DREAM: Data Representation Aware of Damage to Extend the Lifetime of MLC NAND Flash Memory

Ting Ye[‡], Shenggang Wan[♯,*], Xubin He[§], Weijun Xiao[†], Changsheng Xie[‡]
[‡]*Wuhan National Laboratory for Optoelectronics, Huazhong University of Sci. and Tech., China*
[♯]*School of Computer Science and Technology, Huazhong University of Sci. and Tech., China*
[§]*Department of Computer and Information Sciences, Temple University, USA*
[†]*Virginia Commonwealth University, USA*
{*tingye, sgwan*}*@hust.edu.cn, xubin.he@temple.edu, wxiao@vcu.edu, cs_xie@hust.edu.cn*

Abstract

MLC NAND flash memory uses the voltages of the memory cells to represent bits. High voltages cause much more damage on the cells than low voltages. The free space that need not store bits is leveraged to reduce the usage of those high voltages and thus extend the lifetime of the MLC memory. However, limited by the conventional data representation rule that represents bits by the voltage of one single cell, the high voltages are still used in a high probability. To fully explore the potential of the free space on reducing the usage of high voltages, we propose a novel data representation aware of damage, named DREAM. DREAM uses the voltage combinations of multiple cells instead of the voltage of one single cell to represent bits. It enables to represent the same bits through flexibly replacing the high voltages in some cells with the low voltages in other cells when free space is available. Hence, high voltages which cause more damage are less used and the lifetime of the MLC memory is extended. Theoretical analysis results demonstrate the effectiveness and efficiency of DREAM.

1 Introduction

MLC (Multi-Level Cell, which stores two bits in one cell) NAND flash memory cells suffer from exponentially increasing damage with the growth of the voltage which they are charged to [1, 2]. The potential inside this phenomenon is not fully explored. Hence, a great opportunity to extend the poor lifetime of MLC memory, which is its serious drawback, is missed.

An MLC cell resembles an MOS transistor having two gates: a control gate, and a floating gate surrounded by a thin oxide layer [3]. Isolated by the oxide layer, the floating gate can trap electrons. Four non-overlapped threshold voltage windows (denoted by V_{ML}, which can be V_{11}, V_{10}, V_{00}, or V_{01} from low to high voltage) are formed by the trapped electrons to represent two bits (a most significant bit (MSB) and a least significant bit (LSB)). All MSBs and LSBs in a wordline (a group of cells) form an MSB page and an LSB page, respectively.

Program operations charge cells to modify their voltages; while erase operations remove electrons from cells. The P/E (Program/Erase) operations can significantly damage the oxide layer, making it easier to lose electrons and finally leading to raw bit errors. Therefore, we use raw bit error rates (RBER) as an indicator to represent the damage on MLC cells in this paper.

Most of the damage is caused by high voltages [1], and previous research discovers the damage is cumulative [4]. Hence, solutions are proposed to reduce the usage of high voltages by leveraging the free space that need not to store bits, and can be gained by data compression [5, 6] or found in the out-of-band (OOB) space [7] of wordlines. Via distributing the free space in single cells, i.e., less bits are stored in those cells, the usage of high voltages is reduced and thus leading to an overall low damage.

However, according to our analysis, the potential inside the free space is not fully explored and thus losing a great opportunity to further reduce the overall damage. More specifically, in those conventional approaches, one bit of the free space is only used to eliminate the usage of high voltages in one single cell. Nevertheless, in the meanwhile, the remaining cells which store more bits still suffer from the high voltages. If one bit of the free space is shared by multiple cells, much more cells will not suffer from the high voltages, and thus further reducing the overall damage.

