A TCB algorithms for wear leveling method in FTL-based NAND flash memory

Myungsub Lee

Abstract: Flash memory has the advantages of fast access speed, low power consumption, and low price. Therefore, it is widely used in many sectors of the electronics industry. However, the flash memory also has a disadvantage of a limited number of program/erase (P/E) cycles. Many wear-leveling techniques have been studied to prolong the lifetime of flash memory by equalizing the P/E cycles of the blocks. In this paper, we propose a novel wear-leveling technique called "tracking cold block" (TCB), which enhances a bit-set threshold. To enhance the accuracy of cold block information, we used a cold erase table (CET), which is calculated using the number of invalid pages of the blocks in a one-to-many mode, where one bit is related to many blocks. The experimental performance results illustrate that TCB prolongs the lifetime of flash memory by up to 70% compared with previous wear-leveling techniques.

Index Terms: NAND flash memory, wear leveling, non-volatile memory, lifetime, reliability

I. INTRODUCTION

NAND flash memory can be accessed by page or block units grouped in a series, providing a relatively high level of directness. As a result, NAND flash memory is not only used in portable de-vices, such as USB drives, SD cards, and smart phones, but also has a wide variety of uses such as in solid state drive (SSD) backup storage devices. Despite this advantage, NAND flash memory has a shorter lifetime than the existing backup storage devices because of limited overwriting in the same location. To resolve this problem, many researchers have presented techniques that consider NAND flash memory's special characteristics, for example, the technique for extending NAND memory's lifetime is called wear leveling [1-6].

In wear leveling, data separating blocks into "cold" and "hot" ac-cording to the number of erasures is saved in a table along with various other information, such as page reference counts, reference, and erasure time; this is used to extend the flash memory's lifetime. However, if this data is saved in memory, this introduces the disad-vantage of the increase in memory usage, and the implementation of embedded systems with low specifications becomes difficult. Therefore, researchers are exploring wear-leveling techniques that use a block erase table (BET), which is arranged by bits and can reduce memory usage [7]. As a BET's k value increases, multiple blocks are bundled into a single group to correspond to each bit. Therefore, as the k value increases, memory usage could be further reduced. However, as the k value increases, it seriously limits the performance of wear leveling.

To overcome this disadvantage in BET techniques, the hidden cold block problem has been identified as a cause of reduced perfor-mance, and to resolve this, wear-leveling techniques, which use bit arrangements and bit set thresholds (BST), have been proposed [8]. To resolve the hidden cold block problem, BST determines if the bit mapped to the block group, in which the erasure operation occurs, has been set to 1. Moreover, if the block group's average number of invalid pages is greater than the overall block group's average num-ber of invalid pages, the BET bit is set to 2, thus resolving the hid-den cold block problem due to an increase in the k value. However, BST is only effective if k ≥ 1, which limits the performance increase because of BET.

To resolve these problems, this paper proposes the tracking cold blocks (TCB) technique, which strengthens the extent to which cold blocks are recognized and minimizes the migration of pages due to wear leveling to reduce the overhead and improve the overall memory lifetime.

II. PROPOSED METHOD

We present the TCB, which is a technique to overcome the disadvantages of BET and BST. Before explaining TCB, note that we have changed the name of the BET, which manages the erasure records for block groups, to erase table (ET) to prevent the name from conflicting with the BET used as one of the comparison techniques. In TCB, alongside the existing ET, an additional table is used that memorizes moved blocks judged to be cold blocks. This is called the migrated cold block table (MCT). For wear leveling, TCB targets the blocks in which all the ET and MCT bits are zero to prevent indiscriminate page movement and excludes unsuitable blocks from garbage collection [9, 10, 12] to improve NAND flash memory performance. Fig. 1 shows the structure of the tables used in TCB.

In Fig. 1, the block group, which is a collection of 2k blocks, maps to both ET and MCT bits; the two tables use a unified index. Algorithm 1 describes the detailed process by which TCB uses ET and MCT, and Fig. 2 shows the sequential process of TCB wear leveling when the k value is 0 and when it is 1.

