DOI:10.1145/1785414.1785441

Phase Change Memory Architecture and the Quest for Scalability

By Benjamin C. Lee, Engin Ipek, Onur Mutlu, and Doug Burger

Abstract

Memory scaling is in jeopardy as charge storage and sensing mechanisms become less reliable for prevalent memory technologies, such as dynamic random access memory (DRAM). In contrast, phase change memory (PCM) relies on programmable resistances, as well as scalable current and thermal mechanisms. To deploy PCM as a DRAM alternative and to exploit its scalability, PCM must be architected to address relatively long latencies, high energy writes, and finite endurance.

We propose architectural enhancements that address these limitations and make PCM competitive with DRAM. A baseline PCM system is 1.6× slower and requires 2.2× more energy than a DRAM system. Buffer reorganizations reduce this delay and energy gap to 1.2× and 1.0×, using narrow rows to mitigate write energy as well as multiple rows to improve locality and write coalescing. Partial writes mitigate limited memory endurance to provide more than 10 years of lifetime. Process scaling will further reduce PCM energy costs and improve endurance.

1. INTRODUCTION

Memory technology scaling drives increasing density, increasing capacity, and falling price-capability ratios. Memory scaling, a first-order technology objective, is in jeopardy for conventional technologies. Storage mechanisms in prevalent memory technologies require inherently unscalable charge placement and control. In the nonvolatile space, Flash memories must precisely control the discrete charge placed on a floating gate. In volatile main memory, DRAM must not only place charge in a storage capacitor but must also mitigate subthreshold charge leakage through the access device. Given these challenges, solutions for scaling DRAM beyond 40nm are unknown.[17]

PCM provides a nonvolatile storage mechanism amenable to process scaling. During writes, an access transistor injects current into the storage material and thermally induces phase change, which is detected as a programmed resistance during reads. PCM, relying on analog current and thermal effects, does not require control over discrete electrons. As technologies scale and heating contact areas shrink, programming current scales linearly. This PCM scaling mechanism has been demonstrated in a 32 nm device prototype.[15] As a scalable DRAM alternative, PCM could provide a clear roadmap for increasing main memory density and capacity.

These scalability trends motivate a transition from charge memories to resistive memories. To realize this transition for PCM, we must overcome PCM disadvantages relative to DRAM. Access latencies, although tens of nanoseconds, are several times slower than those of DRAM. At present technology nodes, PCM writes require energy intensive current injection. Moreover, writes induce thermal expansion and contraction within the storage element, degrading injection contacts and limiting endurance to hundreds of millions of writes per cell at current processes. These limitations are significant, which is why PCM is currently positioned only as a Flash replacement; in this market, PCM properties are drastic improvements. For a DRAM alternative, however, we must architect PCM for feasibility in main memory within general-purpose systems.

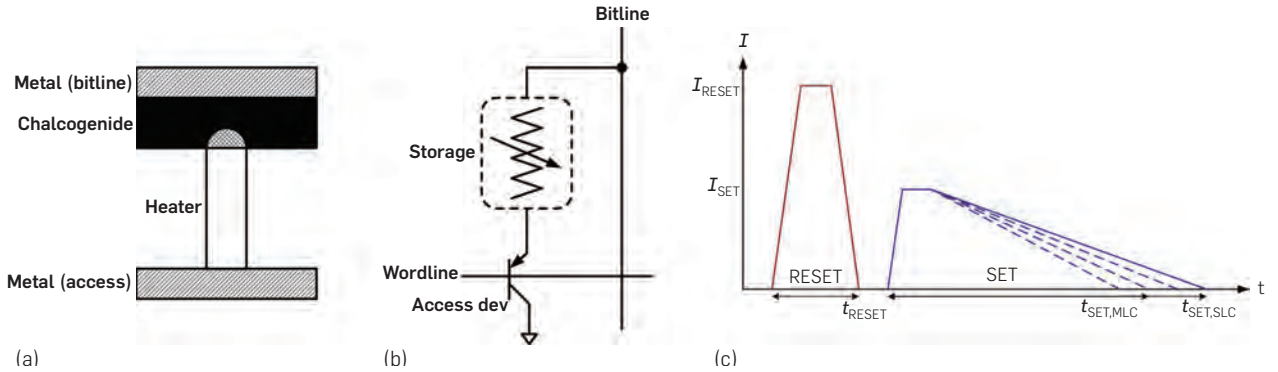
Current prototype designs are not designed to mitigate PCM latencies, energy costs, and finite endurance. This paper rethinks PCM subsystem architecture to bring the technology within competitive range of DRAM. Drawn from a rigorous survey of PCM device and circuit prototypes published within the last 5 years and comparing against modern DRAM memory subsystems, we propose:

- **Buffer Reorganization**: Narrow buffers mitigate high energy PCM writes. Multiple buffer rows exploit locality to coalesce writes, hiding their latency and reducing their energy. Effective PCM buffering reduces application execution time from 1.6× to 1.2× and memory array energy from 2.2× to 1.0×, relative to DRAM-based systems.
- **Partial Writes**: Partial writes track data modifications and write only modified cache lines or words to the PCM array. We expect write coalescing and partial writes to deliver an average memory module lifetime of 11.2 years. PCM endurance is expected to improve by four orders of magnitude when scaled to 32nm.[17]

Collectively, these results suggest PCM is a viable DRAM alternative, with architectural solutions providing competitive performance, comparable energy, and feasible lifetimes.

