JOURNAL OF SEMICONDUCTOR TECHNOLOGY AND SCIENCE, VOL.16, NO.3, JUNE, 2016
http://dx.doi.org/10.5573/JSTS.2016.16.3.330
ISSN(Print) 1598-1657
ISSN(Online) 2233-4866

Improving Energy Efficiency and Lifetime of Phase Change Memory using Delta Value Indicator

Ju Hee Choi and Jong Wook Kwak*

Abstract—**Phase change memory (PCM) has been studied as an emerging memory technology for last-level cache (LLC) due to its extremely low leakage. However, it consumes high levels of energy in updating cells and its write endurance is limited. To relieve the write pressure of LLC, we propose a delta value indicator (DVI) by employing a small cache which stores the difference between the value currently stored and the value newly loaded. Since the write energy consumption of the small cache is less than the LLC, the energy consumption is reduced by access to the small cache instead of the LLC. In addition, the lifetime of the LLC is further extended because the number of write accesses to the LLC is decreased. To this end, a delta value indicator and controlling circuits are inserted into the LLC. The simulation results show a 26.8% saving of dynamic energy consumption and a 31.7% lifetime extension compared to a state-of-the-art scheme for PCM.**

Index Terms—**Phase change memory (PCM), non-volatile memory (NVM), energy saving techniques, write endurance**

I. INTRODUCTION

PCM adopts a chalcogenide alloy (Ge2Sb2Te5, or GST) to store its information [1, 2]. A physical state of a PCM cell is used for indicating 0 or 1, while the level of the electrical currents is used for the conventional memories such as SRAM or DRAM. Each PCM cell has either a crystalline state or an amorphous state. If it is the crystalline state, its value means 0. On the other hand, the amorphous state shows that the corresponding cell has 1.

The main advantage of PCM is that it consumes very little static power. Since the electrical power is not dissipated to sustain the information, GST cells have no leakage power. However, compared to SRAM, the write operations require higher energy consumption and longer latency to be completed because changing the phases needs a significant energy consumption and longer duration time to heat the cells.

To mitigate these problems, the researchers proposed various schemes to reduce the number of the write accesses to PCM-based last-level cache (LLC) [3-5]. The narrow-width values (NWV) [3] is one of the newest schemes, which stores partial data excluding the leading 0s. It employed other previous researches such as the read-before-write (RBW) [4] and the multiple dirty bit (MDB) [5] because they can be orthogonally applied on the NWV.

However, they overlooked reducing the write counts of PCM when the value is not significantly changed. As we analyze it, a large portion of write operations come from small-value changes. Thus, if the delta value is stored instead of updating the new value, the write counts of the LLC will be decreased.

Based on this observation, we propose a delta value indicator (DVI) scheme to save the difference between the value currently stored in the LLC and the new value to be written into the LLC. In the DVI, a delta value array (DVA), which consumes less write dynamic energy, is inserted beside the data array. Instead of updating the data array in the LLC which requires higher write

Manuscript received Oct. 28, 2015; accepted Dec. 28, 2015
Department of Computer Engineering, Yeungnam University, 280 Daehak-Ro, Gyeongsan, Gyeongbuk, 38541, Republic of Korea
E-mail : kwak@yu.ac.kr

dynamic energy, the delta value is stored in the DVA, which reduces the dynamic energy consumption.

The rest of this paper is organized as follows. In Section 2, the related works are briefly reviewed. Section 3 provides the motivation of this paper. In Section 4, we explain a delta value indicator. Section 5 gives the experimental setup and the simulation results. The conclusion is given in Section 6.

II. Related Work

Read-before-write techniques have been proposed to compensate the drawbacks of PCM such as high write dynamic energy consumption and limited write endurance [3-8].

Researchers introduced the read-before-write (RBW) scheme to eliminate the unnecessary write operations to the PCM [4]. As its name stands for, the data in the cache line to be updated are read before initiating the write operation. After comparing the currently stored data and the new data bit by bit, only the bits which should be changed are toggled.

The multiple dirty bit (MDB) scheme is motivated by the idea that parts of words in the LLC are updated during the write-back [5]. The traditional cache has a dirty bit for each line, however, the MDB inserted a dirty bit for each word. Thus, the words in the cache line, which are modified in the upper level cache, are updated instead all words in the cache line are written. In other words, clean words are not re-written. As a consequence, this MDB scheme reduced the number of write operations by avoiding unnecessary writes to the LLC.

