Page-Based Dynamic Partitioning Scheduling for LDPC Decoding in MLC NAND Flash Memory

Kang Wei, *Student Member, IEEE*, Jun Li, *Senior Member, IEEE*, Lingjun Kong, *Member, IEEE*, Feng Shu, *Member, IEEE*, and Francis C. M. Lau, *Senior Member, IEEE*

***Abstract*—To increase program-and-erase (PE) cycles for the multi-level-cell NAND flash memory, a page-based dynamic partitioning scheduling (DPS) algorithm for low-density parity-check codes is proposed in this brief. The proposed scheme provides a dynamic scheduling metric to obtain variable nodes (VNs) with the highest erroneous probabilities in each iteration so as to improve the convergence speed of decoding. Then, two DPS-based belief-propagation (BP) and min-sum (MS) decoding algorithms are introduced, named DPS-based BP (DPS-BP) and DPS-based MS (DPS-MS), respectively, which utilize the interaction between the upper page and the lower page bits to detect the probable charge-shift memory cells. Simulation results show that the proposed DPS-BP and DPS-MS decoding algorithms improve the PE endurance up to about 700 and 1000 cycles against the conventional BP and MS decoding algorithms, respectively, at a bit-error-rate of 10^{-5}. In addition, an effective approximation method is presented to reduce the hardware complexity in practical implementations, in which the size of each partitioned group is preset to raise the efficiency of the VN units.**

***Index Terms*—MLC NAND flash memory, LDPC codes, belief propagation, error performance.**

I. Introduction

NAND flash memory, widely used in solid state drives (SSDs) today, is the most popular type of non-volatile memory (NVM). It can store multiple bits over a single memory cell by employing the multi-level-cell (MLC) or triple-level cell (TLC) technique [1]. Due to the requirements of small flash cell size and high reliability, the serious scaling challenges loom up in NAND flash memory. These challenges are originated from flash device characteristics and the corresponding noise sources, such as programming noise (PN), cell-to-cell interference (CCI), random telegraph noise (RTN) and retention noise (RN) [2].

To overcome the large degradations of reliability and error performance, several error correcting codes (ECCs) have been considered such as Bose-Chaudhri-Hocquenghem (BCH) [3] and low-density parity-check (LDPC) codes [4], [5]. By exploiting data error characteristics introduced by retention errors, [6] proposes a cooperative error correction scheme, which makes the initial information of LDPC decoding for the most significant bit (MSB) page accurate using the decoding result of the least significant bit (LSB) page. In [7], the characteristic of retention errors caused by cell threshold voltage shift is utilized and a retention error aware LDPC decoding scheme is developed to reduce decoding latency. To achieve better error correction performance, a serial reliability-based iterative min-sum decoding (RBI-MSD) algorithm for LDPC-coded MLC flash memory systems with the serial scheduling is proposed [8]. In the serial scheduling, the variable nodes (VNs) or check nodes (CNs) are sequentially updated according to a predetermined order. Serial scheduling compared with the flooding schedule could converge twice as fast [9]. In [10], an adaptive group shuffled belief-propagation (AGSBP) decoding has been introduced, and it provides a set of simple integer-based rules to regroup the VNs dynamically with improved error performance. Thus, it is of great interest to design a suitable serial decoding scheduling scheme for the MLC NAND flash memory.

In this brief, an innovative page-based dynamic partitioning scheduling (DPS) algorithm is proposed to improve the error performance of the MLC NAND flash memory, especially when the memory cells are severely charge-shift. The proposed algorithm exploits the MLC flash device characteristics, i.e., there exists a large endurance gap and a fixed interaction between the upper page and lower page bits. Under this scenario, a detecting counter is obtained by jointly analyzing their read voltage signals and the corresponding decoded codeword bits. In conjunction with the proposed algorithm, DPS-based BP (DPS-BP) and DPS-based MS (DPS-MS) decoding algorithms are introduced, and the VNs with the highest erroneous probabilities are updated in advance. Simulation results show that the proposed decoding algorithms improve the MLC NAND flash channel tolerance and the program-and-erase (PE) cycles endurance compared with the conventional algorithms. Considering the hardware complexity in practical implementations, we propose an effective approximation method for the

Manuscript received October 29, 2018; revised December 9, 2018; accepted January 28, 2019. Date of publication February 4, 2019; date of current version December 6, 2019. This work was supported in part by NSFC for Young Scientists under Grant 61501250, and in part by NSFC under Grant 61501238, Grant 61771244, Grant 61702258, Grant 61472190, and Grant 61602245. This brief was recommended by Associate Editor Y. Liu. *(Corresponding authors: Lingjun Kong; Jun Li.)*

K. Wei, J. Li, and F. Shu are with the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210014, China (e-mail: kang.wei@njust.edu.cn; jun.li@njust.edu.cn; shufeng@njust.edu.cn).

