An Efficient Copy-Back Operation Scheme Using Dedicated Flash Memory Controller in Solid-State Disks

Woo Tae Chang [1,2,a], Yong Tae Lim [2,b], and Jun Dong Cho [1,c]
[1] Sungkyunkwan University, Suwon, Korea
[2] Samsung Electronics Co. Ltd., Hwasung, Korea
Email: {[a] wtzzang, [c] jdcho}@skku.edu, {[a] wt.chang, [b] yongtae.lim}@samsung.com

***Abstract*—Today, flash memory is widely used as storage media in diverse computing environments. Several merits of flash memory such as small form-factor without any mechanical component, high performance and low power consumption replace magnetic storage media. However, flash memory requires emulator called flash translation layer (FTL) to use a common storage media such as hard disk drive (HDD). Merge operation of FTL including copy-back operation reduces the performance of SSD. To address this performance degradation in merge operation, this paper proposes an efficient copy-back operation using flash memory controller in hardware which reduces the merge time and improves FTL performance. Experimental results show that the proposed scheme improves 9% random write I/O performance (IOPS) of SSD compared to the existing technique.**

***Index Terms*—flash memory, flash translation layer, hard disk drive, Solid-state disk, merge operation, copy-back operation, flash memory controller, IOPS**

I. Introduction

Flash memory is becoming increasingly important in storage systems with rapid growth of portable and mobile applications. The price per unit capacity of flash memory is increasingly comparable to HDD.

Flash memory is a non-volatile memory (NVM), which can be electrically erased and programmed; it has many advantages as compared with conventional HDD.

The main advantages of Flash memory are as follows. Firstly, flash memory has small form-factor and light of weight suitable for small device. Secondly, it is resistant to shock, since it has no mechanical components such as hard disk platter, head and spindle motor. Thirdly, flash memory reduces the power consumption. Finally, it provides much high performance than HDD.

However, despites various merits, it has critical features. Flash memory must be erased before writing. And it is erased in blocks, but written in pages. Flash memory utilizes flash translation layer (FTL) to solve its drawback. The FTL enables flash memory device to emulate a common storage media such as HDD. Solid-state disks (SSDs) are the storage device that uses flash memory as their storage media.

The FTL should contain at least four features to be optimized so as to increase the performance: logical-to-physical address mapping, merge Operation, wear-leveling, and power-off recovery.

In this paper we propose an automatic error correction so-called copy-back method using flash memory controller in merge operation. Using this method, we can reduce unnecessary DMA, increase DRAM Bandwidth, and improve the IOPS performance.

This paper is organized as follows. Section II introduces related works of flash memory, FTL, merge operation, and copy-back operation. Section III explains our new scheme of copy-back operation using hardware logic. Section IV provides performance evaluation of the proposed scheme and compares with the traditional scheme. Section V concludes the paper.

II. Background

A. Flash Memory Storage

Flash memory uses cell-based floating-gate transistors as shown in Fig. 1 [1].

- Program Operation: Off-cell, Tunnel injection and Data '0'
- Erase Operation: On-cell, Tunnel release and Data '1'
- Read Operation: Read cell whether it is 0 or 1

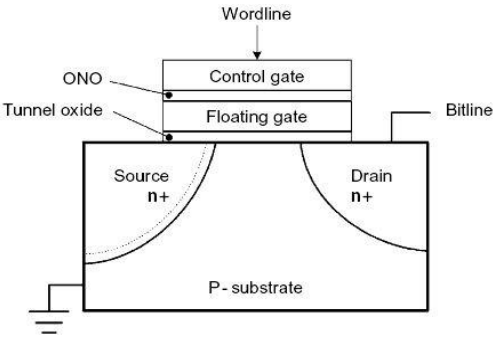

Figure 1. Flash memory cell.

Manuscript received July 31, 2013; revised September 15, 2013.

doi: 10.12720/ijoee.2.1.13-17

Program and Read operations are performed on a page basis, whereas Erase operation is conducted in the block.

NAND flash consists of cells, pages, blocks, planes, die and package as shown in Fig. 2 [2]. However, each NAND device has different size. NAND flash includes extra storage on each page (8KB), found in a spare area of 640B for parity of error correcting code (ECC), because NAND flash requires the ECC to ensure the data integrity.