In essence, the key obstacle of sharing one bit of the free space among multiple cells is the conventional data representation rule that represents bits by the voltage of one single cell. By breaking this rule, i.e., using the voltage combinations of multiple cells to represent bits, the high voltages of some cells can be replaced with the

*Corresponding author: sgwan@hust.edu.cn

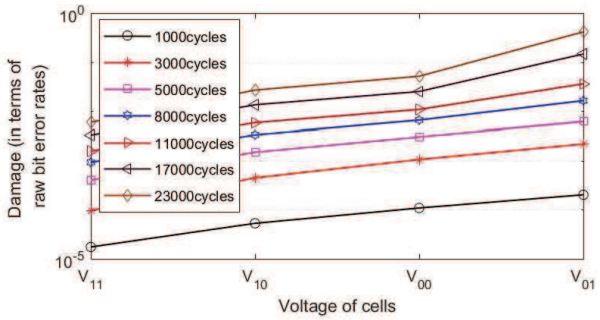

Figure 1: The damage of four voltages under various P/E cycles.

low voltages of other cells to represent the same bits, and thus further reducing the usage of the high voltages.

Motivated by the above analysis, we propose a novel data representation aware of damage, called *DREAM*, to reduce the usage of high voltages in MLC memory via efficiently leveraging the free space. More specifically, in DREAM, bits are represented by the voltage combinations of multiple cells (a cell group) instead of the voltage of one single cell. Through distributing free space in each cell group, and using the lowest damage voltage combinations to represent the bits stored in those cell groups, the usage of high voltages and thus the overall damage can be reduced.

Our theoretical analysis demonstrates the effectiveness and efficiency of DREAM. By using DREAM, the lifetime of MLC memory is extended by 12.5% to 60.8% or 25.8% on average in terms of P/E cycles under various scenarios, compared to the state-of-the-art solution based on adjusting the data layout to extend the lifetime of MLC memory [4].

2 Background and Motivation

P/E operations induce voltage dependent damage on NAND flash memory cells and thus cause raw bit errors. The relevance among the voltage, P/E cycles, and raw bit error rates has been heavily investigated by previous researches [2, 4]. These researches reveal that: (1) the P/E damage on a cell exponentially increases with the growth of the voltage levels the cell charged to as shown in Figure 1; and (2) the P/E damage is cumulative.

Assume all the voltages appear at the same probability. Most of the damage is caused by the high voltages. To clearly illustrate that, consider the following case. Assume the MSB and LSB in a cell are randomized, i.e., 50% chance being 0 or 1, and the normalized damages of V_{11}, V_{10}, V_{00}, and V_{01} are 1, 2.68, 7.34, and 18.2 (derived from the raw bit error rates at 8000 P/E cycles in Figure 1), respectively. The average damage on the cell is about 7.31. In addition, the high voltage V_{01} is applied in a chance of 25%, which contributes $\frac{18.2\times0.25}{7.31}$ or 62.2% of the average damage.

Intuitively, if the usage of high voltages, e.g., V_{01}, can be reduced, the average damage will be significantly reduced. Fortunately, free space that need not to store bits are usually abundant in flash storage, e.g., can be gained by data compression, or found in the OOB space in a wordline. By distributing the free space among cells, the usage of high voltages is reduced and thus leading to less damage to the flash.

According to our analysis, the potential inside the free space has not been fully explored by those aforementioned solutions. Consider a simple case of storing only three bits in two MLC cells C_1 and C_2, i.e., one bit is free and unused. Traditionally, the best storage layout to leverage that unused bit is to store two bits in C_1, and one bit in C_2 as the LSB, respectively. Assume all those three bits are randomized. Because V_{00} and V_{01} are not applied in C_2, the average damage on C_2 is $(1+2.68)/2 = 1.85$. Overall, the average damage on one cell is about $(7.31+1.85)/2 = 4.58$.

As shown in the above example, even in the best traditional storage layout, high voltages are still used in a high probability in some cells, e.g., in C_1, V_{01} with the highest damage would be applied in a chance of 25%, and thus still contribute about $\frac{18.2\times25\%}{2\times4.58} = 49.7\%$ to the average damage; in the meanwhile, some low voltages are not used in other cells, e.g., in C_2, V_{00}, whose damage is lower than V_{01}, is not applied. If V_{01} in C_1 can be replaced by V_{00} in C_2 to represent the same bits, the usage of V_{01} can be further reduced.

