entropy

MDPI

Article

Bilayer LDPC Codes Combined with Perturbed Decoding for MLC NAND Flash Memory

Lingjun Kong [1], Haiyang Liu [2,*], Wentao Hou [3] and Chao Meng [1]

[1] Faculty of Network and Telecommunication Engineering, Jinling Institute of Technology, Nanjing 211169, China; kong@jit.edu.cn (L.K.); mengchao@jit.edu.cn (C.M.)
[2] Institute of Microelectronics, Chinese Academy of Sciences, Beijing 100029, China
[3] College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 1220013235@njupt.edu.cn
* Correspondence: liuhaiyang@ime.ac.cn

Abstract: This paper presents a coding scheme based on bilayer low-density parity-check (LDPC) codes for multi-level cell (MLC) NAND flash memory. The main feature of the proposed scheme is that it exploits the asymmetric properties of an MLC flash channel and stores the extra parity-check bits in the lower page, which are activated only after the decoding failure of the upper page. To further improve the performance of the error correction, a perturbation process based on the genetic algorithm (GA) is incorporated into the decoding process of the proposed coding scheme, which can convert uncorrectable read sequences into error-correctable regions of the corresponding decoding space by introducing GA-trained noises. The perturbation decoding process is particularly efficient at low program-and-erase (P/E) cycle regions. The simulation results suggest that the proposed bilayer LDPC coding scheme can extend the lifetime of MLC NAND flash memory up to 10,000 P/E cycles. The proposed scheme can achieve a better balance between performance and complexity than traditional single LDPC coding schemes. All of these findings indicate that the proposed coding scheme is suitable for practical purposes in MLC NAND flash memory.

Keywords: bilayer low-density parity-check (LDPC) codes; genetic algorithm (GA); perturbation noise; NAND flash memory; multi-level cell (MLC)

check for updates

Citation: Kong, L.; Liu, H.; Hou, W.; Meng, C. Bilayer LDPC Codes Combined with Perturbed Decoding for MLC NAND Flash Memory. *Entropy* **2024**, *26*, 54. https://doi.org/10.3390/e26010054

Academic Editors: Jun Chen and Sadaf Salehkalaibar

Received: 29 November 2023
Revised: 28 December 2023
Accepted: 29 December 2023
Published: 8 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the past decade, NAND flash memories have been extensively applied in consumer electronic devices as they are characterized by nonvolatility, fast read and write speeds, and low power consumption. With the continuous miniaturization of the NAND process, novel storage methods—i.e., multi-level cell (MLC)/triple-level cell (TLC), and three-dimensional (3D) stacking technology—have been applied in order to improve the storage capacity of NAND flash memory [1]. These methods, however, inevitably cause a decrease in the storage reliability of NAND flash memories [2,3], which is characterized by lower program-and-erase (P/E) cycle endurance, increased susceptibility to cell-to-cell interference (CCI), and shorter data retention.

Error correction coding (ECC) is an effective technique for addressing the reliability problem of flash memories. In particular, low-density parity-check (LDPC) codes [4] and polar codes [5] have been applied in ultra-high-density flash memory systems due to their near-capacity performances in symmetric binary discrete memoryless channels (B-DMCs). The work of [6] provided a theoretical framework for constructing and analyzing LDPC codes for asymmetric storage channels. To tackle the asymmetric problem of MLC channels, the authors in [7] proposed an optimization algorithm based on reciprocal channel approximation (RCA) for extrinsic information transfer (EXIT) charts to optimize the degree distributions of an LDPC code, as well as for MLC read voltages. Due to their applications in rank modulation schemes for flash memories, codes in the L_1 metric have been widely

investigated [8]. Two families of error-scrubbing codes based on the L_1 metric and a modular construction, which outperform conventional ECCs for MLC channels, were presented in [9]. According to intracell bit error characteristics and the error spacing strategy, the works of [10,11] improved the min-sum algorithm and bit-flip algorithm, respectively, for decoding LDPC codes (which can speed up the convergence and reduce the decoding delay). In our previous work [12], we designed new LDPC codes and rate-adaptive polar codes by utilizing the intrinsic properties of an MLC NAND flash channel. Specifically, the basis matrix of a protograph LDPC code was constructed based on a sequence of degrees that were optimized by the modified EXIT chart method for flash memories, while the rate-adaptive polar coding was achieved by iteratively calculating the new Bhattacharyya parameters of the memory cell bits. All of these findings suggested that the asymmetric property of errors among different pages can provide us with new insights for the design of ECC schemes for MLC NAND flash memories.

In the signal processing and coding areas, it can be shown that, under certain conditions, additional performance gains for detectors and decoders can be achieved by introducing noises to the received signals [13–17]. The authors in [16] presented a generalized framework and improved perturbation selection criteria for multi-round LDPC decoding schemes, in which pulse perturbations were applied to only a few symbols. A belief propagation list (BPL) algorithm that relies on artificial noise was proposed for decoding polar codes in [17]. The authors of [18] proposed a cyclic redundancy check (CRC)-assisted perturbation algorithm for decoding polar codes, wherbey when the CRC of a simplified successive cancellation (SSC) decoder fails, a number of possible candidate vectors for re-decoding are obtained by adding perturbation noises. In our previous work [19], we proposed several decoding algorithms for polar codes in AWGN channels, which can convert uncorrectable received sequences into the error-correcting regions of the sequences' decoding space.

In this paper, we propose a coding scheme based on the bilayer LDPC codes in [20] for improving the reliability of an MLC NAND flash memory channel, where the channel model is the same as in [4]. Bilayer LDPC codes, an important class of structured LDPC codes used in cooperative relaying systems [21], can provide further diversity and coding gains by combining code words from a source code with the additional parity bits provided by a relay. In this work, we exploited the asymmetric error correction capabilities of MLC pages and placed the extra parity-check bits of a bilayer LDPC code in the lower page, which is a different approach from our previous work [12], where the differences betwene the memory cell bits were used to design ECC codes. These extra parity-check bits were activated only after the decoding failure of the upper page had occurred, which is where the channel state was relatively poor. Then, we applied the perturbation algorithm for the proposed scheme to further improve the decoding performance. The simulation results suggested that our proposed bilayer coding scheme can outperform the traditional LDPC coding scheme in MLC NAND flash channels. In addition, the proposed bilayer coding schemes can be easily extended to 3D flash memory.

Overall, the main contributions and novelties of this paper are summarized as follows:

1. A coding scheme based on bilayer LDPC codes is designed for an MLC NAND flash memory channel, which stores the extra parity-check bits in the lower pages with relatively good channel conditions. To the best of our knowledge, this is the first bilayer LDPC coding scheme that has been successfully designed for an MLC flash channel. The bilayer LDPC decoding algorithm is activated only when the decoding of the upper layer fails. Hence, the decoding complexity increases only slightly at low P/E cycle regions.
2. To further improve the decoding performance, the GA is applied to the proposed bilayer LDPC coding scheme, which produces the genetic noises that rapidly draw the received sequences back to their error-correctable decoding space. The reliability of the storage system is improved as a consequence.

The rest of the paper is organized as follows. In Section 2, we briefly review the related works on bilayer LDPC codes and the perturbation theory. In Section 3, the proposed bilayer LDPC coding schemes with perturbation decoding algorithms are presented. Section 4 presents the experimental results, and Section 5 concludes the paper.

2. Related Works

In this section, a brief review of related works is presented. We first provide an overview of bilayer LDPC codes. Then, we review the perturbation theory.

2.1. Bilayer LDPC Codes

A binary bilayer LDPC code is represented by a Tanner graph, which is denoted by $(\boldsymbol{U}_\alpha \cup \boldsymbol{V}_v \cup \boldsymbol{L}_\beta, E)$, where $\boldsymbol{U}_\alpha = \{u_1, u_2, \cdots, u_{k_1}\}$ and $\boldsymbol{L}_\beta = \{l_1, l_2, \cdots, l_{k_2}\}$ are the sets of check nodes of the upper subgraph and lower subgraph, respectively; $\boldsymbol{V}_v = \{v_1, v_2, \cdots, v_N\}$ represents the set of variable nodes that connect the upper and lower subgraphs; and E is the set of edges such that $E \subseteq (\boldsymbol{U}_\alpha \times \boldsymbol{V}_v) \cup (\boldsymbol{V}_v \times \boldsymbol{L}_\beta)$ (see Figure 1). The red dotted edges in Figure 1 form a 6-cycle in bilayer Tanner graph that contains check nodes from both the upper subgraph and lower subgraph. The key to the construction of a bilayer LDPC code is how to jointly design the upper subgraph (the upper parity-check matrix $\boldsymbol{H}_u$) and lower subgraph (the lower parity-check matrix $\boldsymbol{H}_l$) so that both subcodes and the whole bilayer code have excellent error correction performances while maintaining low complexities.

