https://doi.org/10.7840/kics.2024.49.12.1792

Design and Implementation of a High-Reliability Min-Max Algorithm for LDPC Decoder on FPGA

Heung-Ryol Yoo◆, Yung-Deug Son°

ABSTRACT

LDPC codes improve error correction performance in soft-decision decoding by using the reliability of the received signal, offering greater accuracy than hard-decision decoding. The High Reliability Min-Max Algorithm leverages this advantage by using LLR to reduce computational complexity while improving decoding accuracy. The (837, 726) Min-Max code maintains performance similar to LDPC while being more efficient, reducing hardware resource usage and power consumption. Compared to the (4191, 3602) binary LDPC code, it offers better transmission efficiency. This paper presents the simulation results for a (512, 256) code, and the hardware implementation, using pipeline and parallel structures, highlights the high error correction capability of the proposed Min-Max code. This structure combines the error correction strengths of QC-LDPC with the simplicity of binary LDPC's hardware implementation. The proposed algorithm demonstrates similar error correction performance to LDPC codes but achieves lower complexity and higher transmission efficiency.

Future research will focus on comparing the performance through ASIC implementation to further assess its effectiveness in practical systems. This comparison will offer insights into how the proposed Min-Max code can be implemented for real-world applications requiring efficient and reliable error correction performance.

Key Words : Digital Communication, Error Correction Coding, FPGA, LDPC, Min-max Algorithm

Ⅰ. Introduction

Due to the remarkable advancements in communication technology and computer performance, Low-Density Parity-Check (LDPC) codes have been adopted as standards in various digital communication systems, such as storage systems[1-3] and IEEE 802.11n, 802.3m, 802.16a[4-6]. Recently, in situations where some error correction capabilities of the code may be lost or when the channel is modeled as a Binary Symmetric Channel (BSC), hard-decision decoders are being prioritized over soft-decision decoders. These algorithms use only binary symbols received from the channel during the initialization phase. Based on these symbols, methods such as voting or flipping are employed[7].

Additionally, Sum-Product algorithms and Min-sum algorithms use the log-likelihood ratio (LLR) of the received symbols during the initialization phase. As a result, research continues on balancing decoder performance with computational complexity.

When Robert Gallager first introduced parity-check codes in 1962[8], they did not attract much attention because the code lengths were too long, and the decoder algorithms were too complex. However, due to their simple decoding advantage, LDPC codes have been adopted as decoders in digital communication

※ This paper was supported by the Education and Research Promotion Program of KOREATECH in 2024 and The EDA tool was also supported by the IC Design Education Center(IDEC), Korea.
◆ First Author : Korea University of Technology and Education Department of Electrical Engineering , hryoo@kopo.ac.kr, 정회원
° Corresponding Author : Korea University of Technology and Education Department of Mechanical Facility Control Engineering, ydson@koreatech.ac.kr, 정회원
논문번호 : 202409-203-B-RU, Received September 7, 2024; Revised September 12, 2024; Accepted September 23, 2024

standards, such as 5G communication systems. In the Min-max algorithm[9], the complexity of the algorithm is further reduced at the cost of slightly lower coding gain. In this algorithm, 'maximum' is performed instead of 'sum.' In [10] proposed an architecture for the Min-max algorithm that retains all messages, which can result in large memory requirements if the order of the finite field is not small. Additionally, this architecture incurs very long delay times because it calculates all possible combinations of field elements in each message vector. A parallel Min-max algorithm architecture was also developed in [11]. Although this architecture achieves high speed, it requires a large area due to the high level of parallelism. Moreover, its efficiency decreases when the order of the finite field is large or the code rate is low. In [12, 13] is proposed two decoding algorithms and architectures for the Min-max algorithm, both of which significantly reduce decoding complexity.

Recent research trends focusing on LDPC codes are actively progressing in various fields. In particular, various studies are being conducted to improve the performance and practical applications of Quasi-Cyclic LDPC(QC-LDPC)[14], binary LDPC, and Non-Binary LDPC (NB-LDPC) codes. This study aims to optimize the Min-max algorithm, one of the NB-LDPC codes, and verify its advantage of high error correction capability. Hard-decision decoding has lower complexity than soft-decision decoding, but generally offers reduced error correction performance. However, the proposed High Reliability Min-max Algorithm aims to improve decoding accuracy and efficiency by systematically reducing processing complexity using the LLR of soft-decision decoding. To reduce the high computational complexity of NB-LDPC, we simplified the structure and overcame the difficulties in hardware implementation through an intuitive architectural design. Section 2 presents the software design and simulation of this study. Section 3 describes the hardware design and implementation. Section 4 contains the conclusion.