L. Kong is with the College of Telecommunication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210014, China (e-mail: ljkong@njupt.edu.cn).

F. C. M. Lau is with the Department of Electronic and Information Engineering, Hong Kong Polytechnic University, Hong Kong (e-mail: encmlau@polyu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2019.2897223

DPS scheme, which raises the efficiency of the variable-node units effectively.

II. MLC NAND Flash Memory Channel

Flash device characteristics can be approximated as several noise models, in which the large number of PE cycles limits the operational lifetime. The threshold voltage distribution for erased cells is modeled as a Gaussian distribution and the programmed cells follow the uniform distributions. We model the PN, RTN and RN by Gaussian distribution functions with zero mean and nonstationary standard deviations, which vary with the number of PE cycles. Using the cell precoding technique, the CCI effect of the programmed cells is mitigated. A shifted mean is considered for the threshold voltage distribution function of erased state cells because of this interference and technique [11]. In general, the final threshold voltage distribution functions are calculated by the convolution integral of initial voltage distribution functions with various noise functions and denoted as p_{s_0}, p_{s_1}, p_{s_2} and p_{s_3} [11], [12].

With a non-uniform 6-level quantization scheme [11], the threshold voltage is split into 7 regions, denoted by $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$ and $\mathcal{O}_4$. Memory cells whose observed threshold voltages fall in $\mathcal{E}_1$, $\mathcal{E}_2$ and $\mathcal{E}_3$ can easily incur errors. So, these regions are named erasure regions. In contrast, $\mathcal{O}_1$, $\mathcal{O}_2$, $\mathcal{O}_3$ and $\mathcal{O}_4$ are usually named data regions [7].

For MLC flash memory, LSB and MSB belong to a lower and an upper page, respectively. To exploit the intracell bit-error characteristics, we put the upper page bit and the lower page bit that belong to the same cell into one codeword [13]. In [11], an optimized method for write-voltages V_1 and V_2 is proposed. With such a multipage data organization, lower and upper pages have exactly the same size but unequal error-rate performances. The raw error-rate performances of lower page bits Ψ_l are better than upper page bits Ψ_u versus the number of PE cycles. In addition, lower-page and upper-page bits share the same threshold voltage in one memory cell and suffer from possible charge-shift simultaneously. Although the LDPC decoding improves the bit-error performance a lot for the MLC NAND flash memory, the relationship between the lower page bit and the upper page bit is not utilized during the decoding procedure.

III. Dynamic Partitioning Scheduling Based BP Decoding Algorithm and Its Approximation

A. Charge-Shift and Error Position Detecting

We have already illustrated an example of 6-level non-uniform quantization scheme. Let $\Pr(A_{\mathcal{E}_j}^{s_i})$ denotes the shift probability that the memory cell was orginally programmed as s_i but appeared in the erasure region $\mathcal{E}_j$. Meanwhile, $\Pr(A_{\mathcal{O}_j}^{s_i})$ denotes the shift probability that the memory cell was orginally programmed as s_i but appeared in the data region $\mathcal{O}_j$. Noticeably, suffering from the various noise sources and interferences, it is inevitable that charge-shift may be taken place in several memory cells. In Fig. 1, the shift probabilities of various input symbols are plotted. If the charge-shift happens in a memory cell, it may bring incorrect initial LLRs

