Efficient Page Collection Scheme for QLC NAND Flash Memory using Cache

Seok-Bin Seo[1], Wanil Kim[2], Se Jin Kwon[3]
Department of Computer Engineering
Kangwon National University
Samcheok, South Korea

***Abstract*—Recently, semiconductor companies such as Samsung, Hynix, and Micron, have focused on quad-level cell (QLC) NAND flash memory chips, because of the increase in the capacity of storage systems. The QLC NAND flash memory chip stores 4 bits per cell. A page in the QLC NAND flash memory consists of 16 sectors, which is two to four times larger than that of conventional triple-level cell flash NAND flash memory. Because of its large page size, when the QLC NAND flash memory is applied to the current storage system directly, each page space is not efficiently used, resulting in low space utilization in overall storage systems. To solve this problem, an efficient page collection scheme using cache for QLC NAND flash memory (PCS) is proposed. The main role of PCS is managing the data transmitted from the file system efficiently (according to the data pattern and size), and reducing the number of unnecessary write operations. The efficiency of PCS was evaluated using SNIA IOTTA NEXUS5 trace-driven simulation on QLC NAND flash memory. According to close observation, PCS significantly reduces 50% of write operations compared with previous page collection algorithms, by efficiently collecting the small data into a page. Furthermore, a cache idle-time determination algorithm is proposed to further increase the space utilization of each page, thereby reducing the overall number of write operations on the QLC flash memory.**

Keywords—Solid state drive; storage systems; cache; flash translation layer

I. Introduction

In recent years, the solid-state drive (SSD) market has grown rapidly owing to the capacity increase of storage systems. Accordingly, semiconductor companies such as Samsung, Hynix, Micron, and Toshiba, are carrying out research on low-cost, high-capacity quad-level cell (QLC) NAND flash memory chips. Single-level cell (SLC) NAND memory stores a single bit in a single cell, and multilevel cell (MLC) NAND flash memory stores two bits in a single cell. However, QLC NAND flash memory stores four bits in a single cell; thus, the capacity can be increased more compared with SLC/MLC/tri-level cell (TLC) NAND flash memories NAND flash memories commonly consist of a "page" (a set of sectors) in read/write operation units, and several pages are gathered and compose a "block," which is a unit of erase operation. Furthermore, they have an "erase-before-write" property, i.e., an erase operation must be performed on a block prior to performing data renewal of a page [1].

The biggest differences between QLC NAND flash memory and SLC/MLC/TLC NAND flash memories are the page/block size and performance. In a QLC NAND flash memory, a page consists of 16 sectors, which is larger than that of conventional SLC/MLC/TLC NAND flash memory. Consequently, the read/write/erase operations of QLC NAND flash memory are slow compared with those of SLC/MLC/TLC NAND flash memory. Therefore, a page of QLC NAND flash memory is larger than that of SLC/MLC/TLC NAND flash memory, and its overall performance is reduced. Consequently, when the read/write pattern traces of the file system used in conventional operation systems are applied to a QLC NAND flash memory, the space utilization is decreased, and the number of write operations is increased, and this can be confirmed through a performance evaluation.

In this research, the problem of unnecessary write operations occurring in a QLC NAND flash memory is solved by implementing a page collection method in a flash translation layer [2]. When the file system issues a read/write requests along with its logical addresses, Page Collection Scheme stores the data in a temporary page storage ("cache" or "register") according to its page collection algorithm. Then, the collected data in the cache is transferred to the address mapping layer to be written onto the flash memory. The main issue of this paper is to solve the inefficiency caused by the large page size of QLC NAND flash memories. Therefore, the scope is limited to Page Collection Scheme. However, please notice that Page Collection Scheme can co-exist with any address mapping algorithms, since its main role is to increase interchangeability between the address mapping layer and QLC NAND flash memory.

