Exploring Phase Change Memory and 3D Die-Stacking for Power/Thermal Friendly, Fast and Durable Memory Architectures

Wangyuan Zhang and Tao Li
Intelligent Design of Efficient Architecture Lab (IDEAL)
Department of Electrical and Computer Engineering
University of Florida
zhangwy@ufl.edu, taoli@ece.ufl.edu

Abstract

Emerging three-dimensional (3D) integration technology allows for the direct placement of DRAM on top of a microprocessor, significantly reducing the wire-delay between the two and thereby alleviating memory latency and bandwidth constraints. However, the increase in power density of 3D technology leads to elevated on-chip temperature, which results in an exponential rise in charge leakage of DRAM. Consequently, the refresh frequency of 3D die-stacked DRAM needs to be doubled (or more) to retain data at the expense of additional power overhead. In this work, we investigate using Phase-change Random Access Memory (PRAM) as a promising candidate to achieve scalable, low power and thermal friendly memory system architecture in the upcoming 3D-stacking technology era. Using analytical model, circuit- and architectural- level simulations that capture both physical and electrical characteristics of PRAM, we show that the higher temperature of 3D chips is beneficial to PRAM power savings due to its unique, heat-driven programming mechanisms. Moreover, we show that the Through Silicon Vias (TSVs) ubiquitously used in 3D implementations contribute further PRAM power savings due to their substantially lower resistance to the high PRAM programming current.

To effectively integrate PRAM into a conventional memory hierarchy, we propose architecture and OS support to address its write latency and reliability disadvantages. We present a hybrid PRAM/DRAM memory architecture and exploit an OS-level paging scheme to improve PRAM write performance and lifetime. Moreover, we leverage the error-correcting capability of strong ECC codes to expand PRAM lifespan and use wear-out aware OS page allocation to minimize ECC performance overhead. Our experimental results show that compared to die-stacked planar DRAM, our design reduces the overall power consumption of the memory system by 54% with 6% performance degradation, consequently alleviating the thermal constraint of 3D chips by up to 4.25°C and achieving a speedup of up to 1.1X. We also show that the lifetime can be improved by a factor of 114X using the proposed endurance optimization schemes.

1. Introduction

Dynamic Random Access Memory (DRAM) has been used as the main memory in computer systems for decades due to its high-density, high-performance and low-cost. However, DRAM technologies are facing both scalability and power issues. DRAM is difficult to scale down beyond 50nm[1] due to various limitations associated with device leakages and retention time. DRAM-based main memory is also consuming an increasing proportion of the power budget and has been reported to account for as much as 40% of the total system power [2].

Recently, three-dimensional (3D) integration has emerged as a promising technique. 3D technology creates multiple active dies stacked vertically on a single chip and uses short Through Silicon Vias (TSVs) to interconnect circuits across layers to reduce the wire length and delay. Previous studies[3, 4, 5] have observed impressive performance improvement by stacking memory on top of the microprocessor with low-delay, high-bandwidth connections between them. Despite the performance advantage, the elevated on-chip temperature [3, 4] due to high power density presents significant challenges for DRAM power management [6]. Since the charge leakage of a DRAM cell grows exponentially as the temperature increases, the elevated on-chip temperature will accelerate the degradation of DRAM data retention, which needs to be addressed by increasing DRAM refresh frequency. Consequently, the 3D-stacked DRAM is expected to operate at double (or higher) the current refresh rate [6], leading to a higher refresh power overhead. Worse, this increased power dissipation will further aggravate on-chip temperature and exacerbate thermal constraints. To achieve low power, traditional DRAM power management techniques attempt to eliminate the unnecessary refreshes [6, 7] or put idle banks into power saving mode [8]. However, the temperature dependent DRAM leakage can not be overcome.

In contrast to conventional memory technology, a cutting-edge memory technology, Phase-change Random Access Memory (PRAM), is attracting increasing attention as a promising candidate for next generation memories [9, 10, 11, 12, 21]. PRAM is a type of non-volatile memory that uses the unique behavior of chalcogenide glass, which can be switched between two states (i.e. crystalline and amorphous) with the application of heat. The desirable characteristics of PRAM include random access, fast read access, low standby power, superior scalability (no physical limits down to 20nm technology node [13]), compatible with CMOS process[14] etc. In this paper, we advocate and present a case in which PRAM, as a replacement for DRAM, can be employed to implement 3D-stacked memory systems with lower power consumption and alleviated temperature constraints. Our approach leverages two attractive features provided by PRAM: low standby power and high-temperature friendly operation. The former is a common feature of all non-volatile storages as data can be retained in them even when not powered. The latter is a unique characteristic of PRAM: to store data in PRAM, the temperature needs to be elevated to switch the state of cells. Using analytical and circuit-level modeling that characterizes PRAM in detail, we observe that the programming power of PRAM cells can be reduced as the chip temperature is elevated, in contrast to the exponential

increase in refresh power for DRAM when chip temperature increases. This high-temperature friendly feature makes PRAM superior to DRAM for die-stacked memory systems. In addition, we investigate the power benefit of using 3D TSVs to deliver PRAM programming current. Compared with conventional metal wires, the low resistance of TSVs minimizes the dissipated power along the bit lines. We show that PRAM benefits more from TSVs than DRAM in terms of power savings.

Two major challenges that need to be addressed for design using this emerging memory technology are PRAM high write latency and limited endurance. In this paper, we propose a hybrid main memory design that is composed of a large portion of PRAM used as a primary memory space and a small portion of DRAM that serves as a write buffer to reduce the number of writes to the PRAM partition. To maximize the runtime efficiency of the DRAM-based write buffer, we propose an OS-level paging scheme that takes into account the memory reference characteristics of applications and migrates the hot-modified pages from PRAM to DRAM so that the life time degradation of PRAM is alleviated. This hybrid design with page migration also provides an improved memory write performance over PRAM-only memory design due to the lower write latency on DRAM than that on PRAM. As writes to the PRAM cannot be entirely avoided and the wear-out on a given PRAM cell is exacerbated as the number of overwrites increases, it is crucial to provide an endurance optimization scheme that can maximize and balance the life span of all PRAM cells. Toward this end, we propose a synergetic reliability enhancement approach that combines architecture- and OS -level wear-out optimizations. At the architecture level, our approach uses Error Correction Code (ECC) with varied strength to detect and correct a number of errors in each memory block fetched. However, the performance overhead (increases with the number of errors that need to be corrected due to the enlarged delay in error correction) of using ECC with increased strength is non-trivial. To minimize the ECC-induced error correction penalty in performance and achieve wear-leveling, we propose to use an OS paging scheme to perform endurance-aware allocation.

Our experimental results show that the elevated temperature and 3D TSVs provide up to 46% overall power savings for PRAM programming operations. Our baseline hybrid PRAM/DRAM memory architecture incurs negligible power overhead with 3% performance improvement over PRAM-only main memory and leads to up to 4X lifetime extension. The proposed dynamic ECC strength technique improves the endurance of PRAM by 28X over the design with no ECC and the endurance-aware OS page allocation reduces the ECC-induced performance overhead by 10%. Putting it all together, we show that our die-stacked hybrid PRAM/DRAM with architecture and OS support is capable of saving 51% of main memory energy (54% saving of memory power with 6% performance penalty). The eliminated power dissipation relieves thermal stress of 3D chips, which in turn allows our system to run up to 10% faster under a typical thermal constraint.

The rest of this paper is organized as follows. Section 2 provides a brief background on PRAM. Section 3 characterizes the benefits of 3D technology on PRAM programming power. Section 4 proposes the die-stacked hybrid PRAM/DRAM memory system and its architecture and OS support. Section 5 describes our experimental methodologies, simulation frameworks and workloads. Section 6 presents our evaluation results. Section 7 discusses related work and section 8 concludes the paper.

2. Background

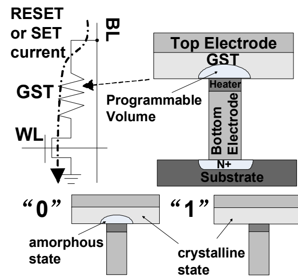


Figure 1. The basic structure of a PRAM cell[16]*

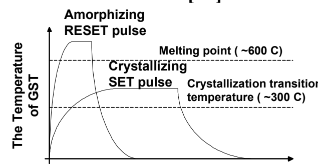


Figure 2. The programming pulses of PRAM[14]

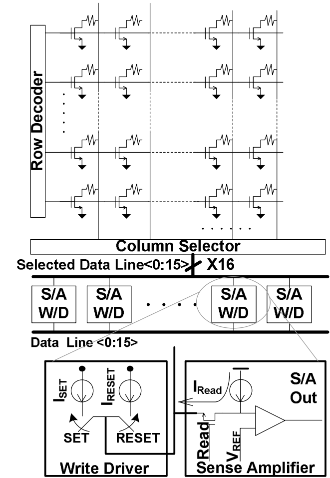


Figure 3. Sub-array architecture of PRAM memory

*The SET and RESET states of the PRAM correspond to a stored binary "1" or binary "0".

