A high-performance and power-efficient design of Processor cache hierarchy using STT-RAM technology with Multi-Retention Time Intervals

ABSTRACT

Caching techniques have been an efficient mechanism for mitigating the effects of the processor-memory speed gap. Traditional multi-level SRAM-based cache hierarchies, especially in the context of chip multiprocessors (CMPs), present many challenges in area requirements, power consumption, and design complexity. New advancements in technology enable caches to be built from other technologies, such as Spin-transfer torque random access memory (STT-RAM) and Phase-change RAM (PRAM), in both 2D chips or 3D stacked chips. 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 (e.g., L2 or L3 cache) has been proposed as it minimizes cache leakage power with technology scaling down. Furthermore, the cell area of STT-RAM is only 1/4 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 research proposal, 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 tunnelling junction (MTJ) designs. For the fast STT-RAM bits with reduced data retention time, a novel dynamic refresh scheme is proposed to maintain the data validity.

For lower level caches with relative large cache capacity, we propose a run-time strategy for managing writes between portions of the cache with different retention characteristics so as to maximize the performance and power benefits. A novel contribution, low-overhead and fully-hardware technique is utilized to detect write-intensive data blocks of working set during system operation. This proposal is one year research project.

Keywords – Spin torque transfer RAM, Retention time, Hybrid cache architecture, Power consumption, Performance.

1. Introduction

Some reasons makes spin-transfer torque RAM (STT-RAM) a potential memory technology alternatives to SRAM-based last-level cache memory structures. In a normal processor with a three-level cache hierarchy while technology continuously scale, resulting in the transistor's leakage current to increase exponentially, the actual requirement for large to maximize last-level caches has increased in an effort to improve the system performance along with energy. Being non-volatile, STT RAM consumes near zero leakage power. in comparison to SRAM equivalent, is about four times denser. It is compatibile with the CMOS fabrication process, and simply incurs three additional mask layers to be able to embed MTJ devices on logic dies.

Nevertheless, the particular latency and energy overhead belonging to the write operations are definitely the critical downsides associated with this technology for delivering aggressive or even improved perfromance in comparison to the SRAM-based cache hierarchy. As a way to masking the negative impacts of high write latency and write energy, recently researchers have proposed various mitigation techniques at the architectural level. In particular Zhou, et al developed an early write termination technique which help reduce the latency and energy overhead throughout the avoidance of unnecessary writes to STT-RAM cells.

1.2. Significance of Research

The data retention time indicates nonvolatility of a memory cell. Relaxing this nonvolatility tends to make the memory cells easier to be programmed, and results in a lower write current or faster switching speed. According to the fact that the required data retention time across the particular layers of the memory hierarchy is different, the data stored in last level cache(LLC) does not need many years of retention time[1], so using STTRAM in LLC provides a chance to relaxed the condition on high retention time. reported time interval that data is resident in last level cache is on the order of microseconds so Another methodology is reduceing the STT-RAM write energy by trading retention time for improved performance and reduce write energy, presented in [2]. Smullen et al. has used The area of the MTJ to lower the retention time by decreasing the thermal stability[2]. Since reduction in MTJ area does not affect critical write current density required to switch the magnetization of the free layer, so the MTJ's write time will not reduced. Consequently To reduce both the MTJ write time and write energy, we propose to reduce free layer thickness.

we propose a lower level STT-RAM design for cache with large capacity that has two partitions with different write characteristics and nonvolatility. we have decomposed each set of the shared last-level cache into a small number of lines of low retention time STTRAM cells with high speed and low energy write operation and a large number of STT-RAM lines which have higher retention time. To address the relatively latency and energy overhead associated with the write operations in high retention time STT-RAM arrays, considering the non-uniformity of write accesses into different cache sets and even within same cache set, we propose a data migration mechanism to enhance utilization of low retention time STTRAM array and therefore dynamically redirect most of the write access from higher retention STTRAM arrays in the same set or another sets into high speed under-utilized STTRAM array and consequently improve the cache response time to write accesses.

In this framework, the research deals with two design issues. According to the required refresh time of the Last Level Cache blocks to achieve significant reduction of write latency of STT-RAM cells with lower retention time, we have to decide an appropriate retention time for multiretention STT-RAM arrays against the overhead for data refresh. Besides, another challenge in determining a suitable time duration for the proposed data migration mechanism.

