Peak-to-average Pumping Efficiency Improvement for Charge Pump in Phase Change Memories

Huizhang Luo*, Jingtong Hu†, Liang Shi*§, Chun Jason Xue‡ and Qingfeng Zhuge*
*College of Computer Science, Chongqing University, Chongqing, China.
†Oklahoma State University, U.S.A.
‡Department of Computer Science, City University of Hong Kong, Hong Kong.

***Abstract*—The emerging Phase Change Memory (PCM) is considered as a promising candidate to replace DRAM as the next generation main memory since it has better scalability and lower leakage power. However, the high write power consumption has become a main challenge in adopting PCM as main memory. In addition to the fact that writing to PCM cells requires high write current and voltage, current loss in the charge pumps (CPs) also contributes a large percentage of the high power consumption. The pumping efficiency of a PCM chip is a concave function of the write current. Based on the characteristics of the concave function, the overall pumping efficiency can be improved if the write current is uniform. In this paper, we propose the peak-to-average (PTA) write scheme, which smooths the write current fluctuation by regrouping write units. An off-line optimal Integer Programming (IP) formulation and an efficient online algorithm are proposed to achieve this goal. Experimental results show that PTA can improve the charge pump efficiency to ∼40% with little overhead. Meanwhile, PTA can achieve 17.0% energy reduction on average.**

I. Introduction

In recent years, researchers have been looking for replacements of DRAM. The emerging non-volatile memory (NVM) techniques have attracted many attentions due to their advantages such as high density, ultra-low leakage power, low-cost, and non-volatility. Among all NVMs, PCM [1][2][3][4] has been one of the most promising candidates to replace DRAM as main memory.

Though PCM has many advantages compared with DRAM, its high write power consumption has become one of the main challenges in adopting PCM as main memory. Two major factors contribute to the high write power consumption. First, changing the states of PCM cells requires higher write current and voltage than DRAM. For example, 3.0V and 5.0V are needed for SET and RESET operations on a PCM cell. They are much higher than the 1.5V for DRAM writes. The current required to program a PCM cell is orders of magnitude higher than that for a DRAM cell [5]. In order to address the high write power issue, several research works have been proposed [6][7][8].

The second factor that contributes to the high write power consumption is the current loss in the on-chip charge pumps (CPs). A CP typically consists of several cascaded stages of large capacitors. Each stage elevates the voltage by a certain amount. The current drawn from the supplier is not always delivered to the loader. It will charge parasitic capacitances and leak away as reverse currents [9]. The pumping efficiency, defined as the ratio between output and input power, is a concave function of load current. Figure 1(a) shows an example of the relationship between pumping efficiency and load current. The *X*-axis represents the load current and the *Y*-axis represents the pumping efficiency. One of the most important properties of a concave function is that $f((x+y)/2) \geq (f(x)+f(y))/2$ for any x and y, which means that if the load current is smoothed, the pumping efficiency can be greatly improved.

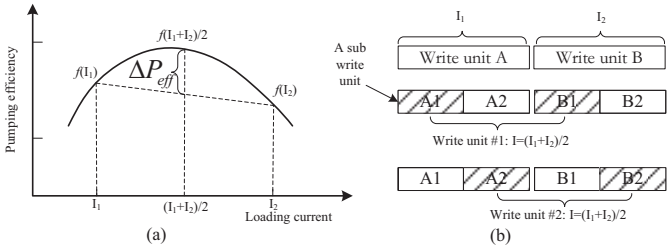

Fig. 1. Peak-to-average write scheme example: write units A and B are regrouped to new write units with the aim of write current smoothing.

As mentioned above, the current requirements of writing a '0' (RESET) and a '1' (SET) are quite different. When RESET operations are the dominant part (called peak write) in a write unit, the load current to a PCM chip is large. Otherwise, the load current is small. Take Figure 1(b) as an example. Assuming that there are two write units, A and B, to be written to the PCM main memory sequentially. '1' is the dominant bit in unit A and '0' is the dominant bit in unit B. The current needed to write A is I_1 and the current needed for B is I_2. The CP efficiency is $f(I_1)$ and $f(I_2)$ for A and B respectively, as shown in Figure 1(a). If we partition the write units into sub-write units and regroup them with the goal of smoothing the current requirements, the CP efficiency can be improved. As shown in Figure 1(b), if we regroup the sub-write unit A1 with the sub-write unit B1 into a new write unit and regroup A2 with B2 into another new write unit, the current requirements for both write units can be $\frac{I_1+I_2}{2}$. Based on the concave function property, we know that $2f(\frac{I_1+I_2}{2}) > f(I_1)+f(I_2)$. From this simple example, we can see that the pumping efficiency can be improved if the current requirements of serial write units become more uniform.

