REAL: A Retention Error Aware LDPC Decoding Scheme to Improve NAND Flash Read Performance

Meng Zhang, Fei Wu*
Wuhan National Laboratory for
Optoelectronics, Huazhong University
of Science and Technology, China
{zgmeng, wufei}@hust.edu.cn

Xubin He, Ping Huang
Department of Electrical and
Computer Engineering,Virginia
Commonwealth University, USA
{xhe2, phuang}@vcu.edu

Shunzhuo Wang, Changsheng Xie
Wuhan National Laboratory for
Optoelectronics, Huazhong University
of Science and Technology, China
{wangshunzhuo, cs_xie}@hust.edu.cn

Abstract—**Continuous technology scaling makes NAND flash cells much denser. As a result, NAND flash is becoming more prone to various interference errors. Due to the hardware circuit design mechanisms of NAND flash, retention errors have been recognized as the most dominant errors, which affect the data reliability and flash lifetime. Furthermore, after experiencing a large number of program/erase (P/E) cycles, flash memory would suffer a much higher error rate, rendering traditional ECC codes (typically BCH codes) insufficient to ensure data reliability. Therefore, low density parity check (LDPC) codes with stronger error correction capability are used in NAND flash-based storage devices. However, directly using LDPC codes with belief propagation (BP) decoding algorithm introduces non-trivial overhead of decoding latency and hence significantly degrades the read performance of NAND flash. It has been observed that flash retention errors show the so-called numerical-correlation characteristic (i.e., the 0-1 bits stored in the flash cell affect each other with the leakage of the charge) in each flash cell. In this paper, motivated by the observed characteristic, we propose REAL: a retention error aware LDPC decoding scheme to improve NAND flash read performance. The developed REAL scheme incorporates the numerical-correlation characteristic of retention errors into the process of LDPC decoding, and leverages the characteristic as additional bits decision information to improve its error correction capabilities and decrease the decoding latency. Our simulation results show that the proposed REAL scheme can reduce the LDPC decoding latency by 26.44% and 33.05%, compared with the Logarithm Domain Min-Sum (LD-MS) and Probability Domain BP (PD-BP) schemes, respectively.**

I. Introduction

NAND flash with the advantages of high performance, large storage capacity, low power consumption, shock resistance, and non-volatility is widely used in various computer systems and consumer electronics products as a storage device (e.g., cellphones, digital cameras, MP3 players, and solid-state disks (SSDs) [8] [9]). As a result of aggressive feature size reduction, each cell is capable of storing more bits information (i.e., 2 bits and 3 bits per cell with multi-level cell (MLC) and triple-level cell (TLC) respectively), which lowers the cost per bit and increases the storage capacity. However, the data reliability in NAND flash cells decreases [6] as each flash cell stores fewer electrons and the voltage gap between adjacent storage states shrinks, causing various interference errors, including retention errors, write interference errors, read interference errors and erase errors [2]. Retention errors, i.e., errors caused by charge leakage over time after flash cells are programmed, have been found to be the dominant flash errors [1] [2] [14]. Therefore, many prior researches have been devoted to alleviate retention errors to improve flash reliability and endurance. One prior solution [12] deploys a data scrubbing technique which periodically refreshes data by reading flash pages and then programming them to new blank pages. While alleviating retention errors, the data refreshing strategy brings about a cost of reducing the performance and lifetime of NAND flash as it causes additional read and write operations. Another solution [7] proposes implementing error correction codes (ECC) in the flash controller to improve the fault tolerance capability. BCH codes [24] [25] are used to guarantee data reliability and have been applied to today's commercial storage systems based on NAND flash [3]. However, flash technology scaling has caused NAND flash cells to become much denser. Especially, when technology node reaches 20nm or even smaller, various interference noises further increase the raw bit error rate (RBER), making NAND flash memory suffer from worse raw storage reliability [6] [13]. Furthermore, after flash cells experience a certain number of P/E cycles, the retention error rate becomes increasingly higher, as a result, BCH codes with stronger error correction capability have to be deployed to ensure data reliability. Unfortunately, such BCH codes require larger extra space to store the encoding redundancy and involve higher implementation complexity for encoding and decoding. Therefore, the flash scaling may render an approach that adopts advanced BCH codes to guarantee reliability at a prohibitively high cost [44]. Moreover, BCH codes are becoming insufficient to cope with the growing RBER [37]. It is obvious that ECC schemes with stronger error correction capability are needed to meet the requirements of uncorrectable bit error rate (UBER) (typically 10^{-15}). The low density parity check (LDPC) codes [19] [20] with promising error correction performance and recent success in commercial hard disk drives (HDD) have attracted much attention and are considered as one choice of ECC for future NAND flash storage systems (e.g., solid state disks (SSDs)) [26] [27] [28].

LDPC codes were first proposed by Gallager in 1962 [19]. One attractive feature of LDPC codes is their highly sparse parity-check matrix with much more "0" than "1" elements, so the benefit is that the constructed sparse parity-check can

*Corresponding author: Fei Wu; Email: wufei@hust.edu.cn reduce the encoding complexity of LDPC codes, and thus decrease the encoding latency, as fewer nodes are involved in the encoding/decoding computations. Similar to BCH codes, LDPC is also a kind of block code. However, they have striking differences. BCH codes are a type of pure algebraic codes, and their encoding and decoding are dependent on the relevant algebraic theory. By contrast, the LDPC decoding is a process of reliability information propagation, and it is some soft decision information for bits reliability (i.e., probability information or logarithm likelihood ratio information) [5] [18]. Moreover, the error correction capability is determined by the accuracy of storage detection (i.e., the quality of the input information) and the number of decoding cycles. If the storage detection is more accurate and the iteration numbers are larger, the error correction capability would be stronger. However, directly using LDPC codes can lead to an increase in decoding latency in NAND flash [5] due to high decoding complexity. To effectively apply LDPC codes, it requires to design an effective method to minimize the decoding overheads. It has been observed that retention errors of NAND flash exhibit the numerical-correlation characteristic in each storage cell [2]. In this paper, inspired by the observed characteristic, we propose REAL: a retention error aware LDPC decoding scheme to improve NAND flash read performance. REAL takes the numeric-correlation characteristic into account when performing LDPC decoding. The incorporation of the characteristic as additional decoding decision information increases the reliability of bits decision and therefore leads to decreased decoding latency and improved NAND flash read performance.

The major contributions of this paper are as follows:

- We analyze the fundamental causes of retention errors in flash and develop a mathematical model to explain the retention errors caused by cell threshold voltage shift. We also analyze how LDPC decoding latency affects the NAND flash read performance.
- Inspired by the numerical-correlation characteristic of retention errors, we propose REAL: a retention error aware LDPC decoding scheme to improve NAND flash read performance. REAL fully utilizes the characteristic by adequately incorporating it into the LDPC decoding process. The numeric-correlation observation is used as extra decoding decision information (i.e., inherent information contained in the property) to reduce the LDPC decoding latency and improve the read performance of NAND flash.
- In order to make good use of the numerical-correlation characteristic of retention errors, we propose a novel codeword layout such that a codeword contains two bits from the same MLC NAND flash cell, which is different from the traditional scheme where two bits from the same cell are assigned to different codewords. The proposed layout enables REAL to take advantage of the characteristic for reducing decoding overheads.

The remainder of this paper is organized as follows. Section II discusses some background of NAND flash and LDPC codes. Section III analyzes the retention error characteristic and read performance of NAND flash. The proposed LDPC codeword layout and the REAL sheme are described in section IV. The simulation results are given in Section V. Finally, we summarize the related work in Section VI and conclude our paper in Section VII.

II. Background and Motivation

In this section, we introduce the background knowledge about fundamental NAND flash and its operations, the main concept of LDPC codes, and the procedures of LDPC encoding and decoding. We then analyze the performance of LDPC decoding and present our motivation of this work.

A. The Basics of NAND Flash

A NAND flash cell is a floating gate of metal oxide semiconductor field effect transistor (MOSFET) which stores electrons and is surrounded by insulation materials [16]. The floating gate is governed by a control gate to which a certain voltage is applied when data is written into the cell. The insulation material between the floating gate and the control gate is called dielectric and its function is to prevent electrons from leaking out of the gate. The material between the transistor channel and the floating gate is named oxide layer that can preserve injected electrons. Due to the effect of strong electric field, electrons are injected into the floating gate through the oxide layer. These insulation materials ensure that flash cells have the non-volatility property. A floating gate stores data by holding a certain amount of electrons. Read, write, and erase are the basic operations of NAND flash. NAND flash is programmed and read in page granularity and erased in the unit of blocks [29].

