Performance/Thermal-Aware Design of 3D-Stacked L2 Caches for CMPs

GUANGYU SUN, Peking University
HUAZHONG YANG, Tsinghua University
YUAN XIE, Pennsylvania State University

13

Three-dimensional (3D) stacking technology enables integration of more memory on top of chip multiprocessors (CMPs). As the number of cores and the capacity of on-chip memory increase, the Non-Uniform Cache Architecture (NUCA) becomes more attractive. Compared to 2D cases, 3D stacking provides more options for the design of on-chip memory due to numerous advantages, such as the extra layout dimension, low latency across layers, etc. On the other hand, 3D stacking aggravates the thermal problem due to the increase of power density. In this work, we first study the design of 3D-stacked set-associative L2 caches through managing the placement of cache ways. The evaluation results show that the placement and corresponding management of 3D cache ways have an impact on the performance of CMPs. Then, we show that the efficiency of thermal control is also related to the placement of cache ways. For caches implemented with different memory technologies, the placement and management of cache ways have different effects on power consumption and power distribution. Consequently, we propose techniques to improve the efficiency of thermal control for different memory technologies. The evaluation results show the trade-off between performance and thermal control efficiency.

Categories and Subject Descriptors: B.3.0 [**Memory Structures**]: General; C.1.0 [**Processor Architectures**]: General

General Terms: Design, Performance

Additional Key Words and Phrases: 3D, NUCA, thermal control, data migration, power gating

ACM Reference Format:
Sun, G., Yang, H., and Xie, Y. 2012. Performance/thermal-aware design of 3D-stacked L2 caches for CMPs. ACM Trans. Des. Autom. Electron. Syst. 17, 2, Article 13 (April 2012), 20 pages.
DOI = 10.1145/2159542.2159545 http://doi.acm.org/10.1145/2159542.2159545

1. INTRODUCTION

Diminishing returns from increasing clock frequency and exploiting instruction-level parallelism in a single processor have led to the advent of chip multiprocessors (CMPs) [Albonesi and Koren 1997; Davis et al. 2005; Kongetira et al. 2005]. The integration of multiple cores on a single chip is expected to accentuate the already daunting "memory wall" problem [Albonesi and Koren 1997; Burger et al. 1997]. Three-dimensional (3D) integration technology provides an opportunity to stack large memory directly on top of logic cores, which could help alleviate the memory bandwidth challenges in CMPs

This work is supported in part by SRC grants, NSFC 61028006 and 61021001, NSF 0905365, 0903432, and 1017277.
Authors' addresses: G. Sun, Center for Energy Efficient Computing and Applications, Peking University; email: gsun@cse.psu.edu; H. Yang, Department of Electrical Engineering, Tsinghua University; Y. Xie, Department of Computer Science and Engineering, Pennsylvania State University.

DOI 10.1145/2159542.2159545 http://doi.acm.org/10.1145/2159542.2159545

[Davis et al. 2005; Xie et al. 2006]. In addition, 3D-stacked memory can further improve performance by reducing access latency and increasing memory bandwidth. Recently, there has been active research of stacking caches on top of processor cores [Li et al. 2006; Madan et al. 2009; Xu et al. 2009].

As the capacity of caches increases with the number of cores in CMPs, the Non-Uniform Cache Architecture (NUCA) becomes more attractive [Kim et al. 2002; Li et al. 2006; Madan et al. 2009]. In a NUCA cache, the organization of data storage has an impact on the performance of CMPs, because the data access latency depends on the position of data, relative to the accessing core. When we consider the design of a 3D-stacked NUCA cache, the extra layout dimension provides more options for how to place data in caches. In addition, the fast access speed through layers and the high bandwidth between logic processing cores and caches accentuate the management of caches for high performance CMPs.

It is also known that 3D integration may increase power density. The design and management of 3D caches should facilitate thermal control techniques, in order to reduce the peak temperature efficiently. For example, the selective-working method [Albonesi 1999] can be employed to reduce the leakage power consumption of SRAM caches. In this method, only a portion of the cache is working; the rest may be shutdown to save power so that the temperature is reduced. When applying such a method to a 3D-stacked cache architecture, the case is quite different from that of 2D caches. First, the decision of selecting which part of the cache should be shutdown not only depends on its position inside a horizontal cache layer but also is related to its layer location, because of thermal resistance between layers. Second, the efficiency of applying such thermal control techniques might vary for different 3D cache designs.

In addition, for caches implemented with different memory technologies, the design and management of 3D architecture has an impact on power consumption and power distribution. For example, nonvolatile memory (NVM) with very low leakage power consumption is also employed as caches in recent research [Joo et al. 2010; Sun et al. 2009; Wu et al. 2009]. In contrast with cases in conventional SRAM caches, dynamic power consumption dominates in these NVM caches. The total power consumption and power distribution vary a lot for different 3D cache configurations. Consequently, distributions of hotspots are also different from those in 3D SRAM caches, and corresponding techniques are required to apply thermal control efficiently.

In this work, we explore the design space of shared 3D-stacking L2 caches based on the network-on-chip (NoC) structure. Several different cache designs are presented and studied through managing placements of the cache ways. The contributions of this article are listed as follows.

— Our study shows that there are more options for placing cache ways in 3D-stacked caches compared to 2D caches. The cache way placement has an impact on the performance and thermal control of 3D-stacked caches.
— For memory technologies with high leakage power consumption, such as SRAM, reduction of leakage power is the pivot of thermal control. A technique called *shadow tags* is proposed to facilitate thermal management in the 3D cache architecture.
— For NVMs with high-access energy consumption, it is more important to control the power distribution so that the temperatures of hotspots are reduced. Consequently, the data migration policy is adapted to flatten the power distribution for these 3D NVM caches.
— Among all proposed placements of 3D cache ways, the experimental results show a trade-off between performance and thermal control.

The rest of this article is organized as follows. Section 2 presents the background of 3D integration technology and the 3D cache architecture used in this work. Magnetic Random Access Memory (MRAM), which is used for 3D NVM cache design, is also introduced. Section 3 compares different cache way placement methods and their impact on performance of processors. In Section 4, power consumption and distribution of 3D caches with different memory technologies are studied, and two techniques are proposed for efficient thermal control. Finally, we conclude the paper in Section 5.

2. PRELIMINARIES

In this section, a brief introduction to the 3D integration technology is first provided. Then, MRAM technology is introduced and compared to the traditional SRAM. Finally, the 3D cache architecture of this work is presented.

2.1. 3D Integration Technology

As process technology scales, the interconnect has emerged as a dominant source of circuit delay and power consumption. The reduction of the interconnect delay and power consumption is critical for deep submicron designs. 3D ICs have recently emerged as promising means for mitigating these interconnect-related problems [Ababei et al. 2005; Davis et al. 2005; Joyner and Meindl 2002; Xie et al. 2006]. Many 3D integration technologies have been explored recently, including wire-bonded, microbump, contactless, and through-silicon-via (TSV) vertical interconnects [Davis et al. 2005].

3D ICs offer a number of advantages over traditional two-dimensional (2D) designs [Davis et al. 2005].

(1) Shorter global interconnects because the vertical distance (or the length of TSVs) between two layers is usually in the range of 10 μm to 100 μm [Xie et al. 2006], depending on manufacturing processes.
(2) Higher performance because of reduced average interconnect length, as well as bandwidth improvement due to die stacking.
(3) Lower interconnect power consumption due to wire-length reduction.
(4) Higher packing density and smaller footprint.
(5) Support for the low cost integration of mixed-technology chips (e.g., MRAM stacking on top of CMOS processor cores).