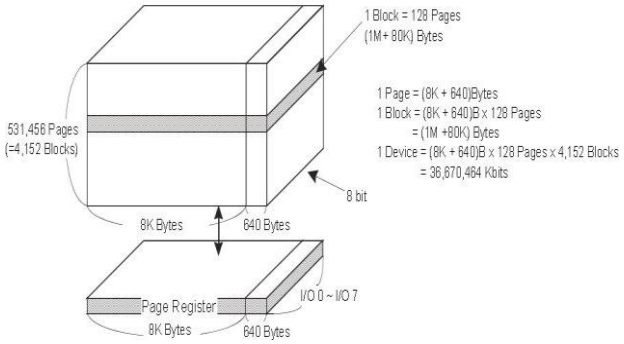

Figure 2. Array organization of NAND device.

B. Flash Translation Layer (FTL)

The objective of the FTL is to provide transparent services for file systems to access flash memory as a block device, to optimize the performance using the least amount of buffer memory, and to extend the endurance by using the wear-leveling technique. It is shown in Fig. 3 [3].

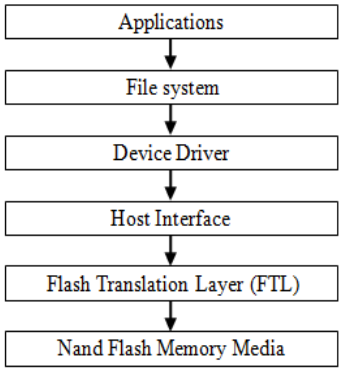

Figure 3. Block diagram of a typical flash memory storage system.

The FTL should contain at least four features as follows [4]:

- Logical-to-physical address mapping
- Merge Operation (Garbage collection, Migration, Compaction)
- Wear-leveling
- Power-off recovery

C. Merge Operation

In general, while performing a write operation, the FTL allocates log block to cache block. But, The log block will be very small amount compared to the data block. As number of write operation increased there can be no available free blocks remained in the log blocks. Merge operation is to make free blocks to resolve this problem. There are three types of merge operations, which are shown in Fig. 4 [5] and [6].

- Switch merge: In Fig. 4(a), log block B becomes a new data block while the old data block A is erased.
- Partial merge: When only part of a block is invalidated by overwriting and the overwritten pages are located in the same block sequentially as shown in Fig. 4(b)
- Full merge: Full merge operation involves the largest overhead. As shown in Fig. 4(c), log block B is selected as the victim block. The valid pages from log block B and its corresponding data block A will be copied into a newly erased block C, and then block A and B will be erased.

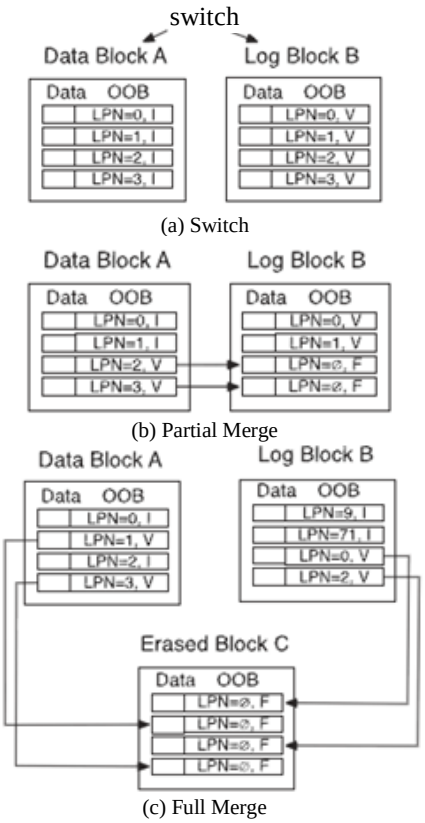

Figure 4. Three kinds of merge operations in FTL.

