Multi Retention Level STT-RAM Cache Designs with a Dynamic Refresh Scheme

Zhenyu Sun, Xiuyuan Bi, Hai (Helen) Li
Polytechnic Institute of New York University, 6 Metrotech Center, Brooklyn, NY, USA
szheny01@students.poly.edu, xbi01@students.poly.edu, hli@poly.edu

Weng-Fai Wong, Zhong-Liang Ong
National University of Singapore, 13 Computing Drive, Singapore
wongwf@comp.nus.edu.sg, zoo@nus.edu.sg

Xiaochun Zhu, Wenqing Wu
Qualcomm Incorporated, 5775 Morehouse Drive, San Diego, USA
xiaochun@qualcomm.com, wenqingw@qualcomm.com

ABSTRACT

Spin-transfer torque random access memory (STT-RAM) has received increasing attention because of its attractive features: good scalability, zero standby power, non-volatility and radiation hardness. The use of STT-RAM technology in the last level on-chip caches has been proposed as it minimizes cache leakage power with technology scaling down. Furthermore, the cell area of STT-RAM is only $1/9 \sim 1/3$ that of SRAM. This allows for a much larger cache with the same die footprint, improving overall system performance through reducing cache misses. However, deploying STT-RAM technology in L1 caches is challenging because of the long and power-consuming write operations. In this paper, we propose both L1 and lower level cache designs that use STT-RAM. In particular, our designs use STT-RAM cells with various data retention time and write performances, made possible by different magnetic tunneling junction (MTJ) designs. For the fast STT-RAM bits with reduced data retention time, a counter controlled dynamic refresh scheme is proposed to maintain the data validity. Our dynamic scheme saves more than 80% refresh energy compared to the simple refresh scheme proposed in previous works. A L1 cache built with ultra low retention STT-RAM coupled with our proposed dynamic refresh scheme can achieve 9.2% in performance improvement, and saves up to 30% of the total energy when compared to one that uses traditional SRAM. For lower level caches with relative large cache capacity, we propose a data migration scheme that moves data between portions of the cache with different retention characteristics so as to maximize the performance and power benefits. Our experiments show that on the average, our proposed multi retention level STT-RAM cache reduces $30 \sim 70\%$ of the total energy compared to previous works, while improving IPC performance for both 2-level and 3-level cache hierarchy.

MICRO'11, December 3-7, 2011, Porto Alegre, Brazil

Categories and Subject Descriptors

B.3.2 [**Memory Structures**]: Design Styles—*Cache memories*

General Terms

Design

1. INTRODUCTION

Continuously increasing capacity as well as cell leakage cause the standby power of SRAM on-chip caches to dominate the overall power consumption of the latest microprocessors. Many circuit design and architectural solutions, such as V_{DD} scaling [12], power-gating [15], and body-biasing [11], have been invented to reduce the standby power of caches. However, these techniques are becoming less efficient as technology continues to scale, causing the transistor's leakage current to increase exponentially. As the alternative of SRAM, the *spin-transfer torque RAM* (STT-RAM) is receiving significant attention because it offers almost all the desirable features of a universal memory: the fast (read) access speed of SRAM, the high integration density of DRAM, and the nonvolatility of Flash memory. Also, its compatibility with the CMOS fabrication process and similarities in the peripheral circuitries makes the STT-RAM an easy replacement for SRAM.

However, there are two major obstacles to use STT-RAM for on-chip caches, namely, its longer write latency and higher write energy. When the write access of a STT-RAM cell operates in the sub-10ns region, the *magnetic tunnel junction* (MTJ) resistance switching mechanism is dominated by *spin precession*. The required switching current rises exponentially as the MTJ switching time is reduced. As a consequence, the driving transistor's size must increase accordingly, leading to a larger memory cell area. The lifetime of memory cell also degrades exponentially as the voltage across the oxide barrier of the MTJ increases. As a result, a 10ns programming time is widely accepted as the performance limit of STT-RAM designs, and is adopted in mainstream STT-RAM research and development [23, 18, 10, 4, 6].

Several proposals have been made to address the write speed and energy limitations of STT-RAM. For example, the early write termination scheme [25] mitigates the performance degradation and energy overhead by eliminating unnecessary writes to STT-RAM cells. The dual write speed scheme [23] improve the average access time of STT-RAM cache by having a fast and a slow cache partition. A classic SRAM/STT-RAM hybrid cache hierarchy with 3D stacking structure was proposed in [18].

The *data retention time* indicates how long data can be retained in a nonvolatile memory cell after being written. In other words, it is the unit to measure nonvolatility of a memory cell. Relaxing this nonvolatility can make the memory cells easier to be programmed, and leads to a lower write current or faster switching speed. In [17], the volume (cell area) of the MTJ device is reduced to achieve better writability by sacrificing the retention time of the STT-RAM cache cells. A simple DRAM-style refresh scheme was also proposed to maintain the correctness of the data.

We note that the access patterns of L1 and lower level caches in a multicore microprocessor are different. Based on this insight, we propose STT-RAM designs with different nonvolatility and write characteristics for use in L1 and lower level caches, or even the different parts within the lower level cache so as to maximize power and performance benefits. A low power counter-controlled refresh scheme is applied to maintain the validity of the data. Compared to the existing works on STT-RAM cache designs, our work makes the following contributions:

- We present a detailed discussion on the tradeoff between the MTJ's write performance and its nonvolatility. Using our macromagnetic model, we qualitatively analyze and optimize the device.
- We propose a multi retention level cache hierarchy implemented entirely with STT-RAM that delivers the optimal power saving and performance improvement based on the write access patterns at each level. Our design is easier to fabricate, and has a lower die cost.
- We present a novel refresh scheme that achieves much lower refresh power consumption than DRAM-style periodic refreshing.
- We propose the use of a hybrid lower level STT-RAM design for cache with large capacity that simultaneously offers fast average write latency and low standby power. It has two cache partitions with different write characteristics and nonvolatility. A data migration scheme to enhance the cache response time to write accesses is also proposed. The proposed hybrid cache structure has been evaluated both in lower level cache of 2-level and 3-level cache hierarchy.

The rest of our paper is organized as follows. Section 2 introduces the technical backgrounds of STT-RAM. Section 3 describes the tradeoffs involved in MTJ nonvolatility relaxation. Section 4 proposes our multi-retention STT-RAM L1 and L2 cache structures. Section 5 discusses our experimental results. Related works are summarized in Section 6, followed by our conclusion in Section 7.

2. BACKGROUND

2.1 STT-RAM

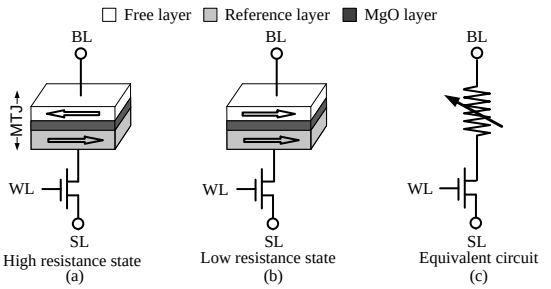

Figure 1: 1T1MTJ STT-RAM. (a) Anti-parallel state, (b) Parallel state, (c) Equivalent circuit.

