Minimizing Cell-to-Cell interference by Exploiting Differential Bit Impact Characteristics of Scaled MLC NAND Flash Memories

Yejia Di*, Liang Shi**, Congming Gao*, Kaijie Wu*, Chun Jason Xue† and Edwin H.M. Sha*
*College of Computer Science, Chongqing University, Chongqing, China
† Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong

***Abstract*—Appealed by the market, flash memory density is being increasingly improved, and the technology scale is being reduced. Currently, scaled multi-level-cell (MLC) flash memory has been the dominant in the global flash memory markets. However, the reliability of MLC flash memory becomes the urgent challenge, where cell-to-cell interference has been well recognized as the major error source. In this work, we propose to minimize cell-to-cell interference through exploiting the differential impacts on the multiple-bit of MLC flash memories. MLC flash memory generally has two or more bits per cell, such as 2-bit/cell or 4-bit/cell, which can be differentially interfered by neighboring cell programming. Based on the understanding of the programming characteristics of MLC flash memory, we found that higher-order bits can be higher interfered and be more significant interference sources. In order to understand the characteristics of cell-to-cell interference on the multiple bits, we first present cell-to-cell interference models for multiple bits, respectively. Then based on the model, a state mapping scheme is designed to minimize cell-to-cell interference through mapping the states of high-order bits. The mapping scheme is motivated by the recent studies on the cell-to-cell interference characteristics of the multiple cell states of flash memory, where different states have varying interferences. In this case, high-order bits should be mapped from high interference states to a low one. A series of experiments show that the proposed scheme is efficient on reducing cell-to-cell interference with negligible overhead.**

I. Introduction

Flash memory has been widely used in our life, such as embedded systems, personal computers, and data centers [1]. During the last decades, flash memory has been well developed through improving the density of flash memory and reducing the bit scale. For example, current flash memory can integrate more than 4-bit per cell [1], and the technology scale is approaching 10-nm [2]. These schemes are effective in reducing the cost of flash memory. However, the development of flash memory has also introduced serious reliability issue [3]. Recent works have been proposed to understand the reliability characteristics of flash memory[4][5][6], and found that reliability issue would hold back the further development of flash memory [7]. One of the major error source, which has been well recognized, is cell-to-cell interference [8]. Cell-to-cell interference would be more serious and become the major challenge, especially for the future high density and small technology scale flash memory. In this work, we will propose approach to minimize cell-to-cell interference for scaled multi-level cell (MLC) flash memory.

*Liang Shi is the corresponding author: shi.liang.hk@gmail.com.

Cell-to-cell interference stems from the voltage shifts induced by neighboring cell programming due to the capacity coupling [9]. To understand cell-to-cell interference characteristics, many prior works [10][5][11] measured cell-to-cell interference on real platforms with different bit densities and technology scales. Based on the understanding of the essential reasons of cell-to-cell interference, subtracted methods have been proposed [12][13][14][8] to minimize the impacts from cell-to-cell interference. Considering that cell-to-cell interference generally increases the voltage shifts, most of these works proposed to redesign the read process by revising the sensing voltage thresholds with fully awareness of the increased voltages. Cai et al. [8] further built a precise model on the read reference voltages based on the states of neighboring programming cells. Different from the above schemes, several other works proposed to reduce cell-to-cell interference during programming stage. Shielding [9][15] was proposed to reduce the coupling capacitors among cells by dividing one page into even/odd sub-pages to avoid the interference in the same page. Wang et al. [16] and Chang et al. [17] identified that neighboring pages are significantly impacted by cell-to-cell interference. They proposed to partition neighboring pages into two programming groups. In this case, the intra group interference was eliminated, and the inter group interference can be minimized if there are invalid pages in the other group. Chang et al. [18] observed that there are slow cells in flash memory, which can induce a larger cell-to-cell interference. They proposed to reduce cell-to-cell interference by speeding up the slow cell programming process. However, none of these works take the differential impacts on the multiple bits of an MLC flash memory into consideration.