According to our analysis, the main obstacle to do the replacement is that the unused bit can only be placed in C_1 or C_2, but not both of them. The key constraint to prevent the unused bit being shared by C_1 and C_2 is the conventional data representation rule that represents bit by the voltage of one single cell. By breaking this rule, the replacement can be easily done, which motivates us to propose our DREAM. By applying proposed DREAM, we can reduce the average damage from 4.58 to 3.22. We will show the analysis in the next section.

3 Design of DREAM

To reduce the usage of high voltages and thus reduce the overall damage suffered by the MLC NAND flash memory, we propose a novel data representation aware of damage, called *DREAM*. The key idea of DREAM is using the voltage combinations of multiple cells instead of the voltage of one single cell to represent bits through breaking the conventional data representation rule. In such a manner, the high voltages in some cells can be replaced by the low voltages in other cells to represent the same bits when free space is gained, and thus further reduce the usage of high voltages compared to the conventional solutions.

The conventional approaches gain the free space using data compression. Two approaches are proposed to place the compression pages. One is putting two compressed pages in a wordline which is known as *implicit compression* and the other is putting as many compressed pages in a wordline as possible which is known as *explicit compression*. Compared to the latter approach, the first one usually suffers from much less FTL complexity which is very important to flash memory. Hence, we focus on applying DREAM into implicit compression in this paper, although DREAM can also work well on explicit compression.

To apply DREAM, at first, the content of write requests will be compressed to gain free space. Then, the compressed data and the metadata, e.g., ECC checksums, should be encoded to their damage-aware representation and additional metadata is generated for decoding. Those additional metadata, and DREAM-coded data and metadata are assembled to page pairs with dedicated layout. Finally, the page pair is stored in a wordline as the MSB and LSB pages, respectively. To read DREAM-coded data, both the MSB and LSB pages of its host wordline should be fetched. Then, DREAM-coded data and metadata are decoded by using the additional metadata. To reduce the modification on the hardware and firmware, we only consider to conduct the encoding and decoding operations on memory controllers in this paper.

3.1 Data Representation Aware of Damage

Table 1: Bit value mapping while storing 1, 2, or 3 bits in 2 cells using DREAM where lowest voltage combinations are used. V_1 and V_2 are the voltages of C_1 and C_2, respectively. '-' means the corresponding combinations are not used.

#Bits / V_1 V_2	1	2	3	#Bits / V_1 V_2	1	2	3
$\mathbf{V_{11}}$ $\mathbf{V_{11}}$	1	11	011	$\mathbf{V_{11}}$ $\mathbf{V_{00}}$	-	-	111
$\mathbf{V_{11}}$ $\mathbf{V_{10}}$	0	10	010	$\mathbf{V_{00}}$ $\mathbf{V_{11}}$	-	-	101
$\mathbf{V_{10}}$ $\mathbf{V_{11}}$	-	01	001	$\mathbf{V_{10}}$ $\mathbf{V_{00}}$	-	-	110
$\mathbf{V_{10}}$ $\mathbf{V_{10}}$	-	00	000	$\mathbf{V_{00}}$ $\mathbf{V_{10}}$	-	-	100

In the DREAM-enhanced memory, every m cells in a wordline form a group. All the m voltages of a group, which are denoted by an m-tuple $(V_1, V_2, ..., V_j, ..., V_m)$ (V_j is the voltage of the jth cell in the group, and can be V_{11}, V_{10}, V_{00}, or V_{01}), together represent i bits. Because each cell has four choices on voltages, all m voltages form 4^m different m-voltage combinations, thus represent up to 4^m (or 2^{2m}) states, i.e., a group can store up to $2m$ bits. The groups can be divided into $2m+1$ types, named with T_i, where i is the number of bits they store. To make the representation damage-aware, for the T_i group, the 2^i lowest damage voltage combinations among 2^{2m} states are selected to represent those i bits. We record this DREAM deployment as DREAM-m, i.e., m cells form a cell group. In this paper, for demonstration purpose without loss of generality, we focus on DREAM-2 where 2 cells in a wordline form a group. Table 1 shows an example to represent different number of bits with DREAM-2 using two cells C_1 and C_2. Two cells can store up to 4 bits. When storing 1, 2, or 3 bits, our DREAM always chooses the lowest voltage combinations. When storing 0 or 4 bits, DREAM uses the traditional mapping which is not shown in this table.

