Microscopic Bit-Level Wear-Leveling for NAND Flash Memory

Yong Song[1], Woomin Hwang[1], Ki-Woong Park[2], and Kyu Ho Park[1]

[1] CORE Lab., Dept. of Electrical Engineering, KAIST, Korea
[2] Dept. of Computer Hacking and Information Security, Daejeon University, Korea
{ysong,wmhwang,kpark}@core.kaist.ac.kr, woongbak@dju.kr

Abstract. By microscopically observing widely used data files, we identified the considerable room for life time improvement in NAND flash memory, which is due to the discovery of a non-uniformity in bit-level data patterns. In an attempt to exploit the discovery, we propose a novel bit-level wear-leveling scheme. Instead of considering only the view of page-level or block-level, we incorporate the non-uniformity in data encoding patterns into wear-leveling scheme. Because of its orthogonality to the existing block-level wear-leveling approaches, our solution can be adopted over the existing solutions without considerable overhead and extend NAND flash's life span up to 36% in case of SLC.

1 Introduction

During the last decade, NAND flash memory has become a de facto solid state storage technology. In the NAND flash memory-based storage systems, it is essential to reliably store data for years, however, NAND flash has endurance problem that each NAND flash cell is worn-out by program/erase cycling and eventually loses its capability to store data. Thus, the endurance problem of NAND flash has been considered in many previous researches and they proposed wear-leveling algorithms, which even out the wearing of different blocks of NAND flash memory.

To enlarge NAND flash's life span, many researches proposed their algorithms for wear-leveling which tries to make wear-out of different blocks of flash memory even [3] [4] [5] [6] [7]. To expand lifetime of NAND flash blocks, commonly, the previous algorithms take page-granularity approaches and use block erase count as a wear-out indicator.This is based on an implcit assumption all cells in same page or same block have the same probability to be worn-out by P/E cycle. To identify whether the assumption is right or not, we obtained data patterns of widely used file data frequently written to disk or accounting for large part of storage, for example, linux source codes, database workloads, encoded video files. After analyzing the data, we found out a case that the bit-level data pattern is not uniform. Base on our observation, we build a simple statistical wear-out model of NAND flash cell and we propose bit-level wear-leveling scheme. The evaluation shows that it expands life span of NAND flash up to 30%.

James J. (Jong Hyuk) Park et al. (eds.), *Mobile, Ubiquitous, and Intelligent Computing*, Lecture Notes in Electrical Engineering 274,
DOI: 10.1007/978-3-642-40675-1_48,

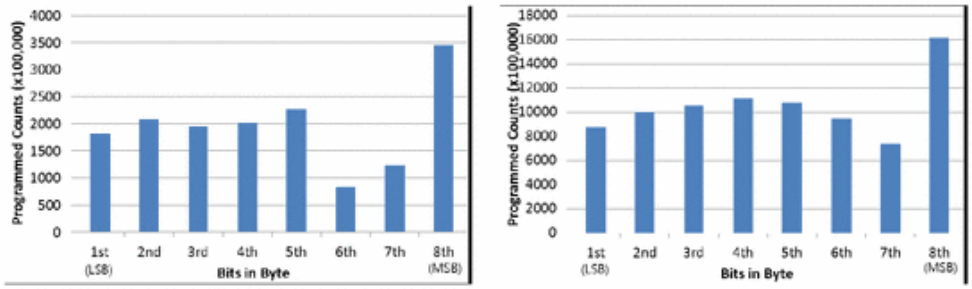


Fig. 1. (a) Bitwise counts of programmed state of Linux source codes, (b)Bitwise count of programmed state of DB workloads

2 Motivation

In case of SLC NAND flash, a cell stores a bit and a page is composed of a series of NAND cells and a block contains multiple pages. To store data on them, a block erase operation should be performed prior to write data, which makes all cells in the block changed to the erased state, representing logic '1'. If logical value of '0' is written to a cell when page writing operation of NAND flash, a state transition of the cell occurs from the erase state to the programmed state, indicating '0'. On the other hands, if logical value of '1' is written to a cell, there is no state transition of the cell and it remains in the erased state. Because a state transition from erase state to programmed state makes a cell worn out [2], each cell can experience different wear-out progress accoriding to the number of its actual programs.

A problem arises from that the previous block-based wear-leveling algorithms treat all cells comprising a block together as if they experience the same number of state transitions between the erase state and the programmed state, thus having the same degree of wear-out. If we assume that the physical endurance of each cell is same, among all cells comprising a block, the cell experiencing the highest number of programs is to be completely worn out most rapidly. If the number of completely worn-out cells is bigger than the number of correctable bit errors by bit error correction algorithm such as ECC or DHC, it is regarded that the life time of the page is ended. Even though other cells still have their life time left and they can operate normally, the page is not used any more. This leads to the underestimation of the life time of the target page. It therefore makes the NAND flash memory block lose chances to prolong lifetime of the block.