Based on this observation, this paper proposes a peak-to-average (PTA) write scheme, which smartly regroups write units on PCM chips to smooth the write current requirement in order to improve the pumping efficiency. The contributions of this paper include:

§Liang Shi is the corresponding author: shi.liang.hk@gmail.com.

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

- Presented that write current variation influences the pumping efficiency of a PCM chip;
- Proposed the PTA write scheme, which can reduce the peak write current and the write variation;
- Proposed an optimal Integer Programming (IP) formulation and an online regrouping strategy to implement the PTA scheme;
- Designed detail simulations under various workloads to show the effectiveness of the proposed PTA scheme.

The rest of this paper is organized as follows. The background of PCM and CP is presented in Section II. Section III presents the problem definition and a motivational example. Section IV presents the proposed PTA scheme and two regrouping strategies. Experimental results are given in Section V. Finally, Section VI concludes the paper.

II. Background

A. PCM Basic

A PCM cell usually consists of a layer of chalcogenide alloy (GST) material, which can switch between low resistance crystalline state (i.e., RESET, represents '0') and high-resistance amorphous state (i.e., SET, represents '1'). The RESET operation is performed by applying a large but short pulse to the GST material and converting it from crystalline to amorphous state. The SET operation is performed by applying a smaller but longer pulse for the reverse state transition. Figure 2 illustrates these two operations. From the figure, we can see that writing a '0' requires much higher current than writing a '1'.

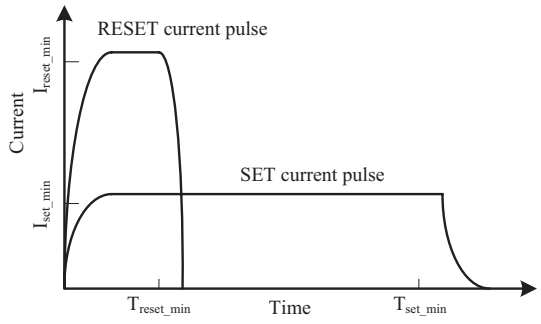

Fig. 2. Electric currents for RESET and SET operations.

B. PCM Memory Architecture

Figure 3 shows a typical PCM memory chip, which consists of 5 components: PCM array, row decoder, column decoder, column selector, and charge pump. PCM array consists of all the PCM cells. These cells are addressed through the wordlines WL_0, WL_1, ..., WL_M that are extended from a row decoder and the bitlines BL_0, BL_1, ..., BL_N that are extended from a column selector. The column selector selects proper bitlines according to the column selection signals, which are supplied by the column decoder. When receiving a write enable signal WE and input data Din, the charge pump supplies currents with appropriate value to the column selector. To write a '0', a RESET current is generated by charge pump (CP). Otherwise, a SET current is generated.

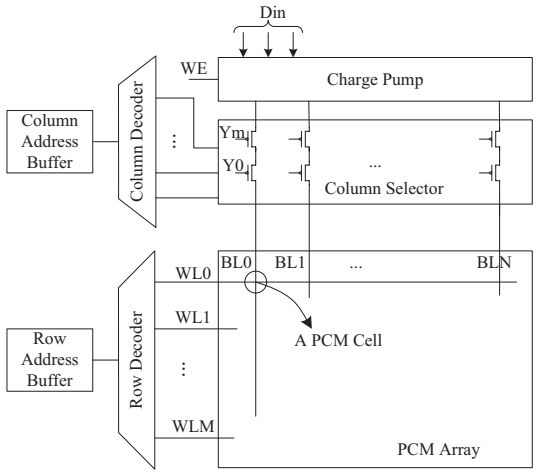

Fig. 3. The baseline memory architecture.