The algebraic structures of the quasi-cyclic (QC)-LDPC codes facilitate their encoding and decoding processes, which leads to their wide applications [22]. Based on the index matrices that determine the shifts of the circulant matrices of parity-check matrices, QC-LDPC codes with a variety of code lengths and rates can be designed.

In this paper, we constructed bilayer QC-LDPC codes for MLC flash memory based on the design approach proposed in [20]. The parity-check matrices of QC-LDPC codes with various code rates can be constructed by selecting appropriate parameters for bilayer QC-LDPC codes. By following Theorem 1, as well as its corollaries in [20], we are able to design $\boldsymbol{H}_u$ and $\boldsymbol{H}_l$ jointly in order to ensure the constructed codes had good properties, e.g., large minimum and stopping distances. We refer the readers to [20] for further details on how to construct bilayer QC-LDPC codes.

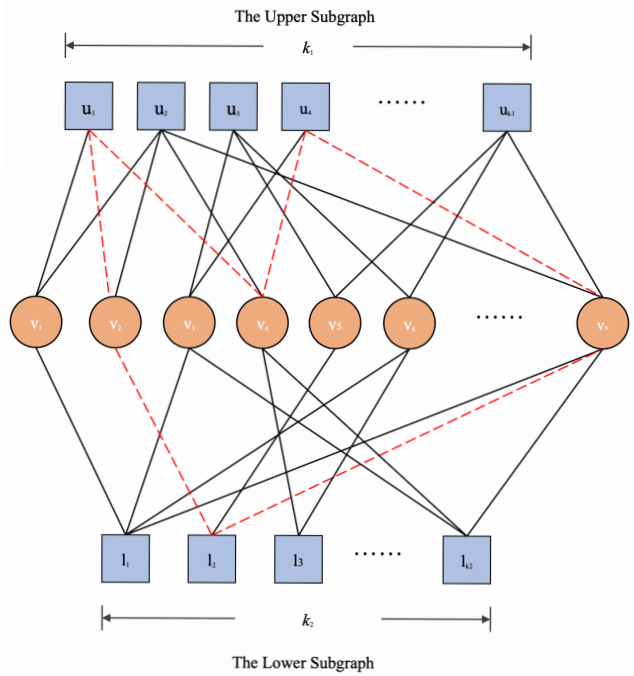


Figure 1. A Tanner graph for a bilayer LDPC code.

2.2. Perturbation Theory

In this subsection, we will briefly introduce the perturbation theory. Benzi [23] discovered the stochastic resonance phenomenon, i.e., when the application of suitable noises can enhance the quality of a signal transmission [24,25]. The concept of stochastic perturbation opened up a new direction in the signal processing area. A perturbed decoding algorithm for a CRC-aided convolutional coding scheme was proposed in [26], which is where a perturbed received signal is decoded by the Viterbi algorithm when the CRC fails. The theory behind the perturbation decoding was analyzed extensively in [27]. As an inevitable interference factor in the storage process, the noise has a huge impact on information storage. Severe noise interference directly leads to the loss of stored information and affects the performance of the storage system. Based on the stochastic resonance phenomenon, artificially generating specific noise in the decoding space may compensate the effect caused by the channel noise and promote success in the decoding process [19].

Figure 2 illustrates the schematic diagram of the decoding space. Every valid code word $\boldsymbol{b} \in \{\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_s}\}$ in a decoding space has a corresponding error-correcting region $\boldsymbol{a} \in \{\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_s}\}$, where s represents the number of valid code words. The decoder can correctly decode when the signal $\boldsymbol{y}$ received is within the error-correcting region. Otherwise, the received signal is beyond the error-correcting area, thus indicating that the current channel noise influence has exceeded the error correction range of the decoder. If a stochastic perturbation noise $\boldsymbol{n}$ is added to a received signal that fails to be decoded, a new perturbation signal $\boldsymbol{y'} = \boldsymbol{y} + \boldsymbol{n}$ is constructed, which may fall into a certain error-correcting region and would then achieve a successful decoding.

However, the direction of random perturbation noise is arbitrary, and there is no guarantee that the noise can definitely help the received signal move to the correct error-correcting region. Therefore, some optimization algorithms can be used to artificially modify random perturbation noises, such that they can evolve in a direction that is helpful for correct decoding. Complex problems in fields like machine learning, combinatorial optimization, signal processing, and channel coding can be solved by the GA—a general framework that does not need gradient information or other auxiliary knowledge in the optimization process [28]. In particular, the GA has been widely used in the field of channel coding, such as in LDPC code construction and decoding [28,29], polar code construction and decoding [30,31], etc. It has been demonstrated in [19] that, in an AWGN channel, a reasonable selection of noise variance, as well as the evolutionary criterion, can obtain the desired perturbation noise and thus improve the decoding performance. In the following section, a GA-based perturbed decoding algorithm is proposed for the application of bilayer LDPC codes in an MLC flash channel.

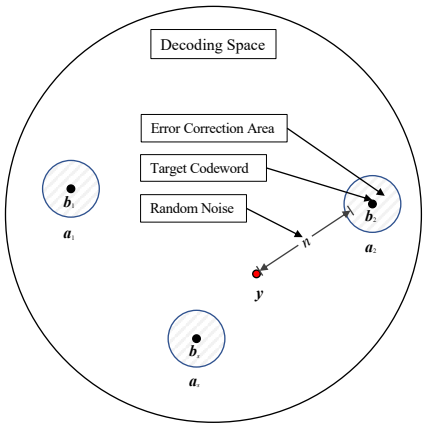


Figure 2. Schematic diagram of perturbation decoding.

3. Bilayer LDPC Codes Applied with a Perturbed Decoding Algorithm

In this paper, QC-LDPC codes were used as subcodes in a bilayer coding scheme that was applied to MLC flash channels. Then, a perturbed decoding algorithm for bilayer LDPC codes in MLC channels was proposed based on the perturbation principle.

3.1. Bilayer LDPC Code Design for an MLC Flash Channel

Assume that the code word begins with the information bits. The information sequence $\boldsymbol{c}^u = [c_1, c_2, \ldots, c_{N-k_1}]$ is encoded by the upper parity-check matrix, which is expressed by

$$\boldsymbol{H}_u \cdot \boldsymbol{x} = \boldsymbol{H}_u \cdot \left[c_1, c_2, \ldots, c_{N-k_1}, p_1, p_2, \ldots, p_{k_1}\right]^T = \boldsymbol{0}, \tag{1}$$

where $[p_1, p_2, \ldots, p_{k_1}]$ contains the parity-check bits generated in the upper layer.

Suppose that the bilayer LDPC code's upper parity-check matrix $\boldsymbol{H}_u$ can be split into two submatrices as follows:

$$\boldsymbol{H}_u = [\boldsymbol{H}_u^{\theta}|_{\,k_1 \times (N-k_1)} | \boldsymbol{H}_u^{\gamma}|_{\,k_1 \times k_1}]. \tag{2}$$

If $\boldsymbol{H}_u^{\gamma}$ is a non-singular matrix, the k_1 parity-check bits can be obtained as follows:

$$\left[p_1, p_2, \ldots, p_{k_1}\right]^T = [\boldsymbol{H}_u^{\gamma}]^{-1} \boldsymbol{H}_u^{\theta} [c_1, c_2, \ldots, c_{N-k_1}]^T. \tag{3}$$

The k_2 parity-check parity bits generated by the lower subgraph of a bilayer LDPC code (with a parity-check matrix $\boldsymbol{H}_l$) are as follows:

$$\boldsymbol{S} = [q_1, q_2, \ldots, q_{k_2}]^T = \boldsymbol{H}_l|_{k_2 \times N} \cdot \boldsymbol{x}. \tag{4}$$

The overall parity-check matrix is

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{H}_u|_{k_1 \times N} & \boldsymbol{O}|_{\,k_1 \times k_2} \\ \boldsymbol{H}_l|_{k_2 \times N} & \boldsymbol{I}|_{\,k_2 \times k_2} \end{bmatrix}, \tag{5}$$

and we have

$$\boldsymbol{H} \cdot \begin{bmatrix} \boldsymbol{X}|_{\,N \times 1} \\ \boldsymbol{S}|_{\,k_2 \times 1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0}|_{\,k_1 \times 1} \\ \boldsymbol{0}|_{\,k_2 \times 1} \end{bmatrix}. \tag{6}$$

For an MLC flash memory cell, the most significant bit (MSB) and the least significant bit (LSB) belong to an upper page and a lower page, respectively. The upper page bit and the lower page bit of the same cell are combined into one code word [4]. The bit error probabilities in flash cells are quite unbalanced, thus resulting in the unequal error rate performances of its upper and lower pages. In general, the raw error rate performance of the lower page is better than that of the upper page as the P/E cycle number increases [10].