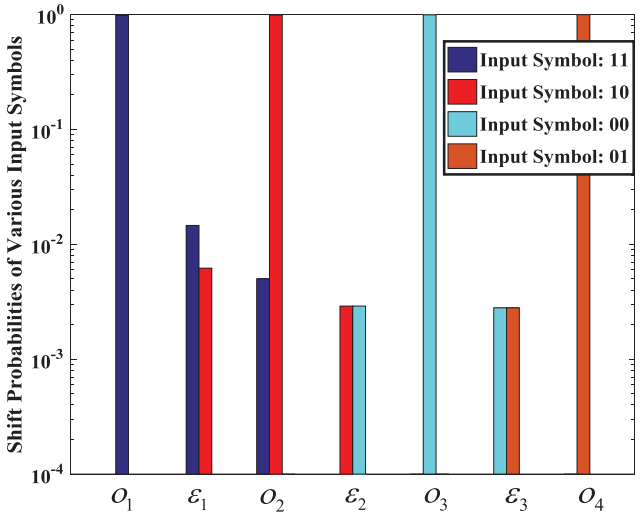

Fig. 1. Shift probabilities of input symbols '11', '10', '00' and '01' when PE = 20K.

and BP decoding failure. An arbitrary ξ is used to present the quantized values of various error probabilities.

Suppose a pair of decoding bits are $\{z_l, z_u\} = \{1, 1\}$, where z_l represents the lower page bit and z_u represents the upper page bit in this memory cell. The unfit conditions are considered in the following. If the memory cell is observed in data region $\mathcal{O}_2$, the second bit may be decoded incorrectly and the corresponding shift probability is $\Pr(A_{\mathcal{O}_2}^{s_0})$. Then, a detecting counter $\boldsymbol{\eta}$ (η_l, η_u) is defined to compare the error probabilities. The value of this detecting counter is set as ξ. If the threshold of the memory cell is observed in erasure region $\mathcal{E}_2$, data region $\mathcal{O}_3$ or erasure region $\mathcal{E}_3$, $\Pr(A_{\mathcal{E}_2}^{s_0})$, $\Pr(A_{\mathcal{O}_3}^{s_0})$ and $\Pr(A_{\mathcal{E}_3}^{s_0})$ are represented as the shift probabilities, respectively. Due to the small values of these shift probabilities, the values of this detecting counter for the first and second bit can be set as 2ξ and ξ. If the memory cell is observed in data region $\mathcal{O}_4$, the value of detecting counters for the first bit is set as 3ξ because of the very small shift probability $\Pr(A_{\mathcal{O}_4}^{s_0})$.

When the decoding bits of the lower page and upper page are $\{z_l, z_u\} = \{1, 0\}$ and the memory cell is observed in data region $\mathcal{O}_1$, the value of this detecting counter for the second bit is set as ξ. If the memory cell is observed in data region $\mathcal{O}_3$ or erasure region $\mathcal{E}_3$, $\Pr(A_{\mathcal{O}_3}^{s_1})$, $\Pr(A_{\mathcal{E}_3}^{s_1})$ are represented as the shift probabilities, respectively. Due to the small values of these shift probabilities, the value of this detecting counter for the first bit is set as ξ. If the memory cell is observed in data region $\mathcal{O}_4$, the value of this detecting counter for the second bit is set as 2ξ because of the very small shift probability $\Pr(A_{\mathcal{O}_4}^{s_1})$.

Suppose the decoding bits of the lower page and upper page are $\{z_l, z_u\} = \{0, 0\}$. If the memory cell is observed in data region $\mathcal{O}_1$, the value of this detecting counter for the second bit is set as 2ξ because of the very small shift probability $\Pr(A_{\mathcal{O}_1}^{s_2})$. If the memory cell is observed in erasure region $\mathcal{E}_1$ or data region $\mathcal{O}_2$, $\Pr(A_{\mathcal{E}_1}^{s_2})$ and $\Pr(A_{\mathcal{O}_2}^{s_2})$ are represented as the shift probabilities, respectively. Due to the small values of these shift probabilities, the values of this detecting counter for the first bit is set as ξ. If the memory cell is observed

TABLE I
THE DETECTING COUNTER (η_l, η_u) OF ALL CONDITIONS FOR THE DECODED BITS OF A MEMORY CELL AND ITS OBSERVED REGIONS

	$\mathcal{O}_1$	$\mathcal{E}_1$	$\mathcal{O}_2$	$\mathcal{E}_2$	$\mathcal{O}_3$	$\mathcal{E}_3$	$\mathcal{O}_4$
$s_0(11)$	$(0, 0)$	$(0, 0)$	$(0, \xi)$	$(2\xi, \xi)$	$(2\xi, \xi)$	$(2\xi, \xi)$	$(3\xi, 0)$
$s_1(10)$	$(0, \xi)$	$(0, 0)$	$(0, 0)$	$(0, 0)$	$(\xi, 0)$	$(\xi, 0)$	$(0, 2\xi)$
$s_2(00)$	$(0, 2\xi)$	$(\xi, 0)$	$(\xi, 0)$	$(0, 0)$	$(0, 0)$	$(0, 0)$	$(0, \xi)$
$s_3(01)$	$(3\xi, 0)$	$(2\xi, \xi)$	$(2\xi, \xi)$	$(2\xi, \xi)$	$(0, \xi)$	$(0, 0)$	$(0, 0)$

in data region $\mathcal{O}_4$, the value for this detecting counter of the second bit is set as ξ because of the small shift probability $\Pr(A^{s_2}_{\mathcal{O}_4})$.

