Journal of Systems Architecture xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy

Krishna Kavi [a,*], Stefano Pianelli [b], Giandomenico Pisano [b], Giuseppe Regina [b], Mike Ignatowski [c]

[a] University of North Texas, United States
[b] University of Pisa, Italy
[c] AMD, United States

ARTICLE INFO

Article history:
Received 8 October 2014
Received in revised form 3 May 2015
Accepted 1 July 2015
Available online xxxx

Keywords:
Memory hierarchy
3D-DRAMs
PCM
Set-associate addressing
Energy modeling
Memory latency modeling

ABSTRACT

In this paper, we describe and evaluate three possible architectures for using 3D-DRAMs and PCMs in the processor memory hierarchy. We explore: (i) using 3D-DRAM as main memory with PCM as backing store; (ii) using 3D-DRAM as the Last Level Cache and PCM as the main memory; and (iii) using both 3D-DRAM and PCM as main memory. In each of these configurations, since the proposed memories are significantly faster than today's off-chip 2D DRAMs for main memories and magnetic hard drives for secondary storage, we introduce hardware assistance to speedup virtual to physical address translation.

We use Simics, a full system simulator, and benchmarks from both SPEC and OLTP suites to evaluate our designs. We use CACTI for obtaining energy and latency values for our configurations. We measure energy consumed and execution performance for the selected benchmarks.

Our studies lead to the following conclusions. The best performance is obtained when 3D-DRAMs are used as last level caches (LLC) and PCM as the main memory. However, this organization performs poorly in terms of energy consumed. Our 3D-DRAM together with PCM as main memory is the best choice in terms of energy consumed. In terms of write-backs, 3D-DRAM as LLC causes fewer writes to PCM than the other organization.

These experiments can be extended to explore specific memory organizations, capacities of 3D-DRAM needed as LLC or main memory and how the hybrid PCM/DRAM memory should be used for specific application contexts.

1. Introduction

Die stacking technology allows for increasing the transistor density by building layers of active silicon, and connecting them with a series of high-speed communication lines known as through silicon vias (TSV) [2,6,7,10,12]. 3D-DRAM technology, in which a variable number of DRAM dies are stacked on processor cores is one of the possible applications of Die stacking [20].

PCM (Phase Change Memory) offers a reliable and scalable non-volatile storage technology. In this technology, during a write operation a driver transistor injects a specific current into the storage material and thermally triggers a phase change. The phase of the material is sampled during a read operation. Unlike DRAMs, PCM cells do not store electrical charge that may decay with time. Thus PCM based storage is read non-destructively. The current needed to induce a phase change can linearly scale with the PCM organization [14–17]. In addition, several improvements have been suggested to increase the reliability and performance of PCM, making this technology attractive as a long-term storage [7,15].

The purpose of this paper is to explore how to use 3D-DRAM and PCM technologies in a processor's memory hierarchy and mitigate the speed gap that currently exists between processors and memories (which is often referred to as the Memory Wall [23]). Our goal is to investigate different organizations for using 3D-DRAMs and PCMs in the memory system. More specifically, we present an evaluation of the following organizations:

i. 3D-DRAM as main memory and PCM as secondary storage (we call this organization CMM, or Cache-like Main Memory);
ii. 3D-DRAM as Last Level Cache and PCM as main memory (we call this organization LLC); and
iii. 3D-DRAM and PCM together as main memory (we call this organization PMM).

* Corresponding author.

http://dx.doi.org/10.1016/j.sysarc.2015.07.009
1383-7621/

Stacked 3D-DRAMs offer much lower access latencies (and higher bandwidths) than off-chip 2D-DRAMs, and PCMs also offer similar advantages over other secondary memory technologies. For these reasons, we propose hardware assistance for virtual to physical address translation and handling page faults. Our feeling is that traditional memory management that relies on several levels of page tables for translating virtual addresses to physical addresses will effectively defeat the advantages of the new technologies. Furthermore, transferring pages between PCM and 3D-DRAM is significantly faster than transferring pages between magnetic disk drives and 2D-DRAM, hence kernel intervention leading to process context switches on page faults should be minimized.

In this paper, we will evaluate our memory organizations and the associated hardware needed to achieve our objective; we measure execution performance and energy consumption for our evaluation. We use several different benchmarks drawn from both SPEC 2006 and OLTP suites, and vary benchmark mixes running on different cores in a multicore system. We use Simics, a full system simulator for our simulations, and CACTI to obtain access latencies and power requirements of our memory systems.

The rest of the paper is organized as follows. In the next section we review the research that is closely related to ours. In Section 3, we describe the underlying hardware components of our memory architectures. Section 4 shows the results obtained using CACTI models for 3D-DRAM memories, PCM memories and the additional hardware structures needed in our architectures (primarily SRAMs). We model these memories using multiple dies, ranks, banks and row buffers. Using values for access latencies and power requirement from CACTI, we evaluate our memory organizations for selected benchmarks. The experimental setup is described in Section 5. Section 6 includes the results and analyses of our studies. Section 7 describes conclusions based on our analyses.

2. Related works

3D stacked DRAMs and PCM (and other non-volatile technologies) are receiving significant interest from the academic and industrial communities due to their latencies and power efficiencies. As such there are many publications that discuss different aspects of these systems or investigate different improvements to such designs. Here we will only review some representative works, and the omission of other works is purely for the purpose of brevity.

Inside an integrated circuit, metal lines are a major source of overhead in terms of latency, area and energy consumed. Different studies show that wire lines require up to 30% of the total power [2,12]. 3D Die stacking technology reduces wire lengths by introducing vertical connections between dies known as *Through Silicon Vias* (TSV) [7].

There are several methods used for stacking two or more dies, including wafer-to-wafer bonding, die-to-die bonding and die-to-wafer bonding with different types of overlays. For our purpose, we will assume die-to-die stacking with face-to-face overlays [6].

2.1. 3D stacked DRAM

3D-stacked DRAMs appear to be an obvious way to take advantage of die stacking, and overcome memory access delays [10,12]. By stacking DRAM dies over the processor[1] and using several TSV connections the memory access latencies can be reduced while increasing bandwidths [5,9,19,20]. Loh proposed two different ways of stacking 3D-DRAM dies: *Wide-3D* and *True-3D-DRAM* [10]. Loh also claims that True-3D-DRAM stacked on processors reduces energy consumed by applications while improving performance [10]. We use True-3D stacking in our studies. There have been other studies that investigated different block sizes for 3D-DRAMs [11], and the use of 3D-DRAMs as Last Level Caches [18]. In yet another study [13], the authors compare the use of 3D-DRAM as main memory and 3D-DRAM as last level cache. This study is similar to ours. However, we also focus on improving address translation by using cache-like indexing.

2.2. PCM as main memory

Qureshi and his colleagues have evaluated using Phase Change Memories as main memory with a small 2D DRAM as a buffer [14]. The DRAM buffers are used to both speedup accesses and reduce write-backs to PCM. The focus of their study is the use of PCM in place of DRAMs for main memory. The DRAM buffer is organized similarly to a hardware cache (but is not visible to the OS), and is managed by the DRAM controller. In our study, however, we evaluate different memory organizations using 3D-DRAM and PCM in the memory hierarchy.

The study by Lee [8] is similar to that of Qureshi [14], in that they also use PCM main memory. Like Qureshi, Lee uses small DRAM buffers between last level caches and PCM to reduce the amount of data written back to PCM. However, Lee studies the use of multiple DRAM buffers instead of a single DRAM based cache in [14].

There have been other studies that examined techniques for improving the performance of PCM-based memory systems and reducing the amount of data written back to PCMs (see for example [8,14,16,17]). These approaches are orthogonal to our work and can be used with our systems.

2.3. Cache-like indexing for Main Memory

For the purpose of improving virtual-physical address translation, 3D-DRAM can be viewed as a cache, even though it is not. Previously we evaluated such an organization and named it Cache-like Main Memory or CMM [5,19]. The CMM organization takes advantage of the 3D-memory being physically near the processor, allowing it to appear both as cache and as main memory. This duality makes the memory perfect either for operations that are more efficient if they use cache-like addressing (fast address translation) or for operations that require main memory like behavior (shared pages, OS management of memory, DMA).

However, our previous studies of CMM were limited for the following reasons:

(a) They did not provide details on the hardware needed.
(b) They did not study power requirements of the organization.
(c) The access latencies for 3D-DRAM and PCM memory were simplistic and rely on average values.
(d) The benchmarks used did not seem to fully stress the memory architectures.

In this paper, we addressed these limitations.

3. Foundation of the architecture

The cache-like indexing mentioned above with CMM designs allows us to minimize the number of levels of page tables needed for translating virtual addresses to physical addresses. Page lookup

[1] Although it is unlikely that a 3D-DRAM will be placed above complex multi-core CPUs due to thermal dissipation issues, there are studies that explore the use of simple cores on the logic layer of 3D-DRAM.

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

is often referred to as Page Walks and may require as many as 7 levels of page tables [1]. Our idea is to use cache-like indexing to eliminate some page table lookups. Consider for example that we use the virtual addresses as tags and search the tags of pages currently resident in memory (somewhat like an inverted page table). Assuming 4 KB pages, 52 bits[2] of a 64-bit virtual address will form the tag that is used to find if a virtual page is resident in the physical memory. If we use fully associative search, then pages can be located anywhere in the memory. Nevertheless, for 32 GB main memory (or 2^{23} pages), we will need $52 * 2^{23}$ tag bits. This is prohibitive both in terms of cost and the time needed for searching. We can consider a direct mapped cache-like addressing, but this limits where a page can be placed in main memory causing unnecessary page-faults. Moreover, this type of addressing complicates placement of shared pages since a shared page may be referenced by different virtual addresses.

