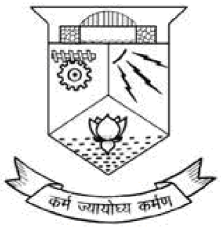
A Phase Change Memory as a Secure Main Memory

A Seminar Report
Submitted in partial fulfilment of
the requirements for the award of the degree of

Master of Technology
in
Computer Science and Engineering

by

Abdul Nassar A A
M105101



Department of Computer Science & Engineering
College of Engineering Trivandrum
Kerala - 695016
2010-11

College of Engineering Trivandrum
Department of Computer Science & Engineering

Certified that this Seminar Report entitled

A Phase Change Memory as a Secure Main Memory

is a bonafide record of the seminar presented by

Abdul Nassar A A
M105101

in partial fulfilment of
the requirements for the award of the degree of
Master of Technology
in
Computer Science & Engineering

Mr. Rameez Mohammed A
Guide
Lecturer
Dept.of Computer Science & Engineering

Dr. M.S Rajasree
Professor and
Head
Dept.of Computer Science & Engineering

Acknowledgement

I would like to express my sincere gratitude and heartful indebtedness to my guide **Mr Rameez**, Lecturer, Department of Computer Science And Engineering for his valuable guidance and encouragement in pursuing this seminar.

I am thankful to **Dr. M S Rajasree**, Head of the Department, **Shine S**, P.G Coordinator, **Shreelekshmi R**, Asst. Professor and **Anwar A**, Asst. Professor and project coordinator Department of Computer Science and Engineering for their help and support.

I also acknowledge my gratitude to other members of faculty in the Department of Computer Science And Engineering and all my friends for their whole hearted cooperation and encouragement.

Above all I am thankful to the God Almighty.

Abdul Nassar A A

Contents

Abstract

Phase Change Memory (PCM) technology appears as more scalable than DRAM technology. As PCM exhibits access time slightly longer but in the same range as DRAMs, several recent studies have proposed to use PCMs for designing main memory systems. Unfortunately PCM technology suffers from a limited write endurance; typically each memory cell can be only be written a large but still limited number of times (10^7 to 10^9 writes are reported for current technology). Till now, research proposals have essentially focused their attention on designing memory systems that will survive to the average behavior of conventional applications. However PCM memory systems should be designed to survive worst-case applications, i.e., malicious attacks targeting the physical destruction of the memory through overwriting a limited number of memory cells.This seminar paper proposes the design of a secure PCM-based main memory that would by construction survive to overwrite attacks.

1 Introduction

Phase Change Memory (PCM) technology appears as a promising technology for designing main memory in future computer systems. PCM presents advantages over DRAMs in terms of static energy consumption as well as integration scalability for future technologies generations; for instance, anticipates durance, i.e., a PCM memory cell can only support a limited number of writes and exceeding this limit might impair its correct functioning. The reported write endurances for PCM memory vary between 10^7 and 10^9 writes on a single cell. Such a limited endurance has been recognized as issue for the design of PCM-based main memory systems. Several propositions have been made to allow a PCM main memory to survive the anticipated lifetime of a computer system ,i.e., 10 to 20 years, in the context of general applications. At the exception of these studies completely ignore the security breach that the limited write endurance of PCM components would create in a main memory. PCM components for main memory would create a main memory through a very simple program overwriting the same memory cells again and again. The potential attack is particularly simple to mount. It can be run by any user without any execution privilege. It is seen that their Region Based Start Gap scheme would survive a few months to a naive overwrite attack consisting in constantly overwriting the same physical memory address. However, the Region Based Start Gap (RBSG) scheme considered in would not survive more than a few days to a slightly more complex attack based on the birthday paradox. More over the RBSG scheme a RBSG scheme supporting page mode would even be less endurant to an overwrite attack.This paper proposes the design of a secure main PCM memory. In order to prevent a malicious user to overwrite some memory cells, the physical memory address (PA) manipulated by the computer system is not the same as the PCM memory address (PCMA) . PCMA is made invisible from the rest of the computer system. The PCM memory controller is in charge of the PA-to-PCMA translation. Hiding PCMA alone does not prevent a malicious user to blindly overwrite some PCM memory blocks. Therefore in the secure PCM-based main memory, PA-to-PCMA translation is continuously modified through a random process. This prevents a malicious user to overwrite some PCM memory words, it also uniformizes the write pressure on the overall memory for every possible type of workloads. For implementing the PA-to-PCMA translation, the PCM memory controller implements a translation table and needs an efficient random number generator. As an example, for write endurance in the 32M range, our study shows that associating a single translation table entry with a 4K memory blocks region should be sufficient. Provided one extra write per 8 program generated writes, our scheme would resist an overwrite attack for 62 % of the expected total memory lifetime. However, endurance to overwrite attacks is obtained at the cost of some performance decrease on applications limited by the main memory bandwidth since one extra block read and one extra memory block write is generated every eight memory block writes. The security also limits the number of possible program-generated writes on the memory to 8/9 th of the total number of possible writes on the memory.

