IMPLEMENTATION OF SOME DECODING ALGORITHMS FOR NB-LDPC CODES ON FPGA

Dam Duc Thuan*, Lai Tien De, Mai Thanh Hai, Tran Van Khan

Abstract: *Non-binary low-density parity-check codes (NB-LDPC) provide better error correction performance in comparison with their counterparts. However, the NB-LDPC decoder has a very high complexity, especially the processing of the check node unit. This paper evaluates the error correction performance of some new decoding algorithms for NB-LDPC codes in different fields with different codeword lengths. The paper also presents the results of the implementation a decoder structure for the NB-LDPC (35,23) over GF(8) on the Spartan 6 board. Analysis and evaluation results show that decoding quality on hardware is equivalent to simulation results on software, demonstrating high feasibility in implementing decoder on a hardware platform, capable of application in devices of the advanced communication systems or high-speed read-and-write data storages.*

Keywords: NB-LDPC codes; STMM; TEC-TMM; Decoding algorithms.

1. INTRODUCTION

Non-binary low-density parity-check codes (NB-LDPC) are defined over GF(q) ($q > 2$), including q elements, which were observed by Davey and MacKay [1]. The NB-LDPC code is better than their binary counterparts in terms of error correction abilities and cluster error correction when the codeword length changes. However, the decoder structure is complex and requires large memory, especially for the check node unit (CNU). This limits the maximum throughput and minimum resources for the decoder structure.

The original decoding algorithm for NB-LDPC codes is the Q-array Sum Product (QSPA) [1] algorithm, which is considered to be the decoding algorithm for optimal error-correcting performance at the cost of high complexity. To reduce decoding complexity, [10] proposed an improved decoding algorithm named Mixed Log-FFT-BP, which has low decoding complexity and suitable for FPGA implementation. Some other approximation algorithms, such as Extended Min-Sum algorithm (EMS) [2] and Min-Max algorithm [3] have been proposed to reduce the complexity of CNU, which is the most complicated process of the NB-LDPC decoder. The min-max algorithm [3] uses comparators instead of adders in [2] to simplify the structure of the CNU as well as avoid increasing arithmetic operations. Currently, the Trellis EMS (T-EMS) algorithm [4, 5] is proposed to increase throughput for the NB-LDPC decoders that accept larger memory costs due to the output messages of CNU, which were created at the same time. In [6], Simplified Trellis Min-max is proposed based on the study in [4], which not only increases decoder throughput but also reduces the complexity of CNU.

The paper evaluates the decoding performance of two new NB-LDPC decoding algorithms, S-TMM and TEC-TMM, and their abilities with the change of codeword lengths and order of Galois field. The paper also presents an implementation of NB-LDPC (35, 23) decoder over GF(8) on Spartan 6 board and evaluates its perfomance in comparison with one which has been done in software.

The rest of the paper is constructed as follows: Section 2 overviews NB-LDPC codes and S-TMM, TEC-TMM decoding algorithms. In section 3, the paper presents the quality assessment results of some NB-LDPC decoding algorithms, which are presented in Section II by the simulation. Section 4 analyzes the implementation of the decoding algorithm for NB-LDPC codes on hardware as well as compares the decoding performance between decoder on FPGA and one in simulation. Finally, conclusions are given in section 5.

2. REVIEW OF NB-LDPC AND SOME NB-LDPC DECODING ALGORITHMS

2.1. Overview of NB-LDPC

The NB-LDPC code is defined by a check matrix **H,** including M rows and N columns. The matrix **H** is sparse, meaning that most of the positions in the matrix have the value 0, the rest of the elements have the non-zero value h_{mn} in Galois field GF(q). Let d_c is the row weight, d_v is the column weight of **H** matrix. NB-LDPC codes are classified as regular and irregular codes, where regular codes are ones with constant d_c and d_v values. NB-LDPC codes are represented by a Tanner graph corresponding to the matrix **H**, with variable nodes representing N columns and check nodes representing M rows. $N(m)$ and $M(n)$ represent the set of variable node d_c connected to m check nodes and d_v check node connected to n*th* check node, respectively. $Q_{m,n}(a)$ and $R_{n,m}(a)$ denote the messages exchanged between n variable nodes and m check nodes (V2C) and from m check nodes to n variable nodes (C2V) for each symbol $a \in$ GF(q). Correct codewords are codewords that satisfy the parity check equation. The iterative decoding algorithm is commonly used for NB-LDPC decoding, including variable node and check node processing, in which the check node processing is always the greatest complexity and will determine the complexity of the algorithm.