We propose a compromise between the use of traditional page tables and cache-like indexing. To reduce the size of tags, we propose to use larger pages, say 32 KB pages. We divide both physical and virtual memories into segments. However, virtual segments contain many more pages than physical segments. Page tables are used to map virtual segments to physical segments. However pages of a virtual segment will compete for pages in a physical segment. We use cache-like indexing to locate a virtual page in the physical segment. Depending on the sizes of virtual and physical segments, we can either use fully associative or use set associative mapping of virtual pages of a virtual segment to physical pages of a physical segment. This still restricts the placement of pages in main memory, but a virtual segment can be mapped to any available physical segment. This is achieved by using one or two levels of page tables.

Consider an example where virtual segments contain 1024 (32 KB) pages and physical segments contain 64 pages. In this case, we need 10-bit tags with pages, where 39 bits of a 64-bit address represent its virtual segment number, given that each segment contains 1024 pages and each page contains 32 KB. Using page tables, the virtual segment is mapped to a physical segment. We use the page number within the virtual segment as a tag to associatively search for its presence in the physical segment. Since we still need page tables to map virtual segments to physical segments, we will use Translation Look-aside Buffers (TLBs) to remember the addresses of the physical segments (see Fig. 1). L-1 and L-2 refer to two levels of page tables used to map a virtual segment to a physical segment. The virtual segment tag is used to find a page in the physical segment.

3.1. True 3D organization

For our studies, we will assume a *True 3D* organization for the 3D-DRAM storage [10]. In a *true 3D-DRAM* organization, the upper layers contain only the DRAM bit cells. The first layer (or logic layer) contains only the control logic such as sense amplifiers, row decoders, row buffers etc. We also place SRAM cells needed for tags on this layer. In a true 3D organization, ranks and banks of DRAM cross multiple layers to reduce the length of data paths and increase clock frequencies. The central bus needed in alternative 3D structures where 2D-DRAMs form layers of the 3D stack, disappears providing real estate for several TSVs (each serving a single or a small group of 3D ranks) that can serve multiple independent memory requests simultaneously.

Another important advantage of separating logic and memory cells is that it simplifies the manufacturing process since we are not mixing different technologies on the same layer. DRAM bit cells are implemented using NMOS while logic is designed using CMOS. In our work, we assume that all the extra logic (such as SRAMs needed for our cache-like indexing, row buffers, and other components of a memory controller) is placed on the logic layer of the 3D structure. In fact, since layer 1 is dedicated for these functions we feel that it should have more than adequate area to accommodate our requirements.

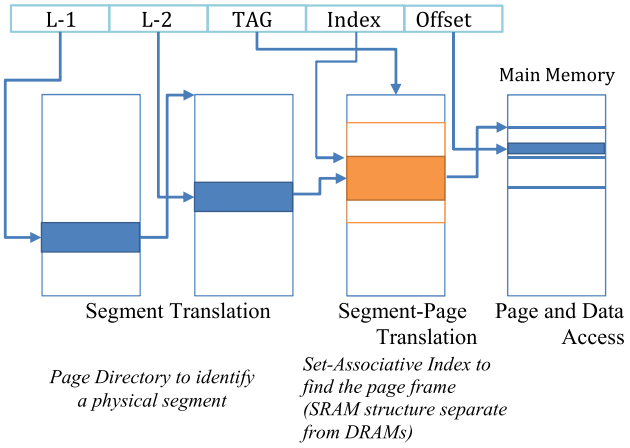


Fig. 1. Cache-like indexing in CMM.

3.2. CMM (3D-DRAM as main memory using cache-like indexes)

To access physically indexed (and tagged) memories in 3D-DRAM, virtual addresses generated by the processor must be translated. To speed up this process CMM uses a TLB to save a significant amount of time by avoiding page table walks for recently accessed pages.

3.2.1. TLB

The CMM TLB associates the physical address of the required data using a pair ⟨ASID, VA⟩, where ASID can be viewed as a process id and VA is the referenced virtual address. Unlike the fully associative TLB used in our previous research [5,19], here we use TLBs with limited set-associativity. To reduce the number of entries in the TLB and reduce the number of tag bits needed for our cache-like indexing, we use 32 KB pages, instead of traditional 4 KB pages. Transferring large pages between secondary and primary memories may cause longer transfer latencies. Previously we explored the use of subpages within a page so that only a few subpages at a time are transferred [5,19]. This however requires bit maps for tracking valid and invalid subpages within a page. For example if 128 byte subpages are used, each TLB entry must track 256 subpages. If we use 1 bit to indicate that the subpage of that page is valid and 1 bit to track dirty subpages, each TLB entry needs 512 bits for tracking subpages, in addition to the bits needed to represent the virtual page number. In this paper we will not use subpages. However we will show power requirements for different TLB sizes and associativites in Section 4.

3.2.2. SRAM

It should be noted that a page may still be resident in memory even on a TLB miss. To locate this page, traditional systems use page table walks [1], which may cause substantial delays. In the CMM architecture however, since the memory addressing is based on cache-like indexing, on a TLB miss, we search the physical memory using the virtual page number of the page being accessed, as described previously. However, the tag bits needed for cache-like search are not stored in 3D-DRAM (or physical

[2] In multi-process systems, tags must also include process-ids. In such systems, more than 52 bits are needed for tags.

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

memory), but stored in a separate SRAM, while maintaining a one-to-one relationship between SRAM entries and DRAM pages. In other words, when a tag match occurs in the SRAM at entry *i* this implies that the referenced virtual page is at physical page *i*. Once found, the newly obtained physical address is stored in the TLB for future accesses. If we assume 1024-page virtual segments, our SRAM needs 10-bit tags to match the virtual page number within the current physical segment. Each physical segment (with 64 pages) will consist of eight 8-way sets. Using larger virtual segments will require more tag bits. We use the LRU policy to replace SRAM entries (and corresponding DRAM pages).

3.2.3. DRAM

If no physical page for the virtual page is found in DRAM, as indicated by no tag match in the SRAM, the requested page evicts a current page from the virtual segment, and a TLB entry is created. However, since we assume a PCM (or other SSDs) for backing store with very short access latencies (compared to magnetic disk drives), we will assume no context switch occurs during this process. Context switch is needed only if no physical segment is assigned for the virtual segment of the requested page. These two cases can be viewed as minor and major page faults.

3.2.4. Row buffers

In this study, we model ranks, banks and row buffers of 3D-DRAM. The use of row buffers allows us to more accurately model access latencies, since accesses to data already in the row buffer will be faster than when a new DRAM row has to be activated. Such details were not modeled in our previous studies [5,19].

3.3. LLC (3D-DRAM as last level cache)

In modern systems, it is common to find first and second level private caches and a larger shared third level cache. Third level or last level caches (LLC) have ranged in sizes between 4 MB and 12 MB. In most of these systems, (2D) DRAM is used as main memory, with SRAM-based cache memories. In one organization reported in this paper, we use 3D-DRAM as the LLC, instead of SRAM-based LLCs.

When 3D-DRAM is used as the last level cache, we need two structures:

1. One structure built with SRAM logic is needed to make the 3D-DRAM appear as cache (for cache-like indexing and hold tags).
2. The other structure is implemented with DRAM logic and stores the actual data contained in LLC.

For large DRAMs used as LLC, the number of lines of data in the LLC will be very large and thus the number of SRAM entries will be very large. However in current processors, SRAM based LLCs occupy a large portion of the processor chip (as much as 50%). Since we eliminate such SRAM-based LLCs, we feel that the saved area can be used to build a SRAM to hold the tags for DRAM-based LLCs. We propose to use the same virtual to physical address translation described previously with the CMM organization, but for accessing PCM pages since PCM will now be used as the main memory.

When using 3D-DRAM as LLC, the size of a single memory line is set to 1024 bytes, or 8 times larger than a typical cache line. The underlying memory controller will transfer data equivalent to a cache line to processor caches. Smaller DRAM lines will require more SRAM bits for holding tags.

PCM. In LLC organization, we use PCM as the main memory. In this case, we use CMM-like addressing for mapping virtual pages to PCM pages. For PCM as the main memory configuration, the internal organization of PCM is similar to that of a DRAM organization, using ranks, banks and row buffers.

3.4. PMM (PCM and 3D-DRAM as part of main memory)

In another memory architecture, we use both the 3D-DRAM and PCM together as the main memory. Physical pages are either in the 3D-DRAM or in off-chip PCM. This organization offers advantages of both technologies – faster DRAM accesses plus denser and lower energy PCM memories.

It should be noted that this choice is different from those of [8,14,25]. In those studies, a (2D) DRAM is used as a buffer to PCM-based main memory. In our study, 3D-DRAM and PCM together form the main memory of the processor. While it is possible to use the 3D-DRAM portion of the memory as buffers by migrating pages from PCM to 3D-DRAM, that is not our goal in this study. Once 3D-DRAM is stacked on processor cores, its capacity cannot be expanded. One must use additional off-chip 3D or 2D DRAMs to expand the memory of such a system. Our study shows that an off-chip PCM offers a better choice for scalable memory capacity in terms of density, cost and energy requirements.