Ⅱ. Software Design & Simulation

2.1 Min-max Algorithm

To explain the Min-max algorithm, the message vector coming from variable nodes to a check node can be represented as a tree structure composed of multiple stages. Each stage represents an element of the variable-to-check message vector, and the nodes of each stage are arranged in the order of finite field elements 0, 1, 2, ..., q -1. Fig. 1 shows the trellis structure for $v_d = 6$ and q = 4. The variable-to-check message vector for a single check node $u_{m,n}(v_{m,n})$ is simply denoted as $u_j(v_j)(0 \leq j < d_c)$. Using this trellis representation of the variable-to-check messages, when calculating the check-to-variable message at v_j, a path that passes through one node at each stage is found. However, the j-th stage is excluded. The LLR of the path is the maximum value of the LLRs of the nodes in the path, and the finite field element is the sum of the finite field elements of the nodes. Fig. 2 shows several paths for calculating v_{dc-1} using the method proposed in this paper, showing the LLR and finite field elements of each path. The optimal path for $u_j(\alpha)$ is the path with the minimum LLR among the paths where the finite field element is α, and $u_j(\alpha)$

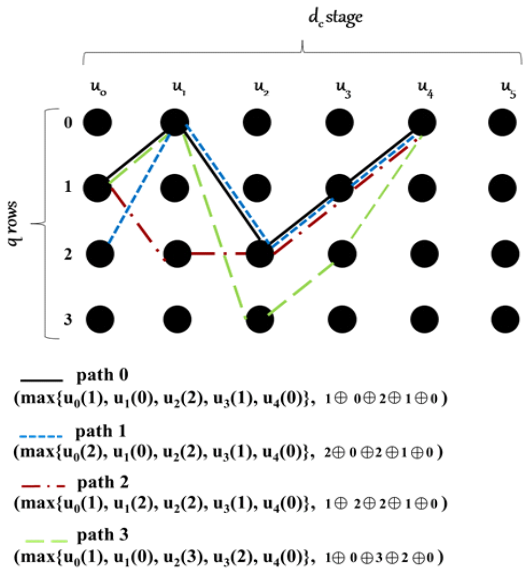

Fig. 1. Message trellis from variable to check: The original trellis structures

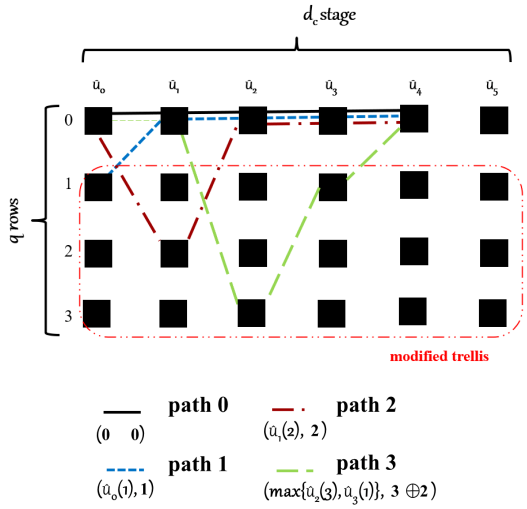

Fig. 2. Message trellis from variable to check: The rearranged trellis structures

is equal to the LLR value of this optimal path.

Additionally, to simplify the algorithm, the variable-to-check message vector rearrangement technique proposed in [15] was adopted. Assuming that the element representing the most reliable message is $\hat{\alpha}_j$, $u_j$is rearranged to $\hat{\alpha}_j$.

2.2 High Reliability Min-max Algorithm

The proposed High Reliability Min-max Algorithm aims to enhance decoding accuracy and efficiency by systematically reducing the complexity of processing. The process involves several key steps:

The initialization phase begins by calculating the sum of alpha values across all codeword positions, denoted as α_{sum}, by summing the alpha values for each position j from 0 to d_c - 1.

In the first stage, initial values are computed for each position j within the codeword. For any given $0 \leq j < d_c$, the initial values $\tilde{u}_j(\beta)$ are determined by checking if β is equal to the bitwise combination of the estimated alpha value at position j and another alpha value.

The second one involves identifying the minimum and second minimum LLRs for each possible value of β. The minimum LLRs, M_{first_min}, are the smallest LLR values along with their corresponding positions, and the second minimum LLRs, M_{second_min}, are the second smallest LLR values.