2.2. MRAM Technology

The basic difference between MRAM and conventional RAM technologies (such as SRAM/DRAM) is that the information carrier of MRAM is a Magnetic Tunnel Junction (MTJ), instead of electric charges [Hosomi et al. 2005; Zhao et al. 2006]. Each MTJ contains two *ferromagnetic* layers and one *tunnel barrier* layer. One of the ferromagnetic layers (reference layer) has fixed magnetic direction, while the other (free layer) can change its magnetic direction by an external electromagnetic field, or a spin-transfer torque. If the two ferromagnetic layers have different directions, the MTJ resistance is high, indicating a "1" state; if the two layers have the same direction, the MTJ resistance is low, indicating a "0" state.

The MRAM technology discussed in this article is called *Spin-Transfer Torque RAM* (STT-RAM), which has the advantage of scalability. In STT-RAM memory cell design, the most popular structure is composed of one NMOS transistor as the access controller and one MTJ as the storage element ("1T1J" structure) [Hosomi et al. 2005]. As illustrated in Figure 1, the storage element MTJ is connected in series with the NMOS transistor. When a *read* operation happens, the NMOS is turned on, and a voltage is applied between the bit line (BL) and the source line (SL). This voltage will cause a current passing through the MTJ, but it will not invoke a disturbed write operation.

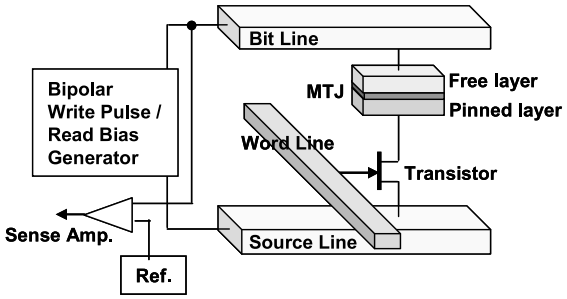


Fig. 1. An illustration of an MRAM cell.

The value of the current is determined by the equivalent resistance of MTJs. A sense amplifier compares this current with a reference current and then decides whether a "0" or a "1" is stored in the selected MRAM cell. When a *write* operation happens, a positive voltage difference is established between SLs and BLs for writing a "0", or a negative voltage difference is established for writing a "1". The current amplitude required to ensure a successful status reversal is called *threshold current*. The current is related to the material of the tunnel barrier layer, the writing pulse duration, and the MTJ geometry [Diao et al. 2007].

2.3. 3D Cache Architecture

As cache sizes increase, the wire delay in deep submicron designs have made Non-Uniform Cache Architectures (NUCA) [Kim et al. 2002] more attractive than ones with uniform access latency. In a NUCA L2 cache, instead of a large uniform monolithic cache, the L2 space is divided into multiple banks which have different access latencies according to their locations relative to a core. Extending the cache simulator CACTI 6.0 [Muralimanohar et al. 2007], which was originally developed for caches in 2D chips, we developed a NoC-based 3D NUCA cache model that incorporated features of 3D integration technology. The key concept is to use NoC routers for communications within a 2D layer while using a special through-silicon-bus (TSB) to communicate vertically between layers [Li et al. 2006].

The example in Figure 2 illustrates a conceptual view of the 3D NUCA structure, which has been used in prior work [Li et al. 2006; Madan et al. 2009; Sun et al. 2009; Xu et al. 2009]. There are four cores located in the bottom layer closest to the heat sink (not depicted). In each core, there is a cache controller connected to a through-silicon-bus (TSB) from which data is moved through layers between processing cores and caches. The TSBs are implemented with TSVs. This bus structure has the advantage of short connections provided by 3D stacking. The latency of traversing on a TSB through several layers is negligible compared to that between two NoC routers in 2D [Loi et al. 2006]. With these buses, the processing cores can have the same access latency to banks in the same location of different layers. Furthermore, hybridization of the NoC router with the bus requires only one additional link (instead of two) on the NoC router, because the bus is a single entity for communicating both up and down [Li et al. 2006].

As shown in Figure 2, processing cores and L2 caches are placed in separate layers. There are 16 cache banks in each layer, which are connected with network-on-chip routers. In this work, the cache banks can be implemented with either SRAM or MRAM technologies. Prior work has shown that such a 3D architecture is considered feasible and efficient for current 3D integration technologies [Xu et al. 2009]. For communication between memory and cores, we provide one TSB for each core to

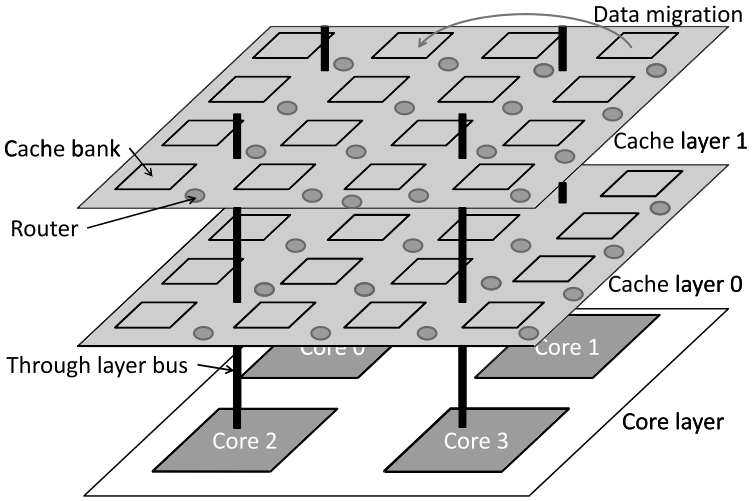


Fig. 2. An illustration of 3D NUCA structure. One core layer, two caches layers, four processing cores, four through-layer-buses, and 32 cache banks.

Table I. Area, Access Time, and Energy Comparison (65nm Technology)

Bank Size	Bank Area	Read Latency	Write Latency	Read Energy	Write Energy	Leakage Power
128KB SRAM	3.62 mm^2	2.252 ns	2.264 ns	0.895nJ	0.797nJ	0.48W
512KB MRAM	3.30 mm^2	2.318 ns	11.024 ns	0.858nJ	4.997nJ	0.059W

benefit from the high bandwidth advantage of 3D stacking. Considering that the TSV pitch size is reported to be only 4–10μm [Loi et al. 2006], thus, even a 1024-bit bus (much wider than our proposed TSB) would only incur an area overhead of 0.32mm^2. In our study, the die area of a 4-core CMP is estimated to be 50mm^2 (discussed later). Therefore, it is feasible to assign one TSB for each core, and the TSV area overhead is negligible.

2.4. Configurations and Assumptions

In this work, our configuration assumes a four-core in-order processor using Ultra SparcIII ISA. The capacity of stacked caches in each layer is based on the areas of caches and cores. We investigated designs such as Sun UltraSPARC T1 [Kongetira et al. 2005], from which we approximated the area of four processing cores as about 50mm^2. The areas of SRAM caches banks are obtained directly from our extended CACTI 6.0 [Muralimanohar et al. 2007]. Therefore, we assume the cache size per cache layer is 2MB. Other configurations are detailed in Table II. The power of each core is also estimated based on data sheets, and the power of an SRAM cache bank is also simulated in CACTI. For MRAM caches, we extend the model in CACTI and adjust it with the MRAM data from the industrial field. Consequently, we develop our MRAM-based CACTI for the simulation in this work. Table I compares the parameters between an SRAM cache bank and an MRAM cache bank.