D. Flash Memory Copy-Back Operation

Copy-back operation is copying data from one page in the flash memory to another row address without data output and data input within the page buffer. As shown in Fig. 5, the operation can be performed through following steps:

- Read for Copy-back : Read (Sensing and dump) from source page to page buffer
- Data-out for error checking or data modifying from page buffer
- Data-in corrected or modified data to page buffer
- Copy-back program: Program (Dump and program) from page buffer to destination page

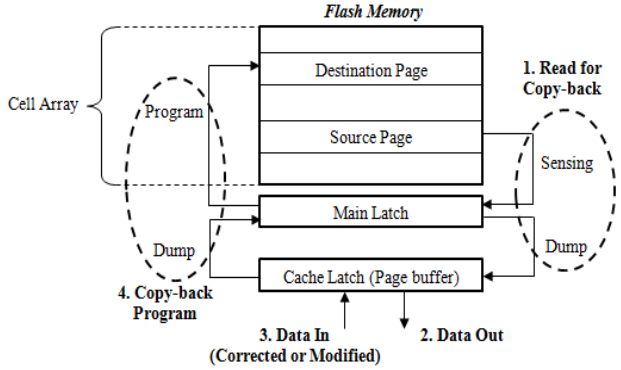

Figure 5. A copy-back operation in NAND flash memory.

E. Traditional Merge Operation

Merge operation in Flash based SSD is copying data from log block or data block in the flash memory to another log block or free block. If there is no error in a page buffer, merge operation is conducted by steps 1 and 4 of Fig. 5. But, because of the occurrence of multiple errors in current flash memory, flash memory requires an ECC in order to ensure the integrity of data during a copy-back operation, and needs steps 2 and 3 of Fig. 5.

Flash memory can not have an ECC engine due to the fact that implementation and size are critical issues. Hence, existing method of copy-back operation is similar to just Read (steps (1) ~ (4) of Fig. 6) and Program (steps (5) ~ (8) of Fig. 6) from the page buffer in flash memory to buffer memory such as DRAM in a SSD controller.

Hereafter, we will call this operation is external copy-back operation which consists of fully steps (1) ~ (8) as shown in Fig. 6.

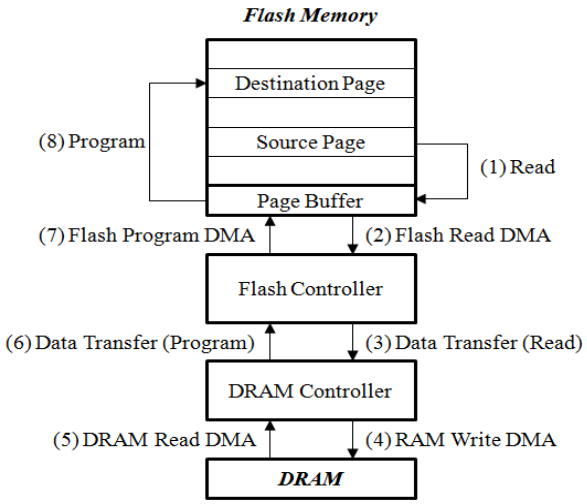

Figure 6. Traditional merge operation (external copy-back).

And if the data needs to be modified or merged between channels, it also should be used the external copy-back operation. In general FTL, by exploiting buffer memory we can conduct merge operation efficiently and manage mapping table. While always using external copy-back operation, additional data transfer, DMA time and hardware/software overhead between DRAM and Flash controller reduce the system performance (IOPS) [7].

III. An Efficient Copy-Back Operation (ECB)

Compared to traditional scheme, the efficient copy-back (hereafter ECB) needs not to transfer the data to DRAM for correcting and detecting error. The ECB is hardware logic to correct and detect error of page buffer in the Flash controller as shown in Fig. 7.

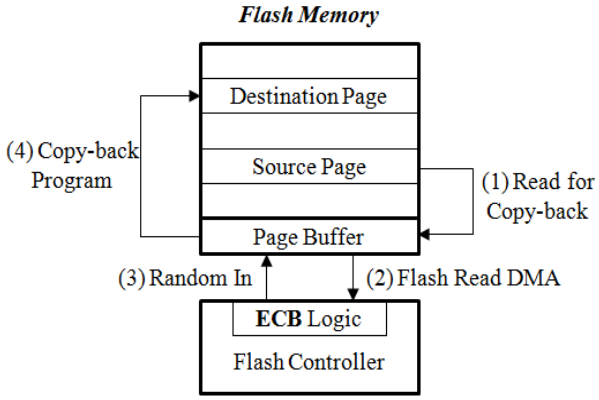

Figure 7. An efficient copy-back(ECB) operation.