Motivated by the observation of intracell unbalanced bit error probabilities in an MLC NAND flash channel, we proposed a coding scheme based on bilayer LDPC codes to improve the reliability and effectiveness of MLC flash memory systems. The specific encoding process established is as follows:

1. Upper Page Encoding: The upper page information sequence $\boldsymbol{c}^u$ is encoded using the matrix $\boldsymbol{H}_u$ to obtain the parity-check vector of length k_1 of the upper page from Equation (3).
2. Generation of the Extra Parity-Check Bits: Specifically, the upper page information bits and the corresponding parity-check bits generated in Step 1 are jointly encoded using the matrix $\boldsymbol{H}_l$ to obtain the extra parity-check bits of length k_2, which are stored on the lower page.
3. Lower Page Encoding: Jointly encode the lower page information sequence $\boldsymbol{c}^l$ and the extra parity-check bits of length k_2 from the upper page to generate the lower page

parity-check bits via a separate channel coding scheme (which is not within the scope of this paper).

4. MLC Programming: The corresponding lower and upper bits are converted into voltages through an iterative incremental pulse programming method. The details of the bilayer LDPC encoding processes are shown in Figure 3.

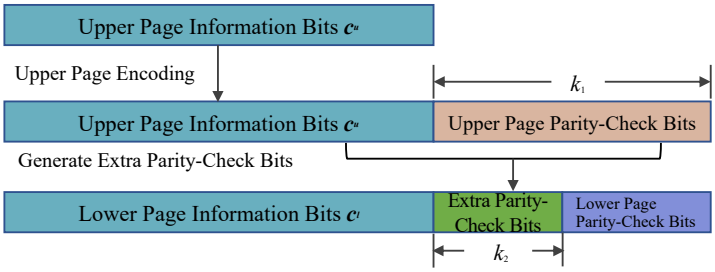


Figure 3. Schematic diagram of the bilayer LDPC encoding process in an MLC NAND flash channel.

In our bilayer LDPC decoding algorithm, the lower and upper pages are decoded separately, as shown in Figure 4. It is only when the upper page decoding fails that the information on the extra parity-check bits in the MLC lower page load and the bilayer LDPC decoding is performed. The specific decoding process is as follows:

1. Read Operation: The voltage in the MLC flash memory is quantified using the soft sensing method. We used the soft-decision storage sensing approach, in which more than one quantization level is used between two adjacent storage states. Thus, the soft information of the code word corresponding to the threshold voltage of each memory cell is obtained.
2. Lower Page Decoding: The lower page information bits and the extra parity-check bits of length k_2 are estimated. The information on the extra parity-check bits is stored for use in the bilayer LDPC decoding.
3. Upper Page Decoding: The upper page is decoded using the matrix $\boldsymbol{H}_u$. If the decoding is successful, the decoding result is output and the decoding process is finished; otherwise, the information on the extra parity-check bits is read from the lower page and the bilayer decoding is carried out in the next step.
4. Bilayer Decoding: The bilayer decoding is performed until the decoding is successful or re-decoded with high-precision quantization.

The details of the proposed decoding algorithm of the bilayer LDPC code applied in an MLC channel are given in Figure 4. Note that, during the whole decoding process, the lower page decoding is employed only when the upper page decoding fails.

3.2. GA-Based Perturbed Decoding Algorithm

In order to further improve the decoding performance in an MLC flash memory channel, a perturbed decoding method based on the GA was proposed. The proposed algorithm, on the premise of making full use of the asymmetric characteristics of an MLC channel, can reduce the number of extra parity-check bits required while ensuring a good decoding performance. The difference in this approach compared to our previous work [19] is that the algorithm proposed in this paper uses only the GA to generate the noises directly to assist the proposed bilayer LDPC coding scheme in an MLC NAND flash memory channel. As shown in Figure 5, when the decoding of a bilayer LDPC code fails, the perturbed decoding algorithm is activated. Then, the genetic noise generated by the GA will be added to the upper page bits for decoding again. The specific steps of the method are as follows:

1. Initialization: In the initialization process, a group of noises with Gaussian distribution is randomly generated by the noise generator as the initialization population. Compared with the channel noise, the artificially added perturbation noise power can be neglected. Genetic noise with a standard deviation between 0 and 0.385 is usually

used. It should be noted that the length of the perturbed noise sequence is the same as that of the received sequence, which conforms to the coded form of the population. Hence, no additional coding operation is required for the population.

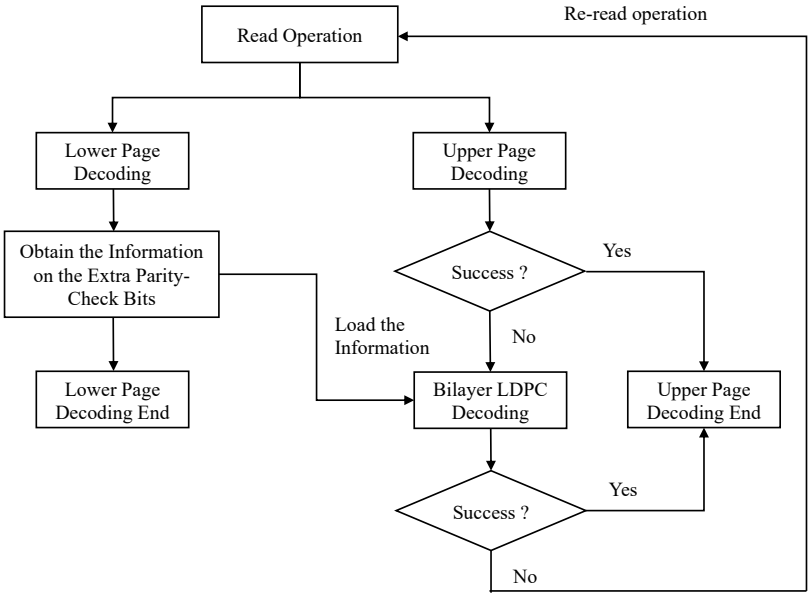


Figure 4. A flow chart of the bilayer LDPC decoding process in an MLC NAND flash channel.

2. Crossover and mutation: By generating new individuals, the crossover and mutation operations expand the scope of the solution space and increase the diversity of the individuals—which are provided in the form of the 'crossover' function and 'mutation' function, respectively—in Algorithm 1. Suppose two parents have chromosomes represented by $\boldsymbol{v} = (v_1, v_2, ..., v_N)$ and $\boldsymbol{w} = (w_1, w_2, ..., w_N)$, respectively, then—via crossover manipulation—the related children can be represented as

$$v_c[j] = \begin{cases} v[j], & j \leq k \\ w[j], & j > k, \end{cases} \tag{7}$$

$$w_c[j] = \begin{cases} w[j], & j \leq k \\ v[j], & j > k, \end{cases} \tag{8}$$

where k is the crossover point, and j is an integer between 1 and N. Suppose an individual's chromosome is represented by $\boldsymbol{v} = (v_1, v_2, ..., v_N)$, and that the mutation probability is p_m, then the mutated chromosome v' can be created as follows:

$$v'_j = \begin{cases} v_j, & \text{with probability } 1 - p_m \\ \text{random value}, & \text{with probability } p_m, \end{cases} \tag{9}$$

where j is an integer between 1 and N, and each gene has an equal probability of being mutated. In this paper, to simplify these operations, the individuals are randomly paired for a single-point crossover during the crossover process.

3. Fitness assessment and selection: A fitness evaluation is the only basis for the GA to update the population. The population's fitness value is calculated by the fitness function, and the next generation is reproduced based on the fitness probability. An individual i's fitness score is computed by

$$f_c^{(i)} = 1 / (1 + \sum_{j=1}^{k_1+k_2} s_j) = 1 / (1 + \sum_{j=1}^{k_1+k_2} \boldsymbol{H}_j \cdot \hat{\boldsymbol{y}}^T), \tag{10}$$

where $\boldsymbol{H}_j$ is the j-th row of the overall parity-check matrix of the bilayer LDPC code, and $\hat{\boldsymbol{y}}$ is the hard decision result of the bilayer LDPC decoder (which corresponds to the received signal $\boldsymbol{y}$). In the population update process, the fitness value of the population gradually increases as $\sum s_i$ decreases, and the probability $P^{(i)}$ of individual i being selected is obtained by the roulette wheel selection of Equation (11), which is provided in the form of the 'selective' function in Algorithm 1. When the decoding is correct, the individual has the highest probability of being selected, and this individual is the optimal in the perturbation decoding algorithm at that time. The following Algorithm 1 shows the details of the perturbed decoding algorithm:

$$P^{(i)} = \frac{f_c^{(i)}}{\sum_{j=1}^{M} f_c^{(j)}}, \tag{11}$$

where M is the number of individuals in the population.