The basic structure of a PRAM cell is composed of a standard NMOS access transistor and a small volume of phase change material, GST ($Ge_2Sb_2Te_5$), as shown in Figure 1. The GST can be switched between two states (i.e. crystalline and amorphous) with dramatically different electrical resistance. The amorphous, high resistance state is used to represent a binary "0", while the crystalline, low resistance state represents a "1". The state of the cell is switched by heating the region of GST to a high temperature threshold through electrical-pulse generated Joule heat [16]. The heating current passed through the cell are I_{RESET} and I_{SET} with different amplitude and duration, as shown in Figure 2. The magnitude of the I_{RESET} is selected so as to just melt a small region of GST that is adjacent to the heater. After as little as 5ns of heating, I_{RESET} is cut off abruptly, which causes the melted region to be quenched into the amorphous, high resistance state. The amplitude of I_{SET} is smaller, but longer, so that the GST is heated to a temperature between melting point and crystallization transition temperature and left in the crystalline, low resistance state. The data stored in a cell is sensed by measuring its resistance when a small read current I_{Read}, less than 100μA[14], is passed through it, thereby consuming very small power for read. PRAM technology has reached a good level of maturity and the fabrication cost of phase change memory cells is small: the number of extra mask steps beyond a standard CMOS process is no more than four [14]. Similar to DRAM, the PRAM is hierarchically organized as sub-arrays, arrays and banks. As shown in Figure 3, a PRAM sub-array consists of a number of cells, decoders for row/column addresses, sense amplifiers (S/As) and write drivers (W/Ds). Unlike DRAM, PRAM employs current sense amplifiers, which are shared and multiplexed across bit-lines due to its large circuit size. For read and write operations, column selector circuit selects 16 bit-lines and makes them connected to S/As for reading or W/Ds for writing. For

DRAM, a row is activated and read into sense amplifiers and data is then taken from the sense amplifiers, selected by the column address. In the case of PRAM write operations, the write current flows from W/D to the cell ground line through column selector, bit-line, GST and access NMOS transistor. For read operations, the read current follows the similar path except that it originates from sense amplifier.

From a power savings perspective, the non-volatile nature of PRAM makes it more favorable than the volatile DRAM because the standby and read power of PRAM are very small. However, 60ns read[17] and 50/120ns[17] to write "1"/"0" for PRAM is much higher than 10ns[17] read/write for DRAM (Note that these latencies don't account for the decoder latency). Thanks to recent technology improvements [19], PRAM has achieved a read access latency that is comparable to current DRAM. Prior studies [18, 39] have proposed using other non-volatile devices, Flash and MRAM, to save memory power. However, using Flash as a general-purpose solution for main memory is still limited by its performance (25µs read and 200µs write latency [18]) and lifespan (1E05 [18]). PRAM is considerably faster than Flash and is able to perform overwrite on any byte without erasing the entire data block. Moreover, PRAM endurance (1E08 as projected by ITRS 2007 Roadmap [17]) is considerably longer than Flash by a factor of 1E03. Although MRAM was used for low power cache design in [39], it isn't suitable for main memory implementation due to considerably lower density than DRAM (cell size $20F^2$ versus $6F^2$[17]) . On contrary, PRAM has a slightly higher density than DRAM ($4.8F^2$ versus $6F^2$ [17]), allowing it to be an alternative device for main memory design. In this paper, we conservatively assume the same density for PRAM and DRAM.

3. PRAM Power Characterization under 3D Integration Technology

By stacking memory directly on top of the microprocessor, [20] reports a 65% performance gain by placing a planar DRAM on top of the processor and [3] further shows an 175% speedup over die-stacked planar DRAM by proposing a true-3D design. However, the increased temperature of 3D chips causes the DRAM to consume more refresh power [6]. The heat-driven programming mechanism of PRAM is more temperature-friendly than DRAM, motivating us to explore PRAM as an alternative candidate for die-stacked memory design. In this work, we perform a detailed characterization of the impact of high temperature on PRAM programming power and further investigate the benefit of 3D TSVs on PRAM programming power. Our results show that PRAM-technology is well suited for 3D-stacked memory implementation.

3.1 Characterizing the Impact of Temperature on PRAM Programming Power

As described in Section 2, PRAM has substantially reduced read and standby power consumption. As a result, overall PRAM power is dominated by its programming power (i.e. SET and RESET power). To characterize the impact of temperature on the PRAM programming power, we use the one-dimensional heat conduction model [16]. This heat conduction model captures the flow of heat in PRAM device that is insulated everywhere except at the two ends, which connect to bit-line and access transistor. The elevated ambient temperature can reach the PRAM devices and help heat them through the two ends. On the other hand, the generated thermal in PRAM devices may also dissipate to the surrounding area, increasing on-chip temperature. Overall, PRAM programming activity has little impact on chip temperature due to the small amount of concurrent writes at anytime. Due to space limitations, we only present our analysis for the RESET operation briefly in this paper (refer to [16] for details) and a similar characterization procedure has been applied to the SET operation.

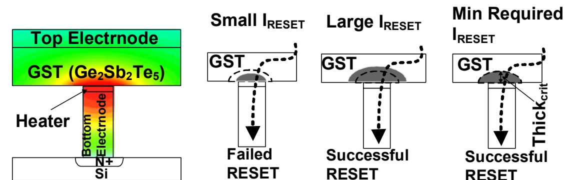


Figure 4. The overall architecture of a PRAM device and the illustration of a successful RESET and the one dimensional temperature profile [16]

The overall structure of the modeled PRAM device is shown in Figure 4. We use the same dimensional parameters as those in [16]. Phase switching of $Ge_2Sb_2Te_5$ is triggered by a Joule resistive heater (q_{Joule}), which can be specified as $q_{Joule} = I_F^2 Rt$ (Eq-1), where I_F is the current passing through the device, R is the resistance of the Joule heating material, and t is the heating time. We use the heat conduction model (Eq-2) to capture the temperature profile shown in Figure 4.

$$\rho_i \frac{\partial T_i}{\partial t} = \frac{\kappa_i}{c_i}\frac{\partial^2 T_i}{\partial x^2} + \frac{\dot{q}_{Joule,i}}{c_i} \text{ (Eq-2)},$$

where x is the distance from the active region of Si substrate, T_i is the temperature, t is the heating time, ρ_i is the density, κ_i is the thermal conductivity, and c_i is the specific heat capacity for the ith layer (i.e. Bottom Electrnode Layer, Heater Layer and GST Layer) with i=1,2,3. As described in Section 2, a temperature rise due to the heat generated by the reset current is required to melt a small region of phase change material adjacent to the heater to achieve a successful RESET operation. The minimal thickness of this required region is defined as $Thick_{crit}$ (as illustrated in Figure 4), which will determine the minimal required programming current. By using Eq-1 and Eq-2, we compute the minimal required programming power as the ambient temperature varies from 40°C to 95°C, a range that covers the temperature of both planar and 3D die-stacked chips [3].

Figure 5 illustrates the effect of chip temperature on a PRAM device's programming power: less programming current is required to RESET/SET a PRAM device at elevated temperature, resulting in smaller programming power. The reduction for RESET and SET power is 15.7% and 19.4% respectively when the ambient temperature increases from 45°C to 85°C. Note that the temperature threshold for RESET and SET is ~600°C and ~300°C respectively. A 19.4% power saving is achieved for SET when the temperature gap is bridged by 13% (40°C/300°C=13.3%). A lower percentage of power saving (i.e. 15.7%) is obtained for RESET due to a smaller gap (i.e. 40°C/600°C=6.7%) bridged. The power saving percentage is greater than that of temperature increment. This is because the flow rate of heat dissipated from the PRAM device through the

two ends is proportional to the temperature gradient between the device and surrounding materials. In addition to bridging the temperature gap, the elevated on-chip temperature reduces this gradient and increases heating efficiency, resulting in further power saving. Our results are largely consistent with [43], which reports experimental data on temperature dependence of RESET power and shows a 25% power saving when temperature is elevated by 16% (100°C/600°C=16%). In contrast to PRAM, DRAM consumes more power at an elevated temperature due to the increased refresh frequency (driven by temperature rise). A typical DRAM refresh interval is 64ms for 2D planar chips [6, 22]. As we employ 3D-stacked on-chip DRAM, the refresh rate is required to be at least doubled if the operating temperature exceeds 85°C [22]. We use the power models from DRAMsim [23] to estimate the power overhead of shortening the refresh interval from 64ms to 32ms when running simulated workloads (see Section 5 for detailed experimental setup). The results show an average 3X increase in refresh power and an average 8% increase in total power for die-stacked DRAM (detailed in section 6.1). As the ambient temperature increases in 3D stacking design, thermal cross talking may cause PRAM programming currents for one cell to interfere the states of adjacent cells. We simulate thermal cross-talk effect for PRAM devices at 65nm technology and our simulation shows the temperature falls exponentially with the increased distance from the programmed cell. This temperature becomes close to the ambient temperature at the adjacent cells, suggesting no thermal coupling effect on 3D chip with elevated temperature.

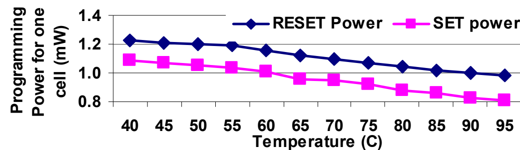


Figure 5. The temperature dependence of a PRAM device's programming power

3.2 Characterizing the Impact of 3D TSVs on PRAM Programming Power

Through Silicon Vias are ubiquitously adopted in 3D design to reduce wire-length, thereby achieving power reduction. In this section, we analyze the power benefit obtained by TSVs in 3D-implmented DRAM and PRAM and show that TSVs benefit PRAM power savings more significantly than they do for DRAM.