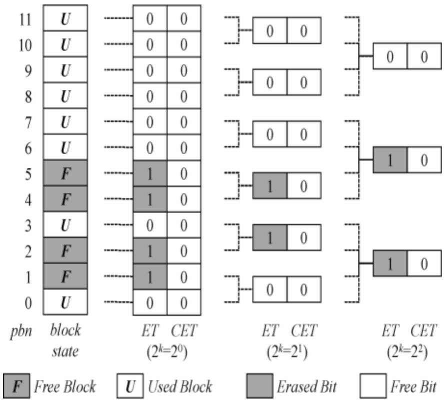

Fig 1 TCB architecture

Revised Manuscript Received on December 22, 2018.

Myungsub Lee1*, Department of computer information, yeungnam university college, South Korea (email: skydream@ync.ac.kr)

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

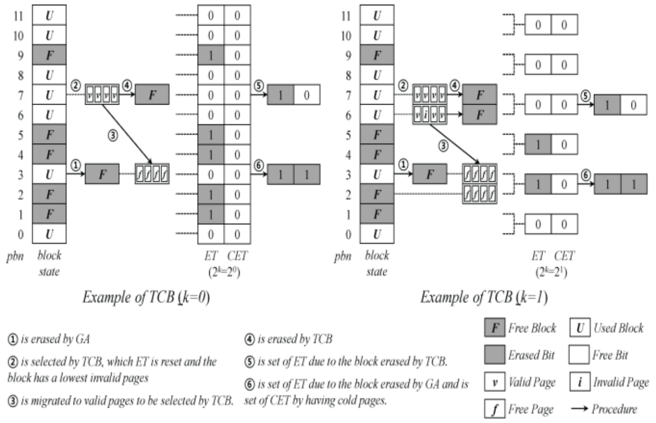
pbn
block state
ET CET
($2^k=2^0$)
Example of TCB (k=0)
ET CET
($2^k=2^1$)
Example of TCB (k=1)
① is erased by GA
② is selected by TCB, which ET is reset and the block has a lowest invalid pages
③ is migrated to valid pages to be selected by TCB.
④ is erased by TCB
⑤ is set of ET due to the block erased by TCB.
⑥ is set of ET due to the block erased by GA and is set of CET by having cold pages.
F Free Block
U Used Block
Erased Bit
Free Bit
v Valid Page
i Invalid Page
f Free Page
→ Procedure

Fig 2 TCB Procedures

According to algorithm 1, after ga_{victim} garbage collection is performed (Fig. 2 ①), if the total block erasure count e_{cnt} divided by the erased block group count f_{cnt} is greater than the threshold value T, wear leveling is performed. If all the block groups mapped to ET or MCT are erased, e_{cnt}, f_{cnt}, ET, and MCT are all reset to 0 (Algorithm 1; Steps 3–13).

Algorithm 1: TCB-Static Wear Leveling Procedure

```
   input: e_cnt, f_cnt, tcb_cnt, k, f_index, ga_victim, tcb_vimtim, ET, MCT, and T
   output: null
1  ga_victim ← GetVictimBlockByGreedy();
2  if e_cnt/f_cnt ≥ T do
3      if f_cnt ≥ size(ET) then
4          e_cnt ← 0;
5          f_cnt ← 0;
6          ResetAllFalgsInET();
7          return;
8      end
9      if tcb_cnt ≥ size(MCT) then
10         tcb_cnt ← 0;
11         ResetAllFalgsInTCB();
12         return;
13     end
14     for f_index = 0 to size(ET) do
15         if ET[f_index] = 0 and MCT[f_index] = 0 then
               // select tcb_vimtim, by equation (1)
16         end
17     end
18     for ga_victim mod k to (ga_victim mod k) + k do
19         if IsErasedBlock(ga_victim) = false then
20             EraseBlock(ga_victim);
21         end
           //migrate pages in tcb_victim to ga_victim
22         MigrateTo(ga_victim, tcb_victim);
23         EraseBlock(tcb_victim);
24         MCT[idx(ga_vimtim)] ← 1;
25     end
26     tcb_cnt ← tcb_cnt + 1;
27 end
```

If the above scenario does not apply, that is, if wear leveling should be performed, the tcb_{victim} block group, which will be migrated, is selected according to Eq. (1) from among the block groups with ET and MCT bits, which are all zero for the same index (Algorithm 1; steps 14–17; Fig. 2. ②).