The data storage device in a STT-RAM cell is the *magnetic tunnel junction* (MTJ), as shown in Figure 1(a) and (b). A MTJ is composed of two ferromagnetic layers that are separated by an oxide barrier layer (e.g., MgO). The magnetization direction of one ferromagnetic layer (the *reference layer*) is fixed while that of the other ferromagnetic layer (the *free layer*) can be changed by passing a current that is polarized by the magnetization of the reference layer. When the magnetization directions of the free layer and the reference layer are parallel (anti-parallel), the MTJ is in its low (high) resistance state.

The most popular STT-RAM cell design is one-transistor-one-MTJ (or 1T1J) structure, where the MTJ is selected by turning on the word-line (WL) that is connected to the gate of the NMOS transistor. The MTJ is usually modeled as a current-dependent resistor in the circuit schematic, as shown in Figure 1(c). When writing "1" (high-resistance state) into the STT-RAM cell, a positive voltage is applied between the source-line (SL) and the bit-line (BL). Conversely, when writing a "0" (low resistance state) into the STT-RAM cell, a negative voltage is applied between the SL and the BL. During a read operation, a sense current is injected to generate the corresponding BL voltage V_{BL}. The resistance state of the MTJ can be read out by comparing the V_{BL} to a reference voltage.

3. DESIGN

3.1 MTJ Write Performance vs. Nonvolatility

The *data retention time*, T_{store}, of a MTJ is determined by the *magnetization stability energy height*, Δ:

$$T_{store} = \frac{1}{f_0} e^{\Delta}. \tag{1}$$

f_0 is the thermal attempt frequency, which is of the order of 1GHz for storage purposes [5]. Δ can be calculated by

$$\Delta = (\frac{K_u V}{k_B T}) = (\frac{M_s H_k V \cos^2(\theta)}{k_B T}), \tag{2}$$

where M_s is the saturation magnetization, and H_k is the effective anisotropy field including magnetocrystalline anisotropy and shape anisotropy. θ is the initial angle between the magnetization vector and the easy axis. T is working temperature. K_B is Boltzmann constant. V is the effective activation volume for the spin-transfer torque writing current. As Eq. (1) and (2) show, the data retention time of a MTJ decreases exponentially when its working temperature, T, rises.

The required *switching current density*, J_C, of a MTJ operating in different working regions can be approximated

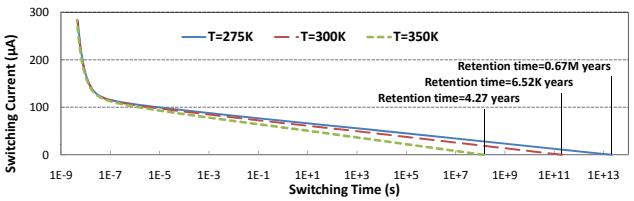

Figure 2: The relationship between the switching current and the switching time of "Base" MTJ design.

as [19, 16]:

$$J_C^{\mathrm{THERM}}(T_{sw}) = J_{C0}(1 - \frac{1}{\Delta}\ln(\frac{T_{sw}}{\tau_0})) \qquad (T_{sw} > 10ns) \tag{3a}$$

$$J_C^{\mathrm{PREC}}(T_{sw}) = J_{C0} + \frac{C\ln(\frac{\pi}{2\theta})}{T_{sw}} \qquad (T_{sw} < 3ns). \tag{3b}$$

$$J_C^{\mathrm{DYN}}(T_{sw}) = \frac{J_C^{\mathrm{THERM}}(T_{sw}) + J_C^{\mathrm{PREC}}(T_{sw})e^{(-A(T_{sw}-T_{\mathrm{PIV}}))}}{1+e^{(-A(T_{sw}-T_{\mathrm{PIV}}))}} \tag{3c}$$

$$(10ns > T_{sw} > 3ns)$$

Here A, C and T_{PIV} are the fitting parameters. T_{sw} is the switching time of MTJ resistance. $J_C = J_C^{\mathrm{THERM}}(T_{sw})$, $J_C^{\mathrm{DYN}}(T_{sw})$ or $J_C^{\mathrm{PREC}}(T_{sw})$ are the required switching currents at T_{sw} in different working regions, respectively. The switching threshold current density J_{C0}, which causes a spin flip in the absence of any external magnetic filed at 0K, is given by:

$$J_{C0} = (\frac{2e}{\hbar})(\frac{\alpha}{\eta})(t_F M_s)(H_k \pm H_{ext} + 2\pi M_s). \tag{4}$$

Here e is the electron charge, α is the damping constant, τ_0 is the relaxation time, t_F is the free layer thickness, $\hbar$ is the reduced Planck's constant, H_{ext} is the external field, and η is the spin transfer efficiency.

As proposed by [17], shrinking the cell surface area of the MTJ can reduce Δ, and consequently decreases the required switching density J_C, as shown in Eq. (3a). However, such a design becomes less efficient in the fast switching region (<3ns) because the coupling between Δ and J_C is less in this region, as shown in Eq. (3b). Moreover, the MTJ device is usually fabricated with the smallest feature size. Further downsizing the MTJ cell surface area will require sub-lithographical patterning technique and is normally not cost-efficient or even doable. Instead, we propose to change M_s, H_k, or t_F to reduce J_c. Such a technique can lower not only Δ but also J_{c0}, offering efficient performance improvement over the entire MTJ working range.

We simulated the switching current versus the switching time of a baseline 45 × 90nm elliptical MTJ over the entire working range, as shown in Figure 2. The simulation is conducted by solving the stochastic magnetization dynamics equation describing spin torque induced magnetization motion at finite temperature [20]. The MTJ parameters are taken from [20], which are close to the measurement results recently reported in [24]. The MTJ data retention time is measured as the MTJ switching time when the switching current is zero. When the working temperature rises from 275K to 350K, the MTJ's data retention time decreased from 6.7×10^6 years to 4.27 years. In the experiments reported in this work, we shall assume that the chip is working at a high temperature of 350K.

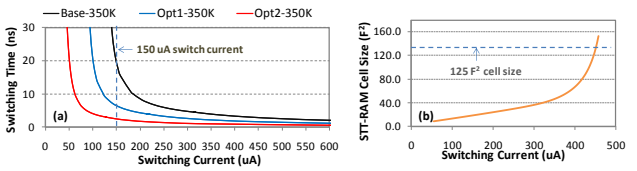

Figure 3: (a) MTJ switching performances for different MTJ designs at 350K. (b) The minimal required STT-RAM cell size at given switching current.

3.2 STT-RAM Cell Design Optimization

To quantitatively study the trade-offs between the write performance and nonvolatility of a MTJ, we simulated the required switching current of three different MTJ designs with the same cell surface shapes. Besides the "Base" MTJ design shown in Figure 2, two other designs ("Opt1" and "Opt2") that are optimized for better switching performance with degraded nonvolatility were studied. The corresponding MTJ switching performances of these three designs at 350K are shown in Figure 3(a). The detailed comparisons of data retention times, the switching currents, the bit write energies, and the corresponding STT-RAM cell sizes of three MTJ designs at the given switching speed of 1ns, 2ns, and 10ns are given in Figure 4.

Significant write power saving is achieved if the MTJ's nonvolatility can be relaxed. For example, when the MTJ data retention time is scaled from 4.27 years ("Base") to $26.5\mu s$ ("Opt2"), the required MTJ switching current decreases from $185.14\mu A$ to $62.5\mu A$ for a 10ns switching time at 350K. At a MTJ switching current of $150\mu A$, the corresponding switching times of all three MTJ designs varied from 20ns to 2.5ns. A switching performance improvement of 8× can be obtained, as shown in Figure 3(a).