A previous version of this article appears in *Proceedings of the 36th International Symposium on Computer Architecture* (June 2009). Parts of this article appear in *IEEE Micro Top Picks from the Computer Architecture Conferences of 2009* (January/February 2010).

Figure 1. Phase change memory. (a) Storage element with heating resistor and chalcogenide between electrodes. (b) Cell structure with storage element and BJT access device. (c) Reset to an amorphous, high resistance state with a high, short current pulse. Set to a crystalline, low resistance state with moderate, long current pulse. Slope of set current ramp down determines the state in MLC.



2. PCM TECHNOLOGY

Given the still speculative state of PCM technology, researchers have made several different manufacturing and design decisions. We survey device and circuit prototypes published within the last 5 years.[10]

2.1. Memory cell

As shown in Figure 1a, the PCM storage element is comprised of two metal electrodes separated by a resistive heater and a chalcogenide, the phase change material. $Ge_2Sb_2Te_5$ (GST) is most commonly used, but other chalcogenides may offer higher resistivity and improve the device's electrical characteristics. Nitrogen doping increases resistivity and lowers programming current while GS may offer faster phase changes.[4,8]

As shown in Figure 1b, PCM cells are 1T/1R devices, comprised of the resistive storage element and an access transistor. Access is typically controlled by one of three devices: field-effect transistor (FET), bipolar junction transistor (BJT), or diode. In future, FET scaling and large voltage drops across the cell may adversely affect reliability for unselected wordlines.[14] BJTs are faster and expected to scale more robustly without this vulnerability.[3,14] Diodes occupy smaller areas and potentially enable greater cell densities, but require higher operating voltages.[11]

Phase changes are induced by injecting current into the resistor junction and heating the chalcogenide. Current and voltage characteristics of the chalcogenide are identical regardless of its initial phase, which lowers programming complexity and latency.[9] The amplitude and width of the injected current pulse determine the programmed state as shown in Figure 1c.

2.2. Operation

The access transistor injects current into the storage material and thermally induces phase change, which is detected as a programmed resistance during reads. Logical data values are captured by the resistivity of the chalcogenide. A high, short current pulse increases resistivity by abruptly discontinuing current, quickly quenching heat generation, and freezing the chalcogenide into an amorphous state (i.e., reset). A moderate, long current pulse reduces resistivity by ramping down current, gradually cooling the chalcogenide, and inducing crystal growth (i.e., set). Requiring longer current pulses, set latency determines write performance. Requiring higher current pulses, reset energy determines write power.

Prior to reading the cell, the bitline is precharged to the read voltage. If a selected cell is in a crystalline state, the bitline is discharged with current flowing through the storage element and access transistor. Otherwise, the cell is in an amorphous state, preventing or limiting bitline current.

Cells that store multiple resistance levels might be implemented by leveraging intermediate states, in which the chalcogenide is partially crystalline and partially amorphous.[3,13] Smaller current slopes (i.e., slow ramp down) produce lower resistances and larger slopes (i.e., fast ramp down) produce higher resistances. Varying slopes induce partial phase transitions changing the size or shape of the amorphous material produced at the contact area, giving rise to resistances between those observed from the fully amorphous or the fully crystalline chalcogenide. The difficulty and high latency of differentiating between a large number of resistances may constrain such multilevel cells (MLC) to a small number of bits per cell.

Wear and Endurance: Writes are the primary wear mechanism in PCM. When injecting current into a volume of phase change material, thermal expansion and contraction degrades the electrode-storage contact, such that programming currents are no longer reliably injected into the cell. Since material resistivity is highly dependent on current injection, current variability causes resistance variability. This greater variability degrades the read window, the difference between programmed minimum and maximum resistance.

Write endurance, the number of writes performed before the cell cannot be programmed reliably, ranges from 1E+04 to 1E+09. Write endurance depends on process and differs across manufacturers. Relative to Flash, PCM is likely to exhibit greater write endurance by at least two to three orders of magnitude; Flash cells can sustain only 1E+05 writes. The ITRS roadmap projects improved endurance of 1E+12 writes at 32nm.[17] With wear reduction and leveling techniques, PCM write limits may not be exposed to the system during a memory's lifetime.

2.3. Process scaling

PCM scaling reduces required programming current injected via the electrode-storage contact. As the contact area decreases with feature size, thermal resistivity increases and the volume of phase change material that must be cooled into an amorphous state during a reset to completely block current flow decreases. These effects enable smaller access devices for current injection. Pirovano et al. outline PCM scaling rules,[14] which are confirmed empirically in a survey by Lai.[9] Specifically, as feature size scales linearly ($1/k$), contact area decreases quadratically ($1/k^2$). Reduced contact area causes resistivity to increase linearly (k), which causes programming current to decrease linearly ($1/k$).

Operational issues arise with aggressive PCM technology scaling. As contact area decreases, lateral thermal coupling may cause programming currents for one cell to influence the states of adjacent cells. Lai's survey of PCM finds these effects negligible in measurement and simulation.[9] Temperatures fall exponentially with distance from programmed cell, suggesting no appreciable impact from thermal coupling. Increasing resistivity from smaller contact areas may reduce signal strength (i.e., smaller resistivity difference between crystalline and amorphous states). However, these signal strengths are well within the sense circuit capabilities of modern memory architectures.[9]

2.4. Array architecture

As shown in Figure 2, PCM cells might be hierarchically organized into banks, blocks, and subblocks. Despite similarities to conventional memory architectures, PCM-specific issues must be addressed. For example, PCM reads are nondestructive whereas DRAM reads are destructive and require mechanisms to replenish discharged capacitors.