When a pair of bits are decoded to $\{z_l, z_u\} = \{0, 1\}$ and the memory cell is observed in data region $\mathcal{O}_1$, the value of this detecting counter for the first bit is set as 3ξ because of the very small shift probability $\Pr(A^{s_3}_{\mathcal{O}_1})$. If the memory cell is observed in erasure region $\mathcal{E}_1$, data region $\mathcal{O}_2$ or erasure region $\mathcal{E}_2$, $\Pr(A^{s_3}_{\mathcal{E}_1})$, $\Pr(A^{s_3}_{\mathcal{O}_2})$ and $\Pr(A^{s_3}_{\mathcal{E}_2})$ are represented as the shift probabilities, respectively. Due to the small values of these shift probabilities, the values of this detecting counter for the first and second bit are set as 2ξ and ξ. If the memory cell is observed in data region $\mathcal{O}_3$, the value of this detecting counter for the second bit can be set as ξ.

Until now, all the conditions of decoding bits in a cell are considered, we conclude the error probabilities for different pages, respectively. The concise description for the detecting counter $\boldsymbol{\eta}$ (η_l and η_u represent lower page and upper page bits respectively) is given in Table I, in which the observed region of a cell is obtained and ξ can be set to 1 for convenience.

B. Dynamic Partitioning Scheduling Based BP Decoding

Let $\mathbf{H}$ be an $M \times N$ parity-check matrix of a binary LDPC code, which has $d_{c,m}$ ones in the m-th row and $d_{v,n}$ ones in the n-th column. When a decoding result z_n is known, there are M syndromes corresponding to the M check equations, given by $f_m = \sum_{n=1}^{N} z_n h_{m,n}$, where $h_{m,n}$ is the element of check matrix $\mathbf{H}$ on the m-th row and n-th column. When detectable errors occur, the syndrome vector $\mathbf{f} = (f_1, f_2, \ldots, f_M)$ will be parity failures. Hence, we define a reliability metric of the hard-decision values as $\mathbf{E} = (E_1, E_2, \ldots, E_N)$ using the syndrome vector $\mathbf{f}$, soft information L_n and column weight $d_{v,n}$. The soft information L_n is updated during the soft decision decoding algorithm. In (1), the minimum $|L_n|$ of the m-th row is selected as w_m, given by

$$w_m = \min_{n \in N(m)} |L_n|. \tag{1}$$

The E'_n collects all the syndromes connected to the n-th received symbol, given by

$$E'_n = \sum_{m \in M(n)} (2f_m - 1) w_m. \tag{2}$$

E_n is the quantization value of E'_n, where α is a positive integer, given by

$$E_n = \left\lceil \frac{\alpha d_{v,\max} E'_n}{\max(E'_n)} \right\rceil. \tag{3}$$

The DPS scheme is summarized below to select a suitable serial decoding scheduling.

Initialization Set $\ell \leftarrow 1$, $t \leftarrow 1$, $\mathcal{G}^{pre} \leftarrow \varnothing$, $\mathcal{G}_\ell \leftarrow \varnothing$ and $\{\Psi_1, \Psi_2\} \leftarrow \{\Psi_u, \Psi_l\}$, obtain w_m, E'_n and E_n using (1), (2) and (3), obtain the detecting counter $\boldsymbol{\eta}$ using Table I.

Step 1 If $t \leq 2$ and $\Psi_t \neq \varnothing$, For all indices of variable-node $v_n \in \Psi_t$, let $\mathcal{G}^{pre} \leftarrow \mathbf{arg}\max E_n$ and remove the selected VNs from Ψ_t, $\Psi_t \leftarrow \Psi_t \setminus \mathcal{G}^{pre}$.