The dynamic power ($P_{dyn} = fCV_C^2$) due to charging and discharging interconnect capacitance is one of the primary contributors to the power consumption of read and write operations for DRAM cells. By using TSVs, the substantial wire length reduction decreases the interconnect capacitance, leading to less dynamic power consumption. Intuitively, since C is proportional to the wire length, given a fixed f and V_c, the dynamic power reduction is proportional to the wire length reduction. However, the dynamic power saving of true-3D DRAM is less than the fraction of wire length reduction (detailed later in this section). This is because the TSVs used to form the vertical bus are of high capacitance [20], which offsets the interconnect capacitance reduction due to the shortened interconnect, leading to a substantial loss in power savings. Differing from DRAM, the power dissipation of write operations for PRAM is dominated by the programming power ($P_{programming} = I^2R$) rather than $P_{dyn} = fCV_C^2$. The reasons are two-fold: first, PRAM write operations are carried out only on those PRAM cells that are selected by both row decoder and column decoder (as is discussed in Section 2). Therefore, only bit lines that connect to those selected PRAM cells will be charged or discharged during write operations, leading to a substantial reduction in C as well as P_{dyn}. In contrast, the read/write operations in DRAM will cause all cells in a selected row to be read out / written back, because capacitors in this row are connected to bit-lines once the row is activated. These capacitors then start to charge or discharge all bit-lines, resulting in a significantly higher C in $P_{dyn} = fCV_C^2$ for DRAM. Second, the magnitude of the PRAM write current required to flow through the direct current path between V_{DD} and GND during write operations is very high. Therefore, the power dissipated along the path due to the resistance outweighs the power required to charge interconnect wires. By using TSVs in die-stacked PRAM design, the resistance is decreased due to the shortened interconnect, thereby saving programming power. More importantly, the physical dimension of TSVs makes their resistance per unit length much smaller. Hence adopting TSVs in die-stacked PRAM retains the power savings benefit due to the reduced resistance (as a result of short wires) along the current path. More interestingly, we observed that the reduced resistance along the current path allows the programming current to be further reduced, while still being able to achieve successful programming operations. Since a smaller magnitude of current is used, power saving can be achieved.

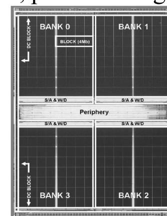


Figure 6. 2D planar PRAM prototype [24]

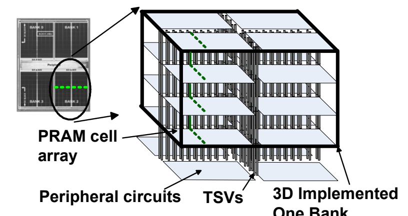


Figure 7. PRAM constructed in true-3D organization

To characterize PRAM power under 3D TSVs, we model a 256Mb PRAM similar to the prototype developed by [24] (the layout is shown in Figure 6). Instead of using the 2D planar architecture and layout, we adopt the true-3D [3] structure announced by Tezzaron Corporation, which constructs on-chip DRAM memory by stacking individual bit-cell arrays in a 3D fashion [5]. Figure 7 shows an overall organization of a true-3D PRAM. The bit-cells belonging to one bank are distributed across the top 4 layers and interconnected by TSVs. A dotted line in the planner design represents a row of memory cells. In a true-3D design, the row is divided into four sections with each section residing on one layer, as is shown by four dotted lines at the top four layers in true-3D organization. The peripheral circuits are located at the bottom layer for speed optimization. We built circuit model and used SPICE simulations to quantify the overall benefit of TSVs on PRAM programming power. Our SPICE results show that PRAM programming power reduces from 1.2mW (planar design) to 0.81mW (3D design), a 32.5% power saving. We built additional circuit models for DRAM and used SPICE simulations to evaluate the dynamic power savings

achieved by a 4-layer true-3D DRAM. Our simulation shows that the percentage of dynamic power savings is only 20%, considerably less than that achieved on true-3D PRAM. By combining the 32.5% power savings due to TSVs and 19.4% power savings due to elevated temperature, we estimate that up to a 46% PRAM power savings can be achieved by 3D.

4. The Proposed 3D Die-Stacked Hybrid PRAM/DRAM System

The low standby power and temperature-driven operation of PRAM technology make it well suited for die-stacked memory systems from the perspective of power savings. However, there are several challenges (i.e. write bandwidth, endurance) that need to be addressed for the practical and effective integration of die-stacked PRAM. In this section, we propose a hybrid, true-3D-implemented PRAM/DRAM memory architecture along with architecture and OS support to address these challenges.

4.1 An Overview of the Proposed Architecture

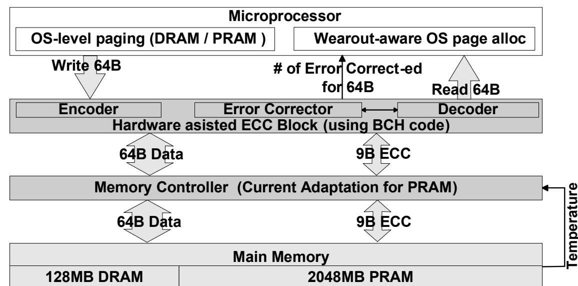
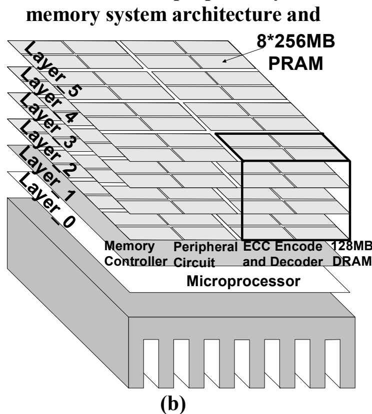


Figure 8. (a) An overview of the proposed hybrid PRAM/DRAM memory system architecture and



(b)

Figure 8. (b) The 3D integration of processor and memory system

Figure 8(a) shows an overview of the proposed PRAM-based 3D memory architecture. Our design consists of a true-3D-implemented 2048MB PRAM and a planar 128MB DRAM partition to exploit the heterogeneous performance and power characteristics of the two. The high-speed, power-efficient write access and the infinite lifespan of DRAM make it suitable to serve as a write partition since doing so improves memory write bandwidth and avoids PRAM's expensive write power and wear-out. On the other hand, the very low power of PRAM allows greater power savings. We use OS-level paging (detailed in section 4.2) to migrate pages to the DRAM partition based on a program's run-time memory reference characteristics.

As the chip temperature affects PRAM programming current, we augment the memory controller with the capability of adapting the current used for programming PRAM cells based on the ambient temperature. We employ 16 digital thermal sensors similar to those used by Intel Core Duo [25] on PRAM layers with 4 sensors per layer to collect run-time temperature profile and feed it back to the memory controller. The sensors are placed in locations where hotspot is likely to appear. With the collected run-time temperature, memory controller sends signals to PRAM peripheral circuits to indicate the current level used for memory writes. We augment the write driver circuit block (shown in Figure 3) to implement a current-adjustment scheme with 8-level tuning resolution. This tuning resolution is determined by making a tradeoff between design complexity/overhead and performance. The estimated area overhead of this scheme is 0.02 mm^2. The memory controller and memory modules are connected by 3D TSVs, which form a fast, vertical interconnect across layers. Note that the width of this TSV-based bus is not limited by the pin-count of the off-chip memory. In this study, we assume a bus with a width of 73 Bytes (i.e. 64 Byte data + 9 Byte ECC code) and the type of ECC code adopted is a BCH code to ensure the reliability of the memory system and improve its lifetime by varying its error correction capability (detailed in section 4.3). The ECC encoding and decoding (including correcting errors if necessary) are performed on each memory access. This ECC functional block is implemented using dedicated hardware. The number of corrected errors by the ECC hardware is passed to the OS (through an interrupt when a correction occurs) to adapt its page allocation scheme by favoring lightly used physical pages over heavily used ones during memory allocation. This achieves a balanced wear-out on the PRAM partition. It also offers a performance benefit, because accessing heavily worn-out physical memory incurs considerable delay in ECC decoding due to the time required for correcting errors.

As shown in Figure 8(b), we assume a quad-core microprocessor with a shared L2 cache and die-stacked on-chip memory and each core is an Alpha 21364 microprocessor core. We organize them as a six-layer 3D-stacked chip by allocating the quad-core on Layer_0 (i.e. the layer that is closest to the heat sink for the purpose of thermal efficiency). Layer_1 is used for planar 128MB DRAM, the memory controller, PRAM memory peripheral circuits, and ECC encoder and decoder with optimized speed for CMOS technology. Layer_2 to Layer_5 are dedicated for PRAM bit-cells(consisting of a PRAM device and an access transisotr), which are designed in a true 3D fashion and realized based on a traditional NMOS technology optimized for density. Note that in our design, we stack DRAM and PRAM in different layers to reduce fabrication cost since the PRAM fabrication process is different than that of the DRAM. The DRAM partition consists of one bank and the PRAM partition has eight banks.