2. Background

2.1. STT-RAM Operation and Peripherals

An STT-RAM cell consists of a Magnetic Tunnel Junction (MTJ) as the storage element combined with an NMOS transistor as the access controller. MTJ contains two ferromagnetic layers that are separated by an oxide barrier layer (e.g., MgO). One ferromagnetic layer has fixed magnetization direction and it is called the reference layer, while the other layer has a free magnetization direction that can be changed by passing a write current and it is called the free layer. The relative magnetization direction of two ferromagnetic layers determines the resistance of MTJ. when the magnetic field of the free layer and reference layer are parallel (anti parallel), the MTJ resistance is low (high), representing a logical 0 (logical 1).

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. As shown in Fig. , When a write operation is performed, a positive voltage difference is established(applied) between the source-line (SL) and bit-line (BL) for writing a “0” or a negative voltage difference is established for writing a “1”. The current amplitude required to ensure a successful status reversal is related to the material of the tunnel barrier layer, the writing pulse duration, and the MTJ geometry. During a read operation, a sense current is injected to generate the corresponding BL voltage. The resistance state of the MTJ can be read out by comparing the BL voltage to a reference voltage.

2.2. STT-RAM Cell Stability and Retention

The nonvolatility of an MTJ is quantitatively measured by the data retention time, which is the maximum time duration for which data can be stored in the MTJ(OR: is a characterization of the expected time until a random bit-flip occurs) and is determined by the thermal stability (Δ) of the MTJ. The thermal stability is approximated by Equation (1), which depends on the geometry and magnetic parameters of the MTJ free layer. A and t_F are the planar area and thickness of the free layer, respectively, while k_B is Boltzmann's constant and T is the operating temperature.

$$\Delta \approx \frac{A \,.\, t_F \,.\, H_k \,.\, M_s}{2\, k_B \,.\, T} \qquad (1)$$

The switching threshold current density J_{C0}, which causes a spin flip in the absence of any external magnetic field at zero temperature (at 0 K), is a figure of merit for MTJ designs and primarily depends on the vertical structure and magnetic properties of the MTJ. is given by Equation (2):

$$J_{c0} = \frac{2e}{h} \,.\, \frac{\alpha}{\eta} \,.\, t_F \,.\, M_s \,.\, (H_k + H_{ext} + 2\pi \,.\, M_s \,.\, X) \qquad (2)$$

3. STT-RAM Cell Design Optimization

When the MTJ (magnetic tunneling junction) switching time is reduced down to 10 ns and below, the switching current required has to increase exponentially [3]. Since the clock frequency of modern computing systems are in the multigiga Hz range, the write speed of STT-RAM cells, which is in the order of tens of nanoseconds, may significantly degrade the overall performance of the system.

This work is based on a key observation: STT-RAMs are most likely to be deployed as on-chip processor caches, and the residency of data in the on-chip cache is much shorter than the default data retention time assumed in STT-RAM designs, namely, 4 10 years [7]. Therefore, it is possible to shorten the data retention time of the MTJ so as to improve its write performance, i.e., switching time and the required switching current. Specifically, relaxing the thermal stability of an MTJ can improve its switching performance while reducing its data retention time [3].

As J_{c0} is independent of MTJ area (Equation 2), the MTJ's write time will remain unaffected by a reduction in its area [2]. So we simulated the required switching current of some different MTJ designs by decreasing the thickness of the free layer and lowering the saturation magnetization with the same cell surface shapes. Besides the nonvolatile MTJ design three other designs that are optimized for better switching performance with degraded nonvolatility were studied. We are able to get 5 volatile MTJs with reduced thermal barriers, which corresponds to retention times of 10sec, 1sec, 500ms, 100ms and 10ms under 125°C, respectively. Raw experimental data is obtained from [5]. The results are plotted in Figure 1 that shows the write energy and write latency of the MTJ(STT-RAM cell) as a function of data retention times. Decreasing the data retention time reduces the write energy and latency per access. The detailed comparisons of data retention times and the corresponding STT-RAM cell write energies and switching currents of six MTJ designs are given in Figure 1.