Step 2 For all indices of variable-node $v_n \in \mathcal{G}^{pre}$, let $\mathcal{G}_\ell \leftarrow \mathbf{arg}\max \eta_n$. Then, remove the selected VNs from $\mathcal{G}^{pre}$, $\mathcal{G}^{pre} \leftarrow \mathcal{G}^{pre} \setminus \mathcal{G}_\ell$ and perform $\ell \leftarrow \ell + 1$. If $\Psi_t = \varnothing$, $t \leftarrow t + 1$. If $\mathcal{G}^{pre} = \varnothing$, go to **Step 1**.

Step 3 If $t > 2$, stop and output $\mathbf{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_\lambda)$.

Note that all the VNs are divided into λ groups corresponding to the $\mathbf{G}$, using the different page bits, predetermined reliability metric $\mathbf{E}$ and detecting counter $\boldsymbol{\eta}$. To begin with, the partitioning number ℓ is initialized to 1, which is to represent the starting point of a new iteration. Following the previous steps, the detecting counter $\boldsymbol{\eta}$ and the reliability metric $\mathbf{E}$ are calculated. The DPS scheme first divides all the VNs into two parts corresponding to upper page Ψ_u and lower page Ψ_l separately. Then, it selects the VNs associated with the largest E_n and records the scheduling order into a previous group $\mathcal{G}^{pre}$. In the following, the first group is generated using one or more VNs in $\mathcal{G}^{pre}$, which have the largest value of the detecting counter $\boldsymbol{\eta}$. When all the VNs are selected and assigned into these groups $\mathbf{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_\lambda)$, the algorithm is accomplished. It is obvious that the three metrics are employed in this algorithm and the partitioning result is obtained as $\mathbf{G}$. The VNs with larger erroneous probabilities will be selected firstly and assigned into the previous group.

According to the DPS scheme, the VNs with different erroneous probabilities estimated by our scheme are sorted in the certain sequence. Based on this updating sequence, we reconsider the conventional BP decoding algorithm. The progress of the modified BP decoding algorithm to update the check-to-bit messages $C^{(k)}_{m,n}$ and bit-to-check messages $B^{(k)}_{n,m}$ in the k-th iteration is given as

a) for $n \in \mathcal{V}$, and each $m \in \mathcal{M}(n)$, compute

$$\begin{aligned} C^{(k)}_{m \to n} &= \varphi\left(B^{(k)}_{n',m}, B^{(k-1)}_{n',m}\right) \\ &= 2\tanh^{-1}\left(\prod_{n' \in \mathcal{J}(m) \setminus n \bigcap \mathcal{G}^{(k)}} \left(\tanh\left(\frac{B^{(k)}_{n',m}}{2}\right)\right)\right. \\ &\qquad \left. \times \prod_{n' \in \mathcal{J}(m) \setminus n \bigcap \mathcal{G}^{(k-1)}} \left(\tanh\left(\frac{B^{(k-1)}_{n',m}}{2}\right)\right)\right), \end{aligned} \tag{4}$$

b) for $m \in \mathcal{C}$, and each $n \in \mathcal{J}(m)$, compute

$$B^{(k)}_{n,m} = \psi(C^{(k)}_{m',n}) = l_n + \sum_{m' \in \mathcal{M}(n) \setminus m} C^{(k)}_{m',n}, \tag{5}$$

c) for $n \in \mathcal{V}$, and each $m \in \mathcal{M}(n)$, process

$$L^{(k)}_n = l_n + \sum_{m \in \mathcal{M}(n)} C^{(k)}_{m,n}, \tag{6}$$

where k is the iteration count, l_n is the intrinsic channel information for the variable node v_n, $\mathcal{J}(m) \setminus n$ is the set of

Algorithm 1: Dynamic Partitioning Scheduling Based BP (DPS-BP)

Data: the maximum iteration $K_{\max}$, the hard-decision $\mathbf{z}^{(0)}$, the extrinsic information l_n, the check matrix $\mathbf{H}$
Result: the final decoding result $\mathbf{z} = (z_1, z_2, \ldots, z_N)$