We use the Simics tool set [Magnusson et al. 2002] for performance simulations. Our 3D NUCA architecture is implemented as an extended module in Simics. As the 3D NUCA is NoC-based, the access latency to the L2 cache is related to data congestion in NoC. Due to the limitation of infrastructure, the accurate congestion in NoC is not modeled in the cache module. Instead, the approximation method in CACTI

Table II. Configuration Parameters

Four cores, 3GHz, 6W/core
Issue Width per core = 1 (in order)
L1-SRAM (private I/D), 16+16KB, 2-way, 64B, 2cycle
L2-SRAM (shared), 4MB, 16-way, 64Bytes, 32banks, 4cycle/bank
L2-MRAM (shared), 16MB, 16-way, 64Bytes, 32banks, 4cycle/bank
Main Memory: 4GB, 300-cycle latency
Number of cache layers = 2 (16banks/layer)
Number of TSB = 4
Hop latency: TSB, V_hop and H_hop = 1 cycle
Router Latency = 2 cycle (without congestion)

6.0 is adapted [Muralimanohar et al. 2007]. The average congestion of each benchmark is evaluated and converted to an extra latency of the router. With this 3D NUCA module, diverse benchmarks from different suites are used in this work. There are some multithreaded benchmarks from SPEC OpenMP2001 and NPB3.2, and there is one thread running on each core when simulated. We also choose benchmarks from SPEC2006 and run them as multiprogrammed workloads (named *mix*). These benchmarks have different transaction intensities to the L2 caches, and therefore provide pervasive simulation results.

The thermal model of the whole process includes a detailed model of cache banks, processing cores, full package, a cooling solution, etc. Then, a popular thermal tool Hotspot 4.1 [Huang et al. 2004] is employed for simulation. The 3D wafer used in this work is bulk-based. The ambient temperature is set to 45°C. We explored a range of thermal conductivities (0.5Watts/meterKelvin–3W/mK) and reported average values of the explored range for the back-end/wiring-layers and the interlayer interconnect layers for various 3D alternatives. In order to evaluate temperatures accurately, we use an iterative method to get runtime leakage power of caches. An initial temperature is assumed and the leakage power of caches is used to calculate a new temperature distribution in Hotspot 4.1. Then the power distribution is updated based on the new temperature. These two steps are iterated until a stable temperature is reached.

3. CACHE WAY PLACEMENT

As introduced in the previous section, the access latency from a processing core to a cache bank of 3D NUCA depends on their relative locations. Obviously, the design and management of cache way placement have an impact on the performance of the CMP. In contrast with the 2D case, the usage of TSBs ensure that accessing cache banks, which are locate in the same position of different layers, takes the same time. Thus, it provides more options for cache way placement with similar performance. In addition, the advantage of high bandwidth is enabled by connecting each core to the NUCA cache separately through its own TSB. The NoC-based structure also enforces parallel communications between processing cores and L2 caches with flexible access routines, which are not available in 2D cases.

In this section, we introduce several placements of cache ways and corresponding management policies. The impact of cache way placement on performance is analyzed. Note that in the performance evaluation, we will not differentiate the MRAM case from the SRAM case. Although the capacities are different for SRAM and MRAM 3D caches, the access latency to each bank is similar in these two cases. The experimental results show similar trends, and we draw the same conclusions about the performance of CMPs using either SRAM or MRAM 3D caches.

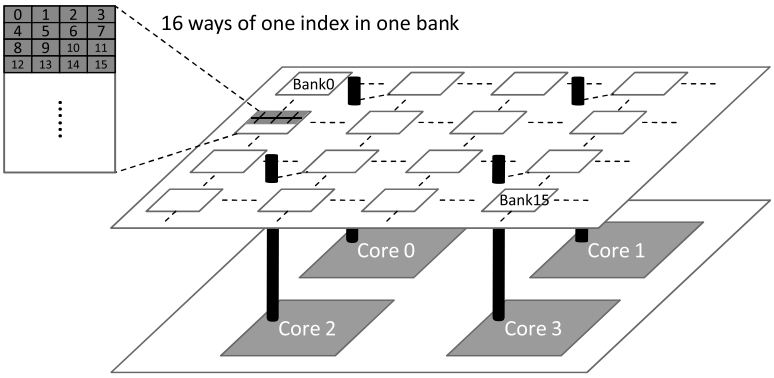


Fig. 3. The baseline cache design. The 16 ways of each cache index are located in the same cache bank.

3.1. Baseline Placement

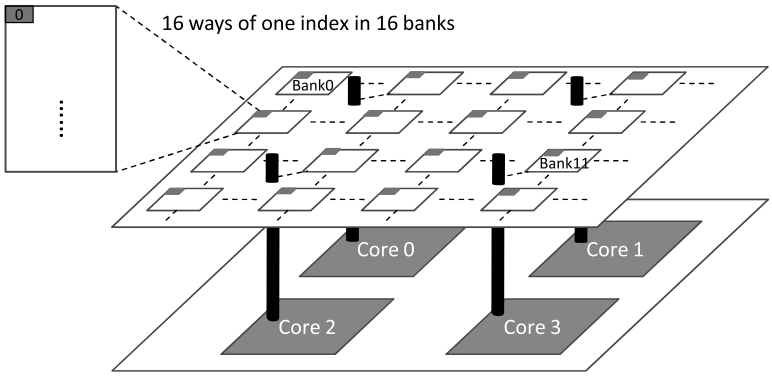
The baseline cache design is shown in Figure 3. The 16 ways of one cache index are placed in the same cache bank. This method of placing all ways together is normally employed in the modern 2D cache design because the address decoding is simple. One access requirement is only delivered to a single cache bank, and the 16 ways are accessed together. Note that only one port of the cache is required to access all 16 cache ways, and the output data are chosen from these cache ways through the comparison of cache tags. With this baseline placement, the data are stored in a fixed cache bank based on its memory address. Then, it takes fixed latency for a core to access all 16 cache ways of the same cache index.

Such a cache placement is very common in the 2D processor with a single core and a uniform L2 cache. However, the case is much different for a CMP using the 3D NUCA. Placing all cache ways of a cache index in one cache bank could cause inefficient cache access so that the performance is degraded. For example, as shown in Figure 3, the cache indexes in bank 0 are the farthest ones from Core 3; those in bank 15 are the farthest ones from core 0. If core 3 frequently accesses cache lines located in cache bank 0, and core 0 frequently accesses data located in cache bank 15, it takes the longest latency for these two cores to access the two cache banks. The cache request from a core needs to travel through five routers to reach the cache bank. Thus, the performance can be improved if the placement of accessed cache lines in two banks could be exchanged. However, the cache access patterns of a core may vary a lot when different applications are running on the core. The access pattern may even change during the runtime of a single application. Therefore, it is impossible to find an optimized placement of all ways of a cache index together in such a baseline 3D NUCA.

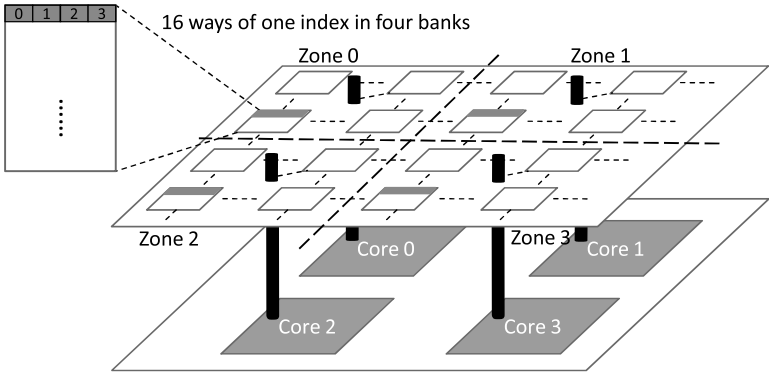
3.2. Distributed Placements

In order to alleviate the inefficient cache access in the baseline placement, distributed placements of cache ways are proposed. The key idea is to spread the cache ways of each cache index throughout the cache layer so that there are at least several ways of a cache line, which are located close to each core. Thus, the data are not allocated to a fixed bank and may be stored in a cache bank close to the processing core.