The narrow-width values (NWV) scheme utilized that the upper half of a small value consists of all zero values [3]. They added a flag bit which is called a narrow bit to indicate whether the data is small value. When the data of an incoming cache line is small value, the flag bit is set and only the lower half of the data is written. During the read operation, if the flag bit is set, access to the upper half of the cache line is omitted.

On top of that, Flip-n-write [6] toggles all bits of the new data if the flipped value requires less number of write operations. SoftPCM is proposed for the error tolerable applications such as video applications [7]. If the currently stored data in PCM are similar to the new data, the memory controller cancels the write operation. FlipMin proposed by Jacobvitz finds a coset of vectors which minimizes the number of bits to be updated and remapping a vector to the original data [8].

In addition, researchers have been proposed hybrid PCM-DRAM schemes [9-11] which combine two kinds of memories. Main memory of the hybrid memory system is mainly composed of PCM with a small portion of DRAM. Reducing PCM write-traffic is motivated by the fact that the write latency of PCM are significantly higher than that of DRAM. To this end, the data are merged and written to improve the write latency and reduce the dynamic energy of PCM main memory [12-14].

II. Motivation

Our proposal starts from the observation that a small-value change generates a large portion of write operations. All previous works we introduced focused on the value to be written itself. However, they overlooked a possibility that it is important that the difference between the currently stored value and the value to be updated. When the delta value of the two data is small, storing the difference can save the energy consumption and enhance the write endurance of the LLC.

Fig. 1 shows the distribution of the differences between the current value and the new value when writing the new data. Note that each value has a 32-bit width. Overall, more than 25% of the write operations occur when the value is increased or decreased by at most 16. In case of *astar,* more than 50% of the write counts are small-value range. On the contrary, some applications such as *namd* and *hmmer* show that the write

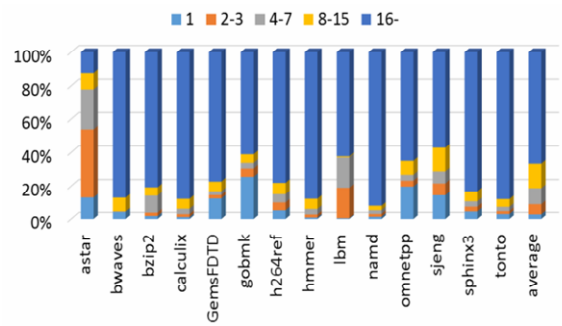


Fig. 1. The distribution of the differences between the current value and the new value. (e.g. the index 2-3 means that the original value subtracted from the modified value is +2, +3, -2, or -3).

counts due to small-value change occupy around 10% of the total write counts.

This observation provides the insight whereby storing this difference value into a new small storage, instead of directly updating PCM cells in LLC, reduces the total write counts of the LLC. Therefore, employing a small device on top of these schemes such as the RBW, the MDB, and the NWV, we achieve greater reduction in the number of write accesses to the LLC, which positively affects energy efficiency and lifetime extension.

III. Delta Value Cache

1. Overall Structure

The DVI contains the difference between the value currently stored in the data array and the new value to be written into the data array as depicted in Fig. 2. The LLC of our mechanism contains the DVA, as well as the tag array and the data array, and some combinational logics for the subtraction, summation, and comparison of two values. Each entry in the DVA consists of DVIs. Each DVI consists of m bits to store the delta value for n bits. In this paper, we assume that n is fixed at 32 bits.

Fig. 3 shows a detailed description of the control logic. The subtraction of the current value and the new value is executed during the write operation. To determine if the difference between two values is small or not, small delta value detection logic is inserted as depicted in the right-side of Fig. 3. If left-most n - m bits of the substractor are zero (upper logic) or one (lower logic), the delta value is within the threshold range ($-2^{m-1} \leq \Delta \leq 2^{m-1} - 1$). In that case, the m bits are written into the DVI and the write enable signal for the data array is disabled. However, if the result is determined to not be a small delta value, the cache block is updated as the new value and the delta value indicator contains 0. During read operation, the current value is obtained by adding the