Initialization: $k \leftarrow 0$, $i \leftarrow 1$, $\mathbf{f}^{(0)} \leftarrow \mathbf{z}^{(0)}\mathbf{H}^T$;
while $\boldsymbol{f}^{(k)} \neq 0$ *and* $k < K_{max}$ **do**
  $k \leftarrow k + 1$;
  Obtain the partitioning result $\mathbf{G}^{(k)} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_\lambda)$ based on the scheme above;
  while $i \leq \lambda$ **do**
    for *all indices of variable-node* $v_n \in \mathcal{G}_i$ *implied in check-node* c_m **do**
      $C_{m,n}^{(k)} \leftarrow \varphi(B_{m,n}^{(k)}, B_{m,n}^{(k-1)})$ using (4);
    $i \leftarrow i + 1$
  for *all indices of check-node* c_m *implied in variable-node* v_n **do**
    $B_{n,m}^{(k)} \leftarrow \psi(C_{m,n}^{(k)})$ using (5);
    $L_n^{(k)} \leftarrow B_{n,m}^{(k)} + C_{m,n}^{(k)}$ using (6);
  $z_n^{(k)} = \begin{cases} 1, & L_n^{(k)} \geq 0 \\ 0, & \text{else} \end{cases}$, for $n = 1, 2, \ldots, N$;
  $\mathbf{f}^{(k)} \leftarrow \mathbf{z}^{(k)}\mathbf{H}^T$;
End

neighbors of check node c_m except variable node v_n, $\mathcal{M}(n)\backslash m$ is the set of neighbors of check node v_n except variable node c_m, $\mathcal{G}$ is the set of the selected VNs, $\mathcal{G}^{(k)}$ is the set of VNs which have been updated for k iterations, $\mathcal{G}^{(k-1)}$ is the set of VNs which have only been updated for $k-1$ iterations, $\mathcal{V}$ is the set of VNs and $\mathcal{C}$ is the set of CNs.

Based on above statements, the detailed steps for the proposed decoding algorithm are shown in Algorithm 2. In the horizontal step, the updating process is divided into λ parts corresponding to the partitioning result $\mathbf{G}$ using Algorithm 1. All values of the check-to-bit messages for the VNs in the same group are updated in parallel and the λ groups are processed sequentially. When all the VNs are scheduled, the horizontal step is completed for one iteration. The vertical step is performed after the horizontal step and the decoded bits $\mathbf{z}^{(k)}$ are obtained based on $L_n^{(k)}$ for the k-th iteration.

C. Approximation for the Proposed DPS-BP Decoding

Considering the practical implementations for a LDPC decoder, there are two groups of processing units the variable-node units (VNUs) and check-node units (CNUs). During the decoding, the computed messages are transmitted between VNUs and CNUs through the connection networks in each iteration [13], [14]. The proposed DPS-BP decoding algorithm finds the VNs belonging to $\mathcal{G}_i$ and then performs the horizontal steps using the VNUs, the number of which is same as the size of $\mathcal{G}_i$. Therefore, the required VNUs of the DPS-BP decoding algorithm architecture are corresponding to the size of each group. However, when the size is not certain and changed, several VNUs may be idle for some time in the decoding process.

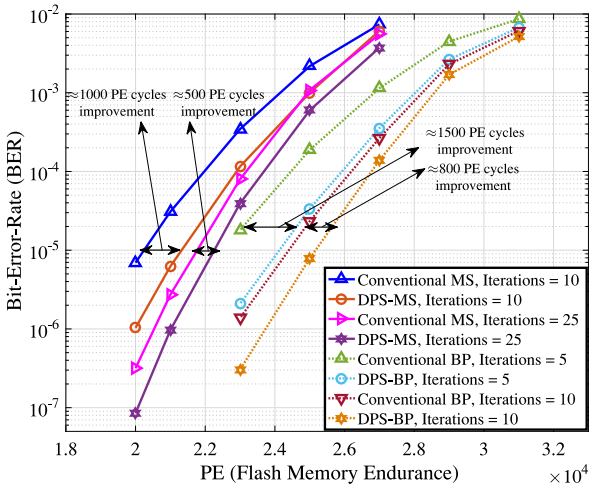

Fig. 2. BER performance of various decoding algorithms with $K_{\max} = 5$, 10 and 25 over different PE cycles.

To avoid the hardware being idle, we propose an approximated DPS method for this decoding algorithm. The main difference between the proposed algorithm and approximated method is the number of the selected VNs in each group. In this approximated scheme, an N-bit codeword $\mathbf{c}$ is divided into λ groups of equal size $N_\lambda = N/\lambda$ and each group contains certain VNs in advance. The certain VNs are obtained by the following approximated scheme. When the satisfied VNs are found in the generating process of $\mathcal{G}_i$, the number is usually unequal to the preset size N_λ in the DPS algorithm. If the number of satisfied VNs is larger than the preset size, the approximated scheme randomly selects adequate VNs to fill in $\mathcal{G}_i$. In contrast, if the number is smaller than the preset size, these VNs are recorded and more VNs are considered until the preset size is reached.