In order to use the ECB, Flash controller conducts Read for Copy-back (step 1 in Fig. 7) command firstly. Then flash data is stored in the page buffer, controller reads data (step 2) from the page buffer and corrects the errors (step 3) using ECB logic automatically. In detail of step 3, ECB logic finds location(Byte information) and position(Bit information) of errors occurred, then flash controller conducts random byte program by bit flipping the data that found by ECB logic. In general, BCH (Bose-Chaudhuri-Hocquenghem) code and LDPC (Low-Density Parity-Check) code are used as the ECC algorithm in the SSD, whereas HDD uses the RS-code [8].

After checking all data in the page buffer, executes Copy-back program operation through step 4 as shown in Fig. 7.

The difference between the two schemes is on step 3 of the ECB and steps 3~7 of traditional merge operation. Step(3) Random In operation time of the ECB increases with the number of errors in a page, on the other hand steps 3~7 of the traditional merge operation is not affected by the number of errors in a page. In other words, before the flash memory is worn out as PE (Program-erase) cycle increases, the performance of ECB is better than the traditional merge operation.

IV. Performance Evaluation

The proposed scheme improves the performance of merge operation. We consider several parameters in the evaluation of Random Write performance.

Fig. 8 illustrates the difference between the ECB and the traditional merge operation under the same conditions. The performance of two methods increases linearly number of channels. However the ECB is constantly performed 9% more than the traditional merge operation.

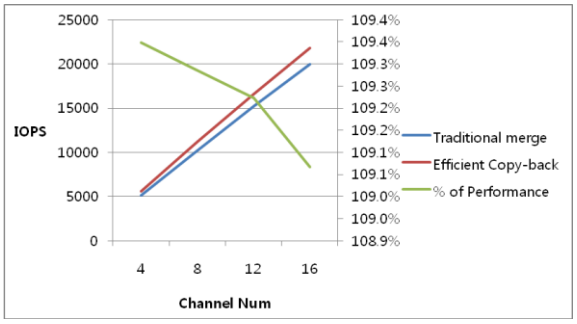

Figure 8. Random write performance by number of channles.

When increasing the number of Banks (Way), the performance of ECB is also better than the traditional merge operation as shown in Fig. 9, it shows a little bit irregular dependency on the number of banks than the change according to the increase in number of the channel.

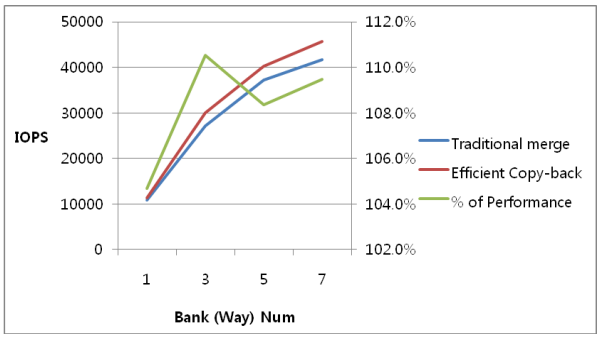

Figure 9. Random write performance by number of banks.

Fig. 10 compares the performance difference by number of invalid page in block, the ECB is also better than the traditional merge operation. However, according to increase in number of invalid page, performance difference is reduced continuously.

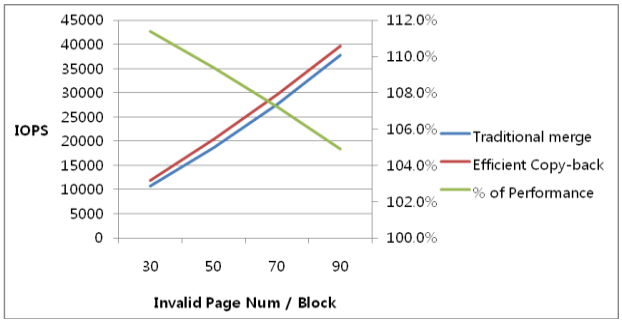

Figure 10. Random write performance by invalid page number / block.

The foregoing experiment results in Fig. 8, Fig. 9 and Fig. 10, are measured at the fixed number of errors less than 100 in a page.

Lastly, as mentioned in Section III, the performance is different by increasing number of errors in a page, in other words the performance of ECB will decrease as compare to traditional merge operation when PE-cycle increase more than the threshold number of errors.

