SLC-enabled Wear Leveling for MLC PCM Considering Process Variation

Mengying Zhao
City University of Hong Kong

Lei Jiang, Youtao Zhang
University of Pittsburgh

Chun Jason Xue
City University of Hong Kong

ABSTRACT

Phase change memory is becoming one of the most promising candidates to replace DRAM as main memory in deep silicon regime. Multi-level cell (MLC) PCM outperforms single level cell (SLC) in terms of capacity while suffering from a weaker cell endurance. Wear leveling strategies are proposed to enhance the endurance but encounters more challenges with the aggravating process variation. Due to endurance variations, balanced write traffic cannot fully exploit the PCM endurance since the weak parts will be worn out sooner than others. In this work, considering process variation, we propose an SLC-enabled wear leveling scheme through dynamic and adaptive mode transformation from MLC to SLC. Instead of redistributing write operations, the proposed scheme dynamically transforms weak and write-dense parts into SLC mode for endurance benefits. The experimental results show that the proposed scheme can improve the endurance by 215% with 4% storage overhead while maintaining the capacity advantage of MLC, compared with the most related work.

Categories and Subject Descriptors

C.3 [**SPECIAL-PURPOSE AND APPLICATION-BASED SYSTEMS**]: Real-time and embedded systems

General Terms

Design

Keywords

PCM, wear leveling, endurance, process variation

1. INTRODUCTION

Nowadays, non-volatile memory has attracted increasing research interests and plays an important role in system design [9][13][14]. Phase change memory (PCM) is proposed to serve as main memory in embedded system as DRAM is currently facing huge challenges due to the scalability limitation as well as the energy efficiency problems [15]. The advantages of PCM include non-volatility, comparable read speed as DRAM, near zero leakage power consumption and good scalability [12].

A PCM cell usually consists of a layer of chalcogenide alloys (GST) material, which can switch between low-resistance crystalline state and high-resistance amorphous state when corresponding currents are applied. There are two utilization modes of a PCM cell, pertaining to single level cell (SLC) and multi-level cell (MLC). When one PCM cell represents one bit using two resistance levels, as shown in Figure 1(a), the cell is referred to as SLC. In comparison, because the resistance values of the crystalline and amorphous states of PCM cells differ by 2-3 orders of magnitude, multiple levels of resistance can be maintained to represent multiple bits in one PCM cell, as shown in Figure 1(b). A PCM of this type is referred to as MLC PCM. Given its potential of high storage density, MLC PCM attracts increasing research interests in optimizing its performance, energy, as well as endurance. In this work, we focus on exploring the endurance enhancement of MLC PCM.

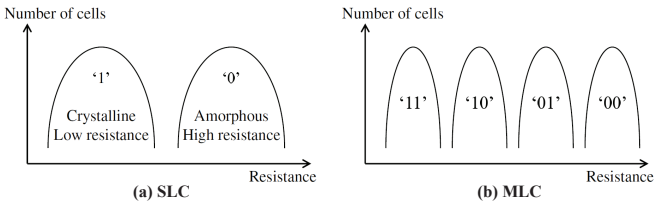

Figure 1: Resistance distributions of PCM cells.

Studies show that PCM can endure a maximum of 10^7-10^8 writes for SLC [5][6] while only 10^5 for MLC [7], which becomes the bottleneck of PCM utilization. To deal with this problem, many techniques are proposed to enhance PCM endurance, which can be summarized into two categories. One category targets the write number reduction in order to elongate the lifetime, e.g. introducing DRAM based cache [11] and differential-write [17]. The other category conducts wear leveling to evenly spread write operations across the memory, e.g. start-gap [10]. However, strategies targeting balanced write distribution cannot work well with the process variation. Due to the endurance variations, different cells can afford various number of writes thus the PCM lifetime is limited by the weakest ones even with balanced write traffic. Wear rate leveling is proposed to balance the write rate, instead of write, among the memory domains to

DAC'14, June 01 - 05 2014, San Francisco, CA, USA

improve the endurance [4]. Wear rate is defined as the number of writes on each domain over the domain's endurance, which takes the endurance variation into account.