- ***Read:*** The stored data can be read by detecting the threshold voltage of a NAND flash cell [16]. When a selected bias voltage is applied to the control gate, it produces an electric field that turns on the transistors, and consequently there appears conduction current on the bit lines. According to the transistors' conduction state, the stored data value can be determined. If a certain number of electrons are stored in the floating gate, an internal electric field is formed and it incurs resistance to the transistors conduction (i.e., more electrons, larger resistance and more difficult to turn on the transistors).
- ***Write:*** The write operation of NAND flash is based on the Fowler-Nordheim (FN) electron tunneling mechanism [35]. When we apply a high voltage to the control gate, the electrons are to move at a high speed in the channel. In the meantime, due to the effect of strong electric field, a part of electrons move through the oxide layer to the floating gate. When the electrons accumulate to a certain amount, the corresponding threshold voltage is reached and a value of "0" is stored.
- ***Erase:*** The erase operation of NAND flash is also based on the Fowler-Nordheim (FN) electron tunneling mechanism [35]. When we apply a high voltage to the silicon base level of all cells, the electrons are removed from

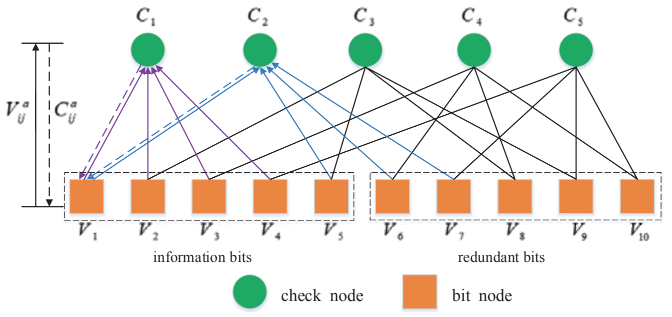

Fig. 1. The check matrix H that is used for encoding and decoding can be transformed to a Tanner Graph (TG) [23], whose purpose is to facilitate the decoding information transfer process. The check nodes correspond to the rows of the check matrix H, and the bit nodes correspond to the columns of the check matrix H. If the element at the ith row and the jth column is nonzero, the ith check node and the jth bit node are connected.

the floating gates to the silicon base level because of the effect of strong electric field. The threshold voltage of floating gate is restored to the initial state, and the state of cells is changed from the programmed (state "0") to the erased (state "1").

B. LDPC Codes

1) LDPC codes: LDPC codes are linear block codes, which are only determined by the check matrix H that directly affects the performance of LDPC codes [18]. Each row of the matrix H represents a check equation E, and each column corresponds to a codeword bit. The amount of "1" in the matrix is much fewer than the amount of "0". The number of "1" in each row, and each column is called row weight λ and column weight ρ, respectively. If λ and ρ are fixed, the LDPC codes are called "regular LDPC codes". If λ and ρ are not fixed, the LDPC codes are named "irregular LDPC codes" [19]. The regular LDPC codes are denoted as (n, λ, ρ). Moreover, the check matrix H can be represented by a Tanner Graph (TG) [41]. The relationship between the check matrix H and the TG is one-to-one mapping. For example, if the check matrix is

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}. \tag{1}$$

Its corresponding TG is shown in Fig. 1. It demonstrates that the TG is composed of check nodes and bit nodes which are connected by edges. If the element at the ith row and jth column $(H_{i,j})$ is nonzero, the ith check node and the jth bit node are connected via an edge, meaning the jth bit node is involved in the computation in the ith equation.

2) LDPC encoding and decoding:

- ***encoding:*** LDPC encoding follows two approaches, general encoding and fast encoding based on the lower triangular matrix. The fast approach with special structure has much lower complexity than the general method [30].
- ***decoding:*** LDPC decoding is based on the BP algorithm [31], which is a kind of soft decision decoding method including two forms of logarithm and probability domains, such as Min-Sum decoding algorithm [36]. BP decoding is a process for reliability information exchange, where such information passes through the edges of the TG (shown in Fig. 1), with reliability degree accumulated for bits decision. In Fig. 1, $V_1, V_2, \cdots, V_j, \cdots, V_n$ represent n bit nodes $(1 < j < n = 10)$, $C_1, C_2, \cdots, C_i, \cdots, C_m$ denote m check nodes $(1 < i < m = 5)$. $C^a_{i,j}$ represents the probability that the check equation C_i is satisfied when the bit node V_j equals to a ("0" or "1"). $V^a_{i,j}$ denotes the probability that V_j equals to a ("0" or "1") when other check equations connected to bit node V_j are satisfied, excluding the check equation C_i (i.e., $C^a_{ij} = P(the\ check\ equation\ C_i\ is\ satisfied \mid V_j = a)$, $V^a_{ij} = P(V_j = a|\ other\ check\ equations\ excluding\ C_i\ connecting\ to\ V_j\ are\ satisfied,\ a = 0\ or\ 1)$.

3) LDPC codes workflow: We illustrate the workflow of LDPC codes with a concrete example that can help readers understand the working principles of LDPC codes. We use the traditional encoding method for encoding and choose the conventional Logarithm Domain Min-Sum (LD-MS) algorithm [43] for decoding. The check matrix H mentioned above is used to construct the LDPC codes and the corresponding TG (shown Fig. 1) is adopted to explain the decoding process of LDPC codes. The symbols used for LDPC encoding and decoding are explained in Table I and in Table II.

TABLE I
THE SYMBOLS USED IN LDPC ENCODING

H	check matrix
$I_{r\times r}$	identity matrix
n	codeword length
r	redundant bits length
$P_{r\times(n-r)}$	general binary matrix
$n-r$	information bits length
$\overrightarrow{A} = (A_0, A_1, \cdots A_{n-1})$: n codeword bits	
$\overrightarrow{B_1} = (A_0, A_1, \cdots A_{n-r-1})$: $n-r$ information bits	
$\overrightarrow{B_2} = (A_{n-r}, A_{n-r+2}, \cdots A_{n-1})$: r redundant bits	

TABLE II
THE SYMBOLS USED IN LDPC DECODING

k	iterations
σ	standard deviation
T_j	initial information
F_{max}	maximum iterations
j	bit nodes $(1 \le j \le n)$
i	check nodes $(1 \le i \le r)$
$Q(i)\backslash j$	the set of $Q(i)$ excluding the bit node V_j
$R(j)\backslash i$	the set of $R(j)$ excluding the check node C_i
$Q(i)$	the set of the bit node V_j connected to the check node C_i
$R(j)$	the set of the check node C_i connected to the bit node V_j
V_{ij}	the information of the bit node V_j connected to the check node C_i
C_{ij}	the information of the check node C_i connected to the bit node V_j

- ***encoding:*** LDPC encoding is the process of generating redundant bits using the check matrix H. We perform a linear transformation on the check matrix H.

$$H \rightarrow \left[P_{r\times(n-r)}, I_{r\times r}\right], \vec{A} = \left[\vec{B_1}, \vec{B_2}\right]$$

The codeword $\vec{A}$ satisfies

$$\vec{A} \cdot H^T = \vec{0}, \left[\vec{B_1}, \vec{B_2}\right] \cdot \left[P_{r\times(n-r)}, I_{r\times r}\right]^T = \vec{0}.$$

We have

$$\vec{B_2} = \vec{B_1} \cdot P^T_{r\times(n-r)}.$$

The r redundant bits are obtained through the above equations.

- ***decoding:*** The conventional LD-MS decoding is described in Algorithm 1, which is the process of loop iterations. The belief information (i.e., probability information or logarithm likelihood ratio information) is passed through the edges of the TG (see Fig. 1) when the interfered codewords are decoded. Here we use an example to illustrate the process of information transmission during decoding. Let's denote a received codeword as $\vec{V} = (V_1, V_2, \cdots, V_9, V_{10})$. When decoding, the codeword $\vec{V}$ needs to satisfy the following equation:

$$H.\vec{V}^T =$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} V_1 \\ V_2 \\ \vdots \\ V_9 \\ V_{10} \end{bmatrix} = \vec{0}$$

Using the matrix multiplication operation, we have

$$\begin{cases} V_1 + V_2 + V_3 + V_4 = 0 \quad (C_1) \\ V_1 + V_5 + V_6 + V_7 = 0 \quad (C_2) \\ V_2 + V_5 + V_8 + V_9 = 0 \quad (C_3) \\ V_3 + V_6 + V_8 + V_{10} = 0 \quad (C_4) \\ V_4 + V_7 + V_9 + V_{10} = 0 \quad (C_5) \end{cases} \tag{2}$$

where $C_1, C_2, \cdots, C_5$ represent 5 check equations which are called check nodes represented by the circles in the TG (see Fig. 1), and $V_1, V_2, \cdots, V_{10}$ are the codeword bits which are called bit nodes denoted by squares in the TG (see Fig. 1). The bit node V_1 is involved in the check equations C_1 and C_2 (see Equation (2)), so there are two edges connecting V_1 to C_1 and C_2, respectively. In a similar manner, we can draw the other connecting edges in Fig. 1. During decoding, information renewal is realized following two-round steps. In the first round, all check nodes information is updated. In the second round, all bit nodes information is updated. Let's take a look on how check nodes C_1 and C_2, and bit nodes V_1 and V_2 are updated. As said, check nodes are updated in the first round and bit nodes are updated in the second round. To update C_1, all bit nodes connected to C_1, excluding V_1, i.e., V_2, V_3, V_4, provide information. Similarly, V_5, V_6, and V_7 provide information to update C_2. All other check nodes are updated in a similar fashion. After finishing the first round of updating all check nodes, it continues the second round of updating bit nodes. Each bit node is updated using the information provided by its connected check nodes which have been updated in the first round. V_1 is updated with information from both C_1 and C_2. The updated V_1 information will then be used to update check nodes in the next decoding iteration. At the end of LDPC decoding, it determines the values of the bit nodes using the updated information.