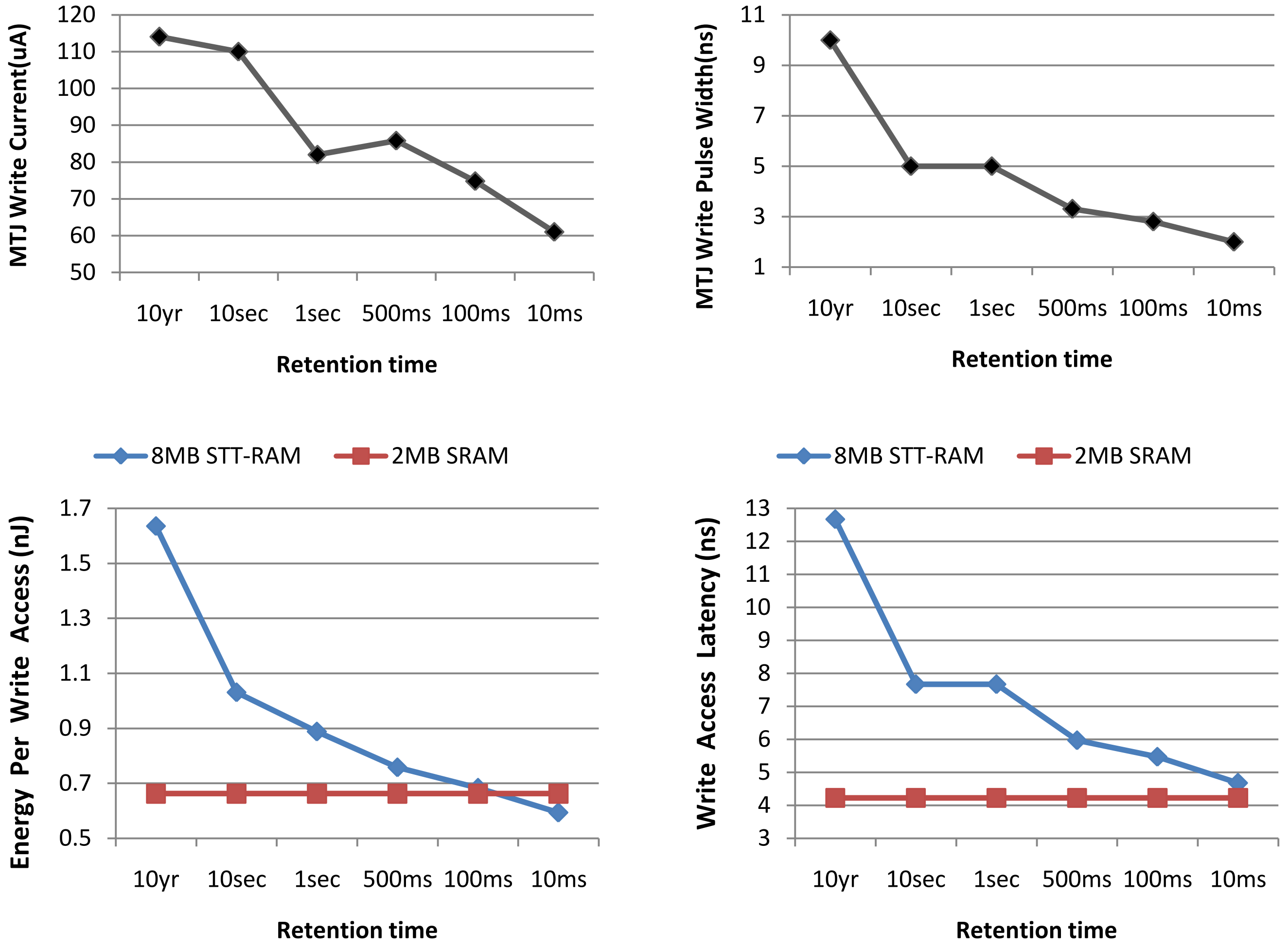


Figure 1: write energy and write latency of the STT-RAM cell as a function of data retention times

Cache Config.	Op.	Retention Time	Dynamic Energy	Pulse Duration	Total latency (+Tag)
8MB STT-RAM	R	10 year	0.530 nJ/access	---	4.283 ns
	W	10 year	1.635 nJ/access	10ns	12.67 ns
8MB MulR(1)	R	---	0.0049ns	---	4.283 ns
	LR-W	10ms	0.593 nJ/access	2ns	4.672 ns
	HR-W	1year	1.635 nJ/access	10ns	12.67 ns
8MB MulR(2)	LR-W	100ms	0.682 nJ/access	2.8ns	5.472 ns
	HR-W	1sec	0.887 nJ/access	5ns	7.672 ns
8MB MulR(3)	LR-W	10ms	0.593 nJ/access	2ns	4.672 ns
	HR-W	1sec	0.887 nJ/access	5ns	7.672 ns
8MB MulR(4)	LR-W	10ms	0.593 nJ/access	2 ns	4.672 ns
	HR-W	100ms	0.682 nJ/access	2.8 ns	5.472 ns

Application characterization gives the basis for evaluating the impact of retention time on the overall system performance. In order to do this characterization, the first step is to investigate the duration for which the cache block should retain the data. A cache block is only refreshed when the block is written. Thus , we record intervals between two successive writes (refreshes) to the same L3 cache block. Figure 2 shows the CDF of L3 baseline cache different inter-write time intervals for applications along with the averages across the entire PARSEC 2.1[10] and SPEC 2006[11] suite applications. We observe from the figures that, on average, approximately more than 95% of the inter-write time intervals are smaller than 10ms.