Using both the PCM and the 3D-DRAM as main memory presents new opportunities and challenges. The main challenge is the placement of a physical page. Should pages be distributed evenly between DRAM and PCM memories? Should more intelligent placement be used such that commonly used pages reside in DRAM for faster accesses? Should we permit migration of pages between these devices?

Intelligent placement requires changes to system software (OS, memory managers). We hope that future research will explore such innovative approaches to take advantage of the heterogeneous main memory organization. In this study we use static page placement and a very simple hardware-based approach for locating pages.

The policy we use is called the Grouped Mixed Policy. We divide the total number of pages into a certain number of subsets that we call groups. Some of the pages of a group reside in the 3D-DRAM while others reside in the PCM and the distribution of pages of a group to PCM and 3D-DRAM is based on the capacities of these devices. Consider for example, the ratio of the 3D-DRAM and PCM capacities is *a:b*. Then, for every group with a + b pages, the first *a* pages will be assigned to the 3D-DRAM and the next *b* pages to PCM.

Other strategies for placing pages in these memories can be explored, but are not considered here.

4. Cacti models

We use CACTI to model our TLBs, SRAMs needed for tags and we use CACTI 3D [22] to model 3D-DRAM memories. More specifically, we obtain values for these parameters:

- memory access time;
- cycle time;
- area;
- dynamic power.

We use these parameters obtained from CACTI in our Simics based simulations to obtain execution times and overall energy consumed by benchmark applications.

4.1. TLBs

We simulated several TLB configurations with results shown in Chart 1 and 2. From the charts, it can be seen that a fully associative TLB is impractical. In fact, this version performs worse both in terms

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

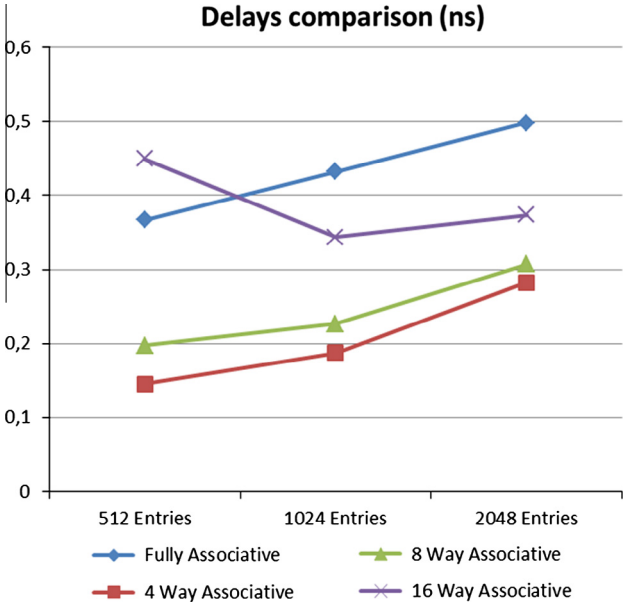


Chart 1. TLB Delays comparison.

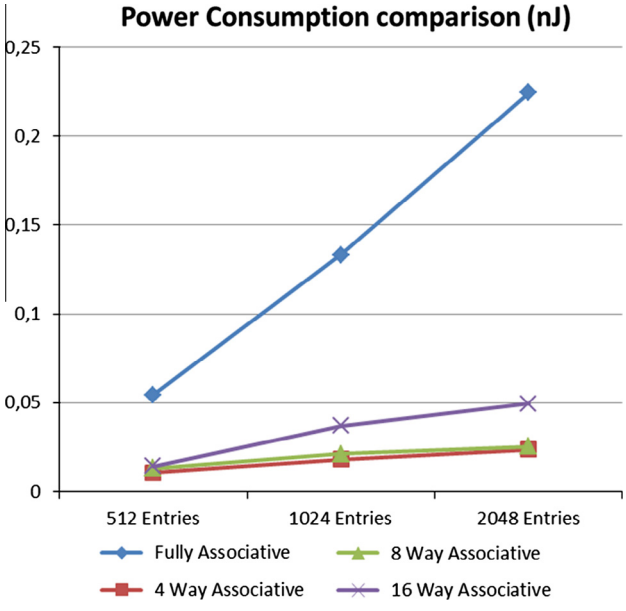


Chart 2. Power consumption comparison.

of latencies and energy consumed. The 16-way TLB appear to exhibit an anomalous behavior for 1024 and 2048 entries. It is our view that this is because of how CACTI operates – it optimizes memory organizations to reduce energy requirements. This can be noted when we see that the 16-way design does not require significantly higher power than that for the 4-way design.

We feel that a 512 entry TLB is too small for the sizes of memories that we propose in our research. This leaves us with the following choices for a TLB:

- 1024 Entries, 4-way associative;
- 1024 Entries, 8-way associative;
- 2048 Entries, 4-way associative;
- 2048 Entries, 8-way associative.

4.2. SRAMs

As noted previously, we use SRAM structures to enable cache-like addressing of our main memories. SRAM stores tags representing (partial) virtual addresses of currently resident physical pages as described previously. The size of the SRAM depends on the size of the main memory, since a tag is stored in SRAM for each main memory page. The CACTI simulation results for our SRAM structures are shown in Charts 3 and 4.

From Chart 3, as expected, delays increase with the size of SRAMs. But delays can be minimized when more banks are used. The energy values shown in Chart 4 exhibit somewhat unexpected behaviors. The energy consumed for 16 GB sized memories are worse with 1 bank than with 2 or 4 banks. This may in part be due to how the CACTI tool is optimizing the SRAM organizations, similar to the results for TLBs described before. We need to choose the number of banks based on how the size of the memory affects the latency values.

While our designs require large SRAMs, we feel that there should be adequate space on the logic layer of a 3D organization.

3D-DRAM. We chose 8, 16 and 32 GB for our DRAMs, but varied the number of ranks and banks. We use one channel for our memory structure. This choice actually penalizes the memory parallelism but it also provide a very simple entry-level 3D-DRAM for the True-3D configuration. To our knowledge, such designs are

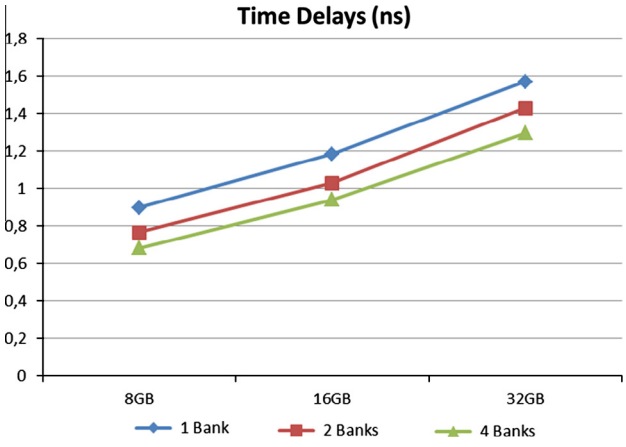


Chart 3. SRAM latencies.

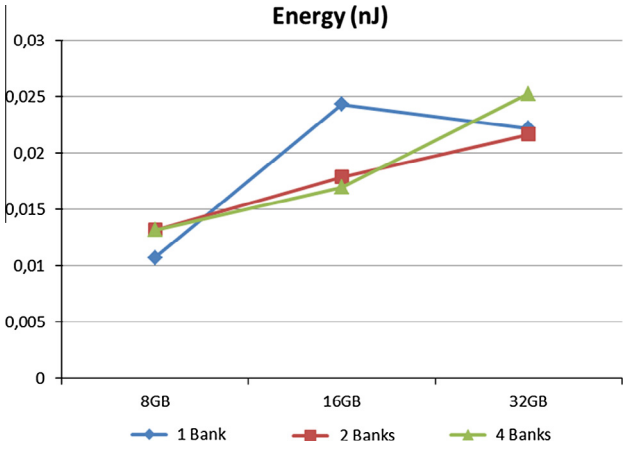


Chart 4. SRAM consumed energy for a single access.

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

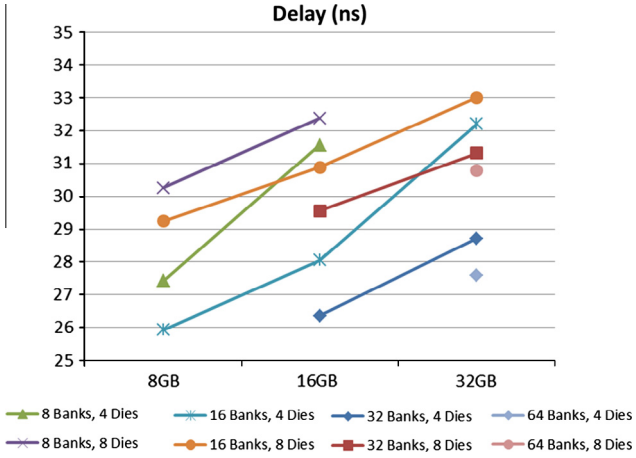


Chart 5. Average latencies for DRAMs.