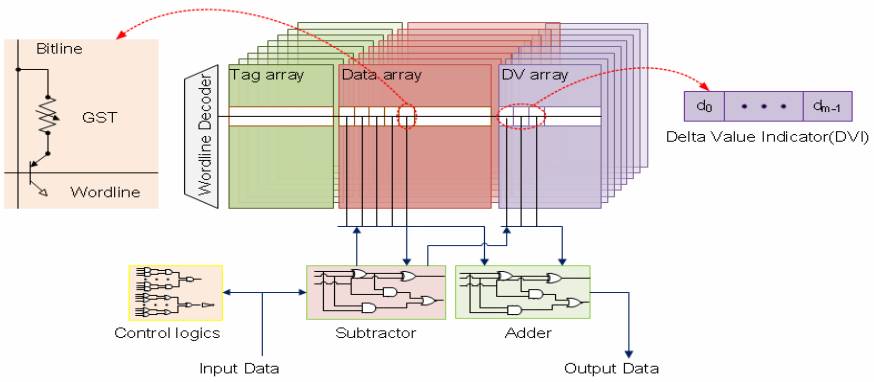


Fig. 2. The overall structure of the DVI. The LLC has the delta value array (DVA) as well as the tag array and the data array.

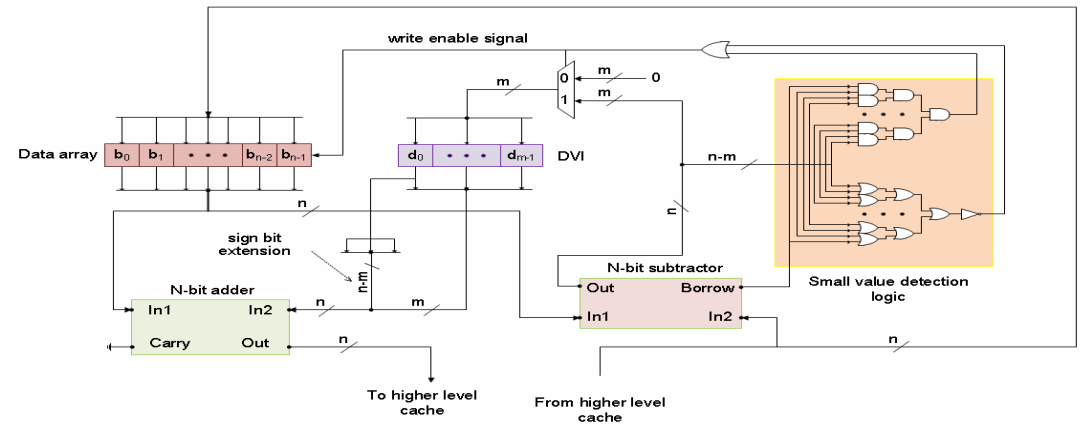


Fig. 3. A detailed description of the DVI circuits. Small delta value detection logic blocks the write enable signal if all upper bits (output[m:n-m-1]) are all 0s or 1s, because the output value is considered as small value.

Table 1. Parameters for NVSim

Design target	Cache
Optimization target	Write dynamic energy
Process node	32 nm
Device roadmap	High performance (HP)
Capacity	16 MB
Word width	256 bits

delta value and the value in the cache. The number of sets and ways of the DVA is the same as those of the tag array or the data array. Some combinational circuits such as subtractors, adders, and control logics, are required. Since our proposal contains all schemes used in the previous works such as the RBW, the MDB, and the NWV without having any factors to worsen the performance, the result of the DVI is always the same or better than their works.

2. Dynamic Energy Consumption Model

To evaluate the energy consumed by the LLC and the DVA, we use NVSim which calculates the several parameters of non-volatile memories [15]. We use high performance (HP) cells under 32 nm technology which is designed for cache. The LLC and the DVA are optimized for low write dynamic energy consumption. Detailed information of the parameters is listed in Table 1. The energy consumption models used in NVSim are divided into three models: read energy consumption (E_{LLC_RD} and E_{DVA_RD}), write energy consumption for SET operation (E_{LLC_WRS} and E_{DVA_WRS}), and write energy consumption for RESET operation (E_{LLC_WRRS} and E_{DVA_WRRS}).

We firstly define E_{LLC_RD} to be the dynamic energy of read operation as follows:

$$E_{LLC\_RD} = E_{HTR} + E_{PRED} + E_{ROWD} + E_{MUX} + E_{RD} + E_{SA} + E_{PREC} \quad (1)$$

where E_{HTR}, E_{PRED}, E_{ROWD}, E_{MUX}, E_{RD}, E_{SA}, and E_{PREC} are the dynamic energy of the h-tree, the pre-decoder, the mux, reading PCM cells, the sense amplifiers, and the pre-charge logic, respectively.