In Eq. 1, v_i is the group's i-th block's valid page ratio; the group with the lowest *cost* is selected to be tcb_{victim}.

$$\text{Cost} = \sum_{i=0}^{2^k} 1 - v_i \tag{1}$$

If tcb_{victim} is selected, its data is migrated to the block group in which garbage collection was most recently performed (Algorithm 1; steps 18–25; Fig. 2. ③). Then, tcb_{victim} is erased, and the mapped ET bit is set to 1 (Algorithm 1; step 20; Fig. 2. ④ and ⑤).

Simultaneously, to represent the migrated block group determined to comprise cold blocks, the MCT bit mapped to ga_{victim} is set to 1 (Fig. 2. ⑥), and the cold block group count tcb_{cnt} is incremented (Algorithm 1; step 26). Algorithm 2 describes the ET update process via erasure.

Algorithm 2: TCB-ETUpdate

```
input: e_cnt, f_cnt, k, b_index, and ET

output: e_cnt, f_cnt

/* ET are updated based on b_index and k in the ET mapping when the block is erased.
*/
e_cnt ← e_cnt + 1; // Increase the total erase count.
// Update the ET if needed
if ET[b_index/2^k] = 0 do
    ET[b_index/2^k] ← 1;
    f_cnt ← f_cnt + 1;
end
```

In Algorithm 2, the ET update occurs if erasure occurs because of garbage collection or wear leveling. If an erasure occurs on a block of a group, the ET bit mapped to the block group's index is calculated and the ET bit is set to 1. In addition, the overall block erasure count e_{cnt} and the erased block count f_{cnt} are incremented (Algorithm 2; steps 1–5).

TCB recognizes that GA is the main technique used, wherein blocks with a large number of erasures are selected as targets for garbage collection. TCB considers blocks in which garbage collection has been performed as hot blocks, and it considers blocks in which the ET and MCT bits are zero, that is, block groups in which erasure has not been performed and the valid page ratio is high to be cold blocks.

As a result, TCB detects hot blocks through garbage collection and uses ET and MCT to strengthen cold block detection. This prevents indiscriminant page migration, and excludes cold blocks from participating in hot blocks' frequent garbage collection to the greatest extent possible. Thus, the garbage collection efficiency is improved and the overhead associated with the page migration policy is reduced. We can expect that the improved wear leveling will increase the NAND flash memory's lifetime and improve its overall performance.

III. RESULTS AND DISCUSSIONS

A. Experimental environment Results

To evaluate TCB's performance, we performed simulated experiments by using SSD (Extension for DiskSim), which is based on DiskSim 4.0; SPC and FIU were used as the workload. SPC has online-transaction-processing properties, and FIU includes host to terminal service operations for a data center. Table 1 shows detailed information on the workload with these properties, and Table 2 shows the parameters used as the experiment settings.

Table 1 Types and Characteristics of workload

characteristics types		ratio of request (%)		avg. request (byte)
		read	write	
SPC	financial1	23.16	76.84	4754.3
	financial2	82.28	17.72	4394.0
FIU	home1	1.12	98.88	7246.9
	home2	12.40	87.60	13079.9
	home3	3.94	96.06	51446.2
	home4	0.03	99.97	16558.1
	online	35.55	64.45	17566.8
	webmail	26.03	73.97	17930.5
	web research	0.14	99.86	22323.3
	web users	29.24	70.76	16942.0

Table 2 Simulation parameters

Description	Value
Total capacity	2GB
Reserved free blocks	15%
Garbage Collection Trigger	# of free blocks < 5%
Flash chip elements	1
Planes per elements	1
Blocks per plane	4096
Pages per block	128
Page size	4KB
Page read latency	60us
Page program latency	800us
Block erase latency	1.5ms
Blocks per lifetime	10^3

B. Experimental result and analysis

TCB performance was analysed by measuring valid-page migration overhead and the time at which blocks first arrived at the lifetime limit, as well as the deviation between blocks over the course of 100 million workload write operations; this was compared with the performances of GA, DP, and BET. Fig. 3 shows the lifetime of NAND flash for each technique when performing the same tasks, assuming the lifetime of NAND flash by using GA on each workload is 1.