The most straightforward method is to distribute all 16 ways uniformly throughout a cache layer. Therefore, there is one way of a cache index located in each bank, which is shown in Figure 4(a). This method of placement is called *uniform-distribution* of cache ways in this article. Note that we use an equal number of cache ways and



(a) Uniform distribution.



(b) Core-based distribution.

Fig. 4. Two distributed placements.

cache banks to simplify the illustration, and such a uniform distribution can always be adopted, even when the two numbers are not equal to each other.

For each cache index, there are several ways of cache lines that are closed to each core. In our 3D cache architecture, each TSB is connected to a router of one cache bank. For example, in Figure 4(a), the lower-right TSB is directly connected to a router belonging to the cache bank that contains cache way 11. Therefore, it takes only one hop for a core to access the closest cache line of each cache index. For each core, it is possible to place the frequently accessed data to the closest bank, no matter to which cache index the data is mapped. Then, the average access latency is reduced.

Management of the cache controller can be improved to facilitate such a distributed placement. In a cache using the baseline placement, when data is fetched into an invalid (empty) cache line, any cache can be chosen as the candidate, because all invalid cache ways are located in the same bank. However, in a cache using the distributed placement, the data should be fetched into an invalid cache line close to the accessing core. For example, we assume that the cache ways of a cache index in bank 0 and bank 11 of Figure 4(a) are both invalid when the lower-right core needs to load some data into this cache index. The data should be stored in the cache way of bank 11 rather than in that of bank 0 in order to keep data close to the accessing core. The priority of each way is different for each core, and the cache controller of each core should be aware of it. Note that this is just a modification of fetching into empty cache lines; the

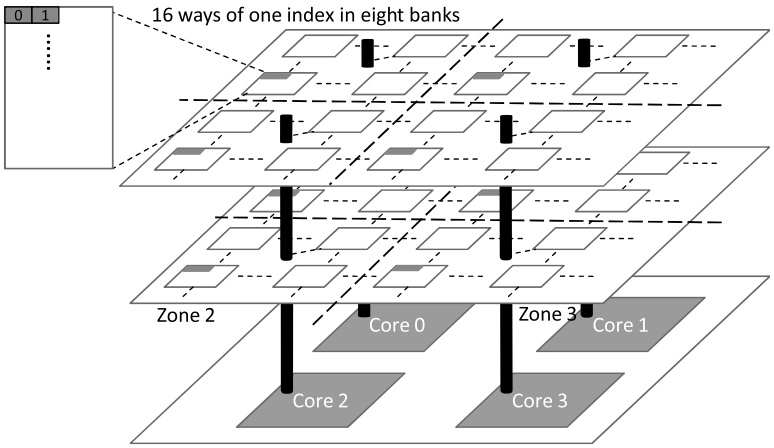


Fig. 5. A cache design with interlayer core-based-distribution placement.

cache line replacement policy is not modified, which is Least Recently Used (LRU) in this work.

Using uniform distribution, however, induces overhead of broadcasting the same cache request to all 16 cache banks. Besides the extra time for broadcasting, it may also aggravate the congestion of the NoC, especially for the application with intensive access to the L2 cache. Therefore, we propose the second method of distribution placement, *core-based distribution*.

Core-base distribution is illustrated in Figure 4(b). Each cache layer is equally divided into four zones. The number of zones is decided by the number of cores. For our configuration described in Table II, there are four cache banks in each zone, which are directly located on top of each core. In this method, we do not distribute 16 ways of a cache index in all 16 cache banks. Instead, the 16 cache ways are divided into four parts, and the four ways of each part are located to one bank inside each zone. For example, in Figure 4(b), cache ways 0–3 of a cache index are located in zone 0.

For each cache index, it is easy to find that there are still some cache ways closed to each core. If these cache lines can store the data frequently accessed by this core, the average access latency is also reduced. Since the request is now only sent to four cache banks instead of 16, the congestion problem in uniform distribution is significantly alleviated, which could help improve performance.

3.3. Interlayer Placements

The cache way placements described so far are all intralayer methods, meaning that all cache ways of a cache index are located in the same cache layer. As we previously addressed, the multilayers of 3D caches provide more design options for the placement of cache ways. In our 3D architecture, the cache access signal on a TSB can arrive at all cache layers within the same number of cycles. This means that, for cache designs just described, if we can fold a cache bank and place each half separately in two cache layers, the access latency from any core to each cache line remains the same.

Figure 5 illustrates the interlayer core-based-distribution placement of cache ways. Although the access latency to a single cache line is not changed, the 16 ways are now located in eight cache banks instead of four. Note, using interlayer placements of cache ways will induce the broadcast of a request in multiple cache layers. However, our experimental results show that the congestion in each cache layer is not increased much because the request is broadcasted in each layer through different routines.

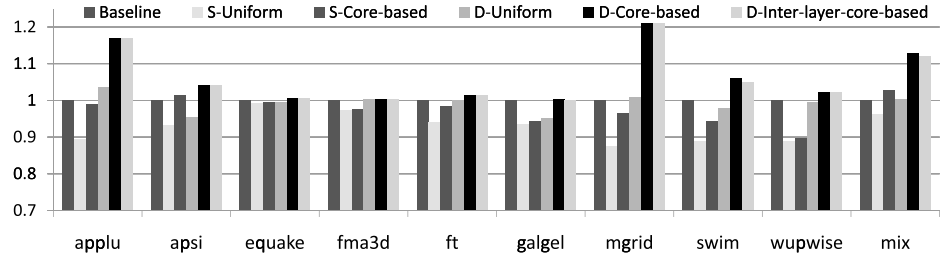


Fig. 6. IPC results of processors using different SRAM cache designs, normalized to the first column. ("S-" means static distributions; "D-" means dynamic distributions; and the first two core-based distributions are intralayer ones.)

Therefore, the performance is kept almost the same when a placement is changed from the intralayer mode to the interlayer mode.

3.4. Data Migrations

Similar to prior approaches [Chishti et al. 2003, 2005; Li et al. 2006], data migration is adopted to move data toward the accessing core. The goal is to move data to the cache line way that is closest to the accessing core. The condition of invoking the data migration depends on the design of real caches. In our L2 3D NUCA, data migration happens when a core accesses a way of a cache line that is not closest to it. The migration policy is tailored to the 3D architecture so that data migrations are limited to the same layer. This limitation can help reduce the congestion of TSBs.

Figure 2 shows an example of data migration after which the core in the upper-left corner can access the data faster. It is easy to find that there is no data migration in the baseline cache design. Data migration for uniform distribution is straightforward. When the data is accessed, it is moved to the cache bank, the router of which is directly connected to the TSV of the accessing core. When applying data migration to the core-based cache distribution, for each cache index, there are four ways of cache lines closest to each core, and the data is migrated to the cache line based on the LRU policy.

Note that the data migration in our 3D architecture is different from the one introduced in the work of Li et al. [2006]. In their 3D cache architecture, a cache line could be migrated to any cache bank, and the core always needs to broadcast the access request to all cache banks to search the data. This strategy will induce congestion, which may cause degradation of performance. On the contrary, in our core-based distribution, the migration only happens among four cache banks that belong to the same index.

3.5. Performance Evaluations

The IPC numbers for running different benchmarks on the CMPs using different cache designs are compared in Figure 6. All results are normalized to those of using the baseline cache design. The methods without data migrations are called *static* distributions, and the others with data migrations are called *dynamic* distributions.