For PCM, write operations are categorized by SET operation and RESET operation. Switching state “0” to state 1” is called SET operation and changing state “1” to state “0” is called RESET operation. The point is that energy consumption of two operations are different.

Table 2. Dynamic energy consumption for access

LLC	Read operation	0.793 nJ
	SET operation	11.663 nJ
	RESET operation	6.257 nJ
DVA (m=1)	Read operation	0.023 nJ
	SET operation	0.428 nJ
	RESET operation	0.259 nJ
DVA (m=2)	Read operation	0.034 nJ
	SET operation	0.791 nJ
	RESET operation	0.453 nJ
DVA (m=3)	Read operation	0.045 nJ
	SET operation	1.153 nJ
	RESET operation	0.646 nJ
DVA (m=4)	Read operation	0.056 nJ
	SET operation	1.515 nJ
	RESET operation	0.840 nJ

Table 3. Area comparison

LLC	0.627 mm^2
DVA (m=1)	0.012 mm^2
DVA (m=2)	0.022 mm^2
DVA (m=3)	0.034 mm^2
DVA (m=4)	0.044 mm^2

Therefore, there are two models for the energy consumption of write operation: E_{LLC_WRS} and E_{LLC_WRRS}.

$$E_{LLC\_WRS} = E_{HTR} + E_{PRED} + E_{ROWD} + E_{MUX} + E_{SET} \quad (2)$$

$$E_{LLC\_WRRS} = E_{HTR} + E_{PRED} + E_{ROWD} + E_{MUX} + E_{RESET} \quad (3)$$

where E_{SET} is the dynamic energy for changing state “0” of PCM cells and E_{RESET} is the dynamic energy for writing “1” to PCM cells.

The dynamic energy consumption models for the DVA are similar to the models for the LLC, however, the energy for the h-tree (E_{HTR}) is removed because the h-tree is not used due to size limitation.

$$E_{DVA\_RD} = E_{PRED} + E_{ROWD} + E_{MUX} + E_{RD} + E_{SA} + E_{PREC} \quad (4)$$

$$E_{DVA\_WRS} = E_{PRED} + E_{ROWD} + E_{MUX} + E_{SET} \quad (5)$$

$$E_{DVA\_WRRS} = E_{PRED} + E_{ROWD} + E_{MUX} + E_{RESET} \quad (6)$$

Table 2 shows the dynamic energy consumption of the LLC and the DVA where m varies from 1 to 4 using our model. The dynamic energy consumption for access of the DVA_m=1, DVA_m=2, DVA_m=3, and DVA_m=4 are 3.67%, 6.78%, 9.89%, and 13.0% of the SET operation nergy consumption for access of the LLC,

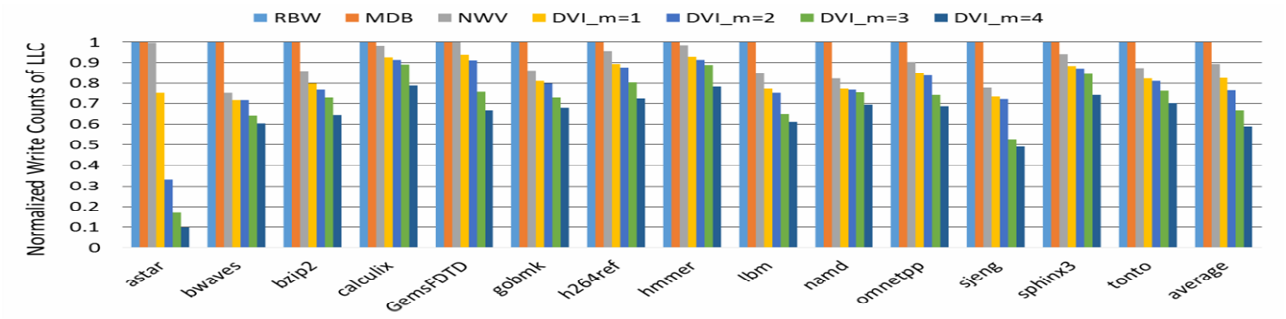


Fig. 4. Normalized write counts of LLC of the RBW, the MDB with the RBW, the NWV, and the DVI (m varies from 1 to 4).

respectively.