This distribution also gives us the basis on which we can choose the optimal retention time. Reducing the retention time too much will make the cache highly volatile leading to degraded performance, while increasing the retention time would negatively affect the write latency.

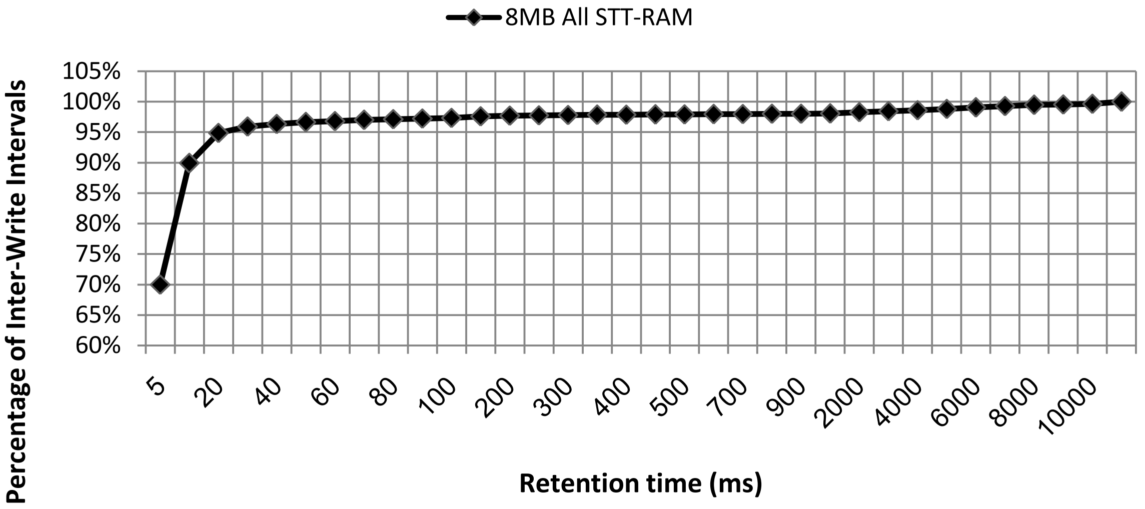


Figure 2: intervals between two successive writes (refreshes) in L3 cache blocks

The 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. Although complete replacement of SRAM cache with the same area STT-RAM cache increases the cache capacity, write operations with no wear-leveling or read priority techniques such as early write termination [6] may impose important challenges especially for write-intensive programs. To alleviate these problems, we partition every cache set into two arrays having different retention characteristics (different write characteristics and nonvolatility). one with large number of high retention STT-RAM lines and the other with few low retention STT-RAM lines, based on the write access patterns at last level cache. In short, we hope to have a cache partition with relative low latency and dynamic energy for write operations.

There exist some non-uniformity for writes within STT-RAM lines of the same set. Figure 3 profiles the write distribution over lines of different sets. This profiling shows how much of the write traffic to each set is destined to the ways with high write request(write-stressed blocks, black bars), 2way of STT-RAM with medium write request (gray bars), and 5 other STT-RAM lines with low write request (white bars).