The MTJ is normally fabricated with the smallest possible size so as to reduce the switching current. The STT-RAM cell's area is mainly constrained by the NMOS transistor which needs to provide sufficient driving current to the MTJ. Figure 3(b) shows the minimal required NMOS transistor size at a given switching current, and the corresponding 45nm STT-RAM cell area. The PTM model was used in the simulation [3] and the power supply V_{DD} is set to 1.0V. Memory cell area is measured in F^2, where F is the feature size at a certain technology node.

Based on the popular cache and memory modeling software CACTI [2], the typical cell area of SRAM is about $125\mathrm{F}^2$. For a STT-RAM cell with the same area, the maximum current that can be supplied to the MTJ is $448.9\mu A$. A MTJ switching time of less than 1ns can be obtained with the "Opt2" design under such as a switching current while the corresponding switching time for the baseline design is longer than 4.5ns. In this paper, we will not consider designs that are larger than $125F^2$.

Since "Opt1" and "Opt2" requires less switching current than the baseline design for the same write performance, they also consume less write energy. For instance, the write energies of "Base" and "Opt2" designs are $1.85pJ$ and $0.62pJ$, respectively, for a switching time of 10ns. If the switching time is reduced to 1ns, the write energy of "Opt2" design can be further reduced down to $0.32pJ$. The detailed comparisons on the write energies of different designs can be found in Figure 4(c).

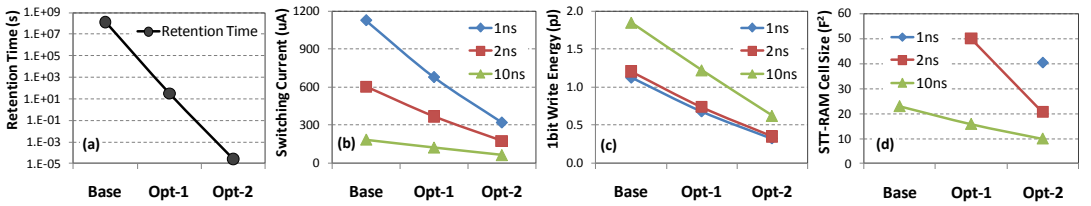

Figure 4: Comparison of different MTJ designs at 350K: (a) the retention time, (b) the switching current, (c) the bit write energy, and (d) STT-RAM cell size.

4. MULTI RETENTION LEVEL STT-RAM CACHE HIERARCHY

Our multi retention level STT-RAM cache hierarchy takes into account the difference in access patterns in L1 and lower level cache. For L1, the overriding concern is access latency. Therefore, we propose the use of our "Opt2" nonvolatility-relaxed STT-RAM cell design as the basis of the L1 cache. In order to prevent data loss introduced by relaxing its non-volatility, we propose a dynamic counter-controlled refresh scheme (M-refresh) to monitor the lifespan of the data, and refresh cells when needed. lower level cache caches are very large compared to L1. As such, a design built with only "Opt2" STT-RAM cells will consume too much refresh energy. Using of the longer retention "Base" or "Opt1" design is more practical. However, to recover the lost performance, we propose a hybrid lower level cache that has a regular and a nonvolatility-relaxed STT-RAM portions. Data will be migrated from one to the other accordingly. The details of our proposed cache hierarchy will be given in the following subsections.

4.1 The Nonvolatility-relaxed STT-RAM L1 Cache Design

As established earlier, using the "Opt2" STT-RAM cell design for L1 caches can significantly improve the write performance and energy. However, its data retention time of $26.5\mu s$ may not be sufficient to retain the longest living data in L1. Therefore, a refresh scheme is needed. In [17], a simple DRAM-style refreshing scheme was used. This scheme refresh all cache blocks in sequence regardless of its data content. Read and write accesses to memory cells that are being refreshed must be stalled. As we shall show in Section 5.2, this simple scheme introduces many unnecessary refreshing operations whose elimination will significantly improve performance and save energy.

4.1.1 Data Retention Monitor

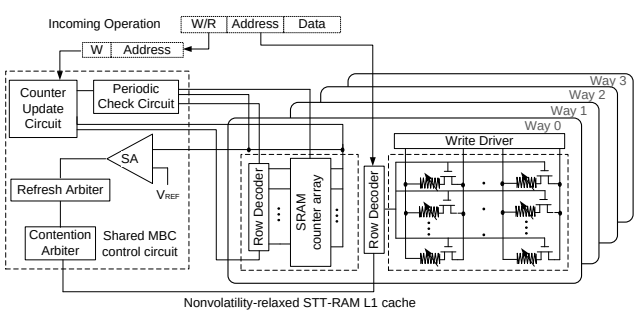

Figure 5: Dynamic counter-controlled refreshing scheme.

To eliminate unnecessary refresh, we use SRAM counters to track the lifespan of cache data blocks. Refresh is performed only on cache blocks that have reached their full lifespan. In our refresh scheme, we assign one counter to each data block in the L1 cache to monitor its data retention status. Only 512 4-bit counters are needed in a 32K bytes L1 cache with 64-byte data blocks. In other words, a mere (4 bits/64 bytes $< 1\%$) overhead.

Reset: On any write access to a data block, its corresponding counter is reset to a lowest level.

Pushing: The STT-RAM cell's retention time is divided into N_{mem} periods, each of which is T_{period} long. A global clock is used to maintain the count-down to T_{period}. At the end of every T_{period}, the level of every counter in the cache is increased by one.

Checking: The data block corresponding to a counter would have reached the maximum retention time when the counter reaches its highest level, and hence needs to be refreshed. Note that the pushing and checking operations can be done simultaneously:

Take, for example, a 32KB L1 cache built using the "Opt2" STT-RAM design . A pushing operation happens once every $26.5\mu s/512/16 = 3.23$ ns, which is more than 6 cycles at a 2GHz clock frequency. A larger cache may mean a higher pushing overhead.

4.1.2 Dynamic Refreshing Schemes to Retain Data

Cache access during refresh: During a refresh operation, the block's data is read out into a buffer, and then saved back to the same cache block. If a read request to the same cache block comes before the refresh finishes, the data is returned from this buffer directly. There is therefore no impact on the read response time of the cache. Should a write request comes, the refresh operation is terminated immediately, and the write request is executed. Again, no penalty is introduced.

Reset threshold N_{th}: We observe that during the lifespan of a cache block, updates happen more frequently within a short period of time after it has been written. Many resets of the cache block data occur far from their data retention time limits, giving us an optimization opportunity. We altered the reset scheme to eliminate counter resets that happen within a short time period after data has been written. We define a threshold level, N_{th}, that is much smaller than N_{mem}. The counter is reset only when its value is higher than N_{th}. The larger N_{th} is, the more resets are eliminated. On the other hand, the refresh interval of the data next written into the same cache block is shortened. However, our experiments in Section 5.2 shall show that such cases happen very rarely and the lifetimes of most data blocks in the L1 cache are much shorter than $26.5\mu s$.

