Loss is Gain: Shortening Data for Lifetime Improvement on Low-Cost ECC Enabled Consumer-Level Flash Memory[*]

Yejia Di[⋆], Liang Shi[⋆*], Congming Gao[⋆], Qiao Li[‡], Kaijie Wu[†] and Chun Jason Xue[‡]

[⋆]College of Computer Science, Chongqing University [†]Tandon School of Engineering, New York University

[‡]Department of Computer Science, City University of Hong Kong

ABSTRACT

Reliability has been a challenge in the development of NAND flash memory, due to its technology size scaling and bit density improvement. To ensure the data integrity, error correction codes (ECC) with high error correction capability have been suggested. However, much higher costs will be introduced which cannot be supported for cost-limited consumer-level flash memory. Thus, low-cost ECCs are usually applied. In this work, a reliability improvement scheme is proposed for low-cost ECC enabled consumer-level flash memory. The scheme is motivated by the finding that low-cost ECC is able to protect shortened encoded data with improved reliability. This is because that the less the encoded data are, the less the errors will be occurred. With this motivation, a design is proposed to construct the shortened data case for a low-cost ECC when it cannot be able to provide the reliability requirement. Second, two relaxation approaches are proposed to relax the space reduction as it has bad effects on flash memory. A model guided evaluation is finally presented, and the results show that the lifetime can be significantly improved with little space reduction.

CCS CONCEPTS

• **Information systems** → **Flash memory**; • **Computer systems organization** → *Dependable and fault-tolerant systems and networks*; Reliability;

KEYWORDS

Error Correction Codes, Reliability, Flash Memory

1 INTRODUCTION

During the last decades, NAND flash memory has been widely deployed as the storage devices, due to many advantages, such as non-volatility, high access performance, power efficiency and so on. For further development, the cost of flash memory is reduced by density improvement and technology scaling, even being staked as 3D flash memory [10]. However, the reliability becomes decreasing, where flash memory is easier to get errors. Thus, ECCs with high error correction capability are recommended for data integrity, such as low density parity codes (LDPC) [23]. However, higher costs will be introduced, including the large implementation areas, high energy consumption and long decoding latencies [19]. Thus, high capability ECCs are not applicable for consumer-level flash memory, which usually has cost limitations. In this paper, we focus on the reliability improvement for flash memory devices equipped with low-cost ECCs.

The reliability issue is due to the physical characteristics of flash memory. Flash memory is possibility to get errors for several reasons, such as program/erase (P/E) cycling [1], retention loss [14][3], program disturb [2], and cell-to-cell interference [1]. A series of methods are proposed to improve the reliability which can be classified into two groups. The first group is to use signal processing schemes, such as finding the optimal reference voltages [1], using a slow programming voltage [22][19], and so on. However, these works need to redesign the chip to support new commands. Another group is to use refresh operations when the storage time of the data is approaching the supported retention time of flash blocks [1]. However, the refresh operations need to be conducted online, which cannot be guaranteed in consumer-level flash memory all the time.

In this paper, a data shortened scheme is proposed to improve the reliability of flash memory equipped with low-cost ECCs. The proposed scheme is motivated by the finding that if ECC only needs to protect part of the ECC unit, the probability of correcting errors successfully is increased, representing reliability is improved. This is because that the number of errors is reduced with shortening protected data. In detail, with 50% of data shortened, the reliability can be improved by 100%. Based on the motivation, the basic idea of the data shortened scheme is to make a tradeoff between storage space and reliability improvement. In this case, the low-cost ECC is enabled with shortened data to continuously protect data integrity. However, there are several challenges for the basic idea: First, the size of shortened part is unknown to the designer. Because the reliability of flash memory is varied during the lifetime, the size of shortened part should be determined correspondingly. Second, the storage space is reduced with the increases of shortened part. If the space is significantly reduced, the lifetime and performance can be impacted. In order to solve above issues, the shortened scheme is designed as follows: First, a mapping unit shorten design is proposed to realize the data shortened case. Basically, the mapping unit of state-of-the-art flash memory is generally much smaller than the flash page size. In this case, once the reliability cannot be supported with the low-cost ECC, one more mapping unit is shortened for a flash page. Second, two space relaxation approaches are proposed. The first one is to exploit the process variation (PV) of flash memory to only shorten the weak blocks. The second one is to exploit the different reliability characteristics of the multiple bits

[*]Corresponding author: Liang Shi. Email: shi.liang.hk@gmail.com.