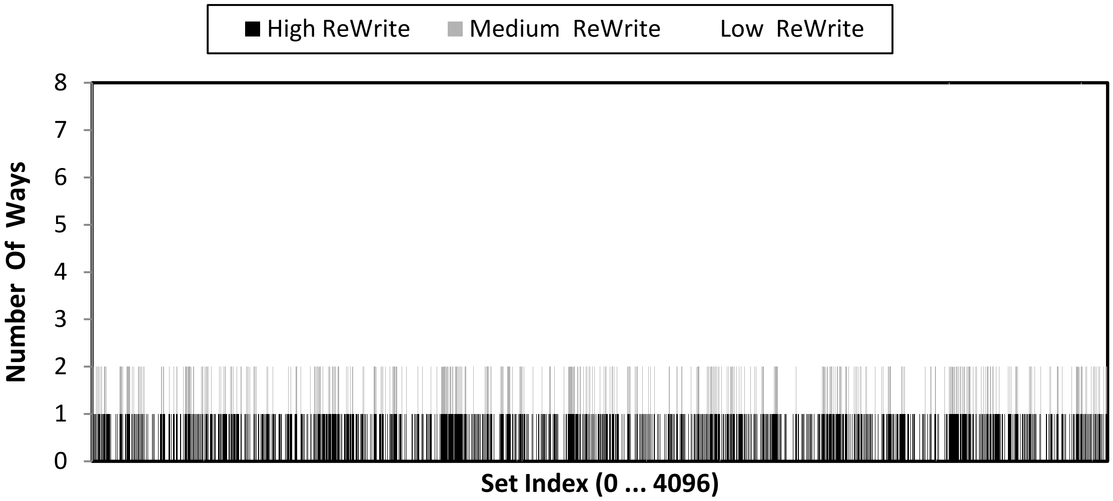
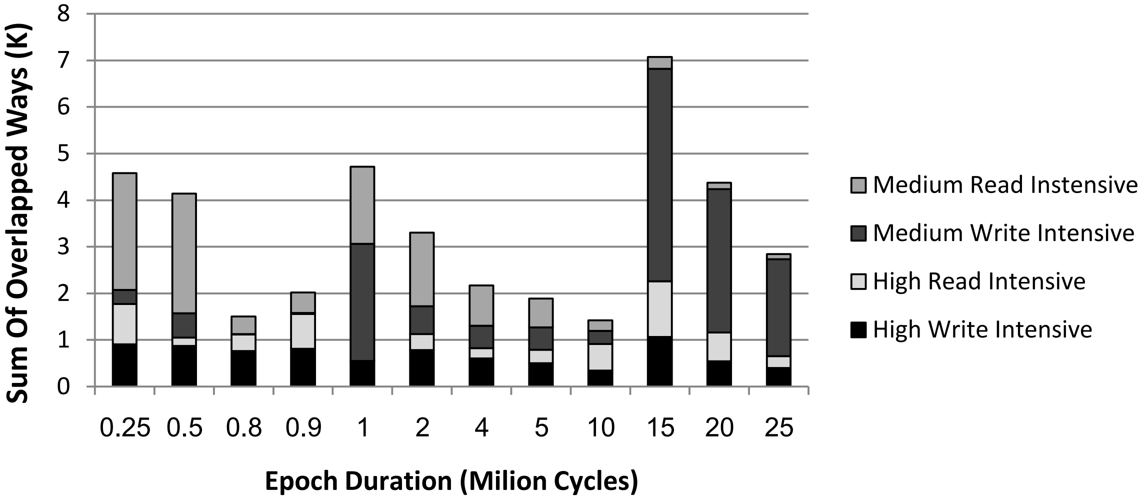
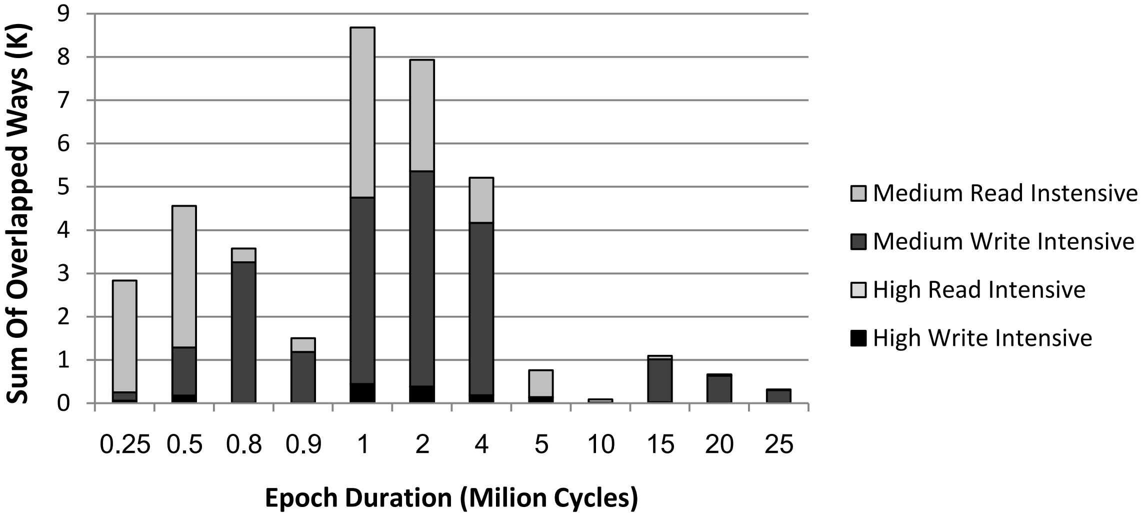