In this paper, we explore the endurance enhancement from a different angle. Instead of redistributing write operations, SLC is employed to relieve risky cells from poor endurance and dense writes. A novel SLC-enabled wear leveling scheme for MLC PCM considering process variation is proposed. The goal is to retain MLC's capacity benefit while leveraging SLC's better endurance. The challenges lie in how to effectively utilize the limited SLC pages with minimized overhead. A dynamic threshold guided approach together with several SLC replacement policies are proposed to dynamically convert MLC pages to SLC mode for lifetime extension. The endurance variation among pages as well as write statistics are smartly encoded into the threshold implementation. In parallel, swapping is carried out to balance the writes among MLC pages. The experimental results show 215% endurance improvement over wear rate leveling [4] with 4% SLC pages. The contributions of this work are summarized as follows.

- Propose a novel wear leveling scheme that introduces SLC mode into MLC PCM to enhance the lifetime;
- Propose a dynamic threshold guided strategy for flexible MLC-SLC conversion, as well as three SLC replacement policies to exploit SLC endurance advantages;
- Evaluate the proposed strategies and compare them to related works in the literature.

The rest of this paper is organized as follows. Section 2 introduces background and related work. Section 3 presents the motivation of this work. Section 4 introduces the proposed SLC-enabled wear leveling strategy for MLC PCM, including dynamic threshold guided MLC transformation and SLC controller policies. Section 5 presents experimental evaluation and analysis. Section 6 concludes this paper.

2. PRELIMINARY

2.1 PCM utilization: SLC vs. MLC

A PCM cell applies its resistance level to represent logic bits. One SLC cell stores one bit with two resistance levels while one 2-bit MLC cell can store two bits with four resistance levels, as shown in Figure 1. Since MLC needs more precise resistance control, it adopts iterative programming-and-verifying write strategy, which reduces the write number that one MLC cell can afford before worn out. It's proved that PCM cell endurance can achieve around 10^7 - 10^8 writes for SLC while 10^5 writes for MLC [7]. Like most of the previous work, the PCM endurance in this paper is defined as the number of programming until one page is worn out and thus can not guarantee the system reliability.

The transformation between MLC and SLC has been studied regarding the capacity difference [8]. In this paper, we introduce SLC into MLC system to enhance the endurance.

2.2 Process variation in PCM

Process variation refers to the variation in the attributes of transistors (e.g. length, width, oxide thickness) when integrated circuits are fabricated. It is the natural occurrence with the increasingly shrinking fabrication size in deep silicon regime. With process variation, critical parameters shift from designed values. PCM cells require different minimum RESET and SET currents, therefore deliver various performance, energy and endurance charateristics.

Zhang et al. [16] explore the sources of process variation in PCM, including bottom electrode contact diameter size, thickness of phase change material, thickness of heater and transistor parameeters within PCM cells, concluding that process variation increases the PCM programming power by 96% and degrades the endurance by 50 times. Cintra et al. [3] also proves that existing wear leveling that assumes fixed endurance for all cells, like start gap [10], achieves negligible endurance improvement with process variation. Considering process variation, Dong et al. [4] proposes the wear rate leveling scheme to enhance PCM endurance. To the best of our knowledge, it is the most related study to our work. Wear rate leveling will be compared with the proposed schemes in the experimental evaluations.

3. MOTIVATION

As one effective metric for endurance enhancement, wear leveling spreads write operations across the memory based on the endurance characteristics. To look into the write traffic, we collect the distribution for 1 million memory writes across 730 memory pages, shown in Figure 2. We can see that the write distribution is dramatically uneven (tens of thousands times of difference). The nonuniform write traffic motivates us to exploit the endurance gap between SLC and MLC cells ($10^8 vs. 10^5$). For example, in the case of Figure 2, the write counts for each page range from 1 to 27124. There are 10 pages whose write counts are bigger than 15000. When this test bench is running on MLC PCM, we can switch these 10 pages into SLC mode to tolerate the extremely high write burden. Due to the tenacity of SLC cells, these 10 pages are released from severe wearing and thus the overall endurance can be improved by 80%. In this paper we introduce an additional small portion of memory space working as SLC to allow some of the original MLC pages to switch into SLC mode to tolerate more write operations.

