VARIATION-TOLERANT NON-UNIFORM 3D CACHE MANAGEMENT IN MEMORY STACKED MULTI-CORE PROCESSORS

by

Bo Zhao

B.S., Electronic & Information Engineering, Beihang University, 2007

Submitted to the Graduate Faculty of
the Swanson School of Engineering in partial fulfillment
of the requirements for the degree of

Master of Science

University of Pittsburgh

2009

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Bo Zhao

It was defended on

June 26th 2009

and approved by

Jun Yang, Associate Professor, Department of Electrical & Computer Engineering

Steven P. Levitan, John A. Jurenko Professor, Department of Electrical & Computer Engineering

Alex K. Jones, Associate Professor, Department of Electrical & Computer Engineering

Thesis Advisor: Jun Yang, Associate Professor, Department of Electrical & Computer Engineering

Copyright © by Bo Zhao
2009

VARIATION-TOLERANT NON-UNIFORM 3D CACHE MANAGEMENT IN MEMORY STACKED MULTI-CORE PROCESSORS

Bo Zhao, M.S.

University of Pittsburgh, 2009

Process variations in integrated circuits have significant impact on their performance, leakage and stability. This is particularly evident in large, regular and dense structures such as DRAMs. DRAMs are built using minimized transistors with presumably uniform speed in an organized array structure. Process variation can introduce latency disparity among different memory arrays. With the proliferation of 3D stacking technology, DRAMs become a favorable choice for stacking on top of a multi-core processor as a last level cache for large capacity, high bandwidth, and low power. Hence, variations in bank speed create a unique problem of non-uniform cache accesses in the 3D space.

In this thesis, we investigate cache management techniques for tolerating process variation in a 3D DRAM stacked onto a multi-core processor. We modeled the process variation in a 4-layer DRAM memory to characterize the latency variations among different banks. As a result, the notion of fast and slow banks from the core's standpoint is no longer associated with their physical distances with the banks. They are determined by the different bank latencies due to process variation. We develop cache migration schemes that utilize fast banks while limiting the cost due to migration. Our experiments show that there is a great performance benefit in exploiting fast memory banks through migration. On average, a variation-aware management can improve the performance of a workload over the baseline (where the speed of the slowest bank is assumed for all banks) by 17.8%. We are also only 0.45% away in performance from an ideal memory where no PV is present.

TABLE OF CONTENTS

1.0 INTRODUCTION 1

2.0 BACKGROUND & RELATED WORK 3

2.1 Process Variations 3

2.2 3D Die Stacking 5

2.3 Non-Unifrom Access of On-Chip Memories 7

3.0 MOTIVATION 9

3.1 3D + Memory 9

3.2 Memory + PV 13

3.3 3D + PV 14

3.4 3D + Memory + PV 15

4.0 ARCHITECTURE & MODELING 16

4.1 3D Stacked Last Level Cache Architecture 16

4.2 PV Modeling 19

4.3 Latency and Leakage Modeling 22

4.3.1 Latency Modeling 23

4.3.2 Leakage Modeling 25

4.3.3 Hardware-Awareness of PV 25

4.4 Energy / Power Modeling 26

4.4.1 CPU Cores 27

4.4.2 L2 Cache, Main Memory, and Crossbar 28

4.4.3 DRAM LLC (L3 Cache) 28

5.0 NON-UNIFORM CACHE MANAGEMENT 32

5.1 Evaluation Settings 32
5.2 Static PV-Tolerant LLC 34
5.3 Dynamic PV-Tolerant LLC 36
5.3.1 Bank Latency Based Migration Policy 36
5.3.2 Tiered Migration Policy 38
5.3.3 Energy Conservation: Intra-Column Two-Tier Migration 41
6.0 CONCLUSION & FUTURE WORK 45
BIBLIOGRAPHY 46

LIST OF TABLES

1 Device parameters from CACTI source code 22

2 Fundamental parameters of Sun UltraSPARC T2 and our design 28

3 Energy / power of a 256KB L2 cache 29

4 Energy / power of a 16GB main memory 29

5 Delay and energy / power of an 8 port, 256-bit wide crossbar 29

6 Subbank energy breakdown 30

7 Energy / power of a 96MB L3 cache (one rank) for read / write evaluation . . 31

8 Energy / power of a 96MB L3 cache (one rank) for migration evaluation . . . 31

9 Baseline chip configurations without PV 33

10 Simulated workloads 34

LIST OF FIGURES

1 Process variations (3σ / Nominal) 3
2 Illustration of WID and D2D variations and their paramenters 5
3 Uniform and Non-Uniform Cache Architecture (UCA and NUCA) 7
4 Planar 3D DRAM and true 3D DRAM 10
5 Subbank components and topology 11
6 From planar memory structure to 3D memory structure 11
7 SRAM cell and DRAM cell 13
8 Architecture of a 5-layer 3D memory stacked on top of an 8-core processor . . 16
9 Sample L_{eff} distribution maps for 4 DRAM cell layers of one chip 21
10 Predicting access latency distribution from gate length distribution 23
11 DRAM data array subbank latency distribution as the result of PV 24
12 Leakage dependency on gate length 26
13 Mapping interconnections from planar DRAM to 3D DRAM 30
14 Performance of the pessimistic baseline 35
15 Performance of the static PV-tolerant LLC 35
16 Illustrations of different dynamic data migration policies 37
17 Performance of the latency-based migration policy 37
18 Performance variations in a 2-tier migration scheme 39
19 Performance summary for a 2-tier migration scheme 39
20 Sensitivity study of tier numbers 40
21 Performance of the proposed 2-tier migration technique 41
22 Percentage of migration activities 42

23 Energy comparison for 2-tier migration and intra-column migration 42
24 Performance of the improved intra-column 2-tier migration technique 43
25 Energy-Delay product results 44

1.0 INTRODUCTION

Scaling and integration for nanoscale CMOS transistors and circuits present promising device density and performance trends. However, one of the major hurdles in scaling in nanometer regime is the difficulty in controlling the device characteristics precisely during fabrication, which leads to *parameter* or *process variations*. Key characteristics in both devices and wires affected by process variations are either structural or electrical [1]. Those parameter variations have significant impact on circuit performance, leakage and reliability. We focus on their impact on circuit performance in this work.

Process variations can be classified into within-die (WID) variations and die-to-die (D2D) variations. WID variations refer to the differences in device features among transistors on one die, while D2D variations refers to such mismatches among different die. Recently, the proliferation of three-dimensional (3D) die-stacked architecture gives rise to considerations in both WID and D2D variations. 3D stacking is a technology that stacks multiple active silicon die on top of each other, and connects them through wafer bonding. The communication among different layers is carried in Through Silicon Vias (TSVs). With this technology, it is possible to either stack multiple 2D die into a powerful 3D chip, or implement a processor using true 3D circuits. In either design choice, process variations impact both the performance within each layer and the communication among different die. That is to say, the performance of the overall 3D processor is constrained by both WID and D2D process variations.

Among various 3D architectures, stacking cache or memory chips directly on top of a 2D multi-core chip [20, 23, 36, 38, 39, 40, 55] has gained popularity. There have been a great amount of efforts on on-chip SRAM cache optimizations under processor variations [11, 15, 16, 45, 46, 62]. However, there is little work on DRAM variations probably because

DRAMs are conventionally off-chip, so their latency changes due to process variation have little impact on the overall performance. However, with the 3D integration of DRAM chips with multi-core chips, such latency variations will become more pronounced, especially where there are both WID and D2D variations in a multi-layered DRAM stack.

In this work, we first model the process variations in a multi-layered 3D DRAM stack integrated with a multi-core processor, then develop memory management techniques to overcome the process variations. Our 3D DRAM architecture follows closely a commercial product from Tezzaron Corporation [21, 22]. From our modeling results, we observe that the DRAM subbanks in 3D space present a fairly wide range of subbank access latencies due to process variations. Such latency variations create a 3D non-uniform access memory for each core on-chip. Similar to the NUCA problem in a 2D CMP, 3D NUCA has a significant impact on the performance of the entire chip. Unlike the 2D NUCA problem, such non-uniformity is dominated by process variations, and less due to the wire delay or interconnection network delay as in a 2D CMP. We develop cache/memory management techniques to overcome the non-uniform access times from different memory subbanks. The basic idea is to migrate data from slow subbanks to fast subbanks to reduce the average DRAM access time. Such migration is shown by our experiments to be very effective, especially when the working set of a workload fits into the fast subbanks due to the large capacity provided by DRAM. In those cases, we even achieve performance improvements over the ideal chip where there are no process variations. We further improve our migration schemes such that the energy increase is minimized. On average, our variation-tolerant migration scheme is 17.8% better than a conservative chip where the speed of the slowest subbank is used as the nominal speed for all subbanks, and only 0.45% away from the performance of the ideal chip. Our ED^2 is also 42% better than a conservative chip and only 3% worse than the ideal chip.

The remainder of this thesis is organized as follows. Chapter 2 introduces the background knowledge within the scope of this work. Chapter 3 further discusses the problems arise from the backgrounds as our motivations. Chapter 4 specifies our system architecture and the modeling of process variations in a true 3D memory organization. Chapter 5 describes our proposed cache migration scheme to overcome process variations. Finally Chapter 6 concludes this paper.

2.0 BACKGROUND & RELATED WORK

2.1 PROCESS VARIATIONS

Process variations (PV) are the deviations from the designed values of certain physical parameters. They come from the poor controllability and imperfection of fabrication processes, as well as wearout mechanisms [1, 41]. PV are becoming relatively larger with technology scaling, due to the increasing difficulty to precisely control the fabrication processes at small feature technologies [2].

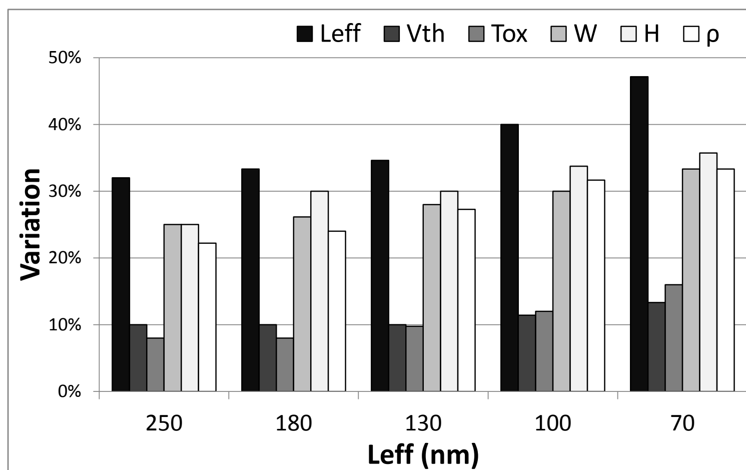

Figure 1: Process variations (3σ / Nominal) based on the prediction in [1]. L_{eff}: effective gate length; V_{th}: threshold voltage; T_{ox}: gate oxide thickness; W: wire width; H: wire height; ρ: resistance per unit length.

Key characteristics in both devices and wires affected by process variations are either structural, such as gate length, wire width, etc., or electrical, such as threshold voltage,

resistance per unit length, etc. [1]. The variation amount of the key six parameters regarding device and wire, and their trends with technology scaling from $250nm$ to $70nm$, are shown in Figure 1. Among all these variations, the effective gate length (L_{eff}) of the device is a major source of variation in CMOS circuits [1, 3, 4, 5], and threshold voltage (V_{th}) variation also draws great attention [4], due to their significant impact on circuit performance, power and reliability. L_{eff} variation is introduced by the non-uniformity in masks and lithography, V_{th} variation is caused by both the density fluctuation of random dopant and L_{eff} variation [2, 4, 6].

From the cause and characteristic point of view, PV can be categorized as systematic and random variations. Systematic variations are repeatable, and they are mostly caused by lithographic aberrations. Thus, they mainly affect L_{eff}. The most important characteristic of systematic variations is spatial correlation, which means that the difference between two devices close to each other is smaller than those that are far apart [5]. In addition, systematic variations also follow the Normal (Gaussian) distribution [2, 6, 11, 41]. Random variations are unrepeatable. They are caused by random doping fluctuation, and therefore, most responsible for V_{th} variations. Random variations are typically assumed to follow the Uniform distribution within the variation range. Studies showed that systematic variations present more impact on performance than random variations [41, 43].

PV can also be classified into with-in-die (WID), die-to-die (D2D, also called with-in-wafer), wafer-to-wafer (W2W, also called with-in-lot), and lot-to-lot (L2L) variations [1, 41]. The WID and D2D variations are most widely studied due to their relatively significant influence on chip performance and yield. WID variations refer to the differences in device features among transistors in one die. Thus, it can result in non-uniform circuit performance and power across a die. D2D variations represent the parameter mismatches between identical die. Thus, it renders the same amount of offset to all transistors across the chip [2]. The relation between WID and D2D can be illustrated in Figure 2. Both follow the Normal distribution with respect to L_{eff}, with the mean value denoted by μ. D2D variations can be seen as the Normal distribution with respect to the μ's of all die.

There have been active efforts recently on WID variation-tolerant designs such as improving die yield [45, 46], improving cache designs [31, 47], improving reliability [48, 49, 50], mit-

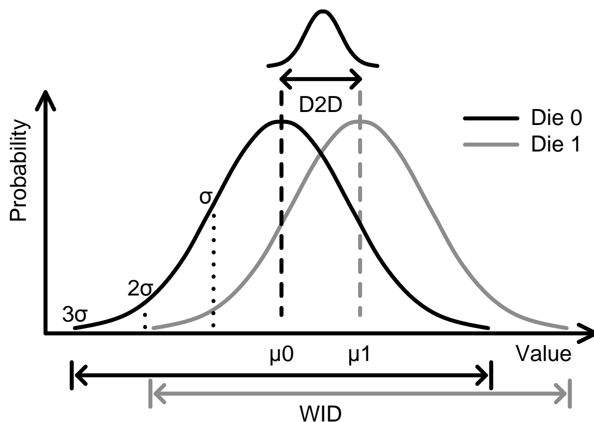

Figure 2: Illustration of WID and D2D variations and their paramenters.

igating latency variations in pipeline stages or various components in a chip [51, 52, 53, 54], and lowering energy consumption or improving energy delay product [26, 30]. However, not much attention has been given to D2D variations mainly because previous emphases were all on a single die. As we will discuss next, the proliferation of 3D die stacked chips will give rise to the considerations of both WID and D2D variations.

2.2 3D DIE STACKING

The continuous pursuit of higher performance has driven designers to put more and more cores, memories, and NoCs on one die. The growing die area and necessity of routing among many functional units and cores lead to very long and complex on-chip interconnection. On the other hand, as technology feature size scales into deep sub-micron territory, devices are improved in speed and power at a faster pace than the interconnection. Hence, the interconnection becomes the dominant factor in delay and power. 3D stacking technology emerges as a solution to this problem. It exploits the vertical dimension of a chip by stacking multiple active silicon die on top of each other and combining them with vertical interconnections which are orders of magnitude shorter than horizontal wires [20]. This technology can dramatically increase transistor density without expanding die footprint, and significantly

reduce the interconnection both within a die and across die in a chip. Although challenges arise with this new technology, 3D stacking promises more flexibility in processor design, and substantial performance improvement and power reduction.

