# Homayoun

# Reference 43

# A Novel Architecture of the 3D Stacked MRAM L2 Cache for CMPs

Guangyu Sun[†], Xiangyu Dong[†], Yuan Xie[†], Jian Li[‡], Yiran Chen[§]
[†]Pennsylvania State University, [‡]IBM Austin Research Lab, [§]Seagate Technology
[†]{gsun, xydong, yuanxie}@cse.psu.edu, [‡]jianli@us.ibm.com, [§]yiran.chen@seagate.com

## Abstract

*Magnetic random access memory (MRAM) is a promising memory technology, which has fast read access, high density, and non-volatility. Using 3D heterogeneous integrations, it becomes feasible and cost-efficient to stack MRAM atop conventional chip multiprocessors (CMPs). However, one disadvantage of MRAM is its long write latency and its high write energy. In this paper, we first stack MRAM-based L2 caches directly atop CMPs and compare it against SRAM counterparts in terms of performance and energy. We observe that the direct MRAM stacking might harm the chip performance due to the aforementioned long write latency and high write energy. To solve this problem, we then propose two architectural techniques:* read-preemptive write buffer *and* SRAM-MRAM hybrid L2 cache. *The simulation result shows that our optimized MRAM L2 cache improves performance by 4.91% and reduces power by 73.5% compared to the conventional SRAM L2 cache with the similar area.*[1]

## 1 Introduction

The diminishing return of endeavors to increase clock frequencies and exploit instruction level parallelism in a single processor have led to the advent of chip multiprocessors (CMPs) [8]. The integration of multiple cores on a single chip is expected to accentuate the already daunting "memory wall" problem [6] and it becomes a major challenge of supplying massive multi-core chips with sufficient memories.

The introduction of the three-dimensional (3D) integration technology [9, 26] provides the opportunity of stacking memories atop compute cores and therefore alleviates the memory bandwidth challenge of CMPs. Recently, active research [4, 13, 22] has targeted SRAM caches or DRAM memories stacking.

Magnetic Random Access Memory (MRAM) is a promising memory technology with attractive features such as fast read access, high density, and non-volatility [14, 27]. However, previous research on leveraging MRAM as on-chip memories is very limited. How to integrate MRAM into compute cores on planular chips is the key obstacle since the MRAM fabrication involves hybrid magnetic-CMOS processes. Fortunately, 3D integrations enable the cost-efficient integration of heterogeneous technologies, which is ideal for MRAM stacking atop compute cores. Some recent work [10, 12] has evaluated the benefits of MRAM as a universal memory replacement for L2 caches and main memories in single-core chips.

In this paper, we further evaluate the benefits of stacking MRAM L2 caches atop CMPs. We first develop a cache model for stacking MRAM and then compare the MRAM-based L2 cache against its SRAM counterpart with the similar area in terms of performance and energy. The comparison shows that: (1) For applications that have moderate write intensities to L2 caches, the MRAM-based cache can reduce the total cache power significantly because of its zero standby leakage and achieve considerable performance improvement because of its relatively larger cache capacity; (2) For applications that have high write intensities to L2 caches, the MRAM-based cache can cause performance and power degradations due to the long latency and the high energy of MRAM write operations.

These two observations imply that MRAM-based caches might not work efficiently if we directly introduce them into the traditional CMP architecture because of their disadvantages on write latency and write energy. In light of this concern, we propose two architectural techniques, *read-preemptive write buffer* and *SRAM-MRAM hybrid L2 cache*, to mitigate the MRAM write-associated issues. The simulation result shows that performance improvement and power reduction can be achieved effectively with our proposed techniques even under the write-intensive workloads.

## 2 Background

This section briefly introduces the background of MRAM and 3D integration technologies.

### 2.1 MRAM Background

The basic difference between the MRAM and the conventional RAM technologies (such as SRAM/DRAM) is that the information carrier of MRAM is Magnetic Tunnel Junctions (MTJs) instead of electric charges [27]. As shown in Fig. 1, each MTJ contains *a pinned layer* and *a free layer*. The *pinned layer* has fixed magnetic direction

[1]This work was supported in part by NSF grants (CAREER 0643902, CCF 0702617, CSR 0720659), a gift grant from Qualcomm, and IBM Faculty Award.

978-1-4244-2932-5/08/$25.00 ©2008 IEEE

while the *free layer* can change its magnetic direction by spin torque transfers [14]. If the free layer has the same direction as the pinned layer, the MTJ resistance is low and indicates state "0"; otherwise, the MTJ resistance is high and indicates state "1".