GLSVLSI'18, Chicago, IL, USA
© 2018 ACM. 978-1-4503-5724-1/18/05. . . $15.00
DOI: 10.1145/3194554.3194571

in multiple level cell (MLC) flash memory. With the significantly increased volume of flash memory devices, we believe that using small space to trade for lifetime improvement is very attractive to the consumer level flash devices. A model guided evaluation is performed, and the results show that the proposed scheme can improve the lifetime significantly with little space reduction. The paper has following contributions:

- Presented the data shortened opportunity of low-cost ECC on reliability improvement;
- Presented a design to enable a low-cost ECC on flash memory by shortening data;
- Proposed two methods to relax the storage space reduction;
- Presented model-guided evaluations. Experimental results show that the proposed approach is able to achieve expected lifetime improvement with tolerable space overhead.

The following of the work is organized as follows: In Section 2, the background and related work are discussed. In Section 3, the data shorten motivation is presented. In Section 4, the mapping unit shortened design is proposed. In Section 5, the evaluations on lifetime, storage space are presented. Finally, the work is concluded in Section 6.

2 BACKGROUND AND RELATED WORK

2.1 Background

There are three operations in flash memory, write, read and erase. The write and read unit of flash memory is a flash page, while the erase unit is a flash block, where a flash block consists of hundred times of flash pages. Due to the asymmetric operation units, out-of-place updates by invalidating the original data and writing updated data to new pages are applied. Besides, a mapping table is used for recording data locations. There are three mapping types: block-level mapping, page-level mapping, and finer-grained mapping, where logical pages are mapped to arbitrary physical pages, any pages, and several finer-grained units. A trend for the future flash memory is the large page size which will be employed to reduce the circuit complexity and improve the I/O bandwidth [10]. For large page size flash memory, finer-grained mapping is recommended, where the mapping unit is smaller than the page size, such as 4KB mapping unit for 32KB page [10].

Flash memory is written by charging electronics to the flash cells, where 2^n voltage states are defined for n bit per cell flash memory, such as 4 states for 2 bits per cell called MLC flash memory as shown in Figure 1. These two bits belongs to different pages, the least significant bit (LSB) page and the most significant bit (MSB) page. During programming, LSB page is first programmed and then is the MSB page for reliability concern. The voltage states are distinct with noise margins for distinguished data read. However, margins have possibilities to be decreased, as there are many adverse effects, including P/E cycling [1], retention loss [1][14][3], program disturb [2], cell-to-cell interference [1], and so on. If voltage is affected from one state to another, data will be read out incorrectly. The possibility of data getting errors is called raw bit error rate (RBER), which is mainly due to the adjacent state changes [12]. As the red arrows shown in Figure 1, we can find that MSB page has higher possibility to get errors than that of LSB page. With the development of multiple bits per cell and wearing of flash memory, RBER will be increased. For data integrity, error correction codes (ECCs) are equipped in flash controller to correct errors, where data are encoded as writing and decoded as reading. Usually, the page size is times of ECC-unit sizes [19]. However, the error correction capability of ECC is limited. In this case, flash memory can only be used reliably with limited lifetime that is with limited P/E cycles.

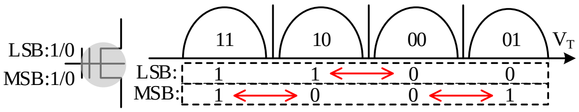


Figure 1: Traditional MLC data states, where the red arrows represent the high-probability error cases with state changes.

2.2 Related Work

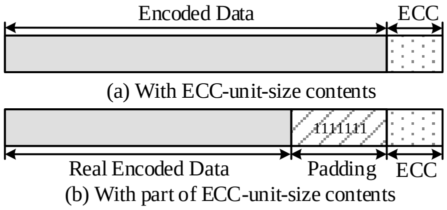
Due to the degraded reliability of flash memory, most of recent works proposed to use strong ECC [23] or multiple ECCs [20] to provide higher error correction capability. However, strong ECC always comes with high costs to the storage systems [19]. In order to solve the reliability issue, many works also proposed to reduce the adverse effects with low-cost ECC for flash memory. Cai *et al.* [1] built a precise model to eliminate the cell-to-cell interference effects. Zhang *et al.* [22] proposed to use slow write to form a larger noise margin. Cai *et al.* [1] proposed to relax retention time by refreshing to reduce the charge loss effects. However, these works either need to redesign the flash chips for new commands, or need to keep system online for refreshing, which are not applicable for consumer-level flash memory. In this work, a method is proposed to increase the reliability of flash memory taking advantage of the characteristic of ECC.