IV. Simulation Results

In order to investigate the error performance and convergence speed of the proposed algorithms for the MLC NAND flash memory, we construct a quasi-cyclic (QC) LDPC code. In this code, each entry of a small base matrix H_B (7×71) is replaced by either a circulant shift of a 64×64 identify matrix or an all-zero 64×64 matrix. The block-length of this code is 4544 bits and code rate is set as 0.9. This irregular code has column-weight $d_v = 5$ and row-weight $d_c = 50$ or 51. In addition, the positive integer α is set to 2 and the range of E_n is from $-2d_{v,\max}$ to $2d_{v,\max}$. The maximum iteration count $K_{\max}$ is set to 5, 10 or 25. For further decoding complexity reduction, the MS algorithm is considered, in which an approximation [13] of the updating formula (4) is employed. The conventional MS algorithm with the DPS method is denoted by DPS-MS and the approximation method is practicable.

To study the performance of the DPS-BP decoding algorithm, in Fig. 2, we plot the bit-error-rate (BER) curves over the number of PE cycles under the maximum iteration number of 5 and 10. Based on the observation, the flash channel tolerance against the PE-cycles-induced errors is significantly improved using the proposed algorithm. For instance, we assume that the required performance of BER is at 10^{-5}.

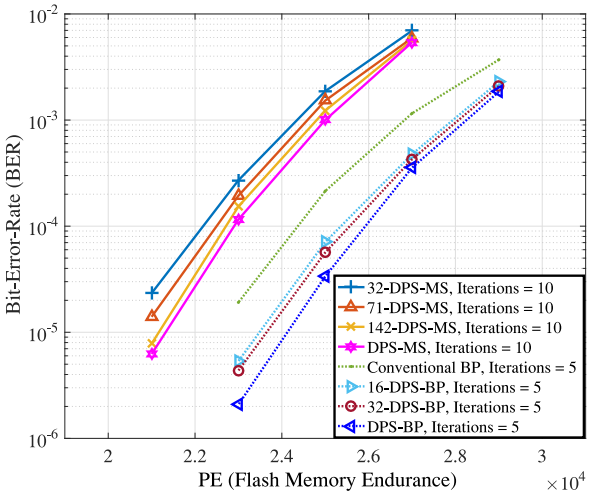

Fig. 3. BER performance of the approximated DPS-BP and approximated DPS-MS decoding algorithms with various λ values over different PE cycles.

TABLE II
AVERAGE NO. OF ITERATION FOR VARIOUS SIMULATED ALGORITHMS WITH QC-LDPC CODE (4544, 448) AND MAXIMUM ITERATION 10 (PCT: PERCENTAGE)

PE	BER	MS	DPS-MS	PCT reduction (%)
27K	10^{-3}	9.41	8.15	13.39
25K	10^{-3}	7.81	5.50	29.58
23K	10^{-4}	6.45	3.88	39.84
21K	10^{-4}	4.39	2.60	40.78

The conventional BP algorithm can approximately endure 23000 and 25000 PE cycles when $K_{\max} = 5$ and $K_{\max} = 10$, respectively. In contrast, for the same BER, the endurance limit of the PE cycles is extended to 24500 and 25700 using the DPS-BP decoding algorithm. Fig. 2 also plots the BER performance of the DPS-MS decoding algorithms. It is shown that, at a BER of 2×10^{-5}, the proposed DPS-MS decoding algorithm improves the PE cycles by up to 1000 cycles under the maximum iteration count 10, and 500 cycles under the maximum iteration count 25 compared with the conventional MS decoding algorithm.

Fig. 3 plots the BER performance of the approximated DPS-MS and DPS-BP decoding algorithms. $\lambda = 32, 71, 142$ and $\lambda = 16, 32$ are set in the approximated DPS-MS algorithm and approximated DPS-BP algorithm, respectively. The required number of VNUs for the approximated decoding algorithm architecture are corresponding to the value of N_λ. Note that the number of VNs in these groups λ yields significant improvement in the BER performance, where the larger λ can bring more improvement but more serial decoding delay.