The most widely accepted vertical connection is *through silicon vias* (TSVs) which serve as a high bandwidth, low latency, and low power die-to-die interface. It has been reported that the pitch of a TSV is less than $10\mu m$ [9, 17]. For example, the Tezzaron Corporation claimed that a 14000-TSV/mm^2 density can be achieved in their 3D stacking technology [21]. The MIT Lincoln Laboratory (MITLL) demonstrated a TSV pitch of $5.6\mu m$ at $180nm$ node [7], and a megapixel CMOS image sensor fabricated in 3D technology that incorporated over one million 3D vias within a $22\times22mm^2$ die size [8]. The length of a TSV is $10\sim50\mu m$, in accordance with the thickness of one die layer, which results tiny resistance and capacitance [7, 20, 23, 38]. Therefore, a sub-FO4 delay is usually assumed for TSVs [36]. It has been reported that the vertical latency for traversing the height of a 20-layer stack is merely $12ps$ [40].

Another significant benefit of 3D stacking technology is the possibility of stacking different processes into one chip. For example, special silicon process layers such as memories, special purpose layers such as analog, RF, and layers in other technologies such as MEMS, biomedical etc. can be stacked with the mature CMOS process technology for advanced computing capabilities. Among those, stacking multiple die of various kinds of memories that require special silicon processes such as DRAM [23, 36, 38, 39, 40], Phase-change memory (PCM or PRAM) [25] and Magnetic memory (MRAM) [27] has gained increasing interest recently. Those researches targeted at exploiting the intrinsic advantages of those memories and develop circuit and/or architectural techniques to overcome their drawbacks for better performance and/or lower power consumption of the entire chip. Next, we will introduce the major challenge in integrating large memories on chip.

2.3 NON-UNIFROM ACCESS OF ON-CHIP MEMORIES

As the general trend in processor development moves from single-core to multi-core, and even many-core, the computing capability of a processor embraces multi-tasks and massively parallel applications. This together with fast growing application data set size, demands extremely high on-chip memory capacity. Today's on-chip L2 and L3 caches are already of multi-megabytes, occupying considerable area in high performance microprocessors. For instance, Intel utilized 12MB L2 cache in their 45nm quad-core desktop / notebook processors, and 8MB L3 cache in server processors. The sizes of L2 and L3 caches will continue to grow to satisfy the application's ever increasing demand in on-chip memory size.

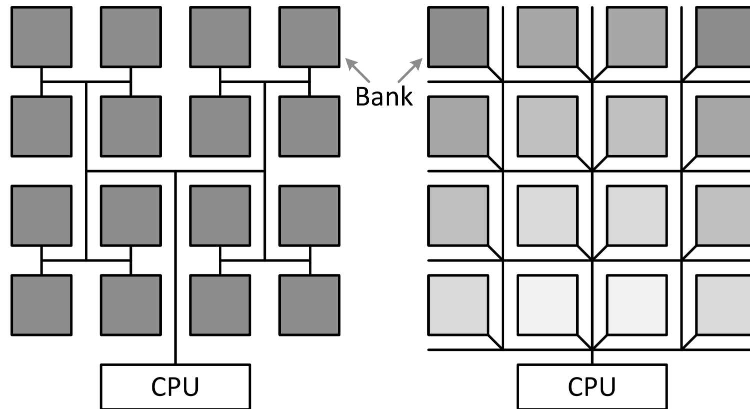

Figure 3: Uniform and Non-Uniform Cache Architecture (UCA and NUCA). Darker color implies longer bank access latency.

Previously, on-chip cache memories were close to CPU and small in capacity and area. Their access latency mainly came from the gate delay in memory arrays. As on-chip interconnection dominates both delay and power at small feature sizes, large L2 and L3 caches have become wire-delay dominated. Hence, a Uniform Cache Architecture (UCA) with single access latency will not be favorable. The concept of Non-uniform Cache Architecture (NUCA) was proposed by C. Kim et al. [18] to handle the increasing global wire delay. The essential difference between UCA and NUCA exists in the interconnection topology, as illustrated in Figure 3. The left half shows an H-tree based UCA, which is the most common layout of the memory structures currently in use. The right half shows a mesh network

based NUCA, with one bank on each of the mesh nodes. In the UCA design, to guarantee a uniform access latency to all banks, H-tree is used for identical routing lengths from CPU to all banks, which equals to the distance from CPU to the farthest bank in the cache. This design sacrifices the speed advantages of the banks nearest to the CPU. By replacing the H-tree with a mesh network, access latency gradient can be observed across banks in NUCA, as shown in the figure, resulting a smaller average latency than the UCA. With different cache line mapping and migration policies previously developed [18], a NUCA can provide better performance and even lower power over UCA.

Various schemes were proposed to reduce the average cache access latency of a NUCA cache. Chishti et al. [56] proposed NuRapid, a design that places frequently used data in banks that are close to the CPU using indirect points from tag to data banks. In CMPs, Cho et al. [57] proposed a page coloring scheme to map the data belonging to a core to its near neighbor cache banks. There have been a few recent studies on NUCA in a 3D chip [27, 58]. However all solutions are fundamentally 2D since the data management is performed still within each layer. In this thesis, we develop a true 3D cache management for a 3D NUCA cache to improve the performance of the entire memory-stacked chip.

3.0 MOTIVATION

The previous chapter introduced the background of process variations, 3D die stacking and on-chip memories. In this chapter, we show how problems arise from an architecture that incorporates all of these three aspects. Step by step, we first show their interactions by grouping them into pairs, and finally derive our target problem by combining all of them.

3.1 3D + MEMORY

Among various 3D architectures, stacking cache or memory chips directly on top of a 2D multi-core chip [20, 23, 36, 38, 39, 40, 55] has gained its popularity due to (1) immediate boost in on-chip memory capacity without increase in die area; (2) reduced latency and energy of core-to-memory interconnect because data can be supplied vertically through TSVs which are orders of magnitude shorter than horizontal wires that connect core with memory within a die [39]; (3) increased bandwidth because the number of chip pins and the area budget of on-chip wires are no longer limitations [39]; and (4) good scalability in number of layers. When combined with different kinds of memory technologies, more benefits can be exploited from their individual characteristics.

Stacking DRAM on top of processors was mostly studied due to the high density, and hence, high capacity benefit of DRAM [36, 38, 39, 40]. Those studies assumed a "planar" 3D DRAM structure, in which each layer is a traditional 2D DRAM die with all components. In such a structure, memory requests in the core layer are first centralized to the through silicon bus (TSB) that connects vertically to memory layers, and then transferred on long-wire H-trees in the memory layers to reach a subbank, as depicted in the left half of Figure 4. A

recent work from S. Thoziyoor et al. [36] proposed the updated version of memory modeling tool CACTI and performed a comprehensive study on performance-energy tradeoffs among SRAM, logic-process DRAM (LP-DRAM, also known as embedded DRAM) and commodity DRAM (COMM-DRAM) on top of a 2D multi-core layer as stacked last level cache (LLC). Their conclusion shows that using commodity DRAM cells with low standby power (LSTP) peripheral circuitry generates the best energy-delay product. This is primarily due to the low power and high density of DRAM arrays which can compensate its low speed with large capacity (and hence low miss rates) within the same die area, when compared with SRAM arrays.

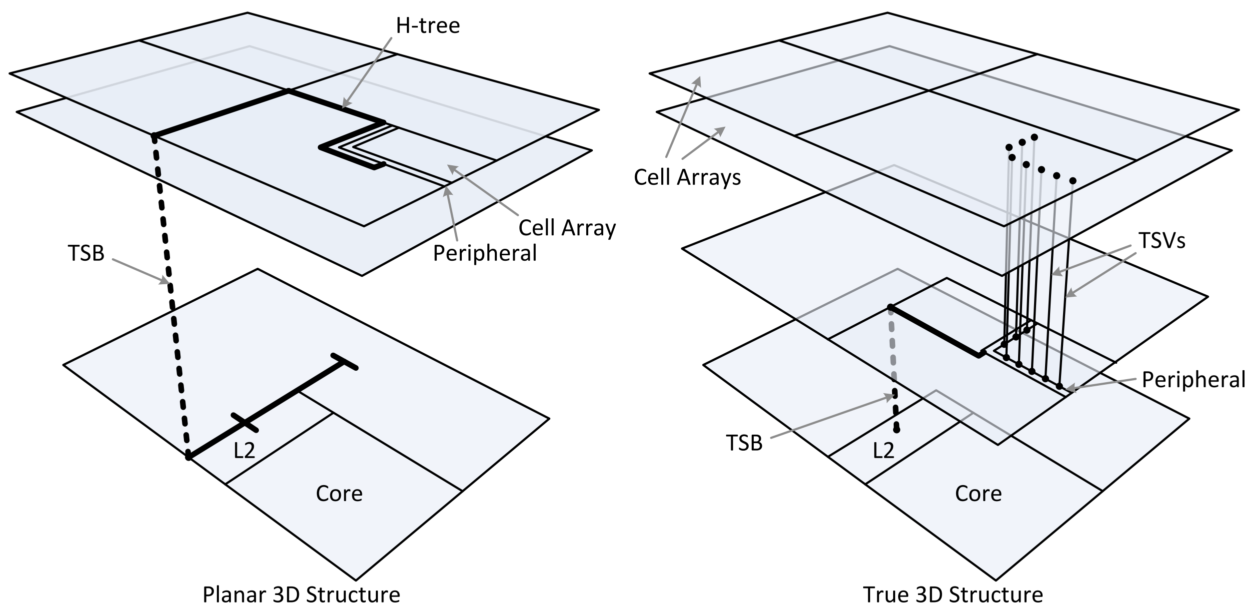

Figure 4: Planar 3D DRAM and true 3D DRAM.

While using "planar" 3D DRAM already shows benefits [36, 38, 39, 40], Tezzaron Corporation proposed an aggressive 3D DRAM structure to fully exploit the bandwidth and device count potentials of 3D stacking [21, 22]. Gabriel H. Loh showed the advantages of such "true" 3D DRAM over "planar" 3D DRAM in his stacked main memory study [23]. In conventional 2D memories, every cell array is equipped with row circuits and column circuits, forming a subbank, as detailed in Figure 5. Those row and column circuits are the peripheral logic of the memory, and they are placed on the same die as the cell arrays. The planar 3D

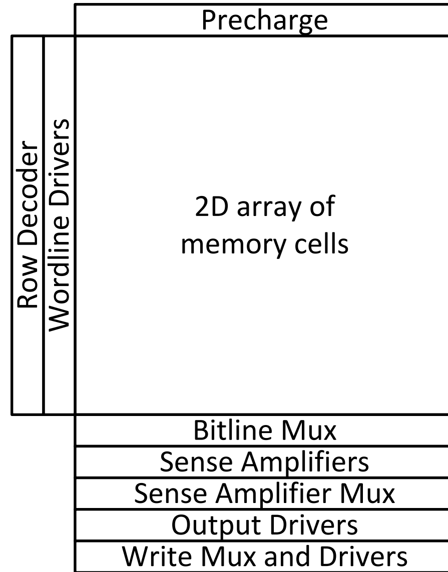

Figure 5: Subbank components and topology.

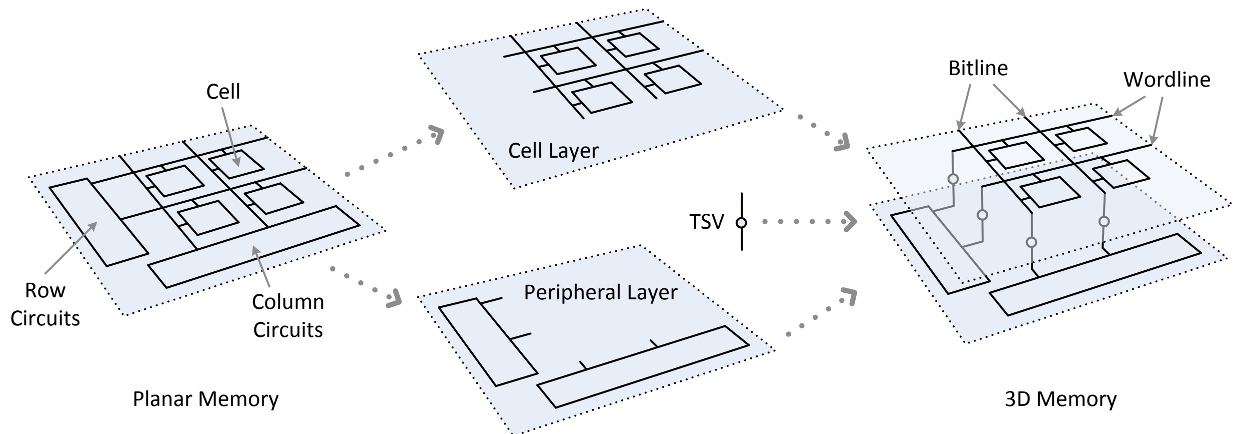

Figure 6: From planar memory structure to 3D memory structure.

memory adopts a naive stacking of such memory die on a processor die. In contrast, in a true 3D memory structure, a traditional planar subbank is distributed vertically by isolating cell array and its peripheral circuits into cell layer and peripheral layer, separately, as illustrated in Figure 6. When stacking several subbanks, all their peripheral logics are placed in a separate layer which connects with the cell layers through TSVs, forming a memory of

$N + 1$ layers where N is number of layers for cell arrays and 1 is the peripheral control logic layer. A memory request can go vertically through its immediate TSB, and then transfers onto a short bus in the peripheral layer to reach a subbank, as shown in the right half of Figure 4. Such a structure has several advantages: (1) compared to the centralized TSB solution, using TSVs at subbank granularity eliminates the bandwidth constraint, as the TSV size and latency will not likely to limit DRAM stacking for several generations [23]; (2) cell layers can be fabricated in traditional nMOS technology which can be optimized for high density. Its fabrication process is also simpler and costs less due to the absence of several main steps in CMOS process; (3) high performance CMOS technology can be employed for the peripheral circuits and interconnections in the peripheral layer to further reduce latency; (4) both clock and power routings can be greatly simplified as they are concentrated in the peripheral layer; (5) since cell layers are in a passive status (driven by the peripheral layer), the peripheral layer will be the most active layer and will contribute nearly all of the power consumption. Hence placing heat sink close to the peripheral layer is good enough for power dissipation.

The true 3D DRAM design brings forth significant performance improvement due to both bandwidth in increase and latency reduction. The bandwidth between DRAM and the processor cores is increased because the data communication channel in true 3D DRAM (TSVs) is far more wider than the off-chip 2D DRAM (package pins). The access latency of a DRAM is decreased because of three reasons: (1) between CPU and main memory, the expensive PCB traces on motherboard and DIMMs are eliminated; (2) between banks, interconnections are reduced due to much smaller die footprint; (3) high performance devices in the peripheral layer can speed up peripherals and interconnections. In contrast, it is important to note that the delay inside the cell array is not improved at all, thus the cell array latency will occupy larger and larger portion of the total memory access latency.

3.2 MEMORY + PV

While process variations attack all kinds of function units in today's large scale ICs, memories are most vulnerable to process variation, for three reasons: (1) memories are array-based, uniform structures. All elements in the array are designed and assumed to be identical, thus there are a huge number of identical critical paths in a memory; (2) to achieve high density, memory cell transistors are usually minimally sized, and a huge number of such devices make up the majority of a memory; (3) memories are occupying more and more on-chip area, and a great amount of WID variations can exist across such a large area. Besides these three reasons for general memory systems, each memory technology exhibits unique vulnerability under PV. Let us take a look at the SRAM and DRAM cell structures, as shown in Figure 7, to understand the impacts of PV on them.