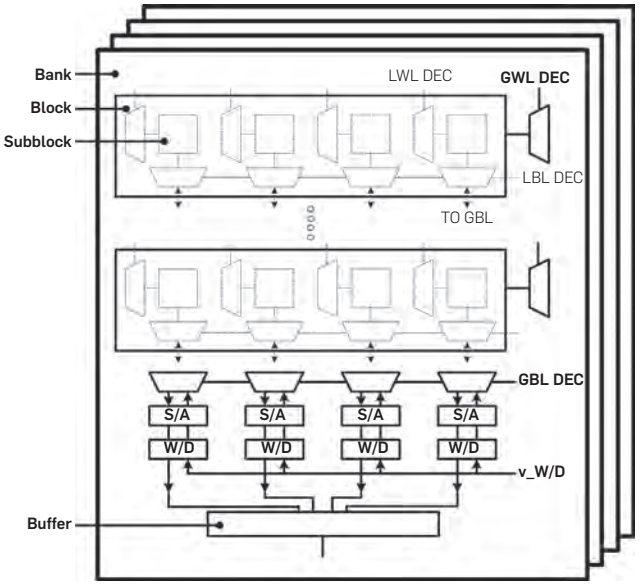
Sense amplifiers detect the change in bitline state when a memory row is accessed. Choice of bitline sense amplifiers affects array read access time. Voltage sense amplifiers are cross-coupled inverters which require differential discharging of bitline capacitances. In contrast, current sense amplifiers rely on current differences to create a differential voltage at the amplifier's output nodes. Current sensing is faster but requires larger circuits.[18]

In DRAM, sense amplifiers serve a dual purpose, both sensing and buffering data using cross-coupled inverters. In contrast, we explore PCM architectures with separate sensing and buffering; sense amplifiers drive banks of explicit latches. These latches provide greater flexibility in row buffer organization by enabling multiple buffered rows. However, these latches incur area overheads. Separate sensing and buffering enables multiplexed sense amplifiers. Multiplexing also enables buffer widths narrower than the array width, which is defined by the total number of bitlines. Buffer width is a critical design parameter, determining the required number of expensive current sense amplifiers.

3. A DRAM ALTERNATIVE

We express PCM device and circuit characteristics within conventional DDR timing and energy parameters, thereby quantifying PCM in the context of more familiar DRAM parameters to facilitate a direct comparison.[10]

Figure 2. Array architecture. A hierarchical memory organization includes banks, blocks, and subblocks with local, global decoding for row, column addresses. Sense amplifiers (S/A) and word drivers (W/D) are multiplexed across blocks.



We evaluate a four-core chip multiprocessor using the SESC simulator.[16] The 4-way superscalar, out-of-order cores operate at 4.0GHz. This datapath is supported by 32KB, direct-mapped instruction and 32KB, 4-way data L1 caches, which may be accessed in two to three cycles. A 4MB, 8-way L2 cache with 64B lines is shared between the four cores and may be accessed in 32 cycles.

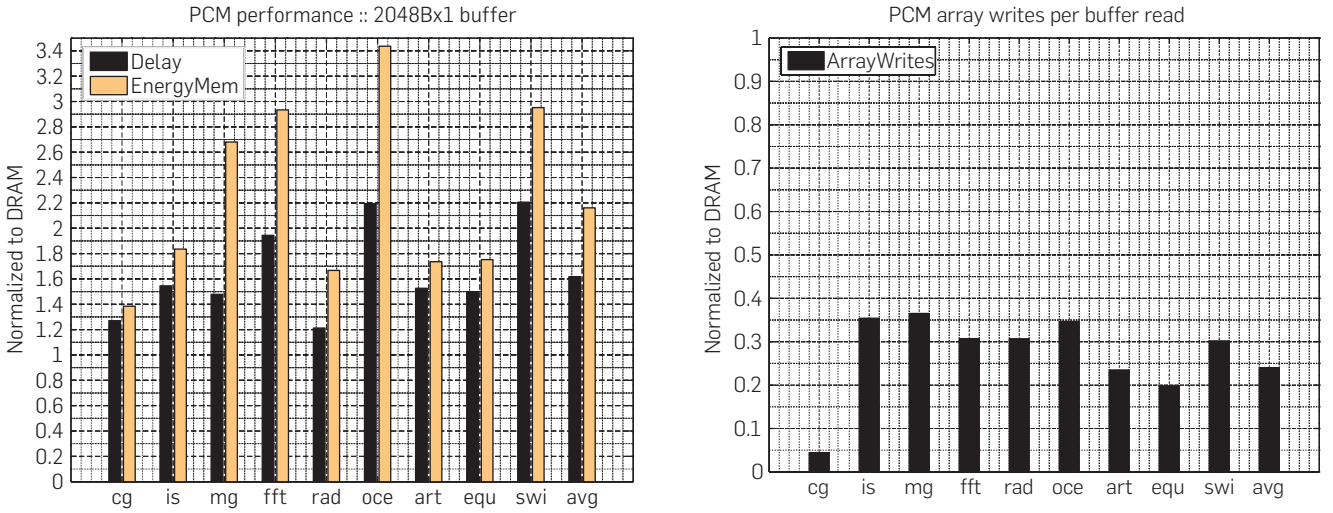
Below the caches is a 400 MHz SDRAM memory subsystem modeled after Micron's DDR2-800 technical specifications.[12] We consider one channel, one rank, and four ×16 chips per rank to achieve the standard 8B interface. Internally, each chip is organized into four banks to facilitate throughput as data are interleaved across banks and accessed in parallel. We model a burst length of eight blocks. The memory controller has a 64-entry transaction queue.

