Journal of Computers Vol. 31 No. 2, 2020, pp. 152-167
doi:10.3966/199115992020043102014

Ultra-Large Last-Level Cache (UL^{3}C) of Phase Change Memory

Hai-Xin Li[1,2,3*], Wei-Liang Jing[4], Ji-Peng Guo[1,2], Yuan Du[1,2],
Zhi-Tang Song[1], Bomy Chen[1,4]

[1] State Key Laboratory of Functional Materials for Informatics, Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 200050, China
{lihaixin, jipengguo, yuandu, ztsong}@mail.sim.ac.cn

[2] University of Chinese Academy of Sciences, Beijing 100049, China

[3] School of Information Science and Technology, ShanghaiTech University, Shanghai, 201210, China

[4] Shanghai Xinchu Integrated Circuit Incorporation, Shanghai 200122, China
{weiliang.jing, bomy.chen}@shxinchu.com

Received 16 July 2018; Revised 25 December 2018; Accepted 9 February 2019

Abstract. With the tendency for applications to grow larger, the computer memory system requires a last-level cache (LLC) of larger capacity and better traits to cater to performance demands. However, the traditional LLC technologies, such as eDRAM, SRAM and STT-RAM, are difficult to enlarge due to their inevitable drawbacks. This paper proposes a phase change memory (PCM)-based ultra-large last-level cache (UL^{3}C) to hold massive data near processors, which enables the cores to obtain information by directly accessing the cache, while there is no need to be concerned regarding the main memory's high latency. Besides, we demonstrate the ideas of fabricating some logic units under the memory array of a 3D PCM to perform extra functions and of implementing the server of fan topology (SOFT) to effectively provide chips with suitable temperatures. We evaluate the UL^{3}C in a full system simulator. The results show that the PCM cache possesses advantages of up to a 39.4% system performance improvement and 91.4% cache power reduction against eDRAM cache. Additionally, enlarging the LLC eightfold further improves the system's performance by 32.1%, with LLC obtaining a 94.5% decrease in the miss ratio and the main memory obtaining a power savings of 73.4% when running large workloads.

Keywords: 3D PCM, computer memory system, larger capacity, last-level cache (LLC), phase change memory (PCM), system performance

1 Introduction

Last-level cache (LLC) has strongly improved the performance of multicore systems. While many workloads tend to be more memory-intensive and have large working sets, the amount of data continues to increase. Most notably, emerging technologies, such as artificial intelligence (AI), high performance computing (HPC) and the like, are unceasingly developing at a high speed [1]. Consequently, the processing system is in need of becoming more and more powerful. Then, a much larger LLC with better traits is needed to meet the demands [2].

To date, the memory technologies being used as LLCs primarily include SRAM (Static Random Access Memory), STT-RAM (Spin-Transfer Torque Magnetic Random Access Memory) and eDRAM (embedded Dynamic Random Access Memory). Although these technologies provide many benefits, they all still have several inevitable shortcomings. As listed in Table 1, SRAM has very low density due to its large memory cell size and cannot avoid large leakage [3]. STT-RAM has very low write speed and high write power consumption, and it is still unreliable [4]. The eDRAM must implement periodic

[*] Corresponding Author

refresh operations, just like the traditional DRAM (Dynamic Random Access Memory), to keep the inner data correct, which consumes a large amount of power. As the capacity increases, the refresh mechanism becomes more complex. The refresh power holds the great majority of eDRAM power dissipation. Moreover, SRAM and eDRAM are both volatile memories, and their leakage problem will become more prominent as the process technology scales down.

Table 1. Comparison of various memory technologies for caches [5-11]

Memory Technology	Cell Size (F^2)	Read Latency	Write Latency	Read Energy	Write Energy	Leakage Power	Standby Power	Volatility
SRAM	146	Short	Short	Low	Low	High	High	Volatile
STT-RAM	54	Short	Long	Low	High	Low	Low	Nonvolatile
eDRAM	65	Short	Short	Low	Low	Low	High	Volatile
PCM	4	Short	Short	Low	Low	Low	Low	Nonvolatile

Nonetheless, eDRAM has currently been shown to be the most suitable technology among the three memories mentioned above when implementing a large LLC because of its high density and high reliability [12-13]. However, with the high-speed development of the new memory technology of PCM (Phase Change Memory), substantial changes will occur. According to Table 1, PCM exhibits a variety of large advantages over eDRAM. PCM is a nonvolatile memory with low standby power and near-zero leakage. Additionally, PCM has a much smaller cell size, namely, much higher density than eDRAM. Moreover, PCM has achieved notably high access speed while consuming low active power. More specific information about PCM and eDRAM is presented in section 2.

Therefore, in this paper, we propose an ultra-large last-level cache (UL^3C) that is implemented by an advanced PCM to significantly improve the system performance with less power consumed.

The main contributions of this paper are as follows:

(1) We demonstrate the memory hierarchy with UL^3C and its benefits. UL^3C is a PCM device that combines up-to-date technologies to gain superfast access speed, and it connects the CPU chip via an OPIO (On-Package I/O) interface in an MCP (Multi-Chip Package). Then, the nonvolatile LLC can hold large amounts of data near the processing cores for a long time, which allows the data to be accessed with high speed and high bandwidth and enables the system to work much more efficiently, especially when running the large workloads of large working sets.

(2) We also propose some relative ideas about UL^3C. Some logic units can be realized under the 3D PCM arrays to serve the UL^3C. SOFT (Server of Fan Topology) can keep the chips working under suitable temperatures. In addition, UL^3C can be utilized in the servers to improve the performance of the data center.

(3) We evaluate this PCM LLC by comparing it with the eDRAM LLC. In addition, we execute the evaluation in a full system simulator. The results show that the PCM LLC achieves a 39.4% system performance improvement and a 91.4% cache power reduction over the eDRAM LLC. Moreover, when we enlarge the LLC by eight times and run large size workloads, the system performance is further improved by 32.1%, with the LLC miss ratio obtaining a 94.5% decrease and the main memory power obtaining a 73.4% savings.

