Partial-SET: Write Speedup of PCM Main Memory

Bing Li*†, ShuChang Shan*, Yu Hu* and Xiaowei Li*
*Key Laboratory of Computer System and Architecture
Institute of Computing Technology, Chinese Academy of Sciences
†University of Chinese Academy of Sciences
Email:{libing2010, shanshuchang, huyu, lxw}@ict.ac.cn

***Abstract*—Phase change memory (PCM) is a promising non-volatile memory technology developed as a possible DRAM replacement. Although it offers the read latency close to that of DRAM, PCM generally suffers from the long write latency. Long write request may block the read requests on the critical path of cache/memory access, incurring adverse impact on the system performance. Besides, the write performance of PCM is very asymmetric, i.e, the SET operation (writing '1') is much slower than that of the RESET operation (writing '0'). In this work, we re-examine the resistance transform process during the SET operation of PCM and propose a novel Partial-SET scheme to alleviate the long write latency issue of PCM. During a write access to a memory line, a short Partial-SET pulse is applied first to program the PCM cells to a pre-stable state, achieving the same write latency as RESET. The partially-SET cells are then fully programmed within the retention window to preserve the data integrity. Experimental results show that our Partial-SET scheme can improve the memory access performance of PCM by more than 45% averagely with very marginal storage overhead.**

I. Introduction

Dynamic Random Access Memory (DRAM) has been main memory in computer system for decades. However, DRAM technology confronts the scaling difficulty of continuously scale-downing towards sub-20nm range. Hence, the emerging memory technologies which pose better scaling capability have attracted increasing attentions. Among them, Phase Change Memory (PCM) is considered as one of the most promising candidates to replace a large portion of DRAM in the main memory [1–3]. Compared with DRAM, PCM has non-volatility, superior scalability, lower standby power and comparable read latency. In addition, the PCM cells can provide higher density with the multiple-level-cell (MLC) technique, which stores multiple bits in a single cell. Meanwhile, the technique that stores one bit per cell is single-level cell (SLC). The large SLC PCM chips are already available [4] and main memory manufacturers started to move SLC PCM device towards mass production [5, 6].

However, PCM is subjected to the long write problem. What's more, the latency of SET and RESET in PCM is quite different. Since a memory line consists of both zeros and ones, the latency of write a memory line is determined by the slower SET operation, which is almost 8 timer larger than RESET latency. Hence, to improve the write performance of PCM, the key point is to reduce the latency of write '1' operation.

Although previous works [1, 2] buffered writes to mitigate the impact of PCM long writes on system performance, once the write is scheduled to serve a bank, the subsequent read access to the different line of the same bank would be blocked and the effective read latency is inevitably increased.

Various techniques have been proposed to mitigate the impacts of slow writes. Qureshi et al. proposed a write cancellation method [7]. It cancels the on-going writes and allows the read request to preempt. Jiang et al. [8] truncates the last few write iterations to finish a write earlier. However, their work views the write '1' and '0' operations as the same and relies on the write-iteration of MLC PCM.

Qureshi et al. [9], Yue and Zhu [10] utilized the asymmetry of SET and RESET operation to mitigate the slow write impact. Qureshi etal. in [9] proposed to pro-actively write '1' to the memory line when the corresponding line gets dirty in the last level cache and the memory bank is idle. When write requests of this line arrived, the memory only executes the fast write '0' operation. Yue and Zhu [10] leveraged both the latency and power asymmetry of writing one and zero in PCM. In this work's memory system, writing a cache line requires multiple serially write units and each unit contains ones and zeros. They divided the process of writing a line into a write 1 stage and multiple write 0 stages. In the write 0 stage, all zeros are written with smaller latency, and in the write 1 stage, more ones are written concurrently without exceeding the power supply. Thereby it completes write with less serially writing units and shorter time than the baseline PCM.