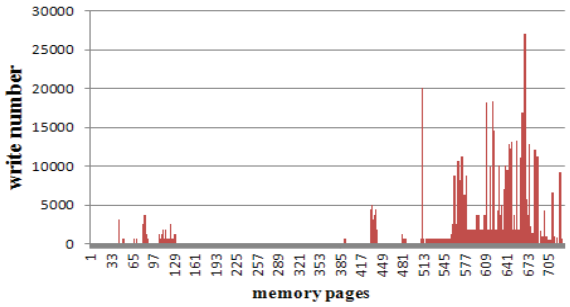

Figure 2: Write distribution for test bench *basicmath_l* in Mibench [1].

However, it is suboptimal to statically determine the MLC candidate for SLC transformation and hold them in SLC mode through the whole program execution. Besides, it becomes more complicated when process variation is involved. In this work, we explore approaches to fully exploit SLC endurance and dynamically manipulate MLC-SLC transformation.

4. SLC-ENABLED WEAR LEVELING FOR MLC PCM

In this section, we will introduce the proposed SLC-enabled wear leveling (SEWL), including dynamic threshold guided MLC transformation scheme and SLC management policies.

The challenge of this work lies in that the MLC-SLC remapping should consider both the write operation distribution in terms of application, and the endurance variation of each page in terms of device. The remapping overhead is another factor to take into account. We propose a dynamic threshold guided MLC-SLC transformation scheme to leverage heavy write workload and weak pages that easy to worn out to SLC cells in order to achieve endurance enhancement for MLC PCM. Additionally, several SLC replacement policies are analyzed and evaluated to fully exploit SLC endurance advantage.

Figure 3 describes the overview of the proposed technique. In the proposed scheme, a small amount of additional memory is introduced to support MLC pages to transform into SLC. If n MLC pages are selected to turn into SLC mode, then extra space with size of n pages are needed since the storage will be doubled after the transformation. Two main challenges are: which MLC pages to select for the transformation and how to fully exploit SLC endurance. These two problems are handled in Section 4.1 and 4.2, respectively, shown as shaded modules in Figure 3. In addition to SLC management, swapping will be conducted among remaining MLC pages for write balance.

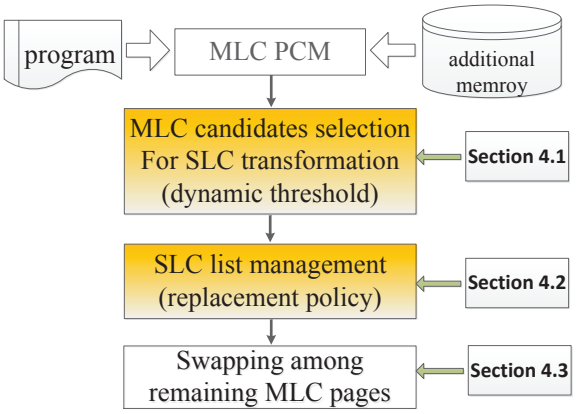

Figure 3: SLC-enabled wear leveling framework.

4.1 Dynamic threshold guided MLC-SLC transformation

For endurance enhancement, pages that are the easiest to wear out should be carefully handled as they dominate the effective lifetime. When considering process variation, on one hand, SLC cells are preferred by heavily written pages; on the other hand, SLC should take care of the "weak" pages that inherently have smaller endurance due to the fabrication variation. To integrate these two aspects, we propose a dynamic threshold guided strategy to dynamically determine the MLC candidates for SLC transformation. Basically, pages whose accumulative write number achieves their specified thresholds will be considered for SLC transformation. The main procedure is as follows.