In this work, a cell-to-cell interference minimization scheme is proposed through exploiting the differential impacts on the multiple bits of scaled MLC flash memories. The scaled MLC flash memories generally have 2 or more bits in one cell. These bits are separately into high-order and low-order bits, where they belongs to different flash pages. The multiple pages are programmed separately for reliability concerns. Based on the organization, we have done an analysis on the characteristics of cell-to-cell interference for the multiple bits. From our analysis, we have two observations: First, the multiple bits are differentially impacted by cell-to-cell interference; Second, different bit programming has varying interference to other cells, where higher-order bits can induce higher interference. In order to understand the characteristics

of the differential cell-to-cell interference on the multiple bits in one cell, several cell-to-cell interference models are presented for the multiple bits respectively. Finally, based on the models, we have proposed a state mapping scheme to minimize cell-to-cell interference among the multiple bits of a cell by exploiting the differential bit impact characteristics. The basic idea of the state mapping scheme is to minimize cell-to-cell interference through mapping the states of high-order bits. The state mapping is motivated by the recent studies on the multiple states of cells, where different states have varying impacts on cell-to-cell interference. The state mapping scheme is designed to map the states of high-order bits from a high interference state to a low one. Experimental results show that the proposed scheme achieves encouraging reduction of cell-to-cell interference. This work has following contributions:

- Presented analysis and models for the multiple bits of a cell on the characteristics of cell-to-cell interference;
- Proposed a state mapping scheme to exploit the differential bit impacts for minimizing cell-to-cell interference;
- Presented experiments to show the effectiveness of the proposed state mapping scheme.

The rest of the paper is organized as follows. Section II is the background and related work. Section III presents cell-to-cell interference minimization scheme by exploiting the differential bit impacts. Experiments are presented in Section IV. Finally, Section V summarizes the paper.

II. Background and Related Work

In this section, we first introduce backgrounds on cell-to-cell interference. Then, related works on reduction of cell-to-cell interference are discussed.

A. Cell-to-Cell Interference

Flash memory cell uses floating gate to store data. During programming operation, electrons are injected in the cell to represent data. For single level cell (SLC), the data is represented by whether there are electrons in the cell. For multiple level cell (MLC), the data is represented by the different number of electrons in the cell. Take 2-bit per cell MLC flash memory as an example. It uses 3 voltage references to differentiate the four states of the cell, as shown in Figure 1(a). By injecting different number of electrons in the cell, the voltage of the cell would fall to different ranges. With this scheme, the data can be well stored in the memory cell. However, apart from the injected electrons, voltage of memory cell is coupled with the programmed voltages of its neighboring cells via parasitic capacitors [14]. In this case, the voltage in a memory cell is varied by not only the corresponding initiative programmed voltage but also the voltage shifts interfered by the neighboring floating gates, which is called cell-to-cell interference [14]. Figure 1(b) shows that the voltage will shift to right due to cell-to-cell interferences, which may cross the voltage thresholds. In this case, the data would be spoiled.

In order to reduce cell-to-cell interference, the organization and operations of flash memory are designed as follows. First, flash memory is organized in wordlines and bitlines, where each wordline has thousands of cells to store data, as shown in Figure 2. The flash cell organization is also called ABL (All-Bit-Line) architecture. During programming, the different

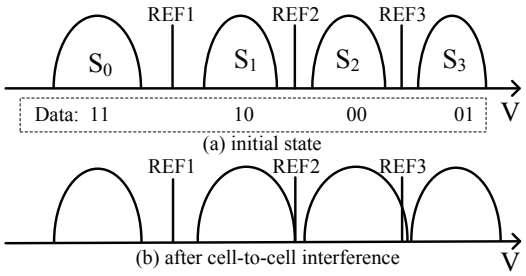

Fig. 1. The cell-to-cell interference effects.

bits of cell are programmed separately. In this case, cell-to-cell interference can be reduced. However, the voltage of memory cell would still be interfered by the programmed voltage of the neighboring cells coupled via parasitic capacitors [14]. The coupling ratio γ is the ratio between the control gate to floating gate capacitance and the total capacitance of the floating gate [19]. Figure 2 shows the coupling ratios for the cell at wordline `i` and bitline `j` (`i`, `j`), where the cell-to-cell interference comes from 8 neighboring cells via parasitic capacitors. Assume γ_x, γ_y and γ_{xy} are the bitline-bitline, wordline-wordline and xy-direction coupling ratios, respectively. The programmed voltage shift $\Delta V_{(i,j)}$ on the (`i`, `j`) cell due to cell-to-cell interference can be formulated as follows:

$$\begin{aligned}\Delta V_{(i,j)} = \gamma_x(\Delta V_{(i,j-1)} + \Delta V_{(i,j+1)}) + \\ \gamma_y(\Delta V_{(i-1,j)} + \Delta V_{(i+1,j)}) + \\ \gamma_{xy}(\Delta V_{(i-1,j-1)} + \Delta V_{(i-1,j+1)} + \\ \Delta V_{(i+1,j-1)} + \Delta V_{(i+1,j+1)})\end{aligned} \tag{1}$$