To access data, their addresses are first sent from the memory controller to PCM array through the address bus. Then the data are transferred through the data bus. Most PCM chips can support 64-bit or more concurrent writes. In this paper, we will use 64-bit writes for illustration purposes. Therefore, one chip is able to support 64 parallel RESETs independently. The concurrent 64-bit write is also referred to as the write unit in this paper. In a write unit, '0' and '1' are randomly distributed, the current requirements of PCM writes may fluctuate dramatically.

C. PCM Charge Pump (CP) Basics and Modeling

1) CP Basics: CP is an electronic circuit that converts the supplied voltage V_{dd} to a DC output voltage V_{out}. Figure 4 shows a typical N-stage CP. Each of the N stages can elevate the voltage by a certain amount. By adding multiple stages, the output voltage can be elevated to a target level, which is multiple times higher than V_{dd}. As shown in the figure, in addition to the N cascaded stages, there is also an output stage. Each of the cascaded stages consists of a pumping capacitor C, switch S_i, signal V_{ck}, and parasitic capacitance C_p. The output stage consists of switch S_{out}, capacitor C_L, and a current generator I_L. To supply a desired current to PCM, during the first half clock period, V_{ck} is low and all the odd switches are closed. The first pumping capacitor is charged to V_{dd} and all the other pumping capacitors in the odd stages receive the charge from the capacitors of the previous stages. During the subsequent half clock period, the signal V_{ck} is high and all the even switches are closed. Now all the capacitors in the odd stages pass the charge to the capacitors in the subsequent stages. Finally, switch S_{out} connects the output load to the final stage.

During this process, the current drawn from the supplier is not always delivered to the loader. There is parasitic power consumed on charging/discharging the internal parasitic capacitance, which does not contribute to the output [10].

2) Charge Pump Modeling: The CP modeling in [9][10] is employed in this work.

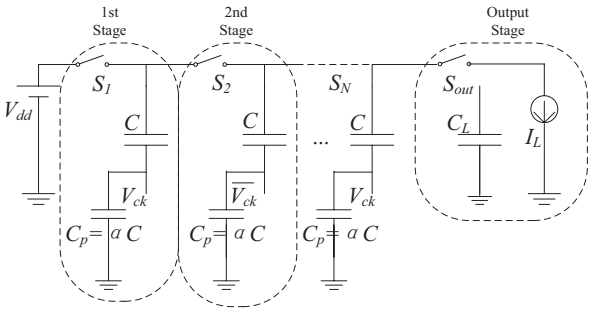

Fig. 4. N-stage CP with parasitic effects [9][10].

Total current supply. The total current consumption can be modeled as:

$$TC = [(N+1) + \alpha \cdot \frac{N^2}{(N+1) \cdot V_{dd} - V_{out}} \cdot V_{dd}] \cdot I_L \quad (1)$$

where α is the proportional factor between the parasitic capacitance and the pumping capacitance. I_L is the output current, which includes all the current that drains from the CP input. I_L mainly consists of three components: 1) the dynamic load current, i.e. read/write currents applied to PCM cells, denoted as I_{load}. This is the current for useful work; 2) the leakage of the load, denoted as I_{load_loss}; 3) the leakage of the CP itself, denoted as I_{CP_loss}. Hence,

$$I_L = I_{load} + I_{load\_loss} + I_{CP\_loss} \quad (2)$$

The pumping efficiency P_{eff} of a charge pump circuit can be expressed as:

$$P_{eff} = \frac{power\_output}{power\_input} \times 100\% = \frac{Vout \cdot I_{load}}{V_{dd} \cdot TC} \times 100\% \quad (3)$$

Substituting Equation (1) and (2) into Equation (3), it turns into

$$P_{eff} = \frac{\frac{V_{out}}{V_{dd}} \cdot I_{load}}{[(N+1) + \alpha \cdot \frac{N^2}{(N+1) \cdot V_{dd} - V_{out}} \cdot V_{dd}](I_{load} + I_{load\_loss} + I_{CP\_loss})} \quad (4)$$

To find the optimum N that minimizes the silicon area and current consumption, Jiang *et al.* [9] found that a single stage CP is preferred for both READ and SET operations and 3-stage CP is necessary for the RESET operation. The N and α becomes constants when the CPs are optimally designed.