Fig. 11 illustrates the results related to the number of errors. In this experiment, the performance of ECB gets dropped than the traditional merge operation, when error occurred is more than 600 errors. But, if error occurs over the ability of ECC, it becomes the uncorrectable state. When the system gets into EOL (End of Life) state due to occurrence many errors, the error recovery mode is an important factor. The drawback of ECB does not make sense, because this system goes to the uncorrectable state if the number of errors is more than 500 per page.

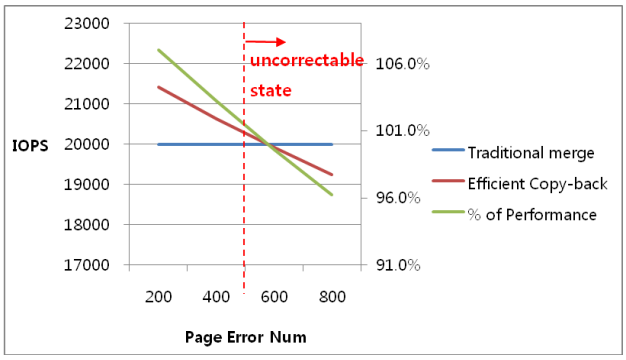

Figure 11. Random write performance by number of page error.

V. Conclusion

In this paper, we proposed an efficient hardware copy-back operation which reduces the merge time and improves FTL performance.

Overall, 9% of random write performance is improved before wearing out of the flash memory as increasing PE-cycle. According to increase in the number of errors in a page, the effect of ECB is decreased. However, this effect is insignificant, because the system goes to uncorrectable state when the number of errors increases over a threshold value. In some cases, the ECB can be applied based on the number of errors on a page.

Henceforth, we will conduct further research about the ECB by the difference of page mapping method of FTL, and try to improve this proposed scheme.

References

[1] K. Paulsen, *Moving Media Storage Technologies*, Elsevier, 2011, ch. 7, pp. 165-185.
[2] Samsung Electronics. 32Gb A-die NAND Flash / Multi-Level-Cell (K9GBG08U0A). Available: http://www.samsung.com
[3] Y. H. Deng and J. P. Zhou, "Architectures and optimization methods of flash memory based storage systems," *Journal of Systems Architecture*, vol. 57, no. 2, pp. 214–227, 2011.
[4] Intel Corporation. Understanding the flash translation layer (FTL) specification. Intel Application Note (AP-684). Available: http://www.intel.com
[5] J. Kim, J. M. Kim, S. H. Noh, S. L. Min, and Y. Cho, "A space-efficient flash translation layer for compact flash systems," *IEEE Transactions on Consumer Electronics*, vol. 48, no. 2, pp. 366-375, 2002.
[6] S. Park, Y. Kim, B. Urgaonkar, J. Lee, and E. Seo, "A comprehensive study of energy efficiency and performance of flash-based SSD," *Journal of Systems Architecture*, vol. 57, no. 4, pp. 354–365, 2011.
[7] IOPS. Performance characteristics. Available: http://en.wikipedia.org/wiki/IOPS
[8] L. Zhang, U A. Tan, and Q. K. Zhang, "Identification of NAND flash ECC algorithms in mobile devices," *Journal of Systems Architecture*, vol. 9, no. 1, pp. 34–48, 2012.

Woo Tae Chang was born in 26th of February 1977 at Seoul, Korea. He is pursuing his Master degree from Sungkyunkwan University (SKKU). He is in charge of the SSD controller. He has been with Samsung Electronics co. Ltd., Memory division, Hwasung, Korea, since 2002.

Yong Tae Lim was born in 1st of January 1970 at Busan, Korea. He received the Master degree in electronics engineering from Yonsei University (YSU) and he is in charge of the SSD controller. He has been with Samsung Electronics co. Ltd., Memory division, Hwasung, Korea, since 2004. From 1997 to 2003, he was with Hynix, Icheon, Korea. His research interests include management of NAND flash memory, soc hardware design and verification.

Jun Dong Cho was born in 21th of July 1957 at Seoul, Korea. He received Ph.D. degree from Northwestern University, Evanston, IL, 1993, in computer science. He was a Senior CAD Engineer at Samsung Electronics, Co., Ltd. He is now Professor of Dept. of Electronic Engineering, Sungkyunkwan University, Korea. He received the Best paper award at the 1993 Design Automation Conference. He has been an IEEE Senior Member since April 1996. His research interests are in the area of System on Chip Design Optimization and lower power designs.