2 PCM Technology

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

2.1 History and background

In the 1950s and 1960s, Dr. Stanford Ovshinsky began researching the properties of a class of amorphous materials. Amorphous materials are those materials that do not exhibit a definite, ordered crystalline structure. By 1968, he reported that certain glasses exhibited a reversible change in resistivity upon a change in phase. In 1969, he also reported a corresponding change in reflectivity that could be induced by laser in an optical storage media. By 1970, the company he and his wife Dr. Iris Ovshinsky founded, Energy Conversion Devices (ECD), published the results of a collaboration with Intel's Gordon Moore. The September 28th, 1970 issue of Electronics2 featured the world's first Phase Change Memory, a 256 bit semiconductor device. Memory (PCM) is a term used to describe a class of non-volatile memory devices that employ a reversible phase change in materials to store information. Matter can exist in various phases such as solid, liquid, gas, condensate and plasma. PCM exploits differences in the electrical resistivity of a material in different phases. This paper describes the basic technology and capabilities of PCM. History and background In the 1950s and 1960s, Dr. Stanford Ovshinsky began researching the properties of a class of amorphous materials. Amorphous materials are those materials that do not exhibit a definite, ordered crystalline structure. By 1968, he reported1 that certain glasses exhibited a reversible change in resistivity upon a change in phase. In 1x969, he also reported a corresponding change in reflectivity that could be induced by laser in an optical storage media. By 1970, the company he and his wife Dr. Iris Ovshinsky founded, Energy Conversion Devices (ECD), published the results of a collaboration with Intel's Gordon Moore. The September 28th, 1970 issue of Electronics2 featured the world's first Phase Change Memory, a 256 bit semiconductor device. Nearly 30 years later, ECD formed a new subsidiary, Ovonyx, a joint venture between ECD and Tyler Lowery, the former CTO, COO and Vice-Chairman of Micron Technology. In February 2000, Intel and Ovonyx announced a collaboration and licensing agreement that spawned the modern age of research and development in PCM. In December of 2000, STMicroelectronics ("ST") and Ovonyx also began a collaboration. By 2003, the three companies had joined forces to accelerate progress on the technology by avoiding duplication in basic, pre-competitive R & D and through expanding the research scope. In 2005, ST and Intel agreed to co-develop a 90 nm PCM technology. In 2007, ST and Intel announced their intention to form a new flash company called Numonyx. In the intervening years since that first work in 1970, much progress has been made in semiconductor manufacturing technology, enabling the practical development of PCM. Also during that time period, phase change materials were perfected for high volume use in rewritable CDs and DVDs. Today, most DVD-RAMs available today use the exact same PCM attributes and capabilities Phase Change Memory blends the attributes commonly associated with NOR-type flash, memory NAND-type flash memory, and RAM or EEpROM.

