FPB: Fine-grained Power Budgeting to Improve Write Throughput of Multi-level Cell Phase Change Memory

Lei Jiang, Youtao Zhang, Bruce R. Childers and Jun Yang
University of Pittsburgh

Introduction

As a promising nonvolatile memory technology, Phase Change Memory (PCM) has many advantages over traditional DRAM. Multi-level Cell PCM (MLC) has the benefit of increased memory capacity with low fabrication cost. Due to high write power and long write latency, MLC PCM requires careful power management to ensure write reliability. However, the existing power management schemes applied to MLC PCM result in low write throughput and large performance degradation.

In this paper, we propose Fine-grained write Power Budgeting (FPB) for MLC PCM. We first identify two major problems for MLC write operations: (i)managing write power without consideration of the iterative write process used by MLC is overly pessimistic; (ii) a heavily written (hot) chip may block the memory from accepting further writes due to chip power restrictions, although most chips may be available. To address these problems, we propose two FPB schemes. First, FPB-IPM observes a global power budget and regulates power across write iterations according to the step-down power demand of each iteration. Second, FPB-GCP integrates a global charge pump on a DIMM to boost power for hot PCM chips while staying within the global power budget. Our experimental results show that these techniques achieve significant improvement on write throughput and system performance.

DIMM and Chip Level Power Budgets

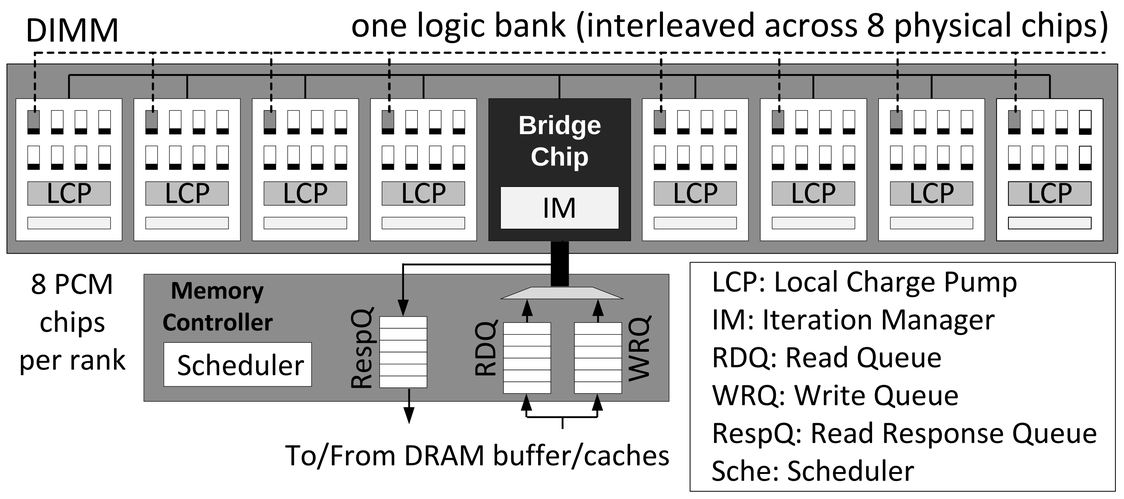


Figure 1: The baseline architecture of a MLC PCM-based memory subsystem (One DIMM).

DIMM level power budget: PCM requires much higher per-cell write power than DRAM. Hay *et al.* calculated that the power provided by a typical DDR3-1066×16 DRAM memory allows up to **560** SLC PCM simultaneous cell writes.