In the third, vectors $m_j(\beta)$ associated with each codeword position j are constructed. For each j, the vectors consist of LLR values $m_j(\beta)$ for each β from 1 to q. The LLR value $m_j(\beta)$ is taken from $m1(\beta)$ if j is not equal to $I(\beta)$, and from $m2(\beta)$ if j is equal to $I(\beta)$.

The fifth is the construction of the minimum basis B_j for each codeword position j. The minimum basis is derived using the sorted vector $\overline{M}_j$ and consists of the smallest p elements from $\overline{M}_j$, ensuring that each element is independent of the others.

In the sixth, the algorithm calculates the values $\hat{v}_j$ for each position j. The initial value $\hat{v}_j$ is set to 0, and for each β, the value $\hat{v}_j(\beta)$ is computed as the maximum of the LLR values of the elements in the basis Bj, combined with their corresponding field elements.

Finally, in the seventh step, the final values v_j are derived for each position j. This is done by aligning the computed values $\hat{v}_j(\beta)$ with the initial alpha values. If α is equal to β combined with α_{sum} and the estimated alpha value at position j, then $v_j(\alpha)$ is set to $\hat{v}_j(\beta)$.

This structured approach effectively reduces computational complexity while maintaining high reliability in the Min-max algorithm's decoding process. By leveraging organized LLR processing and careful management of message vectors, the proposed algorithm offers a robust solution for improving decoding efficiency in error correction coding systems. The proposed algorithm is defined in Fig. 3.

In conclusion, unlike the existing Min-max algorithm, which transmits messages based on the maximum LLR (Log-Likelihood Ratio) value for each path, the High Reliability Min-max algorithm considers the second minimum LLR value, allowing for a more refined path selection process.

The High Reliability algorithm systematically reduces computational complexity by utilizing a minimum basis and reorganizing the message vectors based on β values, offering a more efficient processing flow compared to the existing algorithm. Additionally, the High Reliability approach improves decoding accuracy by calculating the minimum basis

$$Iniatlization: \alpha_{(sum)} = \sum_{0 \le j < d_c} \hat{\alpha}_j$$

$$F1: \hat{u}_j(\beta) = u_j(\alpha) \text{ if } \beta = \hat{\alpha}_j \oplus \alpha \ (0 \le j < d_c)$$

$$F2: compute\ M_{min1} = \{(m1(\beta), I(\beta))\}(1 \le \beta < q)$$
$$M_{min2} = \{m2(\beta)\}(1 \le \beta < q)$$
$$\text{for } j = 0 \text{ to } d_c - 1$$

$$F3: M_j = \{m_j(\beta)\}(1 \le \beta < q)$$
$$m_j(\beta) = \begin{cases} m1(\beta), j \ne I(\beta) \\ m2(\beta), j = I(\beta) \end{cases}$$

$$F4: Sort\ M_j \text{ to obtain } \overline{M}_j = \{(m_j^{(i)}, \alpha_j^{(i)})\}, (1 \le i < q)$$

$$F5: B_j(0) = \Psi(\overline{M}_j) = \{(m_j^{'(i)}, \alpha_j^{'(i)})\}(1 \le i < p)$$

$$F6: \hat{v}_j(0) = 0$$
$$\hat{v}_j(\beta) = max\{m_j^{'(\omega_1)}, m_j^{'(\omega_2)}, \cdots, m_j^{'(\omega_s)}\}$$
$$\text{if } (\beta) = \alpha_j^{'(\omega_1)} \oplus \alpha_j^{'(\omega_2)} \oplus \cdots \oplus \alpha_j^{'(\omega_s)} \ (1 \le s \le p)$$

$$F7: v_j(\alpha) = \hat{v}_j(\beta) \text{ if } \alpha = \beta \oplus \alpha_{sum} \oplus \hat{\alpha}_j$$

Fig. 3. High Reliability Min-max Algorithm

at each codeword position to select the optimal path, whereas the existing Min-max algorithm does not include such additional basis calculations.