We consider parallel workloads from the SPLASH-2 suite (fft, radix, ocean), SPEC OpenMP suite (art, equake, swim), and NAS parallel benchmarks (cg, is, mg).[1, 2, 19] Regarding input sets, we use 1 M points for FFT, 514×514 grid for ocean, and 2 M integers for radix. SPEC OpenMP workloads run MinneSpec-Large data set and NAS parallel benchmarks run with Class A problem sizes. Applications are compiled using gcc and Fortran compilers at the -O3 optimization level.

3.1. Baseline comparison

We consider a PCM baseline architecture, which implements DRAM-style buffering with a single 2048B-wide buffer. Figure 3a illustrates end-to-end application performance when PCM replaces DRAM as main memory. Application delay increases with penalties relative to DRAM ranging from 1.2× (radix) to 2.2× (ocean, swim). On average, we observe a 1.6× delay penalty. The energy penalties are larger, ranging from 1.4× (cg) to 3.4× (ocean), due to the highly expensive array writes required when buffer contents

Figure 3. PCM as a DRAM alternative. (a) Application delay and memory energy. (b) Percentage of buffer evictions that require array writes.



are evicted. On average, we observe a 2.2× energy penalty.

The end-to-end delay and energy penalties are more modest than the underlying technology parameters might suggest. Even memory-intensive workloads mix computation with memory accesses. Furthermore, the long latency, high energy array writes manifest themselves much less often in PCM than in DRAM; nondestructive PCM reads do not require subsequent writes whereas destructive DRAM reads do. Figure 3b indicates only 28% of PCM array reads first require an array write of a dirty buffer.

To enable PCM for use below the lowest level processor cache in general-purpose systems, we must close the delay and energy gap between PCM and DRAM. Nondestructive PCM reads help mitigate underlying delay and energy disadvantages by default. We seek to eliminate the remaining PCM-DRAM differences with architectural solutions. In particular, the baseline analysis considers a single 2048B-wide buffer per bank. Such wide buffering is inexpensive in DRAM, but incurs unnecessary energy costs in PCM given the expensive current injection required when writing buffer contents back into the array.

3.2. Buffer organization

We examine whether PCM subsystems can close the gap with DRAM application performance and memory subsystem energy. To be a viable DRAM alternative, buffer organizations must hide long PCM latencies, while minimizing PCM energy costs.

To achieve area neutrality across buffer organizations, we consider narrower buffers and additional buffer rows. The number of sense amplifiers decreases linearly with buffer width, significantly reducing area as fewer of these large circuits are required. We utilize this area by implementing multiple rows with latches much smaller than the removed sense amplifiers. Narrow widths reduce PCM write energy but negatively impact spatial locality, opportunities for write coalescing, and application performance. However, these penalties may be mitigated by the additional buffer rows.

We consider buffer widths ranging from the original 2048B to 64B, which is the line size of the lowest level cache. We consider buffer rows ranging from the original single row to a maximum of 32 rows. At present, we consider a fully associative buffer and full associativity likely becomes intractable beyond 32 rows. Buffers with multiple rows use a least recently used (LRU) eviction policy implemented in the memory controller.

3.3. Buffer design space

Buffer reorganizations impact the degree of exploited locality and energy costs associated with array reads and writes. Figure 4 illustrates the delay and energy characteristics of the buffer design space for an average of memory-intensive benchmarks. Triangles illustrate PCM and DRAM baselines, which implement a single 2048B buffer. Circles illustrate various buffer organizations. Reorganizing a single, wide

Figure 4. Pareto analysis for PCM buffer organizations.