2.2. Several new decoding algorithms for NB-LDPC codes

Study [6] has proposed the S-TMM algorithm to generate the output messages of the check node in a parallel way by creating the extra column $\Delta Q(a)$. Initially, the vector d_c value from the variable node $Q_{m,n}(a)$ is converted from normal to delta domain by the expression $\Delta Q_{m,n}(a+z_n) = Q_{m,n}(a)$, where z_n is the n*th* hard decision value with the highest reliability so that the first row of the trellis will be the optimal path for the field element 0, denoted by path 0. Next, the decoder searches for the most reliable value, $m1(a)$ for the symbol $a \in$ GF(q), and their respective positions in a trellis. Optimal paths for nonzero field elements will be paths that contain at least one nonzero element with one non-zero LLR value. The considered configuration set is $conf(1,2)$ to be a set of possible paths created from the message $m1(a)$ with maximum 2 deviations. The LLR value of a path is the maximum of the LLR values on that path, and the corresponding field element for each path is the sum of the field elements on that path. For every non-zero field element, there would exist $q/2$ possible cases considering the configuration set. The optimal path for each field element will be the path with the smallest LLR value among the q/2 path under consideration. The LLR value of this optimal path will

be used to construct the value in the extra column for field element a.

Research [7] has proposed TEC-TMM algorithm based on S-TMM algorithm to reduce complexity in hardware structure with almost equivalent quality. In this algorithm, 2 extra columns $\Delta Q'(a)$ and $\Delta Q''(a)$ are created to update the check node output message. The first extra column $\Delta Q'(a)$, and the deviation indices are constructed similar to the algorithm [6]. The second optimal path with the second smallest LLR value used to construct the second extra column is $\Delta Q''(a)$. The check node output messages are updated by using the values of these 2 extra columns and the deviation indices, which are found in the previous step.

3. SIMULATION RESULTS OF DECODING PERFORMANCE OF NEW NB-LDPC DECODING ALGORITHMS

3.1. Simulation model

The simulation is performed on the software to evaluate the decoding performance. The simulation diagram is shown in figure 1, using the Monte Carlo model, which is used to calculate the frame error, with 1 codeword being considered as a frame.

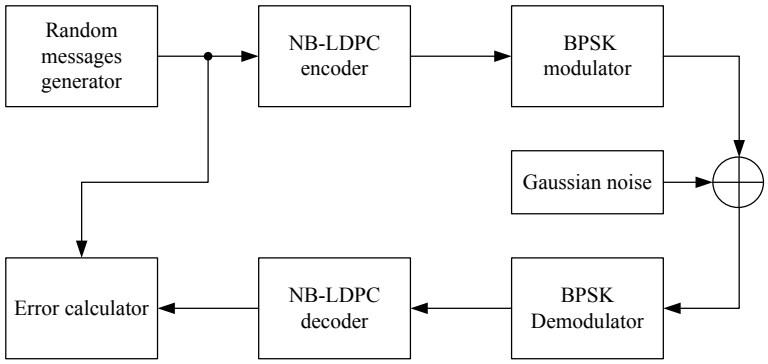

Figure 1. *Simulation diagram used to evaluate decoding performance.*

The functions of the blocks in the diagram above are presented as follows:

Random messages generator: Using the random function to randomly generate messages to be transmitted. Since the code is built on field GF(q), each symbol corresponds to $\log_2(q)$ bits, and the message length is k, so a message has a length of $k.\log_2(q)$ bits. **NB-LDPC encoder:** The message is mapped into symbols and then encoded according to the selected NB-LDPC algorithm. **BPSK Modulator:** The message is remapped as bits of a $n.\log_2(q)$ length and then are modulated. **Transmission channel:** The AWGN channel, the noise is randomly generated according to the Gaussian distribution and added directly to the signal. **BPSK Demodulator**: Calculates the a priori probability for each symbol of the received codeword. **NB-LDPC Decoder:** Input is the a priori probability for each symbol of the codeword, after iterative decoding according to the selected algorithm, we obtain the codeword. **Error calculator:** The check bits are discarded from the decoder output, then information is converted to the bit format and compared with the original message bit sequence.