In the previous statements, we mainly analyse the improvement in error performance using the proposed decoding algorithm. To show the decoding convergence, the conventional MS and our proposed DPS-MS algorithms are executed under 10 maximum decoding iterations. As shown in Table II, the decoding convergence is reduced up to 40.78% in the early state (PE = 21000), and up to 13% when PE = 27000, using the DPS-MS decoding algorithm. Compared with the conventional BP decoding algorithm, the decoding convergence using the DPS-BP decoding algorithm is also improved. Therefore, the proposed algorithms are more effective on the error performance and convergence speed, especially for early states.

V. CONCLUSION

Two DPS-based decoding algorithms, known as DPS-BP and DPS-MS algorithms, have been proposed for the MLC NAND flash memory to mitigate data loss. By virtue of the large endurance gap and fixed relation between the upper page and lower page bits, the decoded bits are calculated and utilized to detect probable charge-changing memory cells, in which VNs with the highest erroneous probabilities are considered and the updated sequences of VNs for the proposed decoding algorithms are redetermined. Through simulations, it is shown that the performance of the proposed DPS-BP and DPS-MS algorithms are improved when compared to the conventional decoding algorithms. In conjunction with the proposed algorithms, the method for the approximated algorithms that fix the number of VNs in each group works at raising the efficiency of the VNUs.

REFERENCES

[1] K. Kim, "Future memory technology: Challenges and opportunities," in *Proc. Int. Symp. VLSI Technol. Syst. Appl.*, Hsinchu, Taiwan, Apr. 2008, pp. 5–9.
[2] Q. Li, A. Jiang, and E. F. Haratsch, "Noise modeling and capacity analysis for NAND flash memories," in *Proc. IEEE Int. Symp. Inf. Theory*, Honolulu, HI, USA, Jun. 2014, pp. 2262–2266.
[3] W. Xueqiang, P. Liyang, W. Dong, H. Chaohong, and Z. Runde, "A high-speed two-cell BCH decoder for error correcting in MLC nor flash memories," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 11, pp. 865–869, Nov. 2009.
[4] Y. Hu *et al.*, "A dominating error region strategy for improving the bit-flipping LDPC decoder of SSDs," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 62, no. 6, pp. 578–582, Jun. 2015.
[5] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 2, pp. 429–439, Feb. 2011.
[6] M. Zhang *et al.*, "CooECC: A cooperative error correction scheme to reduce LDPC decoding latency in NAND flash," in *Proc. 35th IEEE Int. Conf. Comput. Design (ICCD)*, Boston, MA, USA, Nov. 2017, pp. 657–664.
[7] M. Zhang *et al.*, "REAL: A retention error aware LDPC decoding scheme to improve NAND flash read performance," in *Proc. 32nd Int. Conf. Massive Stor. Syst. Technol. (MSST)*, Santa Clara, CA, USA, May 2016, pp. 1–13.
[8] S. Ouyang, G. Han, Y. Fang, and W. Liu, "LLR-distribution-based non-uniform quantization for RBI-MSD algorithm in MLC flash memory," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 45–48, Jan. 2018.
[9] C. A. Aslam, Y. L. Guan, K. Cai, and G. Han, "Informed fixed scheduling for faster convergence of shuffled belief-propagation decoding," *IEEE Commun. Lett.*, vol. 21, no. 1, pp. 32–35, Jan. 2017.
[10] T. C.-Y. Chang and Y. T. Su, "Adaptive group shuffled decoding for LDPC codes," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2118–2121, Oct. 2017.
[11] C. A. Aslam, Y. L. Guan, and K. Cai, "Read and write voltage signal optimization for multi-level-cell (MLC) NAND flash memory," *IEEE Trans. Commun.*, vol. 64, no. 4, pp. 1613–1623, Apr. 2016.
[12] J. Moon *et al.*, "Statistical characterization of noise and interference in NAND flash memory," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 8, pp. 2153–2164, Aug. 2013.
[13] H. Sun *et al.*, "Exploiting intracell bit-error characteristics to improve min-sum LDPC decoding for MLC NAND flash-based storage in mobile device," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 8, pp. 2654–2664, Aug. 2016.
[14] W. Shao, J. Sha, and C. Zhang, "Dispersed array LDPC codes and decoder architecture for NAND flash memory," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 8, pp. 1014–1018, Aug. 2018.