Algorithm 1: Perturbed decoding algorithm.

Input: $\boldsymbol{y}$ // The received signal of the upper page
$\boldsymbol{H}$ // Parity-check matrix of the bilayer LDPC code
M // Population size
T // Maximum number of iterations
$\boldsymbol{n_g}$ // Genetic noise
$\boldsymbol{P_g}$ // Population
p_c // Crossover probability
p_m // Mutation probability
$\boldsymbol{r_e}$ //Extra parity-check bits from the lower page
$\boldsymbol{y}'$ //Perturbed bits
Output: $\hat{\boldsymbol{x}}$ // Estimated code word
Initialization: $\hat{\boldsymbol{x}} \leftarrow 0$, $t \leftarrow 0$
$\hat{\boldsymbol{x}} \leftarrow$ Bilayer LDPC_decoder$(\boldsymbol{y}, \boldsymbol{r}_e, \boldsymbol{H})$
Compute syndrome $\boldsymbol{s}_b$ based on Equation (6)
if $\boldsymbol{s}_b == \boldsymbol{0}$
    break
else
    Initial population $\boldsymbol{P}_g^{(0)} = \{\boldsymbol{n}_g^{(i)} | i = 1, 2, \ldots, M\}$
    while $t < T$
        for i=1 : 2 : M-1
            $(\boldsymbol{n}_g^{(i)}, \boldsymbol{n}_g^{(i+1)}) = crossover(\boldsymbol{n}_g^{(i)}, \boldsymbol{n}_g^{(i+1)}, p_c)$
        end for
        for i=1 : M
            $\boldsymbol{P}_g^{(t)} = mutation(\boldsymbol{P}_g^{(t-1)}, p_m)$
        end for
        for i=1 : M
            $\boldsymbol{y}' = \boldsymbol{y} + \boldsymbol{n}_g^{(i)}$
            $\hat{\boldsymbol{x}} \leftarrow$ Bilayer LDPC_decoder$(\boldsymbol{y}', \boldsymbol{r}_e, \boldsymbol{H})$
            Compute syndrome $\boldsymbol{s}_b$ based on Equation (6)
            if $\boldsymbol{s}_b == \boldsymbol{0}$
                break
            end if
        end for
        Compute the individual fitness $\boldsymbol{f}_c^{(t)}$ based on Equation (10)
        $\boldsymbol{P}_g^{(t)} = select(\boldsymbol{P}_g^{(t-1)})$
        $t \leftarrow t + 1$
    end while
end if
Return $\hat{\boldsymbol{x}}$

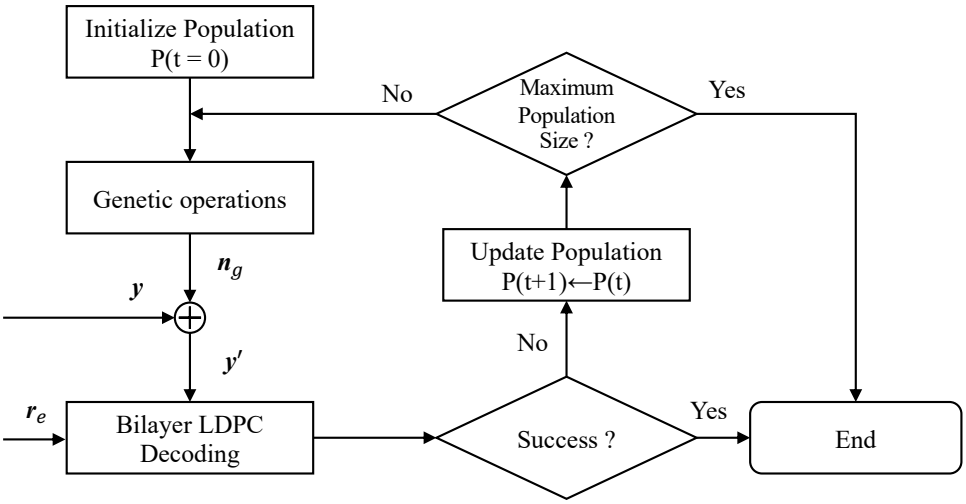


Figure 5. Block diagram of the perturbed decoding algorithm for a bilayer LDPC code.

4. Simulation Results

The BER and FER performances of the bilayer LDPC coding scheme in an MLC NAND flash memory channel are evaluated in this section, where the P/E cycles directly reflect the channel model. For more details on the MLC NAND flash memory channel models, please see [4]. Unless otherwise mentioned, the performances given in this section are the experimental results on the upper page of an MLC. The parameters related to the encoding and decoding algorithms of the simulation are shown in Table 1.

Table 1. Simulation parameters of the bilayer LDPC codes.

Parameters	Values
Code word length N	1404
Length of extra parity-check bits k_2	0/117/311
Code rate R	0.75/0.83
Maximum number of BP iterations I_{max}	20

4.1. Results of the Bilayer LDPC codes

Figures 6 and 7 illustrate the BER and FER performances of the proposed rate–0.75 bilayer LDPC codes with different coding schemes over different P/E cycles, respectively. Scheme I is decoded with only the upper LDPC code, where $k_2 = 0$ means no information on the extra parity-check bits (which is read from the lower page). Schemes II and III are decoded with bilayer LDPC codes, where $k_2 = 117$ and $k_2 = 311$ extra parity-check bits from the lower page, respectively. In addition, the lower page uses a separate LDPC code scheme with the same code length and code rate as the one used in the upper page. The extra parity-check bits received from the lower page are the decoding result of the lower page. In our simulations, the information bit length in the lower page was adjusted to obtain the number of extra parity-check bits, such that the code rate remained the same as that of the upper page.

Figure 6 shows that the proposed scheme outperformed the single LDPC code in Scheme I, where only the upper parity-check matrix $\boldsymbol{H}_u$ was employed. Compared with Scheme I, the proposed bilayer LDPC codes in Schemes II and III extend the lifetime of MLC NAND flash memory by more than 7000 and 8000 P/E cycles at a BER of 10^{-4}, respectively. Compared with Scheme I, the BER performance of the bilayer LDPC codes proposed in Schemes II and III can be enhanced by more than three and two orders of magnitude when P/E = 20,000, respectively.

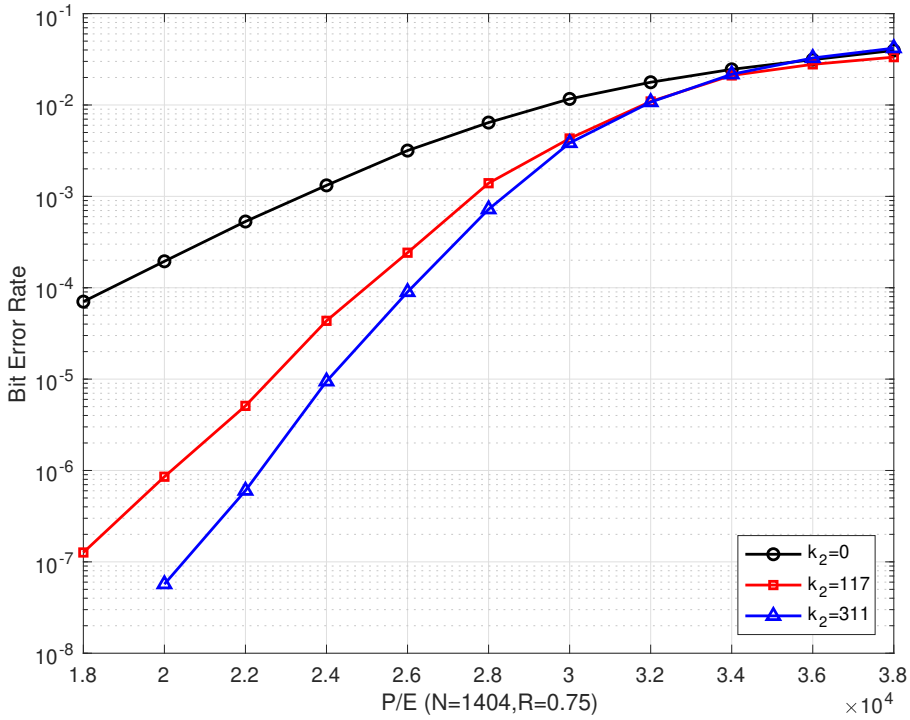


Figure 6. The BER performances of the proposed bilayer LDPC codes with different extra parity-check bits over different P/E cycles (where $N = 1404$ and $R = 0.75$).