3. Area Overhead

We use a MOS-accessed cell model [15] to estimate the storage overhead of the DVA, because we assume that PCM consists of typical 1-transistor-1-resistor (1T1R) structure.

$$AREA_{OVERHEAD} = \frac{n}{c} * m * 3 * \left(\frac{W}{L} + 1 \right) \quad (7)$$

C is the width of a cache block; in this paper, this value is 512, because the size of a cache block is 512 bits. In addition, the W/L means the width-to-length ratio of the cell. Since the size of cell should be enough to endure write current, the W/L ratio, which is closely related to the size of NMOS, is determined by the driving current of NMOS (I_{DS}). If NMOS is operating at the linear region, (8) is applied. Unless, (9) is used for estimating the I_{DS}.

$$I_{DS} = K * \frac{W}{L} * \left[(V_{GS} - V_{TH}) * V_{DS} - \frac{V_{DS}^2}{2} \right] \quad (8)$$

$$I_{DS} = \frac{K}{2} * \frac{W}{L} * \left[(V_{GS} - V_{TH})^2 * (1 + \lambda V_{DS}) \right] \quad (9)$$

As a result of our calculation shown in Table 3, the storage overhead is 1.91%, 3.51%, 5.42%, and 7.02% where the m varies from 1 to 4. The area overhead increases as the m value increases, while the dynamic energy consumption decreases accordingly. Therefore, the m value can be selected by a system designer depending on the requirement of the system.

IV. Simulation Results

1. Simulation Environments

To evaluate the proposed mechanism, we conducted a simulation using the gem5 simulator [16], which is a representative full-system simulator. Memory hierarchy consists of an L1 32KB instruction cache and an L1 32KB data cache as well as 4MB PCM-based L2 cache in our environment. Detail information of the baseline configurations is shown in Table 4.

The benchmark programs were chosen from the SPEC CPU2006 benchmark suite [17]. We executed one billion instructions for each program after fast-forwarding one billion instructions. There are five cache systems for the experiment: the RBW, the MDB with the RBW, the NWV, and DVI with four m values which vary from 1 to 4. Note that the NWV includes the RBW and the MDB. In addition, the DVI is based on the NWV, thus it is assumed that the RBW with the MDB is applied to the

Table 4. Parameters of the simulation environments

Datapath Width	8 inst. per cycle, out-of-order, 2GHz
Fetch Queue	32 entries
Inst. Queue	64 entries
Load/Store Queue	64 entries
Function Units	6 Int ALU, 2 Int Mult/Div, 4 Float ALU, 2 Float Multi/Div, 4 Mem Ports
Branch Predictor	ALPHA 21264 tournament predictor, 16 RAS, BTB 4096 entries
L1 Inst./Data Cache	32KB, directed mapped, 32B line, access latency : 2 cycles
L2 Unified Cache	4MB unified, 16-way, 64B line, access latency : read 27 cycles write 64 cycles
Memory	DDR SDARM, data bus width : 64 bits, access latency : 190 cycles

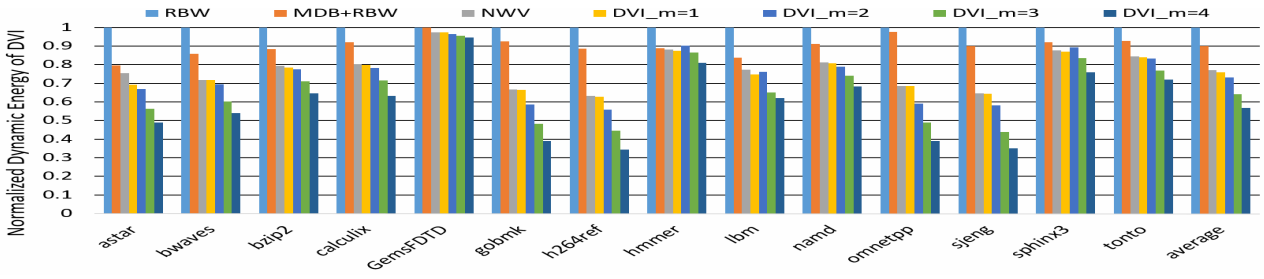


Fig. 5. Normalized dynamic energy of the RBW, the MDB with the RBW, the NWV, and the DVI (m varies from 1 to 4).