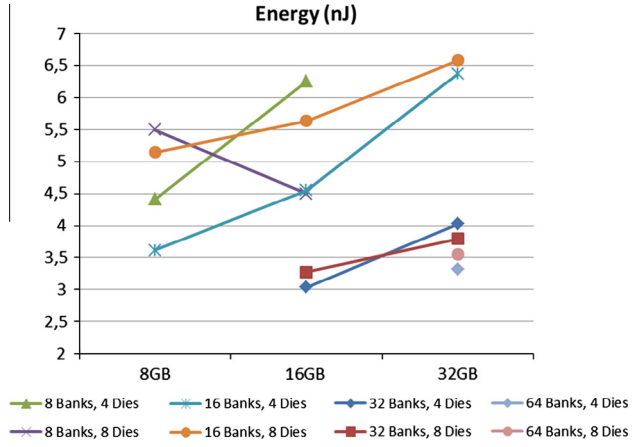


Chart 6. Energy consumed during a memory access.

not yet commercially available. HMC and HBM, which do not use true 3D organization, have proposed 4 channels. But there are serious concerns regarding the amount of energy consumed by these channels, even in the quiescent mode.

We decided to use 4 ranks. This assures a good symmetry inside a single layer and provides space needed for TSVs. Let us remember that the number of ranks must be a power of two. Charts 5 and 6 show the results of our experiments by varying the number of banks and the number of dies.

We feel that banks with more than 512 MB of data are not reasonable. All the results indicate that designs with 8 dies (8 layers of DRAM cells) are not the best choice for our organizations, since they consume more energy and cause longer latencies. We notice that among 4 die alternatives, larger memories perform better with more banks, that is, 16 banks for 8 GB, 32 banks for 16 GB and 64 banks for 32 GB.

4.3. PCM

Initially we explored using available CACTI extensions for modeling PCM, such as the NVSim [24]. However, this tool was not useful for our study because it only simulates PCM at a bank level while we needed to simulate a complete PCM memory device with multiple banks. So we followed the work of Qureshi [14] for modeling PCM. Qureshi states that for a PCM of *x*GB, it's access delays and energy consumption will be 4 times that of a 2D-DRAM with one-fourth the capacity (or *x*/4 GB).

Table 1
PCM multipliers updated with Qureshi's work.

Operation	Old multiplier	New multiplier
Read	4	4
Write	4	32[a]
Read Energy	4	1.3
Write Energy	4	1.3

[a] The multiplier is still 4 but writes are longer (we assumed 8 times longer than a read operation) in PCM due to the difference of time needed for a SET or a RESET operation.

Qureshi made the following assumption: since PCMs are non-destructive and non-volatile, several modifications to 2D DRAM models must be made. The following table shows the model values (multipliers) used for modeling PCM in our system (see Table 1).

5. Experimental setup

To simulate the different memory architectures described in this paper we used Wind River Simics, a full system simulator. Simics includes several tools and modules that can be used to model user-defined architectures. Since we are only studying the memory subsystem, the G-cache module is of most interest to us. This module is originally designed to simulate simple caches, but can easily be expanded to simulate other memories.

We modified the G-Cache module to build all three memory organizations described in this paper, simulating both 3D-DRAM and the PCM behaviors by intercepting every memory request and managing them according to the specific memory organization, and computing the CPU stall time needed to handle the memory operation. We used default values for L1 and L2 caches: 32 KB L1 data and instruction caches, 128 KB L2 cache, all with 128 byte lines, 1 cycle latency to L1 and 10 cycle latency for L2 cache accesses.

We used two different benchmark suites to stress test our three memory organizations:

1. SPEC CPU 2006. Suite of programs created by the SPEC consortium and widely used to represent standard CPU based benchmarks.
2. OLTP. Suite of programs characterized by applications that have large memory footprints and represent Cloud and transaction workloads. Since the systems studied here include very large (32 GB or larger) memories, we felt that our experiments should include server class benchmarks to stress the memory architecture.

Since we used a 4-core *x*86 – 64 "Hammer" system in our simulations, we created several benchmark mixes (mixes of 4 benchmarks each) to test our architecture. In the tables below we present the benchmark mixes. Table 2 lists the benchmark mixes from the SPEC2006 suite, similar to those used in our previous studies [19]. These mixes represent different combinations of memory footprints, from small to large. We will refer to each mix of 4 benchmarks by the name given in the first column of the table.

For the second set of experiments, we created 4 mixes from the OLTP suite and these are listed in Table 3. Again, we use the name in the first column of the table to refer to the corresponding mix of 4 benchmarks.[3]

[3] Thus when we say 'Gobmk' or any other benchmark name, are actually referring to the corresponding mix and not a single benchmark application.

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

Table 2
SPEC 2006 benchmark mixes.

Mix name	Bench 1	Bench 2	Bench 3	Bench 4	Total (GB)
Gobmk	Gobmk	Hmmer	H264Ref	Gromacs	0.046
Gamess	Gamess	Sphinx3	Tonto	Namd	0.027
Sjeng	Sjeng	Libquantum	Leslie3d	Astar	0.192
Omnetpp	Omnetpp	Astar	Calculix	Gcc	0.140
Milc	Milc	Wrf	Zeusmp	Soplex	0.866
Zeusmp	Zeusmp	Leslie3d	Gcc	CactusADM	0.718
GemsFTD	GemsFTD	Mcf	Bwaves	CactusADM	2.262
Mcf	Mcf	Zeusmp	Milc	Bwaves	1.656

Table 3
OLTP benchmark mixes.

Mix name	Bench1	Bench2	Bench3	Bench4	Total (GB)
Auction mark	Auction mark	Auction mark	Sjeng	Stream	20 ÷ 25
Seats	Seats	Seats	Sjeng	Stream	20 ÷ 25
Tatp	Tatp	Tatp	Sjeng	Stream	20 ÷ 25
Epinions	Epinions	Epinions	Sjeng	Stream	20 ÷ 25

We used Sjeng and Stream benchmarks along with OLTP in these mixes to represent server environments that may be presented with large footprint applications along with CPU intensive benchmarks. For L-1 and L-2 caches, we used the same configurations as those of our previous studies [19]: L-1 caches are 32 KB with 4-way associativity and 128-byte lines; L-2 caches are 256 KB with 8-way associativity and 128-byte lines.

In all experiments, we warmed the caches (including 3D-DRAMs) for 500,000,000 instructions and then collected data after another 1,000,000,000 instructions (following similar practice used by other researchers).

5.1. Baseline

To evaluate the efficiencies of our proposed organizations we defined a generous baseline system. The baseline system includes an infinite 2D DRAM for main memory. Thus, it does not encounter page faults. However, the system relies on slower 2D technology. The latencies and the energies modeled are taken from commercially available DDR3 DRAMs with 1 GB for each bank. Also the baseline uses traditional 4 K pages (unlike 32 KB pages used for 3D-DRAM organizations of our work) and relies on multiple levels of page tables for virtual to physical address translation. The baseline uses a finite sized TLB and thus can encounter penalties on TLB misses. We modeled the baseline with TLB miss penalties based on data for commercial systems using AMD processors. We felt that using a very generous baseline allows us to see the true benefits of new memory technologies.

Finally, it should be noted that we only modeled dynamic power and not leakage power for our systems. These studies can be extended to include leakage power models.

6. Results and analysis

6.1. CMM (3D-DRAM as main memory)

In the first memory organization (or CMM that uses 3D-DRAM as main memory and PCM as the secondary memory), we used different TLB configurations. Charts 7 and 8 show the results comparing CMM with the baseline using SPEC2006 benchmark mixes. *For these experiments, we used a 8 GB 3D-DRAM since the memory footprints for SPEC2006 benchmark mixes are relatively small. We explored larger 3D-DRAM sizes for OLTP benchmarks.* Although the baseline is supported with an infinite 2D-DRAM, the baseline is not always better than our CMM. There are several reasons for this. First, the benchmarks used have finite memory footprints, often smaller than the 3D-DRAM configurations we used. Thus infinite 2D-DRAM offers no special advantage. Second, conventional off-chip 2D-DRAMs are slower than 3D-DRAMs. In addition, the baseline uses 4 KB pages (compared to 32 KB in 3D-DRAM). This requires more frequent accesses to TLB and page tables for address translations. We did not include the actual cost of page faults since the baseline has infinite memory, but included the cost of TLB misses.

It appears that for CMM (3D-DRAM as main memory), the 8-way 2048 entry TLB performs better than other configurations. Chart 7 shows that on average, this configuration performs 18% *better than the baseline. In subsequent experiments, we use this TLB configuration.*

Let us now look at the energy consumed (Chart 8). Although the baseline configuration contains infinite DRAM, we used memory sizes that are the same as the 3D-DRAM in CMM organization for the purpose of estimating power-values for the baseline.

Looking at Chart 8, it should be noted that even under this assumption (finite energy consumption by the baseline) our CMM system has comparable performance in terms of energy requirements for SPEC2006 mixes. *More interestingly, as we will*

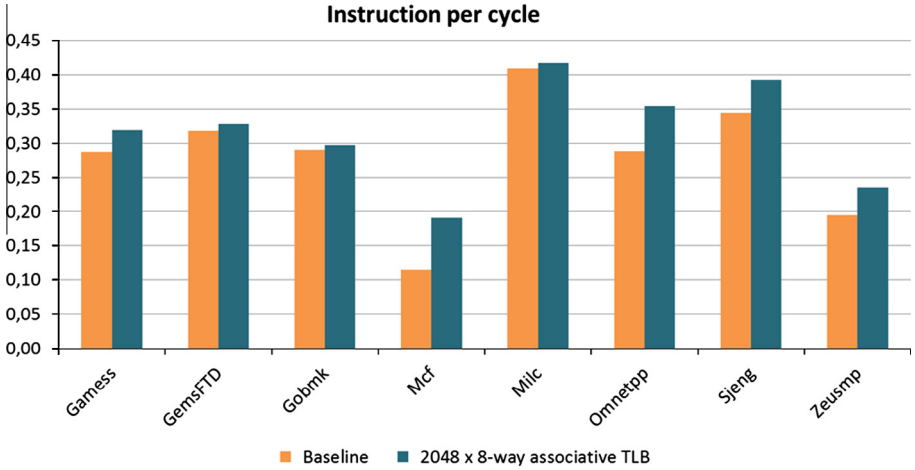


Chart 7. IPC – SPEC 2006 for CMM architecture.