We can find that the static-distributed cache designs cannot improve the performance as expected, and the performance of running some benchmarks is even degraded. This can be explained in two folds. (1) Although there are several ways of cache lines close to each core, the LRU cache replacement policy cannot promise that data frequently accessed are always stored in cache lines close to the processing core. The priority-aware placement mentioned in Section 3.2 places the data close to the accessing core only when the data is fetched for the first time to an empty cache line. (2) The extra overhead of distributed placement may offset the benefits from

reducing access latency. This conclusion is supported by simulation results in Figure 6. The performance is greatly degraded with static-distributed cache way placements when benchmarks *grid*, *swim*, and *wupwise* are running on the CMPs, because these three benchmarks have higher access intensities compared to the others. Therefore, the overhead induced by congestion is severe. We can find that the static core-based method performs better than the uniform method, which also supports the conclusion. One interesting observation is that the performance for benchmark *mix* is improved by using the static core-based distribution, although this benchmark also has intensive cache access, because the *mix* benchmark is composed of several single-thread applications, which data sharing among the cores is rare. There is a higher probability of storing data in cache banks closest to the accessing core.

The results show that we get both benefits and overhead when moving from the baseline placement to the distributed one. In order to improve the performance, data migration is necessary for the distributed placements. Figure 6 also shows the performance results of using different cache designs with the data migration technique. All results are also normalized to those of using the baseline cache design. The results show that performance of CMPs is significantly improved after the data migration mechanism is employed in distributed caches. With data migration, CMPs using the core-based distribution cache achieve higher performance than those using the baseline cache design, but CMPs using the uniform distribution cache still have lower performance than those using the baseline cache, because the overhead induced by uniform distribution is much higher and cannot be offset by the benefits of data migration.

In conclusion, the best performance is achieved when the intralayer core-based distribution with data migration is adopted in the 3D NUCA L2 cache. The performance is almost kept the same when the intralayer core-based distribution is changed to the interlayer mode, meaning that the extra congestion induced in other layers is trivial. However, using the intralayer or the interlayer mode has an impact on the thermal management of the 3D architecture. This is introduced in the next section. Note that the results using MRAM caches are not shown due to page limitation. As we mentioned, we have similar experimental results for MRAM caches, and we can draw the same conclusion with MRAM caches.

4. THERMAL CONTROL OF 3D CACHES

The problem of high power density has always been a challenge for 3D integration technology. As the technology scales down, the increasing leakage power has made the problem even more sever for 3D caches using conventional SRAM technologies. Consequently, the reduction of leakage power is the pivot for thermal control of SRAM 3D caches, which is discussed in Section 4.1. As the potential replacement of SRAM, MRAM has different power and thermal characters in 3D NUCA. In Section 4.2, we will show that it is important to flatten the dynamic power consumption in 3D MRAM caches. The corresponding techniques are proposed for these two cases in order to achieve efficient thermal control.

4.1. Thermal Control of SRAM caches

There has been extensive research on reducing leakage power of SRAM caches in 2D designs [Albonesi 1999; Meng et al. 2005; Powell et al. 2000]. In 3D NUCA, since a single cache layer could be considered a 2D cache, these power-saving methods could be potentially adopted. In this section, we choose the selective-working method [Albonesi 1999] for the thermal control of SRAM caches. We first introduce the basic idea of how it works in our 3D NUCA. Then, we propose a technique for improving this method.

Finally, we discuss the efficiency and runtime cost of applying it to different cache designs in the previous section.

4.1.1. Selective-Working Method. The selective-working method provides the ability to gate (disable) the power of a subset of caches during periods of modest cache activities, while the full cache may remain operational for more cache-intensive periods [Albonesi 1999]. Although CMPs require large on-chip memory, the working set and access intensity to caches are not always at a high level. Especially in the 3D cache architecture, the capacity of L2 caches increases with the number of cache layers. The capacity of the whole cache may be larger than the working set of some applications. For example, in this work, if we shrink the capacity of the L2 cache by a half (e.g. half the cache banks are gated), the cache miss rates of the first four benchmarks in Figure 6 are increased by less than 3%, on average. The performance is almost kept the same. Even for the other applications, the access intensity and runtime working set vary a lot during the whole running process. Consequently, some cache banks may be disabled to reduce power consumption. Note that caches could also be disabled based on the temperature threshold, which is out the scope of this article.

In our 3D NUCA, the cache is controlled in the granularity of the cache bank, meaning that all cache lines inside a bank are powered-on (working) or shutdown (gated) at the same time. Since cache banks are managed individually and only communicated through routers, gating one bank powered-off will not harm the others. Although it is possible to manipulate an individual cache line inside a cache bank, the overhead is much higher, and the design complexity is greatly increased. Furthermore, the temperature is related to the power density. Gating a single cache line of thousands will not change the power density much, which is not efficient.

4.1.2. Shadow Tags. When we apply the selective-working method, the goal is to shutdown as many cache banks as possible without increasing the miss rate much so that performance is kept but the power and temperature are reduced. Therefore, the cache controller should know miss rates before and after gating some cache banks. Then, it can decide how many cache banks should be gated without degrading performance. In order to achieve this, we propose a technique called *shadow tags*, which can predict the runtime miss rate. When the access intensity to the cache is modest, it can predict the miss rate after gating cache banks. Then, these cache banks are disabled if the miss rate is not increased by much. On the other hand, the shadow tag is also aware when the working set increases. Therefore, the disabled cache banks are powered-on again to reduce the miss rate.

A shadow tag is an extra set of cache tags which contains one more state bit, compared to the normal tag. This bit is called a *gating bit* to represent whether the cache line is gated or not. For example, if one cache bank is disabled, gating bits of shadow tags corresponding to cache lines inside the bank are all set to '1', meaning that these cache lines cannot be used. On the contrary, the bits are set to '0' for those cache lines of banks that are still working.

Now, there are two sets of tags in the cache, and shadow tags work similar to normal tags of the cache in predicting the miss rate. Whenever there is a cache access, the request is sent to both normal and shadow tags. A counter is needed to record the number of cache misses of shadow tags during a period. The counter is reset to zero periodically in order to observe the runtime miss rate. For example, assume the cache is fully operating, and we want to predict the miss rate after turning off half of the cache banks. Instead of directly gating cache banks, we just gate the shadow tags corresponding to those cache lines in the banks, as shown in Figure 7. Then, statistics after gating the cache banks are shown in the counter of the shadow tag. If the miss rate

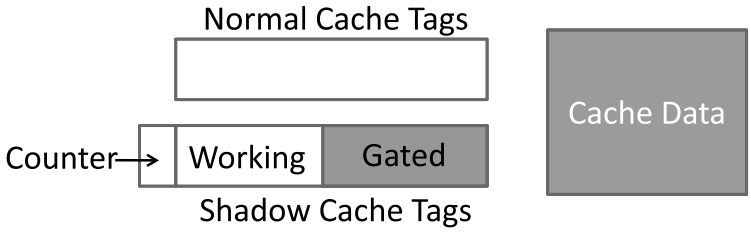


Fig. 7. An illustration of shadow tags.

```
1 SELECT_BANK (){
2   m_c = current miss rate
3   for each working bank_i (in descending order of priorities)
4         turn off shadow tags of bank_i
5         after sampling period t
6         m_g = miss rate after gating bank_i
7         if( (m_g − m_c < threshold)
8               shut down bank_i
9         else turn on shadow tags of bank_i
10  for each gated bank_j (in descending order of priorities)
11        turn on shadow tags of bank_j
12        after sampling period t
13        m_o = miss rate after enabling bank_j
14        if( (m_c − m_o > threshold)
15              enable bank_j
16        else turn off shadow tags of bank_j
17}
```

Fig. 8. The pseudocode of managing cache banks.

does not increase by much, it is safe to shutdown the cache banks for real. Otherwise, the cache banks will keep working, and there is no harm to the performance.