In addition, the error correction performance of the bilayer LDPC codes is significantly enhanced as the number of extra parity-check bits increases. At a BER of 10^{-6}, the bilayer LDPC codes proposed in Scheme II were able to extend the lifetime of the MLC NAND flash memory by more than 2000 P/E cycles when the number of extra parity-check bits was increased from $k_2 = 117$ to $k_2 = 311$. However, the bilayer LDPC codes suffered from a performance degradation as the P/E cycles increased. When the P/E was $\geq$ 34,000, the error correction performances of the three coding schemes were almost the same. This indicated that, in the final period of flash memory, even the enhanced coding scheme cannot improve its reliability.

Figure 7 shows that our proposed scheme outperforms Scheme I. This is consistent with the results in Figure 6. When P/E was $\geq$ 28,000, the error correction performance of Scheme II was only slightly better than that of Scheme III. Even with increasing the number of extra parity-check bits, the improvement in the error correction performance was limited.

Figure 8 shows the comparison of the average number of iterations of the proposed schemes with different coding schemes that were implemented over different P/E cycles when $R = 0.75$. When P/E was < 28,000, the average number of iterations of the proposed coding scheme was slightly higher than that of a single LDPC coding scheme, but it also brought a considerable gain in error correction performance. The reason for this was that the bilayer LDPC decoding algorithm was activated only after the upper page decoding failed. As the P/E cycle number increased, the channel condition of the upper page gradually became worse, and the number of bilayer LDPC decoding increased, which led to an increase in the number of iterations. In this situation, the proposed bilayer coding scheme could not improve the performance.

We also performed simulations on the rate–0.83 bilayer LDPC code, which is one of the proposed bilayer LDPC coding schemes described in Section 3, to further evaluate its error correction performance. The BER and FER performances of the different schemes are shown in Figures 9 and 10, respectively. As the code rate increased, the proposed schemes outperform Scheme I in terms of error correction performance. To further confirm

the importance of the extra parity-check bits from the lower page, we also present the performance of Scheme IV (green line) and Scheme V (purple line) in Figures 9 and 10, respectively, for the ideal case (which assumes no errors in the extra parity-check bits).

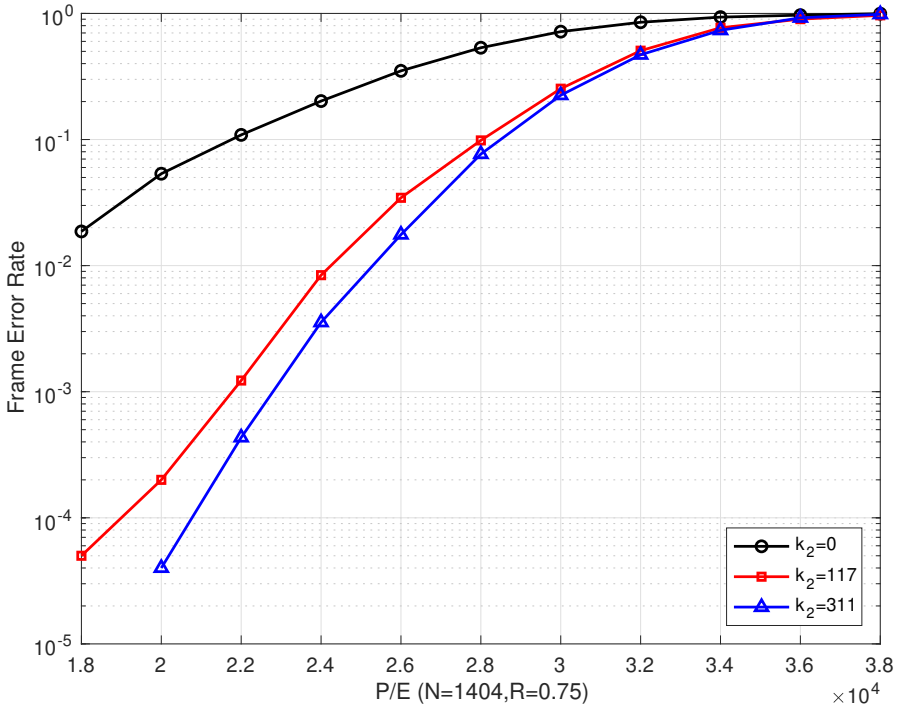


Figure 7. The FER performances of the proposed bilayer LDPC codes with different extra parity-check bits over different P/E cycles ($N = 1404$ and $R = 0.75$).

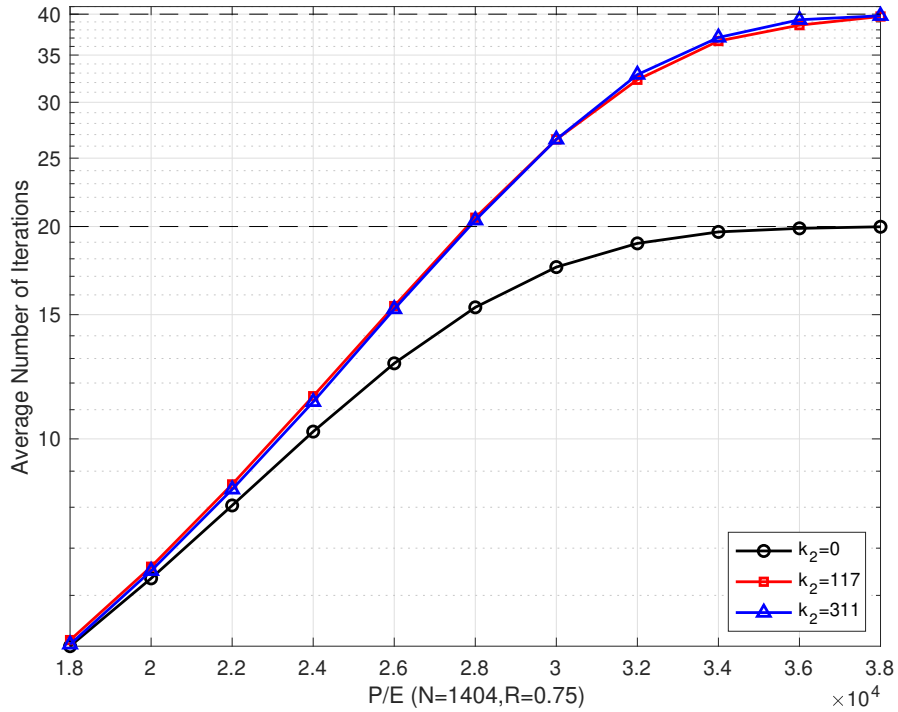


Figure 8. The average number of iterations of the proposed bilayer LDPC codes with different extra parity-check bits over different P/E cycles ($N = 1404$ and $R = 0.75$).

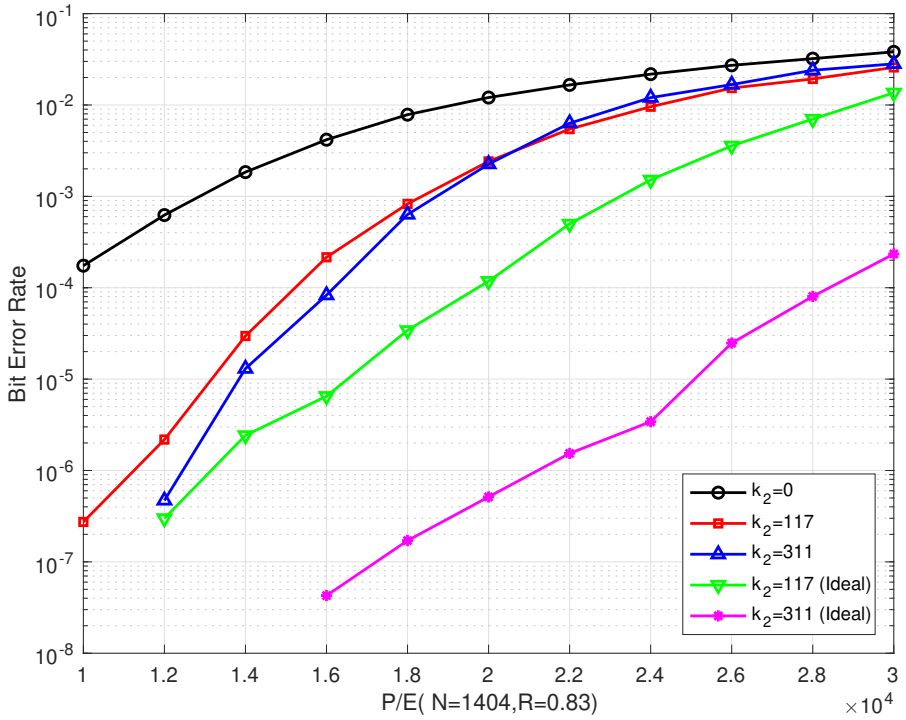


Figure 9. The BER performances of the proposed bilayer LDPC codes with different extra parity-check bits over different P/E cycles ($N = 1404$ and $R = 0.83$).