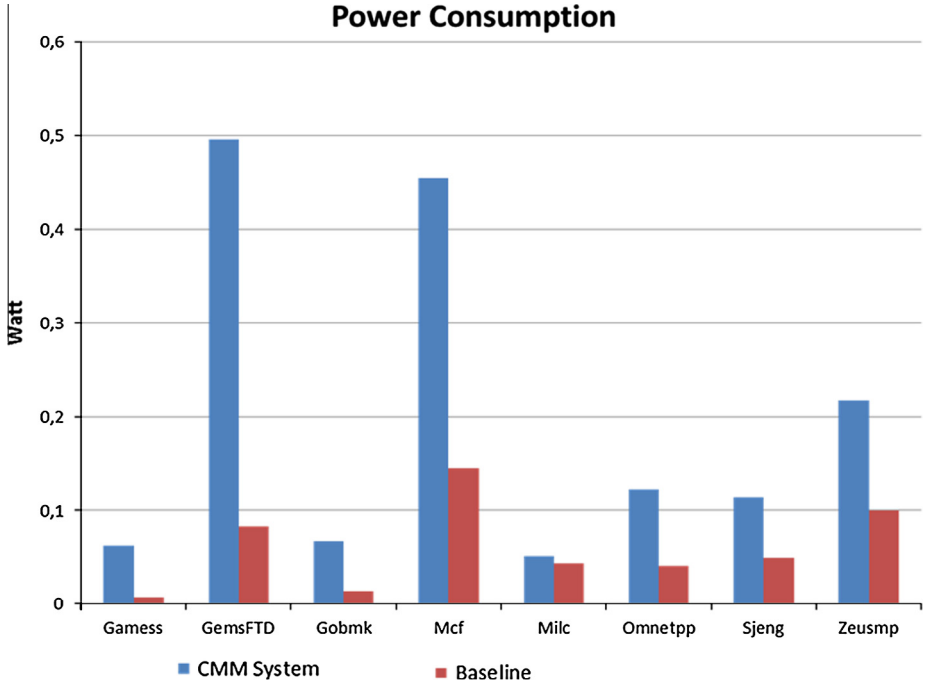


Chart 8. Energy consumption – SPEC 2006 for CMM architecture.

show for one benchmark in Chart 14, TLBs and SRAM structures needed for CMM consume less than 1% of the energy used by the CMM memory system.

OLTP benchmarks have large memory footprints, on the order of 20–25 GB. As can be expected, for these applications the baseline's infinite DRAM becomes advantageous and outperforms our CMM (see Chart 9). The one exception is the TATP benchmark, which experiences a large number of TLB misses. Since we use conventional address translation for the baseline, TLB misses are more expensive than the TLB misses in our CMM organization (CMM uses cache-like indexing and incurs smaller TLB misses).

Note that having a larger 3D-DRAM (16 GB vs 32 GB) is not always beneficial. In some cases the longer latencies associated with larger 3D-DRAMs can defeat the larger capacity, unless more than four dies are used; we used 4 dies. This can be seen when 32 GB 3D-DRAM is used, which is more than sufficient to fully contain the OLTP benchmarks.

It should be noted that while the baseline does better than CMM, the performance differences are not significant. In reality, a practical 2D DRAM based system will face several additional delays due to page faults.

One more clarification regarding our data is needed. While the large 3D-DRAM is divided into ranks and banks, the application pages are not evenly distributed among the memory banks and ranks. This is due to our virtual to physical address translation not being fully implemented. Our current mapping of virtual segments to physical segments results in clustering of physical pages, causing conflicts on memory banks (and TLB entries). These conflicts have caused some performance loss in the CMM system.

A true implementation of our proposed virtual to physical addressing – using segmentation and set-associative addressing – could spread accesses more uniformly across memory banks and lead to fewer bank conflicts and thus better performance. A true implementation requires substantial changes to the operating systems and involves significant resources and manpower.

Let us now consider energy performance for the OLTP benchmarks. Chart 10 confirms our earlier observation that even though the CMM power consumption is still greater than the baseline the values are within comparable range. In some cases, the CMM actually consumes less energy than the baseline.

Note that the energy values for the baseline are based on 2D-DRAM sizes that equal our 3D-DRAM sizes, instead of energy consumed by an infinite 2D-DRAM.

6.2. 3D-DRAM as last level cache (LLC)

Charts 11 and 12 show that our configuration using the 3D-DRAM as the LLC and the PCM as main memory outperforms the baseline configuration (with infinite 2D DRAM). The execution performance gains are particularly impressive for mcf (SPEC2006) and TATP[4] (OLTP). The average performance gain for SPEC2006 benchmark mixes is +27.30% and +45% for the OLTP mixes. In Chart 11, we used 2 GB 3D-DRAM as LLC and 8 GB PCM as main memory. In Chart 12, we use 2 GB LLC and 32 GB PCM.

Although not shown here, we found that using larger than 2 GB 3D-DRAM for LLC shows insignificant performance gains, regardless of the size of the PCM. This may be due to our simulated processor system configuration (4 cores) and the choice of the benchmarks. It should also be noted that even when the LLC is larger than the footprint of a benchmark mix, LLC still incurs cache misses, primarily due to conflicts. Our experiments show that a 2 GB 3D-DRAM as LLC and a 32 GB PCM as main memory is the best choice for the benchmarks tested in our study.

Energy data for SPEC2006 is shown in Chart 13. In Chart 14, we show the data for one of the OLTP benchmark mixes. Other OLTP mixes exhibit similar amounts of energy consumption. What is interesting to note in Chart 14 is that the significant part of the energy consumed in this configuration is due to the 3D-DRAM (as LLC), but the energy consumed increases only marginally when

[4] Again, TATP incurs a large number of TLB misses and our cache-like addressing is the primary contributor to the performance differences between our organizations and the baseline.

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

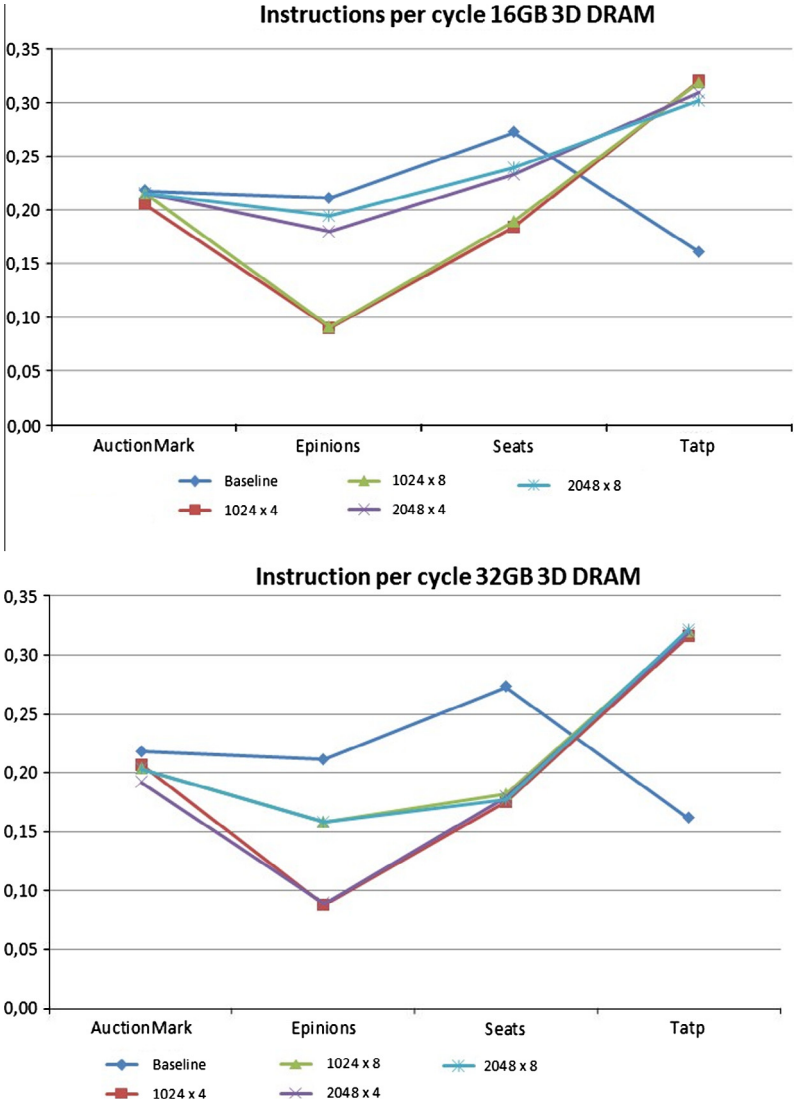


Chart 9. IPC – OLTP for CMM architecture.

the size of the DRAM is doubled. Other structures (SRM, TLB) contribute only a small percent to the overall energy use.