4.2 PRAM-aware OS Paging

Upon a paging request, a conventional OS virtual memory management scheme allocates the next available page(s) without taking into account the characteristics of the underlying storage media. The key idea of our PRAM-aware paging scheme is to favor PRAM over DRAM when allocating cold-modified pages which are infrequently updated, so that the very low standby and read power benefits of PRAM can be fully exploited, while allocating hot-modified pages to the DRAM partition to avoid the write latency and mitigate wear-out of PRAM.

First, we assume pages used for kernel space are always located on the DRAM partition, as they are very likely to be modified intensively. For pages used for user space, we use counters to track the frequency of page updates and adopt a modified Multi Queue Algorithm [26] to classify hot- or cold-modified pages. Specifically, multiple LRU queues (denoted as Q_0, Q_1 … Q_{n-1}) are used with a different rank for each queue. When a page is modified for the first time, its page number is inserted into the tail entry of the queue with the lowest rank (i.e. Q_0) and the modification counter associated with this page is set to 1. Later, if the same page is updated again, its modification counter is updated and the page number is removed from its current LRU queue Q_i and is placed to the tail entry at another queue with an m-higher rank, Q_k , where m is a function of its modification counter f. We use m= log2(f) in our design as a prior study [26] shows this function outperforms others. A periodic demotion of all page numbers in the queues is performed by degrading page numbers from their current queue to a one-lower ranked queue and by discarding those page numbers stored in the lowest-ranked queue. In addition, all modification counters associated with page numbers are halved by shifting right one bit. In the normal Linux OS, any context switch interval between 10−200 ms may be used. We use 10ms as our periodic demotion interval to make our page classification aware of the altered memory reference behavior due to running different programs caused by a context switch. We use 16 queues and we categorize pages in the 8-highest ranked queues as hot-modified pages, while the pages in the remaining 8 queues are regarded as potential hot-modified pages. Previous research has shown that as few as 8 queues can be sufficient to separate hot pages from others [26].

For the initial page allocation, we satisfy the memory requests by allocating physical pages in PRAM. We migrate pages in the 8-highest ranked queues to DRAM and set a bit associated with its entry indicating the location of these pages. If there is not enough space in the DRAM, the pages whose page numbers don't exit in the 8-highest ranked queues will be migrated from DRAM back to PRAM (or written back to disk if no space available in PRAM). Then its page number will be removed from the queue. This prevents the frequent migration back and forth between DRAM and PRAM, since pages evicted from DRAM need to be updated frequently before they are promoted to the 8-highest ranked queues. Page migrations are performed transparently to the program with the aid of the OS, which is responsible for maintaining TLB coherence, copying the page to its new home (we emulate this migration in our simulations by invoking a bcopy() routine) and flashing the cache lines belonging to the pages to be migrated. Due to the die stacking, the high bandwidth between cores and memories allows migrations to be accomplished with significantly lowered latency as compared to a conventional off-chip memory organization. In our case, each queue contains 4,096 entries so that a total of 65,536 page numbers can be stored in 16 queues (each queue entry is 8Byte, among which 4Byte is used for page number and 4Byte is allocated for counter), implemented using a 512KB DRAM with an estimated die area overhead of $0.24mm^2$. Note that we don't allocate an entry for every page. An entry is allocated only when the page is recently modified. The periodic demotion will deallocate entries if their associated pages are not updated frequently enough. The 32,768 page numbers stored in the 8-highest ranked queues correspond to a size of 128MB memory area, which is size of DRAM partition in our hybrid design. The counters are incremented through read-modify-write operations in parallel with memory accesses to avoid performance overhead. This multi-queue algorithm is implemented in memory controller with an area overhead of $0.11mm^2$. We estimated its power overhead as approximately 11.1mW by using circuit-level tools and modified DRAMsim simulator.

4.3 Life Span Optimization using Varying ECC Strength and OS-level Wear Leveling

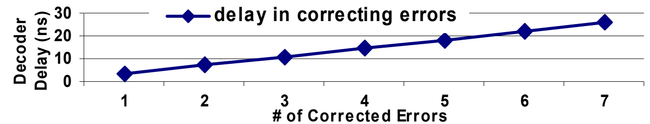


Figure 9. Latency overhead with the increased number of corrected errors

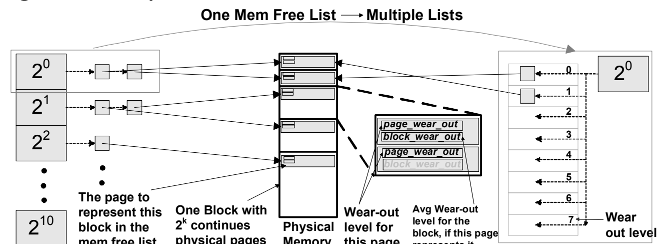


Figure 10. Wear-out aware OS page allocation

As write operations to PRAM are unavoidable, we propose using advanced error correction code (ECC) with OS-level scheme to ensure the reliability and to tolerate multi-bit failure in data blocks for expanding the life span of PRAM with minimum performance overhead. The most commonly used error correction code for DRAM is Hamming code, which can only correct a single-bit error in a message. To increase PRAM life span, we use an advanced ECC, Hamming and Bose-Chaudhuri-Hocquenghem (BCH) code [18], to correct up to 7 errors. Due to the read/write pattern and endurance variation (e.g. caused by process variation), a fraction of cells may fail far earlier than others (i.e. failures occur at the early age of the product). When memory references occur on these cells, ECC will inevitably harm the system performance due to the delay in correcting errors. Therefore, we further propose to feed the number of errors corrected to OS for achieving wear-out aware page allocation. This augmented page allocation scheme favors lightly worn pages over heavily worn ones. Doing so can achieve wear-leveling as well as performance benefit by minimizing performance overhead of BCH.

For each message data of k bits, we can construct a BCH codeword (message data + extra redundancy bits) with a length of n bits to correct up to t errors out of the entire codeword. The length of the codeword, n, should satisfy $n=2^m-1$ and $mt \leq n-k$. In our study, we assume an on-chip bus with a width of 64 Bytes. Due to storage overhead, we limit the number of correctable errors to 7 per BCH codeword. Given a value of 512 bits (64 Byte) for k and 7 for t, m can be found to be 10, and at least 70 (m*t=70) check bits are required. We use 9 Byte (72 bits > 70 bits) redundancy bits for each 64 Byte data. The area overhead is 14.1% (i.e., 9Byte/64Byte=0.141), which is slightly higher than

the overhead (8bit/64bit =0.125) of using Hamming-based ECC. Note that BCH decoding (encoding) occurs on every PRAM read (write) access. The decoding operation also includes error correction if errors are detected. We adopt the hardware-based encoder and decoder design proposed in [27] and estimate that the die area required for this hardware implementation is $1mm^2$. As shown in Figure 9, the performance overhead of BCH decoder rises substantially (i.e. from 3ns to 26ns) as the number of errors increases from 1 to 7. The estimated maximum power consumption due to tracking multiple bits in ECC is 0.2W, which occurs when 7 errors are required to be corrected.

To achieve endurance-aware OS page allocation, we modified the default page allocation algorithm to take wear-out level into consideration. Linux uses three buddy systems to handle page frames in DMA, normal and high memory zones respectively. The kernel delays allocating dynamic memory to user processes until page fault exception occurs. All free page frames are grouped into 11 lists of blocks that contain a variable number of contiguous page frames, as shown in Figure 10. Each block consists of 2^k pages in the kth list and the descriptor of the first page of a block is used to represent this block in the link-list. Blocks within a given list are unordered, so the allocation and de-allocation of pages are performed without considering the wear-out level of physical memory cells. We propose to augment the buddy allocator with wear-out awareness. The key idea is that pages with a lower level of wear-out are allocated before others. As shown in Figure 10, the single list is transformed into multiple lists, one for each wear-out level (e.g. eight lists for maximum seven correctable errors in our case). The wear-out level, which is indicated by the maximum number of errors detected and corrected in all memory references to a physical memory page, ranges from 0 to 7. With our scheme, pages are placed into multiple lists according to their wear-out level and they are unordered in these lists. Therefore, page allocation and de-allocation from a given list still remain an O(1) operation. Our proposed scheme requires a modification of existing page management routines and data structures. A variable called *page_wear_out* is added to the page descriptor to track the the wear-out level of a physical page. The value of *page_wear_out* is updated using the feedback from ECC hardware. The initial value of *page_wear_out* for each page is 0 and it increases as more errors are detected and corrected. A page allocation request is satisfied from the list with the lowest wear-out level first. If there are no free pages available, other lists of higher wear-out levels are considered in-order. For blocks with more than one page, we use another added variable called *block_wear_out*, which represents the average wear-out level of a block, and store the information in the page descriptor of the first page of the entire block. The value of *block_wear_out* needs to be updated during allocation and de-allocation, such as 1) splitting a block with a larger number of pages than the requested memory size in half; 2) merging together pairs of free blocks into a single block with doubled size. In addition, *block_wear_out* is updated when the *page_wear_out* of any page within this block changes. With our approach, pages with lower wear-out level are favored over others and the performance overhead of BCH code is minimized since fewer errors need to be corrected for lightly worn-out regions than for heavily worn-out ones, since less errors occurs in memory reference to the former, thereby less time required in correcting errors. We conservatively categorize a page as heavily worn even if a few cells in a row fail early, because it is possible that more writes will be performed on this row and cause more cells start to fail. Eventually ECC is not capable of correcting all errors. In this case, the entire page cannot be used since OS manages memory at page granularity.