In this paper, our goal is to accelerate write operation and improve the memory performance. Different from the methods in [9] and [10], we reduce the latency of SET. We investigate the SET process in PCM and our insight is that during SET the resistance drops sharply to an extra lower value than RESET resistance with short time. We exploit this feature and advocate a SET pulse (Partial-SET pulse), which has comparable latency with RESET to accelerate the write operation. To the best of our knowledge, this is the first work which applies a short SET pulse to write '1'. To be distinct, we call the conventional long SET pulse as the Full-SET pulse . However, a reliability issue follows the proposed accelerated SET operation. The Partial-SET cells have shorter retention time than that of the Full-SET cells. Thus, we pro-actively fully program the Partial-SET cells within the retention window to guarantee the data integrity. The proposed Partial-SET technique include the accelerated write scheme and reliability guarantee technique.

We evaluate the overall design by simulating a PCM-based main memory system under the SPEC2006 benchmark suite. Our experimental results show that this technique effectively hides the latency asymmetry of SET and RESET operation, and improves the memory access performance by more than 45% on average over the baseline configuration.

The rest of paper is organized as follows. Section II introduces the PCM basics. Section III presents our motivation. Section IV describes the design of Partial-SET scheme. Section V explains the experimental setup and discusses the results.

Corresponding author: Yu HU, E-mail: huyu@ict.ac.cn. This work is supported in part by National Natural Science Foundation of China (NSFC Program) under Grant No.(61076018, 61274030), and in part by National Basic Research Program of China (973 Program) under Grant No.2011CB302503.
978-3-9815370-2-4/DATE14/©2014 EDAA

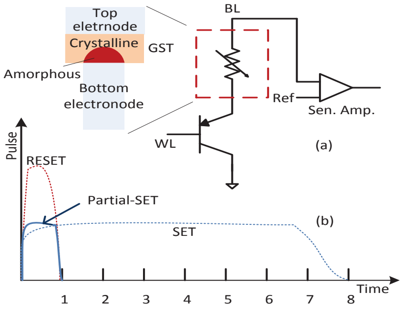


Fig. 1: (a) PCM basic structure and (b) The conventional programming pulse vs the proposed Partial-SET pulse

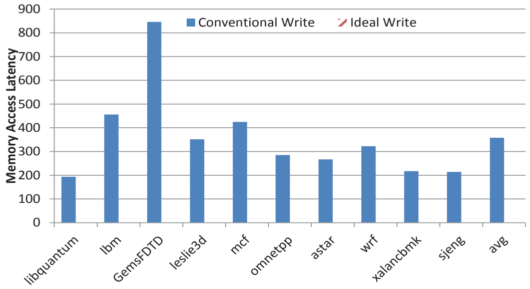


Fig. 2: Memory access latency: Conventional Write vs. Ideal Write

Section VI concludes this paper.

II. Background

Phase change memory is a type of non-volatile memory technology. Fig. 1a illustrates the basic structure of PCM cell. PCM exploits the different resistance of phase change material (typically, GST) to stores data. GST can be switched between the amorphous, high resistive RESET state and the crystalline, low resistance SET, which respectively represents ‘0’ and ‘1’.

Fig. 1b shows the PCM programming pulse. To write ‘0’, a high and short pulse (RESET pulse) is injected into the cell and then is abruptly cut off, leaving the material amorphous. To write ‘1’, a long electrical pulse (SET pulse) is required to heat the cell above the crystallization but below the melting temperature of GST. Then it sustains for a long time until the crystallization completes and the resistance reaches the targeted range. In the readout operation of PCM, the resistance is sensed out with a small short current flowing from the cell. and is compared with the reference value. A larger resistance indicates the stored value is one, otherwise, the value is zero. The reference resistance is adjustable [8] and mostly is the middle value of the resistance range.

III. Motivation

The SET and the RESET pulse are significantly asymmetric in latency, and the SET operation takes typically 8x longer time than RESET (shown in Fig. 1b). Since a memory line generally consists of both zeros and ones, the latency of write is determined by the slower operation, i.e. the SET operation.

We observed the memory access latency could be significantly reduced with a fast SET operation. We compare the memory access latency of two configurations for the workloads from SPEC2006 benchmark as shown in Fig. 2. In Conventional Write, the SET latency is eight times larger than the RESET latency [7, 9, 10]. In Ideal Write, we assume the latency asymmetry in SET and RESET is eliminated and the SET operation is as fast as the RESET. Fig. 2 shows that the average memory access latency for the conventional write is almost 400ns. For read-latency sensitive workloads, such as *GemsFDTD*, the memory access latency is strikingly more than 800ns. In ideal, the memory access latency is reduced to no more than 200ns, achieving more than 50$\%$ reduction.