In fig. 4 compares repetition codes under soft-decision and hard-decision decoding. It demonstrates that soft-decision decoding achieves a lower error probability. In comparison to the (4191, 3602) binary LDPC code, the (837, 726) Min-Max code offers notable advantages due to its shorter length. The reduced length of the (837, 726) Min-Max code implies greater transmission efficiency, as the same level of error correction performance can be achieved with fewer resources. This translates to reduced bandwidth usage and lower power consumption, which are critical factors in many communication systems. Furthermore, the shorter code length also leads to lower decoding complexity, reducing the demand on both hardware and software resources. This reduction in complexity is especially advantageous for real-time communication systems, where rapid and efficient decoding is essential. Overall, the (837, 726) Min-Max code, with its shorter length and lower complexity, provides a more efficient solution while maintaining error correction performance comparable to the longer LDPC codes. In Fig. 5, the proposed High Reliability Min-Max Algorithm, implemented using the soft-decision method described in [14], demonstrates performance similar to the (4191, 3602) binary LDPC code. This comparison highlights the superior performance of the proposed approach in terms of error correction, as it maximizes error correction capability by addressing the limitations of hard-decision decoding methods. Thus, the proposed algorithm achieves high reliability while maintaining efficient processing, further proving its effectiveness in enhancing overall decoding performance.

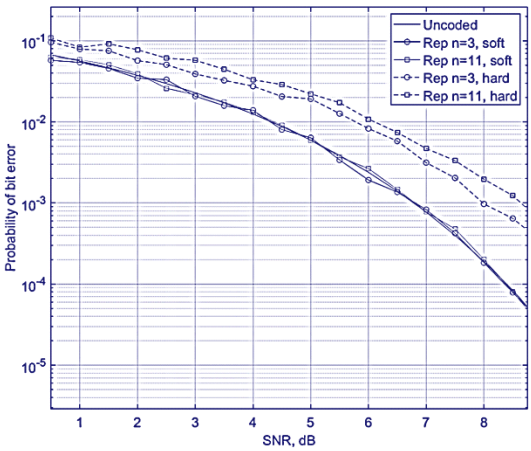

Fig. 4. Performance of repetition code using hard-and soft-decision decoding[16]

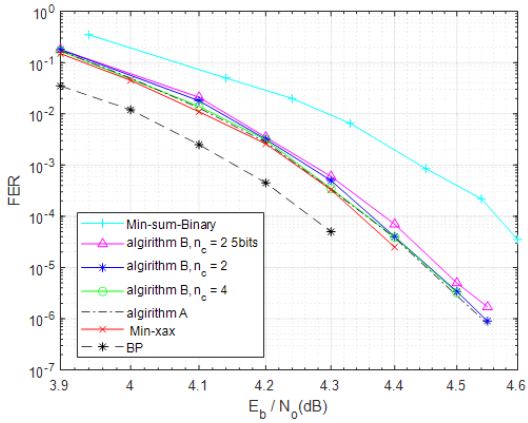

Fig. 5. FER of the (837, 726) Min-Max code and the (4191, 3602) binary LDPC code over the AWGN channel

Ⅲ. Hardware Design and Verification

The proposed design first sorts a limited number of v-to-c messages with the highest reliability, then derives the c-to-v messages from the sorting results using a path construction procedure without storing other intermediate messages. This algorithm consists of a total of 8 stages, where the output of each stage is used as the input for the next stage. We describe the input and output values for each stage and detail the operations according to the given algorithm. The ultimate goal is to recover the alpha value. In the step-by-step explanation, the initialization phase calculates α_{sum} by performing an XOR operation on $\hat{\alpha}$ and α. This value is used as an input in subsequent stages. In the first stage, $u_j(\beta)$ is obtained by performing an XOR operation on $\hat{\alpha}$ and α, and in the second stage, the minimum and second minimum values among several $u_j(\beta)$ values are selected to obtain M_{first_min} and M_{second_min}. In the third stage, $m_j(\beta)$ is calculated using M_{first_min} or M_{second_min}. In the fourth stage, $m_j(\beta)$ values are sorted to obtain data_out, and in the fifth stage, the data_out values are set as the final message and alpha_out.

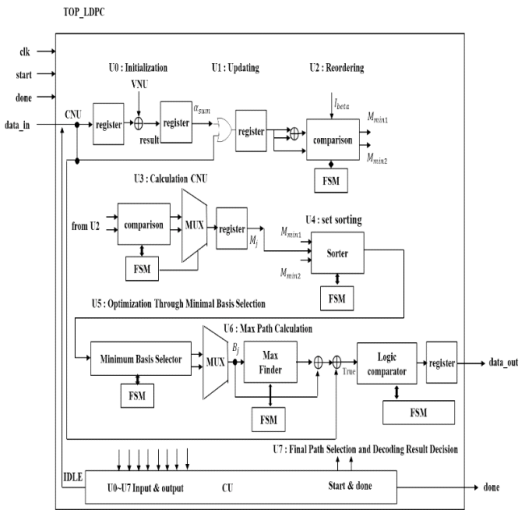

Fig. 6. Hardware Architecture of Proposed Algorithm

In the sixth stage, max_value and xor_result is calculated using data_in and alpha_in, and these values are used to obtain α_{sum}. Finally, in the seventh stage, v_out is calculated using max_value, xor_result, and α_{sum}. Through this process, the original alpha value can be recovered.