Figure 3: non-uniformity for writes within STT-RAM lines of the same set

During an execution interval of a program, the writes into sets are usually concentrated into some few lines which can vary as its write working set changes. considering the actual non-uniformity associated with write accesses into different cache sets and also within exact same cache set, we've got proposed a new dara migration process to boost utilization of minimal retention time STTRAM assortment(array) and as a consequence (and so) dynamically redirect(remapping) the majority of the(most of the) write accesses from higher retention time STTRAM arrays from the exact same set or a different set, (straight) into high speed under-utilized STTRAM array. Then using some management policies, we target to have an ideal cache for applications with various working set size which dissipates less energy, has comparable performance with current SRAM cache and therefore increase the cache response time to write accesses. So Write intensive blocks are primarily allocated from way 0 for a faster write response.





We first conduct circuit-level simulation to extract timing and energy of the proposed and baseline architectures. The STT-RAM cache uses almost the same interface and peripheral logic as the SRAM. Due to these similarities, we use CACTI tool [8] to extract latency and energy values for peripheral logic (at 45nm technology) added by the scaled latency and energy consumption of the STT-RAM model reported in [9]. SRAM parameters for baseline or hybrid structure are also driven from the CACTI model.

4. WRITE MANAGEMENT

During an execution interval of a program, the writes into sets are usually concentrated into some few lines which can vary as its write working set changes. considering the actual non-uniformity associated with write accesses into different cache sets and also within exact same cache set, we've got proposed a new dara migration process to boost utilization of minimal retention time STTRAM array. Then using some management policies, we target to have an ideal cache for applications with various working set size which dissipates less energy, has comparable performance with current SRAM cache and therefore increase the cache response time to write accesses.

The management mechanisms are arranged into two categories: intra-set and inter-set redirection (remapping). By moving data of a working set within a cache set, intra-set policies tries increasing the write-utilization of low retention STT-RAM lines.inter-set policies considering non-uniformity of write requests across various sets of a cache, to dynamically merge a set which has super write-intensive high retention STT-RAM lines (with large write working set, WWS) with one set which has temporary under-utilized low retention line. Hence, frequently-written data from high retention STT-RAM blocks of the former can now move to low retention STT-RAM block of the latter.

4.1. Intra-set Redirection

During a program execution interval, there are usually lines of a set whose contents are frequently rewritten, while the situation in other line is exactly opposite. Considering such access locality of write references, During various phases of an application execution, we desire to maintain write-stressed data into the low retention STT-RAM array within each set; while high retention STT-RAM blocks are mainly used to keep remaining infrequently-written data. Hence, a swapping has to be occurred when set's current high retention STT-RAM lines recently receives large writes.

The first issue to decide on is how to detect write-intensive data blocks (currently forming WWS of a set) based on which it should be placed in low retention STT-RAM line. to count writes in any line, We use a saturating counter named Line Saturation Counter, LSC, to monitor and register the temporal locality of recent writes within a set. The LSCs of a set determines how recent writes were distributed based on which write-stressed lines are selected (when saturated) and placed in low retention STT-RAM line. Saturation counter is incremented when a line gets a write request, and a line is considered as write-intensive as its LSC saturates. So, a concrete range for the LSC, from which a block is considered as write-intensive one during a program execution interval, is an important parameter that has to be determined. another challenge in determining a suitable time duration for 2 proposed data migration mechanism. Therefor we find an optimum execution time interval based on the write access patterns in lines of a set, looking for the best time stamps based on which write-stressed lines are selected (when saturated) and redirection to low retention STT-RAM line is trigered. when the time interval starts, if a high retention STT-RAM counter gets saturated, a data swap between high retention and low retention STT-RAM lines within set can be initiated.

Thus counter value of low retention STT-RAM line, is compared with the saturated value of write-stressed high retention STT-RAM line. If the SRAM line is more recently written and has larger LSC value, no swapping between SRAM and STT-RAM is occurred. Otherwise, the candidate data in the high retention STT-RAM line is written into the low retention STT-RAM line and the content of the low retention STT-RAM line is moved to the write buffer for future write into the high retention STT-RAM line. At the end of a program execution interval, Either a swapping is occurred or not, the LSC of all lines in the set is reset to dynamically track relative write requests to the lines, now on.