Chip level power budget: Another power restriction is the chip-level power budget. Since PCM writes require higher voltages than V_{dd}, PCM chips integrate CMOS-compatible **charge pumps** to supply required voltage and power. Studies have shown that the area of a charge pump is proportional to the maximum current that it can provide.

$$A_{tot} = k \cdot \frac{N^2}{(N+1)\cdot V_{dd} - V_{out}} \cdot \frac{I_L}{f}$$

Charge pump area — **Total write current**

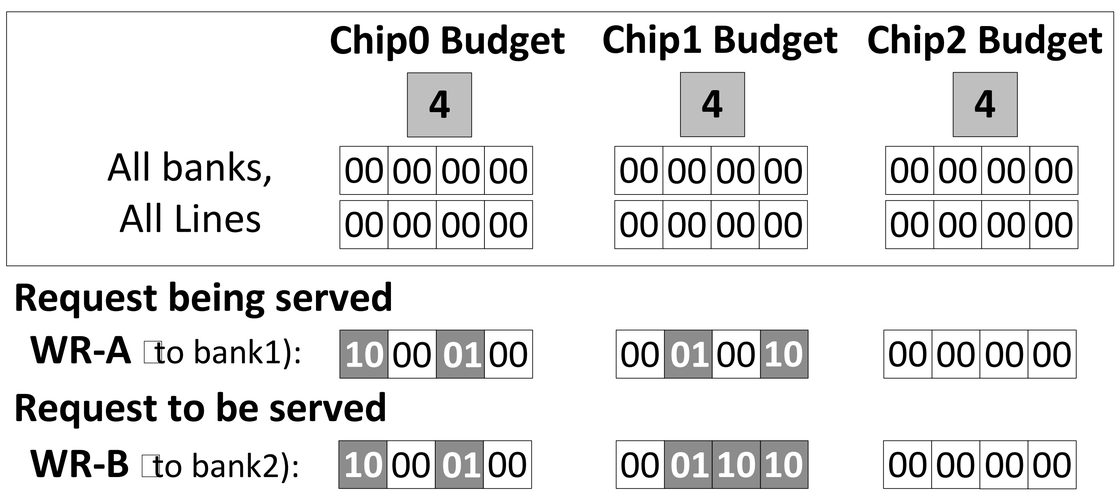


Figure 2: Writes blocked by chip level power budget.

The write throughput of MLC PCM may be constrained by a chip power budget. In Figure 2, we assume (i) one bank spreads across three chips; (ii) the memory initially contains all 0s; (iii) the chip power budget can support 4 cell changes; (iv) the system is serving request ***WR-A*** when request ***WR-B*** arrives. They write to different banks and change 4 and 5 cells respectively (shown as shaded boxes with white font).

While these two writes change 9 cells in total and the DIMM power budget allows 12 cell changes, WR-B cannot be issued as the sum of cell changes for chip 1 is 5, which is larger than the chip's budget. If WR-B is issued, both writes may fail as there is not enough power for reliable programming.

A typical charge pump occupies 15% to 20% of a PCM chip's area. Thus, it is undesirable to enlarge the charge pump to increase its maximum output current/power.

Iteration Power Management (IPM)

Figure 3 illustrates how iteration power management works. The scheme is token driven in order for writes to proceed. And there must be enough power tokens available to satisfy the number of bit changes required by a write. Each token represents the power for a single cell RESET.

Figure 3 (a) shows a simple per write power allocation heuristic, This heuristic tracks **A**vailable **P**ower **T**oken (APT), and then releases a write only when there are enough available **PT**s. **WR-A** is served first. APT is reduced to 30 until WR-A finishes. In this case, **WR-B** stalls until **WR-A** returns its tokens.