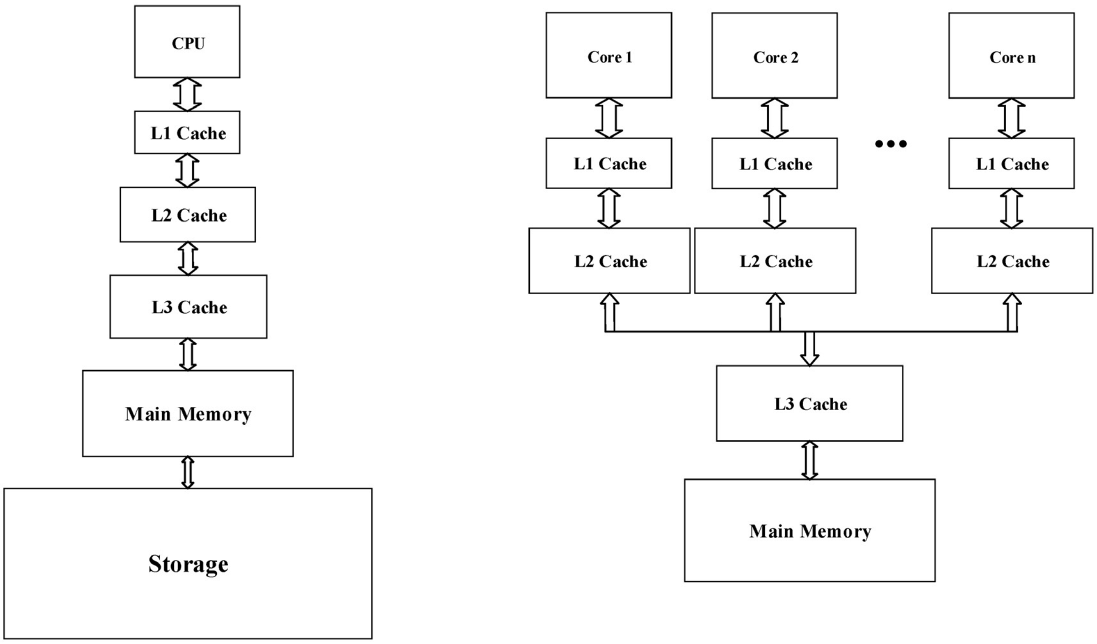
The whole content of this paper is organized as follows. Section 1 shows the motivation and summary of this work. Section 2 introduces some background related to this work. Then, Section 3 demonstrates the memory hierarchy equipped with UL^3C and its related proposals. Section 4 presents our evaluation methodology of UL^3C. Additionally, Section 5 illustrates the experimental results and analysis. Finally, we conclude this paper in Section 6.

2 Background

Typically, the basic memory system of a computer can contain several levels of cache, as shown in Fig. 1(a). In this figure, the L3 cache is what we call the LLC. For a multiprocessor organization, see Fig. 1(b), each processor core has its own private caches, and the LLC before the backing store (main memory) is shared by all of the cores.

2.1 eDRAM

eDRAM is the most commonly used large LLC in the multilevel cache hierarchy, as can be seen in many mainstream products, such as Intel's CPUs of Haswell [14] and Skylake [15], and IBM's POWER processors [16] targeted on servers. eDRAM has higher density than SRAM, and its access speed is faster than the traditional DRAM. The memory cell of eDRAM can be divided into two types [17]. One of them is the structure of 1T1C, which means the basic storage unit that consists of a transistor and a dedicated capacitor. The data a cell stores are represented by the charge quantity in the dedicated capacitor. The other cell is called the gain cell, which is implemented with two or three transistors. The charge is kept in the gate capacitance of its storage transistor. In other words, eDRAM must use some form of capacitor to save data. At the same time, it is a common understanding that a capacitor can easily leak charge. When the charge quantity drops below the threshold, which makes the stored data change [18], the error occurs then. Therefore, eDRAM must refresh itself to make sure the data are correct [12]. Refresh is an operation to reload data from lower memory and recharge the eDRAM cells. The operation not only costs a large amount of energy but also decreases the system performance. The implementation of the refresh wastes an interval of time. In addition, during the refresh period, eDRAM is frozen and is prohibited from being accessed by the CPU (Central Processing Unit) [18]. That will significantly delay the CPU in obtaining the data that it needs. The read operation to eDRAM also causes a loss of charge in the capacitor, which results in the saved data being destroyed [13]. Thus, the data must be written back again.



(a) Basic computer memory system

(b) Cache hierarchy for a multicore organization

Fig. 1.

2.2 PCRAM

PCM is a form of nonvolatile RAM (Random Access Memory) that stores data by the state of the phase change material transforming between crystalline and amorphous [11]. It guarantees the memory cells to be read unmistakably for a long time, and a read operation will not make the inner data corrupted. Promisingly, the PCM consumes a very low amount of power. Its memory cell has almost zero leakage. In addition, the PCM attains a small cell size, which is beneficial to achieve a high density [11]. Moreover, with the development of three dimensional (3D) memory [19] technology and the scale-down in the CMOS process, the memory size increases rapidly. 3D technology builds an array in the vertical direction, which overcomes the obstacles of the memory cells to extend planarly because of Moore's law. As a global representative, Intel has been developing a series of PCM products based on the 3D XPoint technology [20], which they jointly invented with Micron. The capacity of each Optane memory module that they promote for consumers has realized 32 GB [21-22], and the Optane persistent memory for a datacenter reaches 512 GB per module [23]. They are much likely to be larger in the not-too-distant future. With respect to the performance, PCM's actual access speed is already close to that of DRAM [24]. In addition, many studies have shown that PCM possesses great potential to be extremely fast. For example, [10] points out that the $Sc_{0.2}Sb_2Te_3$ compound they designed can make the writing speed of the PCM reach 0.7 ns. Further, [25] achieves 0.5 ns of crystallization by applying a constant low voltage via prestructural ordering effects. Besides, a 2.54 ns reading speed is used in [9]. Moreover, the structure of a differential 2R cross-point [26] or the method of using a reference column proposed by [27] can be applied to significantly improve a PCM's reading speed.

3 UL^{3}C

3.1 Organization

The implementation of the memory hierarchy with our UL^{3}C is shown in Fig. 2, in which the CPU contains multiple cores. UL^{3}C is an off-chip cache that employs state-of-the-art PCM. Similar to the traditional computer storage architecture, our hierarchy still keeps three tiers: cache, main memory and mass storage. SRAM forms the on-chip caches, which can be single level or multiple levels. Main memory is made up of pure DRAM DIMMs (Dual In-line Memory Modules), and the mass storage could consist of SSDs (Solid State Drives) or HDDs (Hard Disk Drives). Specifically, UL^{3}C is encapsulated with a CPU chip in the same package through MCP technology [14]. The package block diagram is depicted in Fig. 3. CPU holds the controller of the UL^{3}C as well as the tags (in SRAM) and the enhancements needed in the power control unit. The OPIO interface [14] is applied to interconnect the CPU and UL^{3}C, which not only provides them with high bandwidth to communicate but also decreases the routings and board size, while consuming less power.

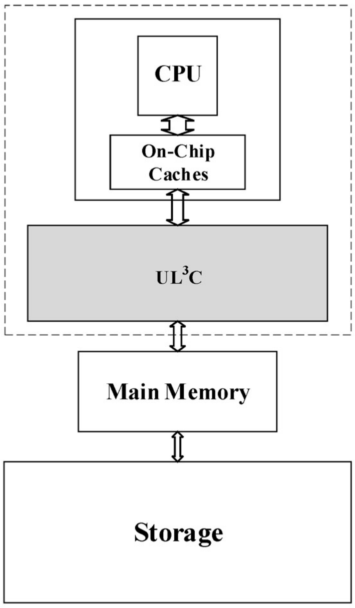


Fig. 2. Memory hierarchy with UL^{3}C

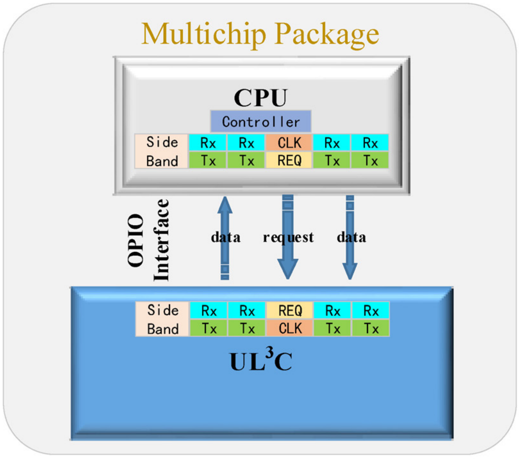


Fig. 3. CPU and UL^3C package block diagram [14]