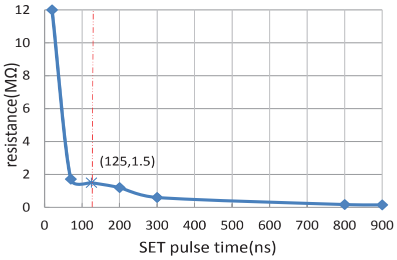


Fig. 3: The resistance is a function of a pulse duration in SET operation

Therefore, if the operations of SET and RESET are symmetric, the PCM performance would be remarkably improved. What’s more, for SET operation, the attained resistance is a function of pulse amplitude and duration [11, 12]. In the following section, we explore this feature and proposed an accelerated SET operation.

IV. Partial-SET Scheme

In this paper we use a short SET pulse to write ‘1’ which has the same width with that of RESET and is called as Partial-SET as indicated in Fig. 1b.

A. Basic Idea

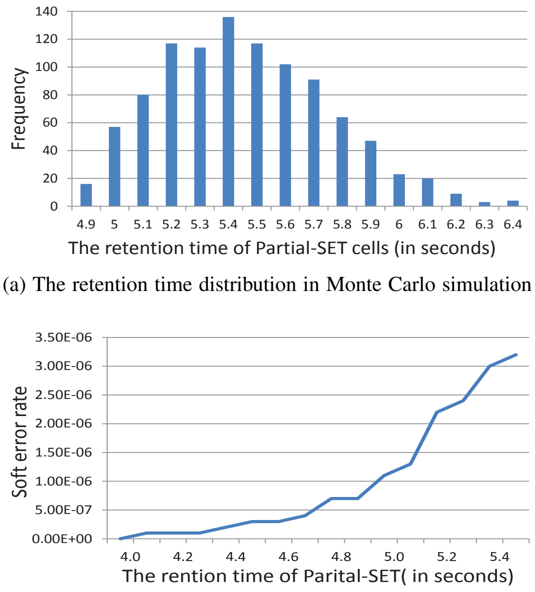
Our idea is inspired by the work in [12] which monitoring the resistance change during the SET operation. Fig. 3 shows that the resistance transition with the SET pulse. In SET process, the resistance drops steeply within a short duration. In the remaining time of SET, the resistance continuously decreases to the pre-defined SET state.

According to the Fig. 3, when a SET pulse sustains for 125ns which is the RESET latency, the resistance decreases to 1.5M ohm, which is eight times lower than the initial RESET resistance. Based on the feature in SET process, we advocate the Partial-SET pulse to reduce the write latency. Ideally, the Partial-SET pulse could achieve the same performance as that of the Ideal Write configuration. However, we find out there exists reliability challenge in Partial-SET cells.

B. Retention Window of Partial-SET cells

The resistance of PCM cell increases over time [13, 14] due to the metastable of the amorphous portion in GST. [15–19]. The phenomenon is resistance drift and exhibits a power-low model, $R_t = R_0 \times t^\nu$, where R_0 is the initial resistance of cells after write, t is the elapsed time (in seconds) and ν is the drift exponent. When R_t crosses the the reference value between RESET and the SET, the data in PCM cells are invalid.

For the cells programmed with Full-SET pulse, R_0 could be $1K\Omega$, which is low enough to retain data for years. However, for the resistance of Partial-SET cells is $1.5M\Omega$ and is relatively much closer with the RESET state. Hence their retention capability is sharply poor. We estimate the retention time of Partial-SET cells with Monte Carlo simulation. In our simulation, the exact value of ν for Partial-SET state is based on the work in [17, 18]. Moreover, the logarithm of R_0 and ν will follow the normal distribution of $N(lg1.5+6, 0.17)$ and $N(0.1, 0.14)$ respectively. We simulate 10^6 Partial-SET cells, and repeat 10^3 times. Fig. 4a is the results. The x-axis represents the simulated retention time of Partial-SET cells