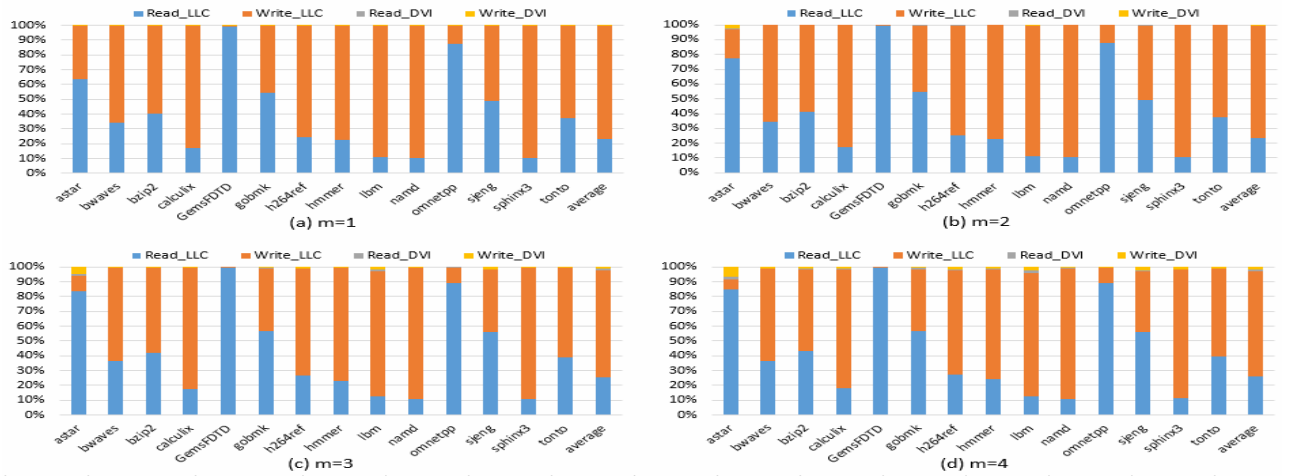


Fig. 6. Breakdown of dynamic energy of the DVI (m varies from 1 to 4).

DVI. Finally, it is assumed that there is an inclusion property between the LLC and the L1 cache as like many modern processors [18].

2. Normalized Write Counts of LLC

To investigate the effectiveness of our proposal, we examined the normalized write counts of the baseline with several DVI configurations, as shown in Fig. 4. The standard of normalization is the write counts of the RBW. Since the write counts of the RBW is the same as that of the MDB, all their values are 1. The NWV shows a 13.1% reduction in the write counts of the LLC. The number of write accesses to PCM is decreased by 41.2% at m=4, whereas the write counts were reduced by 33.2% at m=3.

For DVI_m=2, the number of write operations was reduced by 23.3%. When the value of m is 1, the variation in the write counts is 17.4%. All applications show the better results than the baseline. Especially, nearly 90% of the write counts are reduced for *astar*. These results are consistent with the distribution of the differences between the current value and the new value as depicted in Fig. 1.

3. Normalized Energy Consumption of DVI

Fig. 5 provides the normalized dynamic power consumption of the RBW, the MDB with the RBW, the NWV, and the DVI. The DVI with m=4 achieved 47.9%, 42.2%, and 17.0% energy savings on average compared to the RBW, the MDB with the RBW, and the NWV, respectively. Inspecting the result, the trend is found that the energy consumption decreases as the m value increases.

Even though there are a few exceptions such as *lbm* and *sjeng*, our proposal averagely reduced the dynamic energy consumption by storing the delta value instead of updating the new value.

To study the energy overhead of access to the DVI, we divided the results in the graph into four kinds of energy consumption as shown in right side in Fig. 6. RD_LLC

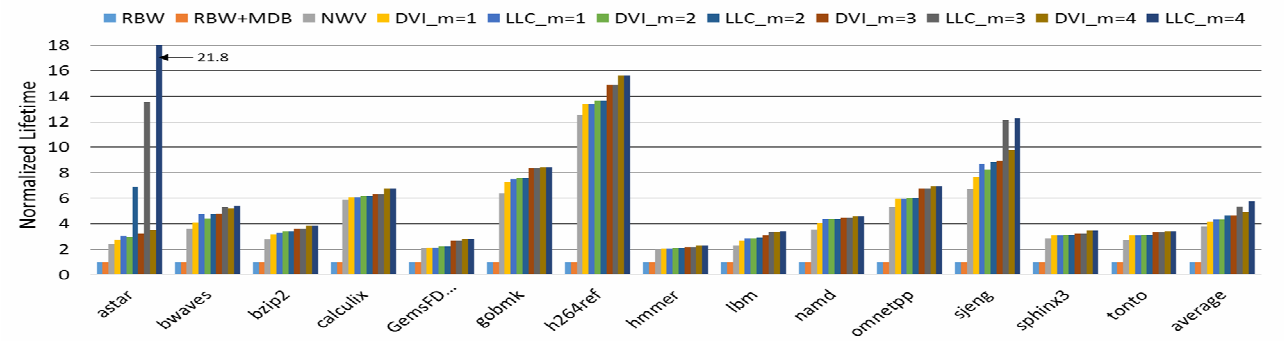


Fig. 7. Normalized lifetime of the RBW, the MDB with the RBW, the NWV, and the DVI (m varies from 1 to 4).