This allows us to choose the best alternative depending on our needs, for instance, a larger last level cache may be more useful when the application can use more cache capacity, or use smaller caches to save energy and minimize the system cost.

6.3. PCM and 3D-DRAM as part of main memory (PMM)

Charts 15 and 16 show the results for SPEC2006 mixes, for our last memory organization that uses 3D-DRAM and PCM together as main memory. Once again, *mcf* shows a significant performance gain over the baseline. The average performance gain for SPEC2006 mixes over the base line is 49% (Chart 15). These results are similar to the data obtained when 3D-DRAM is used as the LLC. This is expected since these applications do not really exercise the large sizes of the memories and thus the 3D-DRAM may contain most of the physical pages (similar to LLC).

However, this memory configuration using both 3D-DRAM and PCM as main memory appears to perform better than the configuration that used 3D-DRAM as main memory and PCM as secondary storage (the first organization or CMM; see Chart 7).

The energy consumed by this architecture (Chart 16) that uses both the 3D-DRAM and PCM as main memory is higher than the CMM organization (see Chart 8). This is because the PMM organization requires additional structures for locating a page either in PCM or 3D-DRAM, and larger SRAM entries to cover the entire main memory system.

For OLTP benchmark mixes, we use several memory configurations. These configurations include larger memories and vary the ratio of 3D-DRAM to PCM capacities. The Chart 17 shows the IPC for these different memory configurations and the baseline configuration.

Here we did not achieve the same performance improvements over the baseline as observed with SPEC2006 mixes (see Chart 15). This is probably due to the larger footprints of the applications and the baseline with infinite 2D DRAM can be beneficial to these applications.

The configurations where the ratio between 3D-DRAM and PCM is either 1:2 or 1:1 perform best and they show an average 30.52% performance gains over the baseline. This may be because a page has a greater chance of residing in the faster 3D-DRAM (if the ratio is larger, more pages will map to PCM).

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

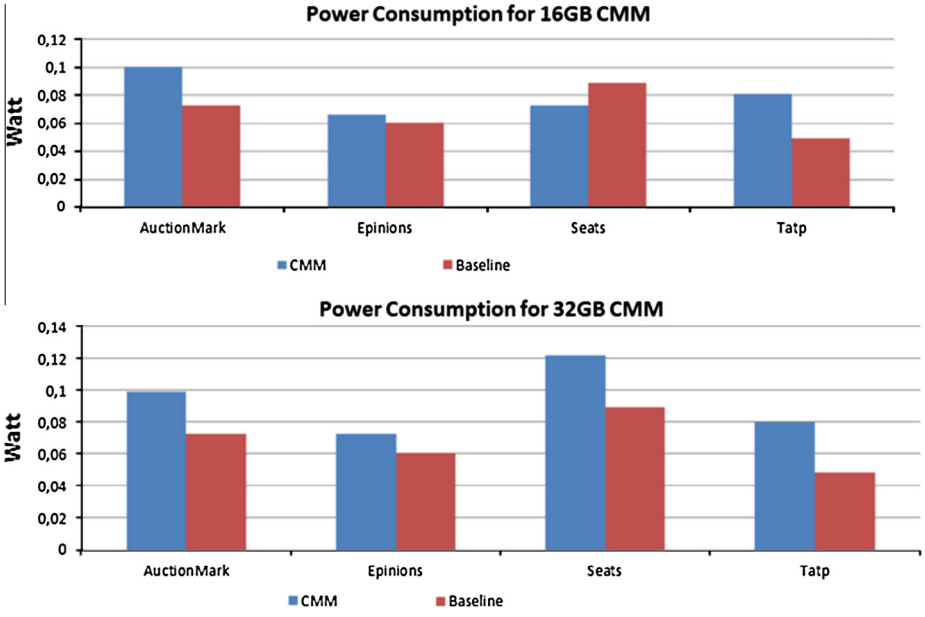


Chart 10. Energy consumed: OLTP for CMM architecture.

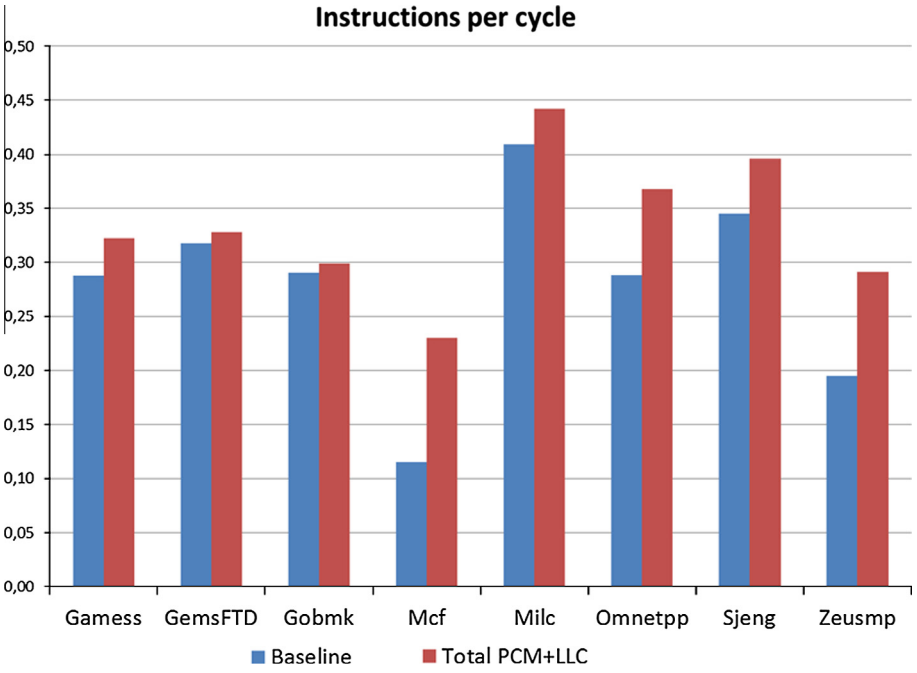


Chart 11. IPC – SPEC 2006 for LLC architecture.

Chart 18 shows the results for energy consumed by OLTP mixes when PMM organization is used. The heterogeneous memory architecture used here does not consume significantly higher energy than the baseline system.

6.4. Write-backs

Since PCM devices have limited write endurances, the number of times PCM cells are modified should be minimized. We now show the data on how our memory organizations compare in terms of the amount of data written back to PCM. Here we show data only for two memory organizations: 3D-DRAM as main memory and PCM as secondary memory (i.e., CMM configuration); and 3D-DRAM as LLC and PCM as main memory (i.e., LLC configuration).

Table 4 shows the data for the various benchmark mixes and different CMM and LLC organization. The table shows the amount of data written back to PCM and the total amount of data transferred from PCM. From the table it can be seen that LLC writes back fewer bytes to PCM than the CMM organization. The amount of data written back in either case for SPEC2006 mixes is small. The differences between CMM and LLC organizations (in terms of data written back) is more significant for OLTP benchmarks, since they present larger memory footprints.

It should be noted that when 3D-DRAM is used as main memory (CMM) and a page is evicted, 32 KB of the page are written back to

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

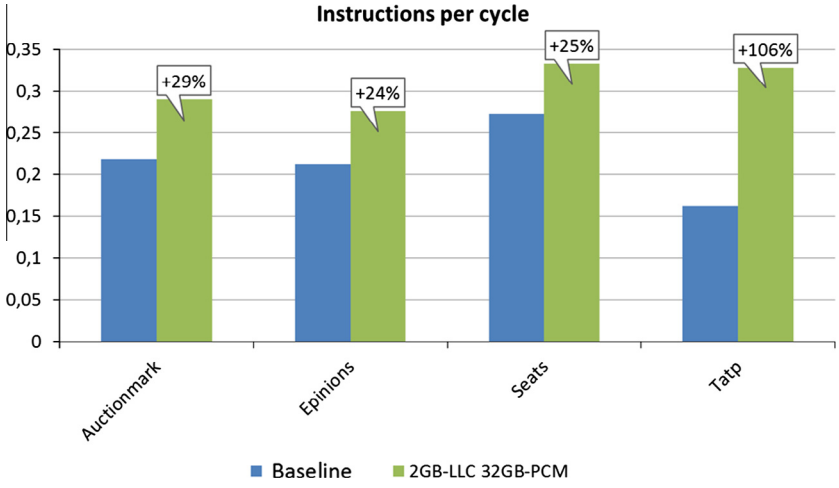


Chart 12. IPC – OLTP for LLC architecture.

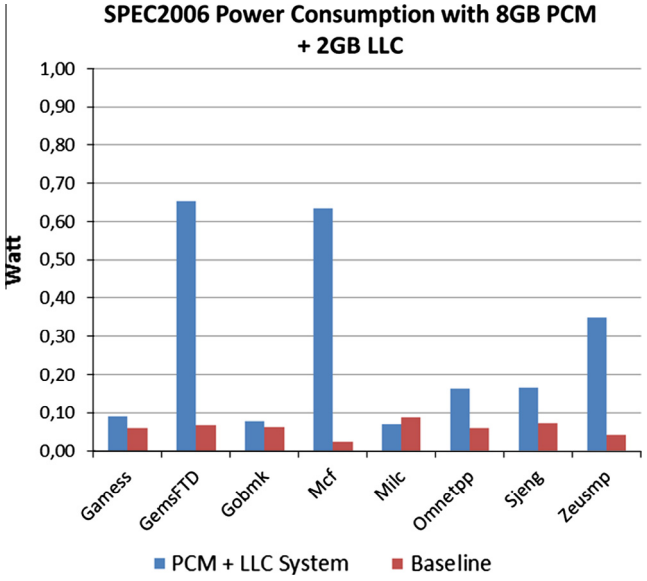


Chart 13. Power consumption – SPEC 2006 for LLC architecture.