On the other hand, flash memory can tolerate some space reduction for lifetime improvement. In tradition, bad block management (BBM) mechanism is applied for flash memory allowing some bad blocks, where occurred errors cannot be corrected by ECC [17]. Usually, the ratio of bad block is 5%. Thus, the BBM mechanism can only prolong lifetime in a little degree. And then, a set of finer-grained space reduction methods are proposed to prolong the lifetime. For example, previous works found that the wearing of flash pages are different [8][17]. Thus, the flash page can be used as the bad unit. Then, the space reduction speed is reduced. Besides, other works also proposed to revive to-be bad pages to further improve lifetime. There is a theory that if only one bit is stored in the multiple bits cell, the reliability can be improvement. Thus, several works proposed to only store one bit per cell when the reliability cannot be supported by ECC [7][21]. Furthermore, Lin *et al.* [11] proposed to reuse two bad pages as one normal page. In this work, a graceful space reduction method is proposed for lifetime improvement. More importantly, a low-cost ECC is used.

As the algebra operation principle of BCH codes, the encoded data length can be shorter than the designed length, called shortened BCH codes [4]. This is because the shortened contents can be regarded as all '0's in the algebra operation. And the shortened BCH codes can provide better reliability [4]. The proposed work is enlightened but different from the shortened BCH codes, as the data alignment of the ECC unit must be obeyed.

3 DATA SHORTENED MOTIVATION

In this section, the motivation on the proposed scheme is presented. The motivation is on the finding that if the encoded data of ECC unit is shortened, the error correction probability will be increased. Considering ECC alignment of flash page, data shortened case is shown in Figure 2. Figure 2(a) shows the traditional ECC structure with ECC-unit-size contents. While Figure 2(b) shows the shortened case, where part of ECC-unit-size contents are encoded by ECC called real encoded data, and the remaining space with all '1's are called padding. With the case of Figure 2(b), the size of contents requiring protection by ECC becomes shorter, so that the number of errors will be reduced.



(a) With ECC-unit-size contents

(b) With part of ECC-unit-size contents

Figure 2: The ECC structures with different-size encoded data.

Here, the reliability benefit is formulated as follows, where available raw bit error rate (RBER) is regarded as the reliability of flash memory, as it represents the maximum tolerated probability to get errors of raw data. The higher the available RBER is, the longer the lifetime will be achieved [1]. For low-cost BCH *(n,k,t)*, where the encoded data length is k, the generated codedata length is n, and it can correct t errors. Then the probability of m errors occurred is:

$$pr(m,n,RBER) = \binom{n}{m} \cdot RBER^{m} \cdot (1 - RBER)^{n-m} \qquad (1)$$

If m is lower than t, errors can be corrected by ECC and the UBER will be 0. Otherwise, m is larger than t, and there are uncorrectable errors. Thus, the uncorrectable bit error rate (UBER) has the following relationship with RBER.

$$UBER = \frac{\sum_{m=t+1}^{n} m * pr(m,n,RBER)}{n} \qquad (2)$$

With the Formula, Figure 3 presents the quantified results of the relationship of BCH(17264, 16400, 57). As shown in the figure, the larger the RBER is, the larger the UBER is. However, in general, the UBER should be guaranteed to be lower than a threshold for reliability, e.g. 10^{-15}.

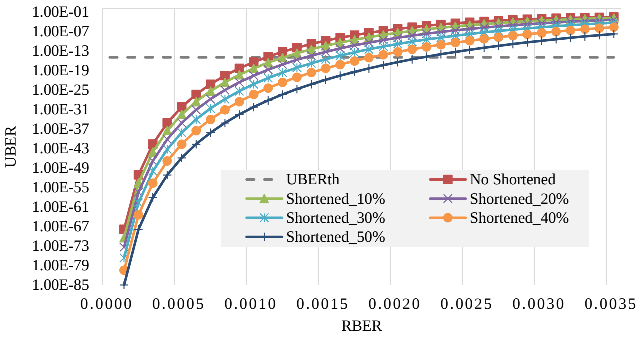


Figure 3: With increased shortened percentage, larger RBER can be tolerated. Where UBERth represents the UBER threshold.