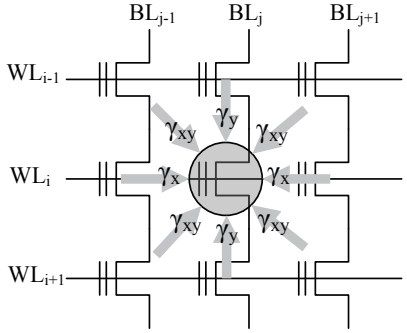

Fig. 2. The cell-to-cell interference effects.

The voltage interference is proportional to the coupling ratios. Previously, a shielded bitline sensing method was proposed to reduce the bitline-to-bitline interference [9]. In this case, the bitline-to-bitline interference can be avoided. Previous works [13][12][20][21] presented that $\Delta V_{(i,j)}$ can be formulated as follows:

$$\begin{aligned}\Delta V_{(i,j)} = \gamma_y(\Delta V_{(i-1,j)} + \Delta V_{(i+1,j)}) + \\ \gamma_{xy}(\Delta V_{(i-1,j-1)} + \Delta V_{(i-1,j+1)} + \\ \Delta V_{(i+1,j-1)} + \Delta V_{(i+1,j+1)})\end{aligned} \tag{2}$$

In this work, we will use the above formula to represent the voltage shift induced by cell-to-cell interference.

B. Related Work

Cell-to-cell interference has been well studied during the last decades since flash memory has been produced. Recent works on cell-to-cell interference can be classified into three groups: measuring the characteristics of cell-to-cell interference on real devices [10][5][11], preventing the interference from neighboring cells [16][17][18], and minimizing the impacts of cell-to-cell interference [12][13][14][8]. ABL architecture applied a preferred programming page-order for flash memory, where cell-to-cell interference on the programmed pages can be minimized [15][8]. Cai et al. [10] observed that the interference mostly stems from the direct neighboring cells, and both the states of programmed cells and programming cells have influence on cell-to-cell interference. Tanakamaru et al. [11] proposed to optimize error correcting code efficiency through fully awareness of the errors induced by cell-to-cell interference. Cai et al. [8] built a model on the optimal read reference for the cell under cell-to-cell interference with ABL architecture. They proposed a neighbor-cell assisted correction method to improve the bit error rate when reading. Chang et al. [17] first proposed a partitioning strategy to prevent the interference from the neighboring cells. The partition scheme was designed to separate the neighboring cells into two programming groups. In this case, the interference intra group can be avoided. The inter group interference, if there are invalid pages in the first group, can be minimized. Besides, Chang et al. [18] found that there are slow cells in flash memory. They found that these slow cells are the dominate of cell-to-cell interference. They proposed to speed up the programming process of the slow cells to minimize the interference. However, none of these works take the differential bit impacts of the MLC flash memory on cell-to-cell interference into consideration. In this work, we will exploit the differential bit impacts for the minimization of cell-to-cell interference for scaled MLC flash memory.

III. Exploiting Differential Bit Impacts for Cell-to-Cell Interference Minimization

In this section, we first present the differential bit impacts for scaled MLC flash memory on the cell-to-cell interference. Then, a state mapping scheme is proposed for the minimization of cell-to-cell interference. Note that based on the number of bits per cell, flash memory can be classified in 2-bit per cell (multiple level cell), 3-bit per cell (triple level cell), and more-bit per cell flash memory. In order to simplify the descriptions, we use 2-bit per cell MLC flash memory in the following. The proposed work can be easily extended to more-bit per cell flash memory because they have similar characteristics.