For each page, a threshold proportional to a page's endurance is initialized. Here we assume that the endurance of different domains in PCM can be obtained via post-fabrication testing procedures. Thus initially weak pages will have lower thresholds and vice versa.

During program execution, each page keeps track of the number of write operations on this page. Once the write number of one page achieves its threshold, it reports to the SLC controller to apply for transformation to SLC mode. SLC controller is a module that approves transformation from MLC to SLC and manages SLC pages. After receiving the status report, the SLC controller evaluates and updates the SLC list. The page resets its write counter and starts over the counting from zero. The SLC controller dynamically assesses the MLC candidate pages and transforms them into SLC mode by updating the SLC list while receiving threshold-reaching messages from pages. Every time one page is transformed into SLC, its threshold is incremented by one level.

Figure 4 shows an example. During program execution, pages count write operations in their own domains. At time t_1, write access number on page j reaches its threshold. Note that the prelimiary thresholds for different pages are different due to the process variation aware initialization. After page j reports to SLC controller, its write counter is cleared accordingly, and its threshold is augmented by one level. The SLC controller will decide if page j should be transformed to SLC mode. If page j is transformed to SLC, it can tolerate much more writes with less wearing, thus significantly slow down its wearing. A page's SLC privilege ends if SLC controller accepts another transformation request and decides to evict this page out of the SLC list. Similarly, page i can also report to SLC controller when the write number achieves its threshold. For pages like page k in Figure 4, it may never reach its threshold as extremely few writes access this page. These pages will never be qualified to transfer to SLC mode since they have no worn out risk. As illustrated in Figure 3, these pages will be engaged in swappings to achieve write balance.

The inherent rationality of this dynamic thresholds lies in the following aspects. Firstly, the threshold initialization integrates the endurance information of each page so that the weak pages have a larger chance for SLC transformation

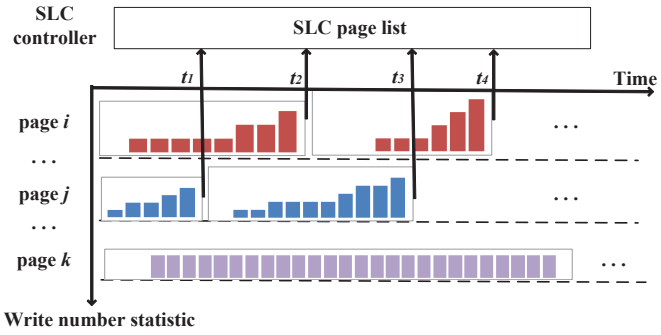

Figure 4: The proposed dynamic threshold guided MLC transformation scheme. Pages record their own write counters. Once the write number reaches its threshold, the page reports to the SLC controller to apply for SLC transformation, followed by clearing it counter and augmenting its threshold by one level.

because of a lower threshold. Secondly, the threshold scheme identifies and deals with hot/cold pages. Hot pages with more write operations reach their thresholds quickly and apply for SLC transformation more frequently, so that worn out risk can be relieved. Thirdly, the dynamic threshold augmentation balances the chance of transformation among different pages. For a hot page, even though it has heavier write burden than others, it benefits from SLC mode after transformation because of the stronger tenacity. Thus after each transformation, its threshold is increased.

4.2 SLC controller

With selected MLC candidate pages, SLC controller will determine the dynamic remapping from MLC to SLC with the objective of fully exploiting the endurance advantage of SLC. One SLC list is maintained and updated by the SLC controller. The list size is bounded by the size of extra available memory, which determines the maximal number of SLC pages in the system. When one new request is received from a page, if the current SLC page number is less than the maximal limitation, the controller will directly approve this transformation and append this page into SLC list. Otherwise, the controller needs to either reject this transformation or evict one SLC page from the list and accept the current request. As the allowed number of SLC pages is limited, it is vital to explore how to fully exploit the SLC endurance advantage. In this section, we propose three different of SLC control policies: FIFO (first in first out), LRU (least recently used) and LW (least worn), as shown in Figure 5. These policies will be evaluated in the experimental section.