Algorithm 1 The LD-MS decoding process

```
Input: The interfered noise channel soft information $\vec{X} = (x_1, x_2, \cdots, x_n)$
Output: The decoded codewords $\vec{V} = (V_1, V_2, \cdots, V_n)$
 1: Initialize $T_j = 2x_j/\sigma^2$ , $C^k_{ij} = 0, V^k_{ij} = T_j, k = 0$, $i \in R(j), j \in Q(i) \, (1 \le i \le 5, 1 \le j \le 10)$ and set the max number of iterations as $F_{max}$.
 2: Update check nodes information
      $C^k_{ij} = \prod_{m \in Q(i)\backslash j} \mathrm{sgn}(V^k_{im}) \cdot \min\left\{\left|V^k_{im}\right| : m \in Q(i)\backslash j\right\}$
 3: Renew bit nodes information
      $V^k_{ij} = T_j + \sum_{h \in R(j)\backslash i} C^k_{hj}$
 4: Make decoding decisions
      $V^k_j = T_j + \sum_{i \in R(j)} C^k_{ij}$
 5: if $V^k_j \ge 0$ then
 6:     $a_j = 0$
 7: else
 8:     $a_j = 1$
 9: end if
10: if $\vec{V} \cdot H^T = \vec{0}$ or $k = F_{max}$ then
11:     break
12: else
13:     for $k$ from 1 to $F_{max}$ do
14:         repeat the process from the step 2 to the step 13 until the step 10 is satisfied.
15:     end for
16: end if
```

C. LDPC Decoding Performance Analysis

The speed of LDPC decoding affects the read performance of NAND flash. Fast decoding algorithms can effectively reduce the response time of storage systems and enhance the overall read efficiency [3] [40]. The traditional probability domain BP algorithm [38] is a sum product decoding algorithm, which involves a large number of complex multiplication operations, increasing the LDPC decoding delay and the system response time. The LDPC decoding performance is mainly influenced by the following factors.

(a) The accuracy of information propagation between the bit nodes and the check nodes, i.e., reliability information.

The LDPC decoding is the process of reliability information (i.e., probability information or logarithm likelihood ratio information) propagation. Each bit node has a reliability information, which is updated iteratively by information transmission between bit nodes and check nodes. Finally, the decoder makes the best decision based on the updated bit nodes reliability information. If the propagated reliability information between the bit nodes and the check nodes is more accurate, the decoder can make the decision in a faster way.

(b) Decoding iterations.

The purpose of the decoding iterations is to accumulate bits decision reliability. If the number of decoding iteration is larger, the reliability of bits decision is higher. However, this increases the decoding overhead and degrades the NAND flash read performance.

(c) Additional information provided.

Providing additional decision information to the decoding process, e.g., the numerical-correlation of retention errors, can speed up the process of accumulating bits decision reliability and thus reduce the decoding latency. The additional information can decrease iterations in the LDPC decoding process. Note that no additional information is available in the traditional LDPC decoding process.

(d) The codeword length.

Long codeword LDPC codes can improve the performance of error correction. However, using a long codeword length affects the decoding performance. Furthermore, it makes the decoding process more complex and requires a large number of decoding logic units, which not only introduces storage overhead, but also increases the decoding delay and the response time of NAND flash storage systems. We should choose a suitable length of LDPC codes to achieve a balance between decoding complexity and error correction performance.

(e) The impacts of retention errors, i.e., electrons leakage speed.

The speed and quantity of electrons leakage can affect the decoding performance of LDPC codes. When the speed and number of electrons leakage reach a threshold, the numerical-correlation of retention errors becomes more apparent, which can provide the additional decision information with high precision and enable the decoding to converge fast.

D. Motivation

As the NAND flash density continues to increase, which brings about deteriorating noise tolerance of NAND flash cells, the information reliability has been decreasing rapidly. Besides, retention errors are identified as the most dominant errors in NAND flash devices [1] [12] [14], which impact data reliability and flash lifetime. Furthermore, when the flash cells experience a certain number of P/E cycles and the retention error rate becomes higher, BCH codes become inadequate to ensure data reliability and have to be replaced by stronger ECC schemes [3]. Based on these considerations, we are motivated to utilize LDPC codes with optimized error correction performance to guarantee data reliability. However, simply using the LDPC codes without optimizations inevitably increases the decoding latency and thus lowers the NAND flash read efficiency. We notice that the LDPC decoding performance can be significantly improved if flash error characteristics are taken into account during the decoding process. In particular, it has been observed that retention errors of NAND flash exhibit the characteristic of numerical-correlation in every cell. The characteristic can be beneficial to developing effective ECC techniques. We propose to leverage the feature in the LDPC decoding process in order to reduce the decoding latency and improve the read performance of NAND flash, which is discussed in depth in section IV.

III. Retention Errors in NAND Flash

In this section, we analyze the causes of bit errors and the retention error characteristic. We summarize our observations based on the analysis and also provide the read performance analysis of NAND flash.

A. Bit Error Analysis

NAND flash, based on the number of bits stored in each cell, is divided into three types [13]: single-level cell (SLC), multiple level cell (MLC) and triple level cell (TLC). The SLC flash stores 1 bit information per cell and has two data states (i.e., “1” or “0”) by using two different threshold voltages to represent the stored data. There are four logic states (i.e., “11”, “10”, “01”, “00”) in an MLC cell that stores 2 bits information, which adopts four preseted different ranges of threshold voltage levels to represent the data. Similarly, the TLC cell stores 3 bits data by drawing into eight threshold voltage levels, representing eight storage states.

The employment of MLC and TLC improves the storage capacity and lowers the price per bit. However, the MLC and the TLC storing more bits in each cell have more threshold voltage states. As a result, the amount of charges stored in each cell is diminished and the gap between two adjacent voltage thresholds becomes smaller. As a consequence, the cells are more sensitive to voltage variations. Variations can easily lead to threshold voltage shifts, which causes voltage window overlapping. When reading the cell data, it is difficult to determine which threshold voltage window the applied reference voltage falls into, in which case errors (shown in Fig. 2) occur. Especially, as flash memory process technology scales to smaller feature size, the inter-cell interference grows dramatically. As a result, noise resistance ability is greatly compromised. In addition, it is more likely for bit flip to occur, causing higher RBER. Meanwhile, owing to inevitable process variability, the threshold voltage exhibits normal distribution [9]. Hence, we model the threshold-voltage distribution of the cell state as

$$p\left(x\right)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^{2}}{2\sigma^{2}}} \tag{3}$$

where μ and σ are the mean and standard deviation of the cell threshold voltage [4], respectively.

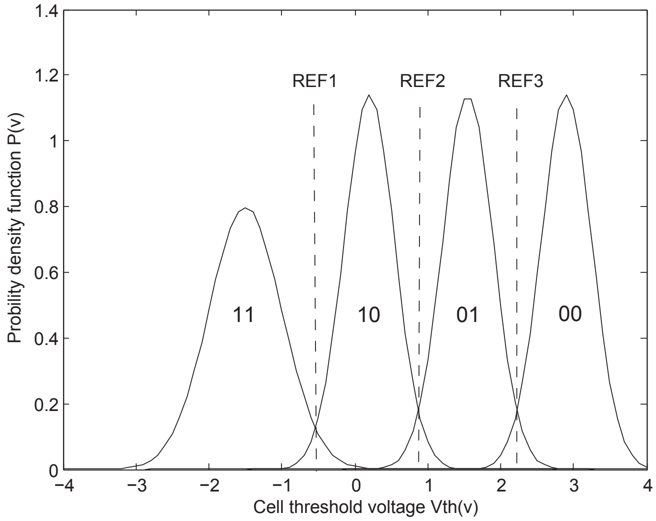

Fig. 2. Threshold voltage overlap [4]. NAND flash cells are subject to various interferences that lead to the threshold voltage overlap. When the reference voltage falls into the overlapping area, we are not able to judge the accuracy of the data that has been read. The overlapping part is the obscure area that we cannot accurately detect the stored bits.

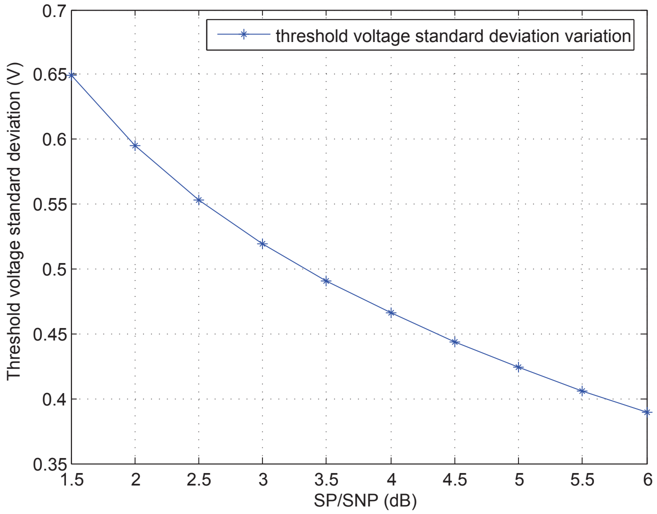

Fig. 3. Threshold voltage standard deviation of the cell variation with SP/SNP. The threshold voltage of the NAND flash cell that exhibits a trend of fluctuation is affected by the retention noise. The threshold voltage varies with SP/SNP.