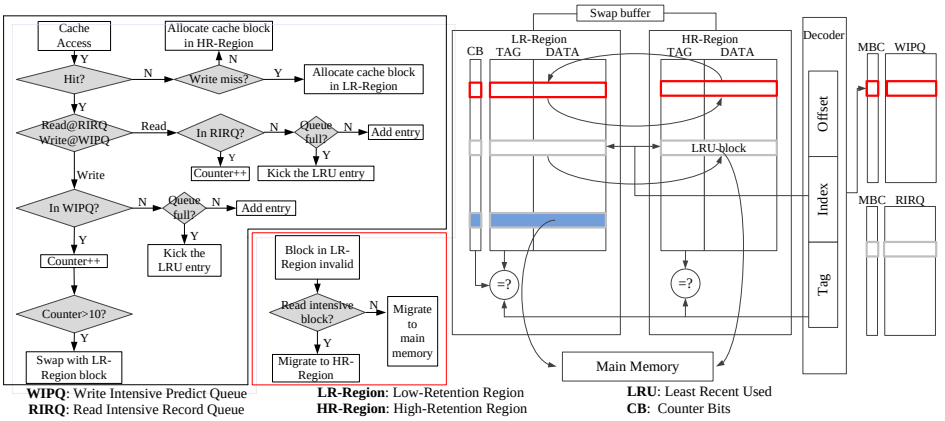

Figure 6: Hybrid lower-level cache migration policy: Flow graph (left). Diagram (right).

4.2 Lower Level Cache with Mixed High and Low Retention STT-RAM Cells

The data retention time requirement in the mainstream STT-RAM development of 4∼10 years was inherited from Flash memory designs. Although such a long data retention time can save significant standby power of on-chip caches, it also entails a long write latency (∼ 10ns), and large write energy [18]. Relaxing the nonvolatility of the STT-RAM cells in the lower level cache will improve write performance as well as save more energy. However, if further reducing retention time to μs scale, *e.g.*, 26.5μs of our "Opt2" cell design, any refresh scheme becomes impractical for the large lower level cache.

The second technique we proposed is a hybrid memory system that has both high and low retention STT-RAM portions to satisfy both the power and performance targets simultaneously. We take a 16 way lower-level cache as a case study as shown in Figure 6, way 0 of a 16-way cache is implemented with a low retention STT-RAM design ("Opt2")) while ways 1 to 15 are implemented with the high retention STT-RAM ("Base" or "Opt1"). Write intensive blocks are primarily allocated from way 0 for a faster write response, while read intensive blocks are maintained in the other ways.

Like our proposed L1 cache, counters are used in way 0 to monitor the blocks' data retention status. However, unlike in L1 where we perform a refresh when a counter expires, here we move the data to the high retention STT-RAM ways.

Figure 6 demonstrates the data migration scheme to move the data between the low and the high retention cache ways based on their write access patterns. A *write intensity prediction queue* (WIPQ) of 16 entries is added to record the write access history of the cache. Every entry has two parts, namely, the data address and a 16-level counter.

During a read miss, the new cache block is loaded to the *high-retention* (HR) region (ways 1-15) following the regular LRU policy. On a write miss, the new cache block is allocated from the *low-retention* (LR) region (way 0), and its corresponding counter is reset to '0'. On a write hit, we search the WIPQ first. If the address of the write hit is already in WIPQ, the corresponding access counter is incremented by one. Note that the block corresponding to this address may be in the HR- or the LR-region of the cache. Otherwise, the hit address will be added in to the queue if any empty entry available. If the queue is full, the LRU entry will be evicted, and replaced by the current hit address. The access counters in the WIPQ are decremented periodically, for example, every 2,000 clock cycles, so that the entries that are in the queue for too long will be evicted. Once an access counter in a WIPQ entry reaches a preset value, $N_{\mathrm{HR}\rightarrow\mathrm{LR}}$, the data stored in the corresponding address will be swapped with a cache block in the LR-region. If the corresponding address is already in the LR-region, no further action is required. A read hit does not cause any changes to the WIPQ.

Likewise, a *read intensity record queue* (RIRQ) with the same structure and number of entries is used to record the read hit history of the LR-region. Whenever there is a read hit to the LR-region, a new entry is added into the RIRQ. Or if a corresponding entry already exist in the RIRQ, the value of the access counter is increased by one. When the counter of a cache block B_i in the LR-region indicates the data is about to become unstable, we check to see if this cache address is read intensive by searching the RIRQ. If B_i is read intensive, it will be moved to HR-region. The cache block being replaced by B_i in the HR-region will be selected using the LRU policy. The evicted cache block will be send to main memory. If B_i is not read intensive, it will be written back to main memory.

In a summary, our proposed scheme uses the WIRQ and RIRQ to dynamically classify cache blocks into three types:

1. *Write intensive:* The addresses of such cache blocks are kept in the WIRQ. They will be moved to the LR-region once their access counters in WIRQ reach $N_{\mathrm{HR}\rightarrow\mathrm{LR}}$;

2. *Read intensive but not write intensive:* The addresses of such cache blocks are found in the RIRQ but not the WIRQ. As they approach to their data retention time limit, they will be moved to the HR-region.

3. *Neither write nor read intensive:* Neither WIRQ nor RIRQ has their addresses. They are kept in HR-region, or evicted from LR-region to main memory directly.

Identifying a *write intensive* cache blocks also appeared in some previous works. In [18], they check if two successive write accesses go to the same cache block. It is highly possible that a cache block may be accessed several times within

Table 1: Simulation Platform

Max issue width: 4 insts	Fetch width: 4 insts	Dispatch width: 4 insts	Write back width: 4 insts
Commit width: 4 insts	Fetch queue size: 32 insts	Reorder buffer: 64 entries	Max branch in pipeline: 24
Load store queue size: 32 entries	Functional units: 2 ALU 2 FPU	Clock cycle period: 0.5 ns	Main memory: 200 cycle latency
Baseline 2-level cache hierarchy			
Local L1 Cache: 32KB 4-way, 64B cache block; Shared L2 Cache: 4MB 16-way, 128B cache block			
3-level cache hierarchy			
Local L1 Cache: 32KB 4-way, 64B cache block ; Local L2 Cache: 256KB 8-way, 64B cache block; Shared L3 cache: 4MB 16-way, 128B cache block			

Table 2: Cache Configuration

	32KB (L1)						**256KB (L2)**	**4MB (L2 or L3)**					
	SRAM	lo1	lo2	lo3	md	hi	md1	SRAM	lo	md1	md2	md3	hi
Cell size (F^2)	125	20.7	27.3	40.3	22	23	22	125	20.7	22	15.9	14.4	23
MTJ sw time (ns)	/	2	1.5	1	5	10	5	/	2	5	10	20	10
Retention Time	/	26.5μs			3.24s	4.27yr	3.24s	/	26.5μs	3.24s			4.27yr
Read Lat (ns)	1.113	0.778	0.843	0.951	0.792	0.802	2.118	4.273	2.065	2.118	1.852	1.779	2.158
Read Lat (cycles)	3	2	2	2	2	2	5	9	5	5	4	4	5
Write Lat (ns)	1.082	2.359	1.912	1.500	5.370	10.378	6.415	3.603	3.373	6.415	11.203	21.144	11.447
Write Lat (cycles)	3	5	4	4	11	21	13	8	7	13	23	43	23
Read Dyn. Eng (nJ)	0.075	0.031	0.035	0.043	0.032	0.083	0.083	0.197	0.081	0.083	0.070	0.067	0.085
Write Dyn. Eng (nJ)	0.059	0.174	0.187	0.198	0.466	0.958	0.932	0.119	0.347	0.932	1.264	2.103	1.916
Leakage pow (mW)	57.7	1.73	1.98	2.41	1.78	1.82	14.24	4107	96.1	104	69.1	61.2	110

very short time, and then becomes inactive. Our scheme is more accurate and effective as it monitors the read and write access histories of a cache block throughout its entire lifespan. The RIRQ ensures that *read intensive* cache blocks migrate from the LR-region to HR-region in a timely manner that, at the same time, also improves energy efficiency and performance.