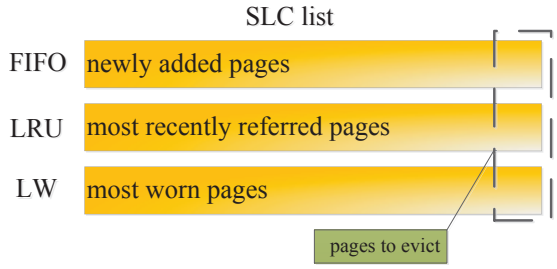

Figure 5: SLC controller with various SLC replacement policies.

4.2.1 SLC replacement: FIFO

First in first out (FIFO) is the most straightforward replacement policy. The pages in the SLC list are maintained based on the order that the pages are added to the list. So when the SLC list is full and a new request is coming, the SLC controller evicts the oldest page in the queue and appends the new page to the list. The evicted page is then switched back to MLC mode thus the space can be reserved for the newly added page for SLC transformation.

4.2.2 SLC replacement: LRU

Another SLC replacement policy is the least recently used (LRU). It is based on the assumption that the most recently used page will be referred again in the near future thus should be kept in the list. In terms of implementation, every write to SLC pages triggers an update of SLC list to record the "recently used" information. The most recently used page is kept to the front of the list and the eviction will occur at the end of the list.

Compared to FIFO, LRU takes the write behavior prediction into consideration. If one application has strong locality, LRU based SLC controller is able to achieve a higher SLC utilization and thus longer PCM lifetime, when compared with FIFO.

4.2.3 SLC replacement: LW

In addition to the write traffic, as previously discussed, the endurance variation also needs attention. When deciding the priority of SLC transformation, the controller essentially needs to consider the pages with hot writes as well as weak endurance. In the least worn (LW) policy, the least worn page, which is evaluated by threshold and endurance, is evicted if necessary.

We adopt the concept of wear rate in [4] to access the wearing of page i, which is defined as:

$$WearRate_i = \frac{writeNumber_i}{endurance_i} \quad (1)$$

In [4], the *writeNumber* for each page are recorded in a counter for wear rate based swapping. In this work, we can derive the *writeNumber* based on thresholds. Based on the definition of thresholds in Section 4.1, with threshold augmentation δ, the *writeNumber* can be calculated as

$$writeNumber_i = T_0 + T_1 + ... + T_n \quad (2)$$

where T_0 is the initial threshold and T_n represents current threshold and $T_n = T_0 + (n-1) \times \delta$.

In LW based SLC controller, the SLC page with the smallest wear rate will be evicted, which is the least risky one in terms of endurance. Consequently, this policy is the closest to the proposed principles of SLC utilization when compared to FIFO and LRU.

4.3 Swapping among MLC pages

Previous discussion explores how to take full advantage of SLC pages so that the overall endurance can be maximized. For the rest of MLC pages, we can do swapping to balance write operations. In the experimental evaluation, we employ the swapping strategy from previous works [4] since this part is not the main focus of this paper.

4.4 MLC-SLC transformation

Once one MLC page is selected to transfer into SLC mode, the content of this page will be rewritten with the assistance of additional memory space. Figure 7 shows an example. Assume additional memory with one page size is introduced (H in Figure 7), indicating that at most one SLC page is

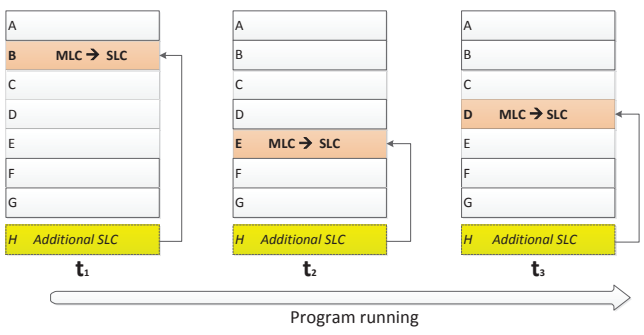

Figure 7: Transformation from MLC to SLC.