Since the UL^3C has large capacity, and the CPU accesses the UL^3C with a rather short distance and high-bandwidth interface, part or even all of the system program can be transferred into the UL^3C. That will allow the CPU to directly launch the system from the cache, which greatly reduces the latency and achieves a real instant-boot. Moreover, unlike DRAM or SRAM, PCM is a nonvolatile memory. The inner data will not be destroyed when the power is supplied unsuccessfully. Hence, most applications that commonly used can be installed in the UL^3C, and a large amount of hot data frequently accessed by the CPU is also supposed to be placed there also.

With that implemented above, the CPU will speedily respond to most requests by directly obtaining data in UL^3C without the necessity to reach the main memory and waiting for it to load the required data up from the drives. This approach enables the hit ratio of the LLC to be significantly improved. In that condition, the times of the CPU accessing the main memory are going to decrease, and the traditional procedures of the data being carried to and fro between the nonvolatile mass storage and cache can be largely reduced. In other words, the data flow mainly exists between the CPU and the UL^3C. During a read operation, the CPU obtains the data directly from the LLC, where the results generated in the CPU are also stored with a write command. However, if the required information unfortunately happens to be not in the UL^3C, the CPU will gain it from the mass storage devices through the main memory.

Therefore, in general, the main memory is not requested to be that critical as in the traditional circumstance. Some methods, such as the technologies proposed by [28] and [29], could be applied to lower the refresh rate of the DRAM, and even some parts of the main memory are allowed to be powered down [30]. As a consequence, the proposed UL^3C has great potential to make a contribution toward saving a large amount of power, while the system performance is significantly improved.

3.2 Architecture Extension

As mentioned in the background, the PCM is not limited to being planar. It strives to be developed with three-dimensional technology. Currently, the memory arrays of the 3D PCM are all implemented in the upper part of the chip, leaving the bottom an enormous space to be unoccupied. For example, Intel's 3D XPoint PCM achieves the double storage-selector stacked memory cells only between its metal 4 and metal 5 [31]. Therefore, to effectively utilize the bottom space, some logic units are expected to be realized under the array to perform the functions that we need. Simply, we can take an example by the technology of CMOS Under the Array (CuA) [32] that was proposed by Micron, a memory giant, several years ago. They proposed that the CMOS logic circuitry inside the 3D NAND be placed under the memory array. This type of innovation not only saves the chip size largely [33] but also achieves faster access times [34].

In our UL^3C, as presented by Fig. 4, the logic units include a powerful controller, which employs a certain type of advanced ML (machine learning) algorithm, and some others that are required, such as a statistical module. The controller is supposed to not only accomplish the basic goals for the UL^3C, such as read/write operations and error correction but also work intelligently. In principle, the statistical module will first mark down the characteristics of the data CPU accessed as well as the regularity of the users, during a certain period of time. Then, the controller analyzes and studies the statistical results sent by the statistical module, to optimally manage the content stored in the UL^{3}C and adjust its storage mode. The management methods include loading hot data stored in the mass storage devices, SSD or HDD, up to UL^{3}C through the main memory, and kicking the cold data that the CPU rarely fetches out of the UL^{3}C to release space. There is no clear boundary between hot data and cold data. How they are defined can be specified by the controller according to its learning results and the internal storage-space condition of the UL^{3}C, or by users conforming to their specific requirements. The process of removing cold data from the UL^{3}C could require several steps. The first is to inspect whether the cold data is reserved in the mass storage devices. If yes, then it directly deletes the cold data. However, if it is not detected in the devices, then it transfers the data down to mass storage in case the CPU demands it at some time later. These optimizations will allow the UL^{3}C to be utilized rationally and the system to work efficiently. Additionally, if the UL^{3}C is large enough, then some parts of the cold data are also permitted in.

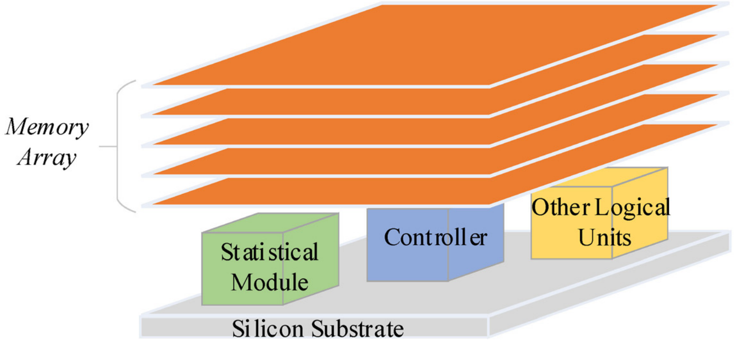


Fig. 4. Logic units under the memory array of the UL^{3}C

In addition, the smart controller is not limited to serving the UL^{3}C. It can also process some lightweight tasks for the CPU when the CPU is busy. Thanks to the proximity to the memory array and the superhigh bandwidth, the controller can obtain the required data and complete the processing instantly. That will make the response speed of the system be greatly accelerated.

In practice, we must consider the problem of heat dissipation because the package simultaneously contains the processors and the UL^{3}C. As shown in Fig. 5, a fan can be placed upon the package, forming an architecture called the server of fan topology (SOFT), to accelerate the cooling of the chips. While the reality is that the fan in a traditional computer adjusts its rotation speed according to the chip's temperature, our fan behaves intelligently on the basis of the CPU's working state. When a peak is predicted to arise instantly, the fan can be controlled to speed up in advance, which is quite different from the traditional case in which the fan does not start an acceleration until the CPU reaches a high temperature. The prediction is executed by some form that is similar to the method used by the statistical module and the smart controller above. Through recording and learning the habits of the CPU, its working state will be tracked and predicted. Then, the fan is going to react in real time. That will effectively keep the cores working at a suitable temperature.

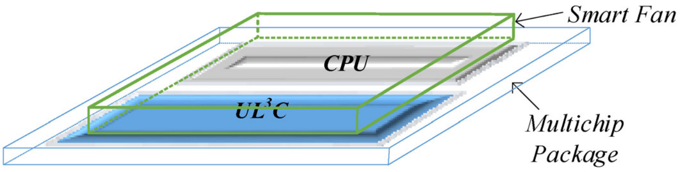


Fig. 5. SOFT architecture

3.3 Embodiment in a Data Center

The UL^{3}C can also bring benefits to the servers of the data center. Theoretically, because of the UL^{3}C's nonvolatility, ultra-large capacity and low access latency, most of the needed data will be stored in it. That will make the processors in the servers tend to access the large cache during most of their time and leave the main memory unvisited. Under this condition, the rarely visited main memory modules in the servers can be logically or physically combined to constitute a main memory pool, which is equally shared by the linked servers. Fig. 6 shows the framework of a data center that we propose. Then, once there is a need for some servers to apply for access to the memory pool, they will obtain a larger memory resource to use. In this way, not only can the individual servers improve their speed of task handling but also the entire data center will run more efficiently. In addition, the problem of memory wall can be alleviated to some extent.