Consider the same case of the example mentioned in Section 2 that three bits are stored in two cells (i.e., $i = 3, m = 2$). Those bits are represented by the voltage combinations of two cells C_1 and C_2. As shown in Table 1, the 2^3 lowest damage voltage combinations among 2^4 states are selected to represent these three bits. Based on this data representation, the average damage on a cell is reduced to 3.22 by applying normalized damages as the example in Section 2. In other words, the average damage is about 29.7% reduction compared to the best case in traditional representation. The reason for this damage reduction is that we do not use the highest voltage V_{01} and apply DREAM to represent bits.

To store i bits in a group, we should first find their corresponding voltage combination $(V_1, ..., V_j, ..., V_m)$. Then, each voltage V_j is explained as two bits in the conventional way, i.e., an MSB and an LSB, e.g., if V_j is $V_{M_jL_j}$, the MSB and LSB will be M_j and L_j, respectively. At last, the m bits $M_1...M_m$ and m bits $L_1...L_m$ are stored in the group as the MSBs and LSBs. In DREAM, the conversion from original i bits to a pair of m bits is an encoding operation. And the reverse conversion is a decoding operation. The encoding and decoding operations can be implemented with either logic computation, or table looking-up. Our prototype on a commodity CPU shows that, the throughput of encoding ranges from 1.4 to 1.7 GB/s and that of decoding ranges from 1.8 to 2.2 GB/s. Because the FPGA-based decoder and encoder are typically much more powerful compared to that built upon CPU, the encoding and decoding operations would have negligible impact on the I/O performance in those DREAM-enhanced MLC memory.

When the voltage combinations are used to represent data, raw bit errors might be amplified. For example, if (V_{10}, V_{00}) represents 110, and (V_{10}, V_{10}) represents 000, two bit flips might occur when V_{00} changes to V_{10} due to electron leakage. To mitigate the amplification of raw bit errors, we come up the following solutions.

First, through generating the ECC checksum from the DREAM-coded data but not the original data, the error amplification will not happen to the data but only happen to the DREAM-coded metadata. Because the amount of the metadata is usually far less than the data (around 10%) [8], the probability of the error amplification will

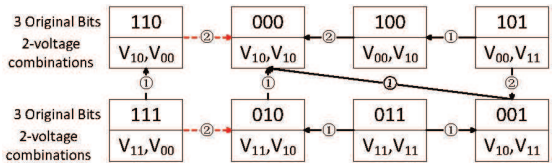

Figure 2: Voltage state transitions in the T_3 group of DREAM-2. The solid arrow indicates voltage state transition with one-bit data flip. The dotted arrow indicates voltage state transition with two-bit data flip. ① indicates adding additional electrons and ② indicates electron leakage.

be dramatically reduced.

Second, the error amplification can be further reduced by carefully choosing a mapping between bits and their corresponding voltage combinations. Generally raw bit errors are induced by retention, program interference, read disturb, and insufficient erases and almost all the raw bit errors are caused by the first two factors [1]. These raw bit errors demonstrate a strong statistic regularity: 90% of the retention errors are induced by the cell voltage state transition from V_{01} to V_{00} or from V_{00} to V_{10} due to electron leakage; and 70% of the program interference errors are induced by the cell voltage state transition from V_{11} to V_{10} by adding additional electrons to flash cells [1]. The error amplification will be dramatically reduced if those voltage state transitions would only flip one bit in all original bits. For example, Figure 2 demonstrates the voltage state transitions in the T_3 group of DREAM-2. We assume that only one voltage state in a 2-voltage combination will make a transition to other voltage states because there is less chance to happen for changing two voltage states at the same time. As shown in Figure 2, there are totally 10 possible transitions. Among these 10 transitions, only two transitions will induce two-bit data flips illustrated with dotted arrows in the figure. The rest of transitions do not cause any error amplification as traditional data representation. In this way, the amplification of bit error is limited to 1.2 (12 flipped bits after using DREAM v.s. 10 flipped bits under traditional representation).