2.2 Memory cell

As shown in Figure 1, the PCM storage element is comprised of two metal electrodes separated by a resistive heater and a chalcogenide, the phase change material. Ge2Sb2Te5 (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

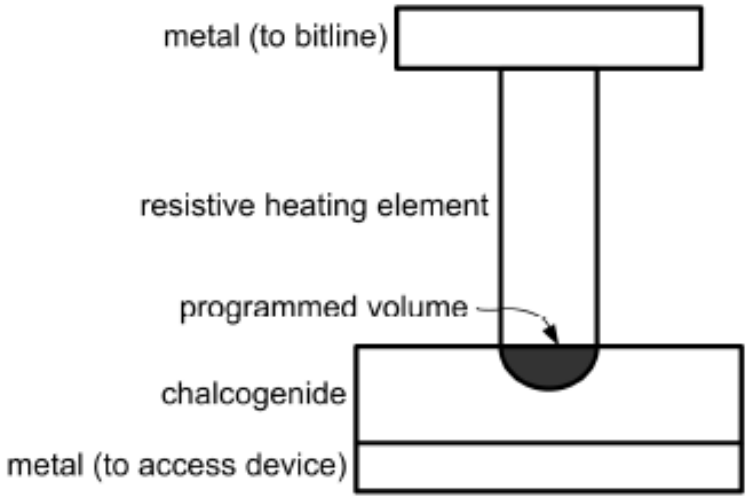


Figure 1: PCM Memory Elements.

while GS may offer faster phase changes. As shown in Figure 2, 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. BJTs are faster and expected to scale more robustly without this vulnerability. Diodes occupy smaller areas and potentially enable greater cell densities, but require higher operating voltages. 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. The amplitude and width of the injected current pulse determine the programmed

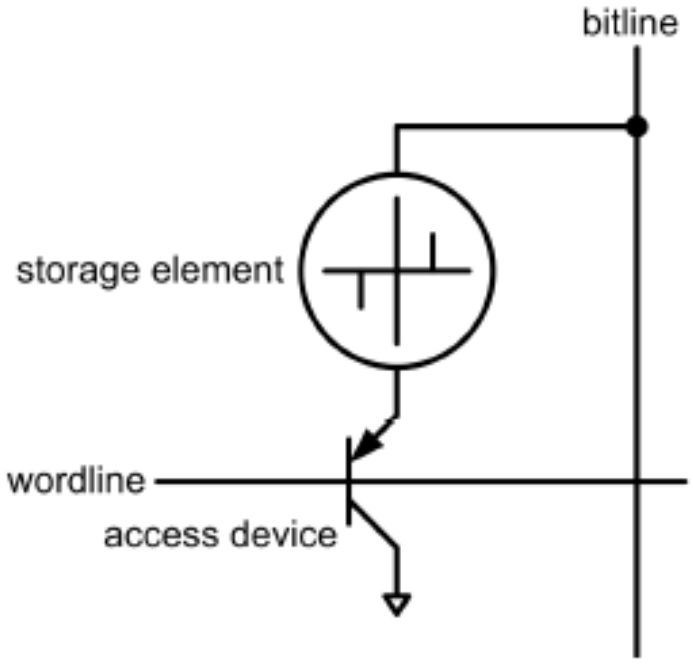


Figure 2: PCM Memory Cell.

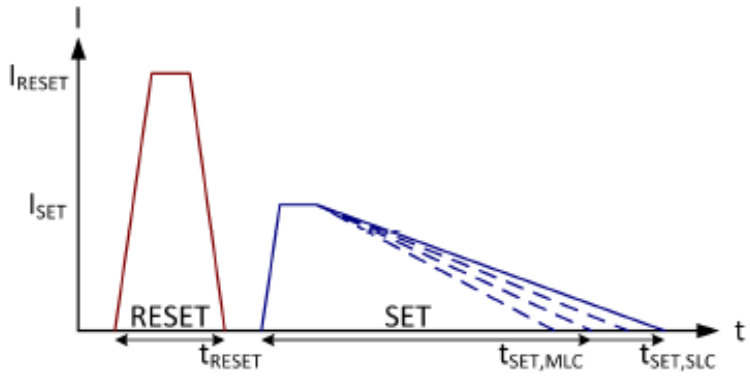


Figure 3: PCM Memory Elements.

state as shown in Figure 3.