The FPGA design process for the Min-Max algorithm began as presented in Fig. 9. Using the Very High-Speed Integrated Circuit Hardware Description Language (VHDL), an LDPC decoder based on the high-reliability Min-Max algorithm was implemented,

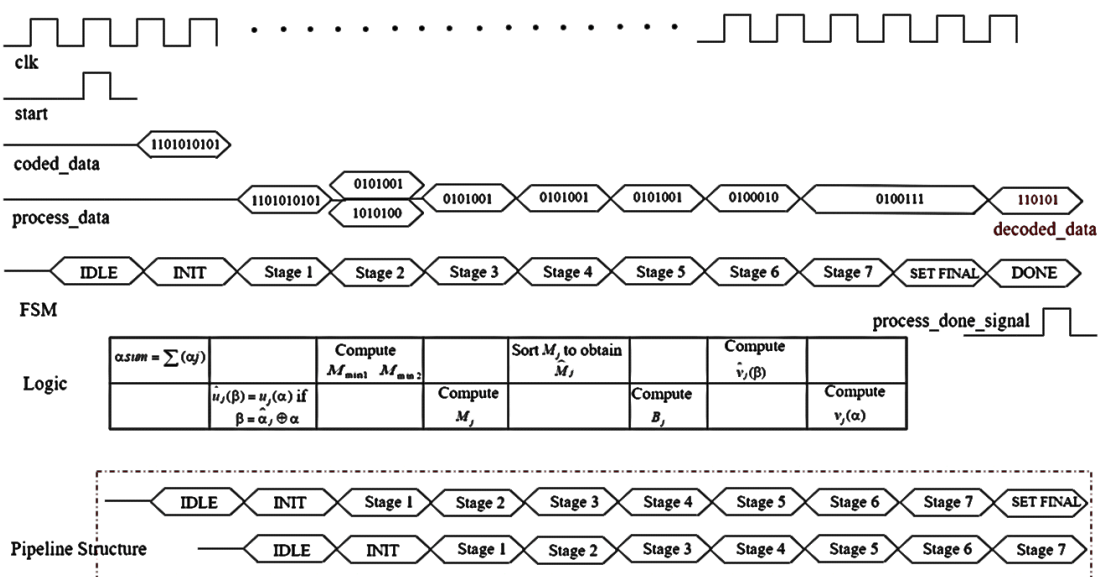

Fig. 7. Timing Diagram of the Proposed Architecture

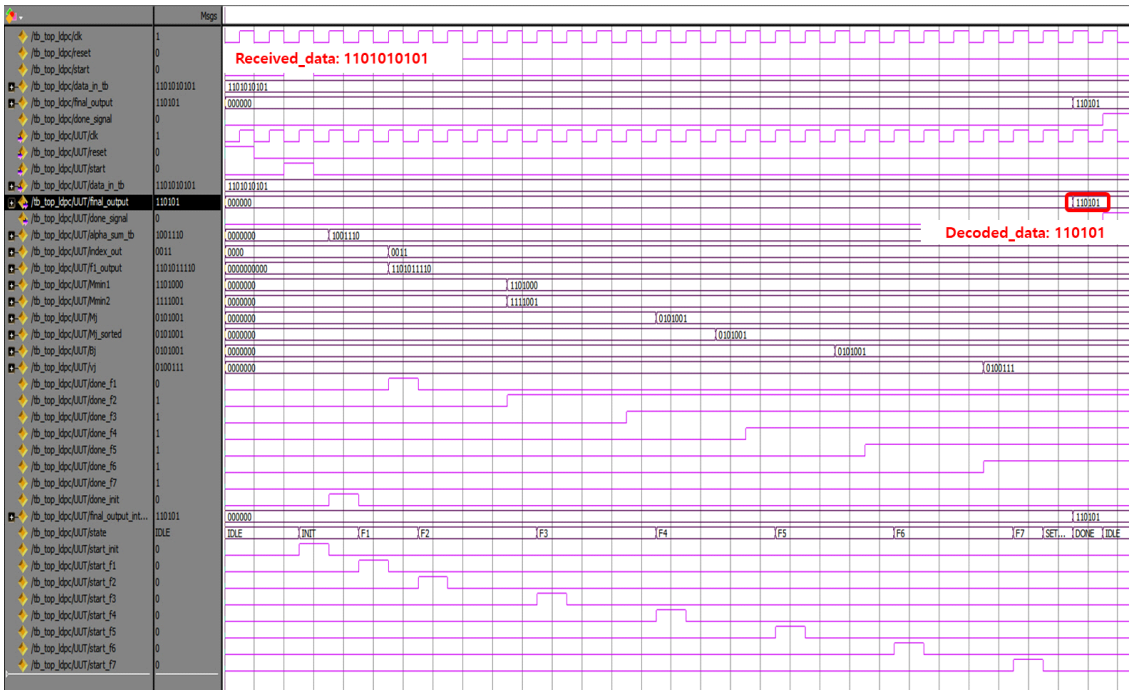

Fig. 8. Simulation Result using EDA Tool

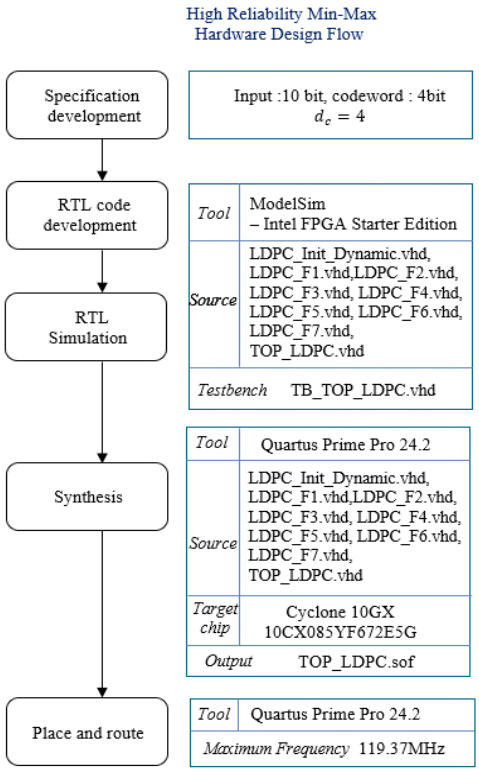

Fig. 9. FPGA Design Flow for Min-max Algorithm

and a Finite State Machine (FSM) was designed to precisely control the decoding process. Through Register Transfer Level (RTL) simulation, it was confirmed that the input value “1101010101” was correctly decoded to the output “110101.” The simulation results were consistent with the expected hardware behavior, and the output signal was verified using an oscilloscope. Fig. 10 illustrates how the designed circuit was placed on the Cyclone 10GX 10CX085YF672E5G chip. After downloading the program to the FPGA, the actual hardware operation was confirmed through the test process presented in Fig. 11. The design operated at 119.37 MHz and utilized 348 LUTs and 255 flip-flops to achieve optimal hardware performance. The test results matched the simulation, and the output signals were accurately verified using the oscilloscope. From a hardware perspective, the proposed design applied parallel processing and a pipeline structure to reduce hardware complexity and maximize performance. XOR operations and optimized lookup tables (LUTs) were used to accelerate the key computational steps of the Min-Max algorithm. Additionally, multiple arithmetic logic