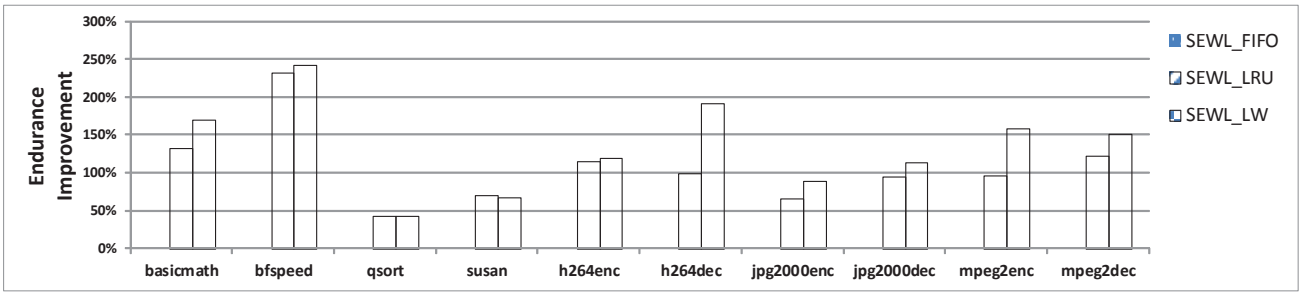

Figure 6: Endurance improvement of SEWL with a SLC percentage of 2%. The results are normalized to those of WRL.

Table 1: Endurance improvement over WRL with SLC percentage of 2%, 4% and 8%.

test bench	SEWL (2% SLC)			SEWL (4% SLC)			SEWL (8% SLC)		
	FIFO	LRU	LW	FIFO	LRU	LW	FIFO	LRU	LW
basicmath	147%	132%	169%	326%	404%	410%	645%	737%	1164%
bfspeed	258%	232%	241%	552%	481%	694%	986%	1271%	2584%
qsort	43%	42%	41%	97 %	132%	102%	155%	251%	193%
susan	67%	70%	66%	103%	124%	125 %	156%	183%	195%
h264enc	114%	114%	119%	120%	102%	454%	139%	117%	716%
h264dec	161%	98%	191%	134%	91%	325%	149%	163%	783%
jpg2000enc	66%	65%	88%	63%	63%	101%	75%	75%	126%
jpg2000dec	93%	94%	113%	108%	58%	140%	95%	92%	206%
mpeg2enc	132%	95%	158%	119%	85%	251%	107%	138%	432%
mpeg2dec	109%	121%	150%	167%	130%	398%	164%	168%	733%
Average	**120%**			**215%**			**433%**		

allowed in this system. At time t_1, Page B is selected for transformation, which means that it needs doubled space to store its original contents. This is realized by spreading the contents into Page B and H. Similarly, at time t_2, Page B is evicted from the SLC list and E is added. Then contents in B and H are compressed into original MLC format and stored into B; contents in E are extended to SLC mode and stored into E and H. Based on this scheme, the additional memory always works in SLC mode and assists MLC pages to transfer into SLC mode. The transformation is implemented by rewriting the original MLC page as well as the additional SLC page. Generally, each additional SLC page maintains a link, indicating the related original MLC page whose content is spread into the current additional page in SLC manner.

5. EXPERIMENT

In this section, we evaluate the effectiveness of the proposed SLC-enabled wear leveling (SEWL) strategy. Corresponding evaluation results are summarized with extensive analysis.

5.1 Experiment setup

In the experiments, we apply a PIN-based trace-driven simulator to compare the proposed SEWL with related works. We modeled 4K cache when deriving memory access trace. Testbenches are selected from Mibench [1] and MediaBench [2]. Endurance variation for memory pages are modeled as Gaussian distributions with standard deviation being 10% of the mean (10^5 for MLC PCM).