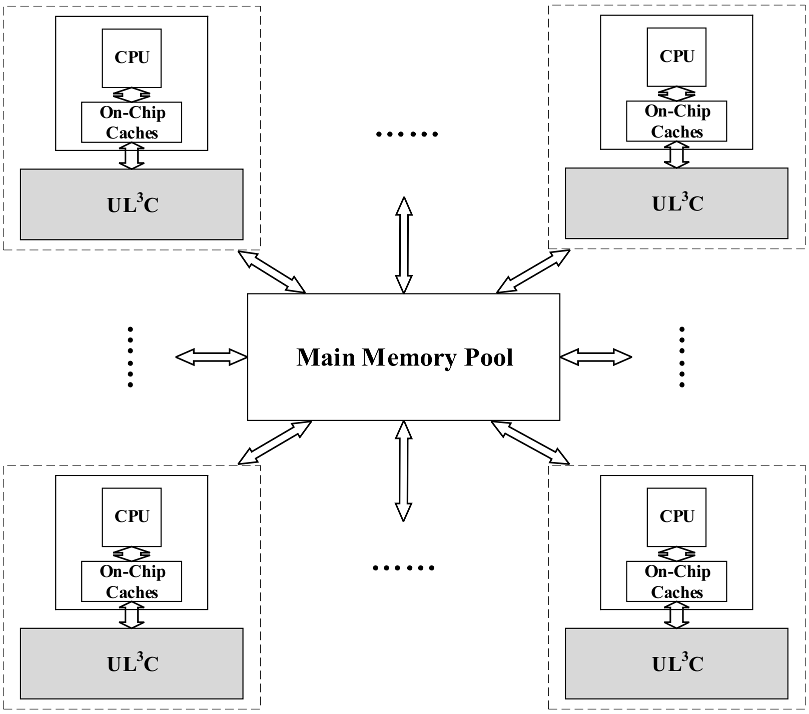


Fig. 6. Framework of a data center that contains a shared main memory pool and with UL^{3}C utilized in the servers

4 Evaluation Methodology

To evaluate the architecture with the UL^{3}C of the PCM that we propose, our study adopts the means of using simulators for convenience. Next, we describe the evaluation method and platform that are used to model the cache memory hierarchy.

4.1 Simulation Method

We select three different PCM models as our LLC and simulate the entire memory hierarchy. As a control, we take the currently most popular large LLC, eDRAM, for reference. The parameter model of eDRAM refers to [13]. For the three PCM models, we distinguish them mainly by their access latency. The first type, which will be *PCM_case1*, is as fast as a real high-performance PCM device. We establish this prototype based on [35] and [36]. The access speed of the second, which is represented by *PCM_case2*, is very close to that of the eDRAM model although slightly slower. The parameters are set by referring to [37]. We make the third type, *PCM_case3*, optimal according to the research findings of [9, 10, 25], in which the researchers have enabled the PCM to achieve a superfast speed.

We divide the simulation work into two parts. During the first part, we model two memory hierarchies, one of the PCM LLC and the other of the eDRAM LLC, and then, we make a comparison of the results to reflect the superiority of the PCM over eDRAM. In the second part, we execute another two simulations: one of a small LLC and the other of a large LLC, to check the further improvements to the system contributed by a larger LLC.

4.2 Simulation System

This work employs an open source, high-speed and high-accuracy simulator —— MARSS [38]. MARSS is based on QEMU, a fast and portable dynamic binary-translation system for emulating processor architectures, and PTLsim, a cycle-accurate full-system x86 microarchitecture simulator. It provides a unique full-system simulation framework for x86 CPUs to simulate/emulate multiple processing cores, coherent caches, on-chip interconnections, DRAM, chipsets, I/O devices and full unmodified binaries of software stacks (including operating systems and libraries). In addition, it supports two datapath models, in-order and out-of-order, of which our study uses the latter.

Additionally, to perform a full system simulation, another cycle-accurate simulator called DRAMSim2 [39] is needed in this work. DRAMSim2 is a memory system simulator and a publicly available DDR2/3 memory system model that can be used in both full system and trace-based simulations. It has a strong focus on being accurate and is easy to integrate. We integrate DRAMSim2 with MARSS in such a way that the whole architecture from CPU down to the main memory is simulated.

Table 2 shows the baseline configuration that we used for our simulation environment. The system utilizes a processor that contains 4 cores operating at 2 GHz. There are three levels of caches in addition to the main memory. As mentioned above, this system works in an out-of-order model. For the caches, the first two levels are built with SRAM for its very low access latency, as is common in current computer architectures. They are all implemented to have 1 bank, 8-way set associative and 64 bytes line size. In the aspect of capacity, the first level (L1) caches, which consist of the instruction cache (Icache) and the data cache (Dcache), are both 32K bytes. The second level (L2) cache is 256K bytes. These two levels of caches are all private and follow the MESI coherence protocol. The third level (L3) cache is the right part that we use as LLC and that we mainly want to model. During all of the simulations, some specifications of LLC are fixed. We configure LLC as a shared and write-back cache with 16-way set associative and 64 B line size. In addition, the L1-Icache and L1-Dcache both have two read ports and two write ports, while the LLC supports only one for the read port and the write port, which is the same as L2. A pseudo policy of Least Recently Used (LRU) cache-line replacement is used in these three levels. Then, we adopt a 4GB DRAM using the parameters of Micron's DDR3 2GB device [40] as our main memory. It is organized as 1 channel of 2 ranks and 8 banks in each rank. The main memory has a 64-bit-width data bus.

Table 2. Simulation system configuration

Components	Specifications
Processor	4 cores, out-of-order (ooo), 2 GHz
L1-Icache	32 KB, 1 bank, 8-way set associative, 64 B line size, private, MESI cache
L1-Dcache	32 KB, 1 bank, 8-way set associative, 64 B line size, private, MESI cache
L2-cache	256KB, 1 bank, 8-way set associative, 64 B line size, private, MESI cache
LLC	16-way set associative, 64 B line size, shared, write-back cache
Main memory	4 GB, 1 channel, 2 ranks per channel, 8 banks per rank, 64-bit data bus

4.3 Benchmarks

Some benchmarks are used in our system to accomplish the evaluation. Here, we choose a widely used benchmark suite from Princeton, PARSEC (Princeton Application Repository for Shared-Memory Computers) [41]. This suite is targeted for the studies of Chip-Multiprocessors (CMPs) with state-of-art algorithms. PARSEC contains 13 workloads, including emerging applications in recognition, mining and synthesis (RMS) as well as systems applications which mimic large-scale multithreaded commercial programs. These workloads are diverse in their working set, data usage (sharing and exchange), parallelization (model and granularity) and off-chip traffic. The suite focuses not only on high performance computing (HPC) but also on emerging desktop and server applications, and it does not have the limitations of other benchmark suites.