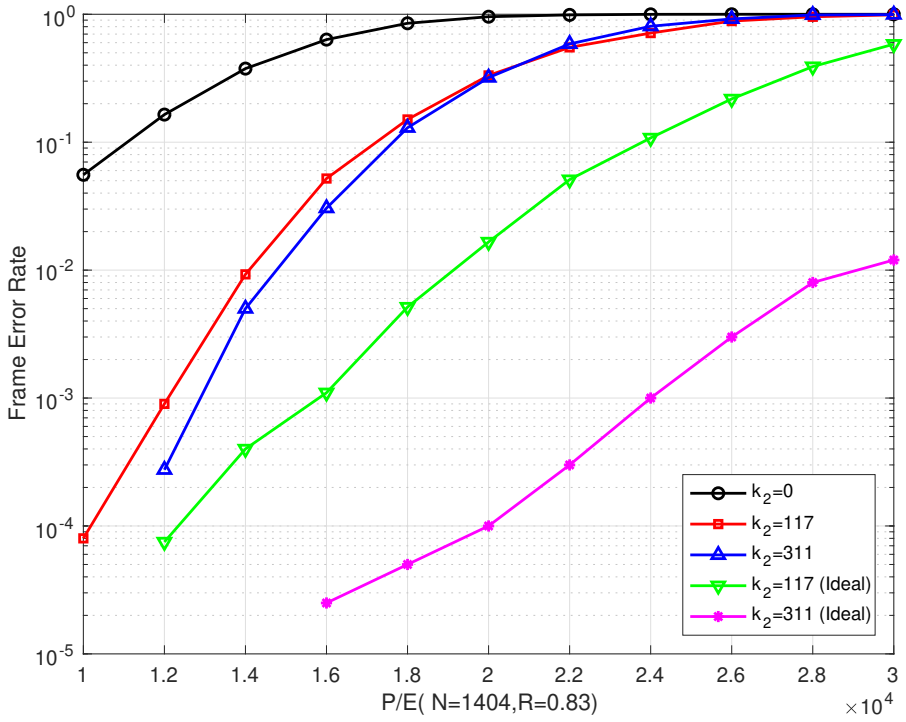


Figure 10. The FER performances of the proposed bilayer LDPC codes with different extra parity-check bits over different P/E cycles ($N = 1404$ and $R = 0.83$).

We can also see from Figures 9 and 10 that the proposed bilayer LDPC codes of Scheme IV and Scheme V can enhance the BER or FER performance by more than two and three orders of magnitude, respectively, compared with that of Scheme I when $P/E = 12{,}000$. Compared with Scheme II and Scheme III, the error correction performance of Scheme IV

and Scheme V in the ideal case was extremely excellent, even in high P/E cycle regions. Compared with Schemes II and III, the BER or FER performance of the proposed bilayer LDPC codes of Schemes IV and V could be improved by more than one and three orders of magnitude, respectively, when $P/E = 16,000$.

From Figure 9, we know that the increase in the number of extra parity-check bits may not necessarily lead to a performance improvement with increasing the number of P/E cycles when $R = 0.83$. This is because the wrong extra parity-check bits can cause an error propagation in the decoding process. In addition, the error propagation intensity may be reinforced with an increase in the number of parity-check bits. In this case, the extra parity-check bits become unnecessary since the decoding delay is increased but the error correction performance is not improved.

Figure 11 shows the comparison of the average number of iterations for several schemes in both ideal and non-ideal cases. We can see from the figure that the average number of iterations of the proposed several coding schemes is almost the same as that of the single LDPC coding scheme in low P/E cycle regions. However, as the P/E cycle number grew, the average number of iterations was higher than that of a single LDPC coding scheme. The average number of iterations of Scheme IV and Scheme V in the ideal case was lower than that in the non-ideal case. This confirms the fact that the errors in the extra parity-check bits can reduce the convergence speed and increase the decoding delay of the proposed method.

In addition, it can also be observed from these figures that the error correction performance improvement of the proposed scheme decreases as the code rate increases. This is because the error correction performance of the upper page becomes worse, and the errors in the extra parity-check bits exacerbate the deterioration.

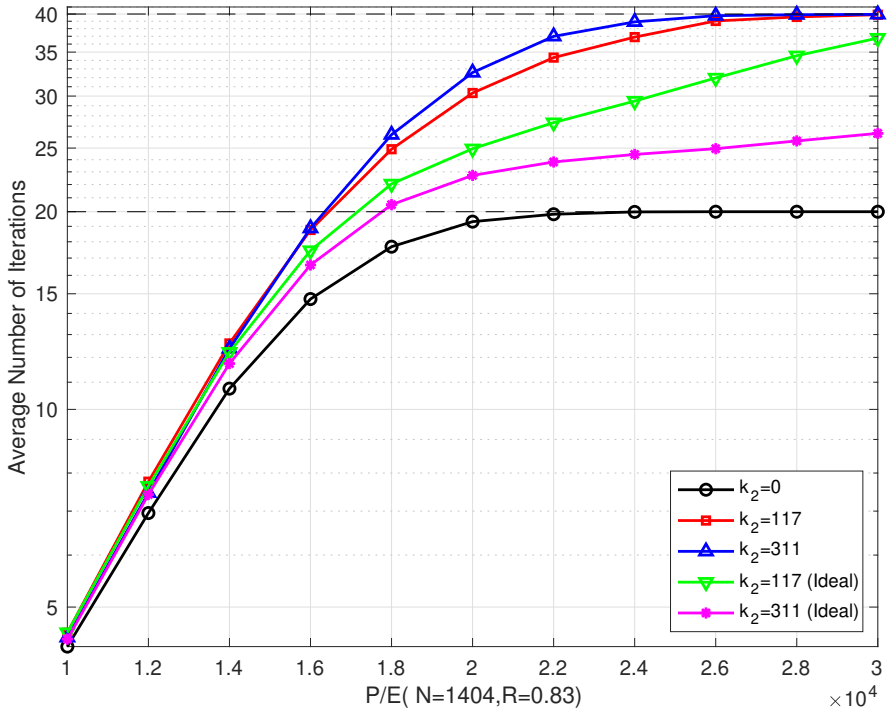


Figure 11. The average number of iterations of the proposed bilayer LDPC codes with different extra parity-check bits over different P/E cycles ($N = 1404$ and $R = 0.83$).

4.2. Perturbed Decoding Algorithm Results

To evaluate the impact of the perturbed decoding algorithm on the performance of the proposed bilayer LDPC codes in MLC flash memory channels, computer simulations were performed. In the GA process, we set the crossover probability p_c and the mutation probability p_m to 0.8 and 0.05, respectively. The number M of the individuals in the

population was set to 10. The number T of generations of the population updates was set to 5, and the standard deviation of the perturbed noise δ was set to 0.25.

From Figure 12, we can see that the perturbed decoding algorithm outperformed other algorithms for our proposed bilayer LDPC codes. Compared with the conventional single LDPC decoding algorithm and the proposed bilayer LDPC decoding algorithm, the proposed perturbed decoding algorithm outperformed them by about 12,000 P/E cycles and 6000 P/E cycles, respectively, at a BER of 10^{-4}. When P/E = 26,000, we can see from the figure that our proposed bilayer LDPC codes with perturbed decoding algorithm can achieve a significant improvement in the BER performance, which was found to be more than three orders of magnitude better than the conventional single LDPC decoding algorithm and two orders of magnitude better than the proposed bilayer LDPC decoding algorithm.

From Figure 13, we can draw the same conclusion that our proposed perturbed decoding algorithm for bilayer LDPC codes performs better than other algorithms. In addition, compared with the proposed bilayer LDPC decoding algorithm, an excellent error correction performance was achieved in both low and high P/E cycle regions.