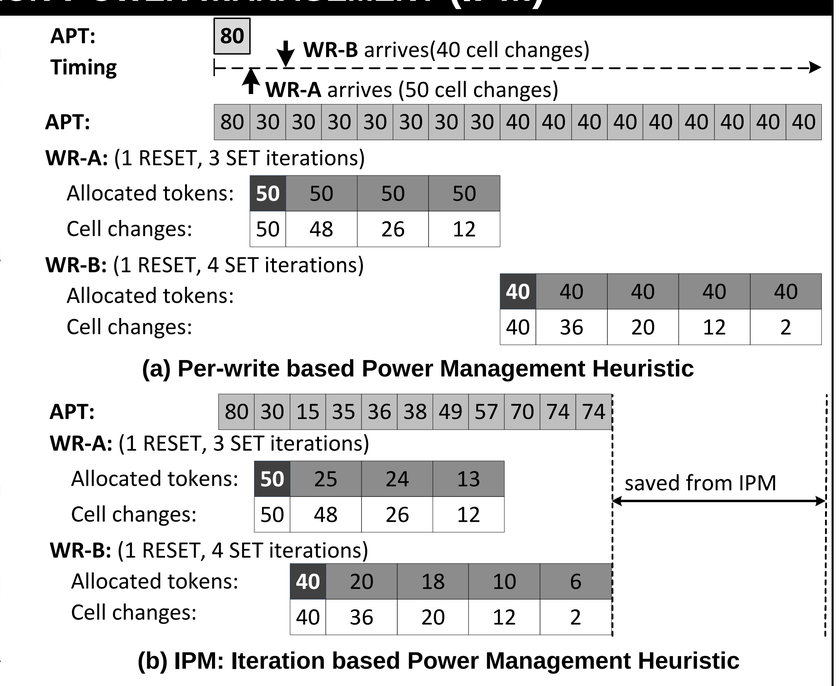


Figure 3: The FPB-IPM: iteration power management (assuming SET is ½ of RESET power and RESET is ½ the length of SET pulse).

To resolve this problem, we designed FPB-IPM to reclaim unused power tokens as early as possible, which increases the number of simultaneous writes. Figure 3 (b) illustrates our improved scheme. Next, after the first RESET iteration, FPB-IPM reclaims $((C-1)/C) \times PT_{RESET}$ tokens, where RESETpower = C×SETpower and PT_{RESET} is the number of tokens allocated in the first iteration. For example, half of the allocated tokens are reclaimed in write iteration 2, as shown in Figure 3 (b). Because a MLC write operation finishes in a non-deterministic number of iterations, the number of cells that need to be written decreases after each SET iteration. The consumed write power also drops as the write operation proceeds. Thus, FPB-IPM also reclaims tokens after SET iterations. To reclaim unused tokens as early as possible, FPB-IPM dynamically adjusts the power token allocation on each iteration.

Global Charge Pump (GCP)

As shown in Figure 4, the Global Charge Pump resides in the bridge chip and uses a dedicated wire to supply the pumped voltage and write current to each PCM chip. Each bank segment (within a PCM chip) has an analog power/ current controller to select write power and voltage from either **LCP** or **GCP** (but not both)..

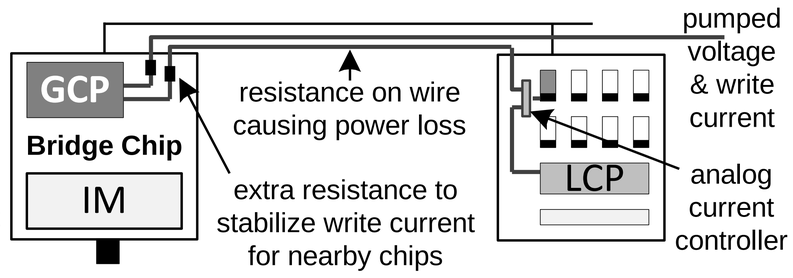
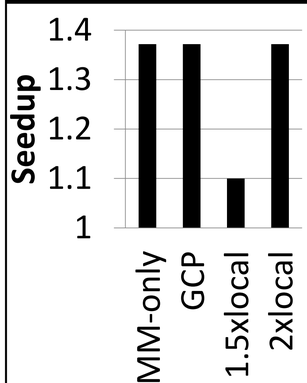


Figure 4: Integrating a global charge pump (GCP)

By default, the maximum power that the GCP can provide is set to the same power as one LCP. The power that the GCP provides to one chip is actually "borrowed" from other chips.

Results



With GCP, system performance nearly achieves the *DIMM-only* with only chip power constraint case. 2 x local pump size also can achieve the same result on performance. But, more 50% pump size does not help at all.

scheme	overhead
2xLocal	100%
1.5xLocal	50%
GCP	12.5%

Baseline has 560 PTs size charge pump. 2xLocal scheme also doubles the pump size overhead. 1.5 x Local has extra 50% pump size overhead. Our GCP only adds extra **12.5%** pump size overhead to baseline.

Lei Jiang University of Pittsburgh lej16@pitt.edu