5. SIMULATION RESULTS & DISCUSSION

5.1 Experimental Setup

We modeled a 2GHz microprocessor with 4 out-of-order cores using MARSSx86 [14]. We assume two-level/three-level cache configuration and a fixed 200-cycle main memory latency. The MESI cache coherency protocol is utilized in L1 caches to ensure consistency, and the lower level cache (L2/L3) uses a write-back policy. The parameters of our simulator can be found in Table 1.

Table 2 shows the performance and energy consumptions of various designs obtained by a modified NVSim [1] simulator. All the "*-hi*", "*-md*", and "*-lo*" configurations use the "Base", "Opt1", and "Opt2" MTJ design, respectively. Note that as shown in Figure 3, they scale differently. SPICE simulations were conducted to characterize the performance and energy overheads of the counter and its control circuity. The reset energy and pushing-checking energy of SRAM counter will be included in the architecture simulation. We simulated a subset of multi-threaded workloads from the PARSEC 2.1 and the SPEC2006 benchmark suites so as to cover a wider spectrum of read/write and cache miss characteristics. We simulated 500 million instructions of each benchmark after their initialization.

We compared the performance (in terms of instruction per cycle, IPC) and the energy consumption of different configurations for both 2- and 3-level hybrid cache hierarchies. We used the conventional all SRAM cache design as the baseline. Our simulation shows that the optimal STT-RAM cache configuration for 2-level cache hierarchy is the combination of (a) a L1 cache of the "L1-lo2" design, and (b) a hybrid L2 cache of using the "L2-lo" in the LR-region and "L2-md2" in the HR-region. The optimal STT cache configuration for 3-level cache hierarchy includes (a) a L1 cache of the "L1-lo2" design, (b) a hybrid L2 cache of using the "L2-lo" in the LR-region and "L2-md1" in the HR-region and (c) a hybrid L3 cache of the "L1-lo2" design in the LR-region and "L3-md2" in the HR-region .

The detailed experimental results will be shown and discussed in Sections 5.2 and 5.3.

5.2 Results for the Proposed L1 Cache Design

To evaluate the impacts of using STT-RAM in L1 cache design, we implemented the L1 cache with the different STT-RAM designs listed in the L1 portion of Table 2 while leaving the SRAM L2 cache unchanged. Due to the smaller STT-RAM cell size, the overall area of L1 cache is significantly reduced. The delay components of interconnect and peripheral circuits also decreased accordingly. Even considering the relatively long sensing latency, the read latency of STT-RAM L1 cache is still similar, or even slightly lower than that of a SRAM L1 cache. However, the write performance of STT-RAM L1 cache is always slower than the SRAM L1 cache for all the design configurations considered. The leakage power consumption of the STT-RAM caches come from the peripheral circuits only, and is very low. The power supply to the memory cells that are not being accessed can be safely cutoff without fear of data loss until the data retention limit is reached.

Figure 7 shows the IPC performance of the simulated L1 cache designs normalized to the baseline all-SRAM cache. On average, implementing the L1 cache using the "Base" (used in "L1-hi") or "Opt1" (used in "L1-md") STT-RAM design incurs more than 32.5% and 42.5% IPC degradation, respectively, due to the long write latency. However, the performance of the L1 caches with the low retention STT-

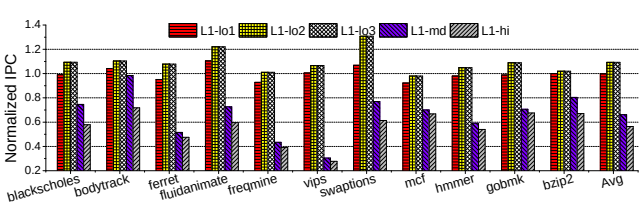

Figure 7: IPC comparison of various L1 cache designs. The IPC's are normalized to all-SRAM baseline.

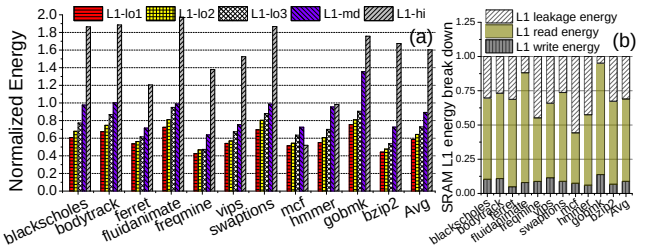

Figure 8: (a) L1 cache overall energy comparison, (b) Break down of L1 SRAM cache energy. The energy consumptions are normalized to SRAM baseline.

RAM design significantly improves compared to that of the SRAM L1 cache: the average normalized IPC's of 'L1-lo1', 'L1-lo2', and 'L1-lo3' are 0.998, 1.092, and 1.092, respectively. The performance improvement of 'L1-lo2' or 'L1-lo3' L1 cache w.r.t the baseline SRAM L1 cache comes from the shorter read latency even though its write latency is still longer, as shown in Table 2. However, L1 read accesses are far more frequent than write access in most benchmarks. In some benchmarks, for example, `swaptions`, the 'L1-lo2' or 'L1-lo3' design achieves a better than 20% improvement in IPC.

The energy consumptions of the different L1 cache designs normalized to the baseline all-SRAM cache are summarized in Figure 8(a). The reported results includes the energy overhead of the refresh scheme and the counters, where applicable. Not surprisingly, all three low retention STT-RAM L1 cache designs achieved significant energy savings compared to the SRAM baseline. The "L1-lo3" design consumes more energy because of its larger memory cell size, and larger peripheral circuit having more leakage and dynamic power, as shown in Table 2. Figure 8 also shows that implementing the L1 cache with the "Base" (used in "L1-hi") or "Opt1" (used in "L1-md") STT-RAM is much less energy-efficient because (1) the MTJ switching time is longer, resulting in a higher write dynamic energy, and (2) a longer operation time due to the low IPC.

Figure 8(b) shows the breakdown of L1 SRAM energy. The leakage energy occupies more than 30% of overall energy. In addition, STT-RAM has lower per bit read energy. Read frequency is around 4.8 times of write frequency on average, resulting in lower dynamic energy of STT-RAM. That's why "L1-lo1", "L1-lo2" and "L1-lo3" STT-RAM can save up to 30% to 40% overall energy compared to SRAM design.

Figure 9(a) compares the refresh energy consumptions of the 'L1-lo2' L1 cache under different refresh schemes. In each group, the three bars from left to right represent the refresh energy consumptions of DRAM style refresh scheme,

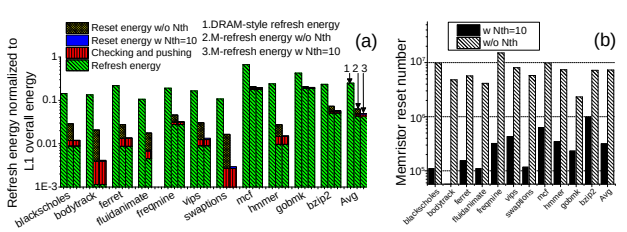

Figure 9: Refresh energy comparison of the different refresh schemes.

refresh scheme without reset threshold N_{th}, and with $N_{\text{th}} = 10$, respectively. The refresh energy consumptions are normalized to the overall L1 energy consumptions when implementing the refresh scheme with $N_{\text{th}} = 10$. Note that the y-axis is in logarithmic scale.