A. Differential Bit Impacts of Cell-to-Cell Interference

1) Cell-to-Cell Interference Characteristics: Current flash memory applies all-bit-line (ABL) architecture to implement a two-around programming algorithm [1] for cell-to-cell interference minimization. The ABL architecture is designed to programming the multiple bits in a well designed programming order. Figure 3 shows the programming order, LSB_{k-2}, LSB_{k-1}, MSB_k, LSB_{k+1}, and MSB_{k+2}. Based on this programming order, cell-to-cell interference can be reduced. For example, when the bit LSB_{k-1} is programming, only the bit LSB_{k-2} is interfered, as shown in Figure 3(a); when the bit MSB_{k+2} is programming, the interfered bits are the two bits in WL_{i-1} and the bit LSB_{k+1}, as shown in Figure 3(b). Let's go through the programming order to find the reason why the interference can be reduced for this programming order. What's more important is to understand the differential bit impacts on cell-to-cell interference.

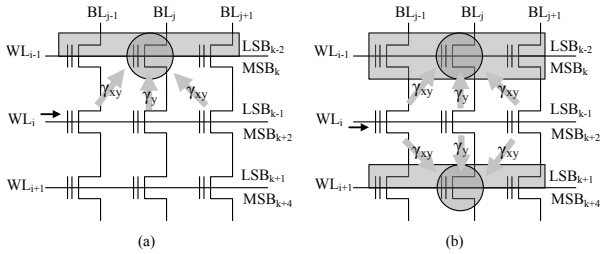

Fig. 3. **Cell-to-cell interference in programming order.** (a) the bit LSB_{k-1} programming. (b) the bit MSB_{k+2} programming.

Table I is a step by step programming example from wordline0 to wordline2. According to the programming order of flash memory with ABL architecture, the page programming flow is: $LSB_0(WL_0) \rightarrow LSB_1(WL_1) \rightarrow MSB_0(WL_0) \rightarrow LSB_2(WL_2) \rightarrow MSB_1(WL_1)$ and so on. In the table, we list the interference between programming.

TABLE I
Various Interference cases for different bit programming with in-page order.

Program order	*Program Bit*	*Interfered Bit*	*Interference Cases*	*Abbr*
1	LSB_0	$--$	*Case* 1	*NULL*
2	LSB_1	LSB_0	*Case* 2	*L2L*
3	MSB_0	LSB_1	*Case* 3	$M2L_1$
4	LSB_2	LSB_1	*Case* 2	*L2L*
5	MSB_1	LSB_0	*Case* 4	$M2L_2$
		MSB_0	*Case* 5	*M2M*
		LSB_2	*Case* 3	$M2L_1$

As described above, there are totally 5 interference cases. In the following, we go through the programming order to analyze the interference.

- Program the LSB in the first wordline. The neighboring bits are not programmed. Thus, the interference is not constructed (Case 1: *NULL*);
- Program the LSB in the following wordlines. The neighboring wordline is programmed with LSB. Based on the features of the two-round programming algorithm [1], the interference on the LSB can be negligible for two reasons: first, LSB has a large voltage range; second, LSB will be reprogrammed during the following MSB programming (Case 2: *L2L*).
- Program the MSB in the first wordline. The neighboring wordline is programmed with only LSB. The interference is also negligible similar to Case 2 (Case 3: $M2L_1$).

- Program the LSB in the following wordlines. This step is same to the second step.
- Programm the MSB in the following wordlines. There are two programmed neighboring wordlines, WL0 and WL2. The 2 bits in WL0 are programmed and the LSB in WL2 is programmed. First, for the LSB in WL0, it could be interfered (Case 4: *M2L*$_2$). This is different from Case 3 because the LSB will not be reprogrammed and it falls in a small voltage margin. Second, for the MSB in WL0, it could be interfered (Case 5: *M2M*). Figure 4 shows the two cases, where the first bit is LSB and the second bit is MSB. Finally, for the LSB in WL2, it is same to Case 3.

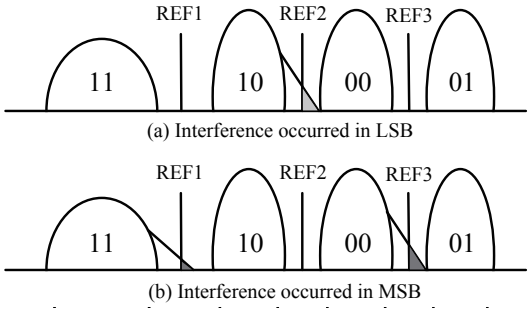

Fig. 4. **Differential Impacted Bits.** (a) Case 4: *M2L*$_2$; (b) Case 5: *M2M*.