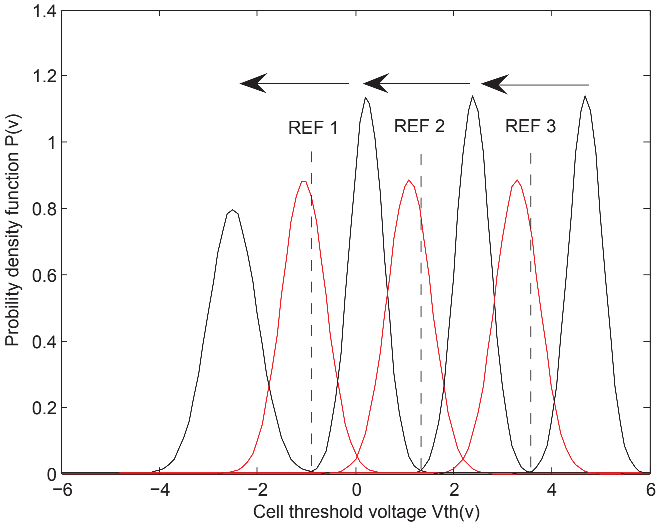

Fig. 4. The threshold voltage shift (red lines) associated with retention errors. With the leakage of the flash cells charges, the threshold voltage is not stable and shows fluctuations, i.e., the threshold voltage mean is reduced and standard deviation is increased.

B. Retention Error Characteristic Analysis

1) Mathematical analysis: Retention errors are caused by the electrons stored in the floating gate leaking gradually [12] [15] through the leakage current, which leads to the cells threshold voltage instability so that the threshold voltage standard deviation changes with SP/SNP (i.e., SP/SNP represents the signal power-to-retention noise power ratio) (shown in Fig. 3). As a consequence, two adjacent threshold voltage windows may become overlapped (shown in Fig. 4) and the boundaries are blurred. It should be noted that the “11” (i.e., an erasure state without electrons injected) voltage window would not shift as there are no electrons in the floating gate in this state. As stated above, the threshold voltage of NAND flash cell follows normal distribution [9]. We denote the mean and standard deviation of the threshold voltage before retention errors occur as μ_k and σ_k, the mean and standard deviation after retention errors appear as μ_i and σ_i. We define $\lambda_1 = \alpha_1 \left|\mu_k - \mu_i\right|$ and $\rho_1 = \beta_1 \left|\sigma_k - \sigma_i\right|$ as transfer factor and instability factor of the threshold voltage, respectively, where α_1 and β_1 are constants and less than 1, k and i are the cell state. If λ_1 and ρ_1 are larger and $\lambda_1 < \eta_1$, $\rho_1 < \omega_1$, where η_1 and ω_1 are the pre-defined thresholds, it demonstrates that the cell dissipates more electrons, which incurs retention errors.

2) Numerical-correlation of retention errors: Retention errors feature the characteristic of numerical-correlation, where the most common retention errors are “00”→“01”, “01”→“10”, “01”→“11” and “10”→“11”, with their relative percentages over all retention errors being 46%, 44%, 5%, and 2%, respectively. The relative percentages among various error transitions are almost constant for different P/E cycles [2]. The numerical-correlation characteristic of the retention errors is very important and can affect the decoding performance of LDPC codes, which can be used to obtain some useful observations that are beneficial to the LDPC decoding.

C. Our Observations

We take MLC NAND flash memory as an example to give our observations. Fig. 5 reveals the bits mapping to the different threshold voltage levels and the relative electron quantities. The left bit called the least significant bit (LSB) and the right bit named the most significant bit (MSB) are associated with the lower page and the upper page, respectively. We have the following several valuable observations based on the numerical-correlation characteristic of retention errors.

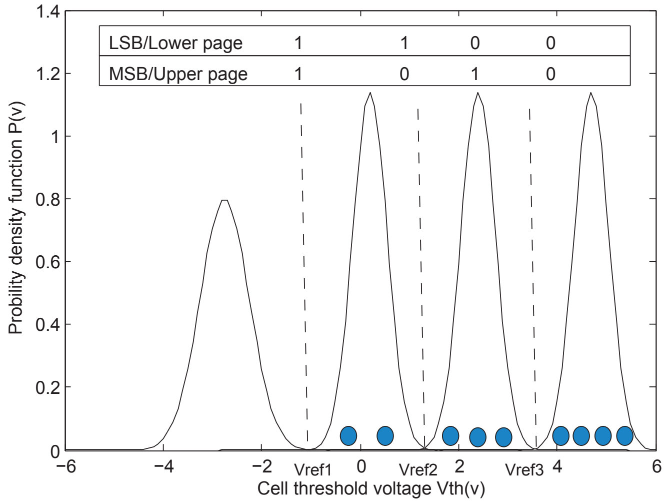

Fig. 5. Mapping information bits to voltage levels preseted in MLC NAND flash. The different bits information corresponds to the different threshold voltage states. The "11" represents the erased state without electrons trapped. The number of electrons injected from the "10" to the "00" states gradually increases. The circles denote the injected electrons.

1) ***Observation 1:*** When the data of a detected lower page is located in the fuzzy area between two adjacent states, we cannot determine whether the data is "1" or "0". Owing to the impact of retention errors, the data value of the original upper page bit has a very large probability of being "1".
2) ***Observation 2:*** When the data of a detected upper page resides in the fuzzy area between two adjacent states, the value of the original lower page bit has a very large probability of being "0", because the proportion of "00" → "01" conversion is more likely to happen than other transitions.
3) ***Observation 3:*** When the data of a detected upper page is "0", due to the effect of retention errors, the data value of the original lower page bit has a very high probability of being "0".
4) ***Observation 4:*** When the data of a detected upper page is "1", the data value of the original lower page bit has a very high probability of being correct.
5) ***Observation 5:*** When the data of a detected lower page is "0", due to the influence of retention errors, the data value of the original upper page bit has a very high probability of being "0", as it is easier for the state "00" to suffer charge leakage than the state "01".
6) ***Observation 6:*** When the data of a detected lower page is "1", due to the effect of retention errors, the data value of the original upper page bit has a very high probability of being "1", for a similar reason as above.

The observations overview is shown in Fig. 6, in which ① ②③ ④ ⑤ ⑥ represent the above 6 observations that are beneficial to the error correction of LDPC codes, as they provide additional decision regarding the original codeword bits reliability.

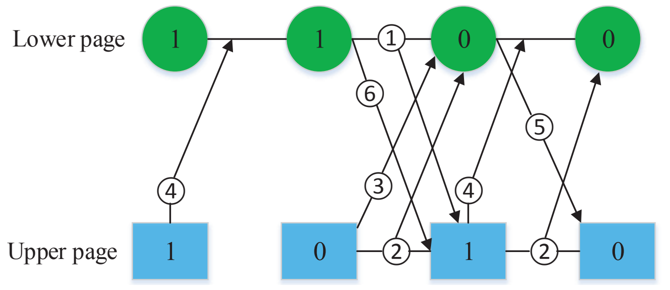

Fig. 6. Observation overview. Symbols ① ② ③ ④ ⑤ ⑥ correspond to the 6 observations respectively, which can be applied to the LDPC decoding process in order to reduce the decoding latency and improve the read performance of NAND flash.

D. Read Performance Analysis of NAND Flash

The LDPC decoding latency can affect the overall read performance of NAND flash, which is modelled as

$$
\begin{aligned}
T^1_{ORL} &= T^1_{FPTC} + T^1_{LDDL} + T^1_{ITDB}\\
T^2_{ORL} &= T^2_{FPTC} + T^2_{LDDL} + T^2_{ITDB}\\
\eta &= \frac{T^1_{ORL} - T^2_{ORL}}{T^1_{ORL}} \times 100\% =\\
&\frac{T^1_{LDDL} - T^2_{LDDL}}{T^1_{FPTC} + T^1_{LDDL} + T^1_{ITDB}} \times 100\% =\\
&\frac{(T^1_{LDDL} - T^2_{LDDL})/T^1_{LDDL}}{T^1_{FPTC}/T^1_{LDDL} + T^1_{ITDB}/T^1_{LDDL} + 1} \times 100\%.
\end{aligned}
$$

Finally, we have

$$
\eta = \frac{\varphi}{T^1_{FPTC}/T^1_{LDDL} + T^1_{ITDB}/T^1_{LDDL} + 1} \times 100\%, \qquad (4)
$$

where

$$
\varphi = (T^1_{LDDL} - T^2_{LDDL})/T^1_{LDDL}
$$

which is the ratio of decoding latency to be reduced. T^1_{ORL} is the overall read latency when the conventional LDPC decoding methods are adopted, T^2_{ORL} is the whole read latency of the proposed REAL scheme, and η is the performance improvement. T^1_{FPTC} and T^2_{FPTC} represent the latencies of reading data from a flash array page to the LDPC decoder and are assumed to be constant. T^1_{ITDB} and T^2_{ITDB} are the latencies of transferring the decoded codeword information from the LDPC decoder to the I/O data buffer, and are also assumed to be constant. That is

$$
T^1_{FPTC} = T^2_{FPTC}, T^1_{ITDB} = T^2_{ITDB}.
$$

T^1_{LDDL} is the decoding latency with the conventional LDPC decoding approaches, while T^2_{LDDL} is the decoding latency with the developed REAL scheme. In particular, T^1_{LDDL} and T^2_{LDDL} are the main parts of the overall read delays, which take up a lot of time and influence the promotion of NAND flash read performance. In formula (4), the values of T^1_{FPTC}/T^1_{LDDL} and T^1_{ITDB}/T^1_{LDDL} are very small, because T^1_{FPTC} and T^1_{ITDB} are very small (we refer to Micron's technical manual on 16, 32, 64, 128Gb NAND Flash Memory [39]), and T^{1}_{LDDL} is large. The percentage of increased decoding performance is close to the improvement of the overall read performance. The process of reading data is shown in Fig. 7. When data is read from the NAND flash, the read latency includes three components: 1) the time taken to transfer data from the NAND flash array pages to the LDPC decoder, 2) the time to perform LDPC decoding, and 3) the time of transferring decoded codeword information from the LDPC decoder to the I/O data buffer.

