]
24. Aqil, C.; El Alami, R.; Akharraz, I.; Ahaitouf, A. Threshold Multi Split-Row algorithm for decoding irregular LDPC codes. In Proceedings of the International Conference on Applied Mathematics, Taza, Morocco, 19–20 October 2017; pp. 88–93.
25. Usman, S.; Mansour, M.M. An Optimized VLSI Implementation of an IEEE 802.11 n/ac/ax LDPC Decoder. In Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS), Seville, Spain, 12–14 October 2020; IEEE: Piscataway, NJ, USA, 2020.
26. Zhong, Z.; Huang, Y.; Zhang, Z.; You, X.; Zhang, C. A flexible and high parallel permutation network for 5G LDPC decoders. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 3018–3022. [CrossRef]
27. Benhayoun, M.; Razi, M.; Mansouri, A.; Ahaitouf, A. Low-Complexity LDPC Decoding Algorithm Based on Layered Vicinal Variable Node Scheduling. *Model. Simul. Eng.* **2022**, *2022*, 1407788. [CrossRef]
28. Wu, X.; Jiang, M.; Zhao, C.; Ma, L.; Wei, Y. Low-rate PBRL-LDPC codes for URLLC in 5G. *IEEE Wirel. Commun. Lett.* **2018**, *7*, 800–803. [CrossRef]
29. Bae, J.H.; Abotabl, A.; Lin, H.P.; Song, K.B.; Lee, J. An overview of channel coding for 5G NR cellular communications. *APSIPA Trans. Signal Inf. Process.* **2019**, *8*, e17. [CrossRef]
30. Razi, M.; Benhayoun, M.; Mansouri, A.; Ahaitouf, A. An improvement and a fast DSP implementation of the bit flipping algorithms for low density parity check decoder. *Int. J. Electr. Comput. Eng.* **2021**, *11*, 4774. [CrossRef]
31. Tarver, C.; Tonnemacher, M.; Chen, H.; Zhang, J.; Cavallaro, J.R. GPU-based, LDPC decoding for 5G and beyond. *IEEE Open J. Circuits Syst.* **2021**, *2*, 278–290. [CrossRef]
32. Sun, K.; Jiang, M. A hybrid decoding algorithm for low-rate LDPC codes in 5G. In Proceedings of the 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China, 18–20 October 2018; IEEE: Piscataway, NJ, USA, 2018.
33. Li, F.; Zhang, C.; Peng, K.; Krylov, A.E.; Katyushnyj, A.A.; Rashich, A.V.; Tkachenko, D.A.; Makarov, S.B.; Song, J. Review on 5G NR LDPC Code: Recommendations for DTTB System. *IEEE Access* **2021**, *9*, 155413–155424. [CrossRef]
34. Venkatesh, D.Y.; Mallikarjunaiah, K.; Srikantaswamy, M.; Huang, K. Enhancing 5G LTE communications: A novel LDPC decoder for next-generation systems. *Inf. Dyn. Appl.* **2024**, *3*, 47–63. [CrossRef]
35. Li, H.; Xu, H.; Chen, C.; Bai, B. Efficient construction of quasi-cyclic LDPC codes with multiple lifting sizes. In *IEEE Communications Letters*; IEEE: Piscataway, NJ, USA, 2024.
36. Richardson, T.; Kudekar, S. Design of low-density parity check codes for 5G new radio. *IEEE Commun. Mag.* **2018**, *56*, 28–34. [CrossRef]
37. Nguyen, T.T.B.; Nguyen Tan, T.; Lee, H. Efficient QC-LDPC encoder for 5G new radio. *Electronics* **2019**, *8*, 668. [CrossRef]
38. 5G NR Multiplexing and Channel Coding (3GPP TS 38.212 Version 15.2.0 Release 15). ETSI. 2018. Available online: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.02.00_60/ts_138212v150200p.pdf (accessed on 1 June 2020).

39. Ren, Y.; Harb, H.; Shen, Y.; Balatsoukas-Stimming, A.; Burg, A. A Generalized Adjusted Min-Sum Decoder for 5G LDPC Codes: Algorithm and Implementation. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2024**, *71*, 2911–2924. [CrossRef]
40. Hamidi-Sepehr, F.; Nimbalker, A.; Ermolaev, G. Analysis of 5G LDPC Codes Rate-Matching Design. In Proceedings of the 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; IEEE: Piscataway, NJ, USA, 2018.
41. Wu, H.; Wang, H. A high throughput implementation of QC-LDPC codes for 5G NR. *IEEE Access* **2019**, *7*, 185373–185384. [CrossRef]
42. Wu, X.; Jiang, M.; Zhao, C. Decoding optimization for 5G LDPC codes by machine learning. *IEEE Access* **2018**, *6*, 50179–50186. [CrossRef]
43. Yang, N.; Jing, S.; Yu, A.; Liang, X.; Zhang, Z.; You, X.; Zhang, C. Reconfigurable decoder for LDPC and polar codes. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; IEEE: Piscataway, NJ, USA, 2018.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.