Employing shadow tags, the selective-working technique is managed as the pseudocode shown in Figure 8. In the first part, for each working cache bank that could be gated, the miss rate after its gating is predicted by updating the shadow tags. Then, the predicted miss rate is compared to the current one without gating the cache bank. If the miss rate is not increased much, the real cache bank is gated. In the second part, for each gated cache bank, the miss rate after turning it on is also predicted periodically so that more cache banks are enabled with a large working set.

In 3D NUCA, cache banks in different layers have different priorities for being disabled/enabled, because the heat sink is usually placed next to the core layer. Cache banks in different layers have different contributions to heat dissipation. Consequently, the cache banks in layers far from the heat sink have higher priorities for being power gated, and cache banks close to the heat sink have higher priorities for being turned on.

The counter is reset to '0' after each sampling time t so that the miss rate is predicted periodically. In this work, the period is set to $10k$ instructions, based on experimental results of benchmarks. The threshold is another important parameter which depends on running applications. In this work, it is set to 3% of runtime miss rate to keep the performance degradation less than 0.5% for all benchmarks. Note that cache misses caused by losing data in the gated cache bank are also counted.

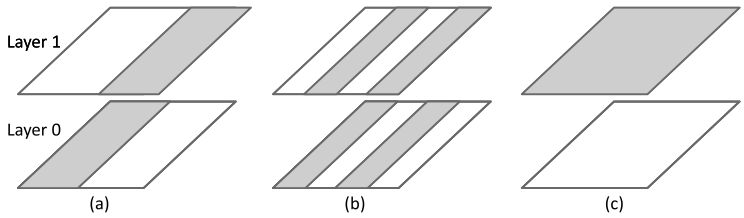


Fig. 9. Three cases of gated caches with the selective-working method (the dark areas are gated banks).

The usage of shadow tags induces trivial overhead. The shadow tag works in parallel with the normal cache tags, and it never interferes with the processing of real data. Thus, it will not degrade performance. Simulation results from CACTI show that the total area of the L2 cache is increased by less than 3% with the extra shadow tags, which is trivial for our 3D cache, because one advantage of 3D technology is saving area. The increase of the total power of the L2 cache is less than 5%. Compared to the power saving by applying the selective method, the power overhead can also be neglected. A 12-bit counter is large enough to count the cache misses in the sampling period. Estimation using the RTL-level synthesis tool shows that the area overhead of the counter can be neglected, compared to the total area of the L2 cache.

4.1.3. Evaluations Results. The efficiency of applying the selective-working technique is related to the choice of gated cache banks. Even when the number of gated cache banks is kept the same, the temperature reduction could vary depending on the locations of the gated cache banks. We show this impact by comparing the peak temperatures when half of the cache banks are gated in different patterns. Note that half of the cache banks are gated to simplify the illustration. In real cases, the numbers of gated cache banks change dynamically. Only the average temperature for all benchmarks is shown. Although the dynamic power is different for each benchmark, the leakage power in CMOS technology is quite dominant. Thus, the peak temperatures for all benchmarks are almost the same, and we only show the average one due to the page limit.

Figure 9 shows three cases of gating half of the cache banks. The total power consumption saved is obviously the same for the three cases. It can also be found that the average power density is reduced to about half of the fully operating cache for these three cases. However, the last case should get the lowest peak temperature, because the active cache banks are all located in the lower cache layer, which is close to the heat sink (not shown in the figure). The corresponding peak temperatures of the processors are calculated and listed in Table III, and the results support our analysis. The results support the conclusion that it is more efficient to disable the cache banks far from the heat sink with the selective-working method.

The design complexity of cache-index decoding may be increased in order to enable the selective-working method. For example, in the baseline 3D NUCA, some cache ways of a cache index in the gated bank need to be moved to working cache banks; otherwise, these indexes are not available because all cache ways of an index are located in the same bank. Thus, address decoding after gating cache banks is different from the original one.

For the same reason, some data are lost after gating cache banks. For example, in the cache design of intralayer core-based distribution, if cache banks in the top cache layer are all gated, half of the caches in the lower layer should be reserved for the cache indexes in the top layer. Therefore, half of the data in the lower cache layer have to be flushed and lost. Table IV compares the design complexity of different cache

Table III. Peak Temperatures of Processors Using Three Cases of Gated Caches

	Case (a)	Case (b)	Case (c)
Temp. (Kelvin)	366.84	362.14	356.73

Table IV. Comparisons of Design Complexity and Data Loss

	extra decoding	data flushing
baseline	yes	yes
uniform	yes	yes
core-based	yes	yes
inter-core	no	no

Note: (Core-based and intercore represent intralayer and interlayer core-based distributions, respectively.)

designs in order to achieve the gated case shown in Figure 9(c). We can find that the interlayer core-based distribution has the lowest design complexity of decoding, and there is the least amount of data loss after gating the cache banks. In the interlayer core-based distribution, even if all the cache banks in the top layer are disabled, the lower cache layer still contains all the cache indexes. Therefore, we do not have to modify the decoding and flush the data for missing cache indexes.

The results in Section 6 show that the performance is kept almost the same when we change the intralayer core-based distribution to the interlayer mode. If we consider both the performance and thermal management, the interlayer core-based distribution is the best design option because of its low design complexity and the least amount of data loss in thermal management. Thus, we use 3D NUCA with interlayer core-based distribution in our experiments of temperature simulation. The results show that the miss rate after applying the selective-working technique with shadow tags is increased by less than 3%, compared to that of using the fully operating cache. The average performance of all benchmarks is degraded by less than 1%. At the same time, the peak temperature is reduced by 7–15 degrees for these benchmarks. We expect that the temperature reduction can be more significant as the number of cache layers increases.

4.2. Thermal Control of MRAM Caches

In this section, we first discuss power consumption and distribution with different cache way replacements in 3D MRAM NUCA. Then, we propose the corresponding thermal control technique.

4.2.1. Power Analysis. As shown in Table I, an MRAM cache has much lower leakage power compared to one of a similar size of a SRAM cache. Thus, the dynamic power consumption has a more significant impact on the total power consumption of MRAM caches (as compared with that of SRAM caches). Furthermore, the energy consumption of a *write* operation to MRAM caches is much higher than that to SRAM caches. As the write-through policy is commonly used in L1 caches of modern processors, the write intensity to L2 caches is normally very high. Thus, in MRAM 3D NUCA, the dynamic power consumption is dominant and is even comparable to the leakage power consumption of SRAM caches [Sun et al. 2009].

Due to dominant dynamic power consumption, the power density distribution of the 3D MRAM NUCA is mainly decided by the access intensity to each bank. If some cache banks are accessed frequently, the power density of these banks can be very

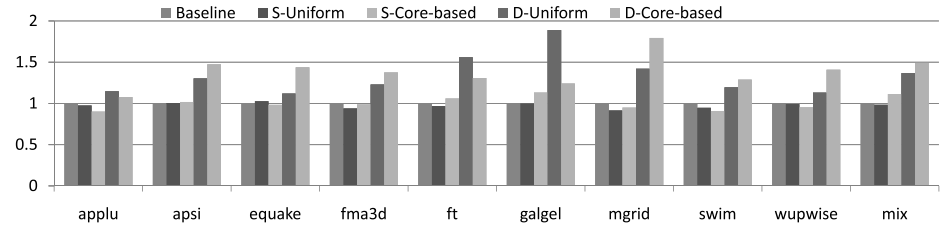


Fig. 10. Total power consumption of MRAM caches with different cache placements, normalized to the first column. ("S-" means static distributions; "D-" means dynamic distributions.)