IV. The Proposed REAL Scheme

In this section, we describe the proposed REAL scheme that takes advantage the numerical-correlation characteristic of retention errors to reduce the LDPC decoding latency. To effectively leverage this feature, we suggest a novel layout of LDPC codewords in which a codeword contains two bits data from the same MLC NAND flash cell. It is different from traditional schemes where two bits belong to different codewords. The suggested layout offers the opportunity to leverage the numerical-correlation characteristic of retention errors for the proposed REAL scheme.

A. A Novel Layout of LDPC Codewords

Conventionally, each MLC NAND flash cell stores two bits that belong to different codewords. Typically, the data bits in the lower page are arranged in one codeword, and the data bits located in the upper page are arranged in another codeword. There is no connection between the two bits located in the same NAND flash cell at the time of decoding. However, the two bits in the same cell have the characteristic of mutual influence with the leakage of electrons, which is called the numerical-correlation characteristic of retention errors mentioned above. This characteristic can facilitate decoding by providing additional bits decision information, which can quickly correct the codeword sequence that contains the error bits. However, the traditional decoding scheme does not leverage the numerical-correlation characteristic of retention errors. We try to incorporate this characteristic into the LDPC decoding process as part of the decoding information propagation. Therefore, we propose a novel layout of LDPC codewords to make a codeword contain two bits data from the same MLC NAND flash cell. The original encoded codeword sequence is cut from the middle part and is stored in different logical page locations (shown in Fig. 8). The advantage of the codeword layout is that we can make full use of the mutual influence characteristic (i.e., the numerical-correlation characteristic of retention errors) of two bits data located in the same cell at the time of decoding. Moreover, the proposed layout provides the chance and environment that can effectively utilize the characteristic for the proposed REAL scheme and fully show the potential value of numerical-correlation characteristic. Likewise, when decoding, the method can offer some additional decoding information to increase the bits decision reliability, which can decrease the decoding iterations and reduce the decoding latency.

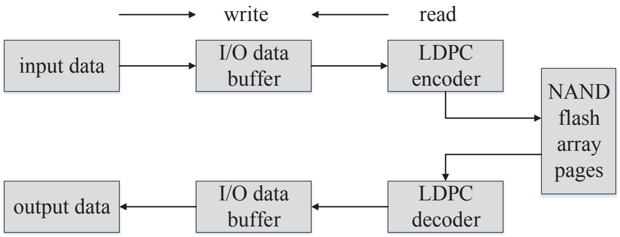

Fig. 7. Read and write data using LDPC.

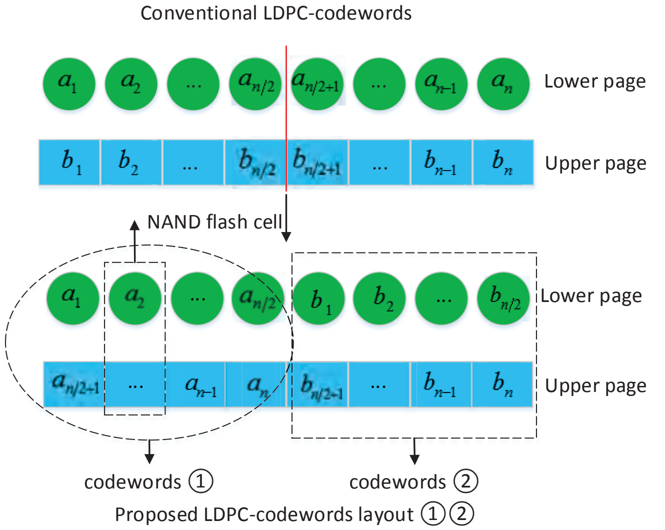

Fig. 8. Our proposed LDPC codeword layout. A LDPC codeword contains two bits data from the same MLC NAND flash cell, which is different from the conventional arrangement scheme in which the two bits belong to different codewords. Our codeword layout can make full use of the numerical-correlation of retention errors to provide accurate bit decision information and reduce LDPC decoding latency.

B. Proposed REAL scheme

In this paper, we propose the REAL scheme, which utilizes the belief propagation process between the bit nodes and the check nodes to correct the error bits in the received codewords that are caused by various interference noises. Besides, the proposed scheme makes full use of the extra decoding decision information inside the bit nodes provided by obtained different valuable observations, which improves both the error correction strength and the decoding convergence speed. The main idea of the proposed scheme is to add the observed additional information into the LDPC decoding process to decrease the decoding iterations and improve the NAND flash read performance. However, the conventional LDPC schemes merely rely on the information propagation between the bit nodes and the check nodes without additional bits decision information added.

1) Information propagation process of the proposed scheme: The proposed REAL scheme can be described as follows. When the decoder receives a LDPC codeword, the reliability information of each bit node can be obtained. Likewise, based on the reliability of gained bit nodes, the reliability degree of the check nodes connected to the bit nodes is calculated. The computed reliability about the check nodes is used to update the reliability information of the bit nodes that are associated with these calculated check nodes. The reliability information between the two types of nodes is repeatedly iterated so that the interfered codeword bits are corrected fleetly. In fact, the information propagation procedure of the REAL scheme is to reckon the maximum posteriori probability P_k for each codeword bit V_i, which is the probability of the ith codeword bit $V_i = a$ ("0" or "1"), if all the check equations are established. Fig. 9 illustrates the information propagation and renewal process of the proposed REAL scheme. The set $R(v_j) = \{c_i : H_{i,j} = 1\}$ represents all the check nodes connected to the bit node v_j and the set $Q(c_i) = \{v_j : H_{i,j} = 1\}$ denotes all the bit nodes linked to the check node c_i. We denote $Q(c_i) \backslash v_j$ as the set of the $Q(c_i)$ excluding the bit node v_j and $R(v_j) \backslash c_i$ as the other set of $R(v_j)$ excluding C_i. When the stored LDPC codewords are interfered by the retention noise, owing to the effect of the device mechanisms of NAND flash, there is initial probability information P_i and I_i about the decoding decision that are alternatively stored in the lower page and the upper page and also called storage prior information (SPI). In the decoding process of the REAL scheme, firstly, when we compute the maximum posteriori probability P_k (i.e., the correct probability of the bit node v_j) for the bit node v_j, all the check nodes connected to the bit node provide the correct probability for the v_j. Then the decoder based on the offered probability makes the best hard decision HD. Likewise, the decoder tests whether all the hard decision bits satisfy the check equation $CH^T = 0$ or not. If not, the bit node v_j and the check node c_i should be updated. In the phase of updating the bit node v_j, the corresponding bit node v_k provides additional decision information E_j that the bit node v_j is correct because of the numerical-correlation characteristic of retention errors. When we calculate the best hard decision HD, the E_j is also considered as the extra decision information that helps to improve the estimation accuracy. The proposed scheme invokes the E_j which can realize accurate estimation for the bit node v_j. Therefore, the decoding latency is reduced and the NAND flash read performance is improved.

2) *Encoding and decoding procedure of the REAL scheme:* In the transversion of decoding, we consider logarithm likelihood ratio ($LLR(p) = \ln \frac{p(x=0|v_{th})}{p(x=1|v_{th})}$) information as the maximum posterior probability information to change the large number of multiplication and division operations into addition and subtraction, lessening the difficulty of computing and storage. If $LLR > 0$, it illustrates that the probability of the bit node $v_j = 0$ is larger. Otherwise, the bit node is more likely to be "1".