The experimental results in Fig. 3 show that TCB improved the lifetime than GA, DP, BET, and BST by an average of 111%, 114%, 69%, and 69%, respectively. These results occurred because TCB was able to detect hot blocks chosen through garbage collection and reduce the frequency by which they were selected as targets for wear leveling. However, the minimization of garbage collection in hot blocks uses valid page migration; this method necessarily has an overhead, and thus it is necessary to measure the overhead related to migration count and compare it to existing methods.

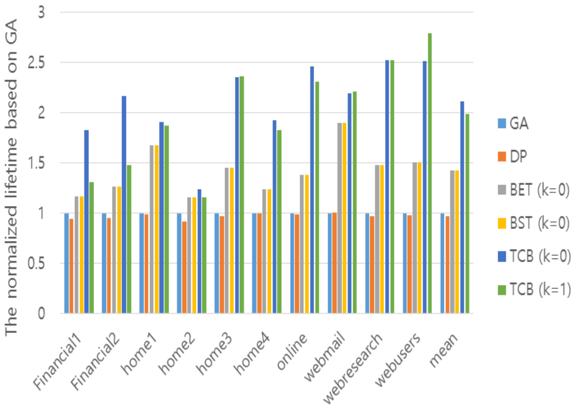

Fig 3 Lifetime of NAND flash memory

Fig 4 shows the average number of valid page migrations during 100 million workload performances for each technique.

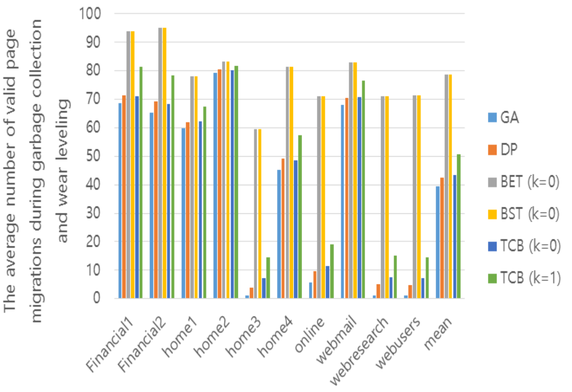

Fig. 4 Average of valid pages migrations during garbage collection and wear leveling

The experimental results in Fig. 4 show that TCB's number of valid page migrations was measured as an average of 2.4 times that of GA and DP. In addition, was measured as 35.3 times smaller than the average performances of BET and BST. These results show that there is almost no difference when comparing TCB's page migration policy overhead with that of GA, which does not perform wear leveling, and when compared to BET and BST, the TCB overhead was reduced by approximately 30%. Fig. 5 shows the wear leveling between blocks for each technique performing the same tasks in each workload if the wear leveling between NAND flash blocks performed by GA is set at 1

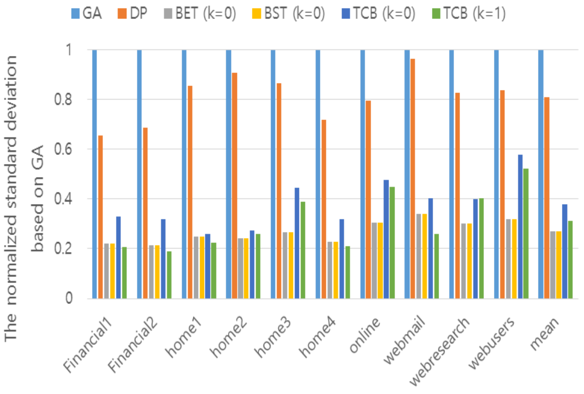

Fig 5 Normalized standard deviation

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Fig 5 shows that the wear deviation between blocks for TCB decreases on average by over 50% compared to those of GA and DP, while it increases by around 12% compared to those of BET and BST. However, when we consider the increase in NAND flash lifetime and the minimization of overhead from valid page migration, these numbers are sufficiently reduced. Moreover, in all experiments, problems with performance reduction were observed to occur when TCB's k value increased. Owing to BET's properties, BET-based techniques are all negatively affected by an increase in the k value. However, for TCB, even when the k = 1, it showed better performance than BET or BST when the k = 0.