The energy consumption of the simple DRAM-style refresh scheme accounts for more than 20% of the overall L1 cache energy consumption on average. In some extreme cases of low write access frequency, for example, `mcf`, this ratio is as high as 80% because of the low dynamic cache energy consumption.

The total energy consumption of our proposed refresh scheme consists of the checking and pushing, the reset, and the memory cell refresh. By accurately monitoring the lifespan of the cache line data, our refresh scheme significantly reduced the refresh energy in all the benchmarks. As we discussed in Section 4.1.2, the introduction of the reset threshold N_{th} can further reduce the refresh energy consumption by reducing the number of counter resets. This is confirmed in Figure 9(a) and (b). The number of counter reset operations are reduced by more than 20× on average after setting a reset threshold N_{th} of 10, resulting in more than 95% of the reset energy being saved. The energy consumption for the refresh scheme is very marginal, accounting for only 4.35% of the overall L1 cache energy consumption.

5.3 Evaluating the Hybrid Cache Design in 2-level Cache Hierarchies

First, we evaluate the proposed hybrid cache design within L2 cache in 2-level cache hierarchies. In comparing the different L2 cache designs, we fixed the L1 cache to the 'L1-lo2' design. In our proposed hybrid L2 cache, way 0 assumes the 'L2-lo' design for the best read latency and the smallest leakage power among all three low retention STT-RAM designs. Ways 1 to 15 are implemented using the 'L2-md1', 'L2-md2', or 'L2-md3' (all "Opt1" MTJ designs) because a 3.24s retention time is good enough for most applications, and they have the minimal refresh overhead. The three resultant configurations are labeled as 'L2-Hyb1', 'L2-Hyb2', and 'L2-Hyb3', respectively. We compare our hybrid L2 cache with the single retention level STT-RAM design of [17] and the *read/write aware high performance architecture* (RWHCA) of [22], and label them as 'L2-SMNGS' and 'L2-RWHCA', respectively. For 'L2-SMNGS', we assumed that the L2 cache uses 'L2-md1' because its cell area of $22F^2$ is compatible to the $19F^2$ one reported in [17]. Instead of using 'L2-hi' in ways 1 to 15, 'L2-RWHCA' uses 'L2-md2' as it has an access latency that is similar to the one assumed in [22] but a much lower energy consumption. Except for Hybrid, all other L2 STT-RAM schemes use the simple DRAM refresh when refresh is needed. To be consistent with the previous section, we normalize the simulation results to the all-SRAM design.

Figure 10(a) compares the normalized IPC results of the different L2 cache designs. As expected, the regular STT-RAM L2 cache with 'L2-hi' design shows the worst performance among all the configurations, especially for benchmarks with high L1 miss rates, and L2 write frequencies (such as `mcf` and `swaptions`). Using relaxed retention STT-RAM design 'L2-SMNGS' improves performance but on the average it still suffers 6% degradation compared to the all-SRAM baseline due to its longer write latency. Among the three hybrid schemes we proposed, 'L2-Hyb1' is comparable

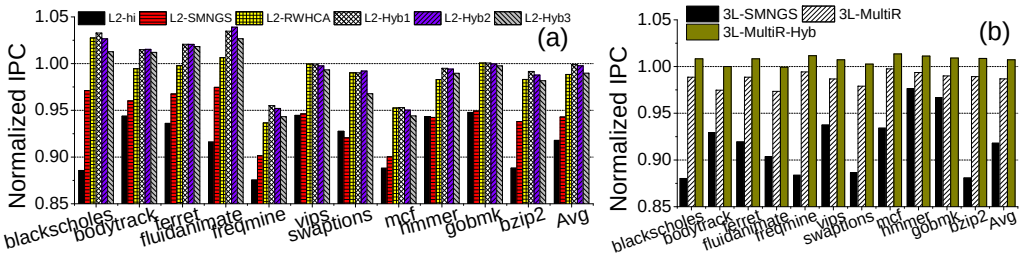

Figure 10: Performance comparison of different (a) 2-L cache designs (b) 3-L cache designs. The IPC's are normalized to all-SRAM baseline.

in performance (99.8% on average) to the all-SRAM cache design. As we prolong the MTJ switching time by reducing STT-RAM cell size in 'L2-Hyb2' and 'L2-Hyb3', IPC performance suffers. However, all our hybrid L2 caches outperform both 'L2-SMNGS' and 'L2-RWHCA' due to their lower read latencies.

Since the savings in leakage energy by using STT-RAM designs in the L2 cache is well established, we compared the *dynamic* energy consumptions of different L2 cache designs. The energy overheads of the data refresh in LR-region, and the data migration between LR- and HR-regions in our hybrid L2 caches are included in the dynamic energy. Due to the lower write energy in the LR-region, 'L2-Hyb1' has the lowest dynamic energy consumption, as shown in Figure 11 (left). As the STT-RAM cell size is reduced, the write latency and write energy consumption increased. Thus, the corresponding dynamic energy of 'L2-Hyb2' and 'L2-Hyb3' grow rapidly. Figure 11 (right) shows the leakage energy comparison. Compared to 'L2-RWHCA' which is a combination of SRAM/STT-RAM [22], all the other configurations have much lower leakage energy consumptions. 'L2-hi', 'L2-SMNGS', and 'L2-Hyb1' have similar leakage energies because their memory array sizes are quite close to each other. However, 'L2-Hyb2' and 'L2-Hyb3' benefit from their much smaller memory cell size.

The overall cache energy consumptions of all the simulated cache configurations are summarized in Figure 12(a). On the average, 'L2-Hyb2' and 'L2-Hyb3' consumes about 70% of the energy of 'L2-SMNGS', and 26.2% of 'L2-RWHCA'. In summary, our proposed hybrid scheme outperforms the previous techniques in [17] and [22] both in terms of performance, and (by an even bigger margin) total energy.

5.4 Deployment in 3-level Cache Hierarchies

We also evaluate four designs for a 3-level cache hierarchy whose parameters were given in Table 2. The designs evaluated in this work include: (1) the all SRAM cache hierarchy, (2) '3L-SMNGS', (3) a multi retention 3-level STT-RAM cache hierarchy ('3L-MultiR') with "L1-lo2" , "L2-md2" and "L3-hi", and (4) a multi retention 3 level STT-RAM cache hierarchy ('3L-MultiR-Hyb') with "L1-lo2", as well as proposed hybrid cache design as its lower level cache (both L2 and L3). In '3L-MultiR-Hyb', 'Hyb1' is used in L2 cache for the performance purpose, while 'Hyb2' is used in L3 cache to minimize the leakage energy. Just like 'L2-SMNGS', '3L-SMNGS' [17] uses the "md1" STT-RAM design in all the three level of caches. In [17], the IPC performance degradations for using the single retention STT-RAM ('md1') were from 1% to 9% when compared to an all-SRAM design.

Our simulation result of '3L-SMNGS' (8% performance degradation on average) matches this well. Comparatively, the average IPC performance degradation of '3L-MultiR' is only 1.4% on average, as shown in Figure 10(b) . The performance gain of '3L-MultiR' over '3L-SMNGS' comes mainly from "L1-lo2". '3L-MultiR-Hyb' has the best performance which is 8.8% and 2.1% better than '3L-SMNGS' and '3L-MultiR' on average. Most of the write access in L2 and L3 cache of '3L-MultiR-Hyb' are allocated into the fast region, boosting up the system performance. Under the joint effort of "L1-lo2" and hybrid lower level cache, the '3L-MultiR-Hyb' can even achieve a slightly higher IPC can all-SRAM design.