4.2. Inter-set Data Redirection (Remapping)

This non-uniform write accesses between sets results in fast wear-out of some sets than some others. To increase cache lifetime and write utilization of low retention STT-RAM array, it is worthwhile to balance write accesses among the sets by reallocating some of those high retention STT-RAM lines in the write-stressed sets onto the low retention STT-RAM line of the under-utilized ones. Simply stated, some mechanism is required to increase set association and remap the hot data of the former set to low retention STT-RAM line of the latter.

In our proposal to measure the write intensity of any cache set we consider existence of second saturated high retention line in current time interval in each set to initiate a inter-set data redirection procedure. When a merge is triggered, the saturation counter of low retention STT-RAM line of the set is used to determine which set can accept hot data from current set.

Any merge policy should address four issues: 1) which sets can receive write-stressed lines of a merge-requesting set, 2) how to perform merge and replace data, 3) how to search the cache for a requested line in the presence of merge, 4) when and how to break a previously established merge.

4.2.1. Destination set selection algorithm.

The destination of enlarging set association can be selected either statically or dynamically. In the static merge, the index of redirection destination set is a function of the requesting set's index [10]. Such static scheme incurs minimum overhead but is not largely-efficient. we use the semi-dynamic approach which increases the chance of static merging by increasing the number of candidate destination set of an association.

In our proposal, each 8 sets with difference in 3 most significant bits of the tag index form a merge group. So, sets that can be merged are physically far enough to decrease effect of locality of references. If there is a second line with saturated counter in high retention STT-RAM lines from start of time interval, our algorithm picks its redirection destination set from the merge group which it belongs to. Besides, a set may hold either low retention STT-RAM line that correspond to itself (native lines) or the lines that are displaced from high retention STT-RAM lines. To solve this ambiguity, each low retention STT-RAM line has an identification bit , I. This bit takes "1", if its data block is native to the low retention STT-RAM block and gets "0", if it is displaced from high retention STT-RAM lines of current set or another set. Therefor to select best destination set, a comparator determines minimum saturation counter value of low retention line of all sets in that merge group which has low retention STT-RAM line with an identification bit I=1(this means that there is no saturated block in high retention lines of the candidate set in current time interval). Only If the difference between saturation counter of the high retention line from requesting target set and saturation counter value of the destination candidate set is more than a threshold, data replacement algorithm is triggered.

Note, the destination set selection algorithm only evaluates sets in the same merge group that are not now merging with some others and has low retention STT-RAM line with an identification bit I=1.

4.2.2. Data replacement algorithm.

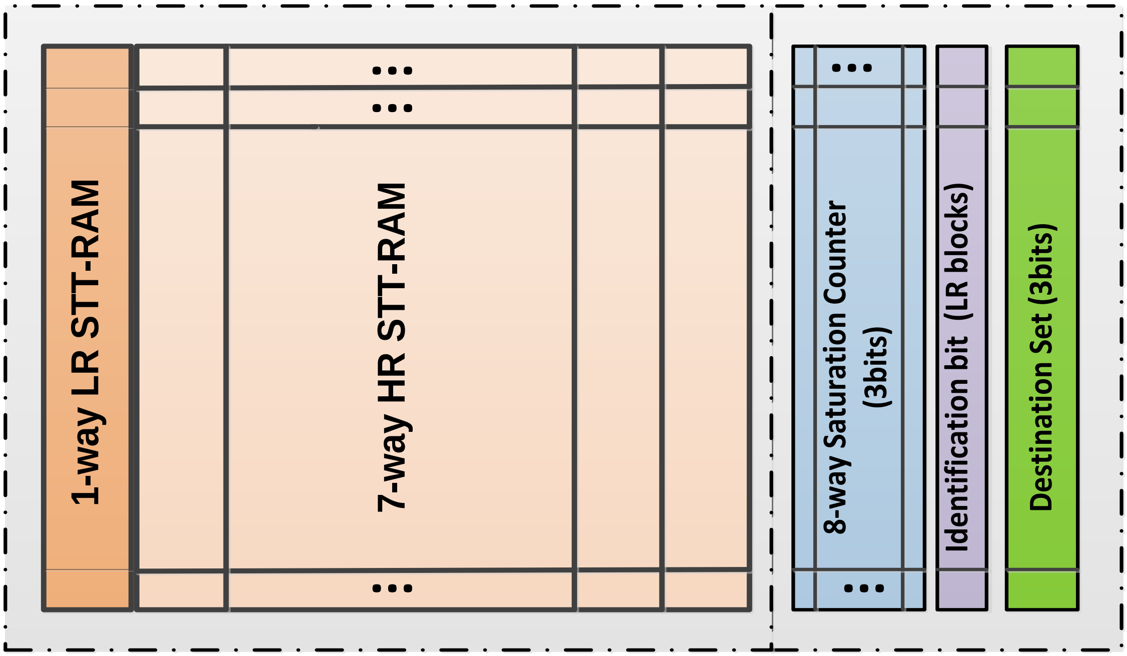
Each set is corresponds to a 3-bit merge destination (MD) storage which saves 3 most significant bits of another set for a merge. Initially, the MD contains 3 most significant bits of the set itself that implies no merge. So 3-bit merge destination determines that low retention STT-RAM line from each set hold either memory lines that correspond to itself (native lines) or the line that is displaced from its coupled. To establish a merge, the MD of the target set and destination set are exchanged and form a coupled set. From now then, any hot data at high retention STT-RAM line in the target set can move to low retention STT-RAM line of the destination set. In other words, low retention STT-RAM line of the destination set help filtering writes to high retention STT-RAM lines of the target. We limit maximum replacement of the lines from a set to another by number of low retention STT-RAM line at each set.