The wear rate leveling framework (WRL) proposed in [4] is applied as the baseline. In this strategy, program execution is divided into fixed intervals, which is specified by a certain number of write operations. During each interval, write behavior is predicted, then the wear rate of pages are derived followed by swapping pages with big wear rates and those with small wear rates. In our simulation, the swapping number is dynamically adjusted by only swapping pages whose wear rate discrepancy is larger than the predefined threshold.

For the proposed SEWL strategy, we implement the dynamic threshold scheme as well as three SLC replacement policies (FIFO, LRU, LW) for SLC management. The remaining MLC pages swap in the same manner as WRL.

5.2 Evaluation results

We evaluate the normalized endurance on test benches selected from different categories in Mibench and MediaBench. As we introduce additional space to support MLC-SLC transformation, the endurance improvements are evaluated with different introduced memory size. This size directly determines the number of SLC pages allowed in the system. We refer to this parameter as SLC percentage, i.e. the maximal SLC page percentage allowed in PCM.

Figure 6 shows the evaluation results with a fixed 2% SLC pages. For each test bench, the proposed SEWL is evaluated combined with three SLC management policies. The endurance results are normalized to those of WRL. We can observe significant endurance improvements. On average, SEWL combined with FIFO, LRU and LW achieve endurance improvement of 119%, 106% and 134%, respectively, compared to the baseline of WRL.

Table 1 summarizes the endurance improvements of SEWL over those of WRL with different SLC percentages. One general observation is that the NE increases with larger SLC sizes. The average endurance improvements of SEWL over WRL with 2%, 4% and 8% SLC pages are 120%, 215% and 433%, respectively.

5.3 Discussions

5.3.1 SLC size and replacement

Table 1 reveals that the endurance improvements rise with bigger SLC percentages. It is because that more SLC pages deliver a bigger endurance advantage during MLC-SLC transformation, while at the cost of a larger storage overhead. Flexible solutions can be derived by trading off endurance improvement and the SLC size introduced.

.

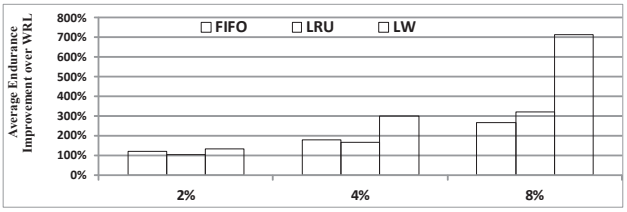

Figure 8: Endurance improvements of SLC controllers with various page replacement policies.

From Figure 8 we can see that all three SLC replacement policies can achieve significant endurance improvements when cooperated with dynamic threshold scheme. FIFO and LRU based SLC replacements have comparable performance while LW outperforms them for almost all the testbenches. The reason is that LW based SLC controller considers both the workload behavior and endurance variations by wear rate evaluation. Thus the pages with heaviest write burden and weakest endurance have more chance to be held in SLC list. LW based SLC replacement can deliver a 713% endurance improvement over WRL with 8% SLC on average.

5.3.2 Dynamic thresholds

The initial thresholds are determined based on corresponding endurance, where the endurance variations are considered. The setting of thresholds essentially decides the potential of SLC transformation. Low initial thresholds indicate more frequent SLC transformation and replacements, while higher thresholds will take pages a longer time to achieve the threshold and apply for SLC transformation.

After one page reports to the SLC controller for transformation, its threshold will be augmented by one level. Considering that the page is relieved from endurance risk during SLC phase, the threshold augmentation potentially increases the transformation chances of other pages.

5.3.3 Overhead

The storage overhead of the proposed scheme includes the extra memory space introduced for SLC transformation, and the records for threshold, write number and wear rate. The record overhead is similar with the wear rate leveling scheme [4]. The experimental results show 215% endurance improvement with extra 4% storage for SLC transformation, which is acceptable.

The performance will not be affected since the SLC access is faster than the MLC access. The performance overhead of one transformation is essentially one rewrite. Once a page is transferred to SLC mode, the performance will be generally improved because of the efficiency of SLC writes.