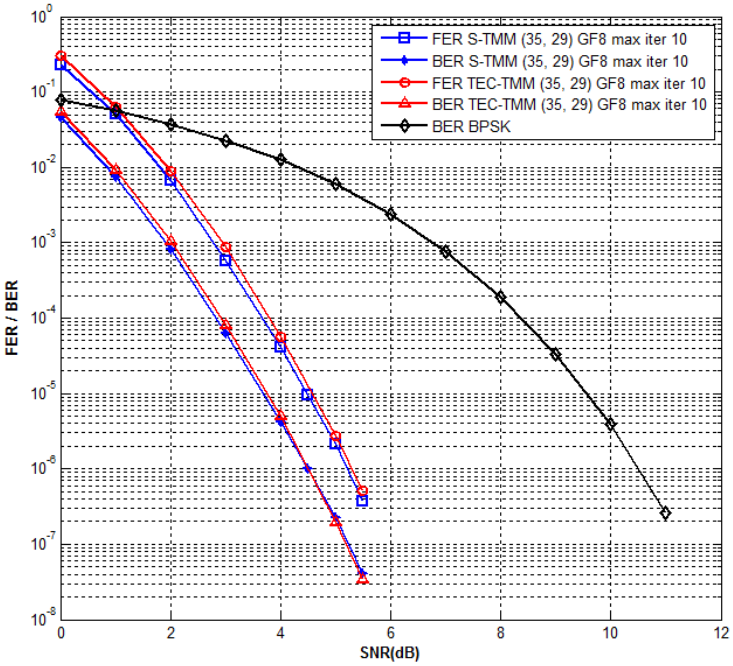

Figure 2. *Decoding performance of S-TMM and TEC-TMM algorithms of NB-LDPC (35,23) code over GF(8).*

3.2. Decoding performance evaluation of S-TMM and TEC-TMM algorithms

Figure 2 shows the results of the decoding performance evaluation of the S-TMM and TEC-TMM decoding algorithms with NB-LDPC (35,23) code over GF(8) using the AWGN channel, with the maximum number of iterations is 10.

The obtained results show that the S-TMM algorithm and the TEC-TMM algorithm bring almost equivalent decoding performance, with a gain of 5dB at BER 10^{-3} and 6.2dB at BER 10^{-6}. With the same decoding abilities, the TEC-TMM algorithm reduces hardware complexity by replacing 2-min finder in S-TMM with 1-min finder in TEC-TMM, the number of bits exchanged between the variable node and the check node is also significantly reduced [7, 9], especially for large fields, when d_c is very larger than $q/2$.

3.3. The decoding performance evaluation of the TEC-TMM algorithm with different length codes

The decoding performance of NB-LDPC codes is higher as the codeword length increases [1]. Figure 3 shows the simulation results of two NB-LDPC (75,45) code and NB-LDPC (120, 69) code according to the TEC-TMM algorithm over GF(16), other parameters of the model have left unchanged.

Simulation results pointed out that NB-LDPC(120, 69) code, which has a longer codeword has a word length that $120 \times 4 = 480$ bits, gives better decoding performance than NB-LDPC(75, 45) code with a codeword length of 300 bits. The longer code gives a gain of 0.5 dB at BER 10^{-5}, and approximately 1dB at BER 10^{-7}. Therefore, with NB-LDPC codes, the longer code, the better decoding performance, especially in low SNR regions.