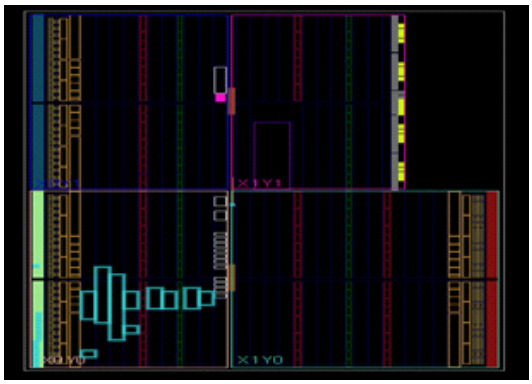

Fig. 10. Chip Layout of FPGA

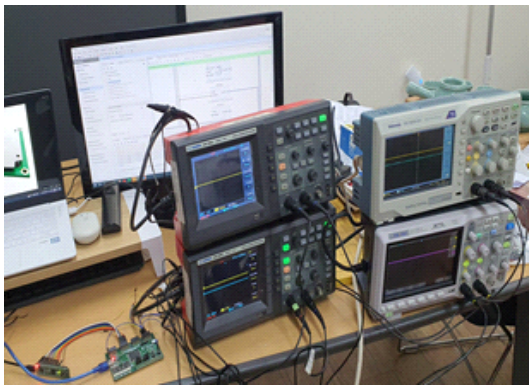

Fig. 11. Test Process for the Programmed FPGA

units (ALUs) were arranged in parallel to efficiently handle high-complexity operations such as comparison and sorting. These optimizations, combined with the inherent parallel processing capabilities of the FPGA, significantly improved throughput while minimizing resource usage.

Additionally, in the software implementation, as shown in Fig. 12, a 6-bit output was generated from a 10-bit input using the C programming language. The software verification results were consistent with the FPGA hardware implementation, confirming that the proposed algorithm was accurately implemented in both software and hardware.

Moreover, the bit error rate (BER) calculation function was implemented using the calculate_ber function, which compares the input bits with the decoded bits and calculates the proportion of incorrectly decoded bits. This function compares the decoded bits with the final output to compute the error rate. In the

Fig. 12. Actual Test Result using C language

current execution, the BER was computed to be 0.000010 according to the 5G standard, as shown in the console output.

Additionally, the throughput calculation function was implemented using the calculate_throughput function, which measures the program's execution time and calculates the number of bits processed per second. Based on the input frequency of 200 MHz, the program recorded a throughput of 119.37 MHz.

Fig. 13 illustrates that in terms of resource usage, this paper significantly reduces the consumption of FFs and LUTs, maximizing resource efficiency, whereas in [17] utilizes more resources to achieve higher performance. In terms of performance, the earlier design achieves a throughput of 165.787 MHz by employing more resources, while this paper minimizes resource usage while still maintaining an adequate performance of 119.37 MHz Furthermore, in terms of design objectives, the earlier approach prioritizes performance, accepting higher resource consumption, whereas this paper emphasizes resource-saving while still delivering efficient performance. Thus, in [17] is suitable for environments where performance maximization is critical, even if it requires significant resource consumption, while the design proposed in this paper offers an optimal solution for environments where hardware constraints necessitate resource efficiency while maintaining adequate performance.

Comparison Items	Ref [17]	Proposed
FFs	7,385	255
LUT	8,713	348
Throughput (MHz)	165.78	119.37

Fig. 13. Comparative Analysis of Different LDPC Decoder Designs

Ⅳ. Conclusions

Hard-decision decoding generally has lower complexity than soft-decision decoding, but it typically offers reduced error correction performance. However, the proposed High Reliability Min-Max Algorithm seeks to enhance decoding accuracy and efficiency by leveraging Log-Likelihood Ratio (LLR) to systematically reduce processing complexity. In this paper, we present a decoder based on this algorithm, which is a type of NB-LDPC. By utilizing the strengths of NB-LDPC's low computational complexity, we have developed an efficient algorithm. The proposed Min-Max Algorithm demonstrated excellent error correction performance in simulations. In future work, we aim to further validate the performance through Application-Specific Integrated Circuit (ASIC) implementation and compare it with other existing methods.

References

[1] S. Jeon and B. V. K. V. Kumar, "Performance and complexity of 32 k-bit binary LDPC codes for magnetic recording channels," in *IEEE Trans. Magnetics*, vol. 46, no. 6, pp. 2244-2247, Jun. 2010. (https://doi.org/10.1109/TMAG.2010.2043067)

[2] K. Ho, C. Chen, Y. Liao, H. Chang, and C. Lee, "A 3.46 Gb/s (9141,8224) LDPC based ECC scheme and on-line channel estimation for solid-state drive applications," *2015 IEEE ISCAS*, pp. 1450-1453, 2015. (https://doi.org/1010.1109/ISCAS.2015.7168917)

[3] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in nand flash memory," in *IEEE Trans. Cir. and Syst. I : Regular Papers*, vol. 58. no. 2, pp. 429-439, Feb. 2011. (https://doi.org/10.1109/TCSI.2010.2071990)

[4] P802.11 - Standard for Information Technology- Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks - *Specific Requirements Part 11: Wireless Local Area Network (LAN) Medium Access Control (MAC) and Physical Layer (PHY) Specifications* (https://standards.ieee.org/ieee/802.11/10548/)