(a) The retention time distribution in Monte Carlo simulation

(b) The soft error rate of Partial-SET cells at various retention time

Fig. 4: The simulated result of retention time for Partial-SET cells

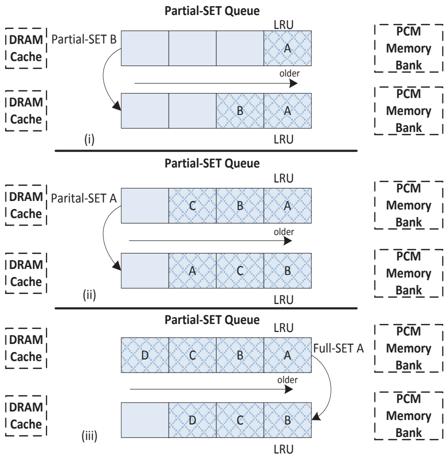


Fig. 5: An example of Partial-SET queue management

and the y-axis is the count of a certain retention time in the simulation. According to the Fig. 4a, the retention time of the Partial-SET resistance level distributes in a range from 4.9 seconds to 6.4 seconds, while in most cases the Partial SET cell begin to lose the stored data after 5.4 seconds.

Furthermore, we estimate the soft error rate of the Partial-SET memory line with various elapsed times. As illustrated in Fig. 4b, the readout value would be wrong if the elapsed time is larger than four seconds. According to the results, the retention window of Partial-SET cells is four seconds in this work. To cope with the retention reliability of Partial-SET cells, we proactively fully set these cells during the retention window.

C. Architectural Support

In this work, the PCM circuit supports two types SET pulse, one is Partial-SET, which has the same latency with RESET to accelerate write operation and the same amplitude as SET pulse, the other is the conventional SET pulse called as Full-SET. The programming circuit is modified to support the Partial-SET pulse. Meanwhile, the reference resistance is set to be 5x lower than the RESET resistance to distinguish ‘1’ and ‘0’ [8]. To prevent the data lose, the memory lines that are Partial-SET would complete a Full-SET in the retention window. Therefore, we modify the memory architecture to facilitate the Full-SET operation.

Compared with the original PCM design, we add the Partial-SET queue to each bank. Each entry in Partial-SET queue has two fields, one is the address of Partial-SET line and the other one records the elapsed time after writing. Since PCM supports the read-modify-write technique and the old value can be readout before write [3, 20], the Partial-SET queue would not allocate storage for the data value.

When a write request arrives, if the read request queue is not empty and the Partial-SET queue has spare entries, the line of the request would be written with Partial-SET pulse. The elapsed time of the entry in Partial-SET queue is reset every time its memory line is written with the Partial-SET pulse. The entry is released from Partial-SET queue when the corresponding line in memory array accepts a Full-SET write.

Besides, the Partial-SET queues support issuing the Full-SET requests. When the Partial-SET queue is full, or has a entry which reaches the retention time, the scheduler would evict the entry with largest elapsed time or the expired one from the Partial-SET queue, and issue a Full-SET request to the corresponding lines in PCM array.

Fig. 5 illustrates three kinds of scenarios in Partial-SET scheme with a 4-entry Partial-SET queue. The entries with grid pattern store the addresses and elapsed times. The capital letters represent the addresses of Partial-SET lines. When a Write B request is executed with the Partial-SET pulse and the Partial-SET queue has spare entries, then the address of B is added into the queue (Fig. 5 (i)) and its elapsed time is initialized as zero. In (ii), the line A is written with a Partial-SET pulse again, then its elapsed time is reset to zero, thus Entry B has the largest elapsed time. Subsequently, the C and D are written with the Partial-SET pulse and inserted into the Partial-SET queue. When the elapsed time of Entry A reaches the retention time (Fig. 5 (iii)), the scheduler evicts it and issues a Full-SET request to the memory line A. If a write request from upper level is written with the Full-SET pulse, its related entry in the Partial-SET queue would be released.