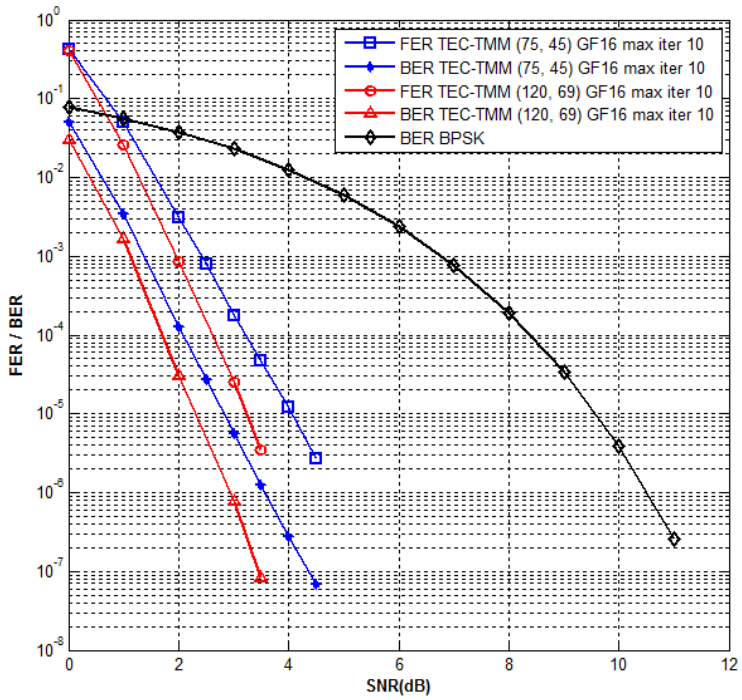

Figure 3. *Decoding performance of TEC-TMM algorithm of NB-LDPC (75, 45) and NB-LDPC (120, 69) codes over GF(16).*

3.4. The quality evaluation of TEC-TMM algorithm decoding in two different fields

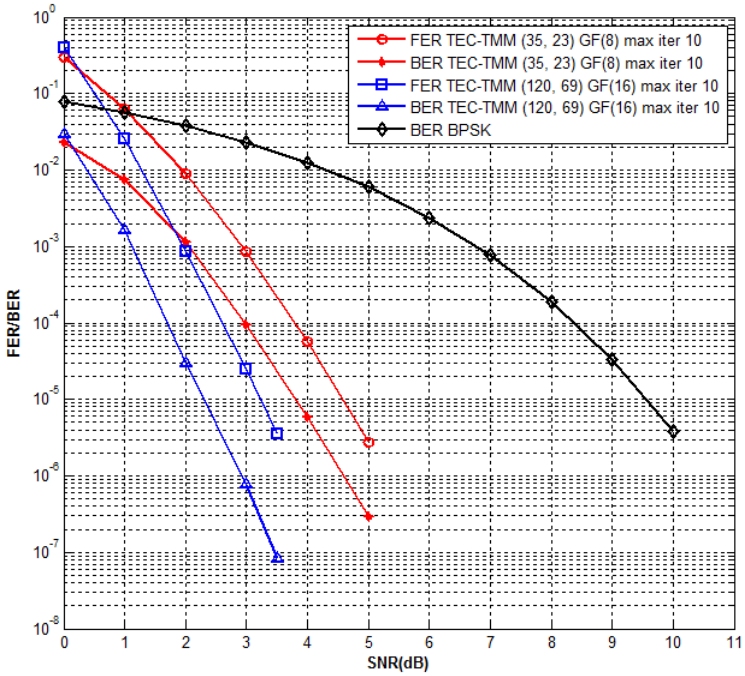

Figure 4. *Decoding performance of TEC-TMM algorithm of NB-LDPC (35, 23) code over GF(8) and NB-LDPC (120, 69) code over GF(16).*

One of the factors affecting the performance of the NB-LDPC codes is that the order of Galois field, according to [1], codes which are represented in the higher-order field give better decoding performance than their representation on lower-order ones, because when represented through non-zero elements of the field, information is not only contained in these symbols, but also in each bit of each element. The higher-order field, the larger number of bits per field element, the smaller information contained in each bit, thus less impact on the content of the message when an error occurs. This is illustrated on the results simulating the decoding performance of NB-LDPC (35, 23) code over GF(8) and NB-LDPC (120, 69) code over GF(16) according to the TEC-TMM algorithm (figure 4).

The simulation results show that the NB-LDPC (120, 69) code over GF(16) gives much better decoding performance than the rest, giving approximately 2dB gain at the bit error rate BER 8.10^{-7}.

4. DESIGN OF NB-LDPC DECODER ON FPGA

4.1. Implementation of NB-LDPC decoder on FPGA

On the basis of the new decoding algorithms that were analyzed in Section 2, the authors implemented the design of the decoder on the FPGA platform. Figure 5 shows the structure of the NB-LDPC decoder, which will be implemented on hardware. Specifically, the script used for the design is as follows:

- Decoder is designed for NB-LDPC (35, 23) code over GF(8);
- Check matrix **H** is built according to the array dispersions method [8];
- Using board Xilinx Spartan 6 XC6SLX25.