means the energy consumption of the read operation of the LLC and WR_LLC indicates the energy consumption of write operation of the LLC. The dissipated energy for read operation and write operation of the DVI is expressed as RD_DVA and WR_DVA. When the value of m is 1, the energy overhead is less than 1% for all applications. Although the energy overhead is increased as the value of m increases, even in the case of DVI_m=4, the extra energy consumption generated by the DVI does not exceed 3% on average. All benchmarks show less than 3% energy overhead except *astar* and *lbm*, which show 8.6% and 3.5% extra energy consumption, respectively.

4. Lifetime Improvement of DVI

The lifetime extension of the proposed scheme is shown in Fig. 6. The lifetime is defined by the time when a first worn-out cell occurs; thus, the lifetime extension is calculated by the ratio of the lifetime of the RBW over the lifetime of our proposal. Overall, the write endurance is improved by 15.64 times at maximum, and thus the lifetime is prolonged by 4.94 times on average at m=4. Compared to the NWV, lifetime is extended by 31.7%. In general, the normalized lifetime increase as the m value grows for each application. In addition, every application has longer lifetime than that of the RBW and the NWV.

We also depict the lifetime extension of the LLC separated from the DVI in Fig. 7. The DVI consists of the DVA and the LLC, thus the lifetime of the DVI is determined as the shorter lifetime between the DVA and the LLC. Since the DVA is also composed of PCM cells, the DVA can be a constraint of the lifespan of our proposal when the heavy write operations are conducted on the DVA. In this case, the lifetime of the whole DVI memory system is limited to the lifetime of the DVA. For example, *astar* achieved a 21.8 times lifetime extension for the LLC, while the actual lifetime of the whole memory system is prolonged by 3.5 times. However, all applications except *astar, bwaves, lbm*, and *sjeng* show the normalized lifetime of the DVI is the same as the normalized lifetime of the LLC. Moreover, for *bwaves* and *lbm*, the lifetime difference between the DVA and the LLC is less than 3%. In general, employing the DVA helps the lifetime extension.

V. CONCLUSIONS

This paper proposed a delta value indicator (DVI) to reduce the write counts in PCM-based LLC. First, our proposal showed and analyzed the portion of write counts due to small-value change. This discovery resulted in employing small storage to hold the difference between the original value and the new value to be written. The experiment result showed that our scheme achieves a 34.1% reduction in write counts, which led to saving dynamic energy consumption by 26.4% and prolong the lifetime by 31.7% compared to the NWV including the RBW and the MDB.

ACKNOWLEDGMENTS

This work was supported by the 2014 Yeungnam University Research Grant.

REFERENCES

[1] C. S. Hwang and C. Y. Yoo. "Atomic layer deposition for semiconductors," Springer, 2014.

[2] S. Raoux, et al, "Phase-change random access memory: A scalable technology," *IBM Journal of Research and Development*, Vol.52, No.4, pp.465-479, Jul., 2008.

[3] G. Duan and S. Wang, "Exploiting narrow-width values for improving non-volatile cache lifetime," *In Proceedings of the Conference on Design, Automation and Test in Europe,* pp.52:1-52:4, Mar., 2014.

[4] Y. Joo, D. Niu, X. Dong, G. Sun, N. Chang, and Y. Xie, "Energy-and endurance-aware design of phase change memory caches," *In Proceedings of the Conference on Design, Automation and Test in Europe*, pp.136-141, Apr., 2010.

[5] S. Wang, J. Hu, and S. G. Ziavras, "On the characterization and optimization of on-chip cache reliability against soft errors," *IEEE Transactions on Computers*, vol.58, no.9, pp.1171–1184, Sept., 2009.

[6] S. Cho and H. Lee, "Flip-n-write: a simple deterministic technique to improve pram write performance, energy and endurance," *In Microarchitecture, 42nd Annual IEEE/ACM International Symposium on*, pp. 347-35, Dec., 2009.