2.3 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 chalcogenid 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 bit-line is precharged to the read voltage. If a selected cell is in a crystalline state, the bit line is discharged with current flowing through the storage element and access transistor. Otherwise, the cell is in an amorphous state, preventing or limiting bit line current. Cells that store multiple resistance levels might be implemented by leveraging intermediate states, in which the chalcogenide is partially crystalline and partially amorphous. 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.

2.4 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 $32nm$ With wear reduction and leveling techniques, PCM write limits may not be exposed to the system during a memory's lifetime.

3 PCM attributes

This new class of non-volatile memory brings together the best attributes of NOR, NAND and RAM.

1. Bit-alterable:
   Like RAM or EEpROM, PCM is bit alterable. Flash technology requires a separate erase step in order to change information. Information stored in bit-alterable memory can be switched from a one to zero or zero to a one without a separate erase step.

2. Non-volatile:
   Like NOR flash and NAND flash, PCM is nonvolatile. RAM, of course, requires a constant power supply, such as a battery backup system, to retain information. DRAM technologies also suffer from susceptibility to so-called "soft errors" or random bit corruption caused by alpha particles or cosmic radiation. Early testing results conducted by Intel on multimegabit PCM arrays for long term data retention show excellent results.

3. Read speed:
   Like RAM and NOR-type flash, the technology features fast random access times. This enables the execution of code directly from the memory, without an intermediate copy to RAM. The read latency of PCM is comparable to single bit per cell NOR flash, while the read bandwidth can match DRAM. In contrast, NAND flash suffers from long random access times on the order of 10s of microseconds that prevent direct code execution.

4. Write/erase speed:
   PCM is capable of achieving write speeds like NAND, but with lower latency and with no separate erase step required. NOR flash features moderate write speeds but long erase times. As with RAM, no separate erase step is required with PCM, but the write speed (bandwidth and latency) does not match the capability of RAM today. The capability of PCM is expected, however, to improve with each process erase speed PCM is capable of achieving write speeds like NAND, but with lower latency and with no separate erase step required. NOR flash features moderate write speeds but long erase times. As with RAM, no separate erase step is required with PCM, but the write speed (bandwidth and latency) does not match the capability of RAM today. The capability of PCM is expected, however, however, to improve with each process generation as the PCM cell area decreases. Scaling is the fifth area where PCM will offer a difference. Both NOR and NAND rely on memory structures which are difficult to shrink at small lithos. This is due to gate thickness remaining constant and the need for operation voltage of more than 10V while the operation of CMOS logic has been scaled to 1V or even less. This scaling effect is often referred to as Moore's Law, where memory densities double with each smaller generation. With PCM, as the memory cell shrinks, the volume of GST material shrinks as well, providing a truly scalable solution. PCM employs a reversible phase change phenomenon to store information through a resistance change in different phases of a material. Advances in memory technology and pioneering work conducted by Numonyx has moved the technology to the forefront of the memory industry R & D activity. PCM offers a combination of some of the best attributes of NOR flash, NAND flash, EEpROM and RAM in a single memory device. These capabilities uniquely combined with the potential for lower memory subsystem costs could potentially create new applications and memory architectures in a wide range of systems.

4 A secure PCM based main memory

4.1 Security principles

4.1.1 Invisible PA-to-PCMA translation is required

If a malicious attacker knows the PA-to-PCMA translation then for a given PCM memory block B, he/she is able to figure out the address of the physical memory block that is mapped on B. If the PA-to-PCMA translation is made invisible from the outside of the PCM memory then the attacker can not retrieve the address of the physical memory block mapped on a given memory block.

4.1.2 PA-to-PCMA translations must dynamically

Our analysis of the RBSG scheme has shown that, in order to resist a birthday paradox attack, the PA to PCMA translation of any physical block B has to be modified with a frequency largely higher than one time every $Wmax$ possible writes on B. The PA-to-PCMA translation changes should be completely unpredictable from the outside of the PCM memory; in particular there should be no restrictions on the new translation.