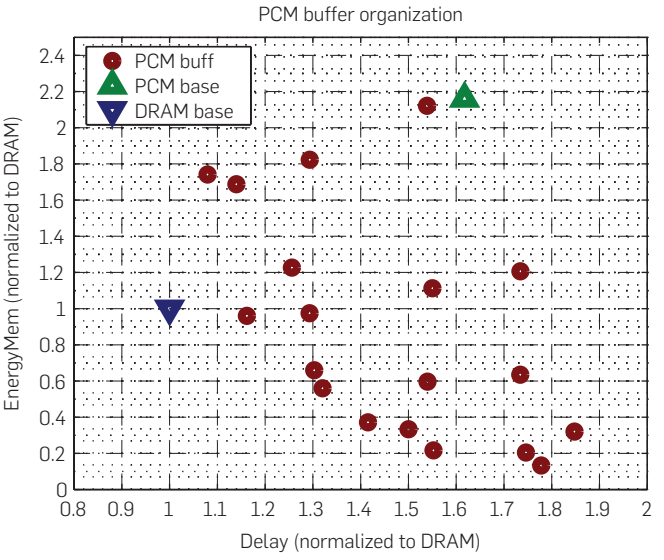
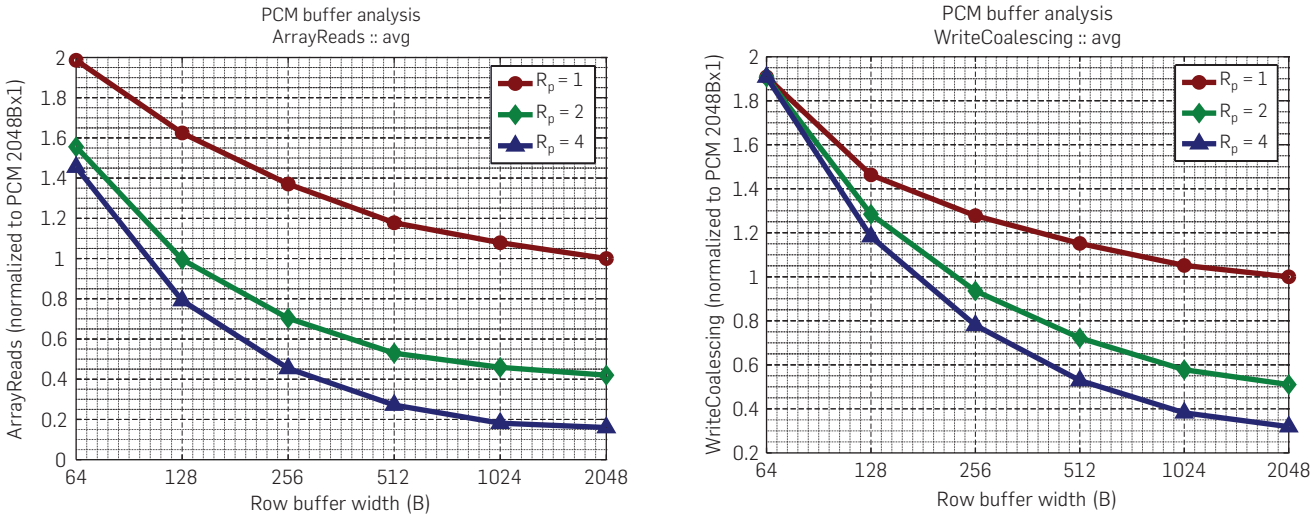


Figure 5. Memory subsystem trends from PCM buffer organizations. (a) Array reads increase sublinearly with buffer width. (b) Array write coalescing opportunities increase with buffer rows.



buffer into multiple, narrow buffers reduce both energy costs and delay. Examining the Pareto frontier, we observe Pareto optima shift PCM delay and energy into the neighborhood of the DRAM baseline. Among these Pareto optima, we observe a knee that minimizes both energy and delay; this organization uses four 512B-wide buffers to reduce PCM delay, energy disadvantages from 1.6×, 2.2× to more modest 1.2×, 1.0×.

The number of array reads is a measure of locality. Figure 5a shows the number of array reads increases very slowly as buffer width decreases exponentially from 2048B to 64B. For a single buffered row (R_p = 1), a 32× reduction in buffer width produces only a 2× increase in array reads, suggesting very little spatial locality within wide rows for the memory-intensive workloads we consider. The single row is evicted too quickly after its first access, limiting opportunities for spatial reuse. However, we do observe significant temporal locality. A 2048B-wide buffer with two rows (R_p = 2) requires 0.4× the array reads as a 2048B-wide buffer with only a single row (R_p = 1).

Writes are coalesced if multiple writes modify the buffer before its contents are evicted to the array. Thus the number of array writes per buffer write is a metric for write coalescing. Figure 5b illustrates increasing opportunities for coalescing as the number of rows increase. As the number of rows in a 2048B-wide buffer increases from one to two and four rows, array writes per buffer write fall by 0.51× and 0.32×, respectively; the buffers coalesce 49% and 68% of memory writes. Coalescing opportunities fall as buffer widths narrow beyond 256B. Since we use 64B lines in the lowest level cache, there are no coalescing opportunities from spatial locality within a 64B row buffered for a write. Increasing the number of 64B rows has no impact since additional rows exploit temporal locality, but any temporal locality in writes are already exploited by coalescing in the 64B lines of the lowest level cache.

Thus, narrow buffers mitigate high energy PCM writes and multiple rows exploit locality. This locality not only improves performance, but also reduces energy by exposing additional opportunities for write coalescing. We evaluate PCM buffering using technology parameters at 90nm. As PCM technology matures, baseline PCM latencies may improve. Moreover, process technology scaling will drive linear reductions in PCM energy.

3.4. Scaling comparison

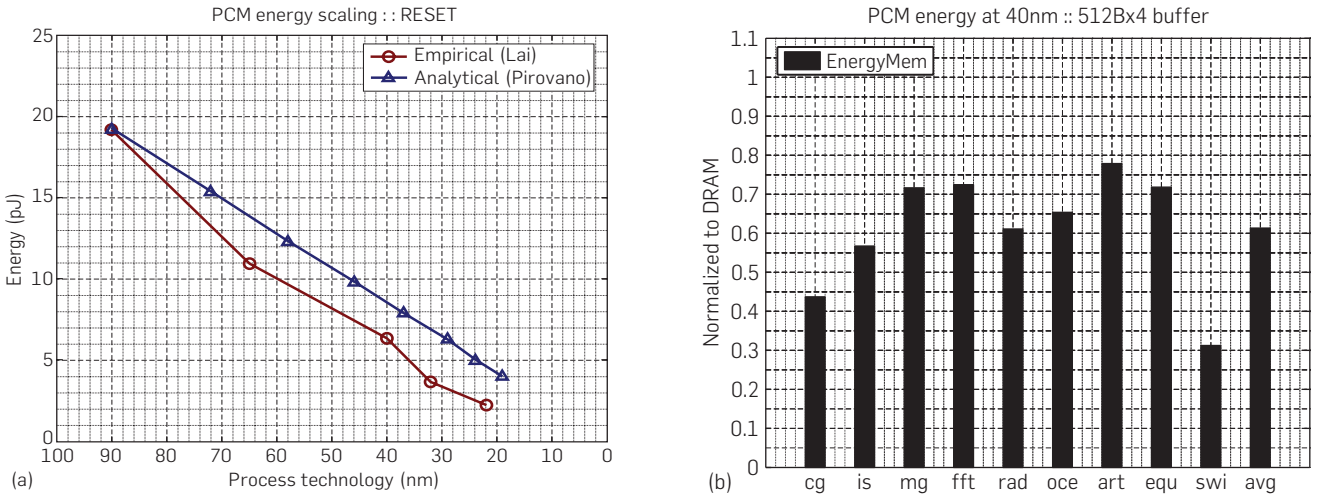
DRAM scaling faces many significant technical challenges as scaling attacks weaknesses in both components of the one transistor, one capacitor cell. Capacitor scaling is constrained by the DRAM storage mechanism, which requires maintaining charge on a capacitor. In future, process scaling is constrained by challenges in manufacturing small capacitors that store sufficient charge for reliable sensing despite large parasitic capacitances on the bitline.