Note that our technique is different than that used by flash, which relies on the Flash Translation Layer implemented at the file-system level , thereby involvling significant performance penalty and complexity. Our proposed scheme is built on exiting OS-page allocation algorithms to achieve wear-leveling with small overhead. Since the wear-out information is stored in memory as run-time data, it may be lost due to power off, OS reboot and reinstallation etc. By feeding back the number of corrected errors from hardware ECC block to OS, our scheme allows the OS to regain physical memory wear-out statistics through a short period of learning process.

In addition to page allocation, we also augment the page replacement policy by using a second-chance algorithm to take wear-out into consideration. More specifically, we use default OS page reclaiming algorithm to select top ten pages that are ready to be replaced. By examining the wear-out level of these pages, the one that with the lowest wear-out level is chosen to be freed or written back to disk if it is dirty. This allows wear-out awareness to be considered even when there isn't plenty of free memory frames available, which may occurs frequently in server system.

5. Experimental Methodology

Table 1. Baseline machine configuration

Parameter	Configuration
Width	4-wide fetch/issue/commit
IQ, ROB, LSQ	64 Issue Queue, 96 ROB entries, 48 LSQ entries
TLB	128 entries(ITLB), 256 entries(DTLB), 4-way, 200 cycle
Branch Pred.	2K entries Gshare, 10-bit global history, 32 entries RAS
I/D L1 Cache	64KB, 4-way, 64 Byte/line, 2 ports, 3 cycle
Integer ALU	4 I-ALU, 2 I-MUL/DIV, 2 Load/Store
FP ALU	2 FP-ALU, 2 FP-MUL/DIV/SQRT
L2 Cache	shared 2MB, 8-way, 128 Byte/line, 12 cycle
Memory	DDR2, 2GB, 8 banks, 667MHz, open page

We simulated a quad-core processor with a shared 2M cache. We assume 3GHz frequency and a 65nm technology with a supply voltage of 1.2V. Table 1 summarizes the baseline machine architecture. We evaluate our techniques by using both SPEC multi-programming workloads and memory-intensive multi-programming workloads shown in Table 2. For SPEC multi-programming workloads, we select benchmarks from SPEC2000 with reference inputs. For each benchmark, we use Simpoint[28] to find representative simulation intervals and select the interval with the highest MPKI. To form multi-programmed workloads, we first categorize benchmarks into: high-miss (MPKI≥20), moderate-miss (5<MPKI≤20) and low-miss (MPKI≤5) categories. In Table 2, the High-, Moderate- and Low- miss workloads (H1-H4, M1-M4, L1-L4) consist of four benchmarks exclusively from each category. The High-Moderate- and Moderate-Low- miss workloads (HM1-HM4, ML1-ML4) are formed by using two benchmarks from each category. To stress the memory system intensively, we use a diverse set of memory-intensive applications from various suites

which feature gigabyte working sets: *Triad* (a streaming benchmark derived from the STREAM suite), *Qsort* (a Unix utility), *Kmean* from MineBench suite and six applications (*BT*, *CG*, *FT*, *LU*, *MG* and *SP*) from NAS Parallel Benchmarks version 3.2 with class "C" input sets. We use four copies of each application to form 4-threaded memory-intensive multiprogramming workloads in Table 2.

Table 2. Workloads

SPEC Multiprogramming Workloads		
High	H1(*mcf, art, apsi, wupwise*)	H3(*fma3d, lucas, apsi, wupwise*)
	H2(*art, art, fma3d, apsi*)	H4(*fma3d, apsi, wupwise, equake*)
Moderate	M1(*swim, swim, gzip, gzip*)	M3(*gzip, gzip, gcc, ammp*)
	M2(*swim, gzip, applu, bzip*)	M4(*gzip, applu, gap, ammp*)
Low	L1(*parser, facerec, vortex, galgel*)	L3(*mgrid, facerec, twolf, mesa*)
	L2(*mgrid, mgrid, facerec, facerec*)	L4(*vpr, vpr, facerec, twolf*)
High - Moderate	HM1(*mcf, art, swim, gzip*)	HM3(*fma3d, lucas, applu, bzip*)
	HM2(*mcf, fma3d, swim, applu*)	HM4(*art, lucas, gzip, bzip*)
Moderate - Low	ML1(*swim, gzip, mgrid, vpr*)	ML3(*applu, bzip, parser, facerec*)
	ML2(*swim, applu, mgrid, parser*)	ML4(*gzip, bzip, vpr, facerec*)
Memory-Intensive Multiprogramming Workloads		
NAS Parallel Bench	4 X *bt*	4 X *lu*
	4 X *cg*	4 X *mg*
	4 X *ft*	4 X *sp*
Triad	4 X *Triad*	
Qsort	4 X *Qsort*	
Kmean	4 X *Kmean*	

Table 3. DRAM/PRAM timing and power parameters[22]

	DRAM	PRAM	
		Average	Aggressive
Timing Parameters (in ns)			
tRCD	15	60	9.9[19]
tRAS	40(2D)/ 27(3D) (1)	NA*	NA*
tCAS	12	12	
tWR	15	150/250	50/120[17]
tRP	15	NA*	NA*
tCMD	6	6	
Power Parameters (1.2V , current in mA)			
Active recharge	135	77(2)	
Precharge power down standby	8	0(3)	
Precharge standby	70	62(4)	
Active power down standby	40	0(3)	
Active standby	75	62(4)	
Read	275	267	
Write	255	325(5)†	263(6)†
Refresh	280	0	

(1)27ns = 40×(1-32%), 32% improvement in tRAS for 3D implementation [3]
(2)77 = standby current (62) + Decoder current (15) obtained by using CACTI 6.0
(3)0 power consumption when master on-off switch for the PRAM is turned off.
(4)62 accounts for the current due to clock signal and peripheral circuit leakage. It is calculated according to a current diagram from Micro technology nodes [22].
(5)325 = DRAM Write Current (255) + PRAM Write Driver (~70mA [24])
(6)263 = DRAM Write Current (255) + [PRAM write current (100uA/bit[19])*16 bit[24]] / PRAM Write Driver Efficiency (20% [24])
† We conservatively use the value of RESET current for both RESET and SET to estimate power consumption, although SET use a smaller current
* doesn't apply to PRAM

To investigate the performance and power impact of the PRAM-based memory architecture, we developed a framework based on a heavily extended Sim-Alpha Simulator [29] integrated with a modified memory model, DRAMsim [23], and the Wattch Power Model [30]. DRAMsim is used for both timing analysis and memory power evaluation, while the power consumption of the cores and caches is calculated by Wattch Power Model. The leakage power of processor cores and caches is assumed to be 15% of their dynamic power based on data reported in [31]. The power overheads of advanced ECC scheme and 16 queues used for OS-level paging are also included in our power estimation. Given the limitation of our simulation infrastructure, we simulate memory-intensive workloads by using memory traces generated from PTLSim/X full system simulator and feeding them into the modified DRAMsim to measure memory performance and power. We assume a disk access latency of 4.2ms [18] on IDE disk. We use conventional DDR2 memory and the configuration is shown in Table 3. For the PRAM-based memory, we perform simulations using two sets of parameter values: average case and aggressive case. For the average case, we find the range of values for a given PRAM parameter through an extensive literature search and use the median value for that parameter. To form the aggressive case, we choose the best value from the published results for each parameter. Our average case is similar to the ITRS 2007 projection for PRAM technologies [17] and our aggressive case mimics PRAM device improvement as technology advances.

As our framework is not a full-system simulator, we developed a memory management model based on source code of several key memory management routines in the Linux Kernel 2.16.15.5 and integrated it into our simulator to mimic the OS page allocation/de-allocation/migration and page table lookup. We have also implemented meta-data structures, such as page descriptors and vm_area structure in our framework. As our simulator doesn't support system calls, we instead trigger our memory management by tracking the memory reference footprints to previously unreferenced pages, similar to page fault expectation handling by OS.

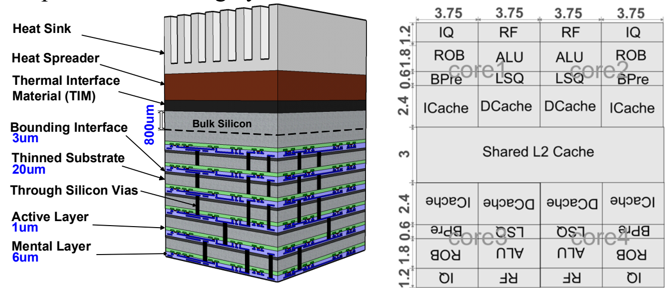


Figure 11. (a) The Cross-section view of the simulated 3D (Left) and (b) the floor-plan of processor layer (unit mm) (Right)

Moreover, we perform detailed thermal analysis to understand the impact of PRAM technology on thermal constrained 3D architecture. The temperature-modeling tool we used is Hotspot4.0 [32]. The floorplan of the core layer is shown in Figure 11(b). The power trace input to Hotspot is generated by Wattch and DRAMsim power model. The layer configuration follows the physical dimension shown in Figure 11 (a). The bulk silicon is modeled as one layer of 800μm in thickness. The thermal parameters of materials used are the same as those in [3] and Hotspot's default heat spreader and heat sink models are set for thermal simulations.