We chose 11 workloads from this suite as our benchmarks, which are *blackscholes*, *bodytrack*, *canneal*, *dedup*, *facesim*, *ferret*, *fluidanimate*, *freqmine*, *raytrace*, *swaptions* and *vips*. They are applied to different domains. All of these benchmarks are configured to be single-process and eight-thread and are executed on an Ubuntu 12.04.5 system. Some benchmarks' features are mentioned in the section on results analysis below.

5 Results

In this section, with a quantitative analysis of the simulation results, we first interpret the impact of the new nonvolatile memory PCM on the system performance and power by comparing it with eDRAM. More importantly, we then demonstrate the huge advantages of a much larger LLC against a small LLC on several aspects of the system. In our evaluation, we consider four main parameters for the criteria: (1) the system performance, represented by IPC (instructions per cycle); (2) the miss ratio of the LLC, the proportion of the misses to all of the visits when the CPU accesses the LLC; (3) the power breakdown of the LLC; and (4) the power breakdown of the main memory.

5.1 LLC with PCM

We have the PCM results normalized based on those of the eDRAM in the figures of this subsection. For the system performance, it is easy to understand that the larger the IPC is, the better the system performs; thus, in this paper, the more suitable memory proves to be the LLC. Fig. 7 presents a different system IPC for those three types of PCM compared with the eDRAM. In general, the four memories show a consistency of good and bad under all benchmarks. The reason is that the *PCM_case1* has the largest access latency, much larger than that of the eDRAM, which results in the smallest IPC. Although the eDRAM leads *PCM_case2* slightly in speed, the eDRAM must regularly execute refresh operations to keep the inner-stored data correctly. Refresh costs a substantial amount of time, and during the refresh period, the eDRAM is not allowed to be accessed. That definitely brings down the system performance, which reflects the eDRAM's IPC as lower than *PCM_case2*'s. For *PCM_case3*, the IPC value is the highest, which represents the best system performance. The reason is that this PCM possesses a rather short latency, shorter than that of the eDRAM. In Fig. 7, the cases of several benchmarks, such as *blackscholes*, *swaptions* and *canneal*, could seem special, that the difference between the four LLCs under these benchmarks obviously differs from that of other benchmarks. The former two occur because the application of *blackscholes* has very small working sets and negligible communication, and *swaptions* is also featured with little communication though medium-sized working sets. From Table 3, we can find that these two benchmarks have the least numbers of reads and writes. As a result, they are executed in a very short time, which causes the system to not be sensitive to the LLCs of different access speeds, thus showing almost the same IPC. For *canneal*, it makes the system performance of diverse LLCs differ quite a lot. This finding arises because *canneal* is a workload with enormous working sets and significant communication, which strictly demands a fast LLC. Thus, *canneal* can easily tell different LLCs apart with distinctly different IPCs. Under this benchmark, PCM brings about a system performance with a maximal improvement by 39.4% compared with eDRAM.

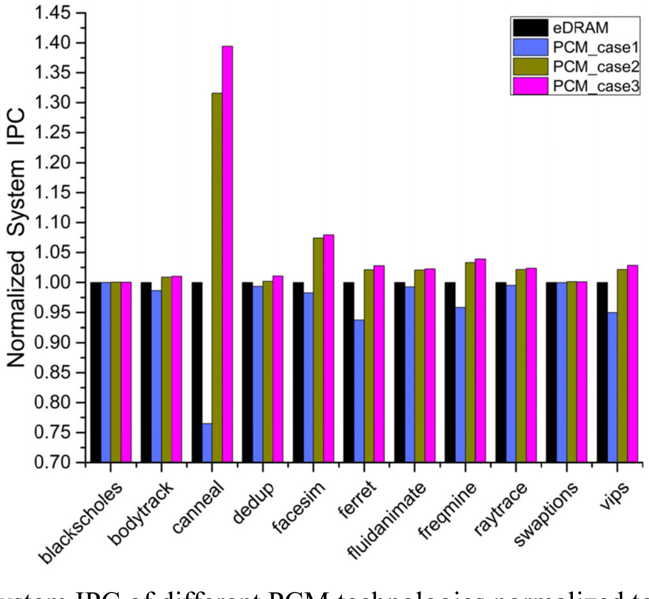


Fig. 7. System IPC of different PCM technologies normalized to eDRAM

Fig. 8 compares the LLC power breakdown of PCM against eDRAM. eDRAM's consumption mostly includes dynamic power, refresh power and leakage power. While PCM is a nonvolatile memory and needs no refresh, PCM costs no refresh energy and mainly consumes the power of dynamic and leakage. Generally, the dynamic activity refers to the access to eDRAM and PCM, which means that their dynamic power is substantially composed of read power and write power. In addition, we can conclude from Fig. 8 that the dynamic power of PCM makes a certain portion of the counterpart of eDRAM under each benchmark, corresponding to the relationship between these two memories' access (read and write) times in Table 3. However, the PCM costs more energy for both per read and per write, and thus, it has higher dynamic power than eDRAM. For the leakage, the memory cell of the PCM has nearly zero leakage current. The PCM leakage power presented in the figure mainly arises from the peripheral circuits, such as the sensing and decoding. At the same time, eDRAM not only has leakage current in the peripheral circuits but also leaks heavily in its memory cells, which results in eDRAM having much higher leakage power than PCM. In addition, eDRAM consumes a large amount because of the frequent refresh operations. Therefore, the PCM can save substantial power compared with the eDRAM in total, 91.4% maximally and 80.5% minimally.

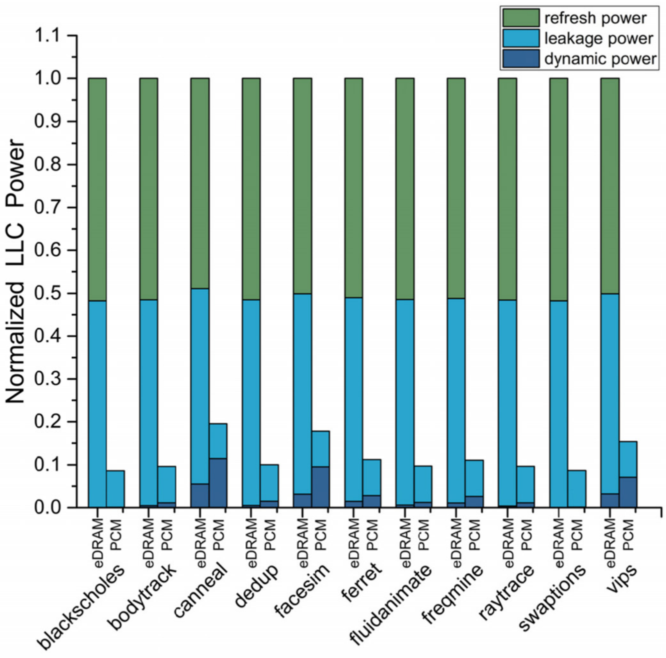


Fig. 8. LLC power breakdown of PCM normalized to eDRAM

Table 3. Average numbers of each benchmark's read and write

Benchmark	Number of read	Number of write
blackscholes	11735	3984
bodytrack	329642	206184
canneal	5821237	9516
dedup	629957	564652
facesim	131356714	10807373
ferret	2573354	1015044
fluidanimate	1036252	334859
freqmine	2727935	1450650
raytrace	827199	36426
swaptions	27164	17043
vips	3635745	7029137