4.2 Principles of a practical secure PCM-based main memory

4.2.1 PA-to-PCMA translation

In the secure PCM-based main memory the PA-to-PCMA translation is performed by the PCM memory controller through the use of a translation table. For a physical memory block B, the address of the corresponding PCM block is computed from an entry read in the translation table and the address B. The PA to PCMA translation must perform a one-to-one address translation from the physical address space to PCM address space. The simplest mapping would be to associate a translation table entry with each physical memory block and ensuring that the translation is a one-to-one block mapping. Such one-to-one block mapping appears as unpractical so associate a single translation table entry with a region of R contiguous memory blocks; for instance if 4K contiguous memory blocks are mapped by a single entry, 64K entries are sufficient to map 16 GBytes. Such a single translation for a large region was already proposed in the context of wear leveling for conventional applications

4.2.2 PA-to-PCMA region address translation

Initializing at boot time the translation table T with a one-to-one region mapping is unpractical. Instead of such an initialization,assume that at initialization time, the translation table T is initialized with only zeros, but that some computation is performed at runtime in addition to the read of the translation table. If memory regions are numbered from 0 to $N-1$, the translation is performed as follows: region B in physical memory is mapped onto region $(T(B).addressxorBxorRinit)$, where $Rinit$ is a random number generated at initialization time. Note that, at initialization time,$T(B)$ is null and that the PA-to-PCMA region address translation is a one-to-one mapping

4.2.3 PA-to-PCMA region displacement translation

The use of a single entry to map a complete region of the physical memory could lead to a possible overwrite attack on trying to write a specific block in all the regions, for instance the first block. In order to avoid such an attack, the displacement in the region is also translated.

Physical memory block X in region B is mapped onto block ($T(B).dispxorXxorDinit$) in region ($T(B).addressxorBxorR_{init}$). As R_{init}, D_{init} is a random number generated at initialization time.

4.2.4 Dynamically changing PA-to-PCMA translation

In order to avoid blind overwrite attacks, the PA-to- PCMA translation must be continuously modified. More precisely, only writes represent an issue. Therefore, the PA-to-PCMA translation modification is triggered randomly and only on memory writes. This random triggering is particularly important: As an example, if the PA-to-PCMA translation change occurs periodically on writes, for instance every 10 writes, an attacker could repeat the sequence of nine consecutive writes on physical block B, one write on physical block C. Physical block C moves in the PCM memory, but block B remains on the same PCM memory location that can be easily overwritten.

4.2.5 How to modify PA-to-PCMA translation

Modifying the PA-to-PCMA translation for a physical region B is implementing through swapping the translations for two physical regions. This guarantees that the PA-to-PCMA translation remains a one-to-one mapping. The region swapping induces the modification of two entries in the translation table. A random physical regionB' is chosen and the PA-to-PCMA translations of$BandB'$ are exchanged, i.e.,

$$T(B).addr := old(T(B').addr)\ xor\ B'\ xor\ B$$

and

$$T(B').addr := old(T(B).addr)\ xor\ B'\ xor\ B$$

4.2.5.1 At the same time, displacement translations inside blocks B and B' are also modified,

$$T(B).disp := old(T(B)).disp\ xor\ RAND$$

and

$$T(B').disp := old(T(B')).disp\ xor\ RAND$$

where $RAND$ is randomly selected. The two memory regions in the PCM memory have to read and swapped accordingly. Randomly swapping memory regions has been already proposed in the context of wear leveling for flash memories . Frequency of PA-to-PCMA translation modifications The cost of a PA-to-PCMA translation modification is proportional to the size R of a region in the memory. The two swapped regions have to be read and rewritten, i.e., a PA-to- PCMA translation modification induces $2R$ memory block reads and $2R$ memory block writes. Therefore, the frequency of the address translation modification should be chosen in order to maintain the total overhead to a reasonable level.This study arbitrarily estimate that inducing in average one extra write on the PCM memory per 8 effective writes IEEE Computer Architecture Letters Phase Change Memory (PCM) is a term used to describe a class of non-volatile memory devices that employ a reversible phase change in materials to store information. Matter can exist in various phases such as solid, liquid, gas, condensate and plasma. PCM exploits differences in the electrical resistivity of a material in different phases. This paper describes the basic technology and capabilities of PCM would be acceptable. That is, in average one out of $16R$ physical memory block writes can trigger a PA-to-PCMA address translation modification. Therefore on receiving a write on a physical memory block, the modification of its PA-to-PCMA translation is randomly triggered with probability $116R$.