3.2 Data Layout

In DREAM, for different types of groups, one certain voltage combination $(V_1,...,V_m)$ might represent different bits, e.g., in DREAM-2, (V_{11}, V_{11}) represents 11 in a T_2 group and 011 in a T_3 group. Hence, additional metadata should be added to identify the type of groups for correct decoding. Specifically, $\lceil log_2(2m+1) \rceil$ bits are used to record the type of a group. To reduce the additional metadata, in a wordline, all the T_i groups are placed together, and form zones (denoted by Z_i). Because the types of groups are usually very small, e.g., $2m+1$, the amount of metadata would not be large.

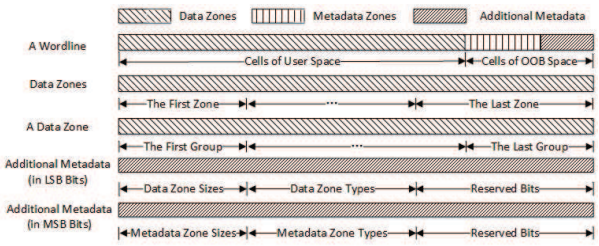

Figure 3: Data layout of a wordline. The DREAM-coded data and metadata are stored separately, and additional metadata are added for correct decoding.

The zones are placed in a sequence and there are no free cells between two adjacent zones. Hence, we only need to know the start address of the first zone Z_0 (a cell number), and the size of each zone (numbers of groups, can be 0), to identify the zones. It is worth noting that the starting address can be calculated from the P/E cycles of its host block [9, 10]. These additional metadata can be further reduced by aligning the size of zones, e.g., making them 2^s-cells aligned. In such a manner, the zone sizes can be further shortened, with the cost of a slightly increasing damage.

Different types of groups suffer different damages. If the starting address of the first zone is fixed, there will be a huge wear unbalance among the cells in a wordline. To balance the damage among the cells in a wordline, that start address will be changed in every P/E cycle, e.g., cyclically added by 2^n (cells). This method also works on the imbalanced damage among the cells in a group.

For the DREAM-coded data, the size of zone Z_i (denoted by S_i, where $0 \le i \le 2m$) is determined by the overall compression ratio r, the number of cells for storing data (user space) N_d in a wordline, and the per P/E cycle average damage caused by the T_i group is D_i. S_i should obey the following two constraints. (1) The total capacity of all the zones should be large enough to store all the data, as illustrated in Equation 1. (2) The total cells in all the zones should not exceed the cells of user space in a wordline, as illustrated in Equation 2. (3) Let p be the number of P/E cycles. The total damage after p P/E cycles on user space is calculated as Equation 3. Hence, S_i can be solved to minimize the total damage through the simplex method [11].

$$\sum\left(\frac{S_i \times 2^s}{m} \times i\right) \ge 2 \times r \times N_d \tag{1}$$

$$\sum(S_i \times 2^s) \le N_d \tag{2}$$

$$total\ damage = \sum\left(\frac{S_i \times 2^s}{m} \times D_i \times p\right) \tag{3}$$

$$in\ all\ the\ above\ three\ equations,\ i = 0,1,2,...,2m$$

DREAM is also applicable to the metadata, e.g., ECC checksums, because the metadata size is usually smaller than that of the out-of-band space in a wordline [12], i.e.,

additional free space is typically available. It is worth noting that, to speed up the decoding, the additional metadata for both the user space and OOB space is not represented in DREAM, and be placed after the last zone of metadata. We demonstrate the data layout of a wordline in Figure 3.