Let

$$\beta = \frac{V_{out}}{V_{dd} \cdot [(N+1) + \alpha \cdot \frac{N^2}{(N+1) \cdot V_{dd} - V_{out}} . V_{dd}]} \quad (5)$$

Then, we will have the pumping efficiency

$$P_{eff} = \frac{\beta I_{load}}{I_{load} + I_{load\_loss} + I_{CP\_loss}} \quad (6)$$

For a cascaded stages CP system, I_{load_loss} dependents on I_{load} [10]. It is weak when I_{load} is small, and becomes significant when I_{load} is large. As shown in Equation (6), when I_{load} is small, the leakage power of the CP itself (caused by I_{CP_loss}) becomes the dominant power loss in a charge pump circuit. In this case, the power efficiency approaches zero. When I_{load} is large, the internal power loss caused by I_{load_loss} is significant. In this case, the efficiency drops drastically as a decreasing function of load current [9]. Generally, the pumping efficiency curve climbs first, then drops drastically as load current increases. The pumping efficiency is a concave function of the load current [10].

III. Problem Definition

In this section, we first present the problem definition. Then, a motivational example is presented to show the main idea of PTA.

A. Problem Definition

One of the most important properties of a concave function is as following:

$$f((x+y)/2) > (f(x) + f(y))/2 \quad (7)$$

The property can be extended to num serial write current requirements, I_1, I_2, ..., I_{num}:

$$f\left(\frac{I_1 + I_2 + ... + I_{num}}{num}\right) \geq \frac{f(I_1) + f(I_2) + ... + f(I_{num})}{num} \quad (8)$$

Equation 8 indicates that the overall pumping efficiency can be improved if the current requirements are uniform. Formally, we define variations among the current requirements as *write variation* WV:

$$WV = \frac{1}{I_{avg}} \sqrt{\frac{\sum_{i=1}^{num} (I_i - I_{avg})^2}{num - 1}} \quad (9)$$

where I_{avg} is the average write current, defined as following:

$$I_{avg} = \frac{\sum_{i=1}^{num} I_i}{num} \quad (10)$$

When serial write units are written to a PCM chip, the imbalanced write current requirements are common. Figure 5 shows the write variation, WV, distributions for a set of different benchmarks. Detailed configurations for the experiments are presented in the experiment section. As shown in Figure 5, the memory chip experiences different degrees of write variations. Overall, the write variations are high. For example, for *basicmath*, 15.5%, 10.3%, 30.9%, 12.3% and 31.0% of the write units have write variation ranging in [0, 0.5), [0.5, 1.0), [1.0, 1.5), [1.5, 2.0), and [2.0, +∞), respectively. Totally, more than 70.0% of the write units' WVs are greater than 1.0.

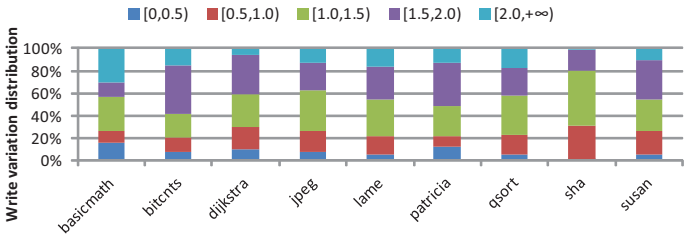

Fig. 5. The write variation distributions for different workloads.

This motivates us to propose a write scheme to reduce write variation. In this paper, we aim to smooth the currents for a serial of write units by regrouping their sub-units. When the write units are queued in the memory controller, each write unit is divided into several sub-units. The sub-units are then regrouped as new write units before they are sent to the PCM chip. The problem, how to obtain a minimal write variation during regrouping, is formally defined as following: there are num serial write units to be written to a PCM chip, where each write unit is divided into sub sub-write units. Assume that the current requirements of each sub-write unit I_{ij} are known. The problem is to regroup the write units such that the overall write variation is minimized. This problem is NP-hard, which can be proved by reducing from the Subset Sum Problem: given a set of non-negative integers, find the subset of the given set with sum equal to a given sum.