IV. CONCLUSION

In this paper, we proposed TCB to overcome the problems in the existing techniques for NAND flash. TCB adds CET to the existing BET to reduce unnecessary page migrations during wear leveling by strengthening the cold block tracing detection conditions. In addition, by migrating cold blocks, the blocks performing garbage collection, that is, the hot blocks. TCB increased the usage rate of cold blocks and minimized garbage collection in hot blocks. The results of TCB performance evaluations show that the lifetime of NAND flash memory was increased from 69% to 114% than by using the existing techniques, and the overhead related to the page migration policy was reduced by approximately 70% compared to BET and BST. Wear deviation was approximately 12% higher compared with those of BET and BST; however, if we consider the increase in lifetime and decrease in overhead due to the migration policy, the experimental results confirmed that the overall performance of the NAND flash memory was improved.

V. ACKNOWLEDGMENT

This paper was supported by in Yeungnam University College Research Grants in 2018.

REFERENCES

1. Myungsub Lee, "Sampling-Based Block Erase Table Method in Wear Leveling Technique for Flash Memory," Journal of Engineering and Applied Sciences, 13(4), pp. 1023-1028, 2018.
2. Myungsub Lee, "A Wear Leveling Technique based on SBET for Flash Memory," Advanced and Applied Convergence Letters, AACL10, pp. 113-115, 2017.
3. Chang L. P, "On efficient wear leveling for large-scale flash-memory storage systems," Proceedings of the 2007 ACM symposium on Applied computing, pp. 1126–1130, 2007.
4. El-Shorbagy, M. A., and A. A. Mousa. "Chaotic Particle Swarm Optimization for Imprecise Combined Economic and Emission Dispatch Problem." Review of Information Engineering and Applications 4. 1 (2017): 20-35.
5. Murugan M & Du D. H. C, "Rejuvenator: A static wear leveling algorithm for NAND flash memory with minimized overhead." Mass Storage Systems and Technologies (MSST), 2011 IEEE 27th Symposium on. IEEE, pp. 1–12, May 2011.
6. Bennoud, Salim, and Mourad Zergoug. "Evaluation and Quantification of Electromagnetic Field Distribution for Different Configurations of Aeronautical Materials." Review of Industrial Engineering Letters 3. 2 (2016): 29-37.
7. Chang. Y. H, Hsieh. J. W & Kuo. T. W, "Improving flash wear-leveling by proactively moving static data," IEEE Transactions on Computers, Vol. 59, No. 1, pp. 53–65, 2010.
8. Kim S. H, Choi J. H, & Kwak J. W, "BST: Hidden Cold Block-Aware Wear Leveling Using Bit-Set Threshold for NAND Flash Memory," IEICE TRANSACTIONS on Information and Systems, Vol. 99, No. 4, pp. 1242–1245, 2016.
9. Shoewu, O., N. O. Salau, A. O. Ogunlewe, and L. I. Oborkhale. "Path Loss Measurement and Modeling for Lagos State GSM Environments." Review of Computer Engineering Research 3. 4 (2016): 69-84.
10. Toapanta, Moisés, Enrique Mafla, and José Orizaga. "Analysis of suitable security protocols for apply a model of identity in the civil registry of Ecuador." Review of Computer Engineering Research 3. 4 (2016): 65-68.
11. Myungsub Lee, "A Tracking Cold Blocks for Static Wear Leveling in FTL-based NAND Flash Memory," The 6th International Symposium on Advanced and Applied Convergence, ISAAC2018, pp. 50-52, 2018.
12. Ali, A., & Haseeb, M. (2019). Radio frequency identification (RFID) technology as a strategic tool towards higher performance of supply chain operations in textile and apparel industry of Malaysia. *Uncertain Supply Chain Management, 7*(2), 215-226.