With part of ECC-unit-size contents, the reliability can be improved with larger available RBER. Assume that the padding length is *l*. If the errors in prior *(k-l)*-length encoded data are corrected, the data integrity can be protected by ECC. Thus, the relationship between RBER and UBER becomes:

$$UBER = \frac{\sum_{m=t+1}^{n-l} m * pr(m, n-l, RBER)}{n} \qquad (3)$$

Here, we define shortened percentage as the fraction between padding length and the traditional encoded data length. And the shortened percentages of the encoding data are measured by varying from 0% to 50%. Figure 3 shows the reliability benefits with part of ECC-unit-size contents, where x-axis is the RBER and y-axis is the UBER. As shown in the figure, with the increasing of shortened percentages, the available RBER is increased. When the shortened percentage is 10%, the available RBER is increased by 12%. While with 50% shortened percentage, the available RBER can be improved by 100%. In this work, the opportunity motivates us to improve the reliability by shortening data.

4 MAPPING UNIT SHORTENED DESIGN

With the data shortened motivation, the reliability can be improvement by shortening data for each ECC unit, where a flash page basically contains multiple of ECC units. In the following, we propose to deploy the data shortened scheme to a 4KB-mapping unit flash memory device with a large page size, such as 32KB. This configuration is a popular setting for state-of-the-art flash device in smartphones and tablets, especially for 3D NAND flash. This is because that large page size is employed to reduce the circuit complexity and improve the I/O bandwidth [10] and finer-grained mapping (e.g. 4KB-mapping) is applied as the request size is usually smaller than the page size [10][9]. Note that the data shortened scheme can also be easily extended to page and block mapping devices without mapping issue due to the smaller request sizes [10][9], and we will discuss them in our future work.

4.1 Overview

In the mapping unit shortened design, once the ECC cannot support the reliability, one or more mapping units are shortened with receiving data. And the data shortened scheme is applied by dividing the received data to each ECC unit. Figure 4 shows an example of the overall structure of the proposed design. In the above half of the figure, assume there are four ECC units for one flash page. L_1 unit data are received for storage and L_2 units are shortened. In the bottom half of the figure, the data are evenly distributed to each ECC unit as encoded data. The paddings are filled with '1's as there are no data. Actually, the cells for paddings will be worn less than other cells [9], and the benefits will be exploited in the future work.

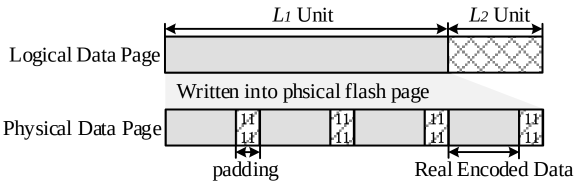


Figure 4: The overview of the mapping unit shortened design.

However, there are challenges on the design. The first one is the shortened data size, where the reliability of flash blocks get worse with the wearing, so that the shortened data size should be determined reliably. The second one is the space reduction, which is increased with the shortened data size, bringing bad effects on flash memory. In the following, the detailed design is first presented. And then, two space relaxation approaches are presented.

4.2 Detailed Design

The basic idea is to gradually shorten data in the granularity of mapping-unit for lifetime improvement. Shortened unit ratio decides the padding ratio, and then introduces related reliability improvement. The larger the shortened unit ratio is, the larger the reliability improvement will be. Here, we call flash page without data shortened is level 0, and with one mapping unit shortened is level 1, and so on. There are totally $\frac{S_{page}}{S_{mapping}}$ levels, and the last level is the flash page with only one mapping-unit received data, where the page size is S_{page}, and the mapping unit size is $S_{mapping}$. For 4KB-mapping with 32KB page size, there are totally 8 levels. In the following, the shortened level is represented as L_2, and the shortened level determination and related write and read processes are presented.

Shortened Level Determination: Initially, flash page is reliable without shortened data. Once the flash page becomes unreliable, the next level is applied to the flash page, and so on. The shortened unit level is determined when reading. During reading, pages in block are read out with acknowledging the error number. If all the error number of the block is less than a threshold, denoted as E_{th}, the current shortened unit level can be used reliability. Otherwise, the shortened unit should be increased to the next level for reliability improvement. If the current shortened level is $\frac{S_{page}}{S_{mapping}} - 1$, the block is regarded as bad. New shortened level will be used for the next round of usage for flash blocks. Each block only needs to maintain one shortened level, which is costless. Note, E_{th} is selected with a smaller number than error correction capability of ECC for conservation.