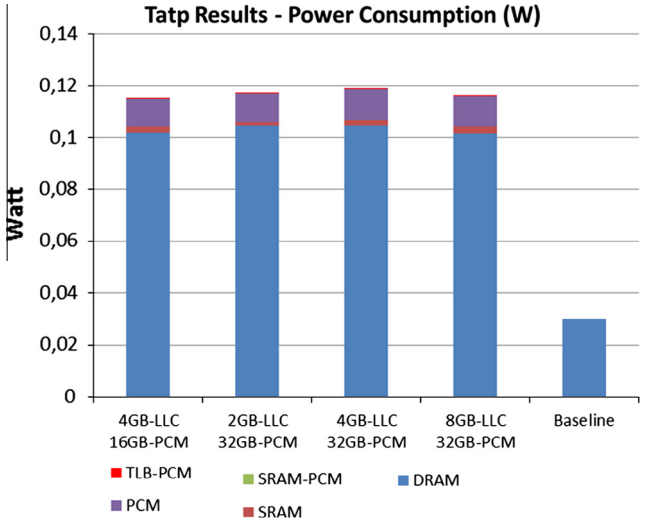


Chart 14. Power consumption for Tatp OLTP benchmark.

the PCM. On the other hand when a cache line in LLC is evicted, only 1024 byes are written back.

Although not included here, for our PMM configuration that uses both 3D-DRAM and PCM as main memory, our simulations show that this organization causes more data to be written back to PCM, since 3D-DRAM is not used as a cache or buffer to PCM. One can expect the amount of data written back will be divided proportionately between PCM and 3D-DRAM based on their capacities, since physical pages are thus mapped.

7. Conclusions

Memory wall [23], which refers to the disparity of speeds between processors and memories, is still a major problem limiting the performance that can be achieved with modern processor technologies. Some new memory technologies such as 3D-DRAM and Phase Change Memories may alleviate this problem to some extent. These technologies present new opportunities and challenges when they are included in a processor memory hierarchy.

In this paper, we explored three different memory organizations for using 3D-DRAMs and PCMs. Each configuration has associated advantages and disadvantages, differing in execution performance, energy consumed and cost. Our goal is to provide initial data that may guide choices on how these new technologies can be used.

7.1. Comparisons

3D-DRAM as LLC configurations, with PCM as the main memory, achieves the best execution performance, but consume more energy than the other two configurations. The energy requirements are due to larger SRAMs needed to store tags for a large LLC (we separate the tags from DRAM and store them in SRAM).

The CMM (3D-DRAM as main memory and PCM as secondary memory) configurations require higher execution times than LLC configuration, but consume smaller amounts of energy. This is expected because CMM uses a longer path to its data using larger pages. At the same time, these organizations need smaller SRAMs.

The last organization that uses both 3D-DRAM and PCM as main memory offers the best trade-off between energy and execution times. This configuration can be further improved with more

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

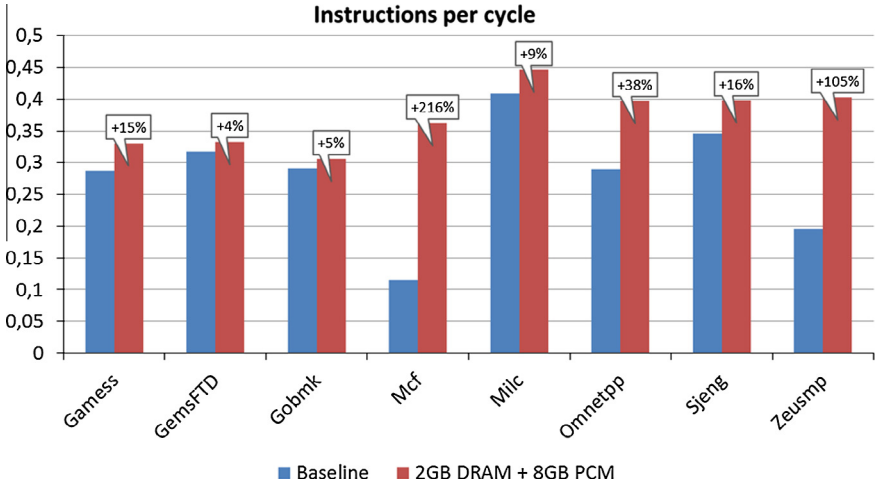


Chart 15. IPC – SPEC 2006 for PMM architecture.

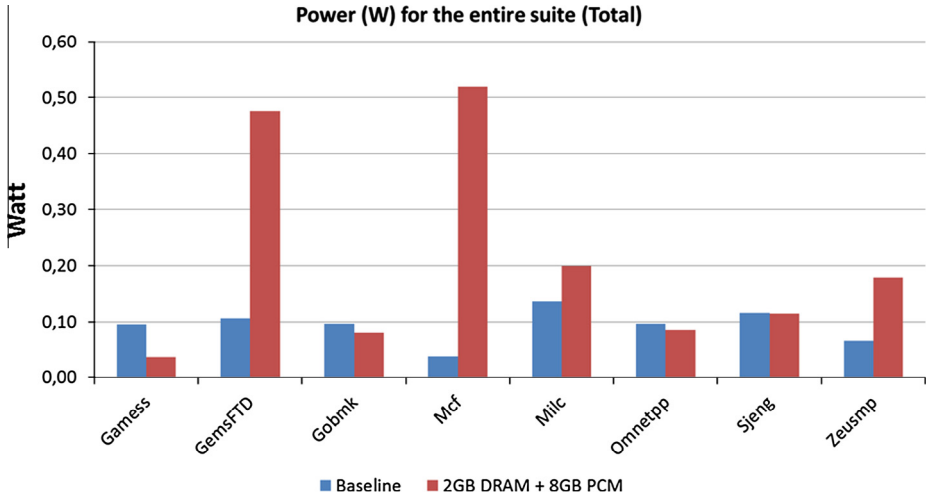


Chart 16. Power consumption – SPEC 2006 for PMM architecture.

intelligent placement of pages in 3D-DRAM and PCM, or migrate pages between these devices. It should be noted that such migrations have some similarities with NUCA caches such as [3,4]. However, unlike NUCAs, pages cannot be replicated in 3D DRAM and PCM. Also, physical addresses change when pages migrate and this leads to flushing of caches. Thus page migration should be used more judiciously.

In terms of data written back to PCM, our studies indicate that using 3D-DRAM as the last level cache results in fewer bytes of data being copied back to PCM when compared to a CMM configuration. However, using both 3D-DRAM and PCM causes more modified data to be written to PCM.

7.2. In summary

In terms of execution performance, 3D-DRAM as LLC is the best choice, followed by both 3D-DRAM and PCM as main memory and finally CMM. In terms of energy consumption, 3D-DRAM together with PCM as main memory is the best choice followed by 3D-DRAM as main memory and finally 3D-DRAM as LLC. In terms of write-backs, 3D-DRAM as LLC is the best, followed by 3D-DRAM as main memory and then PMM organization.

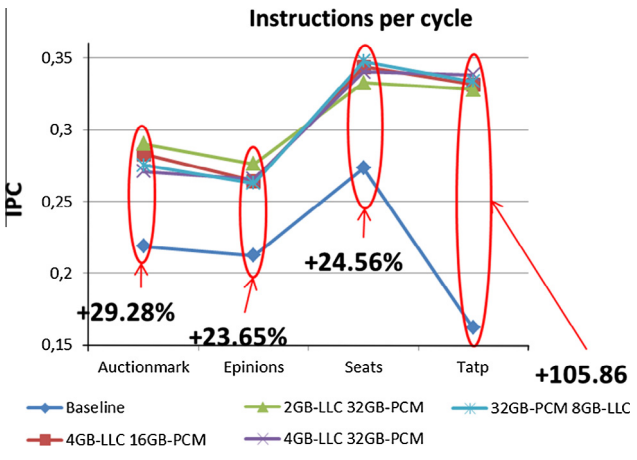


Chart 17. IPC – OLTP for PMM architecture.

Our key contribution is the use of cache-like addressing to aid in the virtual to physical address translation. We have shown that this can improve the execution performance by minimizing page

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

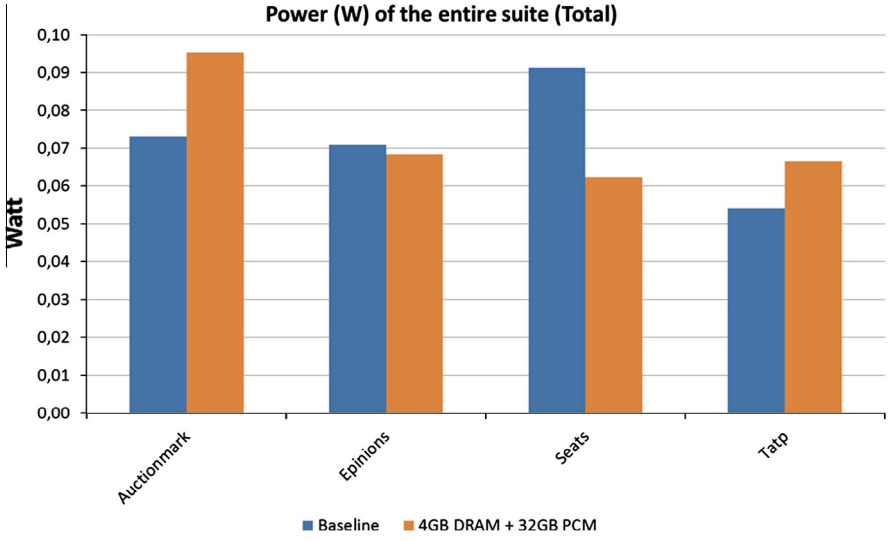


Chart 18. Power consumption – OLTP for PMM architecture.