B. Motivational Example

Assuming that there are 3 serial write units written to a PCM chip. Each write unit I_i can be divided into 4 sub-write units: I_{i1}, I_{i2}, I_{i3}, and I_{i4}. The current requirements of each sub-unit are shown inside each sub-write unit in Figure 6. For the baseline write scheme in Figure 6(a), the current requirements for the 3 write units are 53, 12, and 28, respectively. The write variation is 0.6306. The pumping efficiencies for these 3 writes are 18.9%, 28.2%, and 50.2%, respectively. The overall pumping efficiency is 29.3%.

Figure 6(b) shows the result after regrouping. 3 new write units are generated as: $(I_{11}, I_{22}, I_{23}, I_{24})$, $(I_{12}, I_{21}, I_{13}, I_{34})$, and $(I_{31}, I_{14}, I_{33}, I_{32})$. In this case, the current requirements of the 3 new write units become 34, 30, and 29, respectively. The write variation is reduced to 0.0643. In addition, the pumping efficiencies become 45.1%, 48.4%, and 49.2%, respectively. The overall pumping efficiency is improved to 47.6%. Therefore, regrouping the sub-units can improve the overall pumping efficiency by 18.3%.

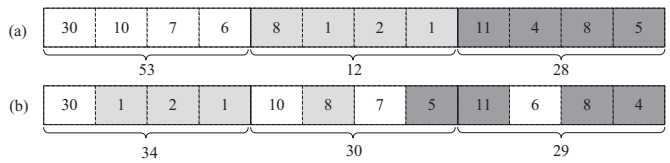

Fig. 6. Motivational example. a) the original write units; b) after regrouping.

It is noted that making an appropriate decision is important for the regrouping process. The improvement for overall pumping efficiency should be as much as possible. However, at the same time the decision-making process should be easy to implement in hardware and should incur little overhead. In the following section, we will present the proposed PTA write scheme, which can achieve great pumping efficiency improvement with little overhead.

IV. Peak to Average Write Scheme

In this section, we will present the PTA write scheme. The PTA write scheme involves renovation on both memory controller and PCM memory chips. Two regrouping strategies in the memory controller are first proposed in subsection IV-A. The modification of PCM memory chips will be described in subsection IV-B.

A. Memory Controller Regrouping Strategies

In order to implement the PTA write scheme, the memory controller first needs to calculate the current requirement and write variation of the write units within a cache line. Then, the write units are divided into equal sub-write units and memory controller regroups the sub-write units to reduce variation. Finally, memory controller sends the regrouped write units to PCM chips.

Two regrouping strategies, Integer Programming (IP) formulation and Partition Strategy (PS), are proposed. IP can generate optimal solution. However, the overhead is relatively large. PS, on the other hand, can generate near-optimal solution with little overhead.

1) IP Formulation: Let num be the total number of write units to be regrouped. Each write units are divided into sub sub-write units. Let $M = num \cdot sub$. Let $X_{i,j}$ be a binary variable, where $X_{i,j} = 1$ means sub-write unit j is grouped into the i^{th} write unit.

First, each sub-write unit j ($1 \leq j \leq M$) must be in one and only one write group. Then $X_{i,j} = 1$ should satisfy the following constraint:

$$\forall_j \sum_{i=1}^{num} X_{i,j} = 1 \tag{11}$$

Second, each group i ($1 \leq i \leq num$) has sub sub-write units. Thus, the following relationship between $X_{i,j} = 1$ and sub is required for each new group:

$$\forall_i \sum_{j=1}^{num \cdot sub} X_{i,j} = sub \tag{12}$$

The objective function is to minimize *write variation* WV:

$$\min \quad \frac{1}{I_{avg}} \sqrt{\frac{\sum_{i=1}^{num} \left(\sum_{j=1}^{M} X_{i,j} \cdot C_j - I_{avg} \right)^2}{num - 1}} \tag{13}$$

where element C_j is the write current of j^{th} sub-write unit and I_{avg} is the average write current defined in Equation (10).

Overheads The time complexity of the IP formulation is $O(2^M)$, which is exponential. It is difficult and expensive to be implemented online. The IP formulation will be used to compute the optimal solution off-line for comparison purpose.