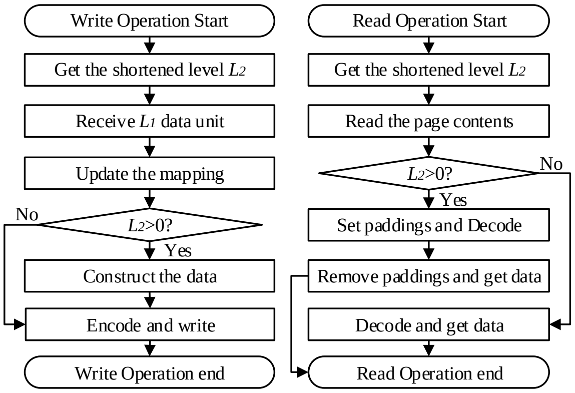


Figure 5: The related write and read processes.

Write Process: The write process is shown in Figure 5. When writing, the shortened level L_2 of the flash block is first got. According to the shortened level, L_1 unit data are received. Then, the mapping table is updated, where one-to-one mapping is used, and the prior L_1 mapping units are mapped with storing these data. Next, data will be encoded and written into flash pages. If the shortened level is 0, that is without data shortened, the data can be directly encoded and then written into flash pages. Otherwise, the data should be constructed first as shown in Figure 4. For constructing, the L_1 unit data are evenly divided to be stored into each ECC unit as real encoded data. For example, when page size is 32KB, mapping unit size is 4KB, ECC unit size is 1KB, and shortened level is 2, the 6 4KB-unit data will be divided, and each 0.75KB to be stored in each ECC unit.

Read Process: For finer-grained mapping flash, the request data is in the finer-grained unit. The read process is shown in Figure 5. When reading, the shortened level L_2 is got first. Second, the whole page data are read out. Then, the decoding and getting data operations are performed according to the shortened level. If the shortened level is 0, that is without data shortened scheme, these operations are directly performed. Otherwise, paddings will be set with all '1's before decoding. After decoding, the paddings are removed, and then the required data are got. Also with the configuration example in write process, the padding size is 0.25KB in each ECC unit which will be set and removed for decoding and getting data.

Expect the write and read operations, there are also operations requiring write or read operations, such as update, garbage collection, wear leveling and so on. For these operations, all data are written and read with related write and read processes.

4.3 Space Relaxation

Data shortened scheme will introduce space reduction for lifetime improvement, introducing user space concern. The user space can be maintained taking advantage of the invisible space, over-provision space (OPS). However, the OPS is usually configured with 15%∼25% of SSD capacity, which is also limited. On the other hand, the use frequency of OPS will become larger, including the introduced update number and garbage collection frequency [6]. Larger use frequency of OPS would affect the performance, though the dominate access pattern of the consumer-level devices is sequential access. In this work, besides the gradual shortened method, two physical characteristics of flash memory are exploited to further relax the space reduction: (1) the block endurance variation, (2) the differential reliability characteristics of the multiple bits in a cell.

4.3.1 Block Endurance Variation Awareness. With the technology scaling and density improvement, process variation (PV) has been a prominent characteristic of flash memory, especially for consumer-level devices. PV indicates that flash blocks have different endurance and larger than the industry data [13][18]. In this work, the variation can be exploited for space relaxation, where only weak blocks need to be shortened. To exploit PV, many measurements can be metrics, such as maximum P/E cycles [15][13], bit error rates (BER) [13][18][8], programming speed [16], and so on[18]. In this work, the maximum error criteria like RBER can work to exploit the reliability variations of flash blocks.

4.3.2 Differential Reliability of LSB and MSB. Multiple bits per cell flash memory has been dominant devices. And the differential reliability characteristics of the multiple bits is exploited here to further relax space reduction. Here, MLC flash memory is discussed, the method can be easily extended to more bits per cell flash memory, such as triple level cell (TLC). For MLC, the data shortened scheme is proposed to be only applied on MSB page. The relaxation approach is based on two observations. First, MSB pages are generally weaker than LSB page as shown in Figure 1. MSB page has larger possibilities to get errors than that of LSB page by almost 2 times [1]. Second, if MSB page is shortened, the reliability of the corresponding part of LSB page is also improved. As shown in Figure 6, when padding in MSB page are set to all '1's, the stored states in LSB page become only '11' and '01', two states. In that case, the possibility of getting errors is decreased almost to 0 for the corresponding contents in LSB page.