5.2 Large Capacity LLC

Large capacity is another focus of our study. Fig. 9 compares the system IPC of two LLCs of different sizes, and the IPC of the large one is normalized to that of the small one. It can be seen that after the cache capacity becomes enlarged, the system IPC improves accordingly. Especially when executing a large workload, using the benchmark of *facesim* as an example, the improvement becomes extremely apparent. The reason is that a small LLC can only accommodate a very limited amount of data simultaneously. The large workloads must be divided into many parts and alternately loaded into the LLC from the main memory, many times, in such a way that each time the CPU sends a data fetching command, only a small portion of the data of the workload will be loaded. The next required data is very likely not in the cache, and then, a cache miss occurs. The larger the volume of the workload is, the larger the probability of the miss will be. Once the CPU misses in the cache, it must access the main memory. However, surely we all know that the access latency of the DRAM is higher by at least an order of magnitude compared to the cache. The CPU will waste a large amount of time to wait for data being transferred from the main memory to the LLC, which causes a significant drag on the system performance. Then, when the LLC size increases, much more data can be held at the same time, which greatly reduces the number of roundtrips to the cache and the latency time of the CPU that receives the data. Thus, using a larger LLC enables the tasks to be processed faster, which reflects a large improved IPC. Through the large benchmark of *facesim*, the large LLC proves an advantage of 32.1% in contrast to the small LLC. Nevertheless, when performing small applications, such as the benchmarks of blacscholes, *swaptions* and several others, the large LLC shows no clear advantage. The reason is that the small capacity of the LLC can also contain most or even all of the data of the small application, thus allowing the CPU to have a small probability of a miss in the cache and enabling the system with the small LLC to catch up with the system that uses a large LLC.

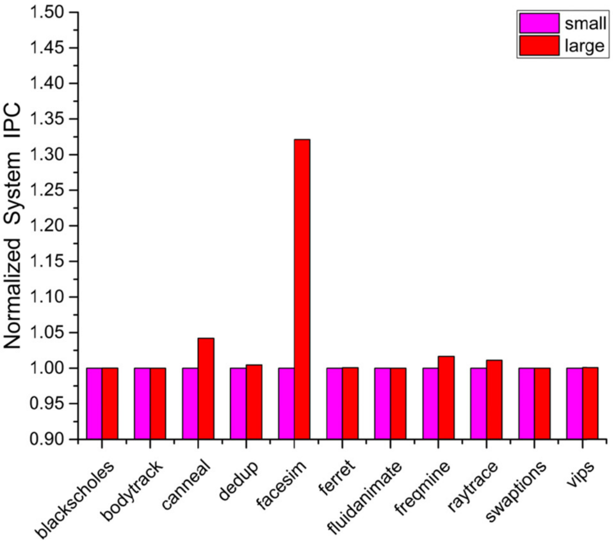


Fig. 9. System IPC of the large LLC normalized to the small LLC

We note that the benchmark of *facesim* in Fig. 7 of the last subsection cannot distinguish different LLCs as obviously as *canneal*, although *facesim* has large working sets and some sharing, as well as the largest numbers of reads and writes, in Table 3. The reason is also that the size of *facesim* is very large but LLC is small, which causes the CPU to have a large probability of missing in the cache. In addition, the much longer latency in accessing the main memory will heavily impact the sensitivity in evaluating the LLCs with different features, which would then present a relatively small IPC differential.

As mentioned above, the CPU will have to access the main memory to fetch the required data when an LLC miss happens. After the command's arrival, the main memory will insert the relative data into the LLC. Usually, the times of the CPU misses in the LLC, including read misses and write misses, are equal to the number of main memory inserts. Table 4 lists the insert numbers of all 11 benchmarks (they are shortened to their initial four letters) when applying a small LLC and a large LLC separately. It is easy to observe that the inserts of all of the benchmarks are reduced when a larger cache is utilized, and in some cases, the reduction is very large. In this paper, the cache's miss ratio is calculated by the formula

$$miss\_ratio = \frac{num\_insert}{num\_read + num\_write}. \quad (1)$$

Table 4. Inserts from the main memory to the small/large LLC of each benchmark

	blac.	*body.*	*cann.*	*dedu.*	*face.*	*ferr.*	*flui.*	*freq.*	*rayt.*	*swap.*	*vips*
small	14571	258646	3012780	925027	112736144	1227387	903334	2375964	804345	41847	4045439
large	11201	96880	425231	384128	6216850	204300	109710	642220	352893	22061	342097

At the same time, the read numbers and write numbers of each benchmark are both stable, as shown in Table 3. Therefore, obviously enlarging the capacity will definitely decrease the miss ratio of the LLC. In Fig. 10, a maximal decrease reaches 94.5%.

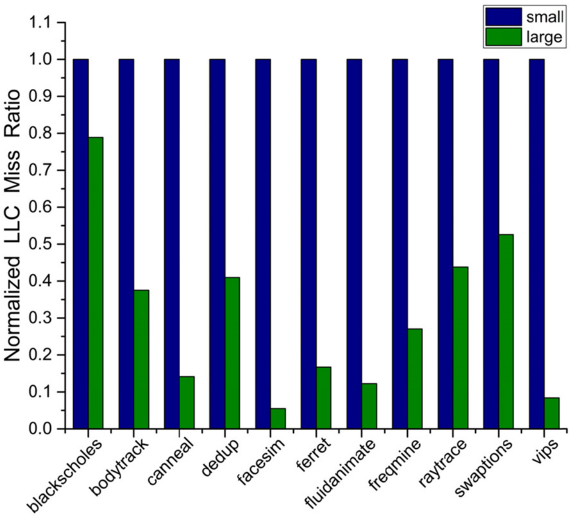


Fig. 10. Miss ratio of the large LLC normalized to the small LLC

Fig. 11 demonstrates the influence of the enlarged LLC to the main memory's power breakdown. A larger LLC will decrease the frequency at which the CPU accesses the main memory, thereby reducing the dynamic power and background power of the DRAM. The burst power, active precharge power and background power are all reduced to some extent under all of the cases in Fig. 11, which contributes to a large savings of the total power, by 73.4% maximally.