4.3. Cache search algorithm.

When a request is directed to a set, tag equality with its native lines are checked. If not found, looking at the MD of the set; it decides to search another set (if MD content differs from its index) or not. The set index for the secondary search is a direct function of current set's index and its MD where tag equality is checked for low retention STT-RAM line. In the case of no tag equality, the LRU policy is followed to bring the requested data into the native cache set.

4.4. Breaking merge algorithm.

During program execution, WWS of the sets varies which may decrease the merge efficiency. Indeed, inefficient established merges have two negative effects: they require search in the coupled set, if tag equality fails at the native set. Besides, they can block some potential merges due to lack of idle sets in a merge group. we may break a merge when high retention STT-RAM lines of the destination set becomes write-intensive and their counters get saturated. The displaced data is written back to the input write buffer and MDs of the coupled sets are exchanged.


1-way LR STT-RAM
7-way HR STT-RAM
8-way Saturation Counter (3bits)
Identification bit (LR blocks)
Destination Set (3bits)

Refrences

[1] H. Naeimi et al., “STTRAM Scaling and Retention Failure”, intel technology journal, volume 17, issue 1,2013.

[2] C.W. Smullen, et al., “Relaxing Non-Volatility for Fast and Energy-Efficient STT-RAM Caches”, *in Proc. of the International Symposium on High Performance Computer Architecture (HPCA),* pp. 50-61, 2011.

[3] Z. Sun, et al., “Multi retention level STT-RAM cache designs with a dynamic refresh scheme.” *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture(MICRO),* pp. 329-338, 2011.

[4] A.Nigam, et al., "Delivering on the promise of universal memory for spin-transfer torque RAM (STT-RAM)," *Proceedings of the International Symposium on Low Power Electronics and Design (ISLPED),* pp.121,126, 2011.

[5] Adwait Jog, Asit K. Mishra, Cong Xu, Yuan Xie, Vijaykrishnan Narayanan, Ravishankar Iyer, Chita R. Das, “Cache revive: architecting volatile STT-RAM caches for enhanced performance in CMPs,” *Proceedings of the Design Automation Conference (DAC),* pp. 243-252, 2012.

[6] P. Zhou, et al., “Energy Reduction for STT-RAM Using Early Write Termination,” *in Proc. of the International Conference on Computer-Aided Design (ICCAD)*, pp. 264-268, 2009.

[7] Z. Diao et al. Spin-transfer Torque Switching in Magnetic Tunnel Junctions and Spin-transfer Torque Random Access Memory. Journal of Physics: Condensed Matter, 19, 2007.

[8] CACTI: An Integrated Cache and Memory Access Time, Cycle Time, Area, Leakage, and Dynamic Power Model, Ver. 5.3, Retrieved in June 2010 from http://www.hpl.hp.com/research/cacti/

[9] X. Dong, et al., “Circuit and Microarchitecture Evaluation of 3D Stacking Magnetic RAM (MRAM) as a Universal Memory Replacement,” *in Proc. of the Design Automation Conference (DAC),* pp. 554-559, 2008.

[10] C. Bienia and K. Li, “PARSEC 2.0: A New Benchmark Suite for Chip-Multiprocessors,” in *Proceedings of Annual Workshop on Modeling, Benchmarking and Simulation (MoBS),* 2009.

[11] SPEC: Standard Performance Evaluation Cooperation. CPU 2006 Benchmark. http://www.specbench. org/