5 PCM memory controller

The design of a secure PCM-based main memory leads to several constraints inside the memory controller.

5.1 Write endurance and region size

The principles above lead to the design of a PCM-based main memory on which an overwrite attack would only be able to consecutively write the same memory block in average 16R times before the physical block is moved in another PCM memory block. In practice, a write attack could succeed in significantly reducing the lifetime of the memory was done. if $16R$ is not small with respect to the write endurance of the cells.Simulations of an overwrite attack on a 16 GBytes PCM memory i.e., 226 256-byte blocks. Regions of respectively 64K and 4K memory blocks were considered. If the memory features a write endurance of $Wmax = 2E$ writes, then the theoretical write endurance of a uniformly accessed PCM memory is 226+E. With a write endurance of only 8 Megawrites (223) per cell, using 64K memory blocks per region is not an option: some memory blocks would be destroyed by a brute force overwrite attack in less than a billion (230) writes. With 4K memory blocks regions, the PCM memory would be able to support an attack consisting of up to 38 % of the theoretical 249 writes. If the write endurance is 32 Mega writes per cell (225) then these respective ratios become 7.4 % for 64K memory blocks regions, and 62 % for 4K memory blocks region for a theoretical maximum of 88.88 % since in average one extra block write is triggered for 8 physical memory writes. If the write endurance is 256 Mega writes per cell (230) then these ratios become 52 % and 79% for 64K and 4K memory block regions respectively. Therefore, if the technology is able to ensure write endurance in the hundreds of millions range then even very large regions could be considered for PA-to-PCM translations.

5.2 Memory controller constraints

The secure PCM main memory would need to integrate extra hardware in the memory controller o implement the secure PA-to-PCMA translation. Memory storage volume The storage volume of the PCM memory controller is a major issue. The main component is the translation table that features an entry per memory region. For a 16 GB memory, the use of regions of 4K 256-bytes blocks would lead to 64K entries, each entry featuring the address of region (14 bits) and a displacement in the region (12 bits) i.e. a total of 26 bits. The total storage cost of the translation table would be 52Kbytes. If a 64K blocks region was used then the size of the PCM translation table would only be 3.25 Kbytes.

5.3 Swapping memory regions logic

The memory controller has to handle the important function of swapping two memory regions on a PA-to- PCMA translation change. This induces a large number of memory reads and writes. An atomic swap of the two memory regions would stop the normal read and write accesses by the computer system. This would be unacceptable. Therefore the memory controller must feature logic to interleave blocks swapping with the normal flow of reads and writes from the computer system. The logic must be able to handle the case where a normal write is overwriting a block belonging to one of the memory regions being swapped. Moreover this normal flow of writes may randomly trigger new region swaps; the memory controller should be able to buffer these swaps. The priority on writes must be dynamically adapted in order to maintain a limited number of regions waiting for swaps, for example at most 8 swaps. As an example,The policy of

randomly splitting the write priority to 1/4th for region swapping and $3/4th$ for normal write was tested. Flow when less than 4 region swaps are waiting and one half for region swapping and one half for normal write flow when 4 or more region swaps. On an experiment on 240 writes and assuming a continuous saturated write flow from the computer system, there was never more than 8 waiting region swaps.

5.4 Extra PA-to-PCMA translation latency

The extra access time to main memory due to PA to- PCMA translation is essentially due to the read of the PA-to-PCMA translation table. This table will be implemented as a SRAM table in the memory controller. For a 16 GB memory using 4K 256-byte blocks region, the read access time of a 52Kbytes SRAM memory would in the range of 2-4 processor cycles and would be marginal compared with the overall main memory access time.

5.5 The random number generator

The secure PCM-based main memory will be able to resist to an overwrite attack if no one is able to follow or reconstruct the PA-to-PCMA translation process. Our proposal heavily relies on a random number generator. The security of our proposal also depends of the security of this random number generator. One can remark that the output of the random number generation used in our memory controller cannot be directly observed from the outside of the PCM memory. Therefore different possible schemes could be implemented ranging from a true hardware random generator to a simpler algorithmic pseudo-random number generator personalized with a huge key at manufacturing time.