6. CONCLUSION

In this paper, we propose an SLC-enabled wear leveling scheme for MLC PCM to enhance the endurance. A small additional SLC memory is introduced to support a portion of original MLC pages to transfer into SLC mode. A dynamic threshold guided strategy is constructed to achieve flexible and effective MLC-SLC transformation. To fully exploit the endurance advantage of SLC pages, three SLC replacement policies are evaluated. The experimental results show 215% endurance improvement on average with 4% SLC pages, when compared with wear rate leveling.

Acknowledgment

The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. CityU 123811) and US NSF CNS-1012070.

7. REFERENCES

[1] Mibench: `http://www.eecs.umich.edu/mibench/`.
[2] MediaBench: `http://euler.slu.edu/~fritts/mediabench/`.
[3] M. Cintra and N. Linkewitsch. Characterizing the impact of process variation on write endurance enhancing techniques for non-volatile memory systems. In *SIGMETRICS*, pages 217–228, 2013.
[4] J. Dong, L. Zhang, Y. Han, Y. Wang, and X. Li. Wear rate leveling: Lifetime enhancement of pram with endurance variation. In *DAC*, pages 972–977, 2011.
[5] R. F. Freitas and W. W. Wilcke. Storage-class memory: the next storage system technology. *IBM J. Res. Dev.*, 52(4):439–447, July 2008.
[6] L. Jiang, Y. Zhang, and J. Yang. Enhancing phase change memory lifetime through fine-grained current regulation and voltage upscaling. In *ISLPED*, pages 127–132, 2011.
[7] L. Jiang, Y. Zhang, and J. Yang. Er: elastic reset for low power and long endurance mlc based phase change memory. In *ISLPED*, pages 39–44, 2012.
[8] H. G. Lee, S. Baek, J. Kim, and C. Nicopoulos. A compression-based hybrid mlc/slc management technique for phase-change memory systems. In *ISVLSI*, pages 386–391, Aug 2012.
[9] T. Liu, Y. Zhao, C. J. Xue, and M. Li. Power-aware variable partitioning for DSPs with hybrid PRAM and DRAM main memory. In *DAC*, pages 405–410, 2011.
[10] M. K. Qureshi, J. Karidis, M. Franceschini, V. Srinivasan, L. Lastras, and B. Abali. Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling. In *MICRO*, pages 14–23, 2009.
[11] M. K. Qureshi, V. Srinivasan, and J. A. Rivers. Scalable high performance main memory system using phase-change memory technology. *SIGARCH Comput. Archit. News*, 37(3):24–33, June 2009.
[12] S. Raoux, G. Burr, M. Breitwisch, C. Rettner, Y. Chen, R. Shelby, M. Salinga, D. Krebs, S.-H. Chen, H. L. Lung, and C. Lam. Phase-change random access memory: A scalable technology. *IBM Journal of Research and Development*, 52(4.5):465–479, 2008.
[13] Y. Wang, L. Bathen, N. Dutt, and Z. Shao. Meta-cure: A reliability enhancement strategy for metadata in nand flash memory storage systems. In *DAC*, pages 214–219, June 2012.
[14] Y. Wang, L. Bathen, Z. Shao, and N. Dutt. 3d-flashmap: A physical-location-aware block mapping strategy for 3d NAND flash memory. In *DATE*, pages 1307–1312, March 2012.
[15] C. J. Xue, Y. Zhang, Y. Chen, G. Sun, J. J. Yang, and H. Li. Emerging non-volatile memories: Opportunities and challenges. In *CODES+ISSS*, pages 325–334, 2011.
[16] W. Zhang and T. Li. Characterizing and mitigating the impact of process variations on phase change based memory systems. In *MICRO*, pages 2–13, 2009.
[17] P. Zhou, B. Zhao, J. Yang, and Y. Zhang. A durable and energy efficient main memory using phase change memory technology. In *ISCA*, pages 14–23, 2009.