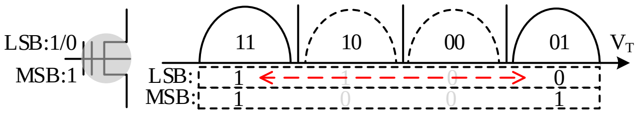


Figure 6: LSB data states with shortening MSB.

5 EXPERIMENT AND RESULTS

5.1 Experiment Setup

A 16GB flash chip is configured as the default storage device. Table 1 presents the parameters. For 32KB page size, 4KB is used as the mapping unit. Thus, there will be 8 shortened levels, and 4KB is the shortened data unit for each level. In flash controller, the low-cost ECC, BCH(17264,16400,57), will be applied.

Parameter	Value
Page/Block Size	32KB/2MB
Plane/Die Size	4GB/8GB
Total Chip Capacity	16GB

Table 1: The parameters for the evaluated consumer-level flash device.

5.1.1 Reliability Model.

In the evaluation, RBER is regraded as the reliability metric. The higher the available RBER is, the more reliable the flash memory is. To quantify the RBER, a reliability model is constructed based on error characteristics of flash memory.

There are two dominant factors on RBER, the wearing (w) and retention time (d), both are proportional to RBER. Through analyzing the measured error characteristics of real flash memory, we find that RBER is exponentially increased with w, and almost linearly increased with d. Furthermore, the RBER can be formulated as follows [14].

$$RBER(w, d) = d_r(w) \cdot d \tag{4}$$

Where d is assumed to be 10 years, and $d_r(w)$ was the deterioration rate, which can be represented as follows, where c and k both are constants.

$$d_r(w) = c \cdot w^k \tag{5}$$

For the wearing of blocks, P/E cycles and its endurance under PV are the main effects. Considering PV, the maximum P/E cycles vary among blocks. Assume that blocks are worn linearly, wearing can be represented as follows, where PE represents its P/E cycles and WD represents the wearing degree for each P/E cycle.

$$w = PE * WD \tag{6}$$

Combining the Formulas from 4 to 6, the reliability model is constructed.

In the evaluation, the default PV distribution on flash memory is assumed as normal distribution, N(8524,1318^2), for the maximum P/E cycles, while the factory index is just 3000 [18]. For each flash block, the wearing degree is the fraction between 3000 and the real maximum P/E cycles.

Four platforms are evaluated here, that is DS_Original, DS_PV, DS_MSB, and DS_ALL. DS_Original represents the gradual mapping unit shortened design. DS_PV takes the block endurance variations into consideration. DS_MSB only applies the data shorten scheme for the MSB page. And DS_ALL takes all relaxation approaches into consideration.

5.2 Experiment Results

In this subsection, the lifetime improvement under various shortened percentages is first presented. With the improvement of lifetime, the data amount under each space reduction is presented. At last, the performance effects are discussed.

Table 2 shows the wearing extension with data shortened scheme. As shown in the table, the wearing of flash can be extended with increasing shortened percentages. Compared with the baseline of 0%, when the mapping unit is shortened one by one, with shortened percentages of 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, the tolerated wearing can be improved by 7.8%, 18.2%, 31.5%, 49.6%, 77.2%, 125%, 238%. As wearing is increased with P/E cycles, improved tolerated wearing means improved lifetime.

Shortened Percentages	0%	12.5%	25%	37.5%
Tolerated wearing (w)	3000	3235	3545	3944
Shortened Percentages	50%	62.5%	75%	87.5%
Tolerated wearing (w)	4488	5317	6749	10142

Table 2: The wearing benefits under various shortened percentages.

With PV, the wearing degrees (WD) of flash blocks are different, so the block can achieve different P/E cycles, based on Formula 6. While the larger the space reduction is, the smaller the available space for data will be. To evaluate lifetime improvement, we simulate to erase blocks one-cycle-by-one and compute the corresponding written data amounts and space reductions. The written data amount represents the amount of data can be written to the flash memory, which is a popular metric in representing the lifetime. During the usage of flash memory, traditional wear leveling scheme is assumed to be applied. Here, without considering PV, the wearing ratio is regarded as 1. Otherwise, the real wearing ratio is taken into consideration considering PV. Figure 7 shows the normalized amount of written data under different space reduction ratios. From the results, several observations can be concluded: First, with the increases of space reduction ratios, the amount of data is significantly increased. Second, to achieve an increased written data amount, the proposed space relaxation approaches can relax the space reduction. In detail, to achieve 10% data amount increases, the DS_Original platform requires 23% space reduction, while the DS_PV requires only 4%, DS_MSB requires 11%, and DS_ALL only requires 2%. Besides, with considering PV, the space reduction ratio becomes smoothing. For DS_ALL platform, the amount of written data can be increased by 34% with only 12% space reduction.