The latest MRAM technology (spin torque transfer ram, STT-RAM) changes the magnetic direction of the free layer by directly passing spin-polarized currents through MTJs. Comparing to the previous generation of MRAM using external magnetic fields to reverse the MTJ status, STT-RAM has the advantage of scalability, which means the *threshold current* to make the status reversal will decrease as the size of the MTJ becomes smaller. In this paper, we use the terms "MRAM" and "STT-RAM" equivalently.

The most popular structure of MRAM cells is composed of one NMOS transistor as the access device and one MTJ as the storage element ("1T1J"structure) [14]. As illustrated in Fig. 1, the storage element, MTJ, is connected in series with the NMOS transistor. The NMOS transistor is controlled by the the word line (WL) signal. The detailed read and write operations for each MRAM cell is described as follows:

- **Read Operation**: When a *read operation* happens, the NMOS is turned on and a small voltage difference (-0.1V as demonstrated in [14]) is applied between the bit line (BL) and the source line (SL). This voltage difference causes a current through the MTJ whose value is determined by the status of MTJs. A sense amplifier compares this current to a reference current and then decides whether a "0" or a "1" is stored in the selected MRAM cell.
- **Write Operation**: When a *write operation* happens, a large positive voltage difference is established between SLs and BLs for writing for "0"s or a large negative one for writing "1"s. The current amplitude required to ensure a successful status reversal is called threshold current. The current is related to the material of the tunnel barrier layer, the writing pulse duration, and the MTJ geometry [11].

In this work, we use the writing pulse duration of $10ns$ [27], below which the writing threshold current will increase exponential. In addition, we scale the MRAM size of previous work [14] down to $65nm$ technology node. Assuming the size of MTJs is $65nm \times 90nm$, the derived threshold current for magnetic reversal is about $195\mu A$.

### 2.2 3D Integration Overview

The 3D integration technology has recently emerged as a promising means to mitigate interconnect-related problems. By using the vertical through silicon via (TSV), multiple active device layers can be stacked together (through wafer stacking or die stacking) in the third dimension [26].

3D integrations offer a number of advantages over traditional two-dimensional (2D) designs [9]: (1) shorter global interconnects because the vertical distance (or the length of TSVs) between two layers is usually in the range of 10 $\mu m$ to 100 $\mu m$ [26] depending on manufacturing processes; (2) higher performance because of reducing the average interconnect length; (3) lower interconnect power consumption due to the wire length reduction; (4) denser form factor and smaller footprint; (5) support for the cost-efficient integration of heterogenous technologies.

In this paper, we rely on the 3D integration technology to stack a massive amount of L2 caches (2MB for SRAM caches and 8MB for MRAM caches) on top of CMPs. Furthermore, the heterogenous technology integration enabled by 3D makes it feasible to fabricate MRAM caches and CMP logics as two separate dies and then stack them together in a vertical way. Therefore, the magnetic-related fabrication process of MRAM will not affect the normal CMOS logic fabrication and keep the integration cost-efficient.

## 3 MRAM and Non-Uniform Cache Access (NUCA) Models

In this section, we describe an MRAM circuit model and a NUCA model which is implemented with Network-on-Chip (NoC).

### 3.1 MRAM Modeling

To model MRAM, we first estimate the area of MRAM cells. As shown in Fig. 1, each MRAM cell is composed of one NMOS transistor and one MTJ. The size of MTJs is only limited by manufacturing techniques, but the NMOS transistor has to be sized properly so that it can drive sufficiently large current to change the MTJ status. The current driving ability of NMOS transistor is proportional to its W/L ratio. Using HSPICE simulation, we find that the minimum W/L ratio for the NMOS transistor under $65nm$ technology node is around 10 to drive the threshold writing current of $195\mu A$. We further assume the width of the source or drain regions of an NMOS transistor is $1.5F$, where $F$ is the feature size. Therefore, we estimate the MRAM cell size is about $10F \times 4F = 40F^2$. The parameters of our targeted MRAM cell are tabulated in Table .

**Table 1. MRAM Cell Specifications**

| | |
|---|---|
| Technology | $65nm$ |
| Write Pulse Duration | $10ns$ |
| Threshold Current | $195\mu A$ |
| Cell Size | $40F^2$ |
| Aspect Ratio | 2.5 |

Despite the difference in storage mechanisms, MRAM and SRAM have the similar peripheral interfaces from the circuit designers' points of view. By simulating with a modified version of CACTI [2], our result shows that the area of a 512KB MRAM cache is similar to a 128KB SRAM cache