[5] IEEE 802.16a-2003-IEEE Standard for Amendment to IEEE Standard for Local and Metropolitan Area Networks-Part 16: *Air Interface for Fixed Broadband Wireless Access Systems — Medium Access Control Modifications and Additional Physical Layer Specifications for 2-11GHz* (https://www.ieee802.org/16/)

[6] IEEE 802.3an - 2006 - IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems LAN/MAN - Specific Requirements Part 3: *CSMA/CD Access Method and Physical Layer Specifications - Amendment: Physical Layer and Management Parameters for 10Gb/s Operation, Type 10GBASE-T* (https://standards.ieee.org/ieee/802.3an/3560/)

[7] W. Ullah, L. Cheng, and F. Takawira, "Prediction and voting based symbol flipping non-binary LDPC decoding algorithms," *Inst. Electr. and Electr. Eng.*, Oct. 2020. (https://doi.org/10.1109/PIMRC48278.2020.9217108)

[8] R. Gallager, "Low-density parity-check codes," in *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21-28, Jan. 1962. (https://doi.org/10.1109/TIT.1962.1057683)

[9] V. Savin, "Min-max decoding for non binary ldpc codes," in *Proc. IEEE Intl. Symp. Info. Theory*, Toronto, Canada, Jul. 2008. (https://doi.org/10.1109/ISIT.2008.4595129)

[10] J. Lin, J. Sha, Z. Wang, and L. Li, "An efficient vlsi architecture for non-binary ldpc decoders," *IEEE Trans. Cir. and Syst.-II*, vol. 57, no. 1, Jan. 2010. (https://doi.org/10.1109/TCSII.2009.2036542)

[11] J. Lin, J. Sha, Z. Wang, and L. Li, "Efficient decoder design for nonbinary quasicyclic ldpc codes," *IEEE Trans. Cir. and Syst.-I*, vol. 57, no. 5, May 2010. (https://doi.org/10.1109/TCSI.2010.2046196)

[12] X. Zhang and F. Cai, "Efficient partial-parallel decoder architecture for quasi-cyclic non-binary ldpc codes," *IEEE Trans. Cir. and Syst. I*, vol. 58, no. 2, pp. 402-414, Feb. 2011. (https://doi.org/10.1109/TCSI.2010.2071830)

[13] X. Zhang and Fang Cai, "Reduced-complexity decoder architecture for non-binary ldpc codes," *IEEE Trans. VLSI Syst.*, vol. 19, no. 7, Feb. 2011. (https://doi.org/10.1109/TVLSI.2010.2047956)

[14] J. Kang, Q. Huang, L. Zhang, B. Zhou, and S. Lin, "Quasi-cyclic ldpc codes: An algebraic construction," *IEEE Trans. Commun.*, vol. 58, no. 5, May 2010. (https://doi.org/10.1109/TCOMM.2010.05.090211)

[15] X. Chen and C. Wang, "High-throughput efficient non-binary ldpc decoder based on the simplified min-sum algorithm," *IEEE Trans. Cir. and Syst. I: Regular Papers,* vol. 59, no. 11, pp. 2784-2794, 2012. (https://doi.org/10.1109/TCSI.2012.2190668)

[16] T. K. Moon, "Error correction coding," *Wiley*, 2021 (ISBN 9781119567486)

[17] R. V. Devi and S. Rajaram, "An efficient FPGA implementation of LDPC decoder for 5G new radio," *RAEEUCCI*, May 2023. (https://doi.org/10.1109/RAEEUCCI57140.2023.10134365)

Heung-Ryol Yoo

Feb. 2002 : B.S. degree Soon Chun Hyang University

Feb. 2007 : M.S. degree, KonKuK University

Sep. 2018-Current : Ph.D. student, Korea University of Technology and Education (KOREATECH)

2022-Current : Professor, Korea Polytechnics

<Research Interests> Digital Circuit, Error correcting Codes, ASIC

[ORCID:0009-0003-7713-3311]

Yung-Deug Son

Feb. 2001 : MS degree, Kobe University of Mercantile Ocean Electro-Mechanical

Mar. 2001-Aug. 2009 : Senior Research Engineer, Hyundai Heavy Industries Co,.LTD

Feb. 2015 : PhD degree in Electrical Engineering, Pusan National University.

Mar. 2016-Current : Professor, Korea University of Technology and Education

<Research Interests> Digital Circuit, Intelligent control, Optimization algorithm, Electric machine drives

[ORCID:0000-0001-7228-301X]