Besides, we also make an evaluation on the performance effects. For evaluation, a widely used simulator, SSDsim [5], is used. The SSDsim is configured with traditional operation latencies shown in [17], and 15% OPS. We collected a typical workload for consumer-level device. The workload is to copy a movie from the computer to a Sandisk USB, which is collected

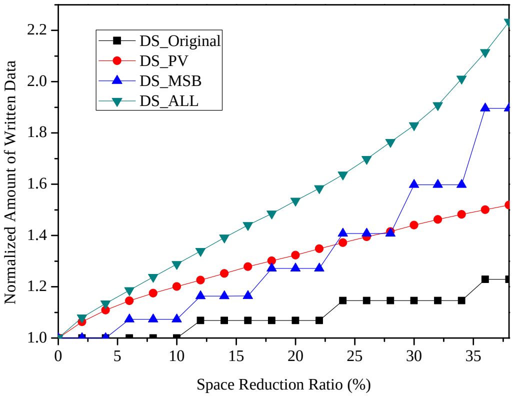


Figure 7: The normalized amount of written data under different space reduction ratios.

by Diskmon. Three mapping schemes are evaluated, including 4KB-mapping, page-mapping and block-mapping. For 4KB-mapping, the mapping is first deployed and then the 4KB-mapping unit shortened design is applied in the simulator. For page-mapping, the shortened design is implemented with receiving part of page-size data. Similarly for block-mapping, the mapping is deployed with only receiving part of block-size data. With given shortened percentage, the receiving data size can be got. As shown in Figure 8, with 12.5% shortened percentage, the write performance is slightly affected, by only 6% for page/block-mapping and 9% for 4KB-mapping, which is acceptable for consumer devices. From the results, we find that the performance effects of page/block-mapping are the same. This is because that all the data size is smaller than the block size, and the effects both exist in the additional page writing. In addition, there is almost no read performance differences as there are only a small amount of sequential read requests.

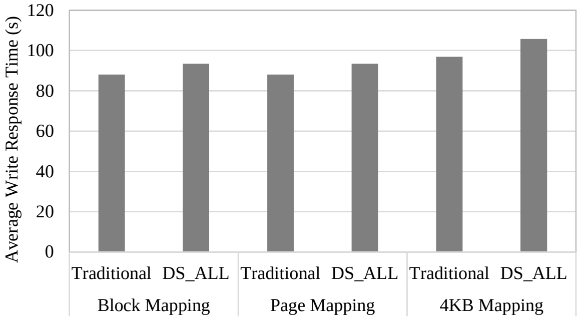


Figure 8: The performance effects.

6 CONCLUSIONS

The reliability of the low-cost enabled flash memory is a challenge. While we find that if encoded data are shortened in each ECC unit, the reliability can be significantly improved. With the motivation, a design deployed on 4KB-mapping flash is proposed to construct the motivation case for lifetime improvement. However, there are two challenges, that is the shortened data determination and space reduction. First, the design is proposed to gradually shortening 4KB-mapping unit with reliable shortened level, and the related write/read process are presented. Second, two space reduction relaxation approaches are proposed for relaxing large space reduction. At last, the experiment results show that the proposed work can improve the lifetime of flash memory significantly with negligible effects.

7 ACKNOWLEDGEMENT

This work is supported by the NSFC (61772092 and 61572411).

REFERENCES

[1] Yu Cai, Saugata Ghose, Erich F. Haratsch, Yixin Luo, and Onur Mutlu. 2017. Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives. *Proc. IEEE* 105, 9 (2017), 1666–1704.

[2] Yu-Ming Chang, Yung-Chun Li, Yuan-Hao Chang, Tei-Wei Kuo, Chih-Chang Hsieh, and Hsiang-Pang Li. 2015. On Relaxing Page Program Disturbance over 3D MLC Flash Memory *(ICCAD)*.