As shown in Figure 14, the proposed bilayer LDPC decoding algorithm and the proposed perturbed decoding algorithm had the same average number of iterations. In addition, the conventional single LDPC decoding algorithm had a slightly lower average number of iterations. However, as the P/E cycle number increased, the gap between the numbers of iterations of the first two algorithms and the conventional single LDPC decoding algorithm gradually widened, especially when the P/E was larger than 30,000 cycles. When combining the above simulation results, we can see that, in low P/E cycle regions, the proposed perturbed decoding algorithm provides a good trade off between the performance and complexity. In high P/E cycle regions, the proposed perturbed decoding algorithm was still effective. However, it cost a large number of decoding iterations, which greatly increased the decoding delay.

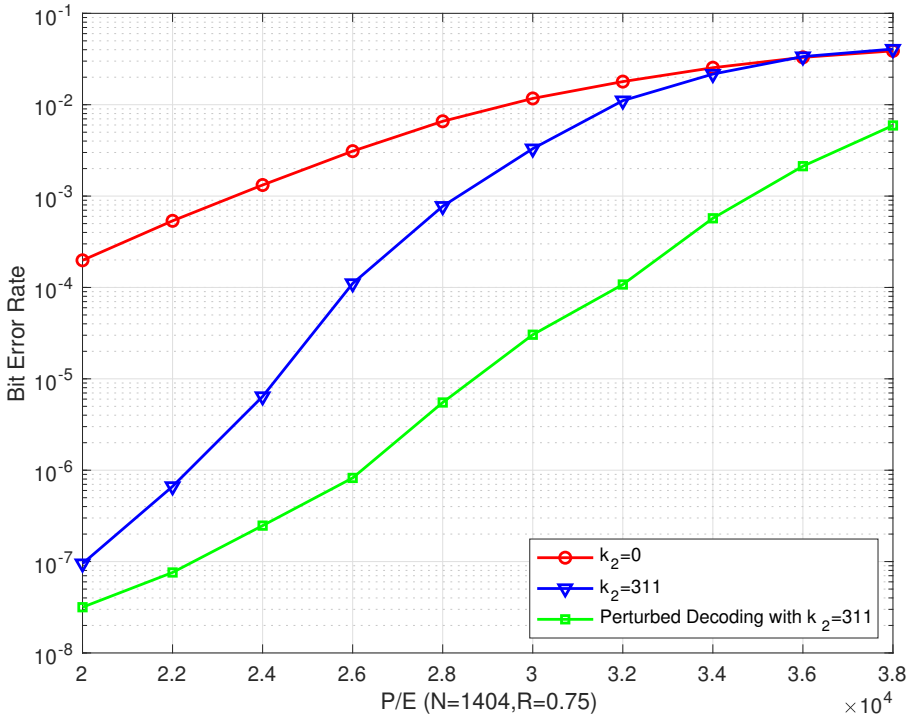


Figure 12. The BER performances of the proposed bilayer LDPC codes with perturbed decoding algorithm over different P/E cycles ($N = 1404$ and $R = 0.75$).

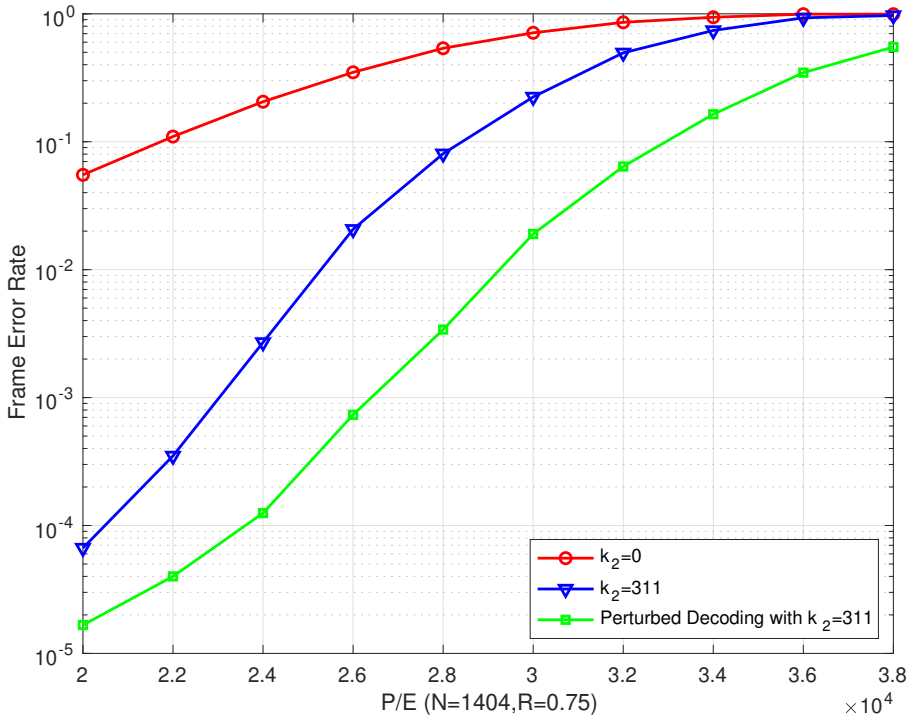


Figure 13. The FER performances of the proposed bilayer LDPC codes with perturbed decoding algorithm over different P/E cycles ($N = 1404$ and $R = 0.75$).

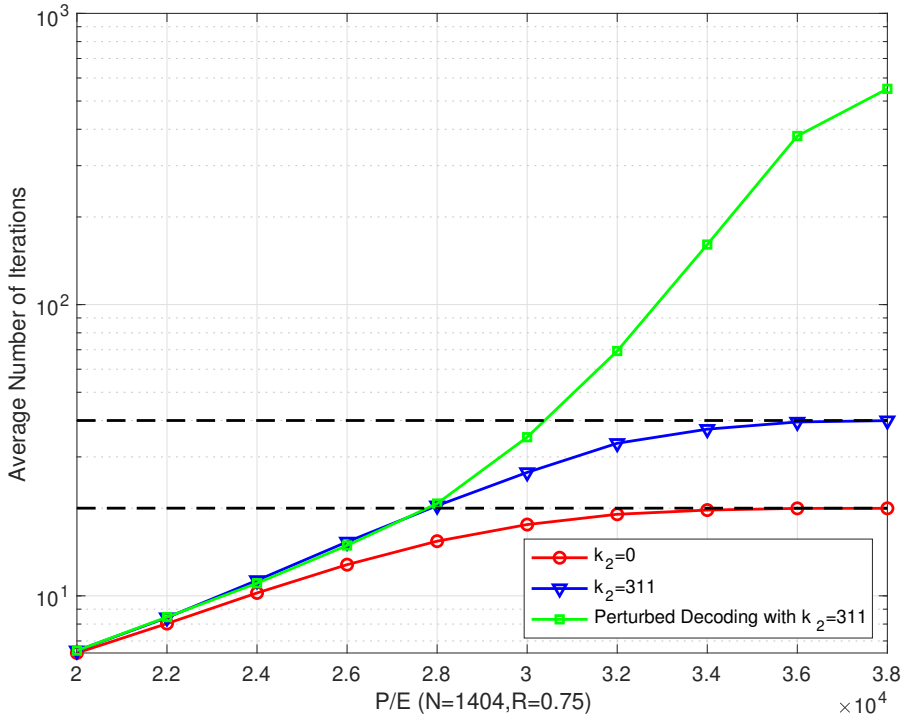


Figure 14. The average number of iterations of the proposed bilayer LDPC codes with a perturbed decoding algorithm over different P/E cycles ($N = 1404$ and $R = 0.75$).

5. Conclusions

In this paper, a coding scheme based on bilayer LDPC codes was proposed for MLC NAND flash channels by exploiting their intracell unbalanced bit error probabilities. When the conventional LDPC decoding of an upper page fails, the bilayer LDPC code is decoded with the extra parity-check bits of a lower page with better channel conditions, which

can improve the reliability of MLC NAND flash systems. Compared with the conventional LDPC codes, the proposed bilayer LDPC codes significantly improve the BER/FER performance by more than three/two orders of magnitude. As the number of the extra parity-check bits increases, the bilayer LDPC codes can achieve a significant enhancement in error correction performance. According to the performance requirement of NAND flash memory, we can adjust the number of the extra parity-check bits of our proposed bilayer LDPC codes. In low P/E cycle regions, the proposed bilayer LDPC codes provide a considerable improvement in error correction performance at the cost of a slightly higher average number of iterations than single LDPC codes. In addition, we applied the perturbation theory to the proposed bilayer LDPC codes, as well as proposed a perturbed decoding algorithm for MLC NAND flash channels that can achieve a desirable error correction performance. In the future, we will try to design cost-effective coding schemes for high P/E cycle regions in high-density flash memory. In addition, we will explore combining the guessing random additive noise decoding (GRAND) [32] with the GA to take advantage of the asymmetric nature of the flash channels to improve decoding performance and reduce latency.