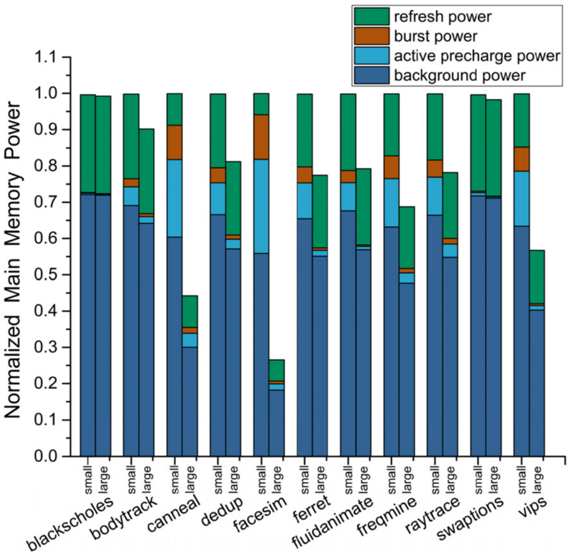


Fig. 11. Normalized main memory power breakdown with LLCs of different sizes

6 Conclusions

We propose to implement a PCM-based UL^{3}C in the computer memory system to address the performance challenge presented by the incessantly developing applications. Thanks to the high density and nonvolatility of the PCM, a large amount of data can be stored in a place closer to the processors. With the continuous progress of PCM technology, this memory's access speed is becoming notably high. In addition, the implementation of interconnecting the UL^{3}C and the CPU chip through an OPIO interface enables the processors to access the LLC with high bandwidth and high speed. A larger cache contributes a higher CPU hit rate. Consequently, the system performance gets a significant promotion. The PCM consumes low power. The UL^{3}C cuts down the procedures of the traditional system uploading data from mass storage to main memory. That saves much power, thus producing savings in the cost. We also propose to realize some extra logic units, such as a smart controller and statistical module, under the memory array of the 3D PCM-based UL^{3}C. The controller can not only manage the data kept in the array but also process some lightweight tasks for the CPU, which further improves the system's performance. Moreover, we also present the idea of SOFT to solve the problem of heat dissipation of the multichip package and to keep the chips working at a suitable temperature.

We conducted an evaluation of the UL^{3}C by simulation. We did see a remarkable performance promotion when executing large workloads, with a large portion of power saved. Specifically, by comparing the PCM that we propose with the traditional large LLC technology of eDRAM, the performance improvement reaches 39.4%, and the cache power drops by 91.4%. After we increased the LLC size to eight times, the results show that the cache's miss ratio becomes 94.5% decreased, and the system performance further improves by 32.1%, while the main memory gains a benefit of 73.4% power consumption saved, as well. As PCM technology develops, we believe that the PCM-based UL^{3}C will not merely stay on theoretical research. Its realization is going to bring a significant renovation to computer systems.

In this work, we have to admit that the endurance problem of the PCM could be a challenge. The PCM has limited endurance by far, as do the other new nonvolatile memory technologies [42]. The cache is a place where the CPU frequently accesses. Even though UL^{3}C is at the last level, it should be durable and able to tolerate high-frequency reads and writes. To overcome this obstacle, much research has been performed, and it has proved that the PCM's endurance has much room for improvement. An example is the noble content-aware bit shuffling (CABS) technique proposed by Miseon Han et al [43]. Therefore, in our future research, we can concentrate on trying to develop a more effective method to maximally prolong the lifetime of the PCM.

Acknowledgements

This work was supported by the National Key Research and Development Program of China (2017YFA0206101, 2017YFB0701703), "Strategic Priority Research Program" of the Chinese Academy of Sciences (XDA09020402), National Integrate Circuit Research Program of China (2009ZX02023-003), National Natural Science Foundation of China (61376006, 61401444, 61504157, 61622408), Science and Technology Council of Shanghai (17DZ2291300).

References

[1] AI: scaling neural networks through cost-effective memory expansion. <https://www.hpcwire.com/2017/06/26/ai-scaling-neural-networks-cost-effective-memory-expansion-2/>.

[2] S. Mittal, R. Wang, J. Vetter, DESTINY: A comprehensive tool with 3D and multi-level cell memory modeling capability, Journal of Low Power Electronics & Applications 7(3)(2017) 23.

[3] M. Imani, A. Rahimi, Y. Kim, T. Rosing, A low-power hybrid magnetic cache architecture exploiting narrow-width values, in: Proc. Non-Volatile Memory Systems and Applications Symposium, 2016.

[4] S. Wang, A. Pan, O.C. Chi, P. Gupta, Tunneling negative differential resistance-assisted STT-RAM for efficient read and write operations, in: Proc. IEEE Transactions on Electron Devices, 2016.

[5] M. Jerry, P.Y. Chen, J. Zhang, P. Sharma, K. Ni, S. Yu, S. Datta, Ferroelectric FET analog synapse for acceleration of deep neural network training, in: Proc. 2017 IEEE International Electron Devices Meeting (IEDM), 2017.

[6] N. Khoshavi, X. Chen, J. Wang, R.F. Demara, Read-tuned STT-RAM and eDRAM cache hierarchies for throughput and energy enhancement. <https://arxiv.org/abs/1607.08086>, 2016.

[7] S. Manisankar, Y. Chung, P-channel logic 2 T eDRAM macro with high retention bit architecture, International Journal of Circuit Theory and Applications 46(7)(2018) 1416-1425.

[8] M. Sato, Y. Shoji, Z. Sakai, R. Egawa, H. Kobayashi, An adjacent-line-merging writeback scheme for STT-RAM-based last-level caches, in: Proc. IEEE Transactions on Multi-Scale Computing Systems, 2018.

[9] X. Dong, N.P. Jouppi, Y. Xie, PCRAMsim: system-level performance, energy, and area modeling for phase-change RAM, in: Proc. IEEE/ACM International Conference on Computer-Aided Design-Digest of Technical Papers, 2009.

[10] F. Rao, K. Ding, Y. Zhou, Y. Zheng, M. Xia, S. Lv, Z. Song, S. Feng, I. Ronneberger, R. Mazzarello, W. Zhang, E. Ma, Reducing the stochasticity of crystal nucleation to enable subnanosecond memory writing, Science, 358(6369)(2017) 1423-1427.

[11] A. Redaelli, Phase Change Memory: Device Physics, Reliability and Applications, Springer International, Cham, Switzerland, 2018.

[12] M. Sato, Z. Li, R. Egawa, H. Kobayashi, An energy-aware set-level refreshing mechanism for eDRAM last-level caches, in: Proc. 2018 IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS), 2018.

[13] M.T. Chang, P. Rosenfeld, S.L. Lu, B. Jacob, Technology comparison for large last-level caches (L3Cs): low-leakage SRAM, low write-energy STT-RAM, and refresh-optimized eDRAM, in: Proc. IEEE International Symposium on High PERFORMANCE Computer Architecture, 2013.

[14] P. Hammarlund, A.J. Martinez, A.A. Bajwa, D.L. Hill, E. Hallnor, H. Jiang, M. Dixon, M. Derr, Haswell: The Fourth-Generation Intel Core Processor, IEEE Micro 34(2)(2014) 6-20.

[15] J. Doweck, W.-F. Kao, A. Lu, J. Mandelblat, A. Rahatekar, L. Rappoport, E. Rotem, A. Yasin, A. Yoaz, Inside 6th-generation intel core: new microarchitecture code-named skylake, IEEE Micro 37(2)(2017) 52-62.