Table 4
Write backs in different configurations.

Architecture	Benchmark mix	Number of write backs	Transferred Bytes
CMM	Gamess	386	49440
CMM	GemsFTD	762	97536
CMM	Gobmk	164	21024
CMM	Mcf	8762	1121568
CMM	Milc	363	46464
CMM	Omnetpp	142	18176
CMM	Sjeng	734	93920
CMM	Zeusmp	755	96672
LLC	Gamess	80	10240
LLC	GemsFTD	136	17408
LLC	Gobmk	74	9472
LLC	Mcf	170	21760
LLC	Milc	43	5504
LLC	Omnetpp	83	10624
LLC	Sjeng	166	21248
LLC	Zeusmp	132	16896
CMM-16 GB	Auction mark	2267650	290259136
CMM-16 GB	Epinions	5990201	766745696
CMM-16 GB	Seats	2867406	367027904
CMM-16 GB	Tatp	565201	72345760
LLC 4 GB + PCM 16 GB	Auction mark	1332	1363968
LLC 4 GB + PCM 16 GB	Epinions	755	773120
LLC 4 GB + PCM 16 GB	Seats	635	650240
LLC 4 GB + PCM 16 GB	Tatp	14602	14952448

table walks and minimizing the need for OS intervention on TLB misses and page faults. Although we use additional hardware structures, they add only very small overheads in terms of energy consumption.

Acknowledgments

This research is supported in part by the NSF Net-centric and Cloud Software and Systems Industry/University Cooperative Research Center (NCSS I/UCRC) and Advanced Micro Devices (AMD).

References

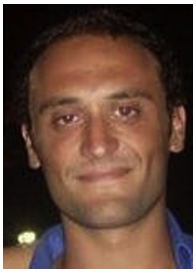
[1] T. Barr, A. Cox, S. Rixner, "Translation caching: skip, don't walk (the page table)", in ISCA-2010, Saint-Melo, (2010).
[2] B, Black, C. Webb, "Die Stacking (3D) Microarchitecture", in: Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture, (2006).
[3] P. Foglia, "Analysis of performance dependencies in NUCA-based CMP Systems", in: Proceedings of the 21st International Symposium on Computer Architecture and High performance Computing, (2009).
[4] N. Hardavellas, "Reactive NUCA: Near-optimal block placement and replication in distributed caches", in: Proceedings of ISCA-2009, (2009).
[5] A. Fawibe, J. Sherman, K. Kavi, M. Ignatowski, D. Mayhew. "New memory organizations for 3D-DRAM and PCMs", in: Proceedings of the ARCS2012: Architecture of Computing Systems, TU Muenchen, Germany, Feb 28-March 02, (2012).
[6] G. Lecarpentier, J.D. Vos., Die 2 Die Bonding, SET S.A.S. (Smart Equipment Technology), 131 Impasse Barteudet, 74490 Saint Jeoire, France & IMEC, Kapeldreef 75, Leuven B-3001, Belgium, (2012).
[7] J. Lau, Through-Silicon Vias for 3D Integration, McGraw-Hill Professional, (2012).
[8] B.C. Lee, E. Ipek, O. Mutlu, D. Burger, Architecting phase phange memory as a scalable dram alternative, Sigarc Comput. Archit. News 3 (37) (2009) 2–13.
[9] C. Liu, Bridging the processor-memory gap with 3D IC technology, IEEE Des. Test (2005) 564–565.
[10] G. Loh, "3D-Stacked Memory Architectures for Multi-core Processors", in Computer Architecture, 2008. ISCA '08. 35th International Symposium on, (2008).
[11] G, H. Loh, and M. Hill Efficiently enabling conventional block sizes for very large die-stacked DRAM caches, New York, NY, in: Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-44), (2011) 454–464.
[12] G. Loh, "Computer Architecture for Die Stacking", International Symposium on VLSI Technology, Systems, and Applications (VLSI-TSA), vol. N/A, p. N/A, 2012.
[13] G. Loh, N. Jayasena, K. McGrath, M. O'Connor, S. Reinhardt, J. Chung, "Challenges in Heterogeneous Die-Stacked and Of–Chip Memory System," in: In the 3rd Workshop on SoCs, Heterogeneous Architectures and Workloads, New Orleans, LA, USA, (2012).
[14] M. Qureshi, "Scalable high performance main memory system using phase change memory technology", in: Proceedings of ISCA-2009, (2009) 24–33.
[15] M. Qureshi, Phase Change Memory – from Devices to Systems, M. & C. Publishers, Ed., Morgan & Claypool Publishers, (2011).
[16] M. Qureshi, "Improving read performance of phase change memories via write cancellation and write pausing". HPCA-2010, Jan (2010).
[17] M. Qureshi, "PreSET: Improving performance of phase change memories by exploiting asymmetry in write times", in ISCA-2012, (2012) 380–391.
[18] M. Qureshi, G. Loh, "Foundamental Latency Trade-off in Architecting DRAM Caches: Outperfornimng impratical SRAM Tags with a simple and pratical design", in: In proceeding of the 2012 45th Annual IEEE/ACM Internatonal Symposium on Microarchitecture (MICRO-45), Washington DC, USA, (2012).
[19] J. Sherman, K. Kavi, B. Potter, M. Ignatowski, A Multi-core Memory Organization for 3-D DRAM as Main Memory, in: H. Kubaitova, C. Hochberger,, M. Danak, B. Sick (Eds.), Architecture of Computing Systems ARCS 2013, Springer Berlin Heidelberg, 2013, pp. 62–73.
[20] H. Sun, "3D-DRAM design and application to 3D multicore systems", IEEE Design & Test of Computers, pp. 36–46, 2009.

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009

[22] S.J.E. Wilton, N. Jouppi, CACTI: an enhanced cache access and cycle time model, IEEE J. Solid-State Circ. 31 (5) (1996) 677–688.
[23] W.A. Wulf, S.A. McKee, Hitting the memory wall: implications of the obvious, Comput. Archit. News 23 (1) (March 1995) 20–24.
[24] D. Xiangyu, X. Cong, X. Yuan, J. Norman, J. Norman P, NVSim: a circuit-level performance, energy, and area model for emerging nonvolatile memory, IEEE Trans. Computer Aid. Des. Integr. Circ. Syst. 31 (7) (July 2012) 994–1007.
[25] W. Zhang, T. Li, "Exploring phase change memory and 3D-stacking for power friendly, fast and durable memory architectures", in: Proceedings of Parallel Architectures and Compiler Technologies, (2009).

Dr. Krishna Kavi is currently a Professor of Computer Science and Engineering and the Director of the NSF Industry/University Cooperative Research Center for Net-Centric Software and Systems at the University of North Texas. During 2001-2009, he served as the Chair of the department. He also held an Endowed Chair Professorship in Computer Engineering at the University of Alabama in Huntsville, and served on the faculty of the University Texas at Arlington. He was a Scientific Program Manger at US National Science Foundation during 1993- 1995. He served on several editorial boards and program committees. His research is primarily on Computer Systems Architecture including multi-threaded and multicore processors, cache memories and hardware assisted memory managers. He also conducted research in the area of formal methods, parallel processing, and real-time systems. He published more than 150 technical papers in these areas. He received more than US $5 M in research grants. He graduated 14 PhDs and more than 35 MS students. He received his PhD from Southern Methodist University in Dallas Texas and a BS in EE from the Indian Institute of Science in Bangalore, India.

Stefano Pianelli received his MS from the University of Pisa in 2014. He spent 6 months at the University of North Texas in 2013, when he conducted the research leading to this publication.



Giandomenico Pisano received his MS from the University of Pisa in 2014. He spent 6 months at the University of North Texas in 2013, when he conducted the research leading to this publication.

Giuseppe Regina was born in southern Italy. He received both his BS and MS degrees in Computer Engineering from the University of Pisa in 2010 and 2013 respectively. He spent 6 months at the University of North Texas in 2013, when he conducted the research leading to this publication. His research interests are in micro architecture fot mobile and cloud applications.

Michael Ignatowski is an AMD Fellow since 2010. Prior to joining AMD, Mike was a Research Senior Technical Staff member at the IBM Watson Research Center. His worked on IBM Power and blade systems, systems and data center power efficiency management. At AMD he is focusing his research on advanced memory technologies and high performance computing. Mike received his BS in Physics from Michigan State University and a MS in Computer Engineering the University of Michigan.

Please cite this article in press as: K. Kavi et al., Memory organizations for 3D-DRAMs and PCMs in processor memory hierarchy, J. Syst. Architect. (2015), http://dx.doi.org/10.1016/j.sysarc.2015.07.009