4 Evaluation

We conduct a mathematical analysis to evaluate the efficiency of DREAM in this section.

4.1 Analysis Methodology

In this work, we define the lifetime as the maximum P/E cycles before reaching the specified bit error rates (BER) under different compression ratios.

The maximum P/E cycles are related to the size of free space, which is determined by data compression ratio r and the size of metadata in OOB space. The maximum P/E cycles are estimated as follows. First, we calculate the zone sizes of user space under a certain data compression ratio according to the method described in Section 3.2. Then we calculate the zone sizes of OOB space under a certain metadata size S_m. We figure out the total damage suffered by user space and OOB space (referred as D_u and D_o, respectively) derived from the Equation 3 and the per-cycle damage of each group type derived from the raw bit error rates at 8000 P/E cycles in Figure 1. Let S_u and S_o denote the size of user space and OOB space, respectively, the overall damage suffered by the page is calculated as $\frac{D_u+D_o}{S_u+S_o}$, which is related to the number of P/E cycles p. Finally, we gradually increase p until the overall damage reaches the specified bit error rates (BER), and the maximum P/E cycles under this r and S_m is p. To calculate the maximum P/E cycles, we just need to change S_m and r.

We assume that the OOB space is set as 20% of the page size and the raw bit error rate of the MLC memory at its end of life is 6.6×10^{-3} [4], which is the average raw bit error rates of the four different cell voltages at 8000 P/E cycles. The metadata size S_m ranges from 5% to 15% of the page size and the data compression ratio r ranges from 0.1 to 0.9.

4.2 Results and Discussion

The maximum P/E cycles is chosen as the metrics for the comparison and the BDC [4] is chosen as the baseline. Figure 4 describes the lifetime of the MLC memory under the DREAM-2 which is normalized to that under the baseline. The results show that compared to the baseline, DREAM significantly improves the lifetime of the MLC memory. The improvement of DREAM ranges from 12.5% to 60.8%, and is 25.8% by geometric mean, compared to the baseline. DREAM demonstrates higher efficiency on improving the lifetime of MLC memory

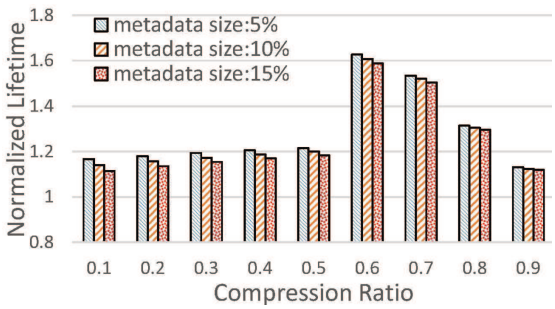

Figure 4: Normalized lifetime increase of DREAM-2 under various compression ratios and metadata sizes.

under the compression ratio from 0.6 to 0.8, because the high voltage V_{01} is used in a high probability (at least 20%) under baseline but in a low probability (at most 10%) under DREAM-2 in such cases.

When the size of metadata increases from 5% to 15% of page size, the benefit of lifetime decreases. This is because the size of free space of OOB reduces when the metadata increases, less cells can be benefited.

We also evaluate the impact to system throughput and overall response time. When DREAM is used in storage system, the system overall response time increased by 6.4% and the throughput decreased by 5.0%.

The impact to system performance can be further reduced if DREAM is only used in specified pages. Since logic pages are either read mostly or written mostly, and a few pages are read and written equally [13, 14], we can separate the read-mostly pages and write-mostly pages, and only apply DREAM in the write-mostly pages.