6. Evaluation

In this section, we present our experimental results, including: 1) power savings achieved by PRAM technology; 2) performance impact of PRAM-based memory design; 3) life span extension by using BCH with varied error correction capability and OS wear-leveling; and 4) thermal stress and performance

improvement due to the relieved thermal constraints in 3D design.

6.1 Power Saving of PRAM Technology

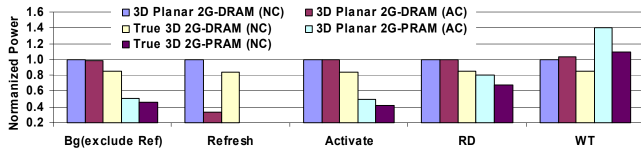


Figure 12. A breakdown of memory power (workload: H4)

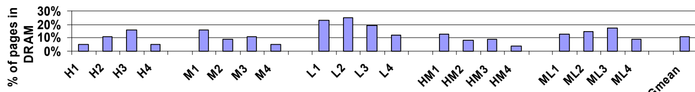


Figure 13. The percentage of pages migrated to DRAM

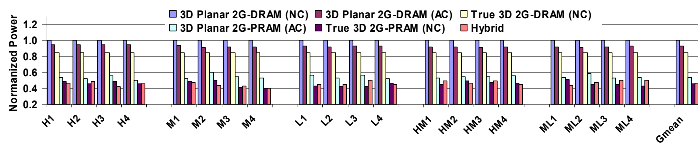


Figure 14. A comparison of overall memory power

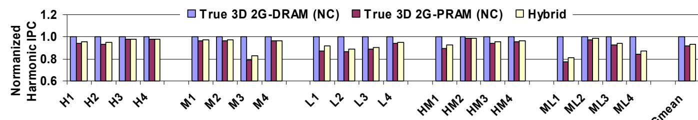


Figure 15. The performance impact of PRAM memory design

We use “3D Planar 2G-DRAM (NC)”, which is a planar on-chip 2G Byte DRAM with normal cooling (NC) solution, as our baseline. We observed that with normal cooling capacity, the peak temperature of all workloads exceeds 85°C, which requires a 32ms refresh interval [22] to ensure reliable operation for DRAM. The “3D Planar 2G-DRAM (AC)” shares the same design configuration as the baseline but with aggressive cooling (AC) capacity so that the on-chip temperature is below 85°C. A 64ms DRAM refresh interval is necessary in this case [22]. The “True 3D 2G-DRAM (NC)” is similar to the “3D Planar 2G-DRAM (NC)”, except that the DRAM is designed in a true-3D fashion. For all PRAM-based designs, the approach of programming current adaptation described in Section 4.1 is used. The “3D Planar 2G-PRAM (AC)” and “True 3D 2G-PRAM (NC)” are planar and true-3D PRAM with aggressive and normal cooling capacity respectively. The “3D Planar 2G-PRAM (NC)” and “True 3D 2G-PRAM (AC)” configurations are not presented because the overall impact of temperature and TSVs on PRAM power is captured by “True 3D 2G-PRAM (NC)” design. The “Hybrid” is our proposed scheme, which uses normal cooling capacity with low cost.

We first present a breakdown of memory power (normalized to the baseline case) on workload H4 to illustrate the impact of different on-chip memory designs. The overall background power includes PRE_PDN, PRE_STBY, ACT_PDN and ACT_STBY and REF [33]. To highlight the impact of elevated temperature (due to 3D die stacking) on DRAM refresh power, we separate the refresh power (Refresh) from the rest of the background power (Bg). The contribution of Bg (exclude Ref), Refresh, Activate, RD and WT to the overall memory power is 21%, 6%, 40%, 16%, 17% respectively in the baseline case. Figure 12 shows that the refresh power reduces dramatically (e.g. by a factor of 3X) when the long refresh interval is used, while the PRAM-based memory does not require refresh due to its non-volatile nature. This non-volatile nature also leads to a lower background power, shown by the first set of bars. Moreover, in contrast to DRAM, the activation of PRAM does not need to charge, discharge and sense all bit lines that connect to a row of cells, which results in a significant active power saving (i.e. more than 50%), as shown by the third set of bars. The last set of bars in Figure 12 shows that the true 3D PRAM implementation reduces memory write power (WT) by 22% compared with a 3D planar PRAM design, which is largely due to the increased number of TSVs used in the design. Although 3D planar PRAM consumes 40% more write power than DRAM, its true-3D design significantly reduces the write power overhead to 10%.

Table 4. The impact of PRAM technology trend on power and performance (Normalized to baseline)

	Power	Performance	Energy
Average	46%	93%	49%
ITRS 2007	46%	94%	49%
Aggressive	34%	98%	35%

Figure 13 shows the percentage of pages that have ever been migrated to DRAM partition for each workload in the hybrid system. Note that page classification is based on modification frequency. Thus, the percentage of pages migrated primarily depend on read/write pattern rather than memory footprint size. As one can see, on average only 11% of pages have ever been migrated to DRAM and we observe 83% of memory accesses are performed on PRAM. This observation shows SPEC multiprogramming workloads sufficiently stress PRAM. Figure 14 shows the overall memory power of different designs across all experimented workloads. For DRAM-based design, the reduction of temperature-induced refresh power from “3D Planar 2G-DRAM (NC)” to “3D Planar 2G-DRAM (AC)” contributes to an average 8% decrease in the total DRAM power. Although the true-3D implemented DRAM achieves an average 15% power reduction due to the reduction in wire-length, the charging and discharging of a large number of bit lines cannot be avoided. The “True 3D 2G-PRAM (NC)” reduces the memory power by 55% compared to the baseline case and achieves the best power efficiency on the majority of all experimented workloads. The significantly reduced background and activation power and the eliminated refresh power are the primary contributors. Compared to the “True 3D 2G-PRAM (NC)”, our proposed scheme (“Hybrid”) also achieves 54% power savings, but introduces a small power overhead for median- and low- miss workloadsdue to the added small DRAM partition. For high-miss workloads, such as HM1-HM4, “Hybrid” achieves slightly more power savings than “True 3D 2G-PRAM (NC)”. This is because our OS page allocation scheme migrates hot-modified pages to DRAM where write operations consume less power than they do on PRAM. Figure 15 presents performance comparison results. Due to the relatively slower read speed of PRAM as compared to “True 3D 2G-DRAM (NC)”, the “True 3D 2G-PRAM (NC)” results in 9% performance degradation on average. By using the proposed OS page allocation and migration, our hybrid design reduces the performance degradation to 6% on average.

Our proposed hybrid design achieves 51% energy reduction, slightly worse than the 54% savings in power, due to the

performance overhead. Furthermore, we show the impact of PRAM technology trend on power, performance and energy in the Table 4. The ITRS 2007 is the proposed design that uses PRAM parameters from International Technology Roadmap for Semiconductors 2007 Edition [17]. As PRAM technology improves, the performance of PRAM will be comparable to DRAM, while the power consumption is expected to be further reduced. This reduction is primary contributed by decreasing the required programming current achieved through device and material improvements.

We further compare our design to several other DRAM-based power management techniques, namely, Smart Refresh [6], Queue-Aware Power Down [34], Min-Rank [35] and Flash-based disk caches [18], which achieve 8.1%, 20% 34% and 41% power saving respectively according to our simulations. Our scheme achieves at lest 13% more power savings. Since our PRAM/DRAM hybrid memory architecture contains a DRAM partition, these proposed techniques can be employed to our design to achieve further power saving. Furthermore, [40] and [42] propose two architectural techniques to reduce PRAM power. [40] proposes to use narrow rows and multiple buffers to improve write coalescing and perform partial writes. [42] takes advantage of redundant bit-writes to eliminate unnecessary writes to PRAM and perform dynamic memory mapping at memory controller to achieve wear-out leveling. These two techniques can be applied to our design to achieve additional 5% power saving.

6.2 Endurance Enhancement

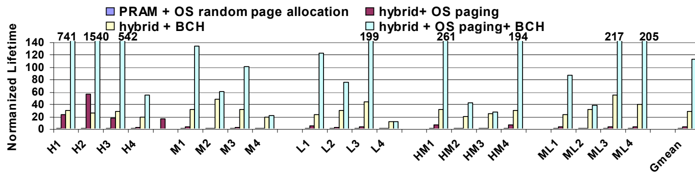


Figure 16. A comparison of endurance improvement

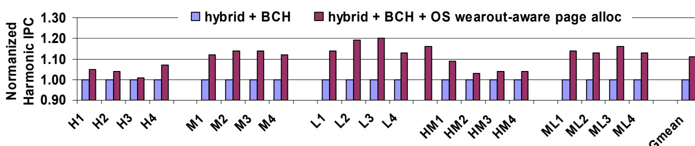


Figure 17. The impact of wear-out aware page allocation on BCH performance

To estimate the wear-out induced failures, we run each workload repeatedly and track the number of writes to each bit. We use 1×10^5 instead of 1×10^8 writes in our lifetime estimation due to the extremely long simulation time to collect 1×10^8 writes to any PRAM cell. With our accelerated estimation method, when a bit has more than 1×10^5 writes on it, we mark it as a failed cell. We define a memory access failure when the number of failed bits accessed in a memory reference is larger than the number of errors that can be corrected. The lifetime of PRAM-based memory is estimated as the number cycles elapsed before the first memory access failure occurs.