The proposed Page Collection Scheme was tested through SNIA IOTTA's Nexus 5 Smartphone Traces-based simulation, and the results showed a large increase in the number of pages that had good space utilization and a decrease in the number of write operations, as discussed in Section 3.The rest of this work is structured as follows. Section 2 shows the characteristics of flash memories, the result of using data patterns for the read/write requests of a conventional host in the NAND flash memories that emerged before QLC NAND flash memory, and the problem occurring when the same patterns are applied to QLC NAND flash memory. In Section 3, the proposed Page Collection Scheme algorithm is described. In Section 4, the comparative

This work was supported by Basic Science Research through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A3B04031440). This study was also supported by 2017 Research Grant from Kangwon National University (No. 000000000).

performance evaluation performed through the SNIA IOTTA Nexus5 Smartphone Traces simulation based on the proposed Page Collection Algorithm is described. In Section 4, a future study is outlined, and a conclusion is provided.

TABLE I. NAND FLASH MEMORY WITHOUT PAGE COLLECTION SCHEME ALGORITHMS

Device Request	*SLC NAND Flash Memory*	*MLC NAND Flash Memory*	*QLC NAND Flash Memory*
Number of pages with space utilization greater than 50%	234,543 Pages	234,544 Pages	52,620 Pages
Number of pages with space utilization of 50% or less	38,024 Pages	38,052 Pages	219,448 Pages

II. PROBLEM DEFINITION

SLC NAND flash memory has the characteristics that an operation of 1 bit per cell is performed, and the 2 KB data area exists per page. In an MLC NAND flash memory, an operation of 2 bits per cell is performed, and a 4- KB data area exists per page [3] [4]. In a QLC NAND flash memory, an operation of 4 bits per cell is performed, and there is an 8 KB data area per page.

Experiments were performed through SNIA IOTTA Nexus5 Smartphone Traces simulation without applying the Page Collection Scheme to the SLC and MLC NAND flash memories. In the results, the percentage of pages that had 50% or lower space utilization in a page was 14% (38,024 and 38,025, respectively), as shown in Table 1, and there was no occurrence of problems caused by space utilization in each page. In the results of performing the simulation for QLC NAND flash memory by using the same method, each page showed a low space utilization, as shown in Table 1. Among 272,568 pages, the number of pages having 50% or lower space utilization was 219,448 (80%).

Algorithm 1. Page Collection Scheme Algorithms

```
INPUT: write(LPN, data, SIZE);
1.WAIT: IF ‘write request’ IS EQUAL 0 GOTO WAIT
IF ‘write date size’ IS BIGGER THAN 6KB THEN
IF ‘LPN’ IS EQUAL ‘Register’ THEN
‘data write in flash memory’ AND
‘clear register’
5.        ELSE ‘data write in flash memory’
6.   ELSE ‘data write in Register
7. GOTO WAIT
```

The detailed comparative analysis in Table 1 revealed the reason why a page could not be filled with data by exceeding 50% space utilization in the QLC NAND flash memory, compared with the SLC and MLC NAND flash memories. Even when the data was applicable for an update, or when a write command requested from the file system was of 16 or more sectors, the starting or ending part with respect to the data size of each sector command did not fully fill a page. To increase the low space utilization of these pages that are operated inefficiently, a page collection method using a cache is proposed.

III. PAGE COLLECTION SCHEME

The page collection method using a cache proposed in this paper is assumed to use the double cache structure described in Fig. 1. Moreover, a random in/out method is used to store and modify data in the resister of a double-cache structure [6]. Furthermore, because the QLC NAND flash memory is based on the double-cache structure, it is assumed that the size of the cache or resister is the size of one page of QLC NAND flash memory.