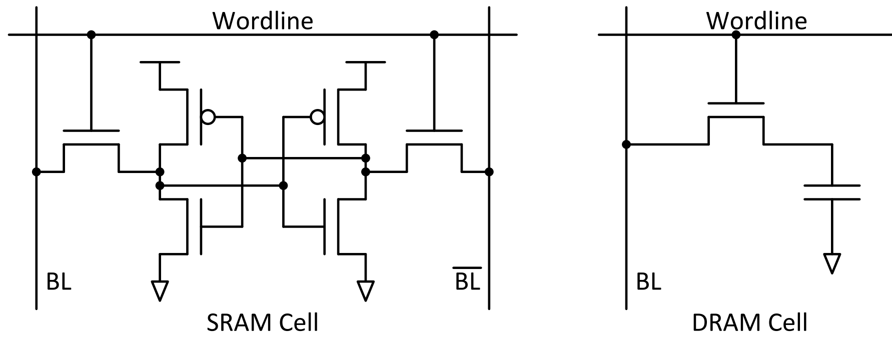

Figure 7: SRAM cell and DRAM cell.

In SRAM, cell functionalities are based on the positive feedback and inner fight of a pair of cross coupled inverters. Hence, the perfect symmetry inside the inverter pair is required. Moreover, the size ratio of cell transistors are very carefully picked to maintain the stability during read, write, and hold, while these transistors are also made as small as possible to satisfy density requirements. Due to the static structure of the cell, SRAM operations are fast, and the PV induced delay variation is small within tens of pico-seconds [11, 14]. However, the most important concern of PV on SRAM is the degraded read/write noise margin, which may lead to read/write failure, and yield reduction, in both normal [10, 11, 12, 13, 14] and ultra low voltage [15, 16] mode SRAM. In contrast, DRAM does

not have such concerns due to its simple cell structure. However, DRAM cell operations rely on the charge sharing between the cell capacitor and bitline capacitance through the access transistor, which has no static driving strength. Hence, DRAM operations are slow and sensitive to the variations on the access transistor. Sang-Hoon Lee et al. showed that there can be $18ns$ read access latency (t_{RAC}) variation in main memory DRAM under PV [44]. Since cell array delay will occupy a great portion of the total access latency in a true 3D DRAM, as analyzed in section 3.1, the PV on cell access transistors will become more important to the overall memory access latency.

3.3 3D + PV

Due to its high flexibility and high yield advantages, the die-to-die bonding has became the most developed 3D integration approach. Obviously, die-to-die variations are introduced by this bonding, which causes different performance and power consumption among the die that are stacked together. With the continuous development of 3D integration, die-to-wafer bonding will likely to be an option as well, and wafer-to-wafer bonding will be the final ultimate solution for highest fabrication throughput when high yield on each wafer can be guaranteed [19, 20]. Hence, 3D integration is expected to further introduce wafer-to-wafer variations, which lead to more severe variation condition in a 3D structure.

Additionally, even without D2D and W2W variations, 3D die integration can already result in a complex variation problem. For example, it is quite practical that several die are designed to be identical but they are then fabricated with different WID variation distributions. When these die are stacked, identical function blocks are joined in vertical dimension but they exhibit inconsistencies due to their distinct variation distributions. This situation is further discussed in section 4.2.

3.4 3D + MEMORY + PV

Combining 3D integration, memory and PV together, PV induced memory access latency variability will exist both across a die and among die, which forms a three-dimensional non-uniform cache/memory access (3D NUCA/NUMA) problem. Beside extending NUCA/NUMA into one more dimension, essential differences exist between this problem and traditional NUCA/NUMA.

Traditional NUCA/NUMA problems arise in wire delay dominated, non-H-tree based memory systems, and the non-uniformity comes from the different distances (thus different wire lengths) from memory blocks to a CPU core [18]. Also, once the chip topology is fixed, the wire lengths are fixed, thus the NUCA problem is deterministic. In contrast, the interconnection effect on a true 3D memory system is minimized, while the cell array delay is maximized in proportion, and the non-uniformity comes from the delay variability of this part. Moreover, such delay variability is caused by the random PV, which differs from layer to layer and chip to chip. Therefore, the PV induced non-uniform memory access in a stacked chip is a non-deterministic, three-dimensional NUCA/NUMA problem.

4.0 ARCHITECTURE & MODELING

To understand the impact of PV on the performance of a 3D memory stacked chip, we will first introduce how a PV model is built for a true 3D DRAM. In this chapter, we will explain the DRAM-on-processor organization, followed by its PV modeling and measurements.

4.1 3D STACKED LAST LEVEL CACHE ARCHITECTURE

Our 3D DRAM organization followed closely the "true" 3D DRAM product of Tezzaron Corporation [21, 22] and the study performed by Gabriel H. Loh [23]. It consists of one core layer, one interface layer, and four DRAM cell layers, as depicted in Figure 8.

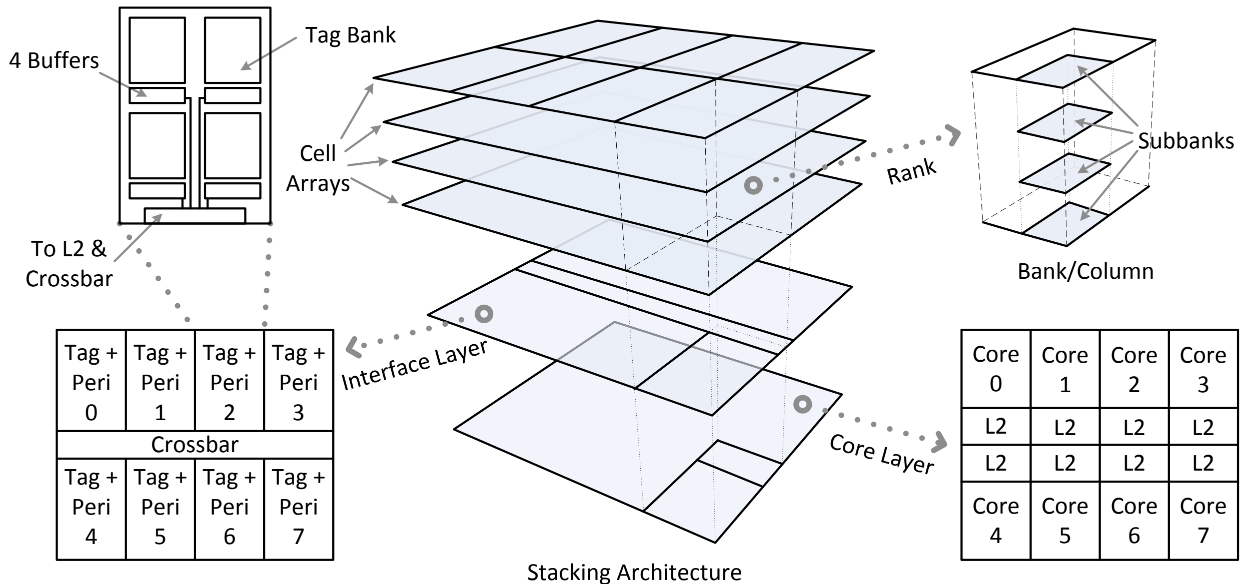

Figure 8: Architecture of a 5-layer 3D memory stacked on top of an 8-core processor.

On the core layer, we incorporated 8 multithreaded 45nm processor cores based on the area parameter of Sun UltraSPARC T2 processor [24]. We define the core area at early stage of modeling since it is a constraint to the die area of all the components, such as the memory banks, that will be vertically aligned atop. In UltraSPARC T2, each core has an area of 12mm^2 in 65nm technology. We scaled this core area to 5.8mm^2 in 45nm technology (∼31% reduction in each dimension). We also assume there is a private 256KB 8-way L2 cache per core. The area of each L2 is ∼2.2mm^2, as estimated by CACTI [36]. The layout of cores and L2's on the core layer is shown on the lower right corner of Figure 8. As we can see, each core slice (core + L2) occupies about 8mm^2, resulting a total die area of 64mm^2. For the 8mm^2 core slice area, it is difficult to fit in gigabyte of DRAM in the entire column of memory banks atop the core slice. Considering that each core is multithreaded, we choose to use the DRAM banks as the last level cache (LLC) rather than the main memory, similar to the design choice used in [36].

The four DRAM cell layers make up of the majority of the shared L3 cache as shown in Figure 8. Each cell layer is divided into 8 blocks. Each block is aligned to one core slice. Each block is further divided into 4 arrays, which are the cell arrays of subbanks consist of cells, wordlines and bitlines. Here, a subbank is the smallest unit with private peripheral circuits. The four blocks stacked directly atop a core compose a rank, which will be used most frequently by the core underneath. Inside each rank, the 4 subbank arrays stacked together are grouped as one bank / column, since their peripheral circuits will be projected to adjoining positions on the interface layer. Therefore, on top of each core, there are 16 subbanks, organized into 4 banks / columns, or 1 rank. The granularity of data array partition is limited by the area of interface layer: smaller subbanks lead to larger subbank count, which means that more area needs to be reserved for peripherals on the interface layer resulting in a more complex layout and lower area efficiency. Although higher capacity can fit into our chip area, we use a rank capacity of 96MB (6MB per subbank), for three reasons: (1) 96MB/rank leads to a 768MB shared L3 cache, which is enough for most of applications; (2) large memory capacity will result in large tag area which is constrained by the available area in the interface layer; (3) enough area needs to be reserved on each cell layer for wordline and bitline TSVs.

Because we decided to stack DRAM as LLC, we need additional transistor budget for the tags which occupy significant die area since the LLC is quite large. It is not beneficial to place these tags in the core layer, as it will expand the die area and increase the cross chip communication delay. Hence, a reasonable design choice is to place them along with the memory peripheral circuits. As high performance CMOS process is implemented in this layer, the commodity-DRAM technology is not favored. However, SRAM will be too big to fit into an $8mm^2$ area. Hence, we turned to logic-process DRAM for smaller area with good speed. The entire tag of a 96MB L3 cache rank occupies about $3.5mm^2$ area, and it is divided into four banks corresponding to the data bank partition. Also, all the 16 subbank peripherals reside in this layer in the manner that the 4 sets of peripheral logic of one column of subbanks are placed together, close to the corresponding tag bank as illustrated in Figure 8. Their row buffers are neighboring to each other for the ease of data migration. The four dedicated 128-bit buses sit in the middle, connecting all the peripherals to the crossbar interface. Note that this interface is also vertically connected to the L2 cache bank in the core layer by the through silicon bus (TSB). Estimated by CACTI, the total area of the 16 peripherals and inter-bank/subbank routings is about $2.67mm^2$. The rest of area in this layer is enough for some control circuits and a 256-bit wide crossbar, which provides connection between each pair of the 8 nodes. It takes charge of shared data access, core communications, and L2 coherence messages. Although the area budget is rich enough, smart physical layout is necessary to arrange all kinds of blocks and wire routings, and also TSV connections, in this layer.

In section 3.1, we discussed three reasons for latency reduction of a true 3D DRAM compared to a 2D DRAM. After seeing the structure of the former, we can conclude one more reason: instead of COMM-DRAM, the faster LP-DRAM is used in tags, which further decreases the total latency of a serial access cache. In addition, due to the area constraint in the interface layer, our subbanks are relatively big with longer, more capacitive and thus slower bitlines. All those reasons put more weight on cell array delay in the total L3 access latency, which further explains why the delay variation of cell arrays is important to L3 access latency.

4.2 PV MODELING

To evaluate the effect of both L_{eff} and V_{th} variations, we modeled only the spatial correlated (systematic) L_{eff} variation, for 2 reasons: (1) systematic fluctuations are the most significant performance limiter [31, 41, 42, 43], and delay variability is dominated by L_{eff} variation across normal and low supply voltage region [4]; (2) systematic variation of V_{th} is completely determined by systematic variation of L_{eff} [2]. S. Herbert et al. also assumed perfect correlation between these two variations in their study [26] according to the analytic model in [4]. Though the relation between L_{eff} and V_{th} variations is characterized analytically in these works, it is inherent in HSPICE.

We used VARIUS [2], a PV modeling infrastructure based on the statistic tool R [28] and its geoR package [29], to model both WID and D2D variations. This model adopts a multivariate Normal (Gaussian) distribution with the well-studied spherical structure [33] for spatial correlation. For a variable that follows the Normal distribution with parameter, mean μ, variance σ, and density ϕ as main inputs, VARIUS outputs its variation map. The intuition of μ and σ of WID and D2D variations can be seen from Figure 2. The density ϕ is a parameter that determines the range of WID spatial correlation. It is expressed as the fraction of a chip's length [2]. As the spatial correlation of two points decreases as their distance grows, ϕ is the distance at which the correlation drops to zero. For example, $\phi = 0.5$ means that the correlation range equals to half of the chip length, hence one device is correlated to all the devices within that radius.

The values of μ, σ and ϕ are determined as follows. Since DRAM cells are minimized for device density, we let μ be the minimum gate length for a given technology size. Previous studies [2, 4, 26, 30, 31, 41] used σ/μ ratio to determine σ. For example, Yu Cao et al. used $\sigma/\mu = 10\%$ at $90nm$ [4], S. Herbert et al. used $\sigma/\mu = 3.2\%$ at $22nm$ [26], S. R. Sarangi et al. used $\sigma/\mu = 4.5\%$ at $32nm$ [2], R. Teodorescu et al. used $\sigma/\mu = 6\%$ at $32nm$ [30], X. Liang et al. used $\sigma/\mu = 7\%$ at $32nm$ [31], K. Bowman et al. predicted $\sigma/\mu = 7\%$ at $50nm$ [41], and we chose our $\sigma/\mu = 7\%$ for WID gate length variation, similar to that in [31] and [41]. For D2D variations, the difference between the μ of each die (e.g., $\mu0$ and $\mu1$ in Figure 2) also follows the Normal distribution. This distribution has the same mean as the

nominal minimum L_{eff} we use, but a different variance, σ'. There is little public-domain information on likely values of σ'. X. Liang et al. used $\sigma'/\mu = 5\%$ at $32nm$ [31], K. Bowman et al. predicted $\sigma'/\mu = 7\%$ at $50nm$ [41], and we conservatively take a 2% for σ'/μ in the D2D L_{eff} variation model. The value of ϕ is related to the die size. A large die tends to have relatively small ϕ. Friedberg et al. experimentally measured this correlation range of gate length variation to be close to half of the chip length ($\phi = 0.5$) with a relatively large chip size of $28 \times 22\ mm^2$ [5], Radu Teodorescu et al. used $\phi = 0.5$ for a die size of $340mm^2$ [30]. Because our chip size is relatively small, we picked $\phi = 1$.