V. Experimental Results

A. Experimental Setup

To evaluate the efficacy of the proposed design, we used a trace-driven memory simulator DRAMsim2 [21] as our simulation platform. Our baseline configuration refers to [9] and is listed in Table I. The baseline has 32 MB 8-way DRAM cache stacked on the 4GB PCM main memory, which has 4 ranks of 8 banks each. Each bank has a request queue of 32-entry to store both write and read requests. The memory controller supports the FR-FCFS schedule policy. The read and write latency are respectively 125ns and $1\mu s$ and the Partial-SET write latency is 125ns [7, 9]. Moreover, the baseline uses the read-modify-write mode [3, 20] to reduce the number of bit flips in write operation.

According to the simulation settings shown in Table I, we calculated the required storage overhead of the proposed method. For Partial-SET scheme, the memory controller is extended with a 32-entry Partial-SET queue per bank, each entry stores the row address and the elapsed time, which

TABLE I: Baseline System Configuration

DRAM cache	32MB, 8-way, LRU, write-back, 64B line size
Memory Controller	32 entries request queue/bank, FR-FCFS scheduling
Main Memory	4GB, 4 ranks, 8chips/rank, 8 banks/chip, 64B line size, 64-bit width
PCM latency	reads: 125ns write (RESET): 1μs Partia-SET: 125ns

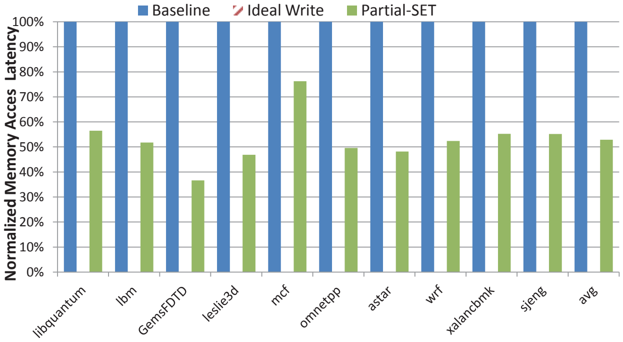


Fig. 6: Normalized Memory Access Latency for Simulated Benchmark

respectively needs 14bits and 33bits storage space. The overall storage overhead of a Partial-SET queue is $(14b + 33b) \times 32$ bit. Compared with a 32-entry request queue which requires $(64B + 14b) \times 32$ bit, the extra overhead is less than 8%.

We ran ten memory-intensive workloads from SPEC2006 [22], performed 10^{10} cycles of each application and used half of them to warmup. The memory traces were collected with the HMTT tool [23].

B. Results

Our objective is to improve the memory access performance. Thus, we compared the memory access latency of three configurations, the conventional write (Baseline), Ideal Write, and Partial-SET as illustrated in Fig. 6. The configuration Ideal Write represents an upper bound on performance improvement. For Partial-SET, we assume a 32-entry Partial-SET queue, which is the same size as the request queue in the baseline system. We normalized the latency with the baseline. The bar labeled *avg* represents the average over all workloads. On average, there is more than 45% memory access latency reduction over the conventional write, which is within 6% of the upper-bound Ideal Write configuration.

VI. Conclusion

PCM is a write-asymmetric memory technology and the speeds of write is constrained by the slower SET operation. In this paper, we propose a short SET pulse, which is called Partial-SET pulse, to accelerate the write operation. However, the Partial-SET memory lines is subjected to poor retention capability. Thus, we proposed to issue Full-SET requests in retention window to preserve the data. As the experimental result shows, the Partial-SET scheme significantly reduces the memory access latency is reduced by more than 45% on average over the baseline. In our future work, we would like to extend this scheme to the MLC PCM technique and alleviate its slow write issue.

References

[1] B. C. Lee, E. Ipek, O. Mutlu, and et al., "Architecting phase change memory as a scalable dram alternative," in *ISCA*, 2009, pp. 2–13.

[2] M. K. Qureshi, V. Srinivasan, and J. A. Rivers, "Scalable high performance main memory system using phase-change memory technology," in *ISCA*, 2009, pp. 24–33.

[3] P. Zhou, B. Zhao, J. Yang, and et al., "A durable and energy efficient main memory using phase change memory technology," in *ISCA*, 2009, pp. 14–23.