Based on the above analysis, we have two observations:

- First, the multiple bits in a cell are differentially impacted by cell-to-cell interference. For example, LSB is interfered only when its corresponding MSB is programmed (Case 4). MSB is only interfered by its neighboring cell's MSB programming (Case 5).
- Second, different bit programming has varying interference to other cells. For example, LSB programming has little interference to other cells (Case 1 and Case 2). MSB interferes its prior neighboring cell (Case 4 and Case 5) and has little interference to its following neighboring cells (Case 3).

2) Differential Bit Interference Models: According to the above observations, multiple bits are differentially impacted by and have differential impacts with cell-to-cell interference. As shown in Section II, cell-to-cell interference would induce increased voltage shift. The models are constructed to show the increased voltage shift for LSB and MSB, respectively.

LSB Interference Model: As shown in Table I, LSB is interfered in three cases, Case 2, Case 3 and Case 4. Thus, the voltage shift of LSB should take these cases into consideration, as follows:

$$\Delta V_{LSB} = \begin{cases} \Delta V_{Case2}, & \text{if the interference case is } L2L; \\ \Delta V_{Case3}, & \text{if the interference case is } M2L_1; \\ \Delta V_{Case4}, & \text{if the interference case is } M2L_2. \end{cases} \quad (3)$$

where Case 2 and Case 3 is negligible. As shown in Figure 4(a), Case 4 can be models as follows:

$$\Delta V_{Case4} = \Delta V_{i,j}, \quad \text{if current state is } 10. \quad (4)$$

where $\Delta V_{i,j}$ is presented in Formula (2) and the current state is MSB with '0' and LSB with '1'.

MSB Interference Model: As shown in Table I, MSB is interfered only in one case, Case 5. Thus, the voltage shift of MSB can be modeled as follows:

$$\Delta V_{MSB} = \Delta V_{Case5}. \quad (5)$$

As shown in Figure 4(b), Case 5 can be models as follows:

$$\Delta V_{Case5} = \Delta V_{i,j}, \quad \text{if current state is } 11 \text{ or } 00. \quad (6)$$

where both LSB and MSB are '1' or '0'. While "11" state can be the negligible one as there are a large margin between state "11" and "10".

Based on the LSB and MSB interference models and the analysis of cell-to-cell interference characteristics, we will propose approach to reduce the interference in the following.

B. State Mapping for Cell-to-Cell Interference Minimization

Based on the LSB and MSB interference models, we can find that cell-to-cell interference is highly correlated with the cell states. For example, LSB is interfered only when the state is "10", and MSB is interfered only when the state is "00". However, previous works stated that severe interferences on these states come from the programming bits with large programming voltages. In the following, we first check the large programming voltage. Then, we discuss the problem for cell-to-cell minimization. Finally, the state mapping scheme is presented.

1) Programming Voltage Interference: As discussed in Section II, flash memory employs the two-round programming algorithm to program the LSB and MSB respectively. Figure 5 is the process of the algorithm, where the 1st round is programming LSB page, while the 2nd round is for MSB page.

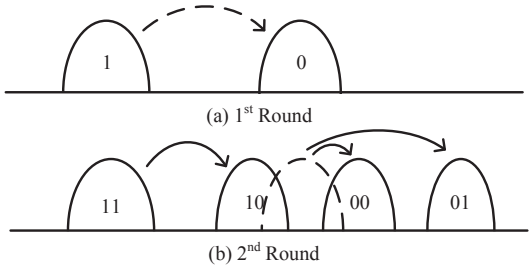

Fig. 5. Two-round programming mechanism for MLC NAND flash memory.

From the two-round program process, we can find that during MSB programming, two cases may induce severe interference. The first one is programming '0' to MSB when LSB state is '1', which is "10"; The second one is programming '1' to MSB when LSB state is '0', which is "01".

2) Problem Statement: Based on LSB and MSB interference models and the voltage interfering, during MSB programming, there are four pairs of interfered and programming states, leading to high interference possibility, as shown in Table II.

Based on above conclusion, the first optimization object of cell-to-cell minimization scheme is to minimize the total number of the four pairs between two wordlines. Second, an MSB programming will interfere two different bits, LSB and MSB. However, the LSB and MSB is protected by the same

TABLE II
FOUR PAIRS OF INTERFERED AND PROGRAMMING STATES DURING MSB PROGRAMMING WITH HIGH INTERFERENCE.