Normalized against an all-SRAM 3-level cache design, the overall energy comparison of 3-level cache hierarchy is shown in Figure 12(b). All three combinations with STT-RAM save much more energy when compared to all-SRAM design. '3L-MultiR' saves slightly more overall energy compared to '3L-SMNGS' because the 'Lo" STT-RAM cell design has a lower per bit access dynamic energy than the "md" design. In '3L-MultiR-Hyb', shared L3 cache which embedded "md2" is much larger than local L2 cache which uses "md1" . Thereby, the leakage of L3 dominates the overall energy consumption. The leakage power ratio between "md2" and "hi" is 69.1/110 (see Table 2). That's why the overall energy of '3L-MultiR-Hyb' is only 60% of '3L-MultiR' whose L3 is "hi".

5.5 Die Cost Comparison

We shall now compare the die cost of using the different cache designs. If we assume that the wafer cost, wafer yield, and defect density are constants in a specific foundry at a given technology node, the number of dies per wafer, N_{die}, and the die yield, Y_{die}, can be modeled by [7]:

$$N_{\text{die}} = \frac{\pi \times (\phi_{\text{wafer}}/2)^2}{A_{\text{die}}} - \frac{\pi \times \phi_{\text{wafer}}}{\sqrt{2 \times A_{\text{die}}}} \tag{5}$$

$$Y_{\text{die}} = Y_{\text{wafer}} \times \frac{1 - e^{-2A_{\text{die}}D_0}}{2A_{\text{die}}D_0}. \tag{6}$$

where ϕ_{wafer} is the diameter of the wafer, Y_{wafer} is the wafer yield, and D_0 is the defect density of the wafer. A_{die} is the die area, determined by the specific cache designs.

Our microprocessor baseline is the Intel Core2 Quad Processor Q8200 fabricated at the 45nm technology node [9]. 50% of the die ($164mm^2$) is occupied by the SRAM caches. The wafer yield is assumed to be 99%. Figure 13 shows the trend of processor die cost C_{die} after replacing the caches with STT-RAM ones by varying Y_{wafer} and the wafer cost C_{wafer}. Here, C_{wafer} and C_{die} are normalized to the baseline with all-SRAM cache design.

For a given curve, the area beneath it represents the allowable C_{wafer} and Y_{wafer} combinations at the given C_{die}.

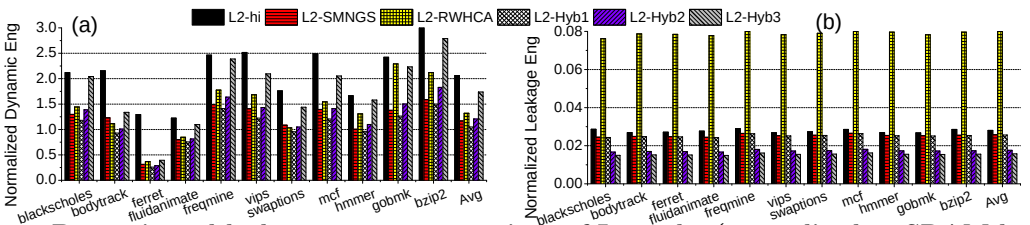

Figure 11: Dynamic and leakage energy comparison of L2 cache (normalized to SRAM baseline).

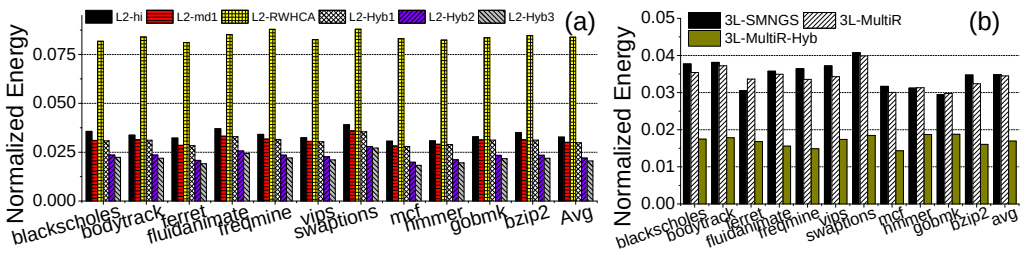

Figure 12: Overall cache energy consumption comparison (a) 2-L cache designs (b) 3-L cache designs (Normalized to the all-SRAM design).

For instance, 'L1-lo2, L2-Hyb2, 80%' is for a multiprocessor with 'L1-lo2' and 'L2-Hyb2' cache hierarchy. The curve constraints the requirement of C_{wafer} and Y_{wafer} when C_{die} of such as processor is 80% of our baseline microprocessor. Or, let's assume that after introducing STT-RAM technology, the Y_{wafer} reduces to 90%. The C_{wafer} has to be less than 1.11, 1.48, or 1.85 if we expect C_{die}(L1-lo2, L2-SMNGS) to be less than 60%, 80%, or 100% of the baseline's die cost, respectively. Our proposed hybrid L2 design can slightly relax C_{wafer}(L1-lo2, L2-Hyb2) to 1.17, 1.53, or 1.95, respectively. Utilizing the proposed multi retention level STT-RAM design to 3-level cache hierarchies can further reduce die cost.

Note that the additional direct fabrication cost introduced by STT-RAM technology is a mere 5% more than that for the standard CMOS process (C_{wafer}=1.05) [13]. In such a situation, we can easily obtain a die cost less than 60% of the baseline, as long as Y_{wafer} is greater than 80%.

6. RELATED WORK

STT-RAM has many attractive features such as the nanosecond access time, CMOS process compatibility and nonvolatility. The unique programming mechanism of STT-RAM – changing the MTJ resistance by passing a spin-polarized current [8] – ensures good scalability down to the 22nm technology node with a programming speed that is below 10ns [21]. Early this year, Zhao, et. al. reported a sub-nanosecond switching at the 45nm technology node for the in-plane MTJ devices [24].

Dong, et. al. gave a comparison between the SRAM cache

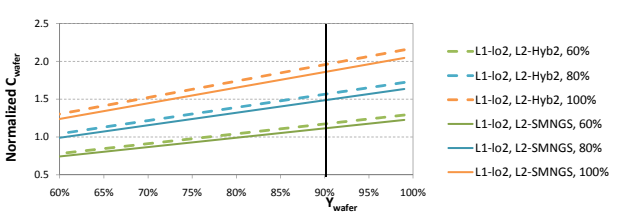

Figure 13: Cost comparison of all-SRAM and all STT-RAM cache designs.

and STT-RAM cache in a single-core microprocessor [6]. Desikan, et. al. conducted an architectural evaluation of replacing on-chip DRAM with STT-RAM [4]. Sun, et. al. extended the application of STT-RAM cache to Chip Multiprocessor (CMP) [18], and studied the impact of the costly write operation in STT-RAM on power and performance.

Many proposals have been made to address the slow write speed and high write energy of STT-RAM. Zhou, et. al. proposed an early write termination scheme to eliminate the unnecessary writes to STT-RAM cells and save write energy [25]. A dual write speed scheme was used to improve the average access time of STT-RAM cache that distinguishes between the fast and slow cache portions [23]. A SRAM/STT-RAM hybrid cache hierarchy and some enhancements, such as write buffering and data migration were also proposed in [18, 22]. The SRAM and STT-RAM cache ways are fabricated on the different layers in the proposed 3D integration. The hardware and communication overheads are relatively high. None of these works considered using STT-RAM in L1 due to its long write latency.