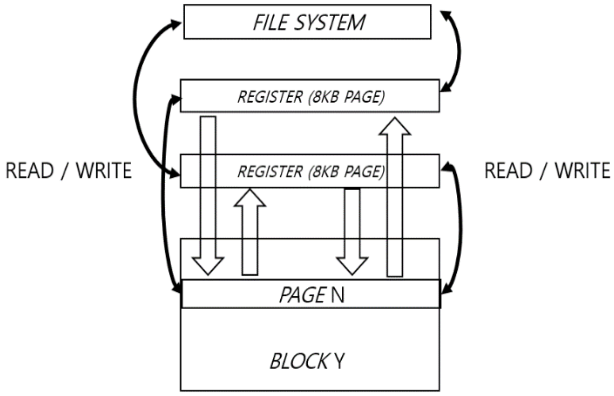

Fig. 1. Double Cache Structure [5].

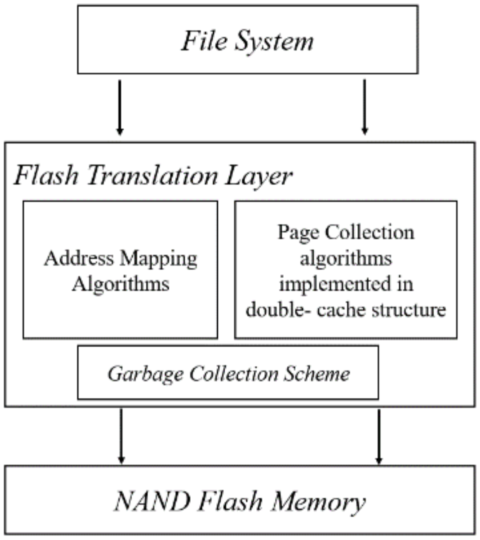
File System
Flash Translation Layer
Address Mapping Algorithms
Page Collection algorithms implemented in double- cache structure
Garbage Collection Scheme
NAND Flash Memory

Fig. 2. FTL Structure with Double-Cache Structure.

In addition, it is assumed that the Page Collection Scheme method and the FTL structure are included in the Page Collection Scheme algorithm having a double-cache structure, as shown in Fig. 2. Based on the above assumption, Algorithm 1 is proposed as follows: when a write request of the OS file system is mage in FTL, the size is determined, and a data write request smaller than a certain size always resides in the cache, ensuring the size of a page will be as close to 8 KB (i.e., 16 sectors per page based on the 512-byte size) as possible.

First, when a write command of the file system is called, the data amount of one page that can be accommodated by the resister is assumed to be less than 6 KB (Lines 1 and 2 of the algorithm). If data of 6 KB or larger come in, they are checked to determine whether a corresponding resister exists, and, if there is a corresponding resister, the write operation of data to the flash memory is performed, and the corresponding resister is emptied for a new write.

IV. Performance Evaluation

The proposed Page Collection Scheme method was implemented based on Algorithm 1 proposed above, and then evaluated. As an outcome of using all the 31 traces provided by IOTTA's Nexus5 Smartphone Traces, Table 2 shows the results of the experiment for the pages that had an increased space utilization compared with the conventional method. In the results of the experiment, when the Page Collection Scheme was not applied, there were 219,948 pages that had low space utilization, as shown in Table 1, but when the proposed Page Collection Scheme method was applied, the number of pages showing low utilization was 110,936, displaying the decrease effect from approximately 80% to 50%.

However, despite the existence of the cache, many pages still had a problem regarding the space utilization. A register exists in a wait state continuously when the resister is not applicable for an update. Accordingly, an algorithm that decides the idle time of the cache is proposed to solve the above problem.

TABLE II. QLC NAND flash memory with Page Collection Scheme Algorithms

Algorithms *Request*	*QLC NAND Flash Memory Without PCS Algorithms*	*QLC NAND Flash Memory With PCS Algorithms*
Number of Pages over 6 KB	52,620 Pages	126,836 Pages
Number of pages less than 6 KB	219,448 Pages	110,936 Pages

V. Future Research Activities

To overcome the limitations of the proposed page collection method using cache, the idle time of the OS of the system is decided. Further, when it is in the category of idle time, the data in the cache are moved to the NAND flash memory, and the corresponding cache is emptied. This way, it is expected that the number of pages having more active and higher space utilization will be increased compared with the conventional methods.