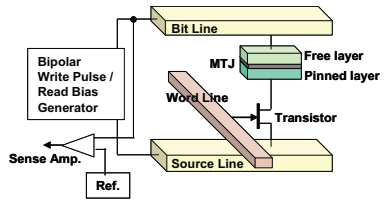


**Figure 1. An illustration of an MRAM cell**

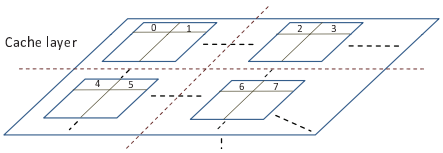


**Figure 2. Eight caches ways are distributed in four banks. Assume four cores and accordingly four zones each layer.**

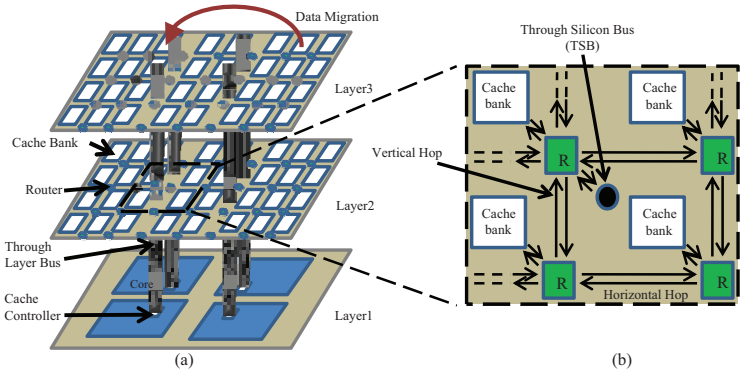


**Figure 3. (a) An illustration of the proposed 3D NUCA structure, which includes 1 core layer, 2 cache layers. There are 4 processing cores per core layer, 32 cache banks per cache layer, and 4 through-layer-bus across layers; (b) Connections amongst routers, caches banks and through-layer-buses.**

whose cell is about $146F^2$ (this value is extracted from CACTI). Table 2 lists the comparison between a *512KB MRAM cache bank* and a *128KB SRAM cache bank*, which are used later in this paper, in terms of area, access time, and access energy.

**Table 2. Comparison of area, access time, and energy comparison(65nm technology)**

| Cache size | 128KB SRAM | 512KB MRAM |
|---|---|---|
| Area | $3.62mm^2$ | $3.30mm^2$ |
| Read Latency | $2.252ns$ | $2.318ns$ |
| Write Latency | $2.264ns$ | $11.024ns$ |
| Read Energy | $0.895nJ$ | $0.858nJ$ |
| Write Energy | $0.797nJ$ | $4.997nJ$ |

### 3.2 Modeling 3D NUCA Cache

As the caches capacity and area increase, the wire delay has made the Non-Uniform Cache Access (NUCA) architecture [18] more attractive than the conventional Uniform Cache Access (UCA) one. In NUCA, the cache is divided into multiple banks with different access latencies according to their locations relative to cores and these banks can be connected through a mesh-based Network-on-Chip (NoC).

Extending the work of CACTI [2], we develop our NoC-based 3D NUCA model. The key concept is to use NoC routers for communications within planular layers, while using a specific through silicon bus (TSB) for communications among different layers. Figure 3(a) illustrates an example of the 3D NUCA structure. There are four cores located in the *core layer* and 32 cache banks in each *cache layer* and all layers are connected by through silicon bus (TSB) which is implemented with TSVs. This interconnect style has the advantage of short connections provided by 3D integrations. It has been reported the vertical latency of traversing a 20-layer stack is only $12ps$ [23], thus the latency of TSB negligible compared to the latency of 2D NoC routers. Consequently, it is feasible to have single-hop vertical communications by utilizing TSBs. In addition, hybridization of 2D NoC routers with TSBs require one (instead of two) additional link on each NoC router, because TSB can move data both upward and downward [20].

As shown in Figure3(a), cache layers are on top of core layers and they can either SRAM or MRAM caches. Figure3(b) shows a detailed 2D structure of cache layers. Every four cache banks are grouped together and routed to other layers via TSBs.

Similar to prior approaches [7, 20], the proposed model supports *data migration*, which moves data closer to their accessing core. For set-associative cache, the cache ways belonging to the set should be distributed into different banks so that data migration can be implemented. In our 3D NUCA model, each cache layer is equally divided into several zones. The number of zones is equal to the number of cores and each zone has a TSB located at its center. The cache ways of each set are uniformly distributed into these zone. This architecture promises that, within each cache set, there are several ways of cache lines close to the active core. Fig. 2 gives an illustration of distributing eight ways into four zones. Fig. 3(a) shows an example of data migration after which the core in the upper-left corner can access the data faster. In this paper, this kind of data migrations is called *inter-migration* to differentiate another kind of migration policy introduced later.