Early this year, Smullen, et. al. proposed trading off the nonvolatility of STT-RAM for write performance and power improvement [17]. The corresponding DRAM-style refresh scheme to assure the data validity is not scalable for a large cache capacity. However, the single retention level cache design is lack of optimization space to maximize the benefits of STT-RAM writability and nonvolatility tradoffs. Also, the MTJ optimization technique they proposed, namely shrinking the cell surface area of the MTJ, is not efficient in the fast switching region (<10ns), as discussed in Section 3.

The macro-magnetic model used in our work was verified by a leading magnetic recording company and calibrated with the latest in-plane MTJ measurement results [24]. However, we note that our model was not able to reproduce the MTJ parameters given in [17], which are overly optimistic in the fast-switching region (< 3ns) in terms of write energy and performance, as well as data retention time.

7. CONCLUSION

In this paper, we proposed a multi retention level STT-

RAM cache hierarchy that trades off the STT-RAM cell's nonvolatility for energy saving and performance improvement. A low retention L1 cache with a counter-controlled refresh scheme, and a hybrid structure for lower level cache with both low- and high-retention portions were presented. Compared to the classic SRAM or a SRAM/STT-RAM hybrid cache hierarchy, we propose one that uses only STT-RAM. This can save significant die cost and energy consumption. Moreover, compared to the previous STT-RAM relaxed retention design that only has a single retention level, our design utilizes multiple retention levels, resulting in an architecture that is optimized for the data access patterns of the different cache levels. Our experimental results show that our proposed multi retention level STT-RAM hierarchy achieves on average a 73.8% energy reduction over the SRAM/STT-RAM mixed design, while maintaining a nearly identical IPC performance. Compared with the previous single-level relaxed retention STT-RAM design, we obtained a 5.5% performance improvement, and a 30% overall energy reduction by having multiple retention levels in 2-level hierarchy. The multi retention STT-RAM cache with proposed hybrid STT-RAM lower level cache achieves on average of 6.2% performance improvement and 40% energy saving compared to the previous single-level relaxed retention STT-RAM design for a 3-level cache hierarchy. Compared to traditional SRAM L1 cache, the L1 cache with a ultra low retention STT-RAM augmented by the proposed refresh scheme can achieve a 9.2% performance improvement, and a 30% energy saving.

With technology scaling, and the increasing complexity of fabrication, we believe that our proposed cache hierarchy will become even more attractive because of its performance, low energy consumption, and CMOS compatibility.

8. ACKNOWLEDGEMENT

The authors would like to thank Dr. Xiaobin Wang from Seagate Technology for the valuable discussion and guidance on this research. This work was supported in part by NSF grant of CNS 1116684.

9. REFERENCES

[1] http://www.rioshering.com/nvsimwiki/index.php.

[2] CACTI. http://www.hpl.hp.com/research/cacti/.

[3] Y. Cao and et al. New paradigm of predictive mosfet and interconnect modeling for early circuit design. In *IEEE Custom Integrated Ckt. Conf.*, pages 201–204, 2000. http://www-device.eecs.berkeley.edu/ptm.

[4] R. Desikan and et. al. On-chip MRAM as a high-bandwidth low-latency replacement for DRAM physical memories. http://www.cs.utexas.edu/ftp/pub/techreports/tr02-47.pdf.

[5] Z. Diao and et al. Spin-transfer torque switching in magnetic tunnel junctions and spin-transfer torque random access memory. *J. of Physics: Condensed Matter*, 19:165209, 2007.

[6] X. Dong and et al. Circuit and microarchitecture evaluation of 3D stacking magnetic RAM (MRAM) as a universal memory replacement. In *Proc. of DAC*, pages 554–559, 2008.

[7] X. Dong and Y. Xie. System-Level Cost Analysis and Design Exploration for Three-Dimensional Integrated Circuits (3D ICs). In *Asia and South Pacific Design Automation Conference*, pages 234–239, 2009.

[8] M. Hosomi and et al. A novel nonvolatile memory with spin torque transfer magnetization switching: Spin-RAM. In *IEEE IEDM*, pages 459–462, 2005.

[9] Intel Core2 Quad Processor Q8200. http://ark.intel.com/Product.aspx?id=36547.

[10] T. Kawahara and et. al. 2 Mb SPRAM (SPin-transfer torque RAM) with bit-by-bit bi-directional current write and parallelizing-direction current read. *IEEE Jour. of Solid-State Ckts.*, 43(1):109–120, 2008.

[11] C. Kim, J. Kim, S. Mukhopadhyay, and K. Roy. A forward body-biased low-leakage sram cache: device, circuit and architecture considerations. *IEEE Trans. on VLSI Systems*, 13(3):349–357, 2005.

[12] S. Kirolos and Y. Massoud. Adaptive sram design for dynamic voltage scaling vlsi systems. In *Midwest Symp. on Circuits and Systems*, pages 1297–1300, 2007.

[13] L. Y. Loh. Mechanism and Assessmenet of Spin Transfer Torque (STT) Based Memory. Master's thesis, Massachusetts Institute of Technology, 2007.

[14] Marss86. http://www.marss86.org/.

[15] P. Nair, S. Eratne, and E. John. A quasi-power-gated low-leakage stable sram cell. In *Midwest Symp. on Circuits and Systems*, pages 761–764.

[16] A. Raychowdhury and et al. Design space and scalability exploration of 1t-1stt mtj memory arrays in the presence of variability and disturbances. In *IEEE Int. Electron Devices Meeting*, pages 1–4, Dec. 2009.

[17] C. Smullen and et al. Relaxing Non-Volatility for Fast and Energy-Efficient STT-RAM Caches. *Proc. of 2011 HPCA*, 2011.

[18] G. Sun, X. Dong, Y. Xie, J. Li, and Y. Chen. A novel architecture of the 3D stacked MRAM L2 cache for CMPs. In *Proc. of 15th HPCA*, pages 239–249, 2009.

[19] J. Z. Sun. Spin-current interaction with a monodomain magnetic body: A model study. *Phys. Rev. B*, 62:570–578, 2000.

[20] X. Wang and et al. Relationship between symmetry and scaling of spin torque thermal switching barrier. *IEEE Trans. on Magnetics*, 44:2479–2482, 2008.

[21] X. Wang and et al. Spin torque random access memory down to 22nm technology. *IEEE Trans. on Magnetics*, 44(11):2479–2482, 2008.

[22] X. Wu, J. Li, L. Zhang, E. Speight, and Y. Xie. Power and performance of read-write aware hybrid caches with non-volatile memories. In *Proc. of DATE*, pages 737–742, 2009.

[23] W. Xu and et al. Design of Last-Level On-Chip Cache Using Spin-Torque Transfer RAM (STT RAM). In *IEEE Trans. on VLSI System*, pages 483–493, 2011.

[24] H. Zhao and et al. Low writing energy and sub nanosecond spin torque transfer switching of in-plane magnetic tunnel junction for spin torque transfer RAM. *J. of App. Phys.*, 109:07C720, 2011.

[25] P. Zhou, B. Zhao, J. Yang, and Y. Zhang. Energy reduction for STT-RAM using early write termination. In *Proc. of the 2009 ICCAD*, pages 264–268, 2009.