5 Conclusions and Future Work

In this paper, we present a data representation aware of damage, named DREAM, to reduce program/erase damage on MLC memory. The basic idea of DREAM is using the low-damage voltage combinations of multiple MLC cells instead of the voltage of one single MLC cell to represent bits, and discarding high-damage voltage combinations when free space are available. The benefits of this scheme is that flash cell damage is reduced and thus improving the lifetime of the MLC memory. Theoretical analysis results have demonstrated the effectiveness of DREAM in terms of improved P/E cycles compared to the state-of-the-art solution based on adjusting the data layout to extend the lifetime of MLC memory under the *implicit compression* strategy.

We consider reducing the read overhead induced by DREAM and combining DREAM with *explicit compression* strategy as our future work.

This research is sponsored by National Natural Science Foundation of China Grants Nos. 61300046 and 61331010, and U.S. National Science Foundation Grants CCF-1717660, CNS-1702474 and CCF-1547804.

References

[1] Yu Cai, Erich F. Haratsch, Onur Mutlu, and Ken Mai. Error Patterns in MLC NAND Flash Memory: Measurement, Characterization, and Analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '12)*, 2012.

[2] Neal Mielke, Todd Marquart, Ning Wu, Jeff Kessenich, Hanmant Belgal, Eric Schares, Falgun Trivedi, Evan Goodness, and Leland R Nevill. Bit error rate in NAND flash memories. In *Proceedings of the IEEE International Reliability Physics Symposium (IRPS'08)*, 2008.

[3] Rino Micheloni, Luca Crippa, and Alessia Marelli. *Inside NAND Flash Memories*. 2010.

[4] Jiangpeng Li, Kai Zhao, Xuebin Zhang, Jun Ma, Ming Zhao, and Tong Zhang. How Much Can Data Compressibility Help to Improve NAND Flash Memory Lifetime? In *Proceedings of the Usenix Conference on File and Storage Technologies (FAST'15)*, 2015.

[5] Yangwook Kang and Ethan Miller. Adding aggressive error correction to a high-performance compressing flash file system. In *Proceedings of theSeventhACMInternationalConferenceonEmbedded Software (EMSOFT 09)*, 2009.

[6] Y. Park and J. s. Kim. zFTL: power-efficient data compression support for NAND flash-based consumer electronics devices. *IEEE Transactions on Consumer Electronics*, 2011.

[7] Youyou Lu, Jiwu Shu, Weimin Zheng, et al. Extending the Lifetime of Flash-based Storage through Reducing Write Amplification from File Systems. In *Proceedings of the Usenix Conference on File and Storage Technologies (FAST'13)*, 2013.

[8] Kai Zhao, Wenzhe Zhao, Hongbin Sun, Tong Zhang, Xiaodong Zhang, and Nanning Zheng. LDPC-in-SSD: making advanced error correction codes work effectively in solid state drives. In *Proceedings of the Usenix Conference on File and Storage Technologies (FAST'13)*, 2013.

[9] Eran Gal and Sivan Toledo. Algorithms and Data Structures for Flash Memories. *ACM Comput. Surv.*, 2005.

[10] Avraham Ben-Aroya and Sivan Toledo. Competitive Analysis of Flash Memory Algorithms. *ACM Trans. Algorithms*, 2011.

[11] George B Dantzig. *Application of the simplex method to a transportation problem*. 1951.

[12] C. Sun, K. Miyaji, K. Johguchi, and K. Takeuchi. SCM capacity and NAND over-provisioning requirements for SCM/NAND flash hybrid enterprise SSD. In *Proceedings of the 2013 5th IEEE International Memory Workshop (IMW'13)*, 2013.

[13] Dushyanth Narayanan, Austin Donnelly, and Antony Rowstron. Write off-loading:Practical power management for enterprise storage. *ACM Transactions on Storage (TOS'08)*, 2008.

[14] Qiao Li, Liang Shi, Chun Jason Xue, Kaijie Wu, Cheng Ji, Qingfeng Zhuge, and Edwin H.-M. Sha. Access Characteristic Guided Read and Write Cost Regulation for Performance Improvement on Flash Memory. In *Proceedings of the Usenix Conference on File and Storage Technologies (FAST'16)*, 2016.