5.6 Write Endurance for conventional applications

The security of our proposed secure PCM main memory is ensured by

1. the invisibility of the access on the PCM IEEE Computer Architecture Letters memory from outside the memory, and

2. the random distribution of the accesses through the PA-to-PCMA address translation change: under an overwrite attack, a given physical block will change PA-to-PCMA address translation after a number of writes that is comparatively small with the write endurance of the PCM memory cell. For a conventional application, for a given physical memory region, the frequency of the PA-to-PCMA address translation changes measured in writes on the region is the same as under an overwrite attack, but the writes are distributed over the whole region. Therefore, for each block, the number of writes is very small compared with the endurance of the PCM memory cell. The write endurance of the overall memory system is therefore very close to its maximum theoretical endurance

6 Future trends in PCM technology

6.1 Economic feasibility of PCM main memory

If within a few years, the write endurance per cell on PCM components reach 256 M writes then it would become feasible to build 16 GBytes (or larger) memory using comparatively very small PA-to-PCMA translation table (for instance, using 64K blocks memory regions): as mentioned above, the secure PCM memory would be able to survive to an overwrite attack at a full 4 GBytes/s write bandwidth for 52

6.2 Page mode is compatible with security

If PCM memories are used as main memory then a page mode would be interesting as on current DRAM to limit the access latency and increase bandwidth when the memory read requests exhibit high spatial locality. Our PA-to-PCMA translation scheme is compatible with such page mode since regions are large enough to accommodate large page, even split across several PCM components.

6.3 Limiting extra write traffic overhead

In this study, the overwrite traffic associated with PA-to-PCMA translation modification could be as large as 1extra PCM block write per 8 physical memory block writes. This overhead can be reduced by decreasing the probability of triggering a PA-to-PCMA translation. This would reduce the total endurance of the system to the overwrite attack, but may still remain acceptable if the cell write endurance is large. For instance, for a 16GBytes memory, if the extra PCM write traffic is limited to 1 extra PCM block write per 32 physical memory writes i.e., a 3.1 % write overhead, the cell endurance is 256 Megawrites and 64K blocks memory regions are used then the secure PCM memory still survives a full 4 GBs/s bandwidth overwrite attach memory block writes. This overhead can be reduced by decreasing the probability of triggering a PA-to-PCMA translation. This would reduce the total endurance of the system to the overwrite attack, but may still remain acceptable if the cell write endurance is large. For instance, for a 16GBytes memory, if the extra PCM write traffic is limited to 1 extra PCM block write per 32 physical memory writes i.e., a 3.1% write overhead, the cell endurance is 256 Mega writes and 64K blocks memory regions are used then the secure PCM memory still survives a full 4 GBs/s bandwidth overwrite attack for 19 % of its 32 years expected lifetime, i.e., about 6 years.

7 Applications

1. PCM in embedded systems:
   A common use of memory in an embedded system is for code storage. Systems requiring a relatively small amount of memory, less than approximately 2Gb, are architected such that code is executed directly from the NOR flash . This memory is often white paper Phase Change Memory: A new memory to enable new memory usage models used as storage memory for an embedded file system. DRAM is often used in these types of systems as a scratchpad memory. In these types of systems, PCM can be used as a code execution memory. With its bit-alterable feature, PCM is able to displace some or all of the DRAM required in the system. In embedded system applications , PCM can reduce the density requirements for DRAM while fulfilling the density requirement of the NAND flash. At the same time, the presence of PCM in this type of system simplifies and improves the performance of file systems stored in the PCM due to the bit-alterability and low latency features