2) Partition Strategy (PS): In the partition strategy, the sub-write units will be divided into two parts by comparing their current requirement values with a threshold Θ. If the current value of a sub-unit is larger or equal to Θ, it is treated as a high current requirement sub-unit. Otherwise, it is treated as a low requirement one. High and low sub-units are evenly distributed to the regrouped write units, such that the numbers of high (low) sub-units in each write unit are nearly equal. In this way, the write variation will be reduced. The threshold Θ is set with the average of current requirements:

$$\Theta = \frac{\sum_{i=1}^{num} \sum_{j=1}^{sub} I_{ij}}{num \cdot sub} \tag{14}$$

Algorithm 1 shows the partition strategy. It employs two indices $Left$ and $Right$, where $Left$ points to the top of high sub-units and $Right$ points to the top of low sub-units. Initially, $Left$=1 and $Right = num \cdot sub$ (lines 1-2). The procedure first calculates the threshold Θ (line 3). Then, the procedure makes a partition (lines 4-12). If the value of I_{ij} is greater or equal to Θ, it is pushed into the left part of $A[\ ]$ (lines 6-7). Otherwise, it is pushed into the right part (lines 9-10). When the partition is finished, the decision is made by allocating the high and low sub-units to write units evenly (lines 13-15). A straightforward yet effective way is to regroup $A[i]$ into the $(i\ mod\ sub)^{th}$ write unit (line 14).

Algorithm 1 Partition Strategy

Input: I_{ij} – The current requirement of each sub-write unit.
$A[\ ]$ – A vector to stack the high and low sub-write units in two directions.
Output: The regroup decision $DECN[\ ][\ ]$.
1: $Left = 1$;
2: $Right = num \cdot sub$;
3: Calculating $\theta = \frac{\sum I_{ij}}{num \cdot sub}$;
4: **for** each I_{ij} **do**
5: **if** $I_{ij} \geq \theta$ **then**
6: $A[Left] = I_{ij}$;
7: $Left = Left + 1$;
8: **else**
9: $A[Right] = I_{ij}$;
10: $Right = Right - 1$;
11: **end if**
12: **end for**
13: **for** $i \leftarrow 1$ to $num \cdot sub$ **do**
14: $DECN[(i\ mod\ sub) + 1][i/sub + 1] = A[i]$;
15: **end for**
16: **return** the regrouping decision $DECN[\ ][\ ]$.

Overheads The running time of PS is $O(M)$, which can be overlapped with the write queue scheduling. The procedure requires extra space to separate the sub-write units in $A[\ ]$, which is equal to the size of the write units. As for the energy overhead, the procedure needs to read the data from the write queues, and then write them to the extra space to make the regrouping decision. According to the experimental results, the energy overhead is negligible.

B. Memory Chip Modification

When regrouped sub-write units and their column addresses are sent from memory controller to memory chips, the PCM chips need to be able to handle the multiple column addresses of sub-write units. Xia *et al.* [11] proposed Dynamic Write Consolidation (DWC) scheme to consolidate multiple writes targeting the same row into one write. In this paper, we adopt their implementing method in memory chip. (sub - 1) column address buffers are added to receive column addresses from address bus. (sub - 1) column decoders are also added to generate column selection signals by decoding the column addresses.

Overheads. The modification to PCM chip does not incur extra performance overhead, since the added decoders work with the original decoders in parallel. The hardware overhead consists of (sub - 1) column address buffers and decoders. The energy overhead is only 0.47% [11].

V. Experiment

A. Experimental Setup

The experiments are conducted in two steps. In the first step, SimpleScalar [12] is used to collect memory access traces (such as write-back addresses, sizes, and contents) from different workloads. The system configurations used in the experiments are shown in Table I. In the second step, these memory traces are fed into a custom memory system simulator. We model power characteristics of read and write operations with a detailed PCM memory system simulator. The PCM device configuration is listed in Table II. The pumping efficiency function is presented in Equation (6). The write units are divided to 4 sub-units as default. We selected 13 benchmarks from Mibench [13] as the workloads.

TABLE I. System configurations.