The decoder structure will be designed to include a decoding unit, data input, data output, logic pins, and clock.

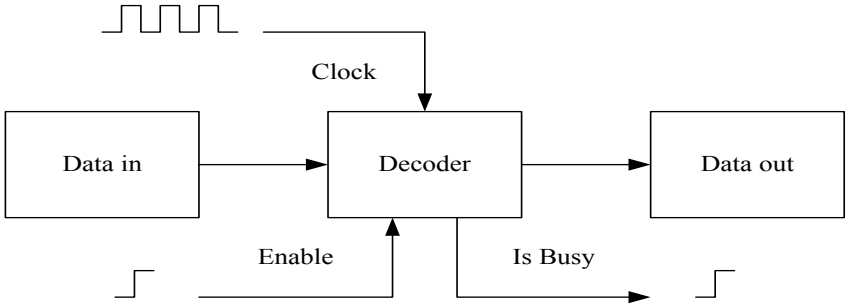

Figure 5. *Structure of NB-LDPC(35, 23) over GF(8) decoder on Spartan 6.*

The data input is the posterior probabilities for each symbol of the codeword that has been normalized and converted to the logarithmic format. The Enable pin is pushed to a high level in 1 clock pulse, the Decoder will start to load data into the memory (RAM), and put Is_Busy pin to high to indicate the decoder is busy. The decoder has finished loading the data and will decode it according to the designed algorithm. When the codeword satisfies the parity check equation or reaches the maximum number of iteration, the decoder will push the codewords out and then lowering Is_Busy to a low level.

A decoder is written in VHDL, the design process includes 2 steps. Step 1:

Design and simulate on ModelSim to check the operation of the decoding that it is the same as the algorithm on the software or not. Step 2: Using the ISE tool of Xilinx, load the design into the Spartan 6 board and check the decoder operation on this hardware. The resource usage results of the designed decoder are shown in table 1.

Table 1. *Resource used for NB-LDPC(35, 23) decoder on Spartan-6.*

Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	6753	30064	22%
Number of Slice LUTs	10113	15032	67%
Number of fully used LUT-FF pairs	5792	11074	52%
Number of bonded IOBs	6	186	3%
Number of BUFG/BUFGCTRLs	8	16	50%

The inspection process is performed according to the schematic shown in figure 6, in which the schematic is divided into 2 parts. The upper part includes error calculations, random message generation, encoding, modulation, and demodulation, all performed on a computer. This is aimed at reducing the use of hardware resources on other functions than decoding, and for the most accurate evaluation results.

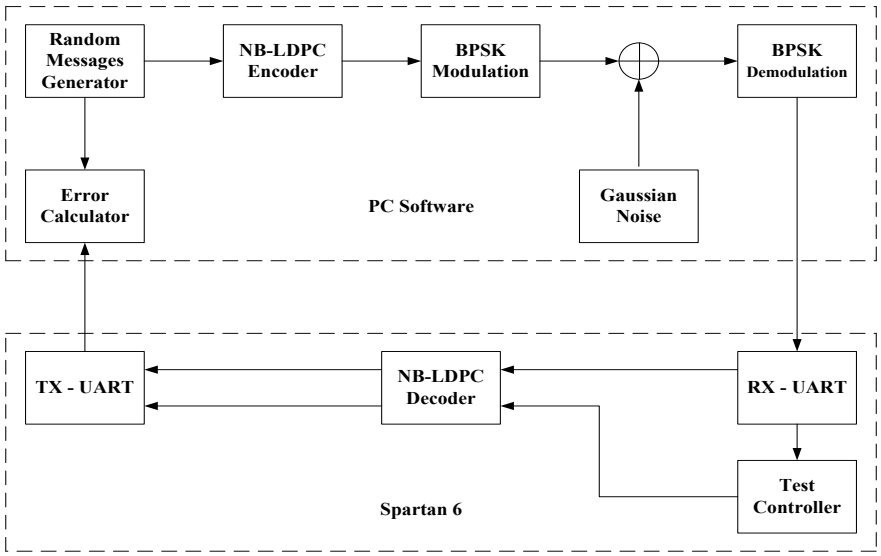

Figure 6. *Decoding performance verification diagram of the decoder on hardware.*

The lower part has the NB-LDPC decoding function, which is the decoding unit designed on the hardware. The data transmission between the computer and the decoder is performed by RX-UART and TX-UART, leading to difficulties in showing the advantages of hardware processing speed, so we only evaluate the performance of the decoder on the hardware compared to the evaluated results on

the computer.