- ***Encoding Procedure:*** For LDPC encoding, we use the general fast encoding method. Let

$$H_{m\times n} = \left[H^1_{m\times(n-m)} | H^2_{m\times m}\right].$$

We use the Gauss elimination algorithm [42] to transform $H_{m\times n}$ into

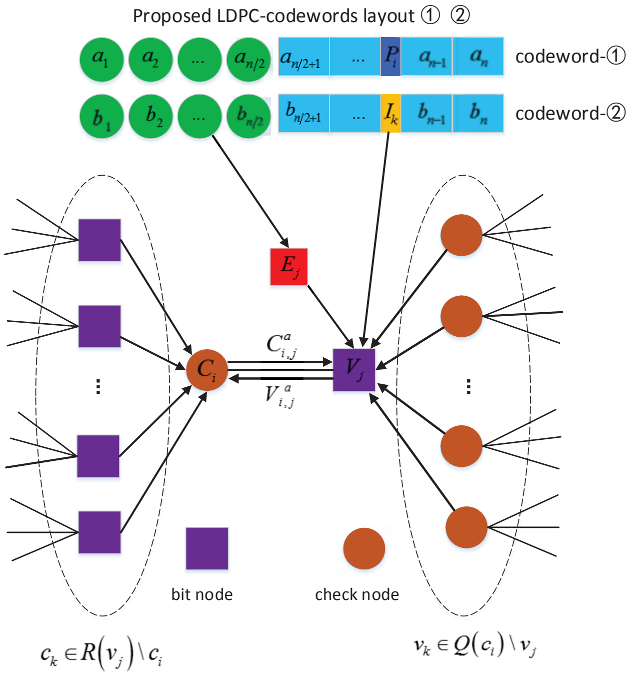

Fig. 9. Information propagation and renewal process of the proposed REAL scheme. When the bit node V_j is updated, the additional decoding information E_j that is provided by the another bit node belonging to the same MLC NAND flash cell due to the numerical-correlation characteristic of retention errors is added to the renewal process of the bit node V_j in order to improve the bit decision accuracy and reduce the LDPC decoding latency.

$$H_{m\times n} = \left[P_{m\times(n-m)} | R_{m\times m}\right],$$

where $R_{m\times m}$ is the identify matrix. We know that $H_{m\times n} \cdot C^T_{1\times n} = \vec{0}$. Let

$$C_{1\times n} = \left[I_{m\times(n-m)} | C_{m\times m}\right].$$

We have

$$\left[P_{m\times(n-m)} | R_{m\times m}\right] \cdot \left[I_{1\times(n-m)} | C_{1\times m}\right]^T = \vec{0}$$

$$C_{1\times m} = I_{1\times(n-m)} \cdot P^T_{m\times(n-m)},$$

where $I_{1\times(n-m)}$ and $C_{1\times m}$ are information bits and check bits, respectively.

- ***Decoding Procedure:***
  - *Step 1:* Initialize $V_{j,i}(0) = P_j$, set the maximum iterations as N_{max}, $0 \le l \le N_{\max}$, $1 \le i \le \frac{1}{2} N_{\max}$.
  - *Step 2:* Check nodes information updating.
    1) In the odd number ($l = 2i - 1$) of iterations, check nodes are updated in a incremental order.

$$C_{i,j}(l+1) = \alpha \prod_{v_k \in Q(c_i)\backslash v_j}^{v_k < v_j} \text{sign}(V_{i,k}(l+1)) \cdot$$

$$\prod_{v_k \in Q(c_i)\backslash v_j}^{v_k > v_j} \text{sign}(V_{i,k}(l)) \cdot \min$$

$$\left[\min_{v_k \in Q(c_i)\backslash v_j}^{v_k<v_j} |V_{i,k}(l+1)|, \min_{v_k \in Q(c_i)\backslash v_j}^{v_k>v_j} |V_{i,k}(l)|\right]$$

2) In the even number $(l = 2i)$ of iterations, check nodes are updated in a descending order.

$$C_{i,j}(l+1) = \alpha \prod_{v_k \in Q(c_i)\backslash v_j}^{v_k<v_j} \mathrm{sign}(V_{i,k}(l)) \cdot$$

$$\prod_{v_k \in Q(c_i)\backslash v_j}^{v_k>v_j} \mathrm{sign}(V_{i,k}(l+1)) \cdot \min$$

$$\left[\min_{v_k \in Q(c_i)\backslash v_j}^{v_k<v_j} |V_{i,k}(l)|, \min_{v_k \in Q(c_i)\backslash v_j}^{v_k>v_j} |V_{i,k}(l+1)|\right]$$

- *Step 3:* Bit nodes information renewal.

$$V_{j,i}(l+1) = P_j + E_j + \sum_{c_k \in R(v_j)\backslash c_i} C_{k,j}(l+1)$$

- *Step 4:* Decoding hard decision HD.

$$V_j(l+1) = P_j + E_j + \sum_{c_i \in R(v_j)} C_{i,j}(l+1)$$

If $V_j(l+1) \le 0$, $v_j = 1$, otherwise, $v_j = 0$.

- *Step 5:* Stop checking the decoding criterion, if the codeword vector $\overrightarrow{v} = (v_1, v_2, v_3, \cdots, v_{n-1}, v_n)$ satisfies the check equation $\overrightarrow{v} \cdot H^T = \overrightarrow{0}$ or reaches the maximum number of iteration $l = N_{max}$, and terminate the decoding iterations and output the vector $\overrightarrow{v}$. Otherwise, let $l = l + 1$, jump to step 2 to repeat the process.

In step 3 and 4, the E_j is the additional decision information provided by the corresponding bit nodes. The calculation of the E_j depends on the 6 observations mentioned above, where there are 6 possible outcomes.

1) The bit node v_j is located in the upper page. The initial information P_c of the corresponding bit node and the hard decision HD_c of decoding have different symbols, namely, $P_c \cdot HD_c < 0$. It demonstrates that the data of detected lower page locates in the fuzzy area of two adjacent states. Based on the Observation 1, the original upper page bit node v_j has a very large probability of being "1". Therefore, we set the E_j as a negative value to make the bit node v_j prone to bit "1". Let $E_j = -1, v_j \in \textit{upper page}$.
2) The bit node v_j is located in the lower page and $P_c \cdot HD_c < 0$. Based on Observation 2, the original upper page bit node v_j has a very large probability of being "0". Therefore, we set the E_j to be a positive value to make the bit node v_j prone to bit "0". Let $E_j = 3, v_j \in \textit{lower page}$.
3) The bit node v_j is located in the lower page and $P_c > 0$ and $HD_C > 0$. Based on Observation 3, the original upper page bit node v_j has a very large probability of being "0". Therefore, we set the E_j to be a positive value to make the bit node v_j prone to bit "0". Let $E_j = 3, v_j \in \textit{lower page}$.
4) The bit node v_j is located in the lower page and $P_c < 0$ and $HD_C < 0$. Based on Observation 4, the original upper page bit node v_j has a very large probability to be to correct. Therefore, we let the $E_j = \alpha P_j$, where α is usually set to be 0.75.
5) The bit node v_j is located in the upper page and $P_c > 0$ and $HD_C > 0$. Based on Observation 5, the data value of original lower page bit has a very large probability of being "0". Therefore, we set the E_j to be a positive value. Let $E_j = 3, v_j \in \textit{upper page}$.
6) The bit node v_j is located in the upper page and $P_c < 0$ and $HD_C < 0$. Based on Observation 6, the data value of original upper page bit has a very large probability of being "1". Therefore, we set the E_j to be a negative value. Let $E_j = -1, v_j \in \textit{lower page}$.

Adding additional decision information into the LDPC decoding process can reduce the decoding latency. When the LDPC codes are decoded, the provided extra information can boost the decision accuracy, which makes all the bits satisfy the check equations at a higher convergency speed.

V. Simulation Results

In this section, we present our simulation experimental setup and evaluation results of the proposed scheme.

A. Evaluation Methodology

In order to test the decoding speed of the proposed REAL scheme, we select the traditional logarithmic domain Min-Sum (LD-MS) and the probability domain BP (PD-BP) decoding schemes as the comparison baselines, using the built Matlab simulation environment. In the simulation experiment, we select the (Quasi-Cyclic LDPC) QC-LDPC code fitting the NAND flash storage systems, the code length of 2KB, the column weight of 4, the row weight of 36, the code rate of 8/9, and utilize the AWGN (Additive White Gaussian Noise) channel to simulate the flash channel. The experiment is divided into three steps. First, we construct the check matrix that does not have four rings by the method of computer searching local optimal solution. Second, information bits are encoded by applying the Gauss elimination algorithm. Third, adding simulated retention error noise and decoding.

B. Evaluation Results

Retention errors are caused by the charge leakage, which are related to the P/E cycles and the retention time. When the flash cells have experienced a large number of P/E cycles, the retention error noise is larger, which results in higher RBER of MLC NAND flash. Fig. 10 shows the change trend of the RBER along with the SP/SNP. The RBER increases with the improvement of the noise intensity. Moreover, the average decoding iteration numbers comparison between the baselines and the proposed REAL scheme at different SP/SNP are shown in Fig. 11. Simulation results show that the proposed REAL scheme can reduce the decoding iterations to a degree much

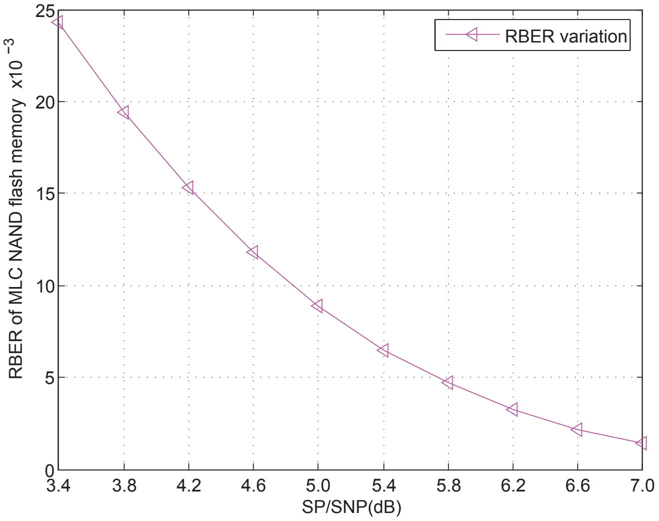

Fig. 10. The RBER of MLC NAND flash varies with SP/SNP.