$No.$	Interfered States (WL_{i-1})	Programming States (WL_i)
1	10	1**0**
2	10	0**1**
3	00	1**0**
4	00	0**1**

error correcting code (ECC). In this case, we need to optimize with balancing the four pairs between LSB and MSB pages.

3) State Mapping: Based on above analysis, we propose state mapping scheme to minimize the total number of four pairs between two wordlines. The basic idea of the scheme is simple and straightforward. During the MSB programming, the prior wordline and the current wordline are checked for the number of the four pairs by flipping the MSB or not.

Figure 6 is an example for the state mapping scheme. Assume there are two wordlines, where LSB and MSB of the first wordline is programmed and the LSB of the second wordlien is programmed. Let's check the effects of the state mapping in reducing the number of the four pairs. For the original case, we can find that the number of the four pairs is 3, where there are 1 for LSB and 3 for MSB. For the state mapping scheme, the programming MSB is flipped. We can find that the number of the four pairs is reduced to 1 for LSB. Based on this exmaple, we can find that it is significant to apply state mapping for the programming MSB page, where the reliability of interfered MSB page will be largely improved.

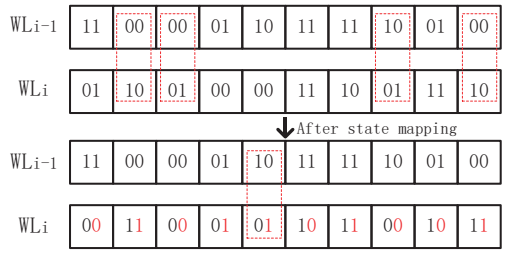

Fig. 6. **Applying state mapping method for to-be programmed MSB page on WL_i.** The red frames record the pairs, and the red states are for the after state mapping one.

Enhanced State Mapping: The decisions on the whole pages are not optimal because the bit states scattered per page are various. To further reduce the cell-to-cell interference, an enhanced state mapping method is proposed, which is designed to partition a page into multiple segments, and making optimization decision for each segment. In this way, we can meet optimization constraints better, and then further minimize cell-to-cell interference.

4) Implementation: Three components should added in the flash controller for the proposed scheme: Encoding Module, Decoding Module and Buffer. Encoding Module is designed to apply state mapping with flipping state '1' and '0' for to-be programmed MSB page. Decoding Module is designed in pairing with encoding module, as the original data may have been flipped. Buffer is for the data of the to-be programmed one and the prior programmed pages. As shown in Table II, the prior programmed pages include prior five pages.

Overhead Analysis: The overhead mainly performs at two aspects.

- There are three module in the implementation structure. Buffer needs 6-page capability, about 12KB with 2KB page, which is negligible. Besides, the performance overhead of encoding and decoding module all can be negligible as the time complexity of our method is O(n);
- There should be several flags to record whether each segment per page is flipped or not, which can be saved in the OOB (out-of-band) of each page without space overhead.

IV. EVALUATIONS

In this section, we first present the experimental methodology. Then, the experiment results are presented.

A. Experimental Methodology

The goal of evaluations is to measure the efficiency of the state mapping method on cell-to-cell interference minimization. First, a set of synthetic results will be presented considering various state distributions. Then, the proposed method is applied for real life files on Linux. Our methodology will compare the number of the four pairs of original case with that after applying state mapping. The experimental platforms include the original situation, situations with the state mapping (SM) method, and the enhanced state mapping (ESM) method.

B. Experiment Results and Analysis

Programming states will have varying influences on interfered states. In this section, we first want to understand the effects on the interference between two states on neighboring wordlines. With this in mind, a set of data is constructed, where two wordlines are initialized with different ratios of two states. For example, S11/S10 with 1/9 means wordline1 has 10% state "11" and wordlinew has 90% state "10". The efficiency of state mapping method with various two state ratios is synthesized, as shown in Figure 7. The efficiencies of

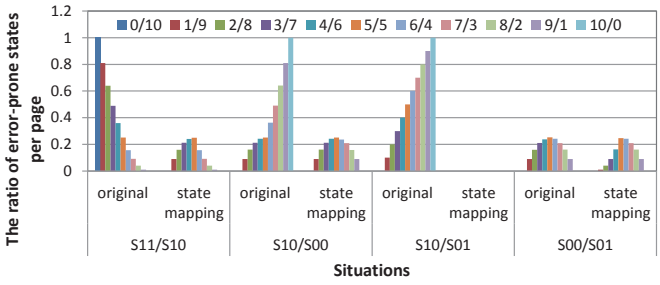

Fig. 7. **The number of the four pairs ratio after state mapping method.** The x axis represents the distribution ratio of two states, and the y axis represents the pair ratio compared to original situation after state mapping method. S_{xx} represents the "xx" state.