Figure 16 shows the lifetime improvement of our proposed technique (hybrid+OS paging+BCH). All results are normalized to the lifetime of the PRAM-only memory system using a random page allocation (PRAM+OS random page allocation). We also present the results of the hybrid PRAM/DRAM architecture that relies on OS paging scheme only (hybrid+OS paging, i.e. no BCH) and BCH only (hybrid+BCH, i.e. no page migration to DRAM partition after initial allocation on PRAM) for reliability enhancement. The "hybrid+OS paging" scheme achieves 4X endurance enhancement (the geometric mean across all simulated workloads). Note that the lifetime is extended by 16X for high miss workloads. This is because the applications in the high-miss category stress memory more intensively than others, leading to a shorter lifetime on the baseline design. We observed that several applications (e.g. *mcf*, *art* and *apsi*) frequently modify a small region of memory. By reallocating those pages from PRAM to DRAM partition, we can substantially reduce the number of writes to PRAM memory and extend the lifetime of the entire memory system. The "hybrid+BCH" scheme allows 28X endurance improvement. Our proposed techniques, which combine BCH and OS paging, extend the life span of the memory system by 114X. The BCH performance penalty increases with the number of errors corrected. As shown in Figure 17, compared with "hybrid+BCH", our proposed techniques (hybrid+BCH+OS wearout aware page alloc) have the ability of achieving wear-leveling, which reduces the number of errors that needs to be corrected per memory access, resulting in 10% performance improvement in geometric mean of all experimented workloads.

6.3 Thermal Relief and Performance Benefit

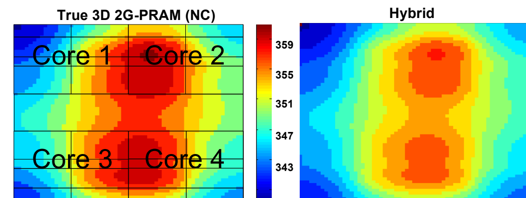


Figure 18. An illustration of on-chip temperature (workload: H3)

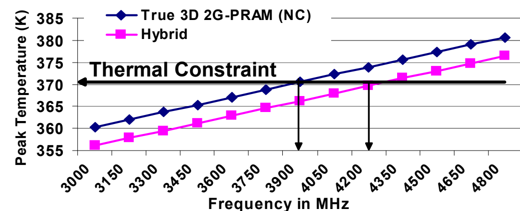


Figure 19. An example of thermal constraint on maximal allowed frequency (workload:H3)

Figure 18 illustrates the temperature distribution (on the workloads H3, i.e. *fma3d+lucas+apsi+wupwise*) of the hottest layer (processor layer) of a 3D DRAM-only system in contrast to that of our proposed design. Figure 20 shows the peak temperature reduction achieved by replacing the conventional DRAM based memory with the proposed PRAM-based design across all experimented workloads. The peak temperature is the sampled maximum temperature across all layers and throughout the entire execution. The temperature reduction ranges form 2.7°C to 4.25°C with an average of 3.3°C (as shown in Figure 20). Arising from reliability concerns, the temperature constraint on a 3D-chip limits the maximum operating frequency of the system. In Figure 19, we present an example of the peak temperature as a function of frequency. Given a thermal constraint, such as 100°C, it places a lower limit on the operating frequency in our proposed hybrid PRAM/DRAM architecture.

Figure 19 shows that the maximum frequency allowed increases by 300 MHz, which translates to an execution speedup of 1.07 on H3 over the DRAM-only design. Figure 20 shows the execution speedup across all workloads. As can be seen, an average speedup of 1.07 is achieved. The results indicate that the lower power of PRAM technology is capable of alleviating the thermal constraint of 3D technology and achieving additional performance gains.

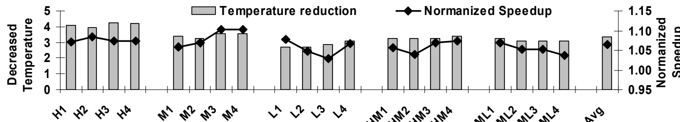


Figure 20. The peak temperature reduction due to the PRAM's power reduction and the execution speedup enabled by the reduced peak temperature

6.4 Memory-Intensive Workload Results

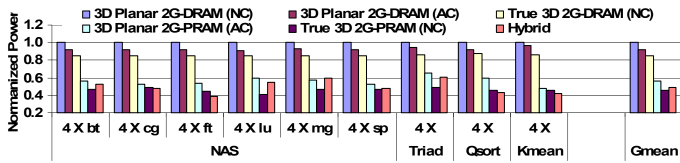


Figure 21. PRAM power impact for memory-intensive workloads

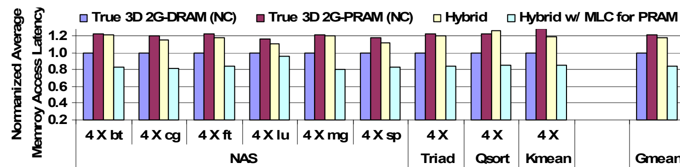


Figure 22. PRAM performance impact for memory-intensive workloads

Figure 21 shows the memory power consumption for each design. Our proposed hybrid design can also effectively reduce memory power by 50% across all memory-intensive workloads with very large memory footprints. Whereas, the average memory access latency for the hybrid design increases by 18% compared to DRAM design, shown in Figure 22. As the performance of memory-intensive application is primarily determined by memory access latency, we use average memory access latency to estimate the performance impact of PRAM due to the limitation of our simulation infrastructure. Although both PRAM and DRAM have approximately the same cell size, adopting the Multi-Level Cell technology [20] substantially increases the density of PRAM, allowing a larger memory capacity with the same die area. "Hybrid w/ MLC for PRAM" in Figure 22 shows the average memory access latency of a hybrid design with MLC technology applied on PRAM partition, resulting in a hybrid memory with 4G effective PRAM and 128MB DRAM. As can been seen, "Hybrid w/ MLC for PRAM" is able to achieve a 28% and 15% reduction in average memory access latency over "Hybrid" and "True 3D 2G-DRAM (NC)" respectively.

7. Related Work

In the past, there have been many studies on DRAM power management at both the architecture and OS levels. At the architecture level, the variation of retention-time among cells is exploited in [6, 7]. The OS-based DRAM power management approaches take advantage of low power modes implemented in today's DRAM chips and maximize its power benefit with optimization techniques in [8, 36]. A common theme of the above approaches is to reduce standby or background power by sending chips to low-power states. However, these techniques are unable to entirely eliminate it because normally they cannot power down all DRAM banks. Our work is unique in that it explores a new type of memory technology, PRAM, which doesn't require data retention power. In addition, our characterizations show that PRAM is more suitable for 3D-stacked memory application compared to others. Similar to prior OS-based approaches, we use a paging-based scheme in our work for the purpose of extending endurance of PRAM. eDRAM has been used for low power on-chip memory. However, eDRAM (like conventional DRAM) requires capacitor to store charges for retaining information. The retention time decreases with technology scaling down. Furthermore, the elevated temperature in 3D exacerbates the leakage issue.

Due to their low power features, nonvolatile memories such as Flash, MRAM and FeRAM are being increasingly used to build storage systems and on-chip structures. [37, 39] evaluates performance and power of 3D stacking cache using MRAM. [38, 41] further evaluates a hybrid cache design, which takes advantage of the best characteristics of each nonvolatile memory technology. Our work focus on exploring PRAM in main memory system design and propose OS support to efficiently address the disadvantages of using PRAM in such application. For main memory design using alternative technologies, [18] first proposed integrating a Flash-based disk cache into memory hierarchy of server platforms to save memory power. The PRAM technology introduced by our work is superior to Flash in terms of performance and lifespan. To our knowledge, [40, 42] are the first two architecture level studies on using PRAM to implement main memory (discussed in section 6.1). Our study differs from these works in two ways: 1) we evaluate PRAM in the context of true 3D-stacked memory architectures, while their designs are limited only to the planar chips; and 2) we propose both architecture and OS-level endurance improvement techniques, whereas their schemes are built only on the architecture layer. Techniques proposed in [40, 42] are orthogonal to our design. By incorporating their schemes into our design, an additional 5% power saving is achieved (discussed in section 6.1).

8. Conclusions

DRAM has been used in computer system main memory design for decades. However, DRAM is facing scalability issues and the power consumption of DRAM-based memory is increasing rapidly. Emerging phase change memory technologies exhibit superior scalability and power efficiency. However, compared to DRAM, PRAM has longer read/write access latencies and is subject to wear-out. In this study, we propose a power-efficient, high-speed and durable memory system design. Our techniques leverage 3D vertical die stacking to reduce PRAM memory access latency. To further absorb the much worse PRAM write delay, our approach uses a hybrid PRAM/DRAM memory architecture and leverages the OS to dynamically allocate physical pages based on workload characteristics. To enhance the endurance of our design, we adopt variable strength ECC and use reliability-aware OS paging to achieve wear-leveling.