Author Contributions: Conceptualization, L.K. and H.L.; methodology, L.K. and H.L.; software, W.H. and L.K.; validation, W.H. and L.K.; formal analysis, W.H.; investigation, W.H. and C.M.; resources, W.H.; data curation, W.H.; writing—original draft preparation, L.K.; writing—review and editing, H.L.; visualization, W.H. and C.M.; supervision, C.M.; project administration, L.K. and H.L.; funding acquisition, L.K. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Project of Basic Science (Natural Science) Research in Higher Education Institutions of Jiangsu Province (grant no. 22KJA510009), the NSFC (grant no. 62271482), the JITSF (grant no. jit-b-202110), the Qing Lan Project in Jiangsu Province, and the Open Research Fund of National Mobile Communications ResearchLaboratory, Southeast University (No.2020D18).

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Cai, Y.; Ghose, S.; Haratsch, E.F.; Luo, Y.; Mutlu, O. Error characterization, mitigation, and recovery in flash-memory-based solid-state drives. *Proc. IEEE* **2017**, *105*, 1666–1704. [CrossRef]
2. Aritome, S.; Shirota, R.; Hemink, G.; Endoh, T.; Masuoka, F. Reliability issues of flash memory cells. *Proc. IEEE* **1993**, *81*, 776–788. [CrossRef]
3. Huang, M.; Liu, Z.; Qiao, L.; Wang, Y.; Shao, Z. An endurance-aware metadata allocation strategy for MLC NAND flash memory storage systems. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.* **2016**, *35*, 691–694. [CrossRef]
4. Aslam, C.A.; Guan, Y.L.; Cai, K. Read and write voltage signal optimization for multi-level-cell (MLC) NAND flash memory. *IEEE Trans. Commun.* **2016**, *64*, 1613–1623. [CrossRef]
5. Yu, X.; He, J.; Li, Q.; Zhang, B.; Wang, X.; Wang, Q.; Huo, Z. DMMC: A polar code construction method for improving performance in TLC nand flash. *IEEE Embed. Syst. Lett.* **2023**, *early access*. [CrossRef]
6. Hareedy, A.; Lanka, C.; Dolecek, L. A general non-binary LDPC code optimization framework suitable for dense flash memory and magnetic storage.*IEEE J. Sel. Areas Commun.* **2016**, *34*, 2402–2415. [CrossRef]
7. Vakilinia, K.; Divsalar, D.; Wesel, R.D. Optimized degree distributions for binary and non-binary LDPC codes in flash memory. In Proceedings of the 2014 International Symposium on Information Theory and Its Applications, Melbourne, Australia, 26–29 October 2014; pp. 6–10.
8. Kabatiansky, G.; Kruglik, S. On codes correcting constant number of errors in l_1 metric. In Proceedings of the Information Technology and Systems (ITaS), Sochi, Russia, 7–11 September 2015; pp. 152–157. [CrossRef]
9. Jiang, A.; Li, H.; Wang, Y. Error scrubbing codes for flash memories. In Proceedings of the 11th Canadian Workshop on Information Theory, Ottawa, ON, Canada, 13–15 May 2009; pp. 32–35. [CrossRef]

10. Sun, H.; Zhao, W.; Lv, M.; Dong, G.; Zheng, N.; Zhang, T. Exploiting intra-cell bit error characteristics to improve min-sum LDPC decoding for MLC NAND Flash based storage in mobile device. *IEEE Trans. Very Large Scale Int. Sys.* **2016**, *24*, 2654–2664. [CrossRef]
11. Hu, Y.; Song, S.; Xiao, S.; Xu, Q.; Xiao, N.; Qin, Z. A dominating error region strategy for improving the bit-flipping LDPC decoder of SSDs. *IEEE Trans. Circ. Sys. II Express Briefs* **2015**, *62*, 578–582. [CrossRef]
12. Kong, L.; Liu, Y.; Liu, H.; Zhao, S. Protograph QC-LDPC and rate-adaptive polar codes design for MLC NAND flash memories. *IEEE Access* **2019**, *7*, 37131–37140. [CrossRef]
13. Kay, S. Can detectability be improved by adding noise? *IEEE Signal Process. Lett.* **2000**, *7*, 8–10. [CrossRef]
14. Sutera, A. Stochastic perturbation of a pure connective motion. *J. Atmos. Sci.* **2010**, *37*, 245–249. [CrossRef]
15. Xiao, D.; Gu, Z. Dynamic perturbation decoding of polar-CRC cascaded code. In Proceedings of the IEEE International Wireless Communications, and Mobile Computing, Limassol, Cyprus, 5–19 June 2020; pp. 42–45.
16. Lee, H.; Kil, Y.S.; Jang, M.; Kim, S.H.; Park, O.S.; Park, G. Multi-round belief propagation decoding with impulsive perturbation for short LDPC codes. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1491–1494. [CrossRef]
17. Arli, A.; Gazi, O. Noise-aided belief propagation list decoding of polar codes. *IEEE Commun. Lett.* **2019**, *23*, 1285–1288. [CrossRef]
18. Gerrar, N.K.; Zhao, S.; Kong, L. A CRC-aided perturbed decoding of polar codes. In Proceedings of the 14th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2018), Chongqing, China, 18–20 September 2018. [CrossRef]
19. Kong, L.; Liu, H.; Hou, W.; Dai, B. Improving decodability of polar Codes by adding noise. *Symmetry* **2022**, *14*, 1156. [CrossRef]
20. Kong, L.; Kim, K.J.; Kwak, K.S. Design of bilayer QC-LDPC codes for decode-and forward based cooperative relaying communication. In Proceedings of the IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 4717–4721. [CrossRef]
21. Ram, E.; Cassuto, Y. Design of bilayer and multi-layer LDPC ensembles from Individual degree distributions. *IEEE Trans. Inf. Theory* **2021**, *67*, 7096–7109. [CrossRef]
22. Fang, Y.; Bi, G.; Guan, Y.L.; Lau, F.C.M. A survey on protograph LDPC codes and their applications. *IEEE Commun. Surv. Tuts.* **2015**, *17*, 1989–2016. [CrossRef]
23. Benzi, R.; Sutera, A.; Vulpiani, A. The mechanism of stochastic resonance. *J. Phys. A Math. Gen.* **1981**, *14*, L453–L457. [CrossRef]
24. McDonnell, M.D.; Abbott, D. What is stochastic resonance? definitions, misconceptions, debates, and its relevance to biology. *PLoS Comput. Biol.* **2009**, *5*, e1000348. [PubMed] [CrossRef]
25. Chen, H.; Varshney, P.K.; Michels, J.H.; Kay, S. Approaching near optimal detection performance via stochastic resonance. In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP), Toulouse, France, 14–19 May 2006; p. III. [CrossRef]
26. Shih, K.; Shiu, D. Perturbed decoding algorithm for concatenated error correcting and detecting codes system. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Helsinki, Finland, 11–14 September 2006; pp. 1–5.
27. Wang, W.; Shiu, D. The theory behind perturbed decoding algorithm. In Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Athens, Greece, 3–7 September 2007; pp. 1–5.
28. Nishikawa, M.; Nakamura, Y.; Kanai, Y.; Osawa, H.; Okamoto, Y. A study on iterative decoding with LLR modulator by neural network using adjacent track information in SMR system. *IEEE Trans. Magn.* **2019**, *55*, 1–5. [CrossRef]
29. Petrović, V.L.; Marković, M.M.; El Mezeni, D.M.; Saranovac, L.V.; Radošević, A. Flexible high throughput QC-LDPC decoder with perfect pipeline conflicts resolution and efficient hardware utilization. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 5454–5467. [CrossRef]
30. Wang, X.; Ma, Q.; Li, J.; Zhang, H.; Xu, W. An improved SC flip decoding algorithm of polar codes based on genetic algorithm. *IEEE Access* **2020**, *8*, 222572–222583. [CrossRef]
31. Elkelesh, A.; Ebada, M.; Cammerer, S.; Brink, S.T. Decoder-tailored polar code design using the genetic algorithm. *IEEE Trans. Commun.* **2019**, *67*, 4521–4534. [CrossRef]
32. Duffy, K.R.; An, W.; Médard, M. Ordered reliability bits guessing random additive noise decoding. *IEEE Trans. Signal Process.* **2022**, *70*, 4528–4542. [CrossRef]

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.