S11/S00 and S11/S01 are negligible because the former can not result in severe cell-to-cell interference, while the latter does not include vulnerable states. From the synthetic results, we can find that there are half of the cases which are significant with state mapping method, and the benefit can be up to 90%.

In our preliminary study, we analyzed several real life files to collect the distributions of bit states. The files studied are

partitioned into pages of 2KB including LSB page and MSB page, and one wordline consists of one LSB page and one MSB page totally 4 KB. All the distribution ratios of many real life files are collected. As shown in Figure 8, the distribution ratios vary with files and a little in pages.

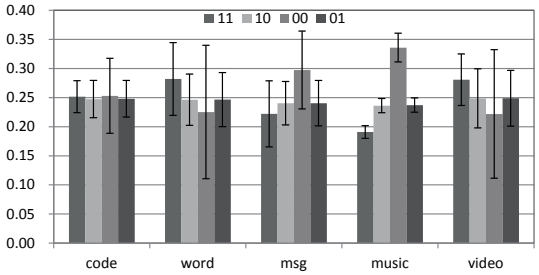

Fig. 8. **The distribution ratio of each states in real life file.** Here, the black lines represent that the standard deviation of each state for many pages.

Figure 9 is the efficiency of state mapping method on real life files. With large standard deviation, enhanced state mapping method achieves significant result, as the number of the four pairs is reduced by 10% in average. However, due to the comparative ratio of each state, the efficiency of global optimization objective is not so significant.

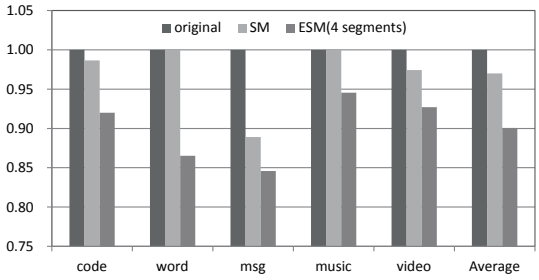

Fig. 9. **The reduced ratio of the four pairs of each file.** Here, each page is divided to 4 segments for ESM.

To understand some detail, we take the first 500 pages of "video" file as an example to show the effect of the proposed schemes. Figure 10 shows the results, where the red one is original one, and the other two are the proposed schemes. We can find that the proposed scheme is encouraging in reducing cell-to-cell interference between wordlines.

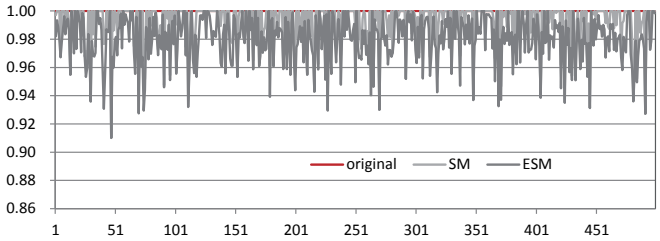

Fig. 10. **The reduced ratio of the four pairs per page for "video" file.**

V. Conclusion

In this paper, differential bit impact on cell-to-cell interference minimization method is proposed. First, with understanding of the characteristics of cell-to-cell interference on the multiple bits, differential bit interference models are constructed. Then, based on the model and motivated by the differential programming voltages, state mapping scheme is designed to minimize cell-to-cell interference through mapping the states of high-order bits. Finally, a series of experiments show that the proposed scheme is efficient on reducing the cell-to-cell interference.

VI. Acknowledgement

This work is supported by the Fundamental Research Funds for the Central Universities (106112016CDJZR185512 and 106112014CDJZR185502), NSFC (61402059, 61472052 and 61572411), and National 863 Programs 2015AA015304.