We repeatedly ran VARIUS to generate a large number of die variation maps. Each map corresponds to the L_{eff} distribution of one die. Each map has one million (1000×1000) sample points. Both WID and D2D samples are in Normal (Gaussian) distribution. We then randomly picked 40 die maps, and randomly grouped 4 maps together, representing the 4 DRAM layers, to form 10 stacked chips. Figure 9 shows a sample L_{eff} distribution in the 4 DRAM cell layers of a chip. The 16-subbanks-per-die floor plan of each layer is superimposed on the variation maps. Four neighboring subbanks of the same location in all 4 layers form a memory rank atop a core, such as rank A and B in solid line shown in the figure. The values of the sample points are converted to different colors on the map. Dark red color implies small value, or short L_{eff} , and light yellow color indicates the opposite. It's quite clear from the maps that gate length variations spread across each layer, and among different layers. Most importantly, when WID and D2D variations are considered together, not only do the subbanks in every layer present L_{eff} variations, but also the subbanks in one vertical rank have a range of L_{eff}. For example, all layers of rank A have relatively short L_{eff}. Therefore, rank A has relatively high average speed. In contrast, rank B has long L_{eff} in Layer 1 and 3. So those two layers are fairly slow, while Layer 2 and 4 are faster. Such a difference gives us the motivation to utilize faster subbanks in a rank / column more frequently for better performance.

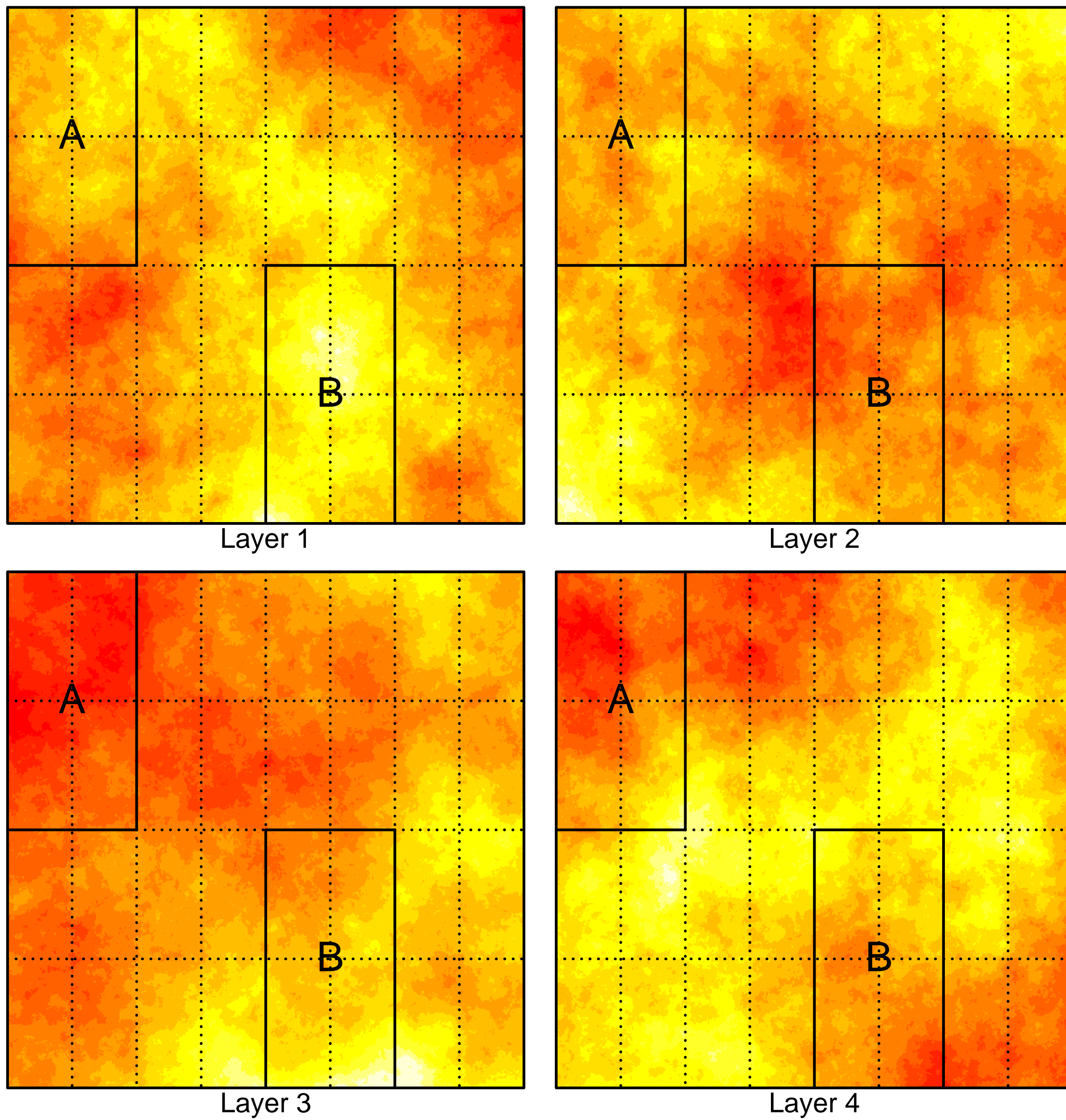

Figure 9: Sample L_{eff} distribution maps for 4 DRAM cell layers of one chip.

4.3 LATENCY AND LEAKAGE MODELING

In statistical circuit design, Monte Carlo simulation is usually used. It takes two main steps: (1) randomly (with certain distribution) pick parameter samples within a given range; (2) run simulation on each of these samples separately. In our latency and leakage evaluations, Monte Carlo simulation was not used, since we need one more step between (1) and (2). Our PV modeling procedure, including generating, picking, and grouping variation maps, is the step to pick random samples, thus it is equivalent to step (1) of Monte Carlo. However, we will not run simulations on all the samples, as Monte Carlo do. For each die, we pick 64 representative samples from the 1 million samples to represent the 32 subbanks. As we will explain later, the biggest and smallest L_{eff}'s in subbanks are picked for latency and leakage simulations separately. The following simulations are run on these chosen samples, which equals to step (2) of Monte Carlo.

For evaluation, we built and simulated in HSPICE the critical path of a DRAM array with DRAM cell, bitline and sense amplifier. For the access transistor in a DRAM cell, we used a long-channel, high V_{th} PTM [34, 35] nMOS model with thick gate oxide, which is mainly based on some parameters in [36] and the CACTI source code, as summarized in Table 1. The cell capacitor was chosen to be $30fF$ [36, 37], and wordline voltage is

Table 1: Device parameters from CACTI source code.

Cell VDD	1.1 V
Physical Gate Length	0.045 μm
Electrical Gate Length	0.0298 μm
Vth	1 V
Gate Width	0.045 μm
Ion	20 μA
Ioff	1 fA
Cell Capacitor	30 fF
Wordline Boosted Voltage	2.7 V

boosted to $2.7V$ [36]. The π RC model was used as the long, very capacitive bitline, and we considered a $40mV$ bitline voltage swing to be a reasonable sensing margin. In each HSPICE run, the cell transistor was modified with a gate length that represents one subbank, and the resulting latency or leakage were then used for that subbank in the subsequent architectural simulations. All HSPICE simulations were performed with a temperature of 90 °C.

4.3.1 Latency Modeling

We studied the latency variability at subbank granularity, and the latency of one subbank is determined by its slowest cell, which corresponds to the biggest value (longest L_{eff}) of all the sample points within the area of this subbank. We then feed those L_{eff} numbers into HSPICE simulation.

All the sample points in a subbank fall into a value range. This range can be wider or narrower, in accordance with the flatter or sharper variation gradient inside this subbank. This range also should be continuous, due to spatial correlation. Figure 10 gives 3 sample subbanks. Subbank 1 represents the fast subbanks such that all of their samples are smaller than the nominal value μ. Subbank 2 is a normal bank with samples around μ. And subbank 3 is an example of slow subbanks. Following the trend of the L_{eff} distribution, shown as the black line in the figure, the number of subbanks in type 2 should be the biggest, and the

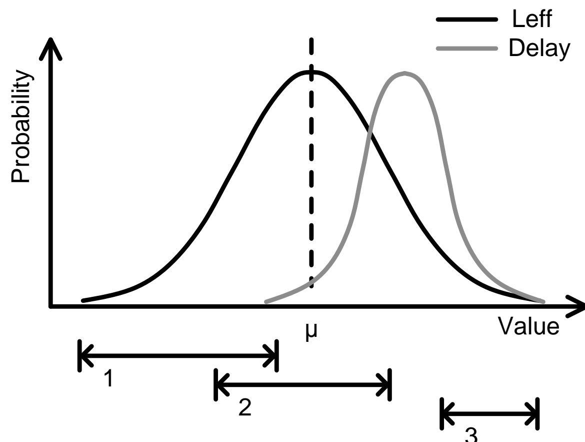

Figure 10: Predicting access latency distribution from gate length distribution.

number of type 1 and 3 subbanks should be smaller. Because the delay of a subbank needs to accommodate its slowest cell, which corresponds to the right-most point in its range, a Normal distribution of subbank access latency can be expected: little subbanks can run faster than nominal case, most subbanks fall between nominal and slowest conditions, as shown by the grey line in Figure 10.

As an nMOS, the DRAM cell transistor is better at passing a logic 0 than 1, this in turn means that reading a logic 1 from a cell will be slower than reading a 0. All of our delay simulations were performed with the configuration of reading a logic 1, in which cell capacitor voltage is set to VDD and bitline voltage is set to half VDD as initial conditions. We ran simulations on 10 modeled chips, the peripheral latency estimated by CACTI was then added to generate the access latency of the L3 cache data array. Figure 11 shows the histogram of the subbank latency as the result of PV, collected from all four memory layers of all 10 chips we modeled. The nominal (no PV) subbank latency is 15 cycles, indicated by the ideal line. This result agrees perfectly with our prediction in Figure 10. The results show that the subbank speed distribution has a larger mean than 15 cycles, which create challenges to PV-tolerant designs. A proper goal is to approach the performance of the ideal

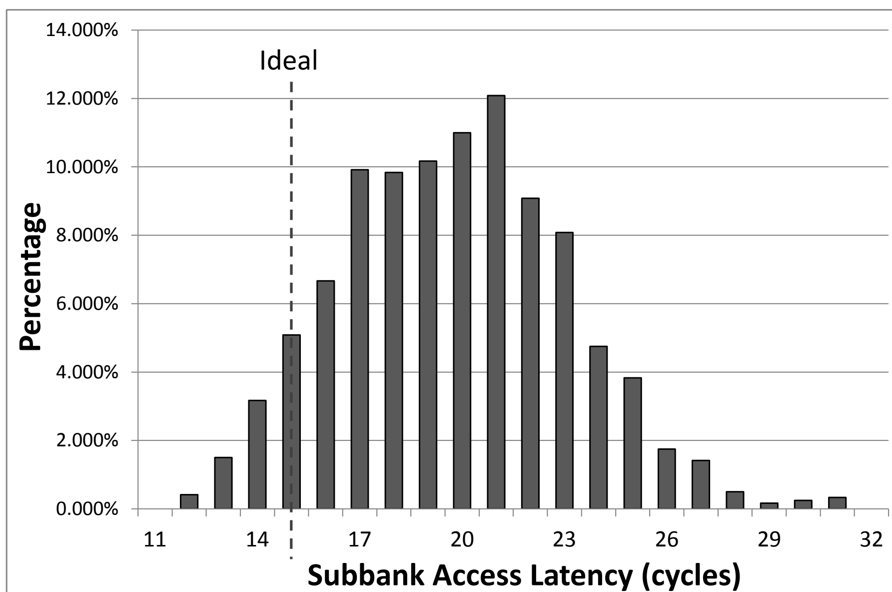

Figure 11: DRAM data array subbank latency distribution as the result of PV.

case by using fast subbanks more frequently, and probably win a little if there exist subbanks that are faster than ideal case.

4.3.2 Leakage Modeling

L_{eff} variations and the induced V_{th} variations also have impact on the leakage of the circuits. When a DRAM subbank is particularly leaky, it should be refreshed more frequently, which affects both performance and energy. To quantify the leakage variations, we also simulated the cell leakage with different L_{eff} values. Due to the inverse proportional relation between gate length and leakage, we found the shortest gate length within each subbank as the leakiest device for HSPICE simulation. Then, we assumed an inverse linear relationship between leakage and refresh period, and scale the refresh period from a normal value of $64ms$ [36]. For example, if one subbank is two times leakier than normal, its refresh period will be $64 / 2 = 32ms$. Note that this is pessimistic because the $64ms$ refresh period is obtained at worst case to satisfy all the cells, while we used it as our average case. In the subsequent architectural simulation, we used subbank level refresh control in a burst refresh mode, meaning that a refresh performs from first line to the last line continuously. A subbank is marked busy and cannot serve any requests during this period of time.

Again, because the nMOS cell transistor is better at passing a logic 0 than 1, a cell will be much leakier when storing a logic 0 than 1. All of our leakage simulations were performed under the condition that the cell capacitor voltage starts at GND and bitline voltage starts at half VDD. An inverse exponential relation between gate length and cell leakage was observed, as shown in Figure 12.

4.3.3 Hardware-Awareness of PV

Since PV magnitude can be detected only at the post-fabrication stage, it is necessary for the hardware to detect such variation and store it on-chip for PV-aware configurations. Tezzaron Corporation incorporates a continuous on-chip testing technology which tests all tiny circuits of the entire memory chip at each power-up [32]. Similar technique such as a built-in self-test circuit that can test and record the retention time of each individual cache

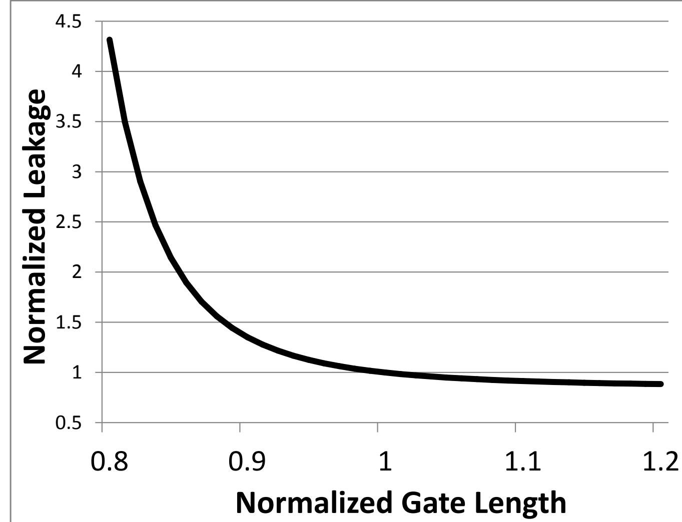

Figure 12: Leakage dependency on gate length.

line after fabrication was also used in previous study [31]. We therefore assumed that the stacked chip can be fully tested after fabrication and the tested results can be stored on-chip for PV-tolerant configurations such as one we propose in this work.

We have discussed how we modeled PV in the DRAM stack of the 3D chip we study. We remark that similar PV also exists in the core layer and the interface layer. Our goal in this work is to reduce the average access time in the LLC. Therefore, it is additive with the optimizations at the core level. The peripheral layer uses high performance devices, and thus accounts for a relatively small part in the entire LLC access time. Therefore, we focus on the latency variations, and the consequent energy results of the DRAM cell layers in this work, and leave the other factors to future work.

4.4 ENERGY / POWER MODELING

Energy / power are modeled in detail for the entire 3D chip. We estimated the core power based on released Sun UltraSPARC T2 specifications, and extracted energy numbers of the memory hierarchy from L2 cache to main memory, as well as a crossbar, from CACTI.