[16] S. K. Sadasivam, B.W. Thompto, R. Kalla, W.J. Starke, IBM power9 processor architecture, IEEE Micro 37(2)(2017) 40-51.

[17] K. Cho, Y. Lee, Y.H. Oh, G.-C. Hwang, J.W. Lee, eDRAM-based tiered-reliability memory with applications to low-power frame buffers, in: Proc. 2014 International Symposium on Low Power Electronics and Design, 2014.

[18] N. Jing, L. Jiang, T. Zhang, C. Li, F. Fan, X. Liang, Energy-efficient eDRAM-based on-chip storage architecture for GPGPUs, IEEE Transactions on Computers 65(1)(2016) 122-135.

[19] G.W. Burr, MJ. Brightsky, A. Sebastian, H.-Y. Cheng, J.-Y. Wu, S. Kim, N.E. Sosa, N. Papandreou, H.-L. Lung, H. Pozidis, E. Eleftheriou, C.H. Lam, Recent progress in phase-change memory technology, IEEE Journal on Emerging & Selected Topics in Circuits & Systems 6(2)(2016) 146-162.

[20] 3D XPoint™: a breakthrough in non-volatile memory technology. <https://www.intel.com/content/www/us/en/architecture-and-technology/intel-micron-3d-xpoint-webcast.html?wapkw=3d+xpoint>.

[21] G. Navarro, G. Bourgeois, J. Kluge, A.L. Serra, A. Verdy, J. Garrione, M.C. Cyrille, N. Bernier, A. Jannaud, C. Sabbione, M. Bernard, E. Nolot, F. Fillot, P. Noé, L. Fellouh, G. Rodriguez, V. Beugin, O. Cueto, N. Castellani, J. Coignus, V.

Delaye, C. Socquet-Clerc, T. Magis, C. Boixaderas, S. Barnola, E. Nowak, Phase-change memory: performance, roles and challenges, in: Proc. 2018 IEEE International Memory Workshop (IMW), 2018.

[22] INTEL® Optane ™ Memory Series 32GB M.2 80MM. <https://www.intel.com/content/www/us/en/ products/memory-storage/optane-memory/optane-32gb-m-2-80mm.html?_ga=2.117025414.1245578153.1531213450-740856301.1527756137>.

[23] Reimagining the data center memory and storage hierarchy. <https://newsroom.intel.com/editorials/re-architecting- data-center-memory-storage-hierarchy/>.

[24] I.S. Kim, S.L. Cho, D.H. Im, E.H. Cho, D.H. Kim, G.H. Oh, D.H. Ahn, S.O. Park, S.W. Nam, J.T. Moon, High performance PRAM cell scalable to sub-20nm technology with below 4F2 cell size, extendable to DRAM applications, in: Proc. 2010 Symposium on VLSI Technology, 2010.

[25] D. Loke, S.R. Elliott, Breaking the speed limits of phase-change memory, Science 336(6088)(2012) 1566-1569.

[26] M. Nakhkash, H. Bardareh, F. Zokaee, H.R. Zarandi, Designing a differential 3R-2bit RRAM cell for enhancing read margin in cross-point RRAM arrays, in: Proc. Nordic Circuits and Systems Conference, 2017.

[27] Y. Lei, H. Chen, X. Li, Q. Wang, Q.Zhang, J. Hu, X. Li, Z. Tian, Z. Song, Enhanced read performance for phase change memory using a reference column, IEICE Electronics Express 14(5)(2017) 20170032.

[28] D. T. Nguyen, H. Kim, H.-J. Lee, I.-J. Chang, An approximate memory architecture for a reduction of refresh power consumption in deep learning applications, in: Proc. 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018.

[29] K. Tovletoglou, D.S. Nikolopoulos, G. Karakonstantis, Relaxing DRAM refresh rate through access pattern scheduling: a case study on stencil-based algorithms, in: Proc. 2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS), 2017.

[30] Deep Power Down (DPD) TN - micron technology. <https://www.micron.com/~/media/documents/ products/technical-note/dram/e0598e21.pdf>, 2014.

[31] J. Choe, Intel 3D XPoint memory die removed from Intel OptaneTM PCM (Phase Change Memory). <http://www. techinsights.com/about-techinsights/overview/blog/intel-3D-xpoint-memory-die-removed-from-intel-optane-pcm/>, 2017.

[32] K. Kilbuck, CMOS under the array. <https://www.micron.com/about/blogs/2015/december/cmos-under-the-array>, 2015.

[33] Introducing 2nd Generation Micron® Mobile TLC 3D NAND. <https://www.micron.com/~/media/documents/ products/presentation/micron-mobile-tlc-3d-nand-launch-deck.pdf>.

[34] T. Tanaka, M. Helm, T. Vali, R. Ghodsi, 7.7 A 768Gb 3b/cell 3D-floating-gate NAND flash memory, in: Proc. IEEE International Solid-State Circuits Conference, 2016.

[35] P.J. Nair, C. Chou, B. Rajendran, M.K. Qureshi, Reducing read latency of phase change memory via early read and turbo read, in: Proc. IEEE International Symposium on High PERFORMANCE Computer Architecture, 2015.

[36] N. Xu, J. Wang, Y. Deng, Y.Y. Lu, B.J. Fu, W. Choi, U. Monga, J. Jeon, J. Kim, K.-H. Lee, E.S. Jung, Multi-domain compact modeling for GeSbTe-based memory and selector devices and simulation for large-scale 3-D cross-point memory arrays, in: Proc. Electron Devices Meeting, 2017.

[37] W.J. Wang, L.P. Shia, R. Zhao, K.G. Lim, H.K. Lee, T.C. Chong, Y. H. Wu, Fast phase transitions induced by picosecond electrical pulses on phase change memory cells, Applied Physics Letters 93(4)(2008) 65.

[38] A. Patel, F. Afram, S. Chen, K. Ghose, MARSS: a full system simulator for multicore x86 CPUs, in: Proc. 2011 48th ACM/EDAC/IEEE Design Automation Conference (DAC), 2011.

[39] P. Rosenfeld, E. Cooper-Balis, B. Jacob, DRAMSim2: a cycle accurate memory system simulator, IEEE Computer Architecture Letter 10(1)(2011) 16-19.

[40] 2Gb: x4, x8, x16 DDR3 SDRAM - Micron Technology. <https://www.micron.com/~/media/documents/products/ data-sheet/dram/ddr3/2gb_ddr3_sdram.pdf>.

[41] X. Zhan, Y. Bao, C. Bienia, K. Li, PARSEC3.0: a multicore benchmark suite with network stacks and SPLASH-2X, ACM Sigarch Computer Architecture News 44(5)(2017) 1-16.

[42] J. Boukhobza, S. Rubini, R. Chen, Z. Shao, Emerging NVM: a survey on architectural integration and research challenges, ACM Transactions on Design Automation of Electronic Systems (TODAES) 23(2)(2018) 14.

[43] M. Han, Y. Han, S.W. Kim, H. Lee, I. Park, Content-aware bit shuffling for maximizing PCM endurance, ACM Transactions on Design Automation of Electronic Systems (TODAES) 22(3)(2017) 48.