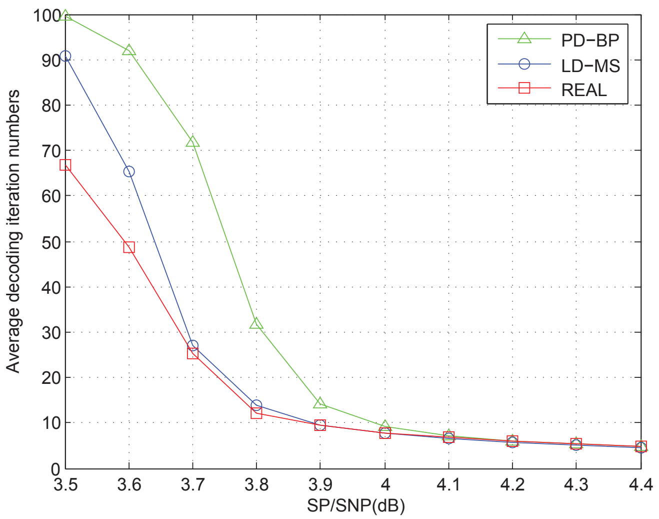

Fig. 11. Comparison of the number of decoding iterations at different SP/SNP.

lower than that of the traditional LD-MS and PD-BP decoding schemes. Besides, the developed REAL scheme exhibits excellent performance when the retention error power is much larger (i.e., SP/SNP is much lower). The charge leakage is more severe at lower SP/SNP. The developed scheme can take full advantage of the valuable observations which are conducive to the LDPC decoding. The proposed REAL scheme can reduce the decoding latency by 26.44% and 33.05% at SP/SNP 3.5 (shown Fig. 12), compared with the LD-MS and PD-BP decoding schemes, respectively. The developed scheme has much faster decoding convergence speed than baselines and thus the read performance of MLC NAND flash is improved.

Besides the decrease of the LDPC decoding latency, the failure rate of the decoded codewords is also lower than the baselines. Simulation results show that the decoding failure rate is much higher at lower SP/SNP (i.e., retention error rate is much larger) by using the baseline decoding schemes than the proposed scheme. If the SP/SNP continues decreasing, the retention error rate becomes much higher. Therefore, the baseline decoding schemes are not able to correct the LDPC codewords, which leads to a larger number of decoding cycles and thus incurs higher decoding delay compared with REAL. However, the proposed decoding scheme is still capable of correcting the LDPC codewords at lower SP/SNP (see Fig. 13) and has much lower decoding failure rate. When the retention error rate is lower (i.e., SP/SNP is larger), the decoding performance is roughly the same between the proposed scheme and the baselines.

C. Read Performance Evaluation of NAND Flash

The LDPC decoding latency has an important influence on the read performance of NAND flash, and it is the main part of the read delay, accounting for most of the time. In this paper, we consider the read delay from the NAND flash array pages to the I/O data buffer and establish the mathematical analysis model (shown in section III) of the relationship between the LDPC decoding latency and the NAND flash read performance. This constructed model plays a role as a bridge linking the LDPC decoding delay and the read performance. Based on this built mathematical model, we evaluate the read performance of NAND flash. In this analysis model, we assume that the time it takes to read data from the flash array pages to the LDPC decoder is constant (i.e., according to the technical reference manual from Micron [39], this time is 25us). We also assume that this time is constant for the data to transfer from the LDPC decoder to the I/O data buffer. We select the PCIe 2.0*8 interface (i.e., 5GB/s), 2KB data pages are transferred from the LDPC decoder to the I/O data buffer. The MLC NAND flash read performance is evaluated at different SP/SNP (shown in Fig. 14).

D. Discussion

In this paper, we demonstrate that our proposed scheme REAL can show outstanding performance in the case of higher P/E cycles and retention error rates. When the retention error becomes higher, the charge leakage problem is more severe and the numerical-correlation becomes more obvious. As a result, the obtained observations are effectively applied to the LDPC decoding process and our REAL scheme has more advantages. In the early stage of NAND flash, the P/E cycles are less consumed and the retention error rate is much lower, and thus the proposed scheme may demonstrate less advantage.

VI. Related Work

There are a number of previously developed techniques to reduce the impacts of retention errors. Cai et al [12] proposed the flash correct-and-refresh (FCR) techniques to improve the lifetime of NAND flash. The key idea is to periodically read, correct, and reprogram (in-place) or remap the stored data before it accumulates more retention errors. For the data retention errors, Park et al [11] proposed incremental

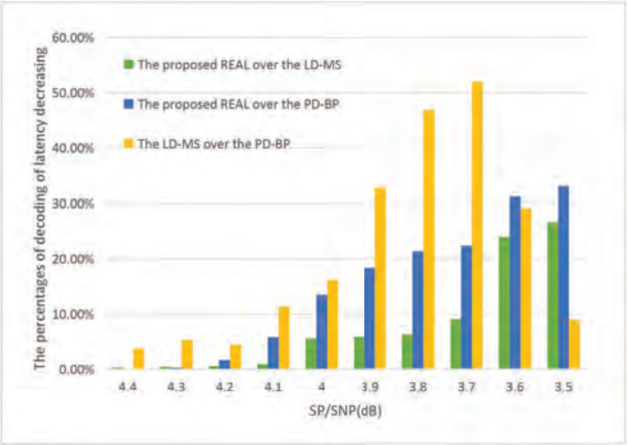

Fig. 12. The reduced decoding latency with REAL.

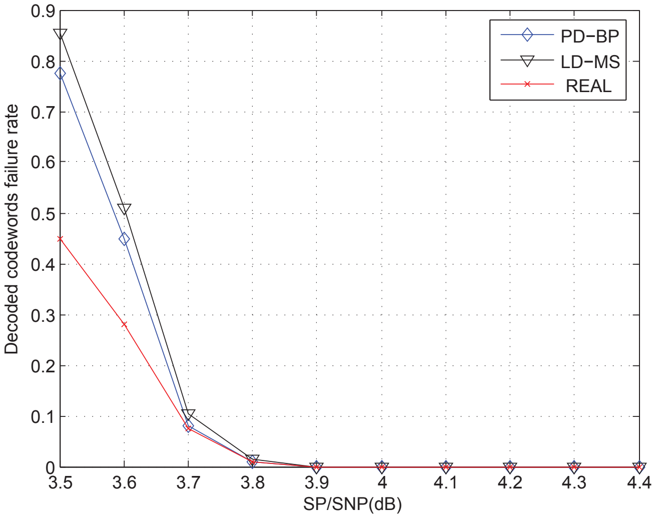

Fig. 13. The comparison of decoded LDPC codewords failure rates.

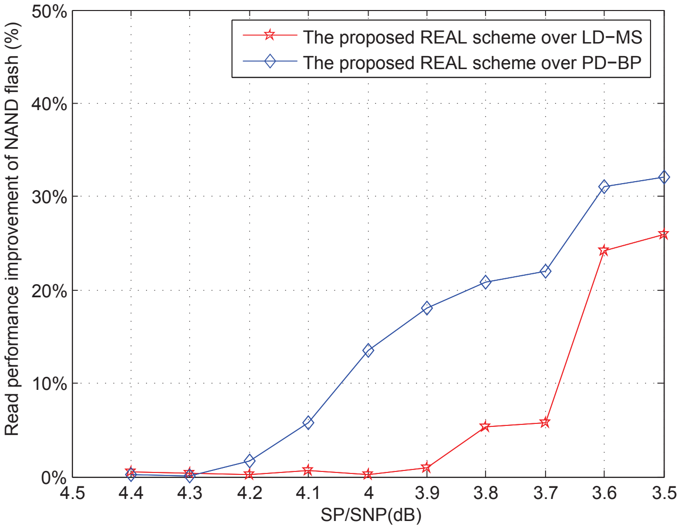

Fig. 14. The improvement in NAND flash read performance of our scheme.

redundancy (IR) that incrementally reinforces error correction capabilities when the data retention error rate exceeds a certain threshold. This method extends the time before data scrubbing occurs, providing a grace period in which the block may be garbage collected. Wang et al [14] provided the programming initial step only (PISO) approach to reduce the number of retention errors, which replenishes the cell charge lost over time by injecting a certain number of electrons into the floating gate. Cai et al [1] developed two techniques (i.e., retention optimized reading (ROR) and retention failure recovery (RFR)). The first technique dynamically adjusts the optimal reading reference voltage adapting to the changes of retention errors. The second recovers data with uncorrectable errors offline by identifying and probabilistically correcting the flash cells with retention errors. In this paper, we reduce retention errors and improve the lifetime of NAND flash from the point of view of the ECC. We propose the REAL scheme, which makes full use of the retention error characteristic (i.e., numerical-correlation) to boost the read performance of NAND flash. The developed scheme combines the error characteristic of NAND flash devices with the LDPC codes, which removes the performance concerns about applying LDPC codes to improve data reliability in flash memory. Moreover, the previous work [2] characterized the error patterns of NAND flash memory and showed that retention errors are caused by charge loss and are the dominant failure mode. These observed error patterns have been the driving forces for the scheme we propose in this paper.