The advantages of this 3D NUCA cache are:(1) placing L2 caches in separate layers makes it possible to integrate MRAM with traditional CMOS process technology; (2) separating cores from caches simplifies the design of TSBs and routers because TSBs are now connected to cache controllers directly, and there is no direct connection be-

**Table 3. Baseline configuration parameters**

| | |
|---|---|
| **Processors** | |
| # of cores | 8 |
| Frequency | 3GHz |
| Power | 6W/core |
| Issue Width | 1 (in order) |
| **Memory Parameters** | |
| L1 cache | private, 16+16KB, 2-way, 64B line, 2-cycle, write-through, 1 read/write port |
| SRAM L2 | shared, 2MB (16x128KB), 32-way, 64B line, read/write per bank : 7-cycle, write-back, 1 read/write port |
| MRAM L2 | shared, 8MB (16x512KB), 32-way, 64B line, read penalty per bank : 7-cycle, write penalty per bank : 33-cycle, write-back, 1 read/write port |
| Write buffer | 4 entry, retire-at-2 |
| Main Memory | 4GB, 500-cycle latency |
| **Network Parameters** | |
| # of Layers | 2 |
| # of TSB | 8 |
| Hop latency | TSB 1 cycle, V_hop 1 cycle H_hop 1 cycle |
| Router Latency | 2-cycle |

tween routers and cache controllers.

We provide one TSB for each core in the model. Considering that the TSV pitch size is reported to be only 4-10$\mu m$ [23], thus even a 1024-bit bus (much wider than our proposed TSB) would only incur an area overhead of $0.32mm^2$. In our study, the die area of an 8-core CMP is estimated to be $60mm^2$ (discussed later). Therefore, it is feasible to assign one TSB for each core and the TSV area overhead is negligible.

### 3.3 Configurations and Assumptions

Our baseline configuration is an 8-core in-order processor using the Ultra SparcIII ISA. In order to predict the chip area, we investigate some die photos, such as Cell Processor [16], Sun UltraSPARC T1 [19], etc. and estimate the area of an 8-core CMP without caches to be $60mm^2$. By using our modified version of CACTI [2], we further learn that one cache layer fits to either a $2MB$ SRAM or an $8MB$ MRAM L2 cache assuming each cache layer has the similar area to that of core layer ($60mm^2$). The configurations are detailed in Table 3. Note that the power of processors is estimated based on the data sheet of real designs [16, 19].

We use the Simics toolset [24] for performance simulations. Our 3D NUCA architecture is implemented as an extended module in Simics. We use a few multi-threaded benchmarks from *OpenMP2001* [3] and *PARSEC* [1] suites. Since the performance and power of MRAM caches are closely related to transaction intensity, we select some simulation workloads as listed in Table 4 so that we have a wide range of transaction intensities to L2 caches. The average numbers of total transactions (TPKI)[2] and write transactions (WPKI) of L2 caches are listed in Table 4. For each simulation, we fast forward to warm up the caches and then run 3 billion cycles. We use the total IPC of all the cores as the performance metric.

**Table 4. L2 transaction intensities**

| Name | TPKI | WPKI |
|---|---|---|
| galgel | 1.01 | 0.31 |
| apsi | 4.15 | 1.85 |
| equake | 7.94 | 3.84 |
| fma3d | 8.43 | 4.00 |
| swim | 19.29 | 9.76 |
| streamcluster | 55.12 | 23.326 |

### 3.4 SNUCA and DNUCA

*Static NUCA* (SNUCA) and *Dymaic NUCA* (DNUCA) are two different implementations of the NUCA architecture proposed by Kim, *et al.* [18]. SNUCA statically partitions the address space across cache banks, which are connected via NoC; DNUCA dynamically migrates frequently accessed blocks to the closest banks. These two NUCA implementations result in different access patterns and variable write intensities. In our later simulations, we use both SNUCA-SRAM and DNUCA-SRAM L2 caches as our baselines when evaluating the performance and power benefits of MRAM caches.

## 4 Direct Replacing SRAM with MRAM as L2 Caches

In this section, we directly replace SRAM L2 caches with MRAM ones that have the comparable area, and show that without any optimization, a naive MRAM replacement will harm both performance and power when the workload write intensity is high.

### 4.1 Same Area Replacement

As shown in Table 2, a 128KB SRAM bank has the similar area as a 512KB MRAM bank does. Thereby, in order to keep the area of cache layers unchanged, it becomes reasonable to replace SRAM L2 caches with MRAM ones whose capacity is 3 times larger. We call this replacement strategy as "same area replacement".

Using this strategy, we integrate as many caches in the cache layers as possible. Considering our baseline SRAM

[2]TPKI is the number of total transactions per 1K instructions and WPKI is the number of write transactions per 1K instructions.

DOCKET ALARM

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

**LAW FIRMS**
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

**FINANCIAL INSTITUTIONS**
Litigation and bankruptcy checks for companies and debtors.

**E-DISCOVERY AND LEGAL VENDORS**
Sync your system to PACER to automate legal marketing.