4.4.1 CPU Cores

We estimated the power of our CPU cores by scaling from the Sun UltraSPARC T2 processor core [24]. T2 has a total power of 65W including 8 cores (with L1 caches), 4MB shared L2 cache and other blocks. Since our L2 cache has different configurations than T2, we will model it using CACTI, we first modeled the L2 cache of T2 and subtracted its power from the total T2 power. Also, as a SoC, T2 combined various kinds of functional units on chip, such as memory control units, Ethernet controller and PCI express interface, and we estimated their total power to be 10W. Then, the power of pure T2 cores can be obtained, as follows:

total power of T2	= 65 W	(from T2 datasheet [24])
power of L2 of T2	= 10.21 W	(from CACTI)
power of other circuit except core and L2	= 10 W	(estimate)
power of cores of T2	= 65 - 10.21 - 10 = 44.79 W	

We further divided the core power into dynamic and static power by assuming that 40% is static power, similar to [36].

dynamic power of cores of T2	$= 44.79 \times 60\% = 26.87\ W$
static power of cores of T2	$= 44.79 \times 40\% = 17.92\ W$

The dynamic power scaling rule is shown in following equation, where α is activity factor, C is capacitance, F is frequency, and V is supply voltage. The specifications of T2 and our design are summarized in Table 2.

$$Power = \frac{1}{2}\ \alpha\ C\ F\ V^2$$

The activity factor depends on the programs run on the cores, and is assumed to be identical between the two designs, thus has no effect on power scaling. The capacitance has linear correlation with technology node. We obtained the dynamic power by applying the

Table 2: fundamental parameters of Sun UltraSPARC T2 [24] and our design.

	Sun UltraSPARC T2	Our Design
Number of Cores	8	8
Supply Voltage	1.1 V	1.1 V
Frequency	0.9 GHz	3 GHz
Technology Node	65 nm	45 nm
L2 Cache	4 MB shared	256 KB private, 2 MB total

power scaling rule:

$$\text{dynamic power of our cores} \quad = (45/65) \times (3/0.9) \times (1.1/1.1)^2 \times 26.87W = 62\ W$$

From 65nm to 45nm, we assumed a 1.5X leakage increase, hence

$$\text{static power of our cores} \quad = 1.5 \times 17.92W = 26.88\ W$$

And finally

$$\text{total power of our cores} \quad = 62 + 26.88 = 88.88\ W$$

4.4.2 L2 Cache, Main Memory, and Crossbar

We directly used the CACTI modeling results for L2 cache, main memory and crossbar, as shown in Table 3, 4 and 5.

4.4.3 DRAM LLC (L3 Cache)

A large memory system is a combination of two types of structures: memory circuits and routing interconnects. Memory circuits consist of a group of identical subbanks, and routing interconnects include all kinds of wires that carry data between a subbank and the controller of this entire memory system. For the energy / power values of a subbank, we use the CACTI

Table 3: Energy / power of a 256KB L2 cache.

data dynamic read	0.104 nJ
data leakage	69.96 mW
tag dynamic read	0.0065 nJ
tag leakage	1.512 mW
total dynamic write	0.155 nJ

Table 4: Energy / power of a 16GB main memory.

dynamic read	5.99 nJ
dynamic write	5.98 nJ
leakage	681.12 mW

Table 5: Delay and energy / power of an 8 port, 256-bit wide crossbar.

delay	495 ps
dynamic energy	0.146 nJ
leakage power	0.044 mW

result directly since the subbank structure of a true 3D DRAM and a "planar" 3D DRAM are the same. The detailed breakdown is shown in Table 6. However, because of the TSVs and TSBs, the routing interconnects are dramatically different now, in both length and topology. We now introduce how this part is modeled.

The left part of Figure 13 shows an example of the traditional planar DRAM layout, as utilized by CACTI. The interconnections in such a layout can be divided into route-to-bank wires, shown as the big H-tree in bold lines, and intra-bank wires, shown as the small H-tree in fine lines. As analyzed earlier, when converting the planar DRAM into a true 3D design, one rank of memory now becomes a stack of four banks, and one bank becomes a stack of four subbanks. As a result, the entire rank, including all banks, is only one quarter of the

Table 6: Subbank energy breakdown.

Component	Energy (nJ)
Row Decoder	0.004
Bitlines	0.027
Sense Amplifiers	0.034
Output Drivers	0.105
TOTAL	0.170

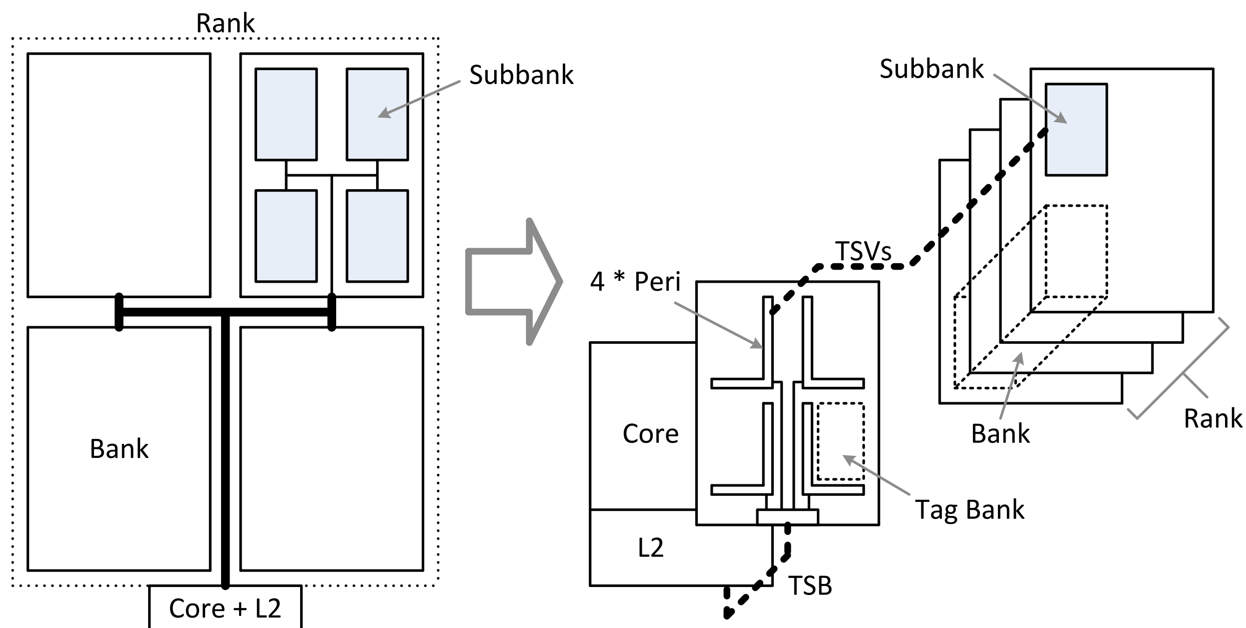

Figure 13: mapping interconnections from planar DRAM to 3D DRAM.

planar design in footprint. As we can see, the original inter-bank H-tree now becomes a small H-tree that used to interconnect four local subbanks. To provide good throughput, we replace this small H-tree with dedicated subbank-to-TSB buses, as shown on the right part of Figure 13. We estimated this bus to be 2/3 of the small H-tree. The original inter-subbank small H-tree has become only the very short wires between the four peripherals of the same column. In this work, we ignore the delay and energy of TSVs, because they are orders

of magnitude shorter than horizontal wires. The energy / power results are summarized in Table 7. We pessimistically subtracted only the route-to-bank part from the CACTI result to obtain our data.

Table 7: Energy / power of a 96MB L3 cache (one rank) for read / write evaluation.

	CACTI	Our Design
data dynamic read	0.568 nJ	0.39 nJ
data leakage	1132 mW	1087 mW
tag dynamic read	0.07 nJ	0.052 nJ
tag leakage	11.32 mW	8.74 mW
total dynamic write	0.66 nJ	0.464 nJ

We also modeled energy / power for data migration polices. The three basic migration steps are (1) read one subbank, (2) send data from one subbank to another subbank, and (3) write one subbank. The energy breakdown for essential migration components are listed in Table 8. 64B wide wires are added between banks / columns for inter-column migration.

Table 8: Energy / power of a 96MB L3 cache (one rank) for migration evaluation.

Component	Energy (nJ)
Data Subbank Read	0.065
Data Subbank Write	0.087
Inter-column Wire (64B, one-way)	0.325
Tag Read	0.052
Row Buffer Swap	0.230

5.0 NON-UNIFORM CACHE MANAGEMENT

In this section, we first present the performance analysis for the 3D-stacked 5-layer DRAM-on-processor. Next, we introduce our proposed cache management schemes pertaining to the 3D architecture. Before discussing any quantitative results, we first describe our evaluation environment and necessary settings.

5.1 EVALUATION SETTINGS

We used Virtutech Simics [59] to model the performance of our 8-core 3D DRAM stacked processor. Major parameters are detailed in Table 9. We extended the g-share module in Simics with PV features such as different subbank access times. In addition, as we will explain later, we modeled the contention in cache subbank accesses as well as the crossbar interconnect for our PV-tolerant designs. Both L1 and L2 are private, and LLC is shared. We used snoopy MESI protocol in L2 to maintain the cache coherence. For the private L2, a miss would trigger a snoop request to all L2s, and then waits for the response back. Such a coherence transaction can take a long time. Fortunately, our LLC has large capacity, and it is highly likely that the local L2 miss can be satisfied in the LLC. Therefore, we parallelize these two operations for performance benefit.

As we can see from Table 9, the nominal LLC subbank latency is 15 cycles. The variation histogram of subbank latency is shown in Figure 11 and explained in subsection 4.3.1. Another subtlety is the latency model for the interface layer. We charged 3 cycles for source-to-destination traffic through the crossbar in an uncontended network. Contention latencies will be further added during the simulation. For tag subbank + row buffers that are imme-

Table 9: Baseline chip configurations without PV.

CPU	8 cores, 3GHz, 2-issue, in-order
Private L1 I/D	32KB/core, 8-way, 64B line, 2-cycle hit time
Private L2	256KB/core, 8-way, 64B line, 2-cycle tag lookup, 3-cycle data hit time
Shared LLC	distributed 3D stacking, 4 data cell layers, 1 interface (peripheral) layer
Organization	tag array: 4 subbanks per core data array: 4 subbanks per core per layer 6MB tag array on the interface layer, 32-bit per tag entry
LLC Configurations	768MB, 48-way, 3-way per subbank 512B line, 5-cycle tag lookup, 15-cycle data hit time 2-cycle delay for tag subbank not immediate to the crossbar
Interconnect	8-port crossbar, 16B wide, bidirectional, 3-cycle end-to-end uncontended latency
Main Memory	16GB, 2 channels, 160-cycle for the critical block

diately next to the crossbar interface (refer to Figure 8), there is only 1 cycle delay on the wire. For others, there are 2 cycle delays. These values are important source of overhead in our proposed migration schemes.

We selected 5 multi-threaded workloads from PARSEC [60] and 10 workloads from SPECCPU-2006. Those workloads are memory intensive, and therefore, more sensitive to process variations in L3. All workloads were compiled with gcc 4.1.0. For PARSEC workloads, we used `simlarg` as the input and simulated the code region of interest from the beginning to the end. For SPECCPU-2006 workloads, we skipped the initialization phase, and simulated 2 billion instructions afterwards. The details of the workloads are listed in Table 10. For multi-threaded workloads, we applied *page coloring* [61] technique to enhance data locality for performance benefit. We ran the workloads twice, the first time distinguish-

ing private and shared pages and the second time pinning private pages to their local LLC banks. For single-threaded workloads, all data are located in the memory rank atop the core.

Table 10: Simulated workloads. “memory” stands for memory footprint. “L2 MPKI” stands for L2 misses per kilo-instructions. “instructions” stands for the number of instructions simulated. All SPEC2006 workloads are run for 2 billion instructions.

SPEC2006	astar	bwaves	gcc	GemsFDTD	lbm	leslie3d	libquantum	mcf	soplex	sphinx3
memory	46M	820M	45M	838M	407M	81M	36M	29M	92M	22M
L2 MPKI	5.4	10.2	6.8	18.3	23.2	16.8	33.8	69.3	24.6	12.7

PARSEC	canneal	facesim	ferret	streamcluster	x264
memory	162M	210M	71M	16M	90M
L2 MPKI	21.0	4.8	3.1	8.7	2.2
instructions	1700M	28500M	38000M	35000M	24300M

5.2 STATIC PV-TOLERANT LLC

We used two baselines as our basis for comparison. The first one is an ideal chip that has no PV, i.e., a chip with uniform LLC. This baseline gives us the performance that we would like to achieve in presence of PV. The second baseline considers the impact of PV, and uses the slowest subbank's latency as the nominal LLC subbank access latency. This baseline obviously produces the worst performance. Figure 14 compares the performance of the two baselines, normalized to the ideal one. The error lines show the performance range of the ten chips, and the bars show their average. In the “l3-worst” results, we did not use the real slowest subbank in the entire LLC as it is typically too slow to make the comparison fair. In reality, cache or memories are typically built with redundancy, and faulty memory lines can be turned off and remapped to the redundant memory lines. We also assume there is such redundancy for PV purposes so that the slowest cache lines are disabled. Hence, in our measurements, we used the third slowest subbank's speed to produce the results. We can see that the pessimistic baseline is 6∼29%, with an average of 16% worse than the ideal baseline. The performance degradation is seen more for the single-threaded workloads because their

L2 MPKIs are higher which indicates that their LLC demands are higher.

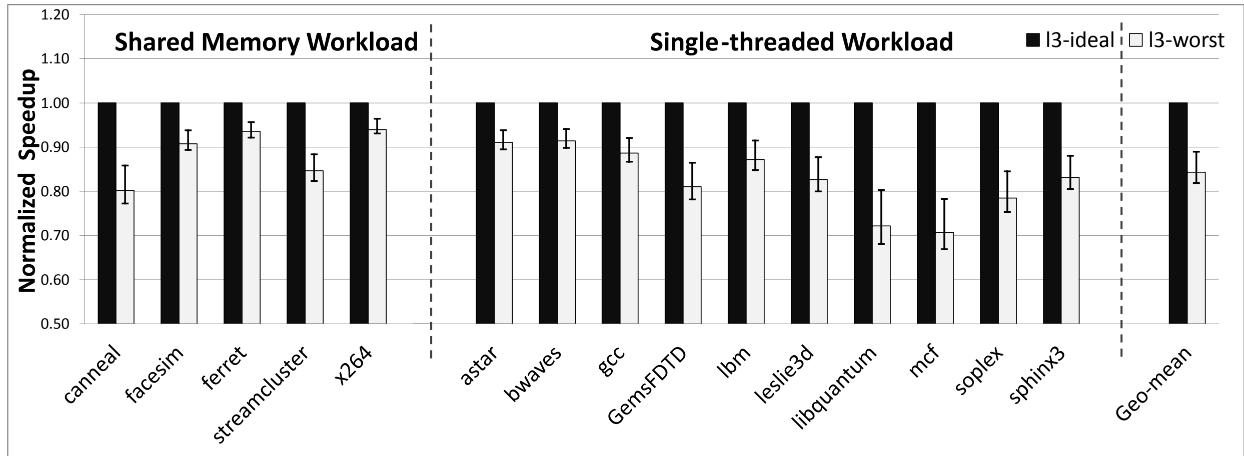

Figure 14: Performance of the pessimistic baseline normalized to the ideal baseline.