References

[1] R. Micheloni, L. Crippa, and A. Marelli, "Inside nand flash memories," in *Springer'10*.
[2] S. Choi, D. Kim, S. Choi, B. Kim, S. Jung, K. Chun, N. Kim, W. Lee, T. Shin, H. Jin, H. Cho, S. Ahn, Y. Hong, I. Yang, B. Kim, P. Yoo, Y. Jung, J. Lee, J. Shin, T. Kim, K. Park, and J. Kim, "19.2 a 93.4mm2 64gb mlc nand-flash memory with 16nm cmos technology," in *ISSCC '14*.
[3] Y. Pan, G. Dong, and T. Zhang, "Exploiting memory device wear-out dynamics to improve nand flash memory system performance," in *FAST'11*.
[4] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Error patterns in MLC NAND flash memory: Measurement, characterization, and analysis," in *DATE'2012*.
[5] L. Grupp, A. Caulfield, J. Coburn, S. Swanson, E. Yaakobi, P. Siegel, and J. Wolf, "Characterizing flash memory: Anomalies, observations, and applications," in *MICRO'09*.
[6] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in MLC NAND flash memory: characterization, analysis, and modeling," in *DATE'13*.
[7] L. Shi, Y. Di, M. Zhao, C. Xue, K. Wu, and E.-. Sha, "Exploiting process variation for write performance improvement on nand flash memory storage systems," *TVLSI '15*.
[8] Y. Cai, G. Yalcin, O. Mutlu, E. F. Haratsch, O. Unsal, A. Cristal, and K. Mai, "Neighbor-cell assisted error correction for mlc nand flash memories," *SIGMETRICS'14*.
[9] T. Tanaka, Y. Tanaka, H. Nakamura, K. Sakui, H. Oodaira, R. Shirota, K. Ohuchi, F. Masuoka, and H. Hara, "A quick intelligent page-programming architecture and a shielded bitline sensing method for 3 v-only nand flash memory," *JSC'94*.
[10] Y. Cai, O. Mutlu, E. F. Haratsch, and K. Mai, "Program interference in mlc nand flash memory: Characterization, modeling, and mitigation," in *ICCD'13*.
[11] S. Tanakamaru, Y. Yanagihara, and K. Takeuchi, "Error-prediction ldpc and error-recovery schemes for highly reliable solid-state drives (ssds)," *JSSC'13*.
[12] G. Kong, T. Kim, W. Xi, and S. Choi, "Cell-to-cell interference compensation schemes using reduced symbol pattern of interfering cells for mlc nand flash memory," in *APMRC, 2012 Digest*.
[13] G. Dong, S. Li, and T. Zhang, "Using data postcompensation and predistortion to tolerate cell-to-cell interference in mlc nand flash memory," *TCS'10*.
[14] J. D. Lee, S. H. Hur, and J. D. Choi, "Effects of floating-gate interference on nand flash memory cell operation," *EDL'02*.
[15] K. T. Park, M. Kang, D. Kim, and S. W. Hwang, "A zeroing cell-to-cell interference page architecture with temporary lsb storing and parallel msb program scheme for mlc nand flash memories," *JSSC'08*.
[16] Y. Wang, L. A. D. Bathen, Z. Shao, and N. D. Dutt, "3d-flashmap: A physical-location-aware block mapping strategy for 3d nand flash memory," in *DATE'2012*.
[17] Y. M. Chang, Y. H. Chang, T. W. Kuo, H. P. Li, and Y. C. Li, "A disturb-alleviation scheme for 3d flash memory," in *ICCAD'13*.
[18] Y.-M. Chang, Y.-C. Li, Y.-H. Chang, T.-W. Kuo, C.-C. Hsieh, and H.-P. Li, "On relaxing page program disturbance over 3d mlc flash memory."
[19] Y. S. Hisamune, K. Kanamori, T. Kubota, and Y. Suzuki, "A high capacitive-coupling ratio (hicr) cell for 3 v-only 64 mbit and future flash memories," *EDM'94*.
[20] G. Dong, N. Xie, and T. Zhang, "On the use of soft-decision error-correction codes in nand flash memory," *TCS'11*.
[21] J. J. Kong, H. Son, J. Lee, J. Kim, K. L. Cho, B. V. K. Vijaya Kumar, and Y. Kim, "Modulation coding for flash memories," in *ICNC'13*.