2. PCM in wireless systems:
   It is common to find nearly independent subsystems for baseband and application processing in wireless systems. At the highest level, these can be considered as independent embedded systems. Generally speaking, both subsystems have the need for a resident execution memory and for storage of small data structures. In many cases, the applications subsystem is also expected to store and perform operations on larger multimedia content. are slower but on the same order of magnitude as the latencies of DRAM, albeit on smaller page sizes, PCM can serve as an outstanding code execution memory and outstanding read-mostly memory for all but the most frequently manipulated data structures. The bit-alterability of PCM eliminates the need for block erase, which reduces the DRAM requirements even further, resulting in a lower cost memory subsystem. PCM promises a scalable memory subsystem solution that provides the best overall cost while meeting the increasing performance demands of high-end, multimedia wireless devices

3. PCM IN SOLID STATE DEVICES
   Managing NAND flash in solid state storage (SSD) subsystems is a challenge due to the block-alterable nature of the NAND technology. It is also challenging to handle increasing levels of error management required when the memory is heavily program/erase cycled or frequently read. PCM can be used in SSD systems to store frequently accessed pages and to store those elements which are more easily managed when manipulated in place. Examples of these types of elements include: parity bits for data stored in NAND, bad block tables, and block and page mapping tables. In this scenario, a small amount of PCM could be used to enhance the manageability of NAND. By minimizing the stress on the NAND memory, higher density MLC NAND is enabled, thus leveraging the capability of PCM to lower the cost of the NAND flash in the subsystem. This caching with PCM will improve the performance and reliability of the subsystem. that multiple erase cycles are likely required to free space to store the new data being written to the device. This increases the number of cycles on the device and further accelerates the time until the maximum endurance limits are reached. The bit-alterable nature of PCM solves the issue of increased write cycles when the device is full. Higher endurance of PCM addresses the needs of these systems when heavy use is expected.PCM in computing platforms As a volatile memory, DRAM consumes power to simply maintain the contents of the memory.

4. PCM as non-volatile memory:
   PCM banks can be turned off when they are not in use to provide reduced power in idle states. More importantly, turning off the banks decouples the relationship between density and power consumption. This results in a PCM subsystem density envelope that is not limited by the power envelope constraints of that system. In addition to non-volatility, PCM offers endurance and write latencies that are compelling for this type of application. This is a key advantage over read-mostly solutions that have been attempted until now.

8 Conclusion

If the promises of the PCM technology are fulfilled then it might become economically feasible to build a main memory from PCM memory component in the next few years. Such a PCM-based main memory will particularly be attractive due to its very low static energy consumption. However to consider such a memory for an industry product , the PCM based memory would have to be able to resist to software overwrite attacks targeting its physical destruction. In this, a first secure PCM based main memory that will resist to overwrite attacks . By hiding the effective PCM memory address from the rest of the computer system and continuously and randomly moving the physical memory blocks in PCM memory, overwrite attacks are made impossible. The proposed PA-to-PCMA translation scheme uniformizes and randomizes the write flow on PCM memory for malicious overwrite attacks as well as conventional non malicious applications. Our scheme requires some hardware overhead in the memory controller (essentially a PA-to-PCMA translation table, the memory region swapping logic and a random number generator). But our scheme allows the overall PCM main memory to resist to an overwrite attack for a very significant fraction of its expected lifetime, e.g. 62 % for 4Kblocks regions and $32M$ write endurance per cell. The scheme presented in this paper induces a significant extra write traffic (1 extra write per 8 effective writes). This consumes some useful memory write bandwidth and may reduce performance on memory intensive applications. A future study will address this issue for conventional (i.e., non-malicious attack) applications.

References

[1] V. Srinivasan M. Franceschini L. Lastras M. K. Qureshi, J. Karidis and B. Abali. "Enhancing lifetime and security of pcm-based main memory with start-gap wear leveling". *Micro*, Dec 2009.

[2] M. J. Breitwisch C. T. Rettner Y.-C. Chen R. M. Shelby M. Salinga D. Krebs S.-H. Chen H.-L. Lung S. Raoux, G. W. Burr and C. H. Lam. "Phase-change random access memory: a scalable technology". *IBM J. Res. Dev.*, 52(4):465–479, 2008.

[3] O. Mutlu B. C. Lee, E. Ipek and D. Burger. "Architecting phase change memory as a scalable dram alternative". *ISCA*, pages 2–13, 2009.