With PV-tolerant LLC designs, we expect that the performance will win over the pessimistic baseline, but inferior to the ideal baseline. Hence, our objective is to approach the performance of the ideal baseline. The first PV-tolerant design is the straight forward static NUCA design, much like that in a 2D CMP. That is, every subbank has its own speed, irrespective of the slowest subbank speed. Therefore, some subbanks will indeed be faster than the nominal speed. However, one design complication is that the cache requests may

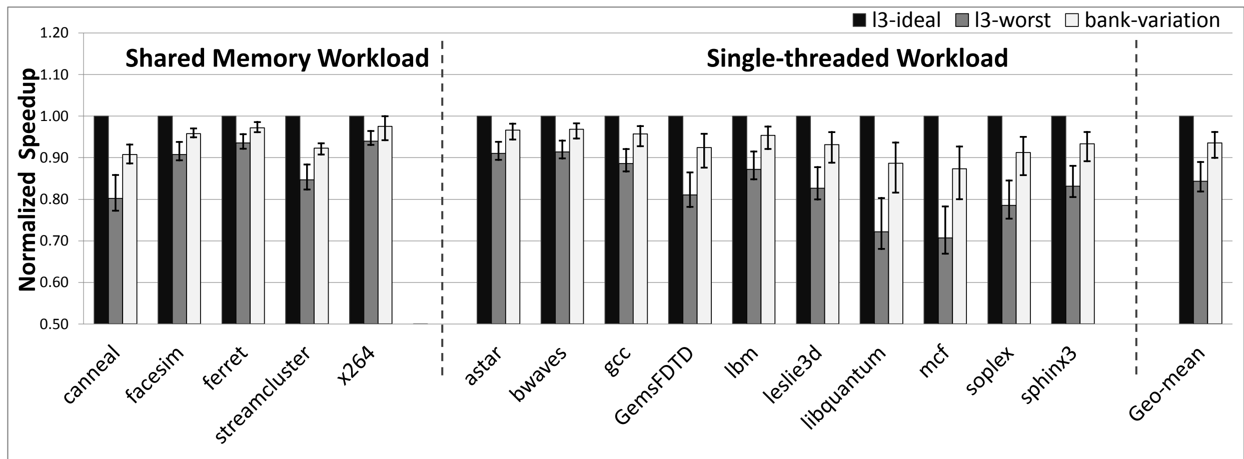

Figure 15: Performance of the static PV-tolerant LLC, normalized to the ideal baseline.

return out-of-order since each subbank has different access latency. This may create contention in the interconnection channel shared among multiple subbanks. In our design, this is the dedicated bus that connects the 4 row buffers of each column of subbanks to the TSV/crossbar port in the interface layer (refer to Figure 8). The contention comes from the vertically aligned 4 memory subbanks. We added a local arbiter to arbitrate their requests for this bus. Figure 15 shows the performances of this scheme compared to the two baselines. As we can see, all workloads immediately benefit from a PV-tolerant design. The average slowdown compared to the ideal baseline is now 6%.

5.3 DYNAMIC PV-TOLERANT LLC

In static PV-tolerant design, data are statically mapped to a subbank. The data in a slow subbank will always be accessed with long latency. We can improve this by applying data migration, similar to the DNUCA design in a 2D CMP [18]. The idea is to migrate data from slow subbanks to fast subbanks. For example, if an LLC access hits in a slow subbank, after the data is sent back to the core, it is moved immediately to a fast subbank such that next time the data can be fetched with lower latency. The question is where to migrate the data, as there might be many subbanks that are faster.

5.3.1 Bank Latency Based Migration Policy

An intuitive migration scheme is to always move the data to the fastest subbank. That is, all 16 subbanks in a memory rank are sorted by their access latencies in ascending order from Subbank-0 to Subbank-15. Once a cache line is accessed in a slow subbank, it is migrated to the fastest subbank (Subbank-0). The victim cache line from Subbank-i are evicted to Subbank-(i+1), $0 \leq i \leq 15$, as illustrated in Figure 16(a). However, this scheme in fact degrades the performance on average, as can be seen from the data series labeled "bank-migration" in Figure 17. The results show that it is even 7% worse than the static PV-tolerant scheme. This is because a lot of contentions were created when all rest 15

subbanks move their data into Subbank-0. All migration activities compete for the free time of Subbank-0, generating long wait time due to this contention. What makes things worse is that Subbank-0 is now also highly demanded as all data are supposed to be located there. Hence, every LLC access is first directed to Subbank-0 then other subbanks, generating even more traffic to Subbank-0.

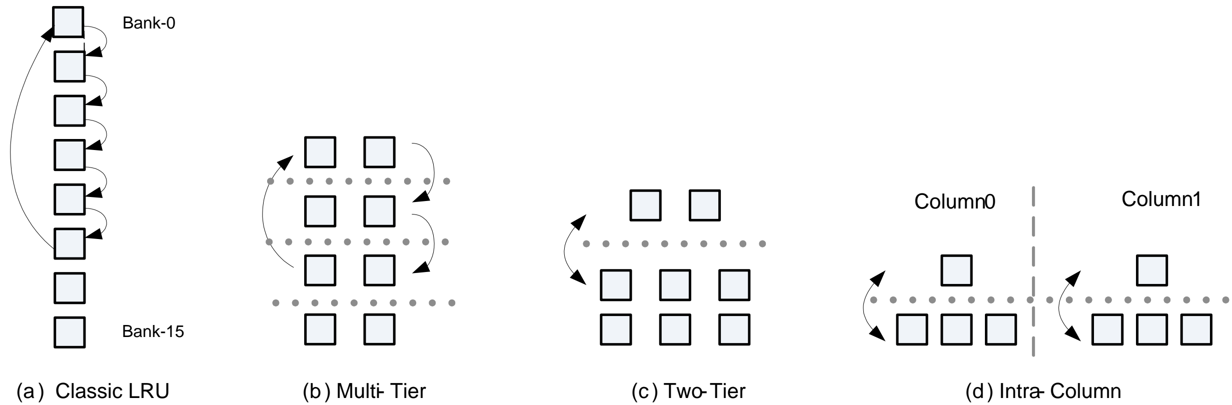

Figure 16: Illustrations of different dynamic data migration policies.

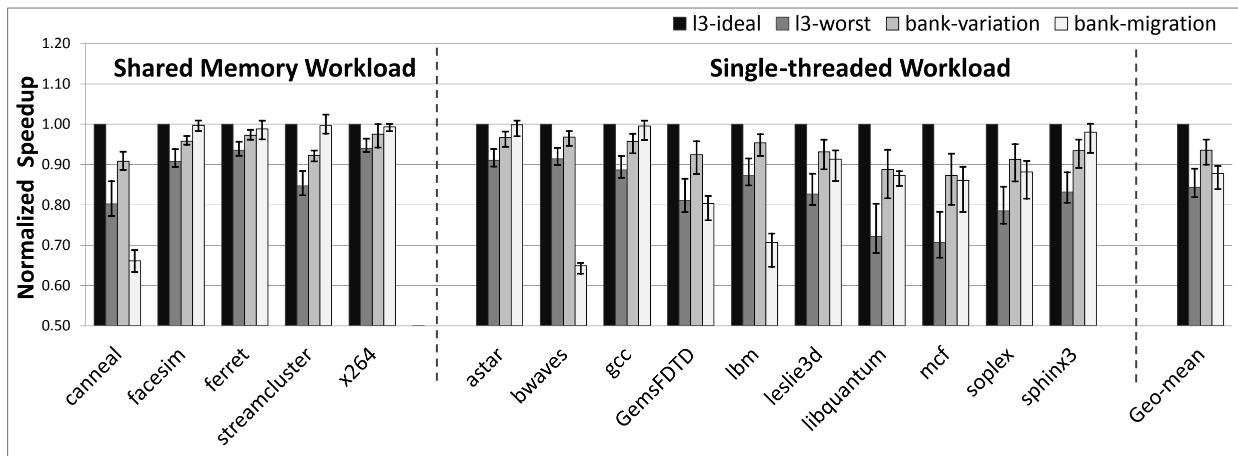

Figure 17: Performance of the latency-based migration policy, normalized to the ideal baseline.

5.3.2 Tiered Migration Policy

If we take another look at Figure 17, we find that some workloads such as `facesim`, `ferret`, `gcc`, `sphinx3` etc. did see improvements with the above migration policy. Some of them are already as good as the ideal baseline. Those are the workloads that have less than ∼10 L2 MPKI, as indicated in Table 10, meaning that their demand on LLC is relatively low. This observation motivates us to revise the migration scheme to offload the requests from Subbank-0 to maybe the second, third, etc. fastest subbanks. More formally, we divide the subbanks in one rank of memory into several tiers according to their speed, and each tier has several subbanks of similar speed. Migration is done such that data is moved from a slow tier to the fastest tier, and evictions go in the opposite direction. Within each tier, cache lines are installed in a Round-Robin manner. Therefore, the pressure originally on Subbank-0 is now evenly distributed into several subbanks, greatly reducing the likelihood of subbank contention. This is illustrated in Figure 16(b).

Next, we study how to divide the subbanks into tiers. We evaluate how many tiers are necessary, and what size of each tier should be. We first start from a simple 2-tier design: a fast tier and a slow tier. We vary the size of the fast tier from 1 to 16, and leave the remaining subbanks to the slow tier. For example, the latency based migration is simply a 2-tier scheme where the fast tier has only Subbank-0, and all the rest 15 subbanks are in the slow tier. We measured the performance of all partitions and normalized them to this configuration. Results are shown in Figure 18. As we can see, there is a tradeoff between the tier size and performance, as the initial size increase in the fast tier does help to improve the performance for most benchmarks due to load distribution. However, when the fast tier size is larger than the slow tier, performance starts to decrease. This is because when the fast tier is bigger, the average LLC access time is also larger which in turn harm the performance. There are also a few workloads, such as `sphinx3` that have nearly monotone decreasing performance. Those workloads generally have small LLC demand and less contention in Subbank-0. Therefore, the smaller the fast tier, the lower the LLC average access time, and the higher the performance. Figure 19 summarizes Figure 18 using Geometric means of all workloads. This curve clearly shows the best choices for the fast tier: 3∼6 fastest

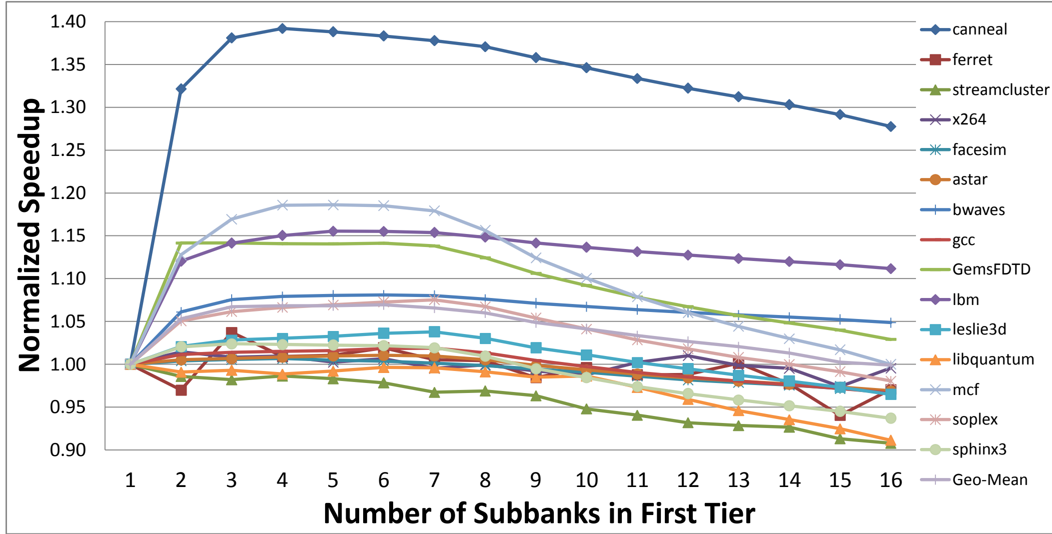

Figure 18: Performance variations in a 2-tier migration scheme.

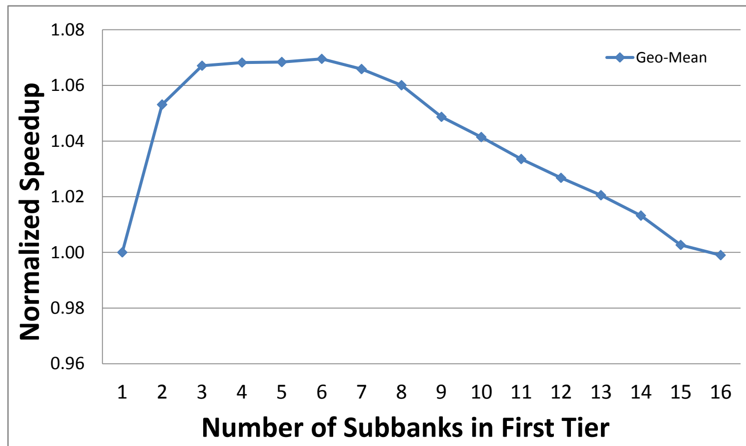

Figure 19: Performance summary for a 2-tier migration scheme.

subbanks give the highest performance. Moreover, we also tested the potential of having a dynamic varying fast tier assuming that the best tier partition can be found dynamically. The Geometric mean of this dynamic optimal value is well below 1% away from the static partition of 3∼6 subbanks. This implies that it is not necessary to perform dynamic searching for the best partition at runtime since a static partition is sufficiently good.

Finally, we keep increasing the number of tiers to see if further performance improvements can be obtained. Figure 20 compares the performance for 2-tier, 3-tier and 4-tier migration

schemes. As we can see, there is no clear performance benefit in increasing the tier count. A two-tier enabled migration scheme is adequate and the migration operation is also simple. A cache line swap between the two tiers is enough to gain performance benefits. Our two-tier partition has 4 subbanks in the fast tier, which amount to $96\times4 = 384$MB memory space. This is fairly large to hold most of our workload's working sets. For workloads that are even larger, it may be advantageous to have more tiers. In all the following studies, we use a 2-tier (4-12) partitioned configuration.