[7] Y. Fang, H. Li, and X. Li, "Softpcm: Enhancing energy efficiency and lifetime of phase change memory in video applications via approximate write," *In Test Symposium (ATS), 2012 IEEE 21st Asian*, pp.131-136, Nov., 2012.

[8] A. N. Jacobvitz, R. Calderbank, and D. J. Sorin, "Coset coding to extend the lifetime of memory," *In High Performance Computer Architecture (HPCA2013)*, pp.222-233, Feb., 2013.

[9] T. J. Ham, B. K. Chelepalli, N. Xue, and B. C. Lee, "Disintegrated control for energy-efficient and heterogeneous memory systems," *In High Performance Computer Architecture (HPCA2013)*, pp.424-435, Feb., 2013.

[10] H. A. Khouzani, C. Yang, and J. Hu, "Improving performance and lifetime of dram-pcm hybrid main memory through a proactive page allocation strategy," *In Design Automation Conference (ASP-DAC), 20th Asia and South Pacific*, pp.508-513, Jan., 2015.

[11] W. Hwang and K. Park, "Hmmsched: hybrid main memory-aware task scheduling on multicore system," *In Proceedings of international conference on future computational technologies and applications*, pp.39-48, May, 2013.

[12] J. Hu, C. J. Xue, Q. Zhuge, W.-C. Tseng, and E. H.-M. Sha, "Write activity reduction on non-volatile main memories for embedded chip multiprocessors," *ACM Transactions on Embedded Computing Systems (TECS)*, Vol.12, No.3, pp.9:1-9:25, Apr., 2013.

[13] F. Xia, D. Jiang, J. Xiong, M. Chen, L. Zhang, and N. Sun, "DWC: Dynamic write consolidation for phase change memory systems," *In Proceedings of the 28th ACM international conference on Supercomputing*, pp.211-220. Dec., 2014.

[14] Y. Huang, T. Liu, and C. J. Xue, "Register allocation for write activity minimization on non-volatile main memory for embedded systems," *Journal of Systems Architecture*, Vol.58, No.1, pp.13-23, Jan., 2012.

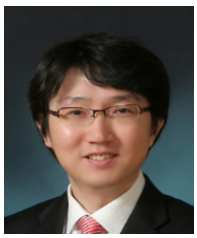
[15] X. Dong, C. Xu, Y. Xie, and N. P. Jouppi, "Nvsim: A circuit-level performance, energy, and area model for emerging nonvolatile memory," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, Vol.31, No.7, pp.994-1007, July, 2012.

[16] J. Power, J. Hestness, M. Orr, M. Hill, and D. Wood, "Gem5-gpu: A Heterogeneous CPU-GPU Simulator," *IEEE Computer Architecture Letters*, Vol.14. No.1, pp.34-36, Jan., 2015.

[17] J. L. Henning, "Spec CPU2006 benchmark descriptions," *ACM SIGARCH Computer Architecture News*, Vol.34, No.4, pp.1-17, Sept., 2006.

[18] Intel core i7 processor, "http://www.intel.com/products/processor/corei7/specifications.htm," accessed 1-Jan-2016.

Ju Hee Choi received a B.S. degree in Computer Science from Yonsei University in 2004 and received a M.S. degree in Computer Science and Engineering from Seoul National University in 2006. He is currently working towards a PhD degree at Seoul National University. Also, he has working in the S.LSI, at Samsung Electronics Co., Ltd from 2006. His research interests include computer architectures, low-power design, multi-core system, and embedded system design.



Jong Wook Kwak received the B.S. degree in Computer Engineering from Kyungpook National University, Daegu, Korea in 1998, the M.S. degree in Computer Engineering from Seoul National University, Seoul, Korea in 2001, and the Ph.D. degree in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea in 2006. From 2006 to 2007, he worked as a senior engineer in SoC R&D Center, Samsung Electronics Co., Ltd. Since 2007, he has been joined as an associate professor in Department of Computer Engineering, Yeungnam University. In 2011, he had been also served as a visiting scholar at Institute of Computer Technology in Seoul National University and during 2012~2013, he was a visiting professor at Georgia Institute of Technology, GA, USA. His lectures cover operating systems, computer architectures, embedded hardware/software designs, etc. His research interests include advanced computer architectures, wireless mobile embedded systems and high-performance parallel and distributed computing.