Outstanding freqeuncy of programs in a specific bit position appears significantly in text-based workloads. We selected Linux source codes as a representative text-based workloads and obtained their bit-level data patterns by traversing Linux source code tree. Figure 1(a) shows the accumulated counts of the case of logical '0' at each bit position per byte, which requires the state transition of the cell from erase state to the programmed state. We can see that the most significant bit (MSB) position in a byte(character) experiences the highest number

of the state transtion than other bit positions. This is because ASCII code for human readable data does not use MSB.

In order to observe data patterns of files to be freqently written to disk, we performed DBT2[1] benchmarks for SQLite[8], an open-source database. The test creates database workloads and simulate heavy user loads for OLTP. We log all of the contents synchronously written to disk through the test. We can see that the bit-level data pattern of the result obtained from the benchmark test is similar to the previous one, as shown in Figure 1(b). Consequently, the difference in degree of wear-out of each cell would be apart more as the ratio of ASCII characters increases and it provides us rooms for prolonging lifetime of the target NAND block.

3 Proposed Scheme

In this section, we present a bit-level wear-leveling scheme based on a simplified statistical model to span the life time of NAND flash memory, which is derived from the observation of the data non-uniformity as described in Section 2. First, we define the fundamental wear-out model of SLC cell, in accordance with the writing pattern of the data. Then, we present the proposed bit-level wear-leveling schemes to make the degree of wear-out of cells uniform.

Our Assumption: Our model is basically derived from the concept of the charge-to-breakdown, Q_{BD}[2]. The Q_{BD} is defined to amounts of charges crossing the tunnel oxide in a NAND flash cell until the oxide is broken down. It is from the following physical characteristics: the oxide of each NAND flash cell is little by little worn-out by repeated program/erase cycling, and the cell eventually loses its capability to store data. Base on the above, the following assumption is obtained. The degree of wear-out of a cell after a P/E cycling is proportional to the amount of charge passing the oxide in the cell and the number of injected and ejected charges during a cycling is always same, and threshold voltage for sensing stored data is also propotional to the amount of stored charges. This assumption is a base of the wear-out modeling for a SLC NAND cell.

SLC can store only one bit value per cell, which basically is a threashold voltage level. Typically, in NAND flash, a logical bit value '1' of an SLC cell denotes the erased state, and a logical bit value '0' denotes the programmed state. W_{slc} is defined to the degree of worn-out of the oxide by charge injection/ejection during a P/E cycling. In case of SLC, a cell having a bit value of '0(Programmed)' is worn out by W_{slc} in a P/E cycling. And a cell having a bit value of '1(Erased)' is not worn-out because there is no charge tunneling the oxide during the cycling. We simply assume that the degree of wear-out of a cell not in programmed state is zero because, in the case, there's no charge to be ejected from FG to substrate through the oxide. Based on the above modeling principles for SLC, the degree of wear-out of each cell is simply represented to the counts of the case that it has a bit value of 0.

In order to even out the usage of each cell, the technique for distributing the erasures and re-writes evenly across the cells is necessary. To achieve it, we

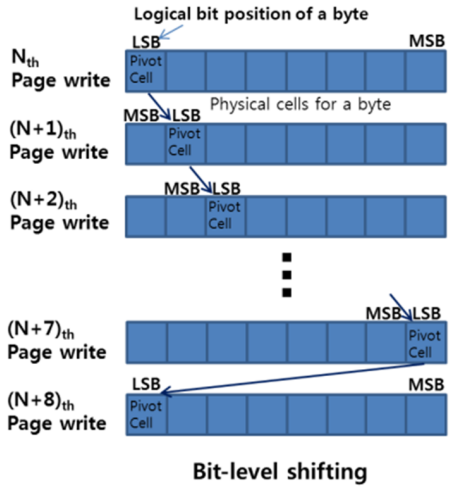
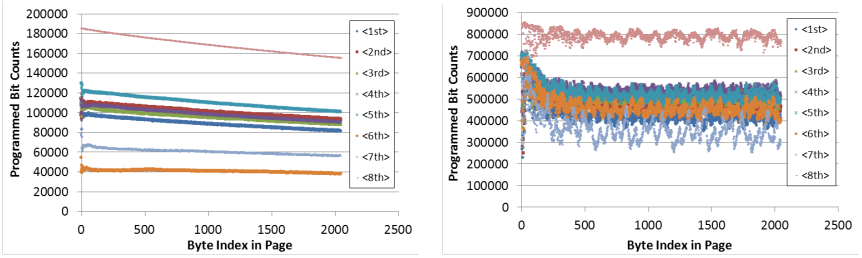


Fig. 2. Our proposed schemes for expanding lifetime of NAND flash cells



(a) (b)

Fig. 3. (a) Page-wide view of wear-out for Linux source code, (b) Page-wide view of wear-out for DB workload

propose a simple bit-level shifting scheme for SLC as shown in Figure 2. To make the frequncy of programmed state of each cell statistically even, every page-level program and erase cycling, a byte of data is written from the pivot bit in the byte and the rest of bits are written from the LSB bit position. The location of pivot bit is shifted by one bit. In practical implementation of this scheme, we can set the location of pivot bit as the value of block erase counts modulo 8.