The scaling scenarios are also bleak for the access transistor. As this transistor scales down, increasing subthreshold leakage will make it increasingly difficult to ensure DRAM retention times. Not only is less charge stored in the capacitor, that charge is stored less reliably. These trends impact the reliability and energy efficiency of DRAM in future process technologies. According to ITRS, "manufacturable solutions are not known" for DRAM beyond 40nm.[17]

In contrast, ITRS projects PCM scaling mechanisms will extend to 32nm, after which other scaling mechanisms might apply.[17] Such PCM scaling has already been demonstrated with a novel device structure fabricated by Raoux.[15] Although both DRAM and PCM are expected to be viable at 40nm technologies, energy scaling trends strongly favor PCM with a 2.4× reduction in PCM energy from 80 to 40nm as illustrated in Figure 6a. In contrast, ITRS projects DRAM energy falls by only 1.5× at 40nm, which reflects the technical challenges of DRAM scaling.[17]

Since PCM energy scales down faster than DRAM energy, PCM subsystems significantly outperform DRAM subsystems at 40nm. Figure 6b indicates PCM subsystem energy is 61.3% that of DRAM averaged across workloads. Switching from DRAM to PCM reduces energy costs by at

Figure 6. PCM Scalability. (a) Reset energy scaling from a survey of empirical prototypes by Lai and an analytical analysis by Pirovano et al. (b) Memory energy projections for 40nm.



least 22.1% (art) and by as much as 68.7% (swim). Note this analysis does not account for refresh energy, which would further increase DRAM energy costs. Although ITRS projects constant retention time of 64ms as DRAM scales to 40nm,[17] less effective access transistor control may reduce retention times. If retention times fall, DRAM refresh energy will increase as a fraction of total energy costs.

4. MEMORY LIFETIMES

In addition to architecting PCM to offer competitive delay and energy characteristics relative to DRAM, we must also consider PCM wear mechanisms. To mitigate these effects, we propose partial writes, which reduce the number of writes to the PCM array by tracking modified data from the L1 cache to the memory banks. This architectural solution adds a modest amount of cache state to reduce the number of bits written. We derive an analytical model to estimate memory module lifetime from a combination of fundamental PCM technology parameters and measured application characteristics. Partial writes, combined with an effective buffer organization, increase memory module lifetimes to a degree that makes PCM in main memory feasible.

4.1. Partial writes

Partial writes track data modifications, propagating this information from the L1 cache down to the buffers at the memory banks. When a buffered row is evicted and contents written to the PCM array, only modified data is written. We consider partial writes at two granularities: lowest level cache line size (64B) and word size (4B).

These granularities are least invasive since modified words are tracked by store instructions from the microprocessor pipeline. In contrast, bit-level granularity requires knowledge of previous data values and expensive comparators. We analyze a conservative implementation of partial writes, which does not exploit cases where stores write the same data values already stored.

Partial writes are supported by adding state to each cache line, tracking stores using fine-grained dirty bits. At the dirty line granularity, 64B modifications are tracked beginning at the lowest level cache and requires only 1b per 64B L2 line. At the dirty word granularity, 4B modifications are tracked beginning at the L1 cache with 8b per 32B L1 line and propagated to the L2 cache with 16b per 64B L2 line. Overheads are 0.2% and 3.1% of each cache line when tracking dirty lines and words, respectively.