4.2. Performance comparison of NB-LDPC decoder performed on FPGA and one by simulation

The decoder, after being implemented on FPGA, will be tested for decoding by comparing the decoding performance that has been evaluated based on the simulation. The decoder performance assessment was presented in section 4.1, which is a combination of the computer software and the implemented decoder. The results compare the decoding ability of the decoder on the hardware, and the simulation results on the software are shown in figure 7.

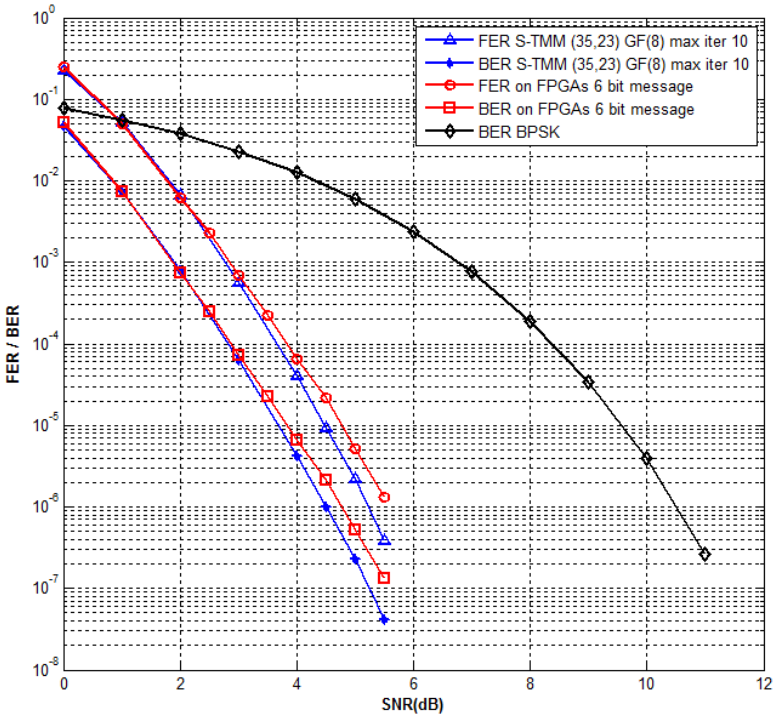

Figure 7. *Comparison of the decoding performance of decoder on software with decoder designed on FPGA.*

The decoder designed on FPGA has a decoding performance equivalent to that of the decoder simulated on the software. The abilities of the decoder on the hardware are reduced by a negligible amount due to the use of fixed-point (12-bit) quantization compared to the floating-point scheme.

5. CONCLUSIONS

The study evaluated and compared the decoding quality of some new decoding algorithms for NB-LDPC codes in different fields, with different codeword lengths. A decoder was performed on the Spartan 6 FPGA board; the analysis showed that the quality of the decoder on the hardware was comparable to that performed on the simulation. The NB-LDPC code exhibits a very high quality with hardware-based feasibility, showing it to be a candidate for high-speed read and write memory and communications applications.

REFERENCES

[1]. Hllatthew C Davey and David JC MacKay, "*Low-density parity check codes over GF(q)*," in Information Theory Workshop, Jun., pp. 165–167, (1998).

[2]. David Declercq and Marc Fossorier, "*Decoding algorithms for nonbinary LDPC codes over GF(q)*," IEEE Trans. Commun., vol. 55, no. 4, pp. 633–643, Apr. (2007).

[3]. Valentin Savin, "*Min-max decoding for non binary LDPC codes*," in Proc. IEEE Int. Symp Inf. Theory, Toronto, ON, Canada, Jul., pp. 960–964, (2008).

[4]. Erbao Li, David Declercq, and Kiran Gunnam, "*Trellis based extended min-sum algorithm for non-binary LDPC codes and its hardware structure*," IEEE Trans. Commun.,vol. 61, no. 7, pp. 2600–2611, Jul. (2013).