VII. Conclusions

In order to reduce the influences of NAND flash retention errors and improve the stored information reliability, we adopt the LDPC codes with optimized decoding performance. Moreover, retention errors of NAND flash cells have the characteristic of numerical-correlation that motivates us to effectively leverage the characteristic in the process of LDPC decoding in order to decrease the decoding latency and thus improve the NAND flash read performance. Thereby, in this paper, we propose the REAL scheme that accounts for the numerical-correlation characteristic into the LDPC decoding process, which can reduce the LDPC decoding latency by 26.44% and 33.04%, compared to the traditional LD-MS and PD-BP decoding schemes, respectively. Besides, the LDPC decoding latency is significant in the overall NAND flash read delay and directly influences the promotion of NAND flash read performance. Simulation results have shown that the proposed scheme improves the read performance of NAND flash.

Acknowledgement

We would like to thank Xiaosong Ma, our shepherd, and the anonymous reviewers for their valuable comments that greatly improved our paper. This research is sponsored by the National Natural Science Foundation of China under Grant No. 61300047, No. 61370063, No. 61300046, No. 61472152 and No. 61572012 and National Natural Science Foundation of Hubei Province No. 2015CFB315, the Education Ministry of Hubei Province of China under No. B2015054, and the U.S. National Science Foundation (NSF) under Grant Nos. CCF-1547804, CNS-1218960, and CNS-1320349, the National High Technology Research and Development Program of China (863 Program) under Grant No. 2013AA013203. This work is also supported by Key Laboratory of Data Storage System, Ministry of Education. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

REFERENCES

[1] Y. Cai, et al, "Data retention in MLC NAND flash memory: characterization, optimization, and recovery," *IEEE High Performance Computer Architecture (HPCA)*, Feb. 2015.

[2] Y. Cai, et al, "Error patterns in MLC NAND flash memory: measurement, characterization, and analysis," *IEEE Design, Automation&Test in Europe Conference&Exhibition (DATE)*, Mar. 2012.

[3] K. Zhao, et al, "LDPC-in-SSD: making advanced error correction codes work effectively in solid state drives," *in Proceedings of USENIX Conference on File and Storage Technologies (FAST)*, Feb. 2013.

[4] G. Q. Dong, et al, "On the use of soft-decision error-correction codes in NAND flash memory", *IEEE Trans. Circuits Syst.I-Regular Paper*, vol. 58,no. 2, pp. 429-439, Feb. 2011.

[5] W. Zhao, et al, "Improving Min-sum LDPC decoding throughput by exploiting intra-cell bit error characteristic in MLC NAND flash memory", *IEEE Mass Storage Systems and Technologies (MSST)*, June 2014.

[6] N. Mielke, et al, "Bit error rate in NAND flash memories," *In Reliability Physics Symposium (IRPS)*, Apr. 2008.

[7] H. Choi, et al, "VLSI implementation of BCH error correction for multilevel cell NAND flash memory" *IEEE Trans. Very Large Scale Integration (VLSI) Systems,* vol. 18, no. 5, pp. 843-847, 2010.

[8] Y. Pan, et al, "Quasi-Nonvolatile SSD: trading flash memory nonvolatility to improve storage system performance for enterprise applications," *High Performance Computer Architecture (HPCA),* Feb. 2012.

[9] Y. Cai, et al, "Threshold voltage distribution in MLC NAND flash memory: characterization, analysis, and modeling," *IEEE Design, Automation& Test in Europe Conference&Exhibition (DATE)*, Mar. 2013.

[10] F. Margaglia, et al, "Improving MLC flash performance and endurance with extended P/E cycles," *IEEE Mass Storage Systems and Technologies (MSST)*, May 2015.

[11] H. Park, et al, "Incremental redundancy to reduce data retention errors in flash-based SSDs," *IEEE Mass Storage Systems and Technologies (MSST)*, May 2015.

[12] Y. Cai, et al, "Flash correct-and-refresh: retention-aware error management for increased flash memory lifetime", *IEEE International Conference on Computer Design (ICCD)*, Sep. 2012.

[13] S. L. Chen, et al, "Reliability analysis and improvement for multi-level non-volatile memories with soft information," *ACM Proceedings of the 48th Design Automation Conference*, June 2011.

[14] W. Wang, et al, "Reducing MLC flash memory retention errors through programming initial step only," *IEEE Mass Storage Systems and Technologies (MSST)*, May 2015.

[15] Y. Luo, et al, "WARM: Improving NAND flash memory lifetime with write-hotness aware retention management," *IEEE Mass Storage Systems and Technologies (MSST)*, May 2015.

[16] R. Bez, et al, "Introduction to flash memory,"*Proc. of IEEE,* vol. 91, no. 4, pp. 489-502, Apr. 2003.

[17] S. Gregori, et al, "On-chip error correcting techniques for new-generation Flash memories," *Proceedings of the IEEE,* vol. 91, pp. 602-616, Apr. 2003.

[18] Z. Wang, et al, "VLSI design for Low-Density Parity-Check code decoding," *IEEE Circuits and Systems Magazine,* vol. 11, no. 1, pp. 52-69, 2011.

[19] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory,* vol. 8, no. 1, pp. 21-28, Jan. 1962.

[20] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory,* vol. 45, no. 2 pp. 399-431, Mar. 1999.

[21] D. H. Lee, et al," Estimation of NAND flash memory threshold voltage distribution for optimum soft-decision error correction," *IEEE Trans. on Signal Processing,* vol. 61, no. 2, pp. 440-449, 2013.

[22] J. H. Kim, et al, "A high-speed layered Min-Sum LDPC decoder for error correction of NAND flash memories," *IEEE 54th International Midwest Symposium on Circuits and Systems (MWSCAS),* Aug. 2011.

[23] H. Zhong, et al, "Block-LDPC: a practical LDPC coding system design approach," *IEEE Trans. Circuits Syst.I-Regular Paper,* vol. 52, no. 4, pp. 766-775, 2005.

[24] R. C. Bose, et al, "On a class of error correcting binary group codes," *Journal of Information and Control,* vol. 3, no. 1, pp. 68-79, Mar. 1960.

[25] S. Lin, et al, "Error control coding: fundamentals and applications (2nd Ed.)," *Prentice Hall,* 2004.

[26] J. Yang, "Novel ECC architecture enhances storage system reliability," *In Proc. of Flash Memory Summit,* Aug. 2012.

[27] E. Yeo, "An LDPC-enabled flash controller in 40nm CMOS," *In Proc. of Flash Memory Summit,* Aug. 2012.

[28] S. Tanakamaru, et al, "Over-10x-extended-lifetime 76%-reduced-error solid-state drives (SSDs) with error-prediction LDPC architecture and error-recovery scheme," *In IEEE International Solid-State Circuits Conference (ISSCC),* Feb. 2012.

[29] P. Cappelletti, et al, "Failure mechanisms of flash cell in program/erase cycling," *In IEDM Tech. Dig,* pp. 291-294, 1994.

[30] P. Wang, et al, "Low-complexity real-time LDPC encoder design for CMMB," *IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP),* Aug. 2008.

[31] E. Sharon, et al, " An efficient message-passing schedule for LDPC decoding," *IEEE Electrical and Electronics Engineers in Israel,* Sep. 2004.

[32] J. E. Brewer, et al, "Nonvolatile memory technologies with emphasis on flash: a comprehensive guide to understanding and using NVM devices," *IEEE Press Series on Microelectronic Systems,* WILEY Press, 2007.

[33] H. Ma, et al, "MLC nand flash retention error recovery scheme through word line program disturbance," *IEEE Next-Generation Electronics (ISNE)*, May 2014.

[34] J. Lee, et al, "Degradation of tunnel oxide by FN current stress and its effects on data retention characteristics of 90-nm NAND flash memory," *Reliability Physics Symposium Proceedings*, Mar. 2003.

[35] R. G. Forbes, "Refining the application of Fowler-Nordheim theory". *Ultramicroscopy*, vol. 79, no. 1, pp. 11-23, 1999.

[36] J. Zhao, et al, "On implementation of min-sum algorithm and its modifications for decoding low-density parity-check (LDPC) codes," *IEEE Trans. Communications,*vol. 53, no. 4, pp. 549-554, 2005.

[37] S. Li, et al, "Improving multi-level NAND flash memory storage reliability using concatenated BCH-TCM coding," *IEEE Trans.Very Large Scale Integration (VLSI) Systems,* vol. 18, no. 10, pp. 1412-1420, Oct. 2010.

[38] J. S. Yedidia, et al, "Divide and concur and difference-map BP decoders for LDPC codes," *IEEE Trans. Information Theory,* vol. 57, no. 2, pp. 786-802, 2011.

[39] Micron Technology,"16, 32, 64, 128Gb NAND Flash Memory User's manual", *http://micron.com.*

[40] L. R. Varshney, "Performance of LDPC codes under faulty iterative decoding," *IEEE Trans. Information Theory,* vol. 57, no. 7, pp. 4427-4444, 2011.

[41] A. Rasheed, et al, "Performance analysis of faulty gallager-B decoding of QC-LDPC codes," *IEEE Telecommunications Forum (TELFOR),* Nov. 2013.

[42] M. Mucha, et al, "Maximum matchings via Gaussian elimination," *IEEE Foundations of Computer Science,* Oct. 2004.

[43] A. Emran, et al, "Simplified variable-scaled min sum LDPC decoder for irregular LDPC codes," *IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2014.

[44] P. Huang, et al, "FlexECC: partially relaxing ECC of MLC SSD for better cache performance," *USENIX Annual Technical Conference (USENIX ATC'14)*, 2014.

[45] J. Meza, et al, "A large-scale study of flash memory failures in the field," *ACM Sigmetrics International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, June 2015.