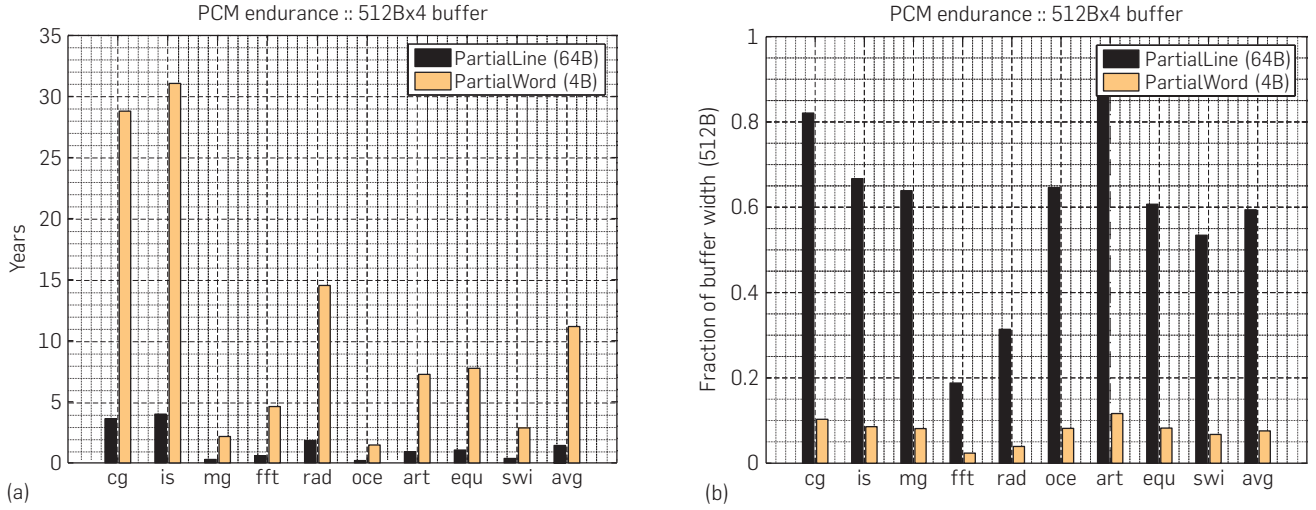
Table 1. Endurance model parameters.

Endurance		
$\hat{W}$	Writes per second per bit	Equation 1
$\hat{L}$	Memory module lifetime (s)	Equation 1
E	Write endurance	1E + 08
Memory Module		
C	Logical capacity (Gb)	2
Memory Bus Bandwidth		
f_m	Memory bus frequency (MHz)	400
M_f	Processor frequency multiplier	10
B	Burst length (blocks)	8
Application Characteristics		
N_w, N_r	Number of writes, reads	sim
T	Execution time (cy)	sim
Buffer Characteristics		
W_p, R_p	Buffer width (B), rows	512, 4
N_{wb}, N_{wa}	Buffer, array writes	sim
δ	Fraction of buffer written to array	sim

4.2. Endurance

Equation 1 estimates the write intensity observed by a memory module driven with access patterns observed in our memory-intensive workloads. Table 1 summarizes the model parameters. The model estimates the number of writes per second $\hat{W}$ for any given bit. We first estimate memory bus occupancy, which has a theoretical peak command bandwidth of $f_m \cdot (B/2)^{-1}$. Each command requires $B/2$ bus cycles to transmit its burst length B in a DDR interface, which prevents commands from issuing at memory bus speeds f_m. We then scale this peak bandwidth by application-specific utilization. Utilization is computed by

Figure 7. PCM Endurance. (a) PCM memory module lifetimes. (b) Fraction of buffer modified (δ).



measuring the number of memory operations $N_w + N_r$ and calculating the processor cycles spent on these operations $(B/2) \cdot M_f$. The processor is M_f faster than f_m. The time spent on memory operations is divided by total execution time T.

$$\hat{W} = \underbrace{\frac{f_m}{B/2} \cdot \frac{(N_w + N_r) \cdot (B/2) \cdot M_f}{T}}_{\texttt{memBusOcc}} \times \underbrace{\frac{N_w}{N_w + N_r}}_{\texttt{writeIntensity}} \times \underbrace{8W_p \cdot \delta \cdot \left(\frac{N_{wa}}{N_{wb}}\right)}_{\texttt{bufferOrg}} \times \underbrace{\frac{1}{C}}_{\texttt{capacity}} \quad (1)$$

Since only a fraction of memory bus activity reaches the PCM to induce wear, we scale occupancy by write intensity to estimate the number of write operations arriving at the row buffers. In the worst case, the entire buffer must be written to the array. However, not all buffer writes cause array writes due to coalescing. N_{wa}/N_{wb} measures the coalescing effectiveness of the buffer, which filters writes to the array. Lastly, partial writes mean only the dirty fraction δ of a buffer's $8W_p$ bits are written to the array. Assuming ideal wear-leveling, writes will be spread across the C bits in the module. Given writes per second $\hat{W}$ and characterized endurance E, a bit will fail in $\hat{L} = E/\hat{W}$ seconds.

In a baseline architecture with a single 2048B-wide buffer, average module lifetime is approximately 1050 h as calculated by Equation 1. For our memory-intensive workloads, we observe 32.8% memory bus utilization. Scaling by application-specific write intensity, we find 6.9% of memory bus cycles are utilized by writes. At the memory banks, the single 2048B buffer provides limited opportunities for write coalescing, eliminating only 2.3% of writes emerging from the memory bus. Frequent row replacements in the single buffer limit opportunities for coalescing.

Figure 7 illustrates significant endurance gains from reorganized buffers and partial writes. 64B and 4B partial writes improve endurance to 1.4 and 11.2 years, respectively. Buffers use partial writes so that only a fraction of the buffer's bits is written to the array. As shown in Figure 7, only 59.3% and 7.6% of the buffer must be written to the array for 64B and 4B partial writes.

4.3. Density versus endurance

PCM cells are presently larger than DRAM cells. Measuring cell size in square feature sizes, which makes the discussion independent of process technology, PCM cells are 1.5–2.0× larger than DRAM cells.