[4] C. Youngdon, S. Ickhyun, P. Mu-Hui, and et al., "A 20nm 1.8V 8Gb PRAM with 40MB/s program bandwidth," in *ISSCC*, 2012, pp. 46–48.

[5] P. Clarke., "Samsung to ship MCP with phase-change," *http://www.eetimes.com/document.asp?_id=1266495*, 2011.

[6] Micron., "Micron Announces Availability of Phase Change Memory for Mobile Devices," in *http://investors.micron.com/releasedetail.cfm?ReleaseID =692563*, 2012.

[7] M. K. Qureshi, M. M. Franceschini, and L. A. Lastras-Montano, "Improving read performance of Phase Change Memories via Write Cancellation and Write Pausing," in *HPCA*, 2010, pp. 1–11.

[8] J. Lei, Z. Bo, Z. Youtao, and et al., "Improving write operations in MLC phase change memory," in *HPCA*, 2012, pp. 1–10.

[9] M. K. Qureshi, M. M. Franceschini, A. Jagmohan, and et al., "PreSET: Improving performance of phase change memories by exploiting asymmetry in write times," in *ISCA*, 2012, pp. 380–391.

[10] J. Yue and Y. Zhu, "Accelerating Write by Exploiting PCM Asymmetries," in *HPCA*, 2013, pp. 282–293.

[11] T. Nirschl, J. Phipp, T. Happ, and et al., "Write strategies for 2 and 4-bit multi-level phase-change memory," in *IEDM*, 2007, pp. 461–464.

[12] G. W. Burr, A. Padilla, M. Franceschini, and et al., "The inner workings of phase change memory: Lessons from prototype PCM devices," in *GC Workshops*, 2010, pp. 1890–1894.

[13] D. Ielmini, D. Sharma, S. Lavizzari, and et al., "Reliability Impact of Chalcogenide-Structure Relaxation in Phase-Change Memory (PCM) Cells;Part I: Experimental Study," *TED*, vol. 56, no. 5, pp. 1070–1077, 2009.

[14] E. Ipek, J. Condit, E. B. Nightingale, and et al., "Dynamically replicated memory: building reliable systems from nanoscale resistive memories," in *ASPLOS*, 2010, pp. 3–14.

[15] Z. Wangyuan and L. Tao, "Helmet: A resistance drift resilient architecture for multi-level cell phase change memory system," in *DSN*, 2011, pp. 197–208.

[16] N. Papandreou, H. Pozidis, T. Mittelholzer, and et al., "Drift-Tolerant Multilevel Phase-Change Memory," in *IMW*, 2011, pp. 1–4.

[17] W. Xu and T. Zhang, "A Time-Aware Fault Tolerance Scheme to Improve Reliability of Multilevel Phase-Change Memory in the Presence of Significant Resistance Drift," *TVLSI*, vol. 19, no. 8, pp. 1357–1367, 2011.

[18] M. Awasthi, M. Shevgoor, K. Sudan, and et al., "Efficient scrub mechanisms for error-prone emerging memories," in *HPCA*, 2012, pp. 1–12.

[19] N. H. Seong, S. Yeo, and H. S. Lee, "Tri-level-cell phase change memory: toward an efficient and reliable memory system," in *ISCA*, 2013, pp. 440–451.

[20] C. Sangyeun and L. Hyunjin, "Flip-N-Write: A simple deterministic technique to improve PRAM write performance, energy and endurance," in *MICRO*, 2009, pp. 347–357.

[21] P. Rosenfeld, E. Cooper-Balis, and B. Jacob, "DRAMSim2: A Cycle Accurate Memory System Simulator," *CAL*, vol. 10, no. 1, pp. 16–19, 2011.

[22] J. L. Henning, "SPEC CPU2006 benchmark descriptions," *SIGARCH*, vol. 34, no. 4, pp. 1–17, 2006.

[23] Y. Bao, M. Chen, Y. Ruan, and et al., "HMTT: a platform independent full-system memory trace monitoring system," in *SIGMETRICS*, vol. 36, 2008, pp. 229–240.