The page collection method that decides the idle time of the cache is an algorithm that decides the idle time of the device or decides the idle time by counting the number of times that the cache was not updated after the write command of the file system in the OS. The algorithms constructed are shown in Algorithm 2 and Algorithm 2-1. In Algorithm 2, a variable was added to Algorithm 1 to check the update status of the cache (Line 5 of Algorithm 2). After the conditions of Algorithm 2 are satisfied, Algorithm 2-1 is performed. If the number of "not-updating the cache" is larger than a certain value or the device is in an idle-time state, the data existing in the cache are moved to the flash memory, and the corresponding memory is emptied for a new write (Lines 1-4 of Algorithm 2-1).

The cache idle-time decision algorithm is different from the conventional algorithm, in which a resister is continuously remaining in the wait state to obtain an expected result even when the page will no longer be updated. The algorithm that decides the idle time of the cache leads to a more efficient use of QLC NAND flash memory overall by moving data to the NAND flash memory according to the criteria, instead of waiting for an update of the resister indefinitely. To prove that the above results of the experiment are significant results, an experiment will be conducted on the current embedded board in future.

Algorithm 2. Cache idle-time decision algorithms

```
1. WAIT: IF 'write request' IS EQUAL 0 GOTO WAIT
2.    IF 'write data size' IS BIGGER THAN 6 KB THEN
3.       IF 'LPN' IS EQUAL 'register' THEN
            'data write in flash memory' AND
            'check register()' AND
            'clear register'
7.       ELSE 'data write in flash memory'
8.    ELSE 'data write in register'
9. GOTO WAIT
```

Algorithm 2-1. Cache idle time decision algorithms

```
<CHECK REGISTER()>
IF 'register non-update' IS MORE THAN $\mathcal{I}$ OR 'device' IS EQUAL 'idle-time' THEN
'data write in flash memory AND
'clear register' AND RETURN
```

VI. Conclusion

In this work, a page collection method that uses cache was investigated for QLC NAND flash memory. In the SLC and

MLC NAND flash memories, the problem of space utilization in each page was not found with respect to the write requests of the conventional file system. However, in the QLC NAND flash memory, problems occurred in many pages because of low space utilization. The proposed page collection method using the cache ensures appropriate space utilization by using a resister, and it performs the write request to the NAND flash memory instead of sending the data directly to the NAND flash memory according to the write request of the file system. In the results of trace-based simulation, the QLC NAND flash memory that applied the proposed method exhibited a decrease in the number of total write operations and an increase in the number of pages having high space utilization. Furthermore, in the future, a study is planned on the page collection method considering the cache idle-time decision, and more pages of QLC NAND flash memory are expected to have high space utilization.

References

[1] M.A. Dongzhe FENG, Jianhua; LI, Guoliang, “A survey of address translation technologies for flash memories,” *ACM Computing Surveys* (CSUR), 2014.

[2] E. Gal and S. Toledo, “Algorithms and Data Structures for Flash Memories,” *ACM Computing Surveys*, 37(2): 138-163, 2005.

[3] MICRON Electronics, “NAND Flash Memory MT29F4G08AAAWP, MT29F8G08BAAWP, MT29F16G08FAAWP,”MICRON Electronics Data sheet, 2018.

[4] TOSHIBA Memory, “Toshiba Mos Digital Integrated Circuit Silicon Gate CMOS, TH58NVG4S0HTAK0,”TOSHIBA Memory Data Sheet, 2017.

[5] MICRON Electronics, “NAND Flash Status Register Response in Cache Programming Operations,” MICRON Electronics Technical Notes, 2018.

[6] Se Jin Kwon, Arun Ranjitkar, Young-Bae Ko, Tae-Sun Chung, “FTL algorithms for NAND-type flash memories,” Design Autom. for Emb. Sys. 15(3-4), pp. 191-224, 2011.