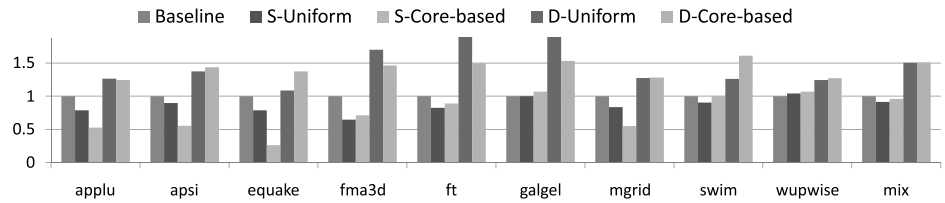


Fig. 11. Variance of MRAM cache bank power density with different cache placements, normalized to the first column. ("S-" means static distributions; "D-" means dynamic distributions.)

high. For some benchmarks in this work, the power density of a frequently accessed cache bank is 1,000x larger than the power density of a cache bank that is seldom accessed. It is known that the temperature distribution is closely related to the power density distribution. Unbalanced power consumption may cause some hotspots in the cache bank with very high power density. In this work, we focus on how to eliminate the cache bank that has a much higher power density than others. In other words, we try to flatten the power distribution among cache banks. Note that the temperature distribution is related to other parameters, such as dimension, thermal resistance, cooling condition, etc, which are also considered in the temperature simulation.

In order to compare the power density distribution for different cache way placements, the access numbers to each cache bank are accessed so that the power density of each cache bank is calculated during the simulation. The comparison of total power consumption is shown in Figure 10 for the different placements discussed in Section 3. For the three cases of static cache way placements, we can find that the difference is not large for all the benchmarks, although the total power consumption varies for different static distributions. When the data migration technique is employed to improve performance, the total power consumption is greatly increased (shown in the last two cases in Figure 10), because data migration induces extra write operations. Since the energy consumption of a write operation to MRAM caches is very high, the total power consumption is greatly increased when the data migration is adopted. Note that the interlayer placement is not considered for MRAM caches, because the selective-working method is not necessary for MRAM caches with low leakage power consumption. The intralayer placements are used in 3D MRAM NUCA to achieve higher performance.

In Figure 11, we calculate the variance of the power density for all cache banks and compare the results among different cache placements. In the figure, a smaller variance means a more uniform distribution of power density among cache banks. Since the total power consumption is similar for static cache way placements, the distributed cache placements with lower variance may help reduce the temperature of hotspots.

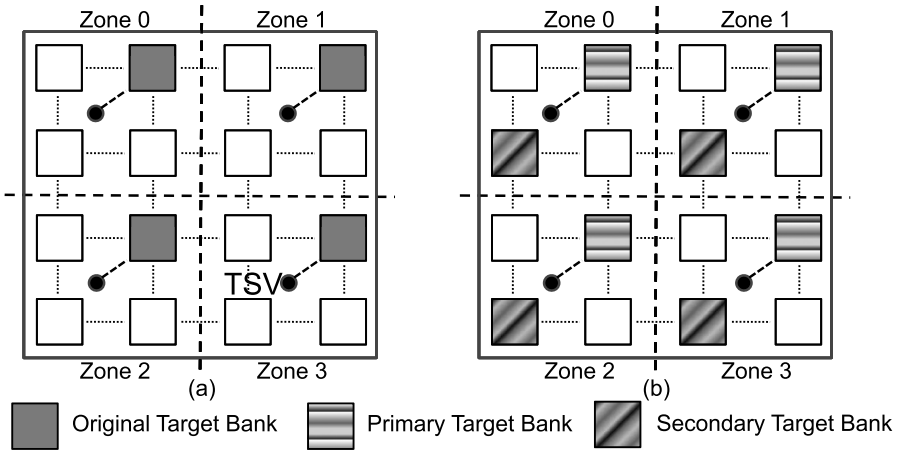


Fig. 12. The planform of a cache layer using (a) original data migration, (b) thermal-aware data migration (core-based distribution).

The results of Figure 11 show that with distributed placement, the cache lines of these indexes are distributed among different cache banks so that the variance of power density is reduced. We can also find that the imbalance of power density is severe in dynamic placements; the reason being that during data migration, data is always moved to the cache bank closest to the accessing core. Thus, such cache banks are far more frequently accessed than others and may cause hotspots with much higher temperatures. In order to mitigate this problem, the thermal-aware data migration policy is proposed in the next section.

4.2.2. Thermal-Aware Data Migration. In the original data migration policy, the data are always moved to cache banks closest to accessing cores. Due to high write energy, the power density of these cache banks is much higher than that of the others. Consequently, hotspots are generated among these cache banks. In order to reduce the power density of these cache banks without inducing much overhead, we propose a technique for improving the data migration policy. In this section, we take the core-based distribution as an example and show how to flatten the power density distribution by thermal-aware data migration.

Figure 12 shows the planform of a cache layer. In each zone of Figure 12(a), the cache bank in dark color is the target cache bank to which the data is migrated in this zone. Because these banks are directly connected to the TSVs, it takes only one hop to access such a bank for the core in the corresponding zone. In order to reduce the power density of these banks, we add an extra candidate for the target cache bank in each zone. As shown in Figure 12(b), there are now two possible target cache banks in each zone. The original target banks in Figure 12(a) are named as primary target banks, and the other target banks are named as secondary target banks.

For each zone, there is only one target bank working at the same time. The two target cache banks in each zone work in turn to avoid a target cache bank being accessing too many times due to data migrations. In order to achieve this mechanism, a threshold is set for the number of data migrations of each target bank. A counter is then needed to record the number of migration operations to each target bank. When the number reaches the threshold, the working target cache bank is switched to the other one in the zone. Consequently, the peak temperature of the hotpot in the primary

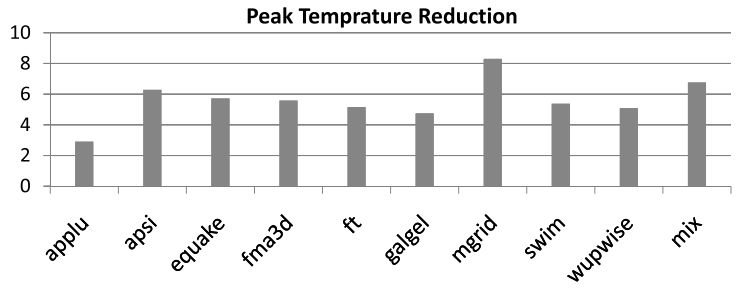


Fig. 13. Peak temperature reduction after using the thermal aware data migration *(core-based-distribution)*.

target bank is reduced. Note that the migration threshold may affect the thermal control. In this work, we set it as $10k$, and it can be adjusted based on real applications.

Since the secondary target cache bank is not directly connected to TSVs, it takes one more hop to access the bank from the core. Thus, performance is degraded a little with thermal-aware data migration. The evaluation results show that, on average, the performance is decreased by less than 2%, because there is only one hop of overhead to access the migrated data. In Figure 13, the peak temperatures after using thermal-aware data migration are compared to those of the original data migration. On average, the peak temperature is decreased by about 5.6 degrees. The maximum temperature reduction is about 8.3 degrees.