Cores	1-cores, 2GHz
L1 I/D cache	2MB, 1-way, 64 Byte/line, 1-cycle hit LRU replacement policy
L2 data cache (LLC)	8MB, 4-way, 64 Byte/line, 6-cycle hit write back, LRU replacement policy
Memory controller	64-entry RDQ and WRQ, MC to bank 64-cycle, scheduling reads first, FCFS, read priority scheduling issuing writes when there is NO read
Main memory	SLC PCM (1GB size, 8 banks 32768 rows/bank, 1024 columns/row memory access bus width: 64 bits)

TABLE II. PCM chip configurations.

Chip	1.8V V_{dd}, 64 concurrent RESET power buget
Charge pump	working frequent: 133MHZ RESET/SET/READ working voltages: 5/3/3V
READ	3V, 8.4 μA, 5.6nJ per line
Write	RESET: 5V, 100 μA, 29.7pJ per bit, 50ns operation latency SET: 3V, 50 μA, 22.5pJ per bit, 150ns operation latency
PCM write unit	size: 64bit, divided into 4 sub-write units

B. Investigated Strategy and Evaluation Metrics

The baseline strategy does not have any optimization. The proposed IP and PS are implemented in the simulator. To show the effectiveness of the proposed techniques, we evaluate write variation, pumping efficiency, and energy in the experiments.

1) Write Variation Reduction: Figure 7 presents the comparisons of write variation among different approaches. As shown in the figure, the results show that the PTA approaches achieve great variation reductions. The write variation is reduced from 1.28 to 0.52 and 0.32 for PS and IP, respectively. It is shown that several benchmarks have significant differences on variation reductions. The reason is that the content of the write also has a large influence on the benefit achieved by the regrouping strategies. The more imbalanced the distribution of '0' and '1' is, the higher the benefit can be achieved. It is also shown that IP works better than PS. However, even though the optimal IP achieves 0.2 more reduction than PS, it comes with large implementation overhead compared with PS.

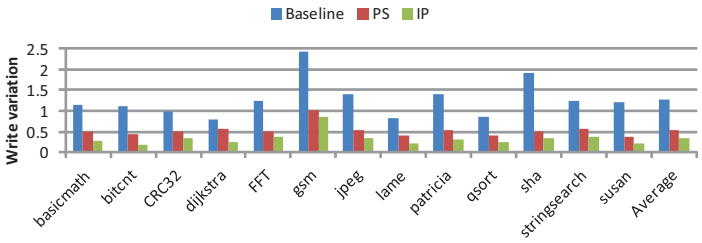

Fig. 7. Improvement of write variations.

2) Pumping Efficiency Improvement: Figure 8 presents the pumping efficiencies achieved by the investigated approaches. As shown in Figure 8, the baseline has a low pumping efficiency (29.8% on average). The proposed approaches achieve significant improvement on the pumping efficiencies. PS achieves 38.8% on average while IP achieves 39.8% pumping efficiencies. This shows that PTA write scheme is very efficient in improving the pumping efficiency.

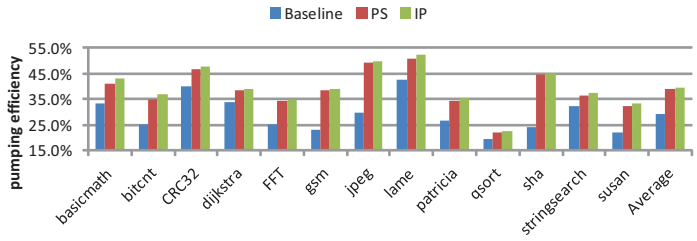

Fig. 8. Improvement of pumping efficiencies under various benchmarks.

3) Energy Saving: The energy saving of PS is also evaluated in the experiments. The energy overhead consists of memory controller overhead and PCM chip overhead. The PCM chip energy overhead is presented in [11]. The memory controller energy overhead consists of Read, Write and ALU operations during the regrouping. We keep track of the memory controller energy overhead in our simulator.

Figure 9 presents energy improvement achieved by the investigated approaches under different benchmarks. As shown in the figure, PS achieved 17.0% energy saving compared to the baseline. Especially, PS improves the energy consumption of benchmark *sha* up to 39.8%. The energy improvement of PTA mainly results from the pumping efficiency improvement. Moreover, different benchmarks have little impact on the energy overhead incurred by PS. The reason is that the energy consumption of PS procedure is not related to the content of input data.