[3] Y. Di, L. Shi, K. Wu, and C. J. Xue. 2016. Exploiting process variation for retention induced refresh minimization on flash memory. In *DATE*. 391–396.

[4] H. Helgert and R. Stinaff. 1973. Shortened BCH codes (Corresp.). *IEEE Transactions on Information Theory* (1973).

[5] Yang Hu, Hong Jiang, Dan Feng, Lei Tian, Hao Luo, and Shuping Zhang. Performance Impact and Interplay of SSD Parallelism Through Advanced Commands, Allocation Strategy and Data Granularity. In *ICS'11*.

[6] Ping Huang, Guanying Wu, Xubin He, and Weijun Xiao. 2014. An Aggressive Worn-out Flash Block Management Scheme to Alleviate SSD Performance Degradation. In *EuroSys*.

[7] Xavier Jimenez, David Novo, and Paolo Ienne. 2012. Software controlled cell bit-density to improve NAND flash lifetime. In *DAC*.

[8] Xavier Jimenez, David Novo, and Paolo Ienne. 2014. Wear unleveling: improving NAND flash lifetime by balancing page endurance.. In *FAST*.

[9] Jung Hoon Kim, Sang Hoon Kim, and Jin Soo Kim. 2015. Subpage programming for extending the lifetime of NAND flash memory. In *DATE*. 555–560.

[10] Myungsuk Kim, Jaehoon Lee, Sungjin Lee, Jisung Park, and Jihong Kim. 2017. Improving Performance and Lifetime of Large-Page NAND Storages Using Erase-Free Subpage Programming. In *DAC*. 1–6.

[11] Han-Yi Lin and Jen-Wei Hsieh. 2015. HLC: Software-based half-level-cell flash memory. In *DATE*.

[12] T. Nakamura, Y. Deguchi, and K. Takeuchi. 2017. AEP-LDPC ECC with Error Dispersion Coding for Burst Error Reduction of 2D and 3D NAND Flash Memories. In *IMW*. 1–4.

[13] Yangyang Pan, Guiqiang Dong, and Tong Zhang. 2013. Error Rate-Based Wear-Leveling for nand Flash Memory at Highly Scaled Technology Nodes. *TVLSI* 21, 7 (2013), 1350–1354.

[14] Heejin Park, Jaeho Kim, Jongmoo Choi, Donghee Lee, and Sam H. Noh. 2015. Incremental redundancy to reduce data retention errors in flash-based SSDs. In *MSST*.

[15] Roman A. Pletka and Saša Tomić. 2016. Health-Binning: Maximizing the Performance and the Endurance of Consumer-Level NAND Flash *(SYSTOR)*.

[16] L. Shi, Y. Di, M. Zhao, C.J. Xue, K. Wu, and E.H.-. Sha. 2015. Exploiting Process Variation for Write Performance Improvement on NAND Flash Memory Storage Systems. *TVLSI* PP, 99 (2015).

[17] D. Wei, L. Deng, L. Qiao, and P. Zhang. 2016. PEVA: A Page Endurance Variance Aware Strategy for the Lifetime Extension of NAND Flash. *TVLSI* (2016).

[18] Yeong-Jae Woo and Jin-Soo Kim. 2013. Diversifying wear index for MLC NAND flash memory to extend the lifetime of SSDs. In *EMSOFT*. 1–10.

[19] Guanying Wu, Xubin He, Ningde Xie, and Tong Zhang. 2013. Exploiting Workload Dynamics to Improve SSD Read Latency via Differentiated Error Correction Codes. *ToDAES* (2013).

[20] Ningde Xie, Guiqiang Dong, and Tong Zhang. 2011. Using lossless data compression in data storage systems: Not for saving space. *TC* (2011).

[21] Ming-Chang Yang, Yuan-Hao Chang, Chei-Wei Tsao, and Chung-Yu Liu. 2016. Utilization-Aware Self-Tuning Design for TLC Flash Storage Devices. *TVLSI* (2016).

[22] L. Zhang, B. Neely, D. Franklin, D. Strukov, Y. Xie, and F. T. Chong. 2016. Mellow Writes: Extending Lifetime in Resistive Memories through Selective Slow Write Backs. In *ISCA*.

[23] Kai Zhao, Wenzhe Zhao, Hongbin Sun, Tong Zhang, Xiaodong Zhang, and Nanning Zheng. 2013. LDPC-in-SSD: making advanced error correction codes work effectively in solid state drives. In *FAST*.