We show that the two emerging technologies can fit well with each other: the 3D TSVs provide significantly reduced wire length and resistance, which reduce both PRAM access delay and power consumption. On the other hand, the high temperature driven operations and low stand-by power make PRAM more thermal friendly to 3D die stacked memory architecture. Experimental results show that our design achieves a 54% power savings with 6% performance loss compared to 3D 2G-DRAM. This power savings reduces 3D thermal constraints and consequently achieves an average 1.07 speedup across all experimented workloads. We also show that the proposed ECC and OS support for PRAM endurance optimization is capable of extending the lifespan by 114X.

9. Acknowledgement

This work is supported in part by NSF grants 0937869, 0916384, 0845721(CAREER), 0834288, 0811611, 0720476, by SRC grants 2008-HJ-1798, 2007-RJ-1651G, by Microsoft Research Trustworthy Computing, Safe and Scalable Multi-core Computing Awards, by NASA/Florida Space Grant Consortium FSREGP Award 16296041-Y4, and by three IBM Faculty Awards.

References

[1]K. Kim, Technology for sub-50nm DRAM and NAND Flash Manufacturing, IEDM, 2005.

[2]C. Lefurgy, K. Rajamani, F. L. Rawson III, W. Felter, M. Kistler, and T. W. Keller, Energy Management for Commercial Servers, IEEE Computer, Vol. 36, 2003.

[3]G.H. Loh, 3D-Stacked Memory Architectures for Multi-Core Processors, ISCA, 2008.

[4]B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G.H. Loh, D. McCaule, P. Morrow, D.W. Nelson , D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen and C. Webb, Die-Stacking (3D) Microarchitecture, MICRO, 2006.

[5]Tezzaron Semiconductors, Leo FaStack 1Gb DDR SDRAM Datasheet.

[6]M. Ghosh and H. S. Lee, Smart Refresh: An Enhanced Memory Controller Design for Reducing Energy in Conventional and 3D Die-Stacked DRAMs, MICRO, 2007.

[7]R.K. Venkatesan, S. Herr and E. Rotenberg, Retention-Aware Placement in DRAM (RAPID): Software Methods for Quasi-Non- Volatile DRAM, HPCA, 2006.

[8]A.R. Lebeck, X. Fan, H. Zeng and C. Ellis, Power-aware Page Allocation, ASPLOS, 2000.

[9]C. Lam, Cell Design Considerations for Phase Change Memory as a Universal Memory, VLSI-TSA, 2008.

[10]M. Gill, T. Lowrey and J. Park, Ovonic Unified Memory - A High-Performance Non-volatile Memory Technology for Stand-alone Memory and Embedded Applications, ISSCC, 2002.

[11]L. Burcin, S. Ramaswamy, K. K. Hunt, J. D. Maimon, T. J. Conway, B. Li, A. Bumgarner, G. F. Michael and J. Rodgers, A 4-Mbit Non-Volatile Chalcogenide Random Access Memory, IEEE Aerospace Conference, 2005.

[12]Intel, STMicroelectronics Deliver Industry's First Phase Change Memory Prototypes. http://www.intel.com/pressroom/archive/releases/20080206corp.htm

[13]G. Atwood and R. Bez, Current Status of the Phase Change Memory and its Future, IEDM, 2003.

[14]D. Salamon and B. F. Cockburn, An Electrical Simulation Model for the Chalcogenide Phase-change Memory Cell, MTDT, 2003.

[15]F. Bedeschi, R. Fackenthal, C. Resta, E.M. Donze, M. Jagasivamani, E. Buda, F. Pellizzer, D. Chow, A. Cabrini, G.M.A. Calvi, R. Faravelli, A. Fantini, G. Torelli, M. Duane, R. Gastaldi and G. Casagrande, A Multi-Level-Cell Bipolar-Selected Phase-Change Memory, ISSCC, 2008.

[16] D.H. Kang, D.H. Ahn, K.B. Kim, J.F. Webb and K.W. Yi, One-Dimensional Heat Conduction Model for an Electrical Phase Change Random Access Memory Device with an 8f2 Memory Cell (f=0:15¹m), Journal of Applied Physics, 94(5), 2003.

[17]ITRS 2007, Emerging Research Device. http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_ERD.pdf

[18]T. Kgil, D. Roberts and T. Mudge, Improving NAND Flash Based Disk Caches, ISCA, 2008.

[19]S. Hanzawa, N. Kitai, K. Osada, A. Kotabe, Y. Matsui, N. Matsuzaki, N. Takaura, M. Moniwa and T. Kawahara, A 512kB Embedded Phase Change Memory with 416kB/s Write Throughput at 100uA Cell Write Current, ISSCC, 2007

[20]G. Loi, B. Agarwal, N. Srivastava, S. Lin and T. Sherwood, A Thermally-Aware Performance Analysis of Vertically Integrated (3-D) Processor-Memory Hierarchy, DAC, 2006.

[21]S. Lai and T. Lowrey, OUM – A 180 nm Nonvolatile Memory Cell Element Technology for Stand-alone and Embedded Applications, IEDM, 2001.

[22]Micron DDR2 SDRAM 2Gb and 4Gb data sheet. http://download.micron.com/pdf/datasheets/

[23]D. Wang, B. Ganesh, N. Tuaycharoen, K. Baynes, A. Jaleel and B. Jacob, DRAMsim: A Memory-System Simulator, SIGARCH Computer Architecture News, Vol. 33, 2005.

[24]S. Kang, W.Y. Cho, B.H. Cho, K.J. Lee, C.S. Lee, H.R. Oh, B.G. Choi, Q. Wang, H.J. Kim, M.H. Park, Y.H. Ro, S. Kim, C.D. Ha, K.S. Kim, Y.R. Kim, D.E. Kim, C.K. Kwak, H.G. Byun, G. Jeong, H. Jeong, K. Kim and Y. Shin, A 0.1-μm 1.8-V 256-Mb Phase-Change Random Access Memory (PRAM) with 66-MHz Synchronous Burst-Read Operation, IEEE Journal of Solid-State Circuits, Vol. 42, 2007.

[25]R. Efraim, H. Jim, A. Cohen and H. Cain, Temperature measurement in the Intel® CoreTM DuoProcessor.

[26]Y. Zhou, P. M. Chen, and K. Li, The Multi-Queue Replacement Algorithm for Second-Level Buffer Caches, USENIX Annual Technical Conference, 2001.

[27]D. Strukov, The Area and Latency Tradeoffs of Binary Bit-Parallel BCH Decoders for Prospective Nanoelectronic Memories, In Proceedings of the 40th Asilomar Conference on Signals, Systems and Computers, 2006.

[28]T. Sherwood, E. Perelman, G. Hamerly, and B. Calder, Automatically Characterizing Large Scale Program Behavior, ASPLOS, 2002.

[29]R. Desikan, D. Burger and S. W. Keckler, Measuring Experimental Error in Microprocessor Simulation, ISCA, 2001.

[30]D. Brooks, V. Tiwari and Margaret Martonosi, Wattch: A Framework for Architectural-Level Power Analysis and Optimizations, ISCA, 2000.

[31]G. Sery, S. Borkar, and V. De, Life is CMOS: Why Chase the Life After?, DAC, 2002.

[32]K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan and D. Tarjan, Temperature-Aware Microarchitecture, ISCA, 2003.

[33]Micron DDR2 SDRAM system-Power Calculator.

[34]Ibrahim Hur and Calvin Lin, A Comprehensive Approach to DRAM Power Management , HPCA, 2008.

[35]Hongzhong Zheng, Jiang Lin, Zhao Zhang, Eugene Gorbatov, Howard David and Zhichun Zhu, Mini-Rank: Adaptive DRAM Architecture for Improving Memory Power Efficiency, MICRO, 2008.

[36]V. Delaluz, A. Sivasubramaniam, M. Kandemir, N. Vijaykrishnan and M. J. Irwin, Scheduler-based DRAM Energy Power Management, DAC, 2002.

[37]X. Dong, X. Wu, G. Sun, Y. Xie, H. Li and Y. Chen, Circuit and Microarchitecture Evaluation of 3D Stacking Magnetic RAM (MRAM) as a Universal Memory Replacement, DAC, 2008.

[38]P. Mangalagiri, A. Yanamandra, Y. Xie, N. Vijaykrishnan, M.J. Irwin, K. Sarpatwari and O.O.A. Karim, A Low-Power Phase Change Memory Based Hybrid Cache Architecture, GLSVLSI, 2008.

[39]G. Sun, X Dong, Y Xie, J. Li and Y. Chen, A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs, HPCA, 2009.

[40]B. C. Lee, E. Ipek, O. Mutlu and D. Burger, Architecting Phase Change Memory as a Scalable DRAM Alternative, ISCA, 2009.

[41]X. Wu, J. Li, L. Zhang, E. Speight, R. Rajamony and Y. Xie, Hybrid Cache Architecture with Disparate Memory Technologies, ISCA, 2009.

[42]P. Zhou, B. Zhao, J. Yang and Y. Zhang, A Durable and Energy Efficient Main Memory Using Phase Change Memory Technology, ISCA, 2009.

[43] I. V. Karpov and S. A. Kostylev, SET to RESET Programming in Phase Change Memories, IEEE Electron Device Letters, Vol. 27, 2006