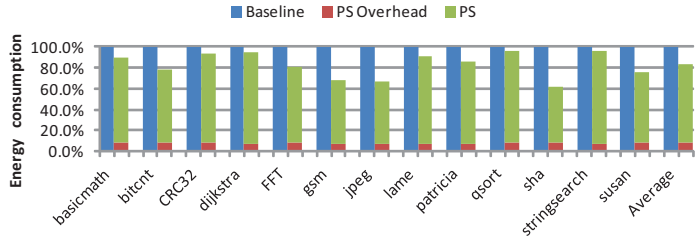

Fig. 9. Energy saving of memory system.

VI. Conclusion

This paper proposes a write scheme, peak-to-average (PTA), to improve the pumping efficiencies of PCM by regrouping the sub-write units. An off-line optimal Integer Programming (IP) formulation and an online technique, Partition Strategy (PS), are proposed to make the regrouping decision. Experiment results show that PTA achieves great improvements compared with baseline technique. The experimental results show that the pumping efficiency is improved from 29.8% to 38.8% by the proposed PS strategy. As for the energy overhead, PTA can achieve 17.0% energy reduction on average.

VII. Acknowledgment

This work is partially supported by the Fundamental Research Funds for the Central Universities (106112014CDJZR185502), NSFC 61402059, NSFC 61472052, NSFC 61173014, National 863 Program 2013AA013202, 2015AA015304, Chongqing High-Tech Research Program cstc2012ggC40005, cstc2014yykfB40007.

References

[1] Y. Choi, I. Song, M.-H. Park, H. Chung, S. Chang, B. Cho, J. Kim, Y. Oh, D. Kwon, J. Sunwoo *et al.*, "A 20nm 1.8 v 8gb pram with 40mb/s program bandwidth," in *ISSCC' 12*, pp. 46–48.

[2] M. Joshi, W. Zhang, and T. Li, "Mercury: A fast and energy-efficient multi-level cell based phase change memory system," in *HPCA' 11*, pp. 345–356.

[3] P. J. Nair, C. Chou, B. Rajendran, and M. K. Qureshi, "Reducing read latency of phase change memory via early read and turbo read," in *HPCA' 15*, pp. 309–319.

[4] M. Zhao, Y. Xue, C. Yang, and C. J. Xue, "Minimizing mlc pcm write energy for free through profiling-based state remapping," in *ASP-DAC' 15*, pp. 502–507.

[5] K.-J. Lee, B.-H. Cho, W.-Y. Cho, S. Kang, B.-G. Choi, H.-R. Oh, C.-S. Lee, H.-J. Kim, J.-M. Park, Q. Wang *et al.*, "A 90 nm 1.8 v 512 mb diode-switch pram with 266 mb/s read throughput," *Journal of Solid-State Circuits*, vol. 43, no. 1, pp. 150–162, 2008.

[6] 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, p. 77, 2013.

[7] B.-D. Yang, J.-E. Lee, J.-S. Kim, J. Cho, S.-Y. Lee, and B.-G. Yu, "A low power phase-change random access memory using a data-comparison write scheme," in *ISCA' 07*, pp. 3014–3017.

[8] S. Cho and H. Lee, "Flip-n-write: a simple deterministic technique to improve pram write performance, energy and endurance," in *MICRO' 09*, pp. 347–357.

[9] L. Jiang, B. Zhao, J. Yang, and Y. Zhang, "A low power and reliable charge pump design for phase change memories," in *ISCA' 14*, pp. 397–408.

[10] G. Palumbo and D. Pappalardo, "Charge pump circuits: An overview on design strategies and topologies," *Circuits and Systems Magazine*, vol. 10, no. 1, pp. 31–45, 2010.

[11] F. Xia, D. Jiang, J. Xiong, M. Chen, L. Zhang, and N. Sun, "Dwc: dynamic write consolidation for phase change memory systems," in *ICS' 14*, pp. 211–220.

[12] D. Burger and T. M. Austin, "The simplescalar tool set, version 2.0," *ACM SIGARCH Computer Architecture News*, vol. 25, no. 3, pp. 13–25, 1997.

[13] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "Mibench: A free, commercially representative embedded benchmark suite," in *WWC-4' 01*, pp. 3–14.