4 Evaluation

We evaluated our bit-level wear leveling scheme based on the proposed simple wear-out model for SLC NAND flash. The evaluation is based on the previously obtained data patterns for different file data workloads, Linux source code files,

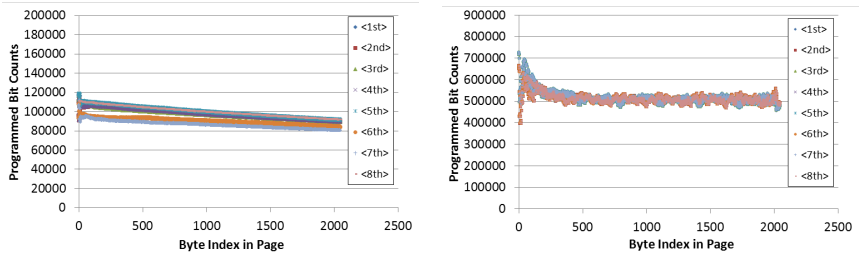


Fig. 4. (a) Results of our wear-leveling scheme for Linux source code, (b) Results of our wear-leveling scheme for DB workload

which are the most good reference for docuement file data composed of mostly ASCII code-based data, and DB update data, which is frequently update to the storage and have big influence on NAND flash's wear-out.

To simulate and visualize the page write operations onto NAND flash memory, we examine data pattern of the two workloads in page-level. We assume that the page size of NAND flash is 2KB. Figure 3(a) and Figure 3(b) show page-wide view of data pattern of each workload for SLC. As previously described, the probability for each bit to be programmed is remarkably different and it is neccessary to be wear-leveled. The reason why the counts of lower indexed bytes in a page are slightly higher than the others in case of Linux source codes, is beacause the size of many files in Linux source tree is lower than the page size and we obtained the data by only accumulating the count of bit value having logic 0. In case of smaller sized file than the size of one page, the dummy data contents in higher offset than the size of files, had better being filled with logic '1' for NAND flash's endurance.

The results in Figure 4(a) and Figure 4(b) are obtained by applying our wear-leveling scheme, bit-level shifting, on the workloads in case of SLC. By applying the proposed scheme, the estimiated wear-out counts of each cell become nearly uniform. With comparing the maximum count, in case Linux source code, the degree of wear-out becomes 36% lower and 15% lower in case of DB workload. Additionally, in case of DB workloads, lower offset of bytes in a page tends to be worn out more than the others in spite of bit-level wear leveling.

5 Conclusion

We introduce our observation that bit-level data patterns of widely-used file data are not uniform and, based on the breakdown mechanism of oxide, we also propose simple wear-out models for SLC NAND flash. To expand endurance of NAND flash, we emphasize the consideration of bit-level wear-leveling scheme and proposed simple approaches. We expect that our solutions can be adopted in the controller-level togther with other wear-leveling scheme or other encoding schemes.

References

1. DBT2 benchmark, http://sourceforge.net/apps/mediawiki/osdldbt/index.php.
2. Bin, Z. W., Ming, X., Zheng, X., and Kang, Z. A. Charge-to-breakdown (Q_{BD}): a method to monitor the ultrathin tunnel oxide in E^2PROM. In *Solid-State and Integrated Circuit Technology, 1998. Proceedings. 1998 5th International Conference on* (1998), pp. 287 –290.
3. Chang, L.-P., and Huang, L.-C. A low-cost wear-leveling algorithm for block-mapping solid-state disks. *SIGPLAN Not. 46*, 5 (Apr. 2011), 31–40.
4. Chang, L.-P., and Kuo, T.-W. Efficient management for large-scale flash-memory storage systems with resource conservation. *TOS 1*, 4 (2005), 381–418.
5. Chang, Y.-H., Hsieh, J.-W., and Kuo, T.-W. Improving flash wear-leveling by proactively moving static data. *IEEE Trans. Computers 59*, 1 (2010), 53–65.
6. Hussain, S., and Mansoor, A. Flash modelling for wearleveling algorithms. In *High Capacity Optical Networks and Enabling Technologies (HONET), 2011* (dec. 2011), pp. 267 –272.
7. Murugan, M., and Du, D. H.-C. Rejuvenator: A static wear leveling algorithm for NAND flash memory with minimized overhead. In *MSST* (2011), A. Brinkmann and D. Pease, Eds., IEEE Computer Society, pp. 1–12.
8. SQLite. Sqlite official home page. http://sqlite.org.