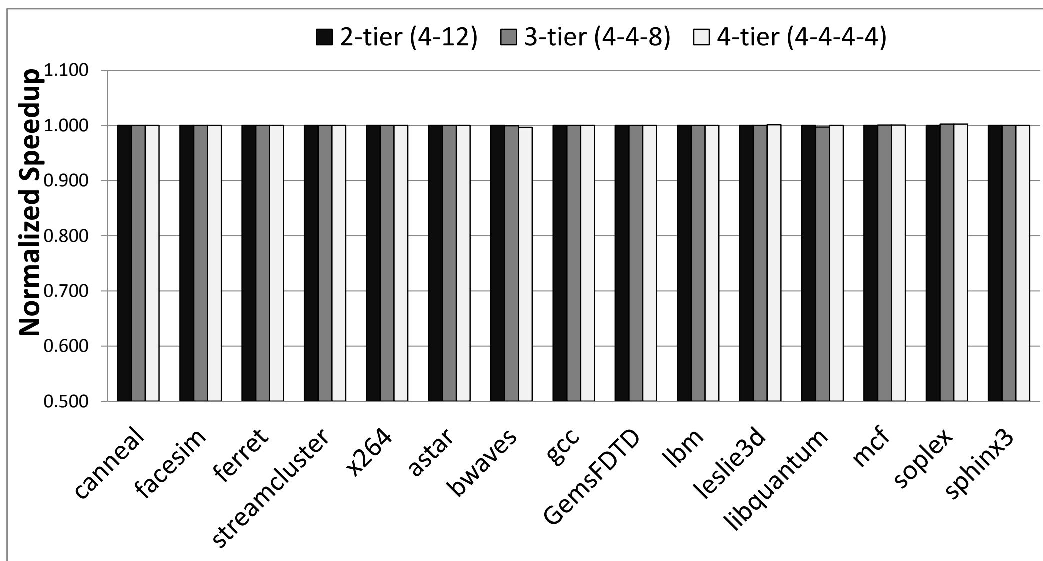

Figure 20: Sensitivity study of tier numbers.

The 2-tier migration scheme (refer to Figure 16(c)) we developed achieves significant performance improvements, as shown in Figure 21. For all workloads except `libquantum`, the maximum performances of the 10 test chips we generated surpass the ideal baseline. The highest is seen for mcf which is 7% faster than the ideal baseline. This is because the 2-tier migration scheme can fully utilize the fast subbanks such that the effective LLC access latency is even shorter than the ideal baseline. For the average performance of each workload (the height of the bars), there are also 3 workloads: `GemsFDTD`, `mcf`, and `sphinx3` that are faster than the ideal baseline. The Geometric mean of all workloads achieves 99.55% of the performance of the ideal baseline. Having a fast tier of 4 subbanks are particularly effective for workloads that have high LLC requirement (or high L2 MPKI) such as `canneal`,

GemsFDTD, lbm and mcf as their performances boost dramatically from "bank-migration" to "bank-migration-2-tier" in the figure.

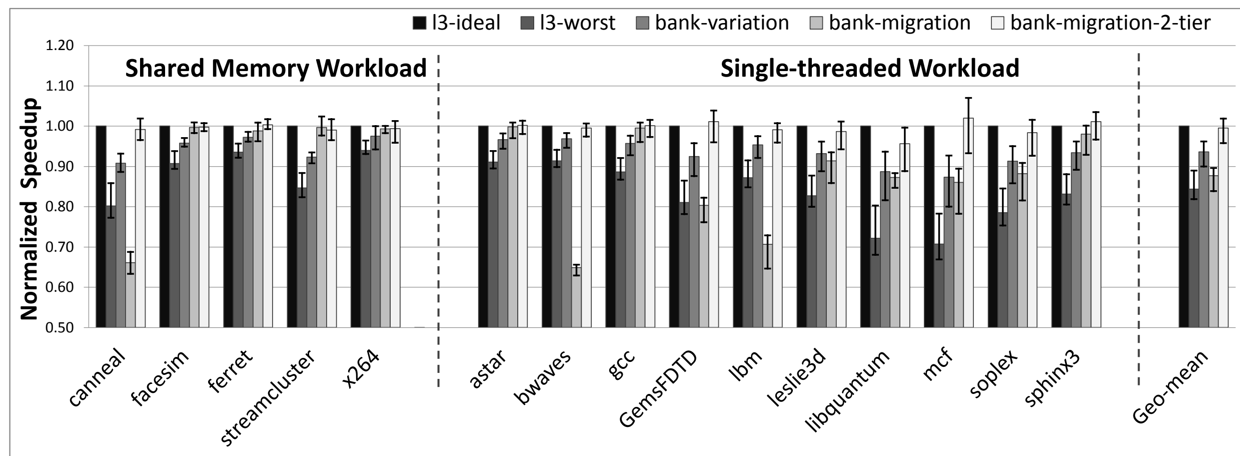

Figure 21: Performance of the proposed 2-tier migration technique, normalized to the ideal baseline.

5.3.3 Energy Conservation: Intra-Column Two-Tier Migration

Having seen the great performance advantages of the 2-tier migration scheme, we now turn into its energy consumption as the migration activities are extra energy overhead. First, let us understand how much migration activities are introduced to the memory. This is reported as the percentage of LLC accesses that trigger migrations in Figure 22. Since our fast tier has 4 subbanks (384MB), most workload's working set can be fit into this capacity. Hence, many workloads do not have noticeable migration overhead — they are only for the cold start stage. For the 6 workloads that have >1% migration activity, we further study their LLC regular dynamic energy consumption vs. migration energy. The results are in Figure 23. The data series labeled with "intra-column" will be discussed later. We can see that for 2-tier design, the migration energy can be more than 50% of the regular access energy for 4 of the 6 benchmarks. Hence, it is necessary to revise our 2-tier migration scheme to lower this part.

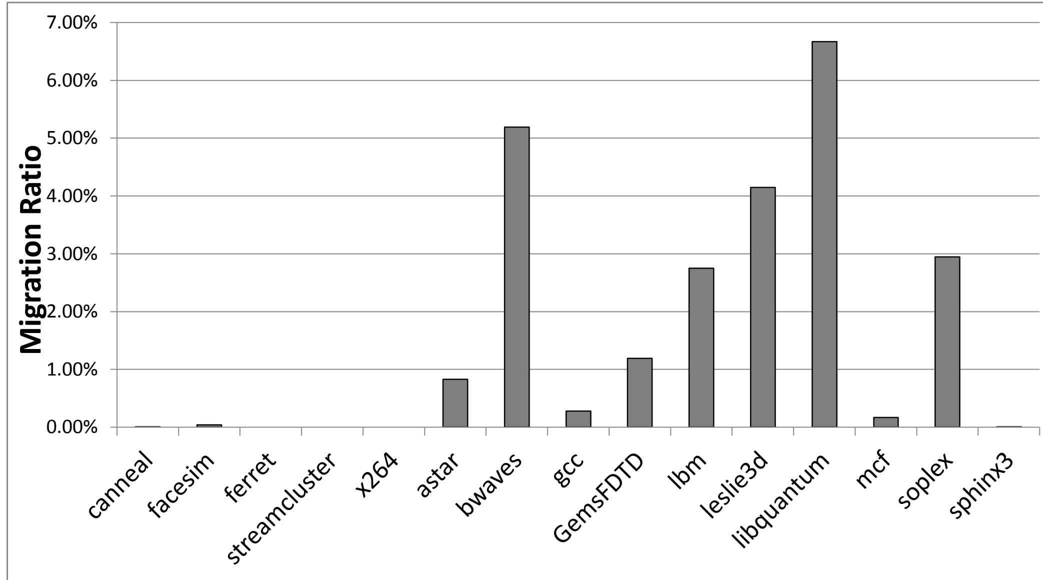

Figure 22: Percentage of migration activities.

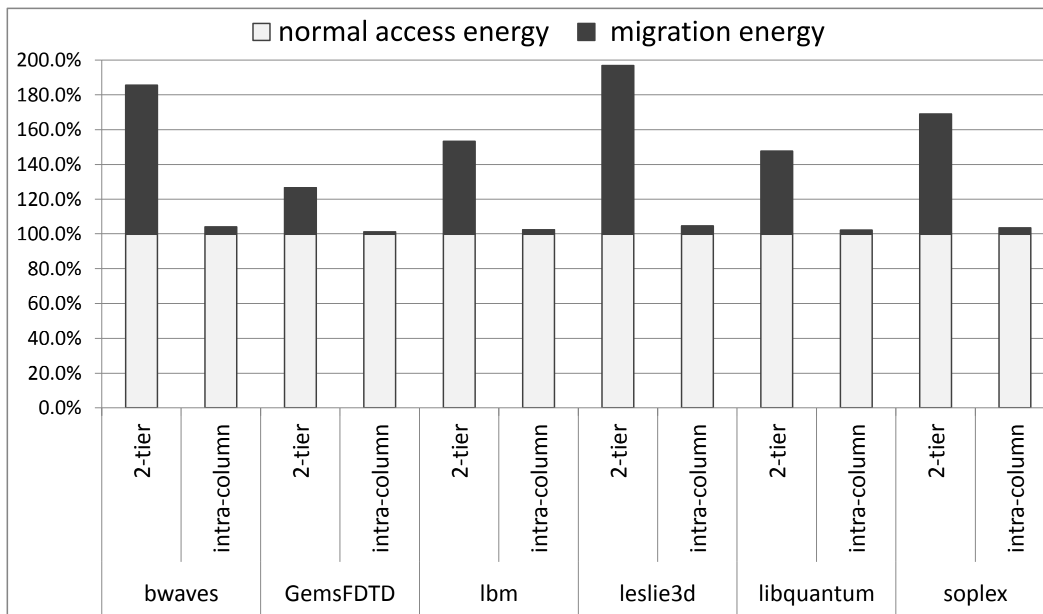

Figure 23: Energy comparison for 2-tier migration and intra-column migration.

To reduce the migration energy, we have observed that a significant portion of the migration energy is consumed on the wires in the interface layer connecting the four row buffers and tag arrays (refer to Figure 8). Table 8 in subsection 4.4.3 lists the energy breakdown

for a LLC access. As we can see, the $0.325nJ$, which is the wires in the interface layer is the largest amount in the table. Since 3D memory design has shorter interconnection wires due to stacking, any 2D wires that cannot be eliminated become more pronounced. The energy spent on the wires in the interface layer is due to the movement of data between subbanks in different columns. Hence, we revise our 2-tier migration scheme into an intra-column migration scheme. Instead of migrating among different columns via the wires on the interface layer, we enforce the migration to happen within each column. There are still 2 tiers in the memory rank. The difference now is that the fast tier is composed of the fastest subbank in each column. A hit into a slow subbank will only move the data to the fastest subbank in its own column. Hence, all migrations happen only within each column of subbanks. This is depicted in Figure 16(d). The intra-column migration technique saves significant energy, compared to the original 2-tier migration scheme. The comparisons are shown in Figure 23. As we can see, intra-column migration has nearly negligible migration energy overhead, which is more than ∼20 times lower than the 2-tier design for all workloads. Also, since intra-column migration distributes fast tier into all columns, it eliminates the possibility that the fast tier concentrates in one column, and thus can guarantee the even distribution of power and temperature across the interface layer. What is more important is that the intra-column

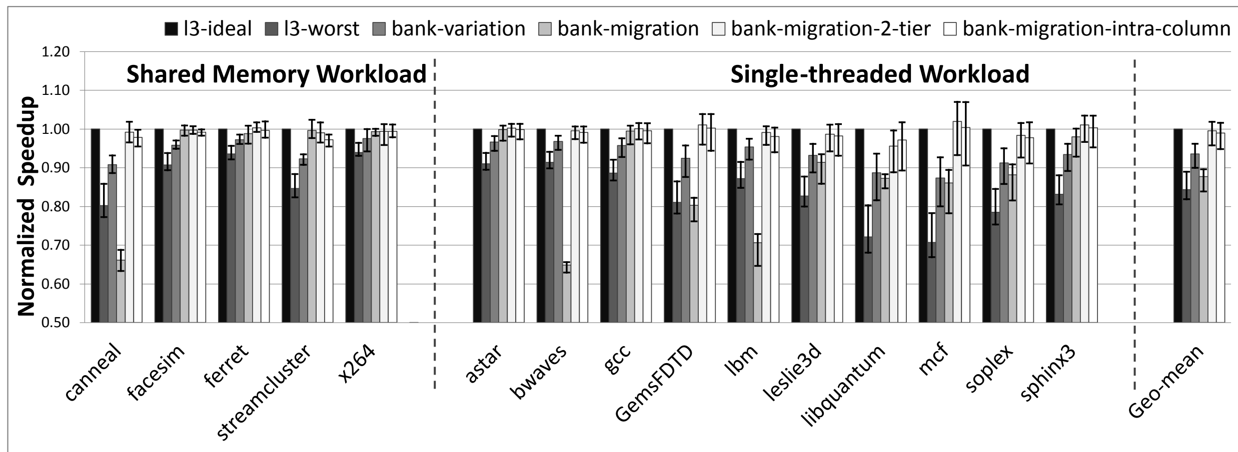

Figure 24: Performance of the improved intra-column 2-tier migration technique, normalized to the ideal baseline.

migration, though it changed the subbanks in the fast tier, was not hurt in performance. This is shown in Figure 24. Both 2-tier migration schemes are very close to each other for all workloads. The average of intra-column is 98.93% of the ideal baseline. They are better than the pessimistic baseline by 18%. Finally, we also measured the energy-delay product for the entire chip using methodologies similar to [36] and compared them among different PV-tolerant schemes. The results are shown in Figure 25. Since our proposed intra-column 2-tier migration scheme achieves almost identical performance as the ideal baseline, and its energy overhead is also negligible, it follows naturally that the energy-delay^2 product is only 3% away from the ideal baseline. They are however 42% better than the pessimistic baseline. We therefore conclude that this is the best PV-tolerant cache management scheme.

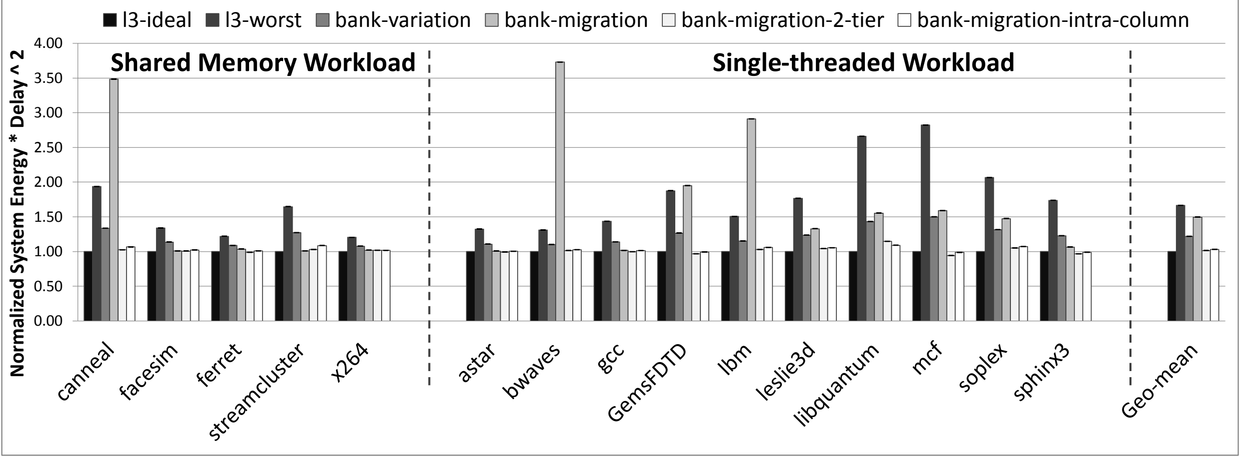

Figure 25: Energy-Delay product results.

There are also a number of recent works that are related to managing the non-uniformity of cache accesses in 3D chips [58], including those that utilize emerging memory technologies such as Phase Change Memories and Magnetic Memories [27, 63]. However, the fundamental cache non-uniformities of those works are still two-dimensional such that they are managed within one layer. Migrating data vertically did not benefit because vertical communication latency was uniform (well below $1us$) due to the small die thickness. However, as we can see from our analyses, such vertical data migration is critical to performance with process variations.