5. CONCLUSION

In this work, we explore the design space of 3D L2 caches through managing the placement of cache ways with respect to performance, power consumption, and temperatures. The experimental results show that using the interlayer core-based distribution cache design can achieve the best performance with data migrations. However, when temperature is also considered, we need corresponding adjustments and techniques to apply thermal control efficiently. For SRAM caches with high leakage power, the interlayer placement is preferred, and the shadow tag technique is employed to apply the selective-working method with low overhead. For MRAM caches, the control of dynamic power consumption is important, especially for dynamic placements. We propose thermal-aware data migration to flatten the power density distribution and reduce the peak temperature. The evaluation results show that the performance is degraded with these thermal control techniques, and the research shows a trade-off between performance and thermal control.

REFERENCES

ABABEI, C., FENG, Y., GOPLEN, B., MOGAL, H., ZHANG, T., BAZARGAN, K., AND SAPATNEKAR, S. S. 2005. Placement and routing in 3D integrated circuits. *IEEE Des. Test Comput. 22,* 6, 520–531.

ALBONESI, D. 1999. Selective cache ways: On-demand cache resource allocation. In *Proceedings of the 32nd Annual ACM/IEEE International Symposium on Microarchitecture*. 248–259.

ALBONESI, D. AND KOREN, I. 1997. Improving the memory bandwidth of highly-integrated, wide-issue, microprocessor-based systems. In *Proceedings of the International Conference on Parallel Computing Technologies*. 126–135.

BURGER, D., GOODMAN, J. R., AND KAGI, A. 1997. Limited bandwidth to affect processor design. *IEEE Micro 17,* 6, 55–62.

CHISHTI, Z., POWELL, M. D., AND VIJAYKUMAR, T. N. 2003. Distance associativity for high-performance energy-efficient non-uniform cache architectures. In *Proceedings of the ACM/IEEE International Symposium on Microarchitecture*. 55.

CHISHTI, Z., POWELL, M. D., AND VIJAYKUMAR, T. N. 2005. Optimizing replication, communication, and capacity allocation in CMPs. *SIGARCH Comput. Archit. News. 33*, 357–368.

DAVIS, J., LAUDON, J., AND OLUKOTUN, K. 2005. Maximizing CMP throughput with mediocre cores. In *Proceedings of the International Conference on Parallel Computing Technologies*.

DAVIS, W. R., WILSON, J., MICK, S., XU, J., HUA, H., MINEO, C., SULE, A. M., STEER, M., AND FRANZON, P. D. 2005. Demystifying 3D ICs: The pros and cons of going vertical. *IEEE Des. Test Comput. 22,* 6, 498–510.

DIAO, Z., LI, Z., WANG, S., DING, Y., PANCHULA, A., CHEN, E., WANG, L.-C., AND HUAI, Y. 2007. Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory. *J. Phys. Condens. Matter 19,* 16.

HOSOMI, M., YAMAGISHI, H., YAMAMOTO, T., BESSHO, K., HIGO, Y., YAMANE, K., YAMADA, H., SHOJI, M., HACHINO, H., FUKUMOTO, C., NAGAO, H., AND KANO, H. 2005. A novel non-volatile memory with spin torque transfer magnetization switching: Spin-RAM. In *Proceedings of the International Electron Devices Meeting*. 459–462.

HUANG, W., STAN, M. R., SKADRON, K., SANKARANARAYANAN, K., GHOSH, S., AND VELUSAM, S. 2004. Compact thermal modeling for temperature-aware design. In *Proceedings of the 41st Annual Design Automation Conference (DAC'04)*. 878–883.

JOO, Y., NIU, D., DONG, X., SUN, G., CHANG, N., AND XIE, Y. 2010. Energy- and endurance-aware design of phase change memory caches. In *Proceedings of the Conference and Exhibition on Design, Automation and Test in Europe*.

JOYNER, J. W. AND MEINDL, J. D. 2002. Opportunities for reduced power dissipation using three-dimensional integration. In *Proceedings of the IEEE International Interconnect Technology Conference*. 148–150.

KIM, C., BURGER, D., AND KECKLER, S. 2002. An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*.

KONGETIRA, P., AINGARAN, K., AND OLUKOTUN, K. 2005. Niagara: A 32-way multithreaded sparc processor. *IEEE Micro 25,* 2, 21–29.

LI, F., NICOPOULOS, C., RICHARDSON, T., XIE, Y., NARAYANAN, V., AND KANDEMIR, M. 2006. Design and management of 3D chip multiprocessors using network-in-memory. In *Proceedings of the International Conference on Computer and Their Applications*. 130–141.

LOI, G. L., AGRAWAL, B., SRIVASTAVA, N., LIN, S.-C., SHERWOOD, T., AND BANERJEE, K. 2006. A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy. In *Proceedings of the IEEE/ACM Design Automation Conference*. 991–996.

MADAN, N., ZHAO, L., MURALIMANOHAR, N., UDIPI, A., BALASUBRAMONIAN, R., IYER, R., MAKINENI, S., AND NEWELL, D. 2009. Optimizing communication and capacity in a 3D stacked reconfigurable cache hierarchy. In *Proceedings of the International Symposium on High-Performance Computer Architecture*. 262–274.

MAGNUSSON, P. S., CHRISTENSSON, M., ESKILSON, J., FORSGREN, D., HÅLLBERG, G., HÖGBERG, J., LARSSON, F., MOESTEDT, A., AND WERNER, B. 2002. Simics: A full system simulation platform. *Comput. 35,* 2, 50–58.

MENG, Y., SHERWOOD, T., AND KASTNER, R. 2005. Exploring the limits of leakage power reduction in cache. *ACM Trans. Architec. Code Optim*.

MURALIMANOHAR, N., BALASUBRAMONIAN, R., AND JOUPPI, N. 2007. Optimizing NUCA organizations and wiring alternatives for large caches with CACTI 6.0. In *Proceedings of the Annual ACM/IEEE International Symposium on Microarchitecture*. 3–14.

POWELL, M., YANG, S., FALSAFI, B., ROY, K., AND VIJAYKUMAR, T. 2000. Gated-vdd: A circuit technique to reduce leakage in deep-submicron cache memories. In *Proceedings of the International Symposium on Low-Power Electronics and Design*. 90–95.

SUN, G., DONG, X., XIE, Y., LI, J., AND CHEN, Y. 2009. A novel architecture of the 3D stacked MRAM L2 cache for CMPs. In *Proceedings of the International Symposium on High-Performance Computer Architecture*. 239–249.

SUN, G., WU, X., AND XIE, Y. 2009. Exploration of 3D stacked l2 cache design for high performance and efficient thermal control. In *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design (ISLPED'09)*. 295–298.

WU, X., LI, J., ZHANG, L., SPEIGHT, E., RAJAMONY, R., AND XIE, Y. 2009. Hybrid cache architecture with disparate memory technologies. In *Proceedings of the Annual International Symposium on Computer Architecture*. 34–45.

XIE, Y., LOH, G. H., BLACK, B., AND BERNSTEIN, K. 2006. Design space exploration for 3D architectures. *J. Emerg. Technol. Comput. Syst. 2,* 2, 65–103.

XU, Y., DU, Y., ZHAO, B., ZHOU, X., ZHANG, Y., AND JUN, Y. 2009. A low-radix and low-diameter 3D interconnection network design. In *Proceedings of the International Symposium on High-Performance Computer Architecture*. 30–42.

ZHAO, W., BELHAIRE, E., MISTRAL, Q., CHAPPERT, C., JAVERLIAC, V., DIENY, B., AND NICOLLE, E. 2006. Macro-model of spin-transfer torque based magnetic tunnel junction device for hybrid magnetic-CMOS esign. In *Proceedings of the IEEE International Behavioral Modeling and Simulation Workshop*. 40–43.

Received March 2010; revised September 2010, May 2011; accepted October 2011