[5]. Erbao Li, Francisco Garcia-Herrero, David Declercq, Kiran Gunnam, Jesus O Lacruz, and Javier Valls, "*Low latency T-EMS decoder for non-binary LDPC codes*," in in Proc. Conf. Rec. 47th Asiloma Conf. Signals, Syst. Comput. (ASILOMAR), Nov., pp. 831–835, (2013).

[6]. Jesús O Lacruz, Francisco Garc´ ıa-Herrero, David Declercq, and Javier Valls, "*Simplified trellis min-max decoder architecture for nonbinary low-density parity check codes*," IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 23, no. 9, pp. 1783–1792, Sep. (2015).

[7]. H. P. Thi and H. Lee, "*Two-extra-column trellis min–max decoder architecture for nonbinary LDPC codes*," IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 25, no. 5, pp. 1787–1791, May. (2017).

[8]. Bo Zhou, Jingyu Kang, Shumei Song, Shu Lin, Khaled Abdel-Ghaffar, and Meina Xu, "*Construction of non-binary quasi-cyclic LDPC codes by arrays and array dispersions,*" IEEE Trans. Commun., vol. 57, no. 6, pp. 1652–1662, Jun. (2009).

[9]. Nghia Pham Xuan, Huyen Pham Thi, Thuan Dam Duc, "*Reduced-complexity trellis min-max decoder for non-binary LDPC codes*", in The National Conference on Electronics, Communications and InformationTechnology, pp. 89-92, Dec. (2018).

[10]. Ziyong Wang, Jiahui Meng, Zhibin Deng, Liang Zhang, Jingpeng Gao, *"FPGA Implementation Scheme of Mixed Logarithmic Domain FFT-BP Decoding Algorithm Based on Non-Binary LDPC Codes"* in 37th Chinese Control Conference (CCC), pp. 8459-8464, Jul. (2018).

[11]. Injun Choi, Ji-Hoon Kim, *"Memory-Reduced Non-Binary LDPC Decoding with Accumulative Bubble Check",* IEEE Transactions on Circuits and Systems II: Express Briefs, (2017).

[12]. Min-Ho Kim, Kyeong-Bin Park, Ki-Seok Chung, *"A Low-Complexity Adaptive Extended Min-Sum Algorithm for Non-Binary LDPC Codes",* 5th International Conference on Computational Intelligence and Applications (ICCIA), (2020).

[13]. Ángel Álvarez; Víctor Fernández; Balázs Matuz, *"An Efficient NB-LDPC Decoder Architecture for Space Telecommand Links",* IEEE Transactions on Circuits and Systems II: Express Briefs, Oct. (2020).

TÓM TẮT

HIỆN THỰC HÓA MỘT SỐ THUẬT TOÁN GIẢI MÃ CHO MÃ NB-LDPC TRÊN FPGA

Mã kiểm tra chẵn lẻ mật độ thấp phi nhị phân (NB-LDPC) cho phẩm chất sửa lỗi tốt hơn so với phiên bản mã nhị phân cùng loại. Tuy nhiên, bộ giải mã NB-LDPC có độ phức tạp rất cao, đặc biệt là quá trình xử lý nút kiểm tra. Bài báo này đánh giá chất lượng sửa lỗi của một số thuật toán giải mã mới cho mã NB-LDPC trên các trường khác nhau với các mã có độ dài từ mã khác nhau. Bài báo cũng trình bày kết quả hiện thực hóa một cấu trúc bộ giải mã triển khai cho bộ mã NB-LDPC (35, 23) trên trường GF(8) trên bo mạch Spartan 6. Kết quả phân tích và đánh giá cho thấy chất lượng giải mã trên phần cứng tương đương với kết quả mô phỏng trên phần mềm, thể hiện tính khả thi cao trong việc hiện thực hóa bộ giải mã trên nền tảng phần cứng, có khả năng ứng dụng trong các thiết bị thuộc hệ thống truyền thông tiên tiến hay các bộ nhớ đọc ghi tốc độ cao.

Từ khóa: NB-LDPC; Thuật toán giải mã; S-TMM; TEC-TMM.

Received 23rd October 2020
Revised 09th November 2020
Published 16th November 2020

Author affiliations:
Military Technical Academy.
*Corresponding author: thuanmta@gmail.com.