6.0 CONCLUSION & FUTURE WORK

By modeling the process variations in a multi-layered 3D DRAM stack integrated with a multi-core processor, we observed that the DRAM subbanks in 3D space present a fairly wide range of access latencies due to both WID and D2D variations. Hence, it is important to consider the impact of PV on the performance in a 3D chip with DRAM stacks. Our evaluation indicates that the performance of a 3D chip with PV can be degraded by 16% if the speed of the slowest subbank is used as the nominal subbank speed. We developed and compared cache management techniques to overcome the non-uniform access times from different memory subbanks, and reached a final scheme based on data migration between tiers within a column of subbanks. Our scheme can overcome PV in memories such that the resulting speed is only 0.45% away from a chip with no PV. The energy overhead is also minimized, resulting an ED^2 which is 42% better than a conservative chip and only 3% worse than the ideal chip. Hence, our scheme is effective and efficient to handle the process variations in such a complex 3D system.

As we focused on the PV in 3D DRAM LLC in this work, PVs in other structures are ignored for simplicity. When the PVs in the entire system are considered, it will become a more realistic, more complex non-uniformity problem. On the other hand, as thermal problem arises as the major challenge in 3D stacked multi-layer system, studying temperature distribution and it induced performance and leakage issues could also be an aspect of future work.

BIBLIOGRAPHY

[1] S. R. Nassif, "Modeling and Forecasting of Manufacturing Variations", *Asia and South Pacific Design Automation Conference*, pp. 145-149, 2001.

[2] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, J. Torrellas, "VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects", *IEEE Transactions on Semicon-ductor Manufacturing*, Vol. 21, No. 1, 2008.

[3] J. Tschanz, K. Bowman, V. De, "Variation-Tolerant Circuits: Circuit Solutions and Techniques", *Design Automation Conference*, pp. 762-773, 2005.

[4] Y. Cao, L. T. Clark, "Mapping Statistical Process Variations Toward Circuit Performance Variability: An Analytical Modeling Approach", *Design Automation Conference*, pp. 658-663, 2005.

[5] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, C. Spanos, "Modeling Within-Die Spatial Correlation Effects for Process-Design Co-Optimization", *International Symposium on Quality Electronic Design*, 2005.

[6] A. Srivastava, D. Sylvester, and D. Blaauw, "Statistical Analysis and Optimization for VLSI: Timing and Power", *New York: Springer*, 2005.

[7] W. R. Davis, J. Wilson, S. Mick, J. Xu, H. Hua, C. Mineo, A. M. Sule, M. Steer, P. D. Franzon, "Demystifying 3D ICs: The Pros and Cons of Going Vertical", *IEEE Design & Test*, Vol. 22, No. 6, pp. 498-510, 2005.

[8] V. Suntharalingam, R.Berger, J.A. Burns, C.K. Chen, C.L. Keast, J.M. Knecht, R.D. Lambert, K.L. Newcomb, D.M. O'Mara, D.D. Rathman, D.C. Shaver, A.M. Soares, C.N. Stevenson, B.M. Tyrrell, K. Warner, B.D. Wheeler, D.-R.W. Yost, D.J. Young, "Megapixel CMOS Image Sensor Fabricated in Three-Dimensional Integrated Circuit Technology", *IEEE International Solid-State Circuits Conference*, Vol.1, pp. 356-357, 2005.

[9] K. Banerjee, S. J. Souri, P. Kapur, K. C. Saraswat, "3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration", *Proceedings of IEEE*, 89(5), pp. 602-533, 2001.

[10] R.Venkatraman, R.Castagnetti, S.Ramesh, "The Statistics of Device Variations and its Impact on SRAM Bitcell Performance, Leakage and Stability", *International Symposium on Quality Electronic Design*, pp. 190-195, 2006

[11] A. Agarwal, B. C. Paul, S. Mukhopadhyay, K. Roy, "Process Variation in Embedded Memories: Failure Analysis and Variation Aware Architecture", *IEEE Journal of Solid-State Circuits*, pp. 1804-1814, 2005.

[12] T. Fischer, A. Olbrich, G. Georgakos, B. Lemaitre, D. Schmitt-Landsiedel, "Impact of Process Variations and Long Term Degradation on 6T-SRAM Cells", *Advances in Radio Science*, Volume 5, pp. 321-325, 2007.

[13] C. Tsai, M. Marek-Sadowska, "Analysis of Process Variation's Effect on SRAM's Read Stability", *International Symposium on Quality Electronic Design*, pp. 603-610, 2006.

[14] S. Mukhopadhyay, K. Kim, H. Mahmoodi, K. Roy, "Design of a Process Variation Tolerant Self-Repairing SRAM for Yield Enhancement in Nanoscaled CMOS", *IEEE Journal of Solid-State Circuits*, Vol. 42, No. 6, pp. 1370-1382, 2007.

[15] J. P. Kulkarni, K. Kim, S. P. Park, K. Roy, "Process Variation Tolerant SRAM Array for Ultra Low Voltage Applications", *Design Automation Conference*, pp. 108-113, 2008.

[16] J. Singh, J. Mathew, D.K. Pradhan, S.P. Mohanty, "Failure Analysis for Ultra Low Power Nano-CMOS SRAM Under Process Variations", *International SOC Conference*, pp. 251-254, 2008.

[17] S. Gupta, M. Hilbert, S. Hong, and R. Patti, "Techniques for Producing 3D ICs with High-Density In-terconnect", *International VLSI Multilevel Interconnection Conference*, 2004.

[18] C. Kim, D. Burger, S. W. Keckler, "An Adaptive, Non-uniform Cache Structure for Wire-Delay Dominated On-Chip Caches", *International Conference on Architectural Support for Programming Languages and Operating Systems*, pp.211-222, 2002.

[19] K. Sakuma, P.S. Andry, C.K. Tsang, K. Sueoka, Y. Oyama, C. Patel, B. Dang, S.L. Wright, B.C. Webb, E. Sprogis, R. Polastre, R. Horton, J.U. Knickerbocker, "Characterization of Stacked Die using Die-to-Wafer Integration for High Yield and Throughput", *Electronic Components and Technology Conference*, pp. 18-23, 2008.

[20] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. Loh, D. McCaule, P. Morrow, D. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen, C. Webb, "Die-Stacking (3D) Microarchi-tecture", *International Symposium on Microarchitecture*, pp. 469-479, 2006.

[21] Tezzaron Semiconductors, "FaStack Memory", http://www.tezzaron.com/memory/FaStack_memory.html

[22] Tezzaron Semiconductors, "3D Stacked DRAM",
http://www.tezzaron.com/memory/Overview_3D_DRAM.htm

[23] G. Loh, "3D-Stacked Memory Architecture for Multi-Core Processors", *International Symposium on Computer Architecture*, pp. 453-464, 2008.

[24] Sun Microsystems, "UltraSPARC T2 processor datasheet",
http://www.sun.com/processors/UltraSPARC-T2/

[25] P. Zhou, B. Zhao, J. Yang, Y. Zhang, "A Durable and Energy Efficient Main Memory Design Using Phase Change Memory Technology", to appear, *International Symposium on Computer Architecture*, 2009.

[26] S. Hebert, D. Marculescu, "Variation-Aware Dynamic Voltage/Frequency Scaling", *International Symposium on High Performance Computer Architecture*, pp. 301-312, 2009.

[27] X. Wu, Y. Xie, J. Li, L. Zhang, E. Speight, R. Rajamony, "Hybrid Cache Architecture with Disparate Memory Technologies", to appear, *International Symposium on Computer Architecture*, 2009.

[28] R Development Core Team, "R: A Language and Environment for Statistical Computing", *R Foundation for Statistical Computing*, 2006, http://www.R-project.org

[29] P. Ribeiro, P. Diggle, "geoR: A package for geostatistical analysis", *R-NEWS*, Vol. 1, No. 2, 2001.

[30] R. Teodorescu, J. Torrellas, "Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors", *International Symposium on Computer Architecture*, pp. 363-374, 2008.

[31] X. Liang, R. Canal, G.-Y. Wei, D. Brooks, "Process Variation Tolerant 3T1D-Based Cache Archi-tectures", *International Symposium on Microarchitecture*, pp. 15-26, 2007.

[32] Tezzaron Semiconductors, "Bi-STAR Technology",
http://www.tezzaron.com/technology/Bi_STAR.htm

[33] N. Cressie, "Statistics for Spatial Data", *New York: Wiley*, 1993.

[34] Arizona State University, "Predictive Technology Model (PTM)",
http://www.eas.asu.edu/~ptm/

[35] W. Zhao, Y. Cao, "New generation of Predictive Technology Model for sub-45nm early design ex-ploration", *IEEE Transactions on Electron Devices*, Vol. 53, No. 11, pp. 2816-2823, 2006.

[36] S. Thoziyoor, J. H. Ahn, M. Monchiero, J. B. Brockman, N. P. Jouppi, "A Comprehensive Memory Modeling Tool and its Application to the Design and Analysis of Future

Memory Hierarchies", *Interna-tional Symposium on Computer Architecture*, pp. 51-62, 2008.

[37] W. Mueller, G. Aichmayr, W. Bergner, E. Erben, T. Hecht, C. Kapteyn, A. Kersch, S. Kudelka, F. Lau, J. Luetzen, A. Orth, J. Nuetzel, T. Schloesser, A. Scholz, U. Schroeder, A. Sieck, A. Spitzer, M. Strasser, P.-F. Wang, S. Wege, R. Weis, "Challenges for the DRAM Cell Scaling to 40nm", *IEEE International Electron Devices Meeting*, pp. 336-339, 2005.

[38] T. Kgil, S. D'Souza, A. Saidi, N. Binkert, R. Dreslinski, T. Mudge, S. Reinhardt, K. Flautner, "Pi-coServer: Using 3D Stacking Technology to Enable a Compact Energy Efficient Chip Multiprocessor", *International Conference on Architectural Support for Programming Languages and Operating Systems*, pp. 117-128, 2006.

[39] C.C. Liu, I. Ganusov, M. Burtscher, S. Tiwari, "Bridging the Processor-Memory Performance Gap with 3D IC Technology", *IEEE Design and Test of Computers*, 22(6), pp. 556-564, 2005.

[40] G. L. Loi, B. Agarwal, N. Srivastava, S. Lin, T. Sherwood, "A Thermally-Aware Performance Analysis of Vertically Integrated (3D) Processor-Memory Hierarchy", *Design Automation Conference*, pp. 991-996, 2006.

[41] K. A. Bowman, S. G. Duvall, J. D. Meindl, "Impact of Die-to-Die And Within-die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration", *IEEE Jounal of Solid-State Circuits*, Vol. 37, No. 2, pp. 183-190, 2002.

[42] S. G. Duvall, "Statistical Circuit Modeling and Optimization", *International Workshop on Statistical Metrology*, pp. 56-63, 2000.

[43] M. Orshansky, L. Milor, L. Nguyen G. Hill, Y. Peng, C. Hu, "Intra-field gate CD variability and its impact on circuit performance", *International Electron Devices Meeting*, pp. 479-482, 1999.

[44] S. Lee, C. Choi, J. Kong, W. Lee, J. Yoo, "An Efficient Statistical Analysis Methodology and Its Application to High-density DRAMs", *International Conference on Computer-Aided Design*, pp. 678-683, 1997.

[45] A. Agarwal, B. C. Paul, H. Mahmoodi, A. Datta, K. Roy, "A Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technologies", *IEEE Transactions on Very Large Scale Integrated Systems*, Vol. 13, pp. 27-38, 2005.

[46] S. Ozdemir, D. Sinha, G. Memik, J. Adams, H. Zhou, "Yield-Aware Cache Architectures", *International Symposium on Microarchitecture*, pp. 15-25, 2006.

[47] A. Das, B. Ozisikyilmaz, S. Zademir, G. Memik, J. Zambreno, A. Choudhary, "Evaluating the Effects of Cache Redundancy on Profit", *International Symposium on Microarchitecture*, pp. 388-398, 2008.

[48] X. Fu, T. Li, J. Fortes, "NBTI Tolerant Microarchitecture Design in the Presence of Process Variation", *International Symposium on Microarchitecture*, pp. 398-410, 2008.

[49] X. Fu, T. Li, J. Fortes, "Soft Error Vulnerability Aware Process Variation Mitigation", *International Symposium on High Performance Computer Architecture*, pp. 93-104, 2009.

[50] S. Sarangi, B. Greskamp, A. Tiwari, J. Torrellas, "EVAL: Utilizing Processors with Variation-Induced Timing Errors", *International Symposium on Microarchitecture*, pp. 423-434, 2008.

[51] E. Chun, Z. Chishti, T. N. Vijaykumar, "Shapeshifter: Dynamically Changing Pipeline Width and Speed to Address Process Variations", *International Symposium on Microarchitecture*, pp. 411-422, 2008.

[52] X. Liang and D. Brooks, "Mitigating the Impact of Process Variations on CPU Register File and Execution Units", *International Symposium on Microarchitecture*, pp. 504-514, 2006.

[53] X. Liang, G.-Y. Wei, D. Brooks, "ReVIVaL: A Variation-Tolerant Architecture Using Voltage In-terpolation and Variable Latency", *International Symposium on Computer Architecture*, pp. 191-202, 2008.

[54] A. Tiwari, S. R. Sarangi, J. Torellas, "ReCycle: Pipeline Adaptation to Tolerate Process Variation", *International Symposium on Computer Architecture*, pp. 323-334, 2007.

[55] N. Madan, L. Zhao, N. Muralimanohar, A. Udipi, R. Balasubramonian, R. Iyer, S. Makineni, D. Newell, "Optimizing Communication and Capacity in a 3D Stacked Reconfigurable Cache Hierarchy", *In-ternational Symposium on High Performance Computer Architecture*, pp. 262-274, 2009.

[56] Z. Chishti, M. Powell, T. N. Vijaykumar, "Distance Associativity for High-Performance Ener-gy-Efficient Non-Uniform Cache Architectures", *International Symposium on Microarchitecture*, pp. 55, 2003.

[57] S. Cho, L. Jin, "Managing Distributed, Shared L2 Caches Through OS-Level Page Allocation", *In-ternational Symposium on Microarchitecture*, pp. 455-465, 2006.

[58] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, M. Kandemir, "Design and Management of 3D Chip Multi-processors Using Network-in-Memory", *International Symposium on Computer Architecture*, pp. 130-141, 2006.

[59] Virtutech Simics, http://www.virtutech.com

[60] The PARSEC Benchmark Suite, http://parsec.cs.princeton.edu

[61] S. Cho, L. Jin, "Managing Distributed, Shared L2 Caches Through OS-Level Page Allocation", *In-ternational Symposium on Microarchitecture*, pp. 455-465, 2006.

[62] A. Das, B. Ozisikyilmaz, S. Zademir, G. Memik, J. Zambreno, A. Choudhary, "Evaluating the Effects of Cache Redundancy on Profit", *International Symposium on Microarchitecture*, pp. 388-398, 2008.

[63] G. Sun, X. Dong, Y. Xie, J. Li, Y. Chen, "A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs", *International Symposium on High Performance Computer Architecture*, pp. 239-249, 2009.