In particular, $8F^2$ DRAM cells provide a sufficiently wide pitch to enable a folded bitline architecture, which is resilient against bitline noise during voltage sensing. However, manufacturers often choose the density of $6F^2$ DRAM cells. The narrow pitch in $6F^2$ designs preclude folded bitlines, increasing vulnerability to noise and requiring unconventional array designs. For example, Samsung's $6F^2$ implements array blocks with 320 wordlines, which is not a power of two, to improve reliability.[5]

In contrast, PCM cells occupy between $6F^2$ and $20F^2$.[10] Part of this spread is due to differences in design and fabrication expertise for the new technology. However, we also observe a correlation between cell size and access device (e.g., the $6F^2$ cell uses the relatively small diode). We favor larger BJTs for their low access times. Cells with BJTs occupy between $9F^2$ and $12F^2$.

Given 9–$12F^2$ PCM cells and $6F^2$ DRAM cells, two-bit multilevel PCM cells are necessary to be competitive with respect to density. Two-bit MLC provide an effective density of 4.5–$6.0F^2$per bit. However, MLC suffer from lower endurance. Process and manufacturing set the read window, which quantifies the difference between the lowest and highest programmed resistances in single-level cells. By programming the cell to intermediate resistances within the same read window, MLC inherently require a larger number of logical states that each occupy a narrower region of the read window. Thus, wear more quickly impacts the ability to differentiate these resistances.

4.4. Assumptions and qualifications

Considering only memory-intensive workloads, this analysis

is conservative. PCM subsystems would more likely experience a mix of compute and memory-intensive workloads. Expected lifetimes would be higher had we considered, for example, single-threaded SPEC integer workloads. However, such workloads are less relevant for a study of memory subsystems. Moreover, within memory-intensive workloads, we would expect to see a mix of read and write intensive applications, which may further increase lifetimes.

Scalability is projected to improve PCM endurance from the present 1E+08 to 1E+12 writes per bit at 32nm with known manufacturable solutions.[17] This higher endurance increases lifetime by four orders of magnitude in our models. ITRS anticipates 1E+15 PCM writes at 22nm although manufacturable solutions are currently unknown.

5. CONCLUSION

The proposed memory architecture lays the foundation for exploiting PCM scalability and nonvolatility in main memory. Scalability implies lower main memory energy and greater write endurance. Furthermore, nonvolatile main memories will fundamentally change the landscape of computing. Software cognizant of this newly provided persistence can provide qualitatively new capabilities. For example, system boot/hibernate will be perceived as instantaneous; application checkpointing will be inexpensive[7]; file systems will provide stronger safety guarantees.[6] Thus, we take a step toward a new memory hierarchy with deep implications across the hardware–software interface. C

References

1. Aslot, V., Eigenmann, R. Quantitative performance analysis of the SPEC OMPM2001 benchmarks. *Sci. Program. 11*, 2 (2003).
2. Bailey, D. et al. NAS parallel benchmarks. In *Technical Report RNR-94-007, NASA Ames Research Center*, March 1994.
3. Bedeschi, F. et al. A multi-level-cell bipolar-selected phase-change memory. In *International Solid-State Circuits Conference*, 2008.
4. Chen, Y. et al. Ultra-thin phase-change bridge memory device using GeSb. In *International Electron Devices Meeting*, 2006.
5. Choi, Y. Under the hood: DRAM architectures: 8F2 vs. 6F2. *EE Times*, February 2008.
6. Condit, J. et al. Better I/O through byte-addressable, persistent memory. In *Symposium on Operating System Principles*, Oct 2009.
7. Dong, X. et al. Leveraging 3D PCRAM technologies to reduce checkpoint overhead in future exascale systems. In *Conference on Supercomputing*, Nov 2009.
8. Horii, H. et al. A novel cell technology using N-doped GeSbTe films for phase change RAM. In *Symposium on VLSI Technology*, 2003.
9. Lai, S. Current status of the phase change memory and its future. In *International Electron Devices Meeting*, 2003.
10. Lee, B., Ipek, E., Mutlu, O., Burger, D. Architecting phase change memory as a scalable DRAM alternative. In *International Symposium on Computer Architecture*, June 2009.
11. Lee, K.-J. et al. A 90nm 1.8V 512Mb diode-switch PRAM with 266MB/s read throughput. *J. Solid State Circuit. 43*, 1 (Jan 2008).
12. Micron. 512Mb DDR2 SDRAM component data sheet: MT47H128M4B6-25. In www.micron.com, Mar 2006.
13. Nirschl, T. et al. Write strategies for 2 and 4-bit multi-level phase-change memory. In *International Electron Devices Meeting*, 2008.
14. Pirovano, A. et al. Scaling analysis of phase-change memory technology. In *International Electron Devices Meeting*, 2003.
15. Raoux, S. et al. Phase-change random access memory: A scalable technology. *IBM J. Res. Dev. 52*, 4/5 (Jul/Sept 2008).
16. Renau, J. et al. SESC simulator. In http://sesc.sourceforge.net, 2005.
17. Semiconductor Industry Association. Process integration, devices & structures. *International Technology Roadmap for Semiconductors*, 2007.
18. Sinha, M. et al. High-performance and low-voltage sense-amplifier techniques for sub-90nm sram. In *International Systems-on-Chip Conference*, 2003.
19. Woo, S. et al. The SPLASH-2 programs: Characterization and methodological considerations. In *International Symposium on Computer Architecture*, June 1995.

Benjamin C. Lee (bcclee@stanford.edu), Stanford University.

Engin Ipek (ipek@cs.rochester.edu), University of Rochester.

Onur Mutlu (onur@cmu.edu), Carnegie Mellon University.